NXTware Remote

Advanced Development and Maintenance Environment for OpenVMS and other Strategic Platforms

Gerrit Woertman
CTO OpenVMS
Business Generating Software
gerrit@bgsoftware.nl
+31 6 51341600
Introduction to NXTware Remote

• Why Modernize Development?

• What is NXTware Remote
  – Advanced tools
  – Rich integration
  – Powerful Debugger

• Benefits of NXTware Remote
Classic OpenVMS program development

• Terminals session to VMS
• Edit program source
• Compile, link, run
• Later DECset with LSE, CMS, MMS
• Or develop on desktop/laptop and use extensively ftp to copy sources both ways
Why modernize your development environment?

• Enhanced Results
  • Improved Quality
  • Increased Productivity
  • Lower Costs

• Enriched Teams
  • Establishes best practices
  • Enrich current capabilities
  • Simplifies onboarding new developers

• Agile Organization
  • Improves development-operations (Dev-ops)
  • Provides great organizational adaptability
The components of a modern development Environment?

• **Primary**
  - Modern integrated development environment (IDE)
    - Powerful editors
    - Integrated graphical debugger
    - Integrated source code management
    - Open framework for adding tools

• **Best practices**

• **Contributing**
  - Task/Requirement management
  - Dev-Ops automation between
    - Source code repository
    - Build
    - Testing
What is NXTware Remote?

- An integrated development/maintenance environment that
  - Increases software engineering productivity
  - Updates and modernizes the development process
  - Easily integrates external tools into the dev process
- Supports best practices and improves code quality
- Provides a
  - Remote development environment
  - Advanced software engineering tools
  - Powerful Debugger
NXTware Remote Features

- Eclipse-based IDE
  - Native HP Language Editors
  - Full integration/certification of 3rd party Eclipse plug-ins

- Remote Development
  - From: Windows, LINUX, MacOS
  - To: OpenVMS, LINUX, UNIX

- Language Support
  - C/C++, COBOL, PASCAL, FORTRAN, BASIC
  - DCL, SQL, various scripting languages

- Modular Extensible
  - IDE and Remote Execution
  - MMS, CMS, SVN, etc
  - 3rd Party plug-ins
NXTware-Remote

OpenVMS

NXTware-Remote

COBOL

C

C++

FORTRAN

PASCAL

BASIC

DCL

JAVA

HTTP, Manager, Database

eclipse

NXTware-Remote plugins
Remote Development Architecture

Remote Communication

WORKSTATION
Windows
LINUX
MacOS

REMOTE SERVER
OpenVMS
LINUX
UNIX

ECLIPSE
NXTWARE REMOTE STUDIO
OPENVMS TOOLS (IDE)

NXTWARE REMOTE SERVER
OPEN VMS

OpenVMS Cluster
VAX
Output
Remote Development

- **Remote architecture**
  - NXTware Remote Studio
    - Eclipse plugin
  - NXTware Remote Server
    - Remote execution engine
  - NXTware Server
    - Cross-platform Communication platform

- **Remote Operation**
  - Studio enables users retrieve and edit code the transfer to a remote server for operations
  - On remote server the source code is built, complied, run and debugged from within Eclipse
Advanced Tools

• Rich Editor Features
  – Easy Navigation
  – Syntax Highlighting
  – Content Assist
  – Mirroring and Synchronizing Files
  – Source Code Templates
  – Macros and Shortcuts
  – Search & Replace
  – Bookmarks
Rich Integration: OpenVMS Centric

- **Oracle RDB**
  (Relational Database Management System)
  integration

- **HP CDD**
  (Common Data Dictionary)
  • CDO, BLDCDO

- **HP ACMS**
  (Application Control Management System)
  • GDF, TDF, IFDL, CDO

- **HP RMS**
  (Record Management Services)
  integration

- **CMS, NXTware Remote**
  advanced integration with CMS
Rich Integration: CMS File Diff
Powerful Debugger: Managing values

Video Demo: https://www.youtube.com/watch?v=3icNPjKjtXg
Rich Integration 3rd Party Plugin and Integration

- Database Management
  - Dbeaver for Database integration (SQL and RDB support)
- Source Code Management Support
  - Quality Center, Subversion, Git, Microsoft Team Foundation, etc
- Requirement and Task Management
  - Mylyn Task Management (Quality Center support)
  - Custom integration
- Quality/Check Style
  - ACRT - Automatic Code Review Tool for COBOL
- Custom Integration
  - Jenkins Build Automation
NXTware Remote and Jenkins

• Brings Continuous integration and Build Automation to OpenVMS
  – NXTware Remote Server acts as a Jenkins proxy on OpenVMS
• NXTware Remote for Jenkins allows developers to
  – Detect build breaks sooner
  – Identify failing tests more quickly
  – Make progress faster
The Modern Development Components

• Primary components
  ✓ Modern integrated development environment (IDE)
    – Powerful editors
    – Open framework for adding capabilities
  ✓ Best practices for development engineering

• Valued-adding components
  ✓ Task/Requirement management

• **The Next Step:** Dev-Ops automation (Continuous Integration) between
  – Source code repository
  – Automated Build
  – Automated Testing
NXTware Remote Benefits

- Proven demonstrated benefits
  - Developers are
    - More productive
    - Generated fewer bugs
    - Created code that was easier to manage
  - Managers are able to better
    - Allocate resources
    - Track efforts
    - Forecast deliverables
  - End users receive
    - Better quality application updates faster
What can NXTware Remote do for you and your users?

- 30%–45% improvement in overall productivity
- 100%–300% improvement in productivity for specific tasks
- 50% decrease in bugs and errors that reach QA
- Overall reduction in the amount of time and energy needed to develop, build and test software
For More Information

Visit [www.ecubesystems.com](http://www.ecubesystems.com)
Watch the NXTware Remote [playlist](http://www.ecubesystems.com)
More Info: [gerrit@bgsoftware.nl](mailto:gerrit@bgsoftware.nl)
Cal Success Story

• Don’t just take our word for it: (bit.ly/1E0m7KN)

https://www.youtube.com/watch?v=1V0gm98Xj4g
NXTware Remote for Jenkins

Short overview of the benefits of Build Automation, Continuous Integration and NXTware Remote for Jenkins

Gerrit Woertman
CTO OpenVMS
Business Generating Software
gerrit@bgsoftware.nl
+31 6 51341600
What is build automation?

• Build automation is the process of automating tasks that software developers do in their day-to-day work including:
  – compiling computer source code into binary code
  – packaging binary code
  – running automated tests
  – deploying to production systems
  – creating documentation and/or release notes
Why Implement build automation?

• The benefits of build automation include
  – Improved product quality
  – Accelerated compile and link process
  – Elimination of redundant tasks
  – Minimization of bad and faulty builds
  – Documentation of build and release history to track issues
  – Removal of dependencies on specific personnel
  – Saved time and money - because of the reasons listed above.
What is Continuous Integration?

• Continuous Integration (CI) is a development process that
  
  – requires developers to integrate code into a shared repository several times a day
    
    • each check-in is then verified by an automated build
      – optionally followed by automated tests
  
  – allows teams to detect problems early
What is Continuous Integration?

Four Steps to Continuous Integration

1. Source Control
2. Build Automation
3. Test Integration
4. Report Commit
Why Jenkins?

• Popular Continuous Integration Server
  – Widely used
  – Extensible
  – Ease to use

• Multi-platform
  – Windows, Linux, Unix
  – But not OpenVMS
Why NXTware Remote and Jenkins for OpenVMS?

• Brings Continuous integration and Build Automation to OpenVMS
  – NXTware Remote Server acts as a Jenkins proxy on OpenVMS

• NXTware Remote for Jenkins allows developers to
  – Detect build breaks sooner
  – Identify failing tests more quickly
  – Make progress faster
NXTware Remote for Jenkins Architecture

Manager

Developers

Success or Failure Notification

Continuous Integration (CI) Server

NXTware Remote: Plugin

Windows
Linux

Source Control Server

Windows, Linux
OpenVMS

Git, SVN, CMS

Rules

Check Changes

NXTware Remote Managed Build/Test

Tort

*Each developer and their manager has access to Code Quality Tools (Sonar, PMD, etc.)
Jenkins with NXTware Remote Plugin

Jenkins Interface

NXTware Remote
Continuous Integration and Jenkins

Jenkins Interface: Returning results from an OpenVMS Build via NXTware Remote

Console Output

Started by user anonymous
Building in workspace C:\Users\Robert\jenkins\workspace\a
$CIL-I-SUPERSEDE, previous value of DCLPATH has been superseded
$CIL-I-SUPERSEDE, previous value of DECSSFS_CASE_PRESERVE has been superseded
$CMS-I-FETCHED, generation 3 of element DISH\USERS;\[NOVLE.cmc]AORDER,F fetched
SYSTEM job terminated at 26-ANS-2014 14:30:02.02

Accounting information:
Buffered I/O count: 156 Peak working set size: 27232
Direct I/O count: 88 Peak virtual size: 225488
Page faults: 2146 Mounted volume: 0
Charged CPU time: 0 00:00:00.13 Elapsed time: 0 00:00:00.44

Finished: SUCCESS
NXTware Remote for Jenkins Process

• CI Process on OpenVMS
  – Create a new job from a CMS repository
  – Build that code, see build results
  – Run tests, see test results
  – Make a change and watch it run through the system
NXTware Remote for Jenkins

**Before**
- Building is slow, error prone
- Testing is onerous
- Code coverage is onerous
- Bugs caught later
- Developers worry about servers
- No change control for deployments
- Slow progress
- Different artifact per environment
- Inconsistent configuration per environment
- Deployments are "hard"
- Integration difficult

**After**
- "fire and forget", consistent
- Testing is automated
- Code coverage is easy
- Bugs caught early and often
- Developers worry about code
- Change control in the right places for deployments
- Rapid progress. Greater agility.
- Identical artifact per environment
- Identical configuration per environments
- Deployments are “click” easy
- Integration simplified
The Modern Development Components

• Primary components
  ✓ Modern integrated development environment (IDE)
    – Powerful editors
    – Open framework for adding capabilities
  ✓ Best practices for development engineering

• Valued-adding components
  ✓ Task/Requirement management

• The Next Step: Dev-Ops automation (Continuous Integration) between
  – Source code repository
  – Automated Build
  – Automated Testing
For More Information

Visit [www.ecubesystems.com](http://www.ecubesystems.com)

More Info: [Pmarquez@ecubesystems.com](mailto:Pmarquez@ecubesystems.com) or [gerrit@bgsoftware.nl](mailto:gerrit@bgsoftware.nl)