

SDK - Software Development Kit - v.1.0

Processo Metadados

Added by [Luís Guilherme Havryluk](#), last edited by [Luís Guilherme Havryluk](#) on Jan 30, 2012 ([view change](#))

Labels

[incubado](#), [processo](#), [cql](#), [metadados](#)

documento **incubado**

Processo (C)onstrução/(Q)ualidade/(L)iberação Metadados

Introdução

Nesse documento será apresentado o funcionamento e as ferramentas auxiliares para o andamento correto do processo de construção/qualidade/liberação de metadados do produto Datasul.

O que é?

O processo CQL é uma divisão de contextos de trabalho na linha de produção de metadados do produto Datasul. Cada contexto tem suas peculiaridades e ambientes específicos como mostrado abaixo (itens no dia em que esse documento estava sendo escrito):

(C)onstrução

Para construir metadados na linha datasul é necessário ter um usuário válido internamente e utilizar o ambiente correto. Esse ambiente, também chamado de desenvolvimento corporativo, utiliza uma base progress e mantém todos os metadados do produto, isto é, os que estão em construção/qualidade/liberação.

(Q)ualidade

Esse ambiente é montado em tempo de alpha testes e a base utilizada é o oracle. No momento em que esse ambiente é colocado no ar os objetos gerados na base metadados são instalados automaticamente (Deploy Automático) preparando o ambiente para testes.

(L)iberação

No contexto de liberação o ambiente é um repositório dentro da ferramenta de controle de versões o TFS. Esse repositório é copiado para a mídia para que os metadados sejam liberados.

Como funciona?

- **Necessidade de sustentação ou inovação:** Equipes recebem chamados e verificam a necessidade de inovação sustentação.
- **Analistas/Desenvolvedores das equipes levantam requisitos:** A partir do chamado os requisitos são levantados e analisados e o objeto metadados entra em fase de (C)onstrução.
- **Desenvolvedor gera o metadados em (C)onstrução e notifica analista da finalização:** Após a (C)onstrução do metadados o objeto deve ser colocado em (Q)ualidade para testes.
- **Analista coloca objeto e todas as suas dependências em (Q)ualidade:** Utilizando a ferramenta gerenciador de estados os objetos e dependências são colocados em qualidade.
- **Testadores utilizam o ambiente de (Q)ualidade para validar/verificar o metadados:** O ambiente de (Q)ualidade é gerado automaticamente para os testers verificarem/validarem os objetos e confirmarem se o objeto pode ser liberado.
- **Analista (L)ibera o metadados:** Uma vez confirmado o teste o metadado é (L)iberado pelo analista e a integração contínua gera o repositório de (L)iberação.
- **Metadados é empacotado na mídia:** Através dos procedimentos da integração contínua os metadados são disponibilizados na mídia que vai para o portal e cliente.

Quais as ferramentas de apoio?

Para auxiliar o andamento e a execução do processo existem as seguintes ferramentas de apoio:

- **Gerenciador de Metadados:** Ferramenta utilizada para desenvolver metadados. Contém os tipos de objetos e versionamento para o correto desenvolvimento para o produto. Nessa ferramenta que o processo começa.
- **Gerenciador de Estados:** Ferramenta para passar os objetos de uma fase para a outra. Esse procedimento ocorre com mudança de atributos na base de dados.
- **Plugin Exportador:** De acordo com os atributos alterados em base de dados essa ferramenta gera a exportação em xml dos metadados. O Controle de delete/add/update de arquivos é feito automaticamente pela ferramenta.
- **Plugin Importador:** Com os dados exportados corretamente o importador gera a base de dados metadados. Algumas validações de integridade são realizadas nessa ferramenta para que o deploy do produto funcione corretamente.
- **Deploy Automático:** Com a base de dados metadados gerada o deploy automático instala todos os objetos e deixa o ambiente pronto para uso. Essa ferramenta realiza alguma validações e é executada no start do Servidor de Aplicação.

Como tudo se integra?

Os colaboradores e as ferramentas são importantíssimos para o correto funcionamento do processo. Para tudo ficar integrado temos um ambiente chamado Integração Contínua responsável por rodar o plugins, compilar os projetos e gerar pacotes.

Sendo assim, com o metadados não é diferente e a Integração Contínua, no contexto do processo, tem as seguintes responsabilidades:

- Rodar o plugin exportador 16hrs/20hrs/22hrs para gerar todos os repositórios (desenvolvimento, qualidade e liberação) de acordo com os atributos da base de dados modificados pelo Gerenciador de Estados.
- Rodar o plugin importador (sequencial ao exportador de qualidade).
- Empacotar o EAR do produto e instalar para deploy no Servidor de Aplicação.
- Gerar a mídia do produto.

No entanto, nem tudo é automatizado e os colaboradores tem um papel fundamental com as seguintes responsabilidades:

- Desenvolvimento/Análise dos produtos metadados.
- Gerenciar os Estados dos objetos para que esses sigam o processo corretamente.

[Home](#)

[Específicos](#)

[Flex](#)

[Getting Started](#)

[Java](#)

[Processo Datasul](#)

[Progress](#)

[Tools](#)

[Trihas](#)

[User](#)

[Page Operations](#)

[Browse Space](#)

- Investigação do processo para verificar/validar se os objetos estão corretamente instalados.
- Testes dos metadados em ambiente de qualidade.

Como vai para a mídia?

A Integração Contínua tem a responsabilidade de montar a mídia do produto, assim, é dela a responsabilidade disponibilizar os metadados liberados na mídia e esse procedimento ocorre da seguinte forma:

1. Os colaboradores liberam os metadados pela ferramenta Gerenciador de Estados.
2. A Integração Contínua executa o plugin exportador para liberação gerando repositório metadados-liberation no controle de versão (hoje o TFS).
3. A Integração Contínua disponibiliza o repositório metadados-liberation dentro da mídia.

Como o cliente coloca no ar?

O cliente recebe a mídia via portal ou outra forma de entrega e instala. Quando instalado o produto está pronto e preparado para gerar a instalação dos metadados e isso ocorre no primeiro start do Servidor de Aplicação.

Caso a instalação seja realizada com sucesso, o deploy automático não é mais executado passando a ser um processo manual(é preciso modificar flags manualmente no produto para que o processo seja executado novamente). Em caso de erro, o processo fica no estado de início e toda vez que o start do Servidor de Aplicação ocorra o deploy é executado.

Com o procedimento de deploy automático temos o conceito de retornar ao padrão de fábrica que é muito útil em algumas situações em que o cliente altera e o produto não funciona.

Quais os problemas conhecidos?

1. **Processo de Desenvolvimento do Progress difere do processo CQL:** O processo de desenvolvimento em progress não tem o mesmo controle e estados que os metadados o que gera algumas inconsistências, principalmente no APPServer Progress, pois um metadados está diferente do objeto progress que ele utiliza.
2. **Dependência de Ambientes principalmente os automáticos:** O processo é dependente dos ambientes para o andamento dos objetos e se algum desses ambientes falham o processo é comprometido impactando o produto.
3. **Deploy Automático impede que o sistema funcione(roadmap):** O deploy automático quando dá erro ele derruba o container WEB do Servidor de Aplicação deixando a aplicação inacessível.
4. **Controle de Versão difere do progress e metadados(roadmap):** O controle de versão do progress não está de acordo com o padrão metadados o que pode gerar problemas.

Ilustração

Abaixo segue a ilustração do funcionamento do processo CQL.



Informações sobre este Documento

Versão	Estado	Compatibilidade	Criador	Criação	Último Revisor	Última Revisão
29	incubado	\$param0	Luís Guilherme Havryluk	Set 14, 2011 17:32	Luís Guilherme Havryluk	Jan 30, 2012 12:48



Your Rating:

Results:

0 rates

11.5.0-SNAPSHOT e superiores

Children (7) [Hide Children](#) | [View in Hierarchy](#)

-  [1. Gerenciamento de Estados C.Q.L](#)
-  [2. Cadastro e Associação de Versões](#)
-  [3. Deploy e Contextos de Trabalho](#)
-  [4. Exportador de Metadados](#)
-  [5. Reverter Versões](#)
-  [6. Importador de Metadados](#)
-  [7. Merge de Metadados](#)

 [Add Comment](#)