National Cancer Institute

caBIG™ cancer Biomedical Informatics Grid ™

Automated Build and Deployment

CaArray 2 Build and Deployment pilot project

August 9, 2007

U.S. DEPARTMENT
OF HEALTH AND
HUMAN SERVICES

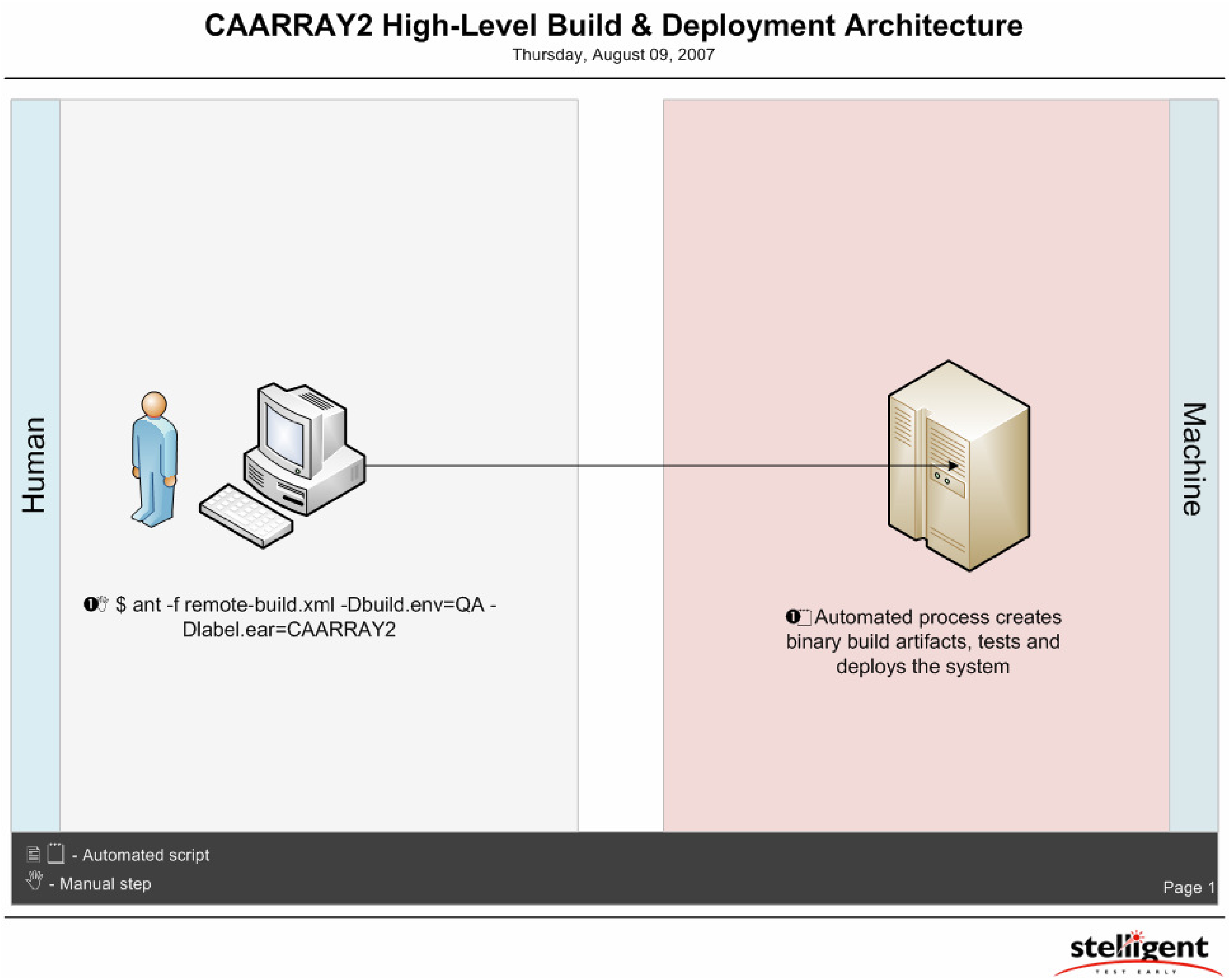
National Institutes
of Health

Goal of caArray 2 Pilot

Make Deployment the "click of a button"

- **What:** ***Speed up delivery time by decreasing manual processes***
  - Integrate configuration into the build process rather than requiring manual intervention
- **How:** ***Convert attributes (currently maintained in Word documents) into files that are machine-readable (an automated system)***
- **Why:** ***Enable caArray 2 to adapt to changes and deliver software faster with fewer deployment errors***

User Actions



CAARRAY2 High-Level Build & Deployment Architecture

Thursday, August 09, 2007

❶ ✋

```
$ ant -f remote-build.xml -Dbuild.env=QA -Dlabel.ear=CAARRAY2
```

❶ ▤ Automated process creates binary build artifacts, tests and deploys the system

▤ - Automated script

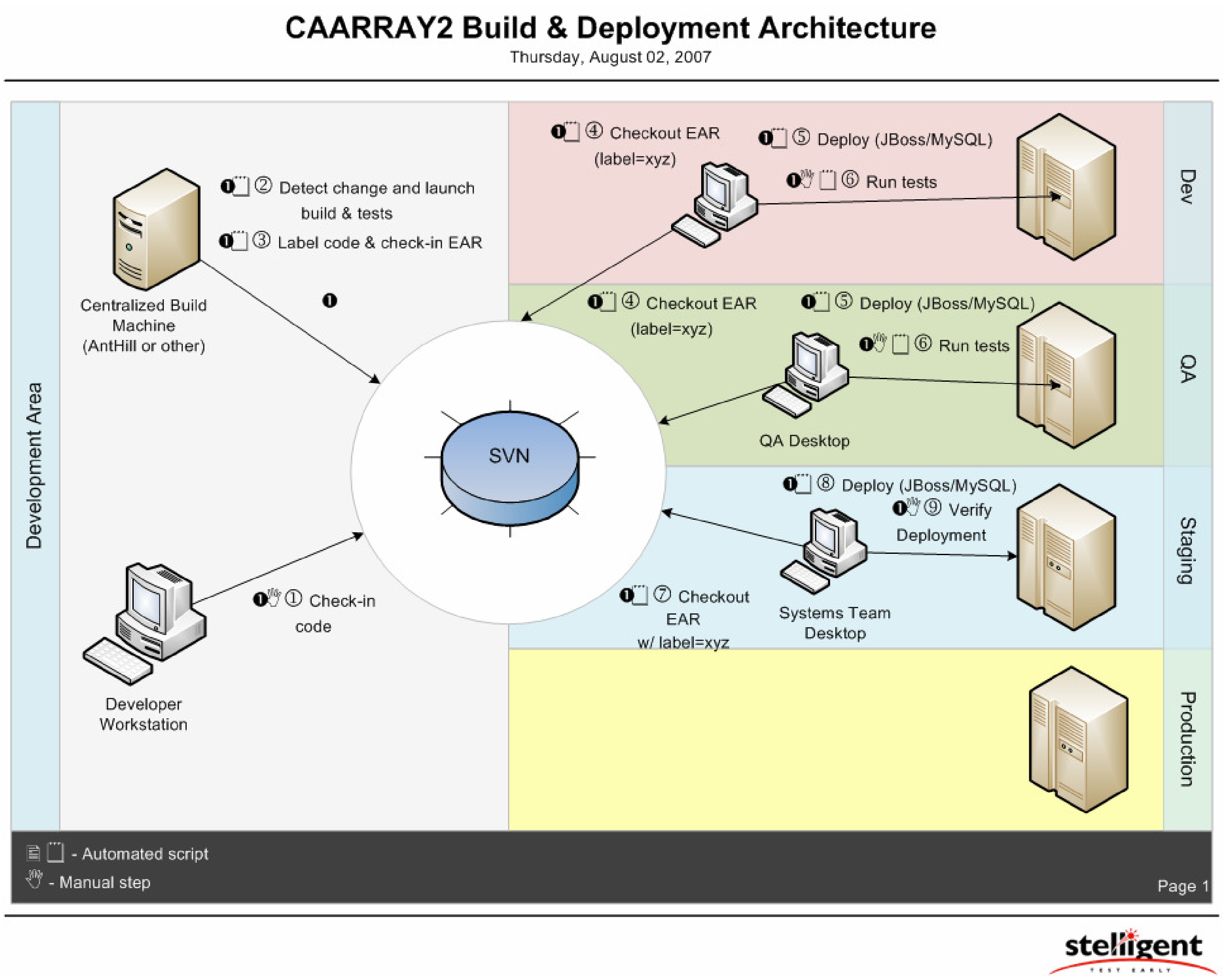
✋ - Manual step

Page 1

Benefits

- **More time testing. Less time troubleshooting error-prone deployment steps**
  - Redeploy in minutes (from a baseline configuration)
  - Development can remotely deploy to DEV environment in a single command
  - QA can remotely deploy to QA environment in a single command
  - MySQL database is updated remotely
  - JBoss is completely configured remotely
- **Single Source Repository**
  - All source code and application configuration is maintained in SVN
- **Maintain Security**
  - `caarray2-key.properties` file is publicly accessible, while the `dev.properties`, `qa.properties` and `staging.properties` can be restricted to authorized users or machines
- **Reduce wild-goose chases (i.e. Build system is king)**
  - All application/deployment configuration is managed through the build scripts
    - Not hard-coded values in the source code
    - Not vendor-specific XML files
    - This includes JBoss and MySQL
- **No duplication of resources**
  - Reuse of caArray's existing build scripts
  - Extensions written for remote deployment without changing existing build scripts

Build and Deployment Architecture for caArray 2 Pilot



CAARRAY2 Build & Deployment Architecture

Thursday, August 02, 2007

Development Area

Centralized Build Machine (AntHill or other)

② Detect change and launch build & tests

③ Label code & check-in EAR

❶

SVN

Developer Workstation

① Check-in code

Dev

④ Checkout EAR (label=xyz)

⑤ Deploy (JBoss/MySQL)

⑥ Run tests

QA

④ Checkout EAR (label=xyz)

⑤ Deploy (JBoss/MySQL)

⑥ Run tests

QA Desktop

Staging

⑧ Deploy (JBoss/MySQL)

⑨ Verify Deployment

⑦ Checkout EAR w/ label=xyz

Systems Team Desktop

Production

- Automated script

- Manual step

Page 1

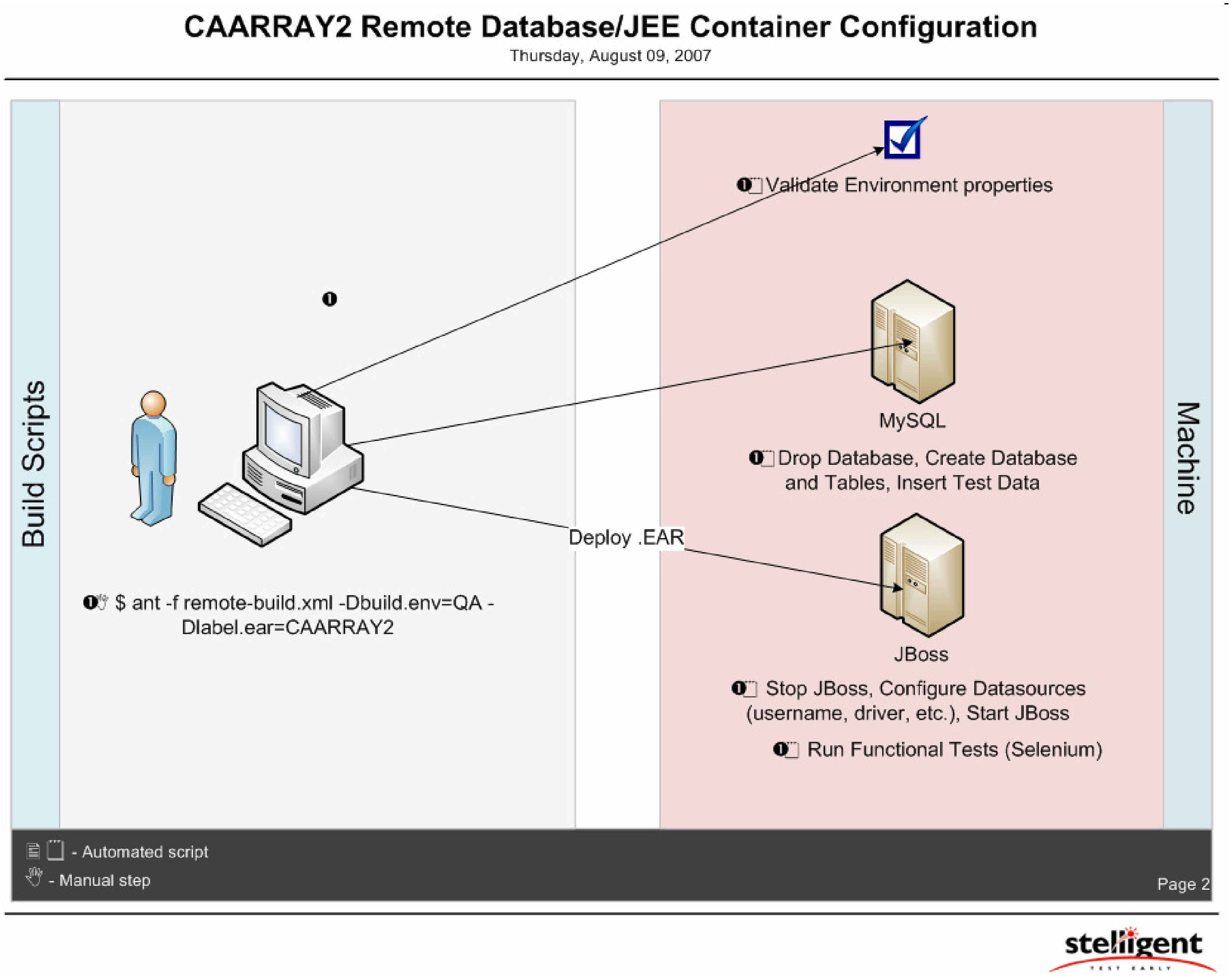
stelligent

TEST EARLY

caArray 2 – High-Level Build and Deployment Steps

1. **Developer checks-in code to SVN**
2. **From a centralized integration build machine, (similar to target environments) a developer runs a build and deployment in the DEV environment**
   1. `ant -f remote-build.xml -Dbuild.env=DEV`
   2. If the build is successful, the Ant script on the build server checks the EAR (or moves to managed directory) into SVN and labels the EAR
3. **QA runs an Ant script from the centralized build machine. This script checks out the packaged EAR file from SVN**
   1. `ant -f remote-build.xml -Dbuild.env=QA -Dlabel.ear=CAARRAY2`
   2. This Ant script configures the QA environment's JBoss and MySQL servers, runs the database integration scripts and deploys the EAR file to the JEE container (JBoss) and runs a suite of automated tests
4. **QA performs a suite of manual tests as necessary in the QA environment**

Remote Deployment Configuration



CAARRAY2 Remote Database/JEE Container Configuration

Thursday, August 09, 2007

Build Scripts

❶

❶ ✋

```
$ ant -f remote-build.xml -Dbuild.env=QA -Dlabel.ear=CAARRAY2
```

Deploy .EAR

Machine

❶ □ Validate Environment properties

MySQL

❶ □ Drop Database, Create Database and Tables, Insert Test Data

JBoss

❶ □ Stop JBoss, Configure Datasources (username, driver, etc.), Start JBoss

❶ □ Run Functional Tests (Selenium)

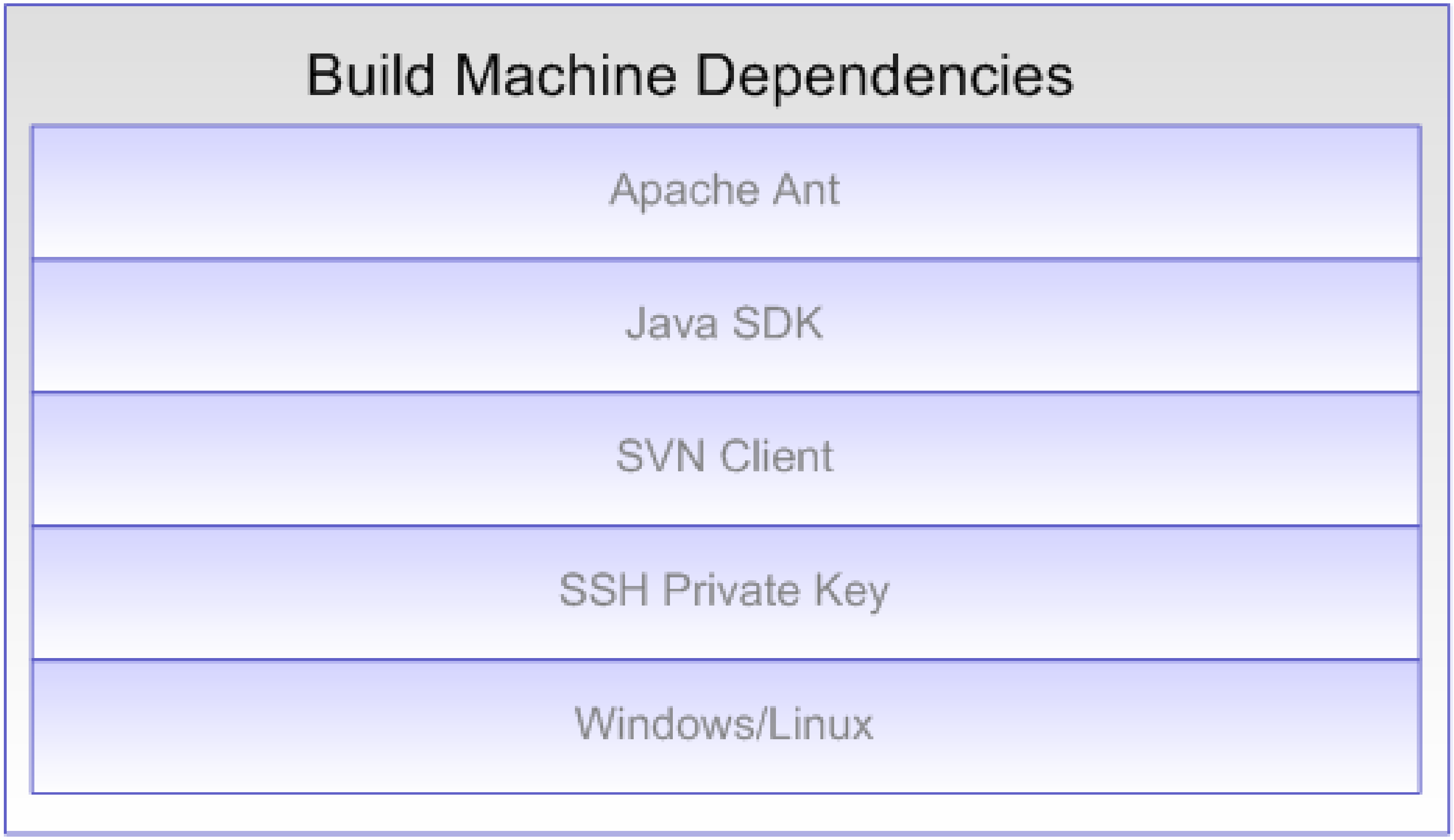
□ - Automated script

✋ - Manual step

Page 2

stelligent TEST EARLY

Build Machine Dependencies



Build Machine Dependencies

Apache Ant

Java SDK

SVN Client

SSH Private Key

Windows/Linux

Developer Scenario for caArray 2

1. **Developer commits code to SVN (ideally, at least once a day)**
2. **As necessary, a developer can run a remote build and deploy to the "DEV" environment**
    1. `ant -f remote-build.xml -Dbuild.env=DEV`
3. **When ready to test, a developer will notify QA to run a deployment by giving QA the Subversion label**
    1. For example, `CAARRAY2`

QA Scenario for caArray 2

1. **QA manages the `qa.properties` file for the caArray 2 project**
   1. This may be managed via SVN, file system or other. However, this file and these values should **not** be made available to public (or even development team?)
2. **QA runs an Ant script from the centralized build machine. This script effectively checks-out the packaged EAR file from SVN**
   1. `ant -f remote-build.xml -Dbuild.env=QA -Dlabel.ear=CAARRAY2`
3. **After successful deployment to the QA environment, QA may run their suite of automated and manual tests**

Questions

- **Managing [environment].properties files (dev.properties, qa.properties)**
  - Authorized SVN access?
  - File system? (to include an automated copy and remove to target environment)
- **SSH access**
  - Systems had a concern with tying user access to an individual
    - Potential Solution: restrict access from build machine

Additional Information

- **This process places more responsibility with the development teams**
  - Attributes must be extracted into simple `.properties` files
  - Build process is responsible for much of the application configuration (JBoss, MySQL, etc.)
- **Eliminates the manual copying and pasting that occurs from Word documents to AntHill/JBoss/Oracle, etc.**
  - This will reduce errors and lessen the time to get software into production