Flexible Configuration Management for a Component-based Software Asset Repository

Presented on behalf of MKS by

Tom Brett
Technical Consultant, Confluence Systems Ltd

MKS Overview

- Provider of enterprise software configuration management solutions since 1984
- Strong Global 1000 penetration
- 26,000 seats of enterprise SCM installed since July, 2001
- HSBC, DaimlerChrysler, Pharmacia, Johnson & Johnson, Continental Teves, RBC
- Public - TSE:MKX

Build Better Software™

About the Author

- 20 years in Software Development
- 12 years in CM
- Automotive, Banking, Telecoms & other
- 3 years with MKS
- Now independent

Software Repository Business Drivers

- Sharing of functional components
  - no reinventing the wheel!
- Flexible assembly of components into products
  - rapid response to new requirements & new customers
- Tailoring of a core product for different customers
  - stay flexible to customer needs while optimising maintainability

Component Based SCM

- Version control at individual file level *(member)*
- Files grouped into component configurations *(subproject)*
- Components grouped re-usably into product configurations *(project)*
- Each component may have its own independent version & branching history

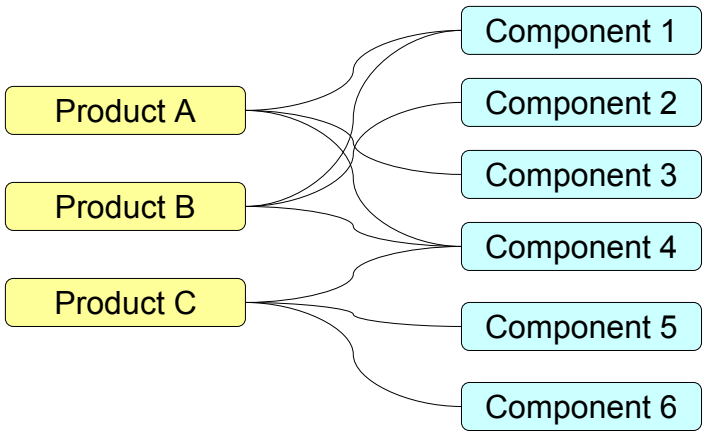
Build Better Software™

Pattern: Component Palette

- Multiple components in flat structure
- Each product includes an arbitrary combination of components
- Symmetrical
- Can treat as 100% re-used code
  - though some components may in fact only be in one product
- Easy to re-use code later
  - even if not originally planned to do so

Build Better Software™

Component Palette – Diagram



Component Palette – typical situation

- Large library, small product
- Each product typically includes only a small % of the whole code base
- Not much release history at product level
- Typical example – mobile phone industry

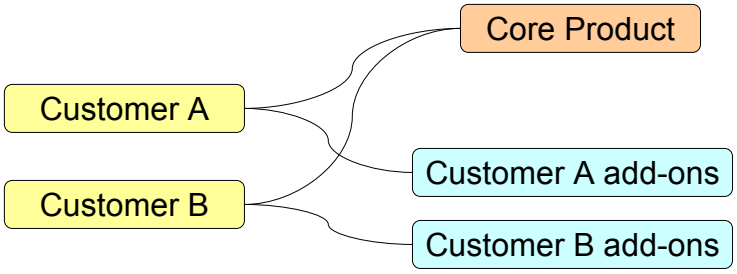
Pattern: Tailored Core Product

- Core product
  - any appropriate structure
  - generally not re-used code
- Customer-specific modifications
  - implemented by branched development of core
- Customer-specific add-ons
  - implemented by re-usable subprojects

Tailored Core – typical situation

- Large product, small library
- Large % code base commonality
- May have long complex release history for each customer
- Typically small number of customers, intensive relationship
- Typical example – industrial automation

Tailored Core – Diagram



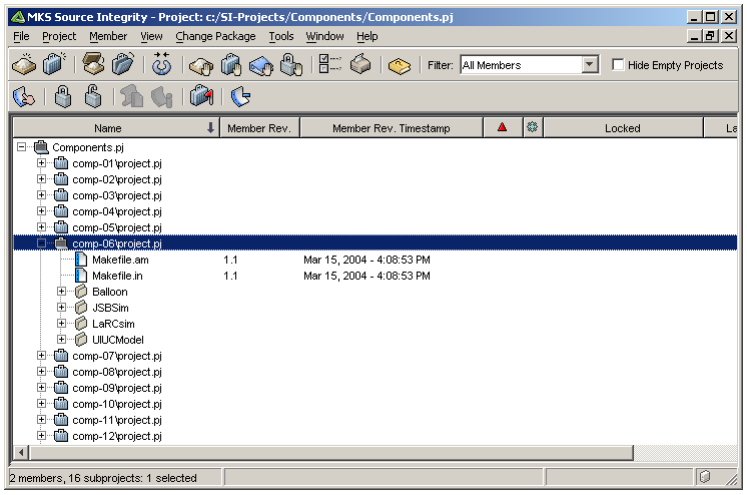
Build Better Software™

Component Based SCM - Tool Support

- Hierarchical project structure
- Branching at project level
- Re-useability at subproject level
  - many-to-one project-subproject linkage
- Tool examples use MKS Source Integrity Enterprise

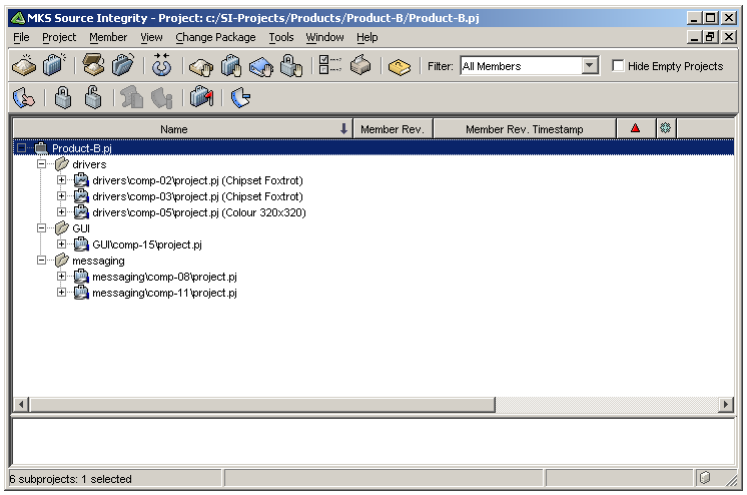
Build Better Software™

Component Palette



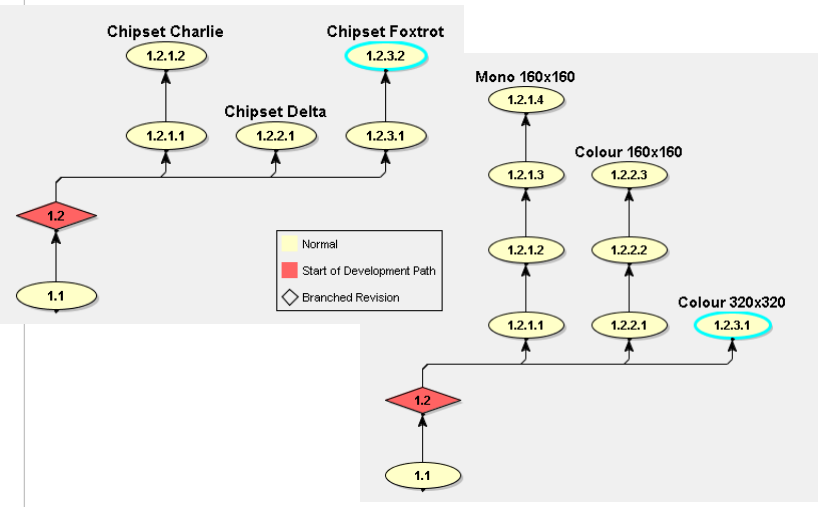
Build Better Software™

Product using components

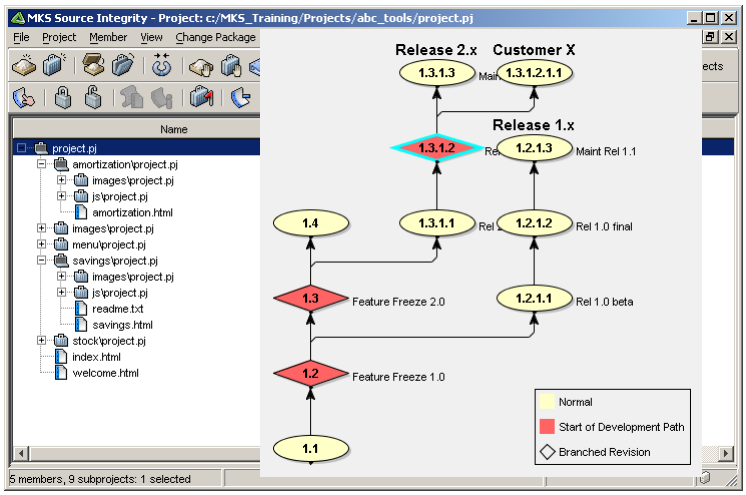


Build Better Software™

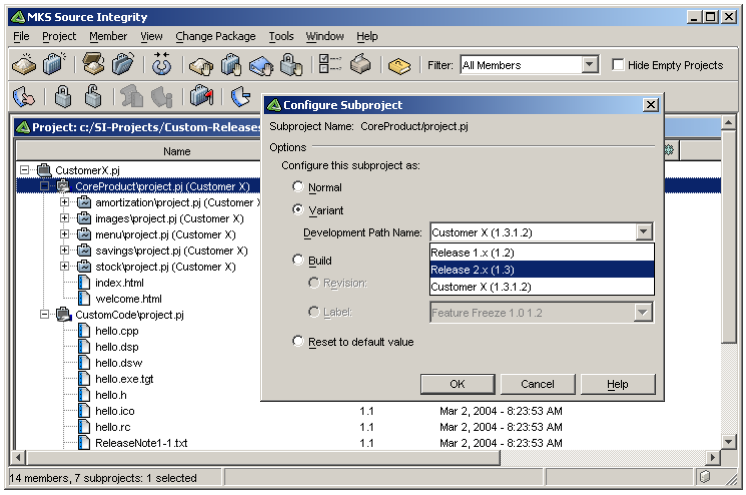
Independent Component History



Core Product



Customer "Container Project"

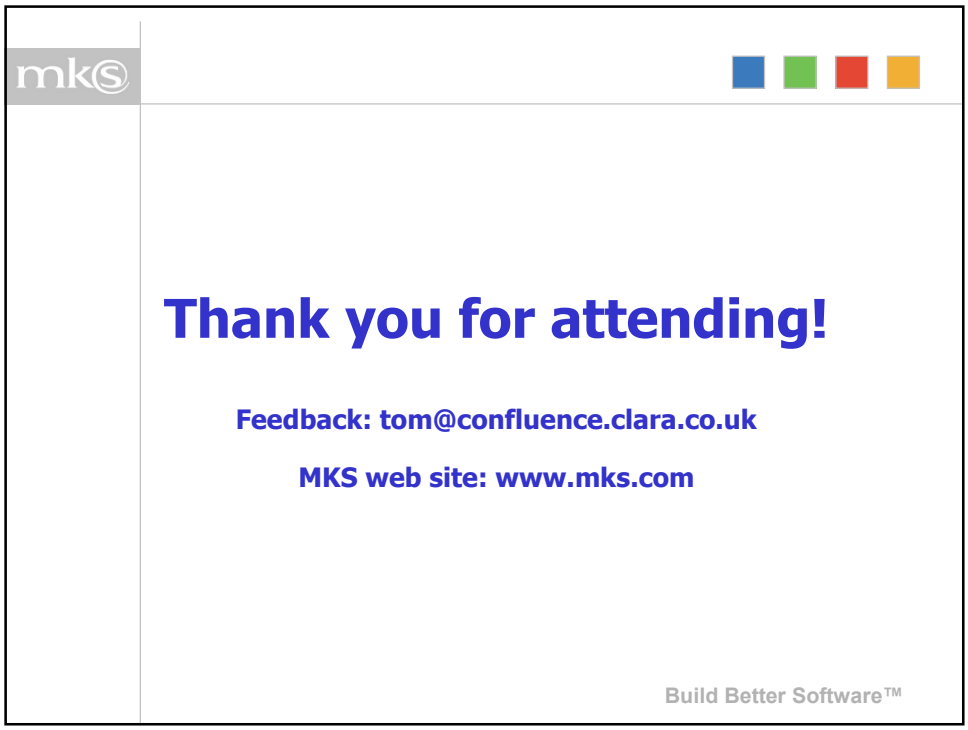


Summary

Component Based SCM achieves:

- Responsiveness
  - fulfilling customer needs
- Re-useability
  - maximising efficiency by avoiding duplicated effort
- Flexibility
  - easily matching configurations to requirements

= COMPETITIVE ADVANTAGE!


mks
Thank you for attending!
Feedback: tom@confluence.clara.co.uk
MKS web site: www.mks.com
Build Better Software™