Rijo Mathew
3,338 posts since
Oct 10, 2005
Currently Being Moderated
2. Jun 5, 2009 8:57 AM in response to: athat90

Re: What are the architectural differences between Documentum ver 5.x & 6.x

1) Significantly enhanced configurability throughout the stack
2) New SOA API
3) Eclipse-based development tooling
4) Enhanced scalability through Branch Office Caching
5) Reporting as part of the Platform
6) Repository virmalization (3rd party repository management

1) Significantly enhanced configurability throughout the stack
Presets - Dramatically reduce need for coding, Customize common settings, Contextual display based on user, Specify object typeACLs, formats, actions, lifecycles, workflows base attributes

Aspects - Provide alternate method for adding properties, Not bound to object types, Tend to be content-centric, not functional, Targeted for developers in D6

Simpler deployability through Java API - Native Java API for easier deployments, better logging, faster performance, increased supportability

2)New SOA API

DFS - Cohesive SOA approach, Decouples ECM functionality from Documentum terminology, Balances building content applications and integrating with non-content applications

3)Eclipse-based development tooling

Composer - Elcipse based IDE, Standard Platform, Open and extensible, Project based Development process

4)Enhanced scalability through Branch Office Caching

Predictive Caching - push subset of content to remote BOCS servers based on: content locations, properties, users, work queues

Write-back Support - allow for writes to local BOCS cache for both import and check-in

5)Reporting as part of the Platform

Reporting Services includes top 20 OOTB reports. Will ship with report engine
Users can tailor pre-built, reports or create their own, Access through JDBC driver
Report Abuse **Reply**

Rijo Mathew
3,338 posts since
Oct 10, 2005
Currently Being Moderated
3. Jun 5, 2009 9:00 AM in response to: athat90

Re: What are the architectural differences between Documentum ver 5.x & 6.x

Java DMCL has been replaced with D6 DFC

D6 DFC - Java DMCL

Deployment

- Enable EAR/War File Deployment for WDK Apps in D6
- Allow IT personnel to deploy WebApps using App Server tools

Simplify Clustered and HA deployments
Accelerate support for new App Server versions
Allow "non-exploded" WAR file deployment for better performance
Sandbox DFC per Application to avoid application conflicts
Simplify troubleshooting by removing transition from Java to C++
¯ Reliability
Pure Java solution
Eliminate C++ DMCL to meet IT reliability goals
¯ No more core dumps on application server tier
¯ No more memory corruptions on application server tier
Cross-platform single code line

Architecture

- Re-architect for deployment in JEE environment

¯ Optimized for server side deployment
¯ Global caches vs. session based caches
Consolidate similar DFC and DMCL facilities
¯ Session management
¯ Object caching
¯ Distributed docbase support

- Allows support for 64 bit OS

Simplified programming model

- Disable Bypassing of BOF logic by direct DMCL use

Decoupling persistent objects from sessions

- Easy debug-ability DFC J-DMCL stack

Performance

- Removed marshalling and un-marshahng data between Java and DMCL (through JNI)
- Highly reusable caches

¯ Session based caches are not sufficient on the middle tier

- Removed duplicate processing for similar functionality between DFC and DMCL
- Less memory consumption
- Performance gain is estimated from 20% to 40% depend on application