



Program Verification Environment

Integrated tool set built on Eclipse IDE and JML infrastructure (JML4 is also built on Eclipse). Each component of MOBIUS is also a complete software package in itself e.g. ESC/Java2.











https://mobius.ucd.ie/wiki/Components







Search

freshmeat KindSoft SRG Trac Mobius - Trac UCD - Trac JML4 Travel Scholar InriaGforge JML Fortress Voting iGoogle Hudson

Mobius

logged in as dcochran Logout Settings Help/Guide About Trac My Notifications My Account

Wiki Timeline Roadmap Browse Source View Tickets New Ticket Search Tags Admin Peer Review

> Start Page Index by Title Index by Date | Last Change Watch Page

MOBIUS Components

Name	Languages Processed	Main Repository	Plugin Repository
BML Library	BML	⇒ src/BmlLib/trunk	⇒ src/BmlLib_plugin
BONc	BON	⇒ src/bon	
Coq Editor	Coq		
Simplify	Mobius VC AST → Simplify	⇒ src/simplify	⇒ src/simplify_plugin
Bico	BC → Bicolano BC (Coq)	⇒ src/bico	
DirectVCGen	Source + FOL → Mobius VC AST	⇒ src/escjava/trunk/MobiusDirectVCGen	
ССТ		⇒ src/CCT/trunk	
BML2BPL	BML → BPL	⇒ src/BML_BPL_Translator	
ETHZ JML2 Plugin	JML		
ESC/Java2	Java + JML → Mobius VC AST	⇒ src/escjava/trunk/ESCTools	⇒ src/escjava_plugin
Examples		⇒ src/examples	
Freeboogie	BPL → Mobius VC AST	⇒ src/freeboogie/trunk/FreeBoogie	
Javafe	Java + Bytecode → Mobius VC AST	⇒ src/Javafe	⇒ src/Javafe_plugin
JML2BML	JML → BML	⇒ src/Jml2Bml/trunk	
JML Class Library		⇒ src/Jmlclasslib	
JML Folding	JML		⇒ src/JML_folding_plugin
Logging		⇒ src/logging	
Plugin Library			⇒ libs/pluginlib
Prover Editor			⇒ src/provereditor_plugin
Umbra	Java Bytecode	⇒ src/Umbra/trunk	⇒ src/Umbra_plugin

Edit this page

Attach file

Delete this version

Delete page

Download in other formats:

Plain Text



Done

Powered by Trac 0.10.4 By Edgewall Software.

Visit the Trac open source project at http://trac.edgewall.org/



Architecture

- Languages: BON, BPL, ByteCode, BML,
 Coq, Java 1.4, Java 1.5, Java 6, JavaCard, JML...
- Actions: Static Analysis, Abstract Syntax Trees, Transformation...
- Interfaces: command line, front-end API, back-end (e.g. theorem provers), interactive plug-ins...

Interaction

- PVE components communicate with each other using a shared language and interface
- Example: Extended Static Checking

JavaFE: Java/JML → AST

ESC/Java2:AST → VC

Simplify: VC → Success/Fail

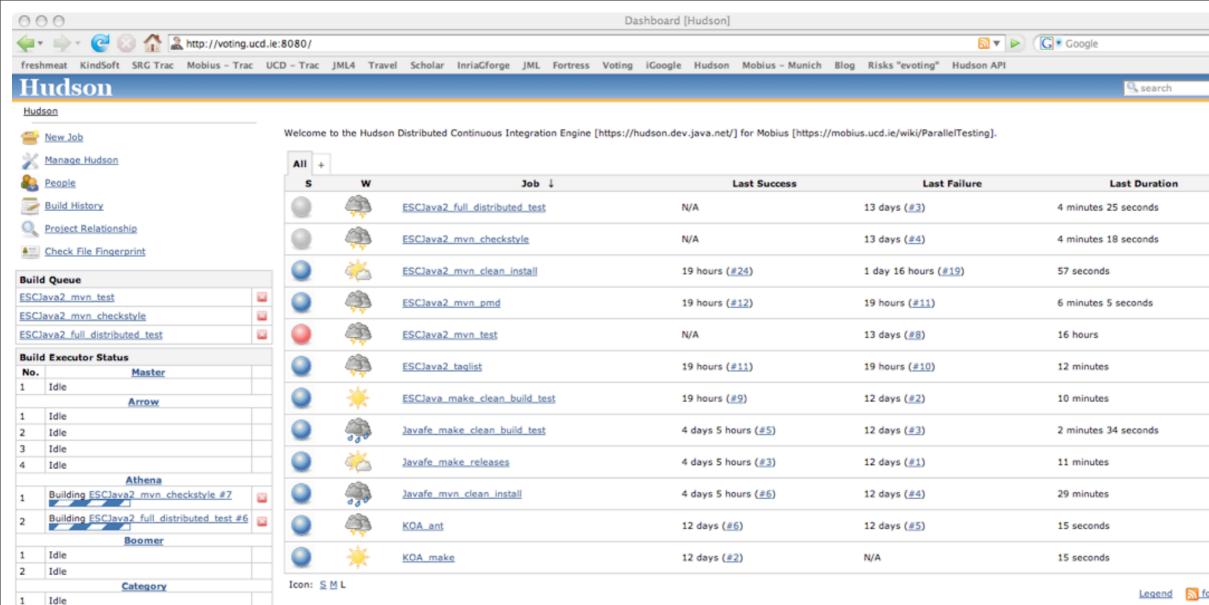
Logging: Success/Fail → Eclipse warnings

ESC plugin: adds markers to the code



Continuous Testing

- JUnit test suites or command line tests
- Maven Project Object Model (POM)
- Hudson Continuous Integration Engine
- http://voting.ucd.ie:8080/
- Build starts as soon as changes committed to subversion



Legend for all for failures

Q *

@ maven | logout
ENABLE AUTO REFRESH

edit description

2

(2)

2

2

(2)

Done

2

3

5

7

1

2

2

Idle

Idle Idle

Idle Idle

Idle

Idle

Idle

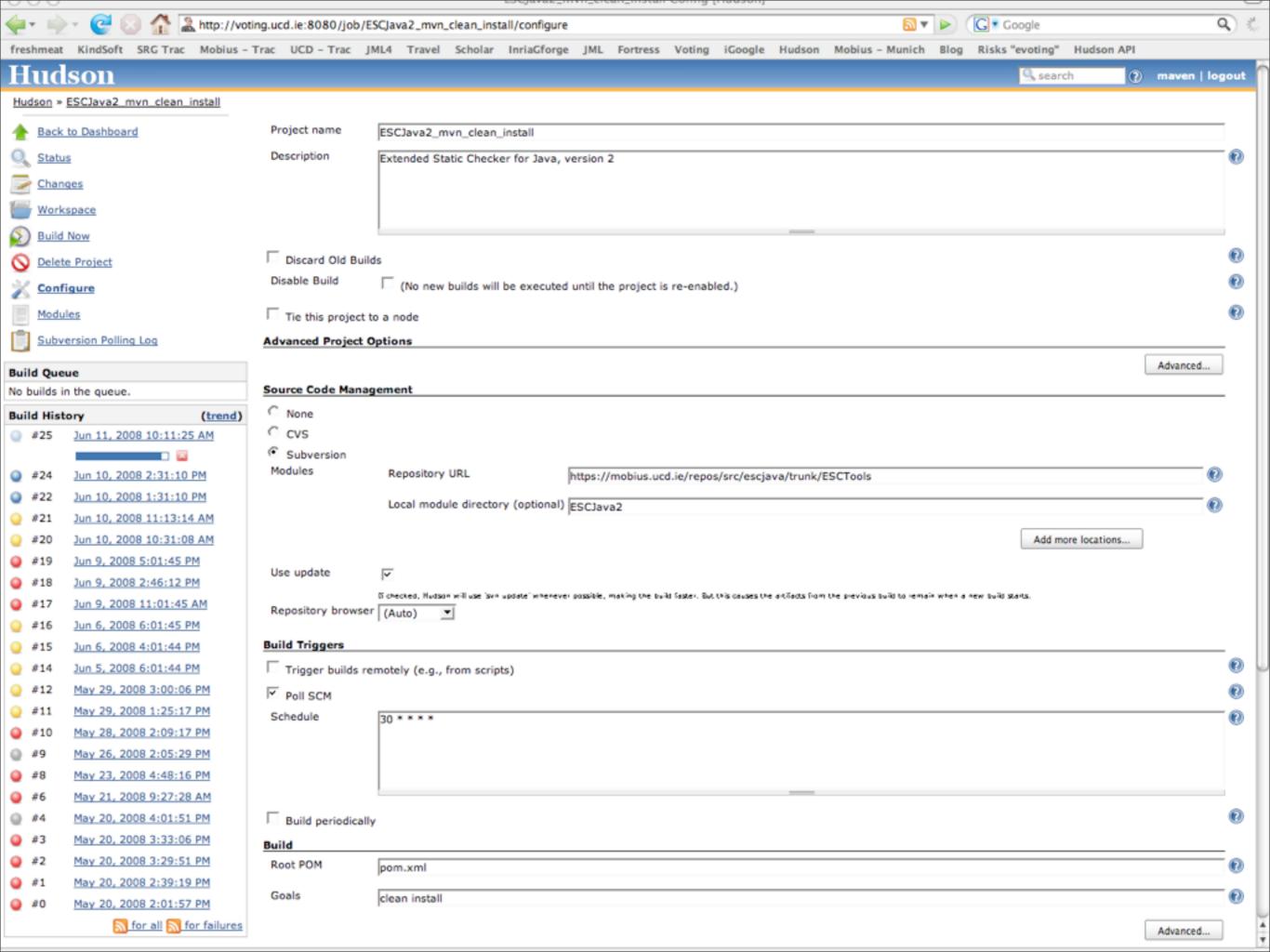
Idle

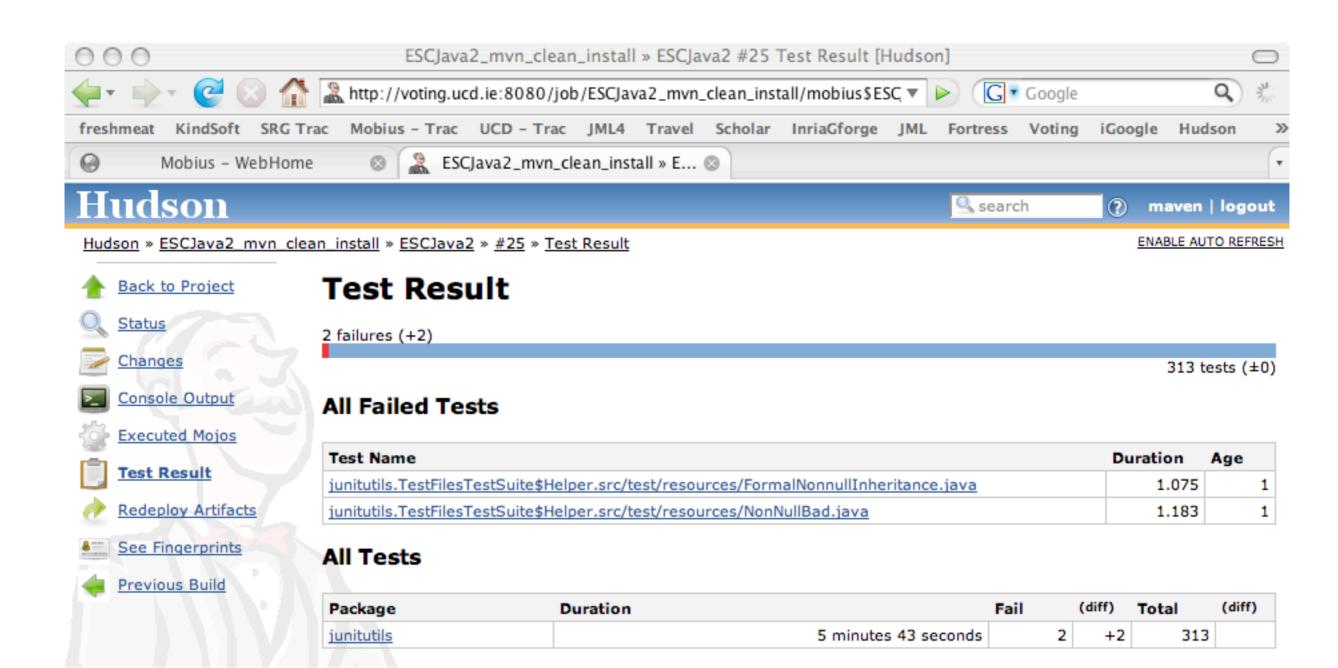
Idle Idle Logic

Morphism

Object

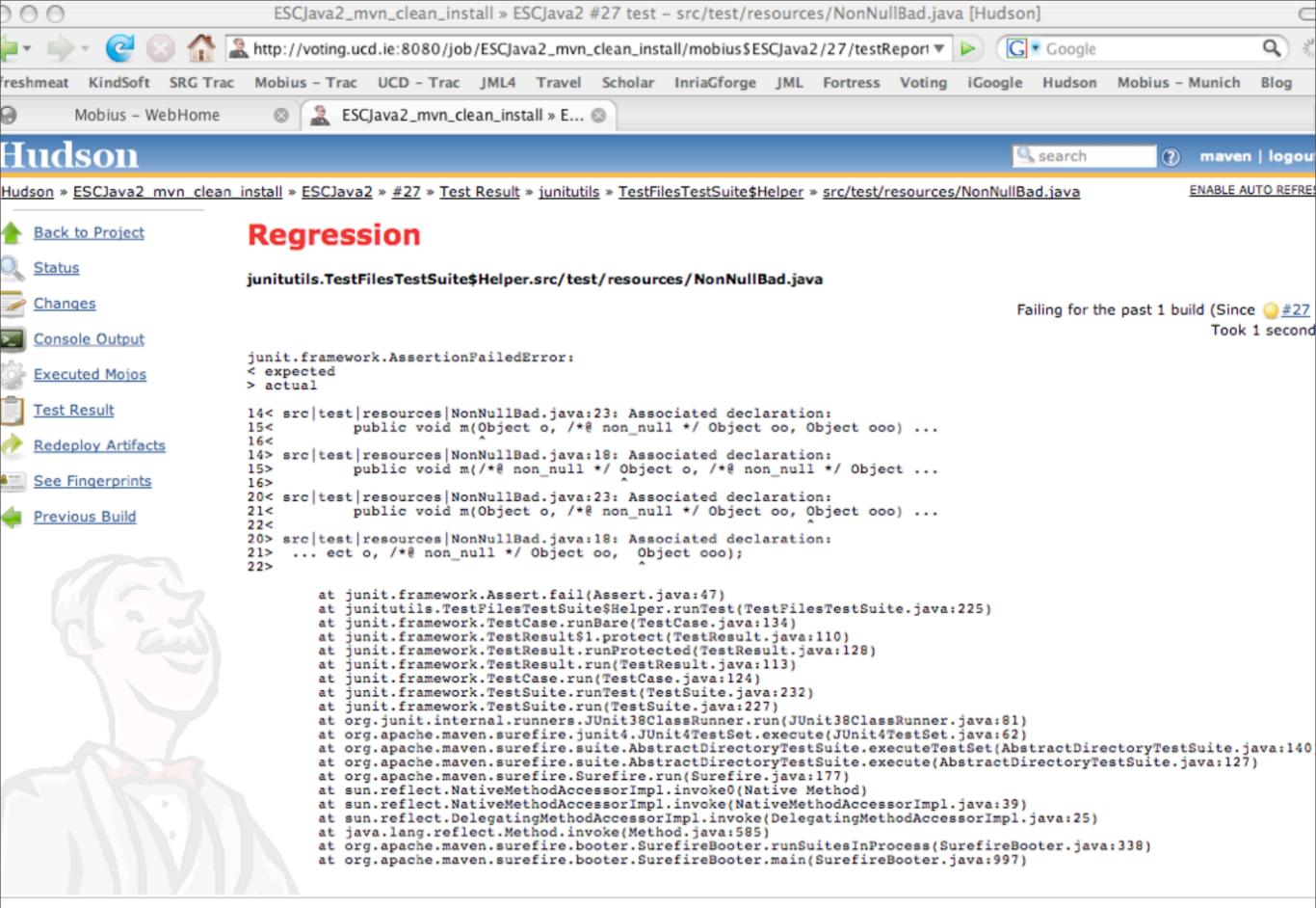
Building ESCJava2 mvn test #10





Hudson ver. 1.215

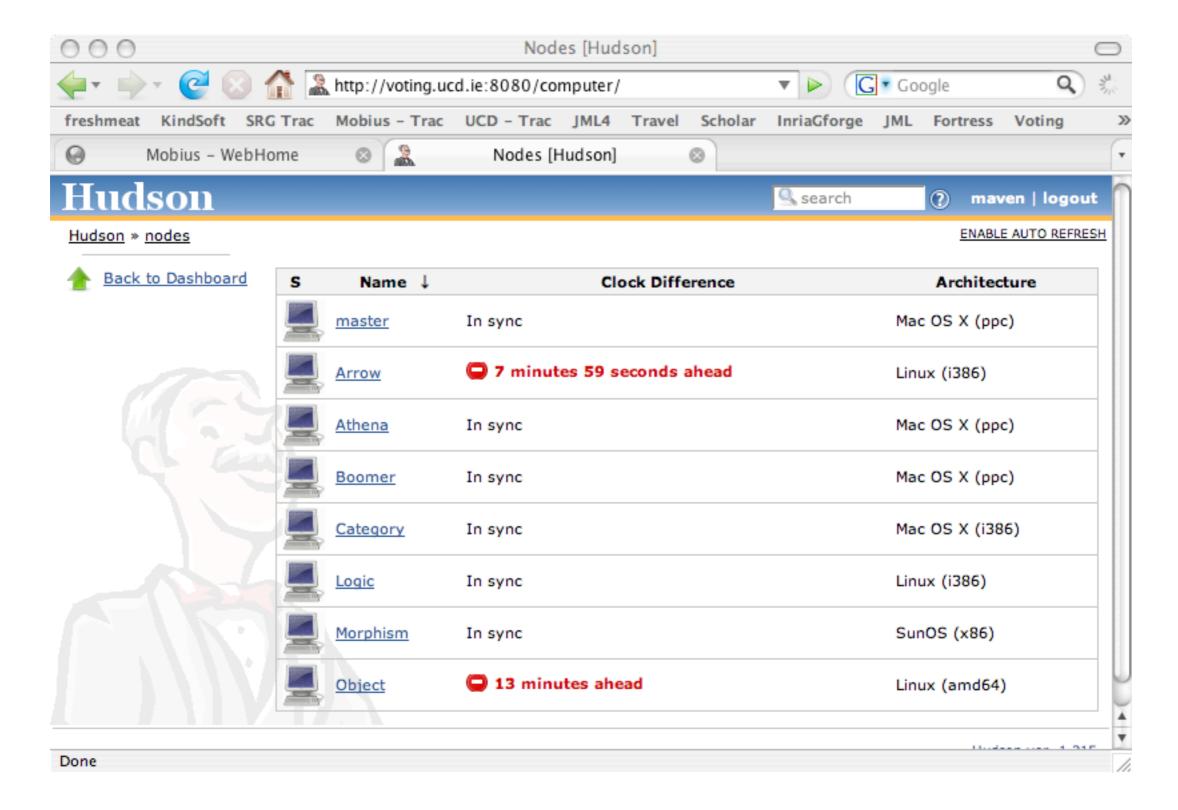
Done



Hudson ver. 1.215



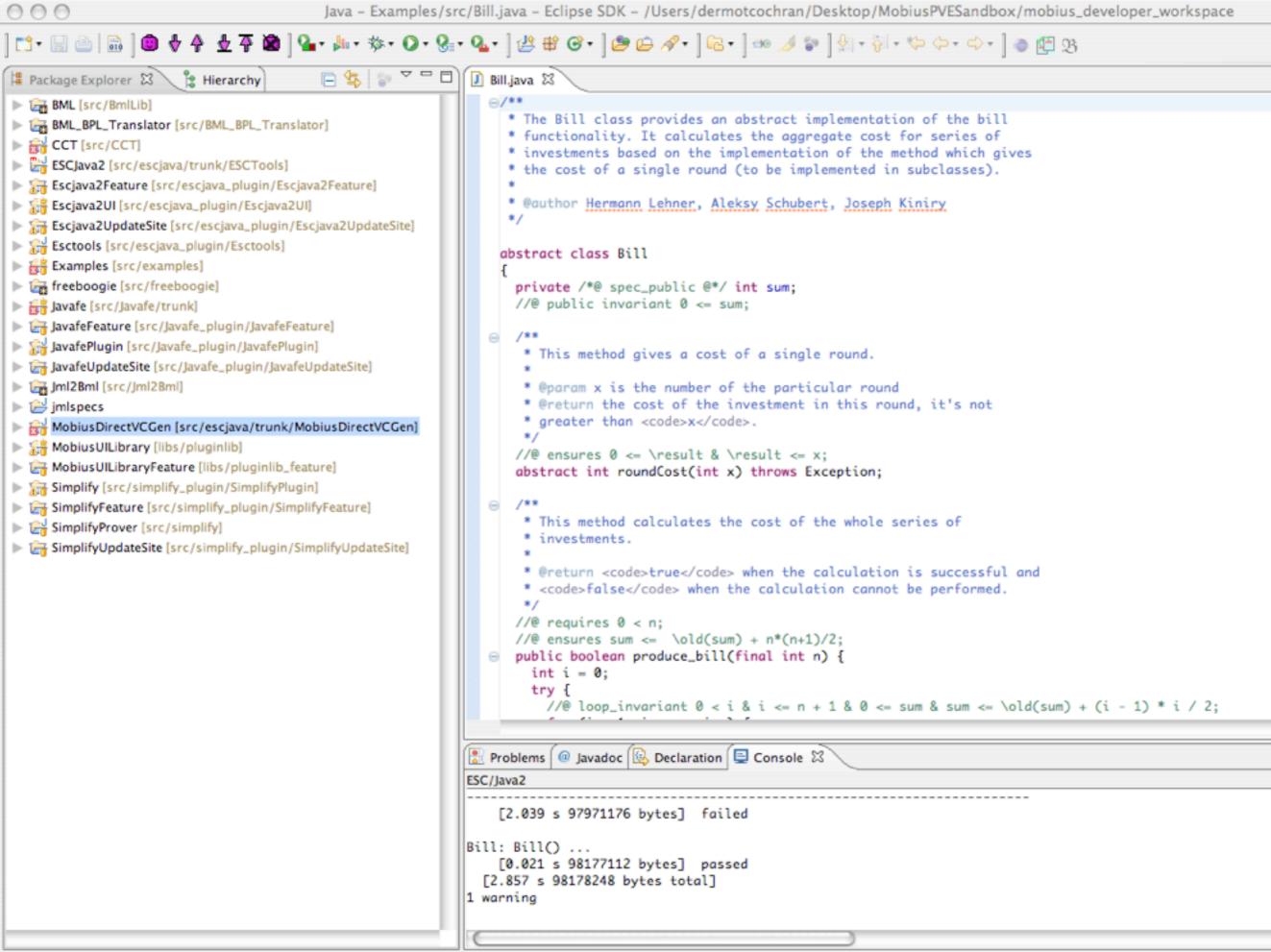
Test Environments





Release Modes

- Standalone command line tools
- Eclipse plugins
- MOBIUS PVE customization of Eclipse
 - Mac OS X (Universal)
 - Linux (32 bit)
 - Windows XP/Vista





Summary/Questions

- Components are standalone but also integrated with Eclipse and JML4
- What remains to be done?
- What support/help do you need to use/ develop/integrate with the PVE?
- Any other questions or issues?