IBM Software Group

【SOA講堂】

Rational 動能開發-BuildForge-開發團隊大利器

SWG Rational Consultant
TommyWu (tommywu@tw.ibm.com)

Agenda

- What is Build Forge?
- What is Build Forge *Not*?
- Rational Build Forge Function Preview
- Technical System Overview & Usage Examples

What is Build Forge?

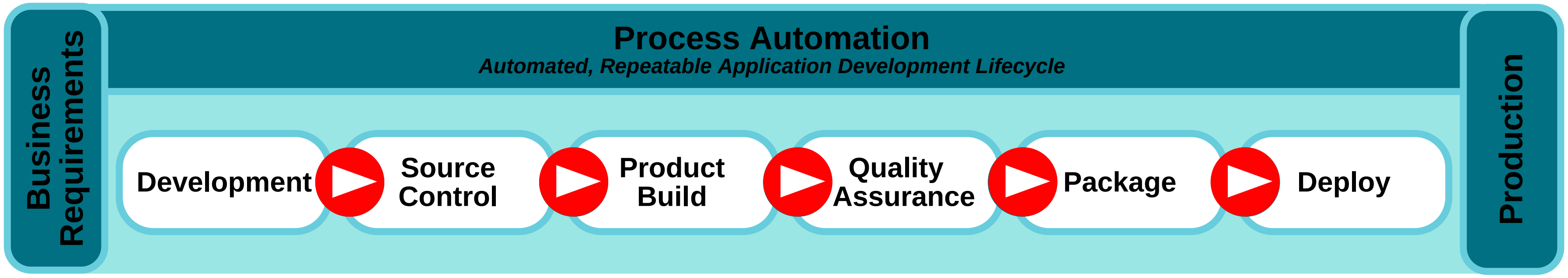
Automation has revolutionized industries such as Automotive Manufacturing, by allowing them to produce better products, faster, and at a lower cost.

Build Forge applies the same principles to the industry of Software Development…

What is Build Forge?

The "Elevator Pitch":

Build Forge is an adaptive process execution framework that automates, orchestrates, manages, tracks and logs all the processes between each handoff within in the assembly line of software development, creating an automated software factory.



- **Automates the creation of a deployable application from source artifacts**
- **Eliminates manual handoffs between code, build, test, and deploy teams**
- **Integrates with existing tools**
- **Aggregates results from multiple systems and processes**

What is Build Forge?

A "Closer to Home" Analogy:

Writing Checks
versus
Online Banking

Manually write checks
Manually pay bills
Manually balance account

Financial "framework"
Automated bill pay
Transaction scheduling
Automatic notifications
Transaction history
Interacts with multiple systems

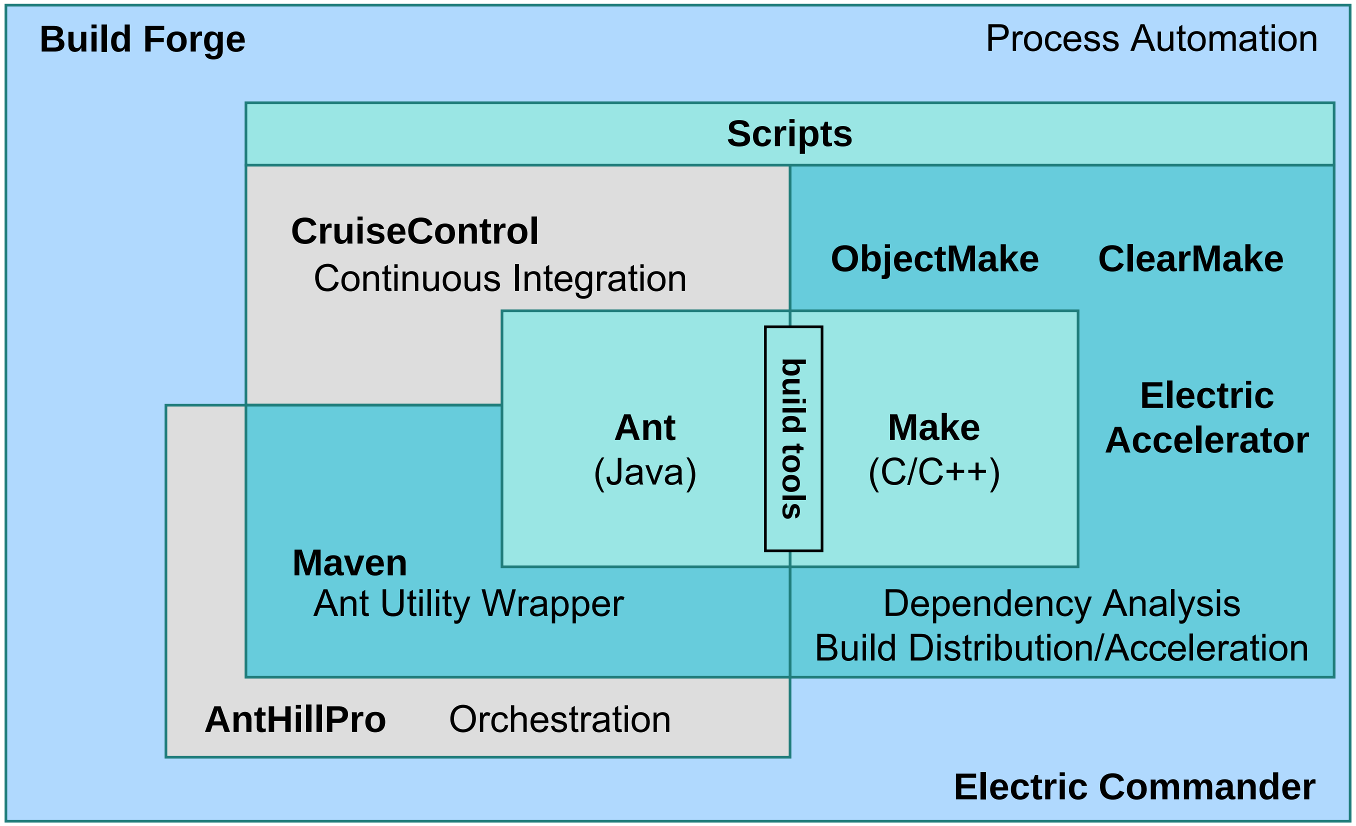
What Is Build Forge NOT?

- **Build Forge is NOT a Build Tool**
  - A build tool generates the executables and non-source files of a program from the program's source files.
  - A build tools is platform-specific and require specialized skills.
  - Tool examples include:

ANT (Java)	ClearMake (Rational gmake)
gmake (Unix C/C++)	Omake (Rational NMAKE)
NMAKE (Windows C/C++)	OpenMake
ObjectMake (Synergy make)	ElectricAccellerator

  - Advanced build tools may help with ***dependency analysis*** and build ***acceleration***.
- ***Don't let the name fool you!*** ***We often use the term "Build" for the whole assembly process…and your customers will too!***

Build Tools, CI, and Process Automation

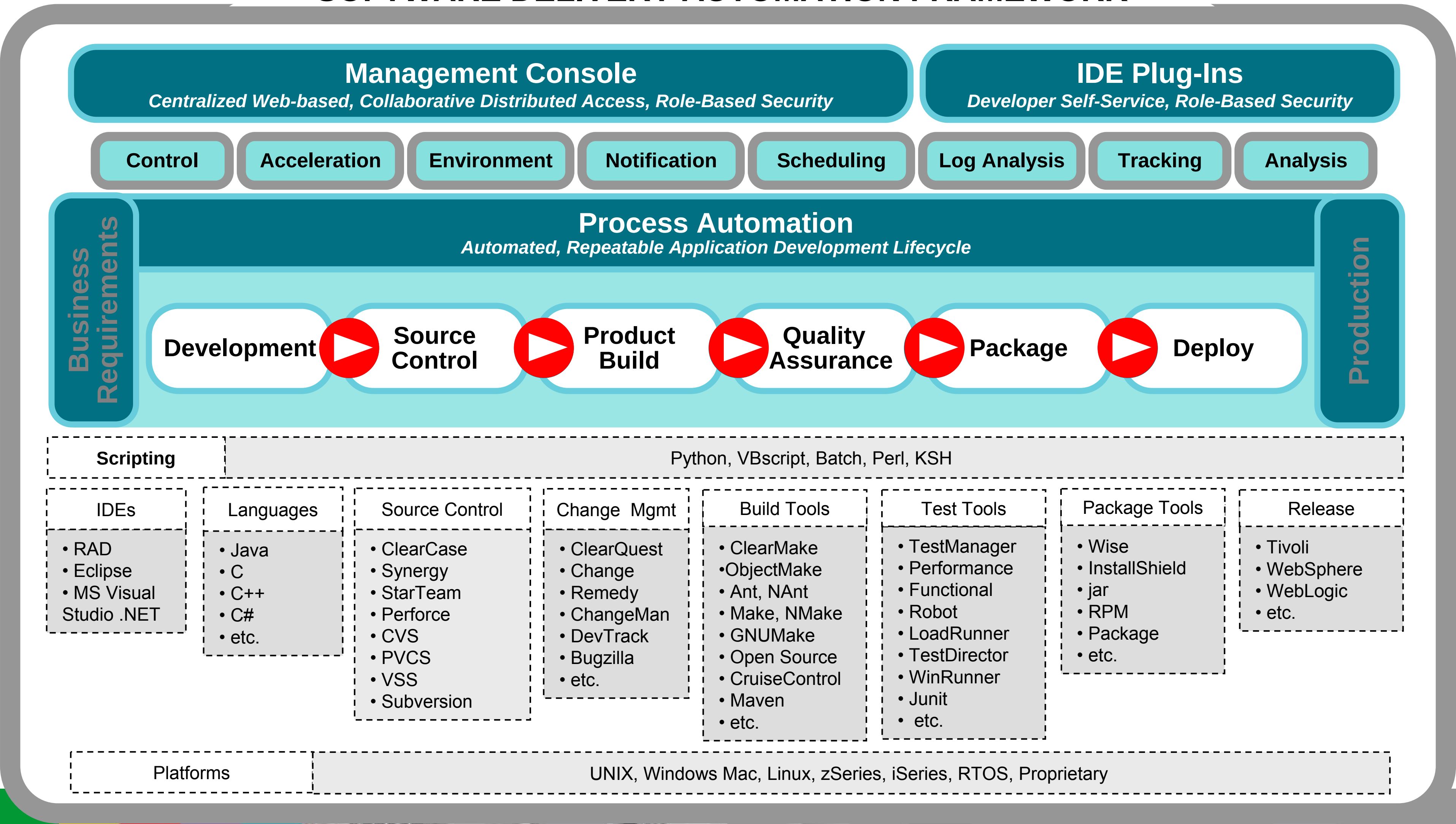


⬜ Replaced by Build Forge

🟦 Executed from Steps in Build Forge

Build Forge System Overview

SOFTWARE DELIVERY AUTOMATION FRAMEWORK



Management Console
Centralized Web-based, Collaborative Distributed Access, Role-Based Security

IDE Plug-Ins
Developer Self-Service, Role-Based Security

Control | **Acceleration** | **Environment** | **Notification** | **Scheduling** | **Log Analysis** | **Tracking** | **Analysis**

Business Requirements

Process Automation
Automated, Repeatable Application Development Lifecycle

Development ▶ **Source Control** ▶ **Product Build** ▶ **Quality Assurance** ▶ **Package** ▶ **Deploy**

Production

Scripting	Python, VBscript, Batch, Perl, KSH

IDEs
- RAD
- Eclipse
- MS Visual Studio .NET

Languages
- Java
- C
- C++
- C#
- etc.

Source Control
- ClearCase
- Synergy
- StarTeam
- Perforce
- CVS
- PVCS
- VSS
- Subversion

Change Mgmt
- ClearQuest
- Change
- Remedy
- ChangeMan
- DevTrack
- Bugzilla
- etc.

Build Tools
- ClearMake
- ObjectMake
- Ant, NAnt
- Make, NMake
- GNUMake
- Open Source
- CruiseControl
- Maven
- etc.

Test Tools
- TestManager
- Performance
- Functional
- Robot
- LoadRunner
- TestDirector
- WinRunner
- Junit
- etc.

Package Tools
- Wise
- InstallShield
- jar
- RPM
- Package
- etc.

Release
- Tivoli
- WebSphere
- WebLogic
- etc.

Platforms	UNIX, Windows Mac, Linux, zSeries, iSeries, RTOS, Proprietary

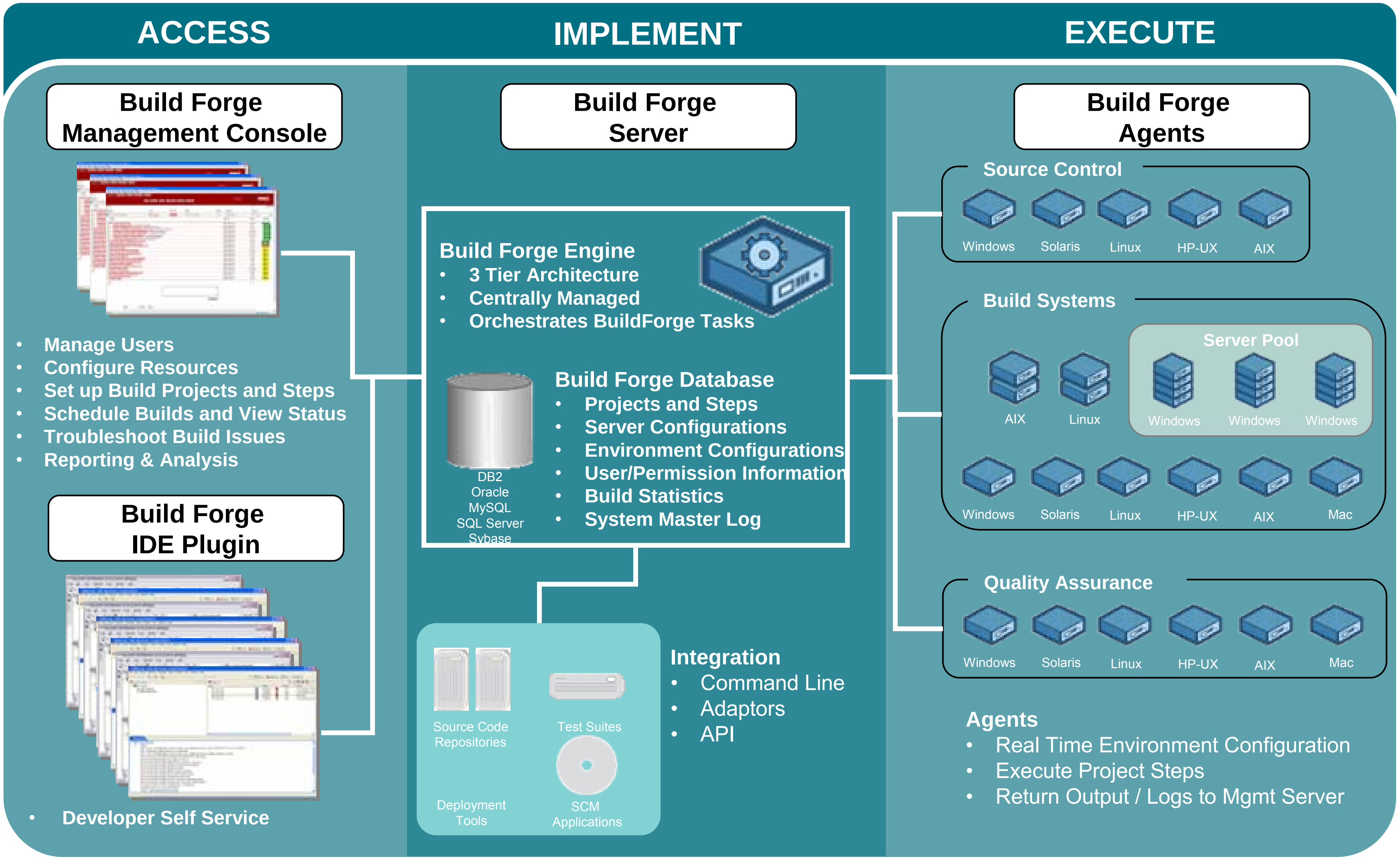
Agenda

- What is Build Forge?
- What is Build Forge *Not*?
- Rational Build Forge Function Preview
- Technical System Overview & Usage Examples

Key points

- Build Forge Architecture
- Projects
- Environments
- Security
- Servers
- Project Execution

Build Forge architecture



- Manage Users
- Configure Resources
- Set up Build Projects and Steps
- Schedule Builds and View Status
- Troubleshoot Build Issues
- Reporting & Analysis

- Developer Self Service

Build Forge Engine
- 3 Tier Architecture
- Centrally Managed
- Orchestrates BuildForge Tasks

Build Forge Database
- Projects and Steps
- Server Configurations
- Environment Configurations
- User/Permission Information
- Build Statistics
- System Master Log

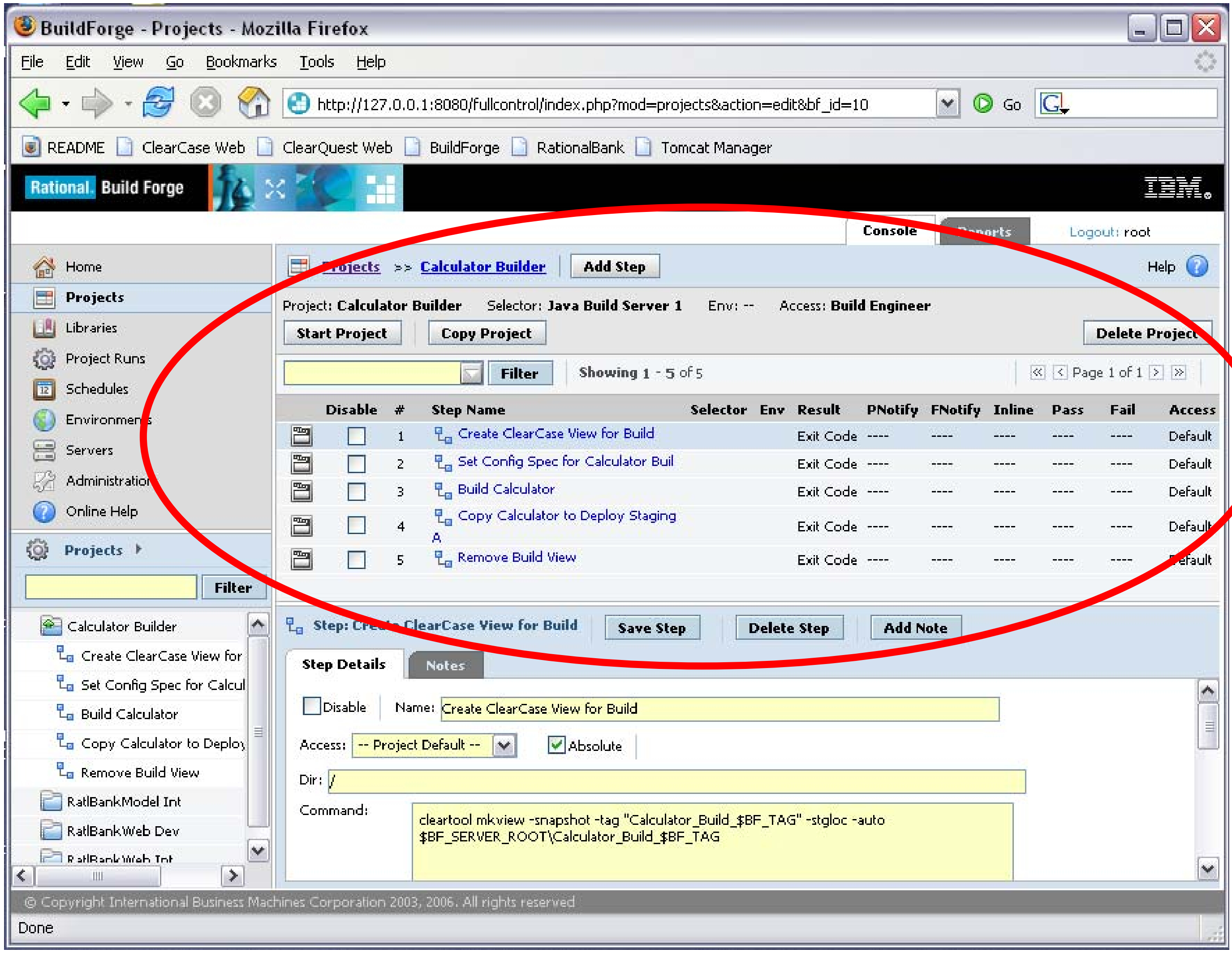
Integration
- Command Line
- Adaptors
- API

Agents
- Real Time Environment Configuration
- Execute Project Steps
- Return Output / Logs to Mgmt Server

Build Forge project

- A set of Steps which can be run in the BuildForge system. A project can be run repeatedly; each run generates a new Build.
- Step: A component of a Project. A Step stores one or more commands which can be executed on a Server. Each command may itself launch an executable file, a batch file, or script that launches many other commands.
- Command: Anything that can be invoked from the command line of a given server (through an Agent) based on the security privileges configured.

Build Forge project



BuildForge - Projects - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://127.0.0.1:8080/fullcontrol/index.php?mod=projects&action=edit&bf_id=10

README ClearCase Web ClearQuest Web BuildForge RationalBank Tomcat Manager

Rational. Build Forge

Console Reports Logout: root

Home
Projects
Libraries
Project Runs
Schedules
Environments
Servers
Administration
Online Help

Projects

Filter

Calculator Builder
- Create ClearCase View for
- Set Config Spec for Calcul
- Build Calculator
- Copy Calculator to Deploy
- Remove Build View

RatlBankModel Int
RatlBankWeb Dev
RatlBankWeb Int

Projects >> Calculator Builder | Add Step | Help

Project: **Calculator Builder** Selector: **Java Build Server 1** Env: -- Access: **Build Engineer**

Start Project | Copy Project | Delete Project

Filter | Showing 1 - 5 of 5 | Page 1 of 1

Disable	#	Step Name	Selector	Env	Result	PNotify	FNotify	Inline	Pass	Fail	Access
☐	1	Create ClearCase View for Build			Exit Code	----	----	----	----	----	Default
☐	2	Set Config Spec for Calculator Buil			Exit Code	----	----	----	----	----	Default
☐	3	Build Calculator			Exit Code	----	----	----	----	----	Default
☐	4	Copy Calculator to Deploy Staging A			Exit Code	----	----	----	----	----	Default
☐	5	Remove Build View			Exit Code	----	----	----	----	----	Default

Step: Create ClearCase View for Build | Save Step | Delete Step | Add Note

Step Details | Notes

☐ Disable Name: Create ClearCase View for Build

Access: -- Project Default -- ☑ Absolute

Dir: /

Command:

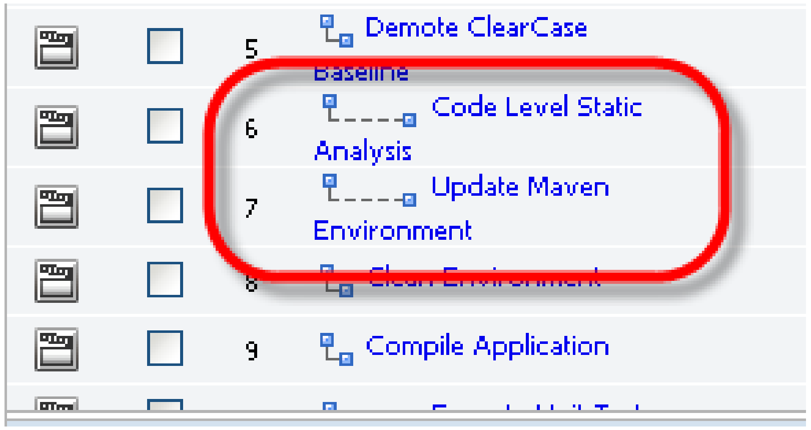
```
cleartool mkview -snapshot -tag "Calculator_Build_$BF_TAG" -stgloc -auto
$BF_SERVER_ROOT\Calculator_Build_$BF_TAG
```

© Copyright International Business Machines Corporation 2003, 2006. All rights reserved

Done

Build Forge project

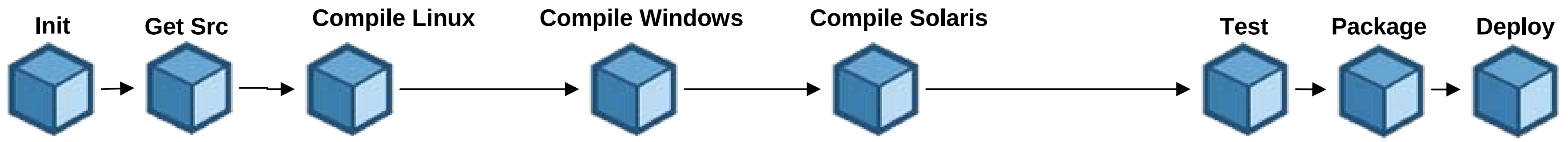
- **Threading:** Threading allows for multiple Steps to run in parallel, thereby increasing the speed of your build.
  - Multiple, adjacent, Threaded steps are thought to be a Thread Block
  - Threaded Steps are indented on the Project Page



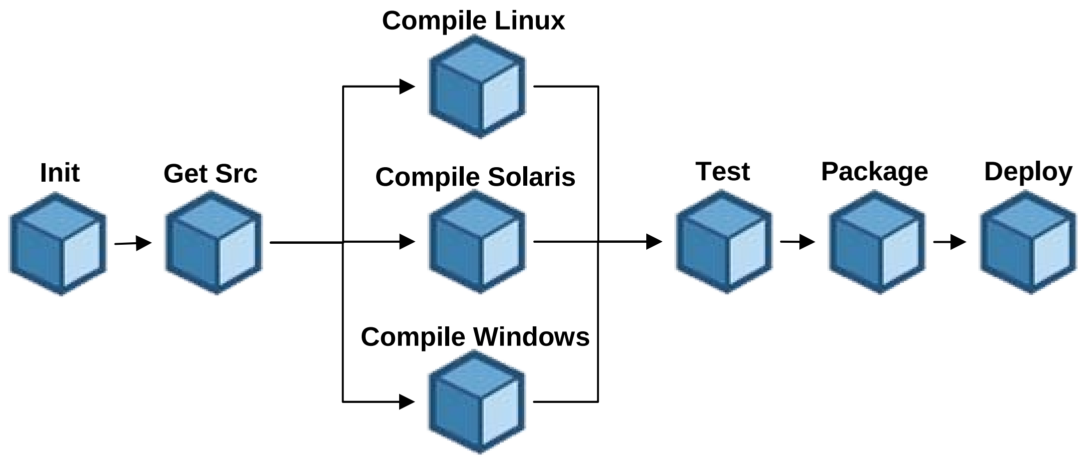
	☐	5	Demote ClearCase Baseline
	☐	6	Code Level Static Analysis
	☐	7	Update Maven Environment
	☐	8	Clean Environment
	☐	9	Compile Application

Build Forge projects - Threading

Without Threading



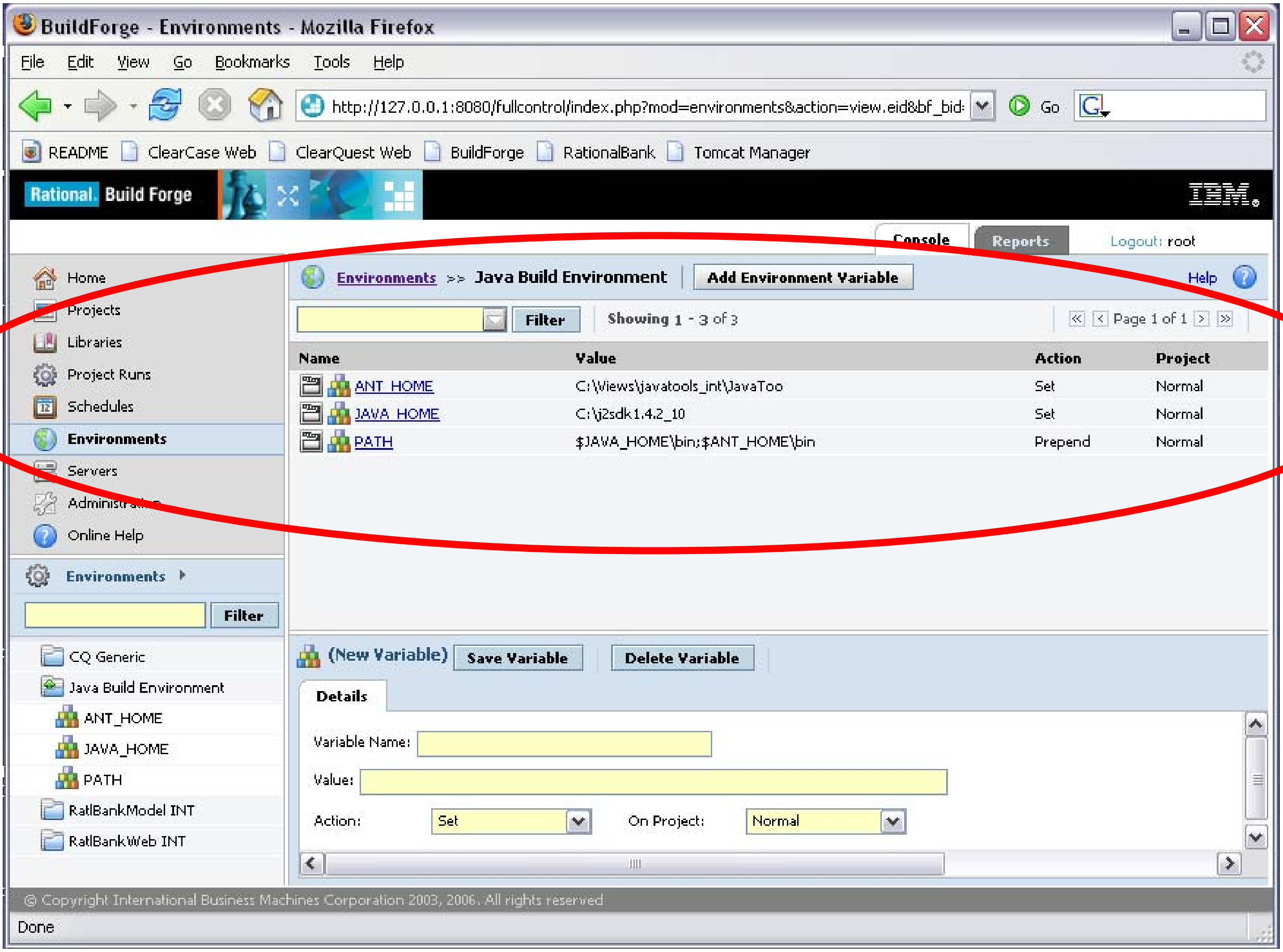
With Threading



Build Forge environments

- The BuildForge system allows you to manage environment variables separately from your Projects, Servers, and Steps
  - You can define Environment Groups
    - Containing one or more Environment Variables
  - You can assign Environment Groups to Servers, Projects, and Steps
    - At runtime, the system assembles the final working environment for a Step from all of the relevant Groups assigned to the Server, Project, and Step

Build Forge environments


BuildForge - Environments - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://127.0.0.1:8080/fullcontrol/index.php?mod=environments&action=view.eid&bf_bid:
Go
README ClearCase Web ClearQuest Web BuildForge RationalBank Tomcat Manager
Rational. Build Forge
Console Reports Logout: root
Home
Projects
Libraries
Project Runs
Schedules
Environments
Servers
Administration
Online Help
Environments
Filter
CQ Generic
Java Build Environment
ANT_HOME
JAVA_HOME
PATH
RatlBankModel INT
RatlBankWeb INT
Environments >> Java Build Environment
Add Environment Variable
Help
Filter
Showing 1 - 3 of 3
Page 1 of 1
(New Variable) Save Variable Delete Variable
Details
Done

Name	Value	Action	Project
ANT_HOME	C:\Views\javatools_int\JavaToo	Set	Normal
JAVA_HOME	C:\j2sdk1.4.2_10	Set	Normal
PATH	$JAVA_HOME\bin;$ANT_HOME\bin	Prepend	Normal

Variable Name:
Value:
Action: Set
On Project: Normal

© Copyright International Business Machines Corporation 2003, 2006. All rights reserved

Build Forge security

- Concepts in the BuildForge security system:
  - **Users** – A user in the BuildForge environment corresponds to a single, named person who interacts with the system in some way.
  - **Access Groups** – An access group is a collection of users who are assigned permissions to perform specific tasks within the BuildForge system.

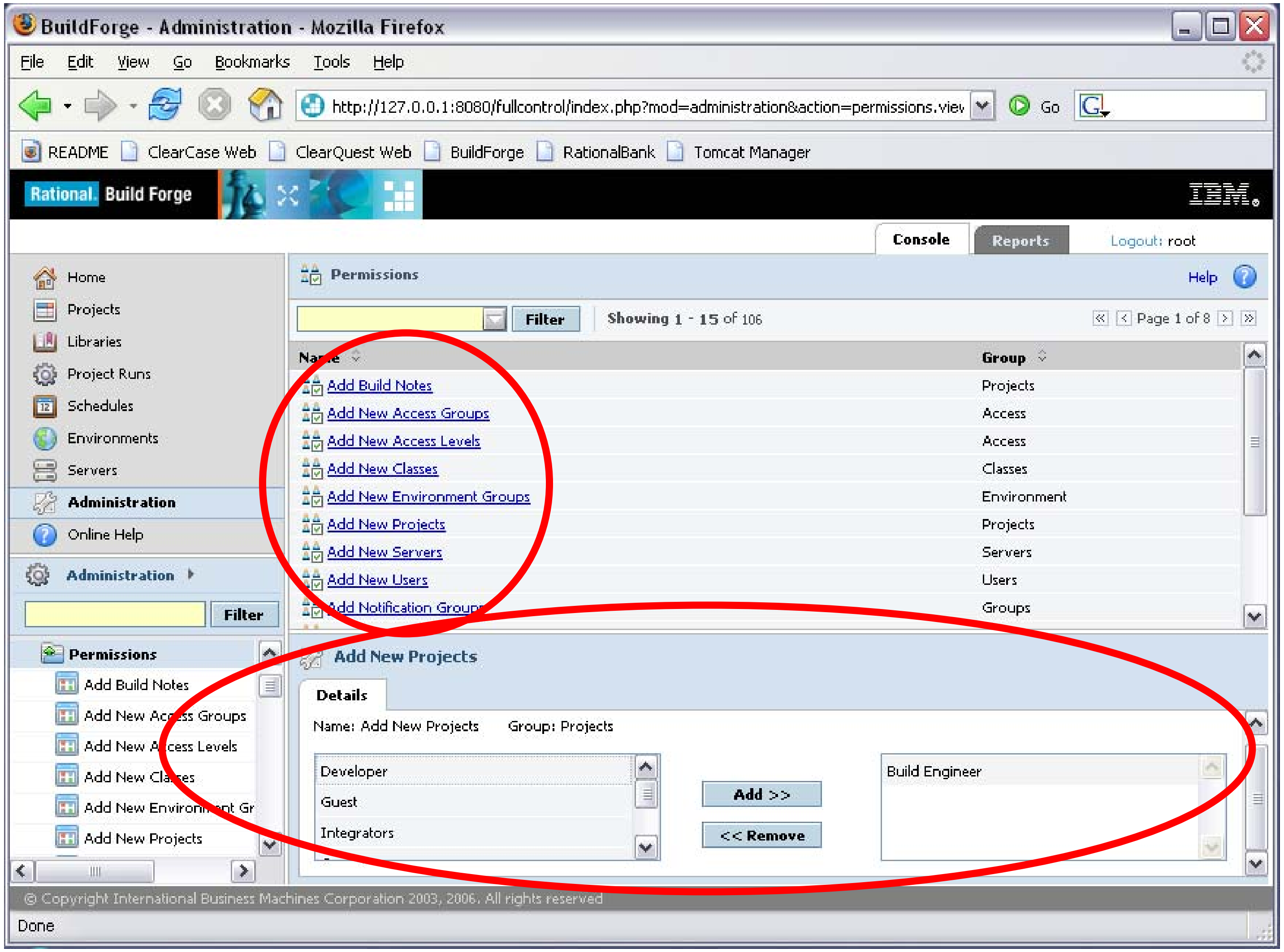
    An Access Group is usually associated with a logical Role within an organization – *the .NET User Interface development team,* for example.
  - **Permissions** – Permissions are typically set on system level functionality and are assigned to Access Groups only – not individual Users.
  - **Ownership** – Specific resources – Projects, Servers, Steps, etc. – in the BuildForge system have a single, named Access Group as Owner.

    Ownership over an entity allows for full edit/delete/execute privileges over that resource.

Build Forge security

- Role-based system
  - Access *Groups* represents a role a *User* has in organization
  - Roles have *Permissions* and *Ownership*
- User access is determined by the **union** of the Permissions of all groups the user belongs to
- Combination of *Permissions* and *Ownership* define what a group can do and/or see
  - To edit a give Project, a User would have to have *Ownership* on the Project AND *Permissions* to Edit Projects
- Example
  - User who is a member of the Guest group (and no other groups) sees only Projects which have the Guest group assigned as their Access property
  - User can only launch projects with Guest access

Build Forge security


BuildForge - Administration - Mozilla Firefox

http://127.0.0.1:8080/fullcontrol/index.php?mod=administration&action=permissions.view

Permissions

Showing 1 - 15 of 106 — Page 1 of 8

Name	Group
Add Build Notes	Projects
Add New Access Groups	Access
Add New Access Levels	Access
Add New Classes	Classes
Add New Environment Groups	Environment
Add New Projects	Projects
Add New Servers	Servers
Add New Users	Users
Add Notification Groups	Groups

Add New Projects

Details

Name: Add New Projects Group: Projects

Available: Developer, Guest, Integrators

Assigned: Build Engineer

© Copyright International Business Machines Corporation 2003, 2006. All rights reserved

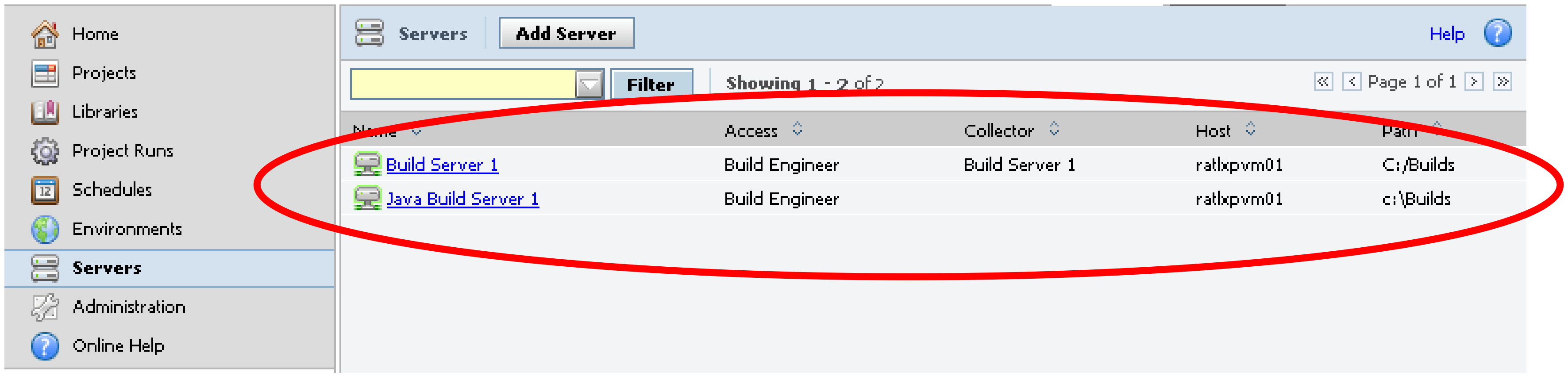
File Edit View Go Bookmarks Tools Help
README ClearCase Web ClearQuest Web BuildForge RationalBank Tomcat Manager
Rational. Build Forge
Console Reports Logout: root
Home Projects Libraries Project Runs Schedules Environments Servers Administration Online Help
Administration
Filter
Permissions
Add Build Notes
Add New Access Groups
Add New Access Levels
Add New Classes
Add New Environment Gr
Add New Projects
Help
Add >>
<< Remove
Done

Build Forge servers

- A Server is a ***logical*** resource in the BuildForge environment that represents any physical system running the BuildForge Agent.
  - A Server is defined within the BuildForge system with certain properties, such as a default path.

- A Server Pool is a group of Servers with similar build properties (running the same operating system and with the same compilers available, for example).
  - The BuildForge system can run a Project on a different Server in the same Pool when the default Server for the Project is busy.

Build Forge servers



Name	Access	Collector	Host	Path
Build Server 1	Build Engineer	Build Server 1	ratlxpvm01	C:/Builds
Java Build Server 1	Build Engineer		ratlxpvm01	c:\Builds

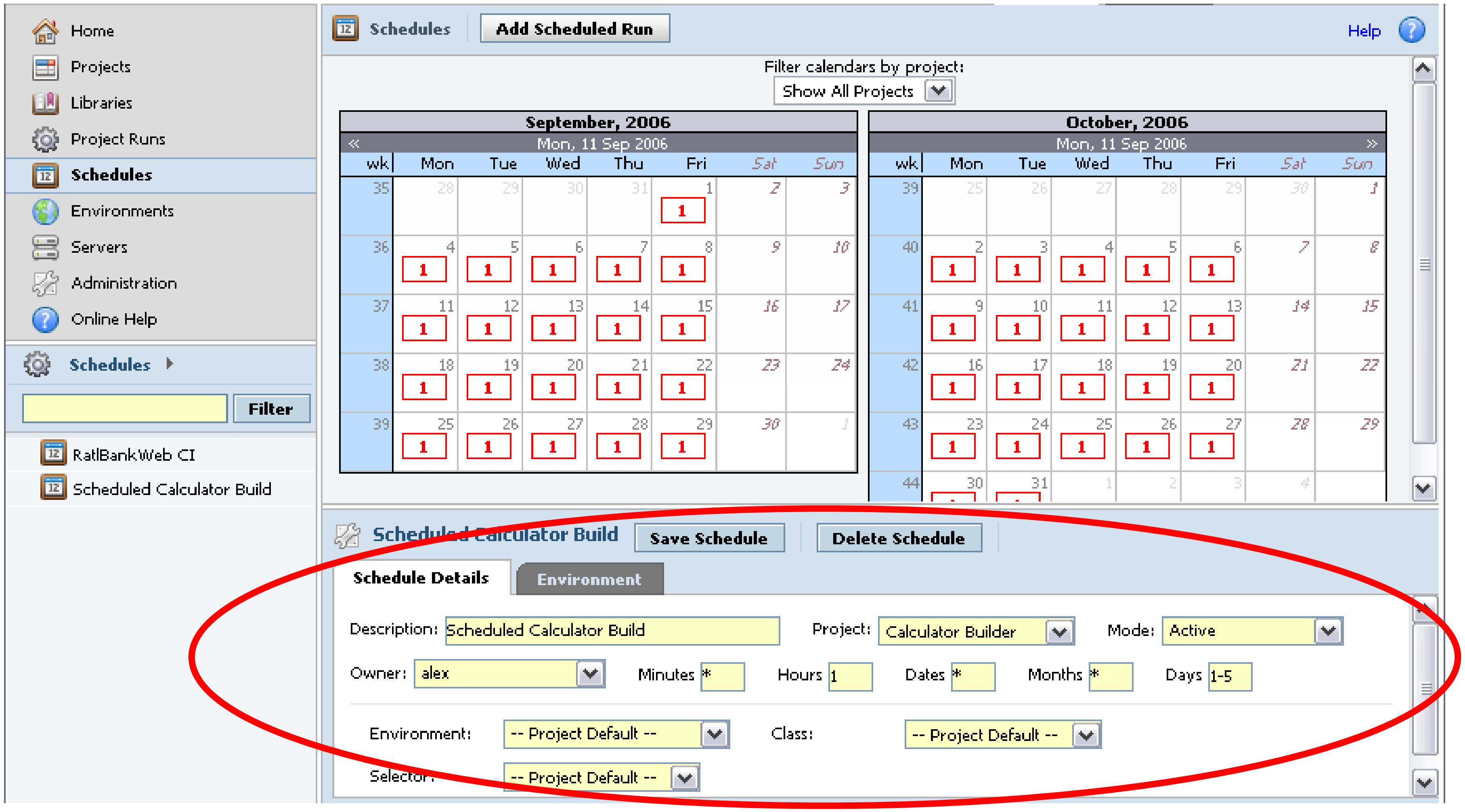
Build Forge project execution

4 Primary Ways to Execute a Project:

Quick Start	Run with default settings
Manual Start	Override default settings
Scheduled	Run a single or recurring instance at some future date(s)
Restart	Rerun a failed Build, starting from the point of failure

Build Forge project execution

Example Schedule



Filter calendars by project: Show All Projects

September, 2006 — Mon, 11 Sep 2006

wk	Mon	Tue	Wed	Thu	Fri	Sat	Sun
35	28	29	30	31	1 1	2	3
36	4 1	5 1	6 1	7 1	8 1	9	10
37	11 1	12 1	13 1	14 1	15 1	16	17
38	18 1	19 1	20 1	21 1	22 1	23	24
39	25 1	26 1	27 1	28 1	29 1	30	1

October, 2006 — Mon, 11 Sep 2006

wk	Mon	Tue	Wed	Thu	Fri	Sat	Sun
39	25	26	27	28	29	30	1
40	2 1	3 1	4 1	5 1	6 1	7	8
41	9 1	10 1	11 1	12 1	13 1	14	15
42	16 1	17 1	18 1	19 1	20 1	21	22
43	23 1	24 1	25 1	26 1	27 1	28	29
44	30	31	1	2	3	4	

Scheduled Calculator Build

Description: Scheduled Calculator Build Project: Calculator Builder Mode: Active

Owner: alex Minutes * Hours 1 Dates * Months * Days 1-5

Environment: -- Project Default -- Class: -- Project Default --

Selector: -- Project Default --

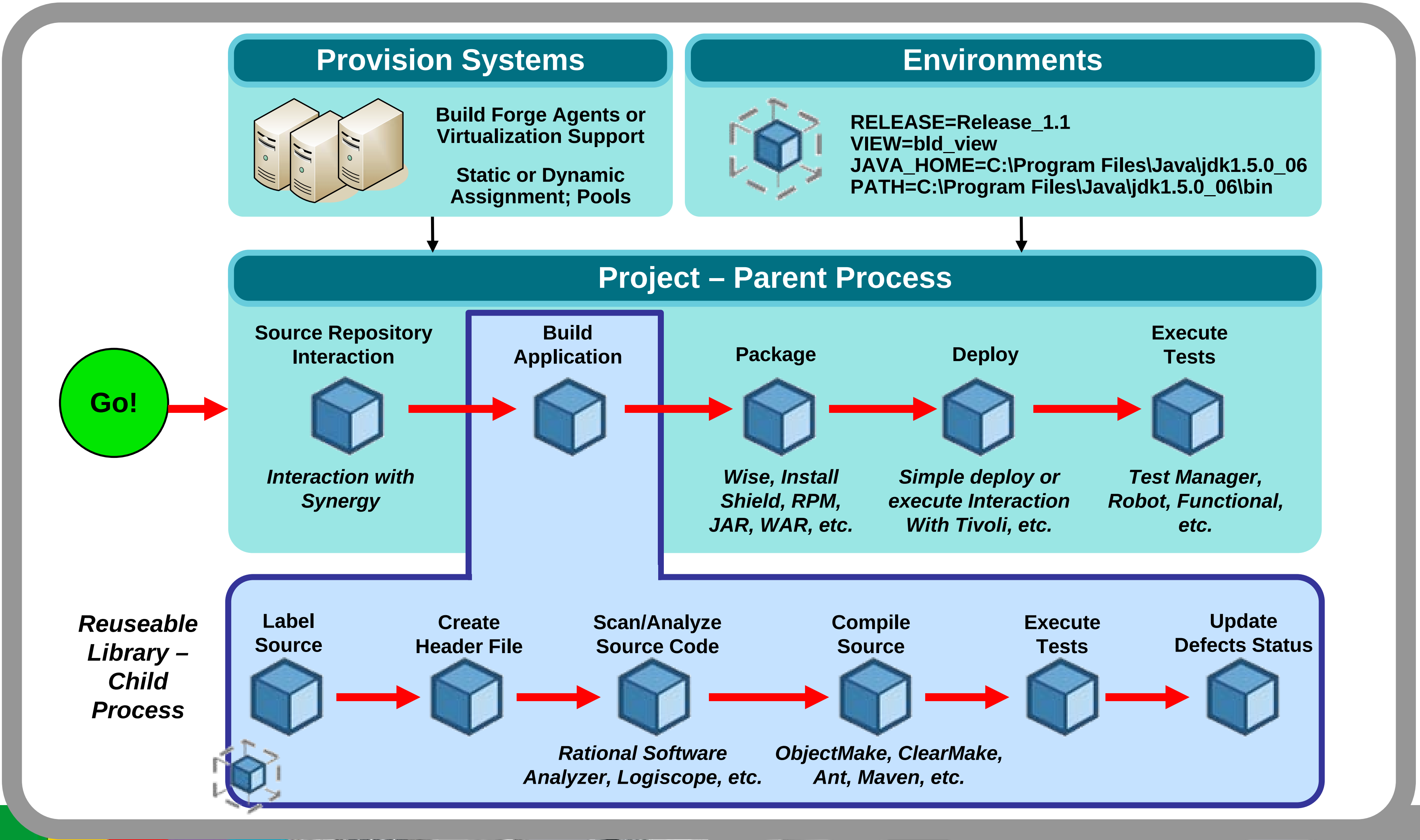
Summary

- The Build Forge architecture provides a robust, flexible foundation for build (and more) process automation.
- We have discussed
  - Build Forge Architecture
  - Projects
  - Environments
  - Security
  - Servers
  - Project Execution
- In the following labs we will dive into a Build Forge environment and see how to use these concepts.

Agenda

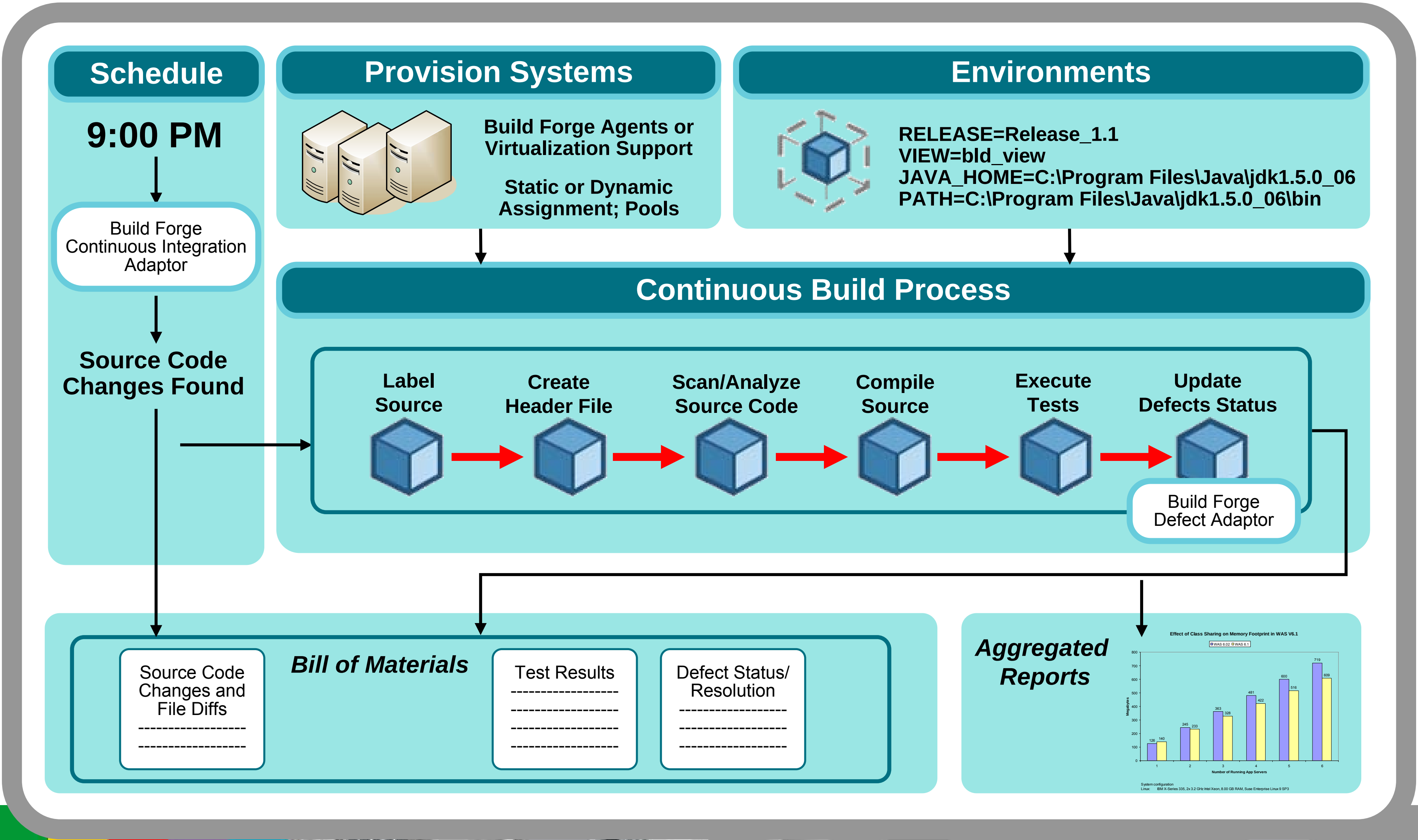
- What is Build Forge?
- What is Build Forge *Not*?
- Rational Build Forge Function Preview
- Technical System Overview & Usage Examples

Example – Basic Process Automation & Reuse

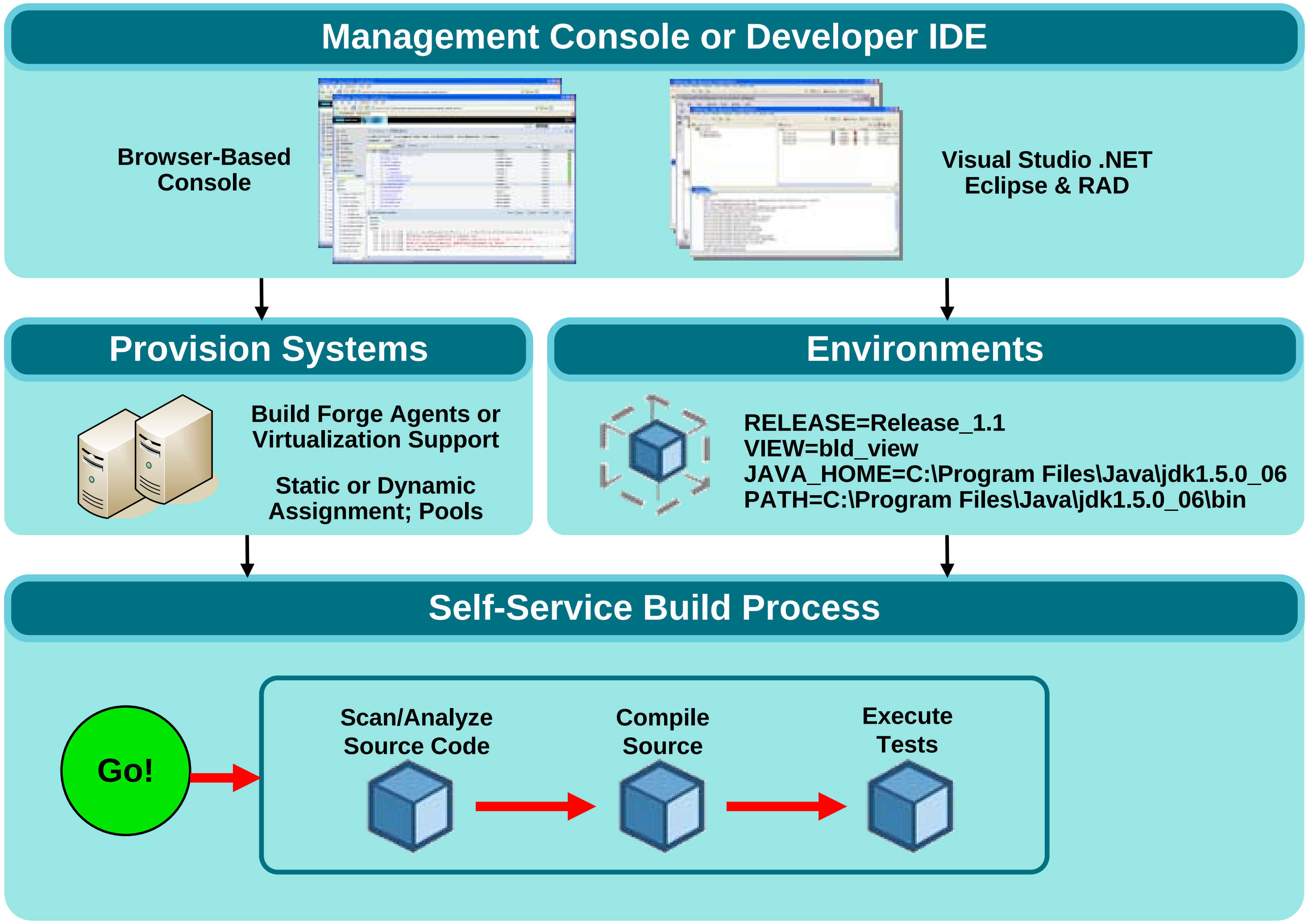


Provision Systems
Build Forge Agents or Virtualization Support
Static or Dynamic Assignment; Pools
Environments
RELEASE=Release_1.1
VIEW=bld_view
JAVA_HOME=C:\Program Files\Java\jdk1.5.0_06
PATH=C:\Program Files\Java\jdk1.5.0_06\bin
Project – Parent Process
Go!
Source Repository Interaction
Interaction with Synergy
Build Application
Package
Wise, Install Shield, RPM, JAR, WAR, etc.
Deploy
Simple deploy or execute Interaction With Tivoli, etc.
Execute Tests
Test Manager, Robot, Functional, etc.
Reuseable Library – Child Process
Label Source
Create Header File
Scan/Analyze Source Code
Rational Software Analyzer, Logiscope, etc.
Compile Source
ObjectMake, ClearMake, Ant, Maven, etc.
Execute Tests
Update Defects Status

Example – Continuous Integration


Schedule
9:00 PM
Build Forge Continuous Integration Adaptor
Source Code Changes Found
Provision Systems
Build Forge Agents or Virtualization Support
Static or Dynamic Assignment; Pools
Environments
RELEASE=Release_1.1
VIEW=bld_view
JAVA_HOME=C:\Program Files\Java\jdk1.5.0_06
PATH=C:\Program Files\Java\jdk1.5.0_06\bin
Continuous Build Process
Label Source
Create Header File
Scan/Analyze Source Code
Compile Source
Execute Tests
Update Defects Status
Build Forge Defect Adaptor
Bill of Materials
Source Code Changes and File Diffs
Test Results
Defect Status/ Resolution
Aggregated Reports
Effect of Class Sharing on Memory Footprint in WAS V6.1
WAS 6.02
WAS 6.1
Megabytes
Number of Running App Servers
126 140
245 233
363 328
481 422
600 516
719 609
System configuration
Linux: IBM X-Series 335, 2x 3.2 GHz Intel Xeon, 8.00 GB RAM, Suse Enterprise Linux 9 SP3

Example – Developer Self Service


Management Console or Developer IDE
Browser-Based Console
Visual Studio .NET Eclipse & RAD
Provision Systems
Build Forge Agents or Virtualization Support
Static or Dynamic Assignment; Pools
Environments
RELEASE=Release_1.1
VIEW=bld_view
JAVA_HOME=C:\Program Files\Java\jdk1.5.0_06
PATH=C:\Program Files\Java\jdk1.5.0_06\bin
Self-Service Build Process
Go!
Scan/Analyze Source Code
Compile Source
Execute Tests

Customer Example – Parallel Execution

Provision Systems

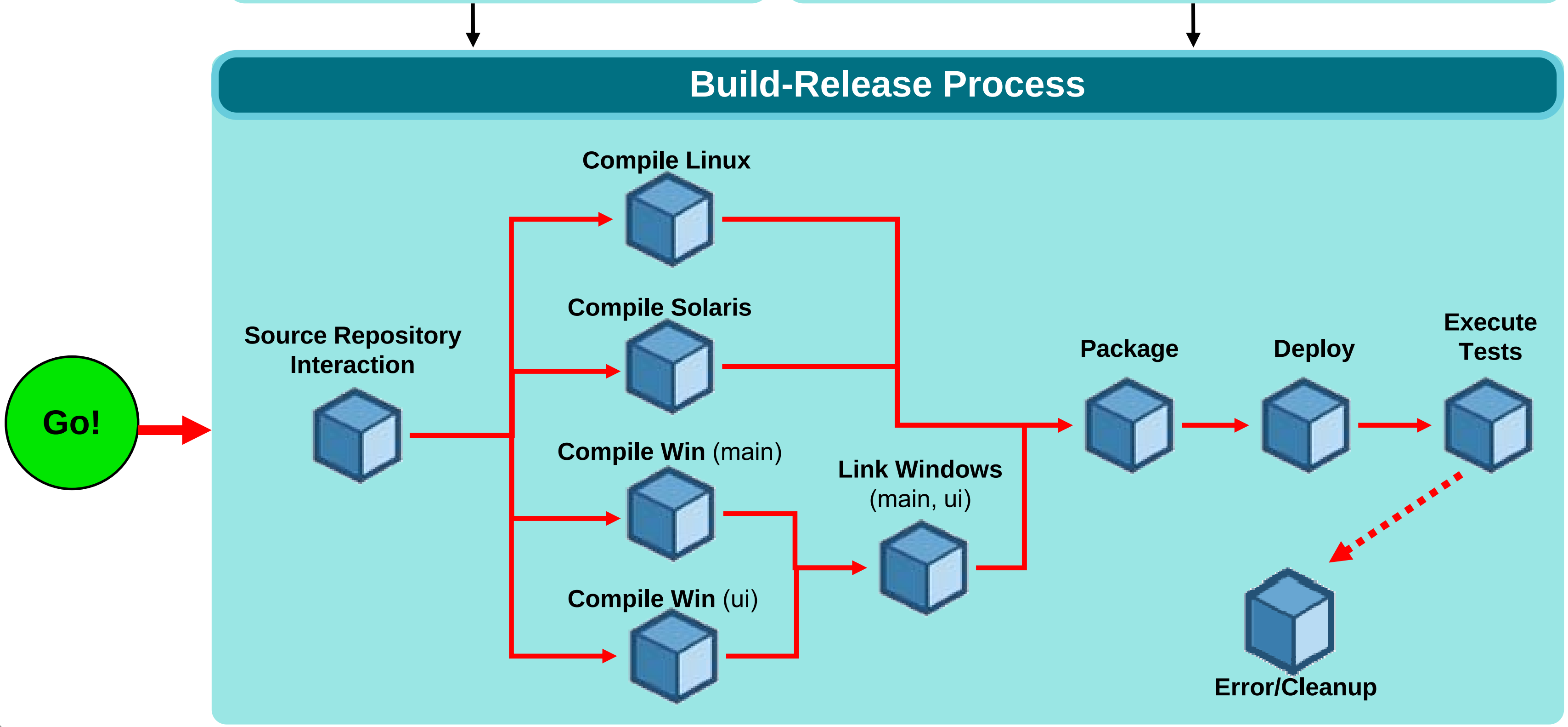
Build Forge Agents or Virtualization Support

Static or Dynamic Assignment; Pools

Environments

```
RELEASE=Release_1.1
VIEW=bld_view
JAVA_HOME=C:\Program Files\Java\jdk1.5.0_06
PATH=C:\Program Files\Java\jdk1.5.0_06\bin
```

Build-Release Process



Selected Customers

VERITAS™ bp Adobe CAMPBELL & COMPANY SINCE 1972 tellabs® LEHMAN BROTHERS SIEMENS FIDELITY NATIONAL FINANCIAL

IBM Morgan Stanley Google™ TEXAS INSTRUMENTS Allstate. You're in good hands.

MetLife® Scotiabank paymetric EMC² smarts

DIGITAL INSIGHT® CLEARCHANNEL

Intuit. invensys. Fidelity Investments® agile Reynolds & Reynolds.

EA.com ELECTRONIC ARTS MULTIMEDIA GAMES Autodesk® EMC² where information lives® VITAL

GE AVAYA Vicarious Visions® symantec. hp PERVASIVE® VITAL PROCESSING SERVICES®

COMPUCREDIT STATE FARM INSURANCE CITADEL SAMSUNG OVERWATCH SYSTEMS John Hancock

BetweenMarkets INTERWOVEN Enterprise Content Management Solutions for Business

Canon CITRIX online Manhattan Associates. Stratus Technologies EXCEL Where Applications Perform Best

CONEXANT SOUTHWEST.COM® OUTCOME POST-MARKETING EXCELLENCE hp LaserJet Medplus WYSE freescale™ semiconductor

Thank
You

BackUp :

Continuous Integration

Continuous Integration

IBM

CI的優點

- 軟體系統的整合往往是一個漫長而痛苦的過程
- 持續整合使得軟體缺陷更易於被改正
  - 因爲前後兩次建構之間所做的改動很小，所以很容易定位軟體缺陷的位置並修復
  - 多個軟體缺陷之間會相互干擾，使得缺陷的定位更爲困難
  - 當缺陷較少時，開發人員心理上也更有餘力去改正它們
- 持續整合也使我們可以更頻繁地發佈版本
  - 更快地修正軟體產品中的缺陷
  - 對使用者feedback的回應更爲迅速
  - 使用者可以更早體驗產品的新功能
  - 打破開發團隊和使用者之間的障礙

Continuous Integration, Martin Fowler, http://martinfowler.com/articles/continuousIntegration.html

CI的原則

- 建立統一的建構管理庫
- 建構過程的自動化
- 對建構結果進行自動化的測試
- 團隊成員應該每天交付開發結果
- 加快建構的速度
- 保證測試環境和生產環境的一致性
- 團隊成員可以方便獲取專案進度資訊
- 自動部署建構結果

建立統一的建構管理庫

- 什麼樣的專案檔應該放在建構庫中
  - 程式碼、需求文件、設計文件
  - 測試腳本、專案屬性檔、資料庫結構(DDL)、安裝腳本、Third-Party Lib
  - 任何一名新成員加入該專案時，能夠很快地 Checkout 檔並且建構整個軟體系統

建構過程的自動化

- 整個的編譯、打包、安裝到測試/生產平臺的過程應該是自動的
  - 手工操作很浪費時間，並且易於出錯
  - 所有的步驟都應該由腳本來自動化地完成
  - 任何一名新成員加入該項目時，能夠很快地建構整個軟體系統並且完成套裝軟體的安裝
- 對於 Build 工具的要求
  - 快速完成建構，自動識別哪些部分是需要重新編譯、哪些部分不需要
- 自動化的建構工具
  - IBM Rational Build Forge
  - IBM Rational ClearCase - Clearmake
  - Make (Unix)
  - Ant (Java)

對建構結果進行自動化的測試

- 建構的過程中應包括對建構結果的一個單元測試
- 該測試失敗的話，就意味著此次建構的失敗
- 單元測試的目的不是爲了保證建構結果完全沒有問題，而是在一定程度上驗證軟體的品質
  - 建構自測並不是一個完美的測試，但是在每次建構中都會被執行
- 進行軟體自測的工具
  - XUnit : JUnit, CUnit
  - IBM Testing Automation Framework

團隊成員應該每天交付開發結果

- 整合主要解決不同開發人員所做修改之間的衝突
- 整合越頻繁，發生的修改衝突也就越小
- 開發人員應該每天交付開發結果，即把每個人開發的結果整合到整個專案中去
- 每日交付鼓勵開發人員把開發任務分解成小塊(幾個小時的工作量)
- 使用 ClearCase UCM 可以很好地支援每日交付
  - 在開發流(development stream)中測試自己開發的那部分代碼
  - 開發活動(activity) 可以幫助分解開發人員的工作量
  - 把開發結果提交(submit)到整合流中
  - 在整合流(integration stream)中測試自己交付的部分跟專案的整體是否有衝突，如果發現衝突則馬上解決

加快建構的速度

- 持續整合的要點是快速提供回饋
- 持續整合中最花時間的部分是建構
  - 調查表明，超過一個小時的建構是讓人不可接受的
- 減少建構花費時間會讓每一個開發人員受益，因爲每一個人都要做每日交付
- 減少建構花費時間的方法 – 分階段的建構
  - 交付建構 (commit build) – 儘量快速
    - 減少建構中測試用例的數量
  - 二次建構 (second build) – 完整的測試

保證測試環境和生產環境的一致性

- 測試的目的是找出軟體在生產系統中可能發生的問題
- 盡可能保持測試環境和生產環境的一致性
  - 相同版本的作業系統、資料庫系統、硬體配置等
  - 任何不一致性都可能導致測試環境和生產環境運行結果的不一致
- 越來越多的測試團隊開始使用虛擬化技術來配置測試環境

團隊成員可以方便取得專案進度

- 持續整合的關鍵是團隊溝通
  - 專案做了多少次建構
  - 當前的交付建構進展狀況如何
  - 每次建構之間實現了哪些需求
  - 每次建構上發現了哪些缺陷、哪些已經改正了
- 這些專案狀態資訊可以很方便地從Rational SDP 平臺上獲取
  - Build Forge – 提供建構狀態資訊
  - ClearCase – 管理每日交付
  - ClearQuest – 記錄每一個建構上所發現的軟體缺陷
  - RequisitePro – 記錄需求被實現的進度

自動部署建構結果

- 在持續整合中，安裝建構結果是一個非常頻繁的操作
- 需要一種自動化的方案來把建構結果安裝到測試和生產系統上去
- 當發現新安裝的建構有問題時，也可以快速回覆到上一個狀態
- 支援自動化部署的工具
  - IBM Rational Build Forge
  - IBM Rational Tivoli Provisioning Manager (對於大規模的部署，如安裝一個新版本到幾百台 ATM 機或伺服器上)