Automating the Assembly of Software Systems with Rational Build Forge and Rational Team Concert

Adrian Daniels

Senior IT Specialist, IBM Rational

adrian.daniels@uk.ibm.com

RU READY TO SAVE THE DAY

IBM Rational Software Development Conference 2008

WHERE TEAMS ARE R-HEROES

IBM

Rational. software

Agenda

- Jazz Team Build

- Why use Build Forge with Team Concert ?

- Build Forge and Team Concert Integrated Experience

- Use Cases

- Architecture

Jazz Team Build

- Built around the philosophy of "Team First"

  – Teams engaged in collaborative software development

    • Developer asks "Have my latest integrated changes broken the build ?"

    • Developer asks "Can I reproduce what the customer is using ?"

    • Tester asks "what build has a bug been fixed in ?"

- Key Goals for Jazz Team Build

  – Build is a first class citizen in the development lifecycle

  – All teams members have awareness and control of builds

  – Full traceability between builds, work items and code

Jazz Team Build - Goals Alive Today

Jazz Team Build – Solution Corner Stones

- Lightweight definitions

  – Build Definition

  – Build Engine Declarations

- Data Model for Build Results

  – Logs, Downloads, Links, Compile/Test Information

  – Traceability to Work Items and Code

- Build Toolkit

  – In the form to Ant Tasks

- Build Engine

  – Simple Build Engine

  – Continuous Integration Loop

Why Use Build Forge with Team Concert (1)?

- Distributed capability

  – Simultaneous execution on many platforms

  – Broader platform support

  – Distributed purge/clean-up

- Capacity/Throughput

- Ability to parallelise/accelerate process execution

- Ability to manage highly varied toolsets without scripting

Why Use Build Forge with Team Concert (2)?

- Governance/Traceability across tools and machines

- Integrated scheduler for non-compile related tasks

- Sophisticated logging of whole build output

- Integrated security

- Ability to manage processes for Jazz and non-Jazz environments simultaneously

- Process automation, not just build automation (process automation is a superset of build automation)

Build Forge and Team Concert Integrated Experience

- Build Forge becomes a normal resource in Team Concert (RTC)
  - RTC Users can pick Build Forge from Build Definition Template list
  - Connection info stored in the Jazz server for smooth flow
- Build Forge objects matched to RTC equivalents
  - Projects have matching RTC Build Definition
  - Project Environments have matching RTC Build Properties
  - Build Forge Engine has matching Jazz Build Engine
- Jazz Build Requests will fire Build Forge Jobs
- Build Forge Job results available directly in Jazz
  - Summary info directly in Jazz
  - Detail data still in Build Forge with direct link-out from Jazz

Use Cases

- Use Build Forge as a Jazz Build Engine

- Running Build Forge Project from within Team Concert

- Build Forge Build Result Linkage

- Team Concert Governance of Build Forge Builds

Use Build Forge as a Jazz Build Engine

- Allows Build Forge to operate as a standard Jazz Build Engine

- Enabled by installing RTC Client Build Forge plug-in and Jazz Server Build Forge plug-in

- User will see Build Forge projects and be able to create or associate Jazz build definitions with them

Running Build Forge Project from within Team Concert

- Facilitated by the use of two plug-ins:
  - Rational Team Concert Client plug-in
  - Jazz Server Build plug-in
- Build Forge projects appear as standard Jazz Build Definitions
- Prior Build Forge run information displayed as Jazz Build Results
  - Summary of Build results information
  - Detailed Build Forge BOM information available through request
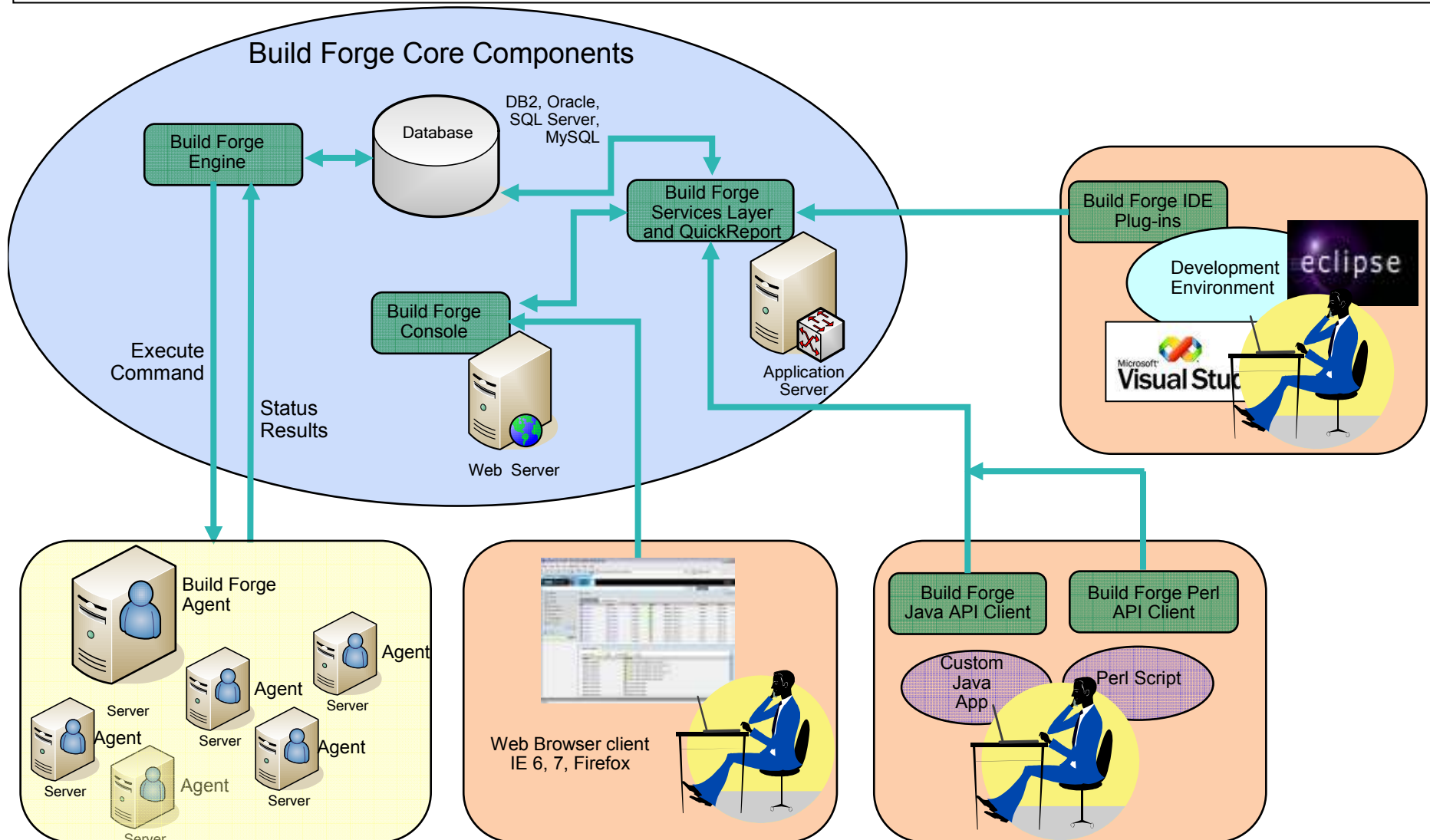
Build Forge Build Result Linkage

- Linked Build Forge Projects represented as true Jazz objects
  - Build Definition Items
  - Build Properties Items
  - Build Result Items
- Build Forge Linked objects also have typical Jazz pointers
  - Work Items
  - SCM artifacts
  - Requirements Items
- Same Options as "Regular" Jazz Objects
  - Example: Work items can be created automatically on failed builds

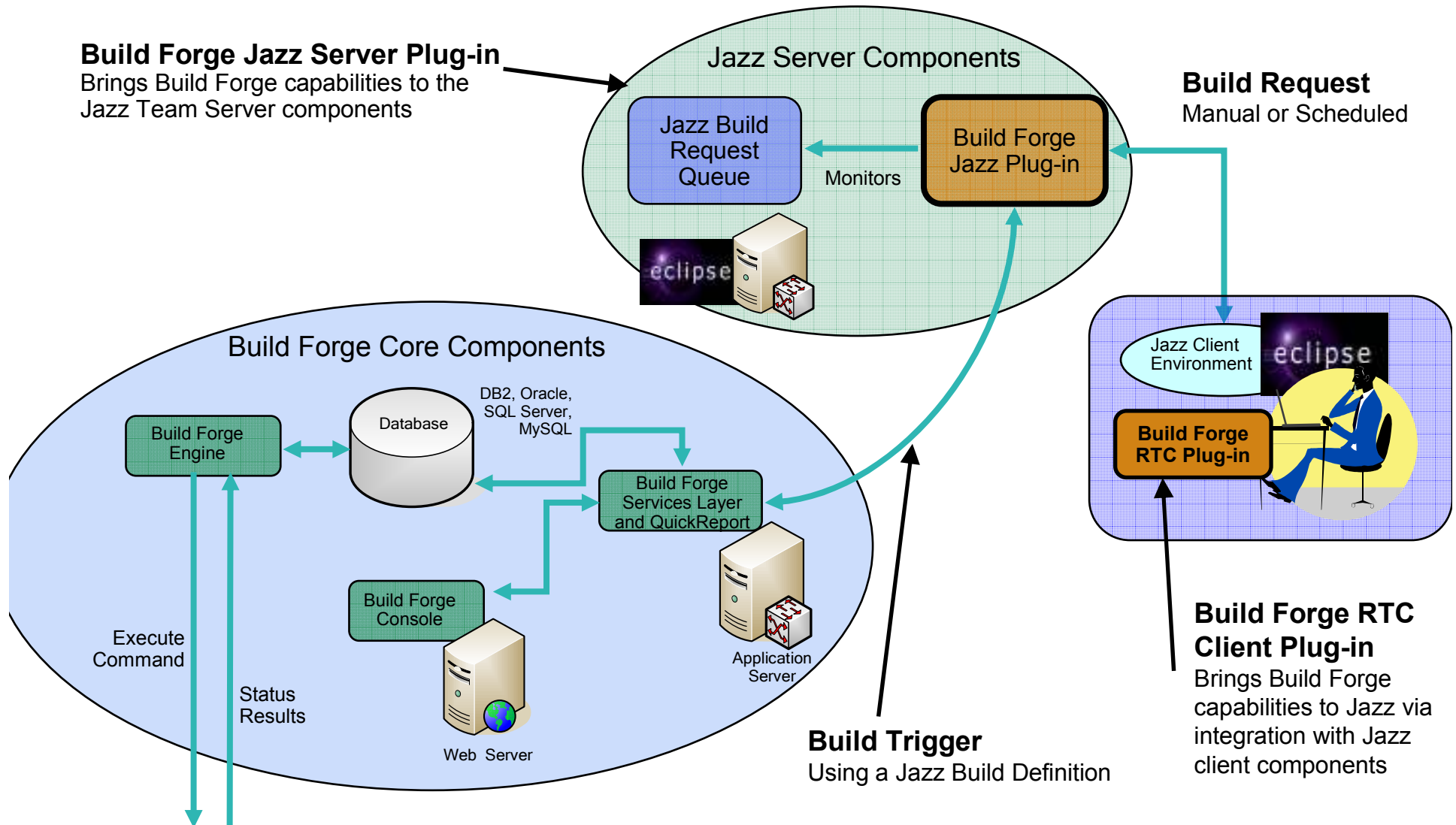# Team Concert Governance of Build Forge Builds

- Build Forge Jobs are part of Jazz process enactment and enforcement

    – Same Governance as work items and SCM

    – Achieved by Build Forge projects being true Jazz build definition items

- Example

    – Custom Jazz rule that requires specific build properties to be enforced on builds would show up as Build Forge Environment entries

Build Forge 7.1 Architecture Recap

Build Forge Core Components

- Build Forge Engine
- Database — DB2, Oracle, SQL Server, MySQL
- Build Forge Services Layer and QuickReport
- Build Forge Console
- Application Server
- Web Server

Execute Command

Status Results

- Build Forge Agent
- Agent / Server (multiple)

- Web Browser client IE 6, 7, Firefox

- Build Forge IDE Plug-ins
- Development Environment — eclipse, Microsoft Visual Studio

- Build Forge Java API Client
- Build Forge Perl API Client
- Custom Java App
- Perl Script

Build Forge 7.1 with Jazz Integration

Build Forge Jazz Server Plug-in
Brings Build Forge capabilities to the
Jazz Team Server components

Jazz Server Components

Jazz Build
Request
Queue

Build Forge
Jazz Plug-in

Monitors

eclipse

Build Request
Manual or Scheduled

Jazz Client
Environment

eclipse

**Build Forge
RTC Plug-in**

Build Forge Core Components

Build Forge
Engine

Database

DB2, Oracle,
SQL Server,
MySQL

Build Forge
Services Layer
and QuickReport

Execute
Command

Status
Results

Build Forge
Console

Application
Server

Web Server

Build Trigger
Using a Jazz Build Definition

**Build Forge RTC
Client Plug-in**
Brings Build Forge
capabilities to Jazz via
integration with Jazz
client components

Learn more at:

- IBM Rational software
- IBM Rational Software Delivery Platform
- Process and portfolio management
- Change and release management
- Quality management

- Architecture management
- Rational trial downloads
- developerWorks Rational
- IBM Rational TV
- IBM Rational Business Partners