

SDK - Software Development Kit - v.1.0

Práticas de Programação

Added by [Poliane Ketterine Valdrich](#), last edited by [Poliane Ketterine Valdrich](#) on Feb 14, 2012 ([view change](#))

Labels
[incubado](#), [processo](#)

Comentários

documento incubado

Use comentários no código sempre que considerar apropriado. Se você parou para pensar para programar vale a pena documentar, ou seja, qualquer trecho de código que sua elaboração exigiu um esforço requer ao menos uma simples explicação.

Estrutura do Código

"Comece pelo começo, siga até chegar ao fim e então, pare." — Lewis Carroll em Alice no País das Maravilhas

Para evitar a proliferação de código emaranhado transformando nossos códigos ABLScript em verdadeiros spaghetti codes recomenda-se estruturar o código conforme a sequência abaixo:

1. Header
Comentário contendo o objetivo do programa
2. Variable definitions
Declaração de variáveis locais e compartilhadas — quando esta área contiver muitas variáveis e tornar o código muito longo recomenda-se o uso de includes
3. Other definitions (exemplo: temp-tables)
Definições das tabelas temporárias que serão usadas devem ser feitas dentro de includes para garantir o reuso onde necessário minimizando os esforços de cópia (control + <c>, control + <v>)
4. Corpo do Script
5. Exit Point

Tamanho do Código

Um código muito grande em geral indica um problema em potencial (*code smell*). Normalmente este tipo de código desempenha mais de uma função e sua complexidade tende a crescer rapidamente tornando difícil o entendimento do comportamento por parte de outros desenvolvedores. Recomenda-se refatorar códigos longos criando procedures que possuem um único objetivo e possuem um ciclo de vida curto.

Boa Prática!

Perceba se seus códigos ABLScript estão muito longos, pois eles podem identificar se está sendo desenvolvida uma regra de negócio no metadados. Regra de negócio nunca deve ser implementada no código ABLScript, pois deveriam estar numa BO.

Qual o tamanho recomendado?

Recomenda-se que um código não ultrapasse uma tela, ou seja, **para ler o código e entender a funcionalidade não deve ser necessário fazer scroll**. Se a área de declaração de variável tomar muito espaço do código ela deve ser realizada dentro de um include - simplificando a leitura. Os demais trechos do código devem ser quebrados usando procedimentos (procedures) de forma que cada procedure seja responsável por uma única tarefa.

Atenção!

Para garantir um padrão uniforme o metadados em breve bloqueará procedures com mais de 200 linhas de código - em um primeiro momento gerando um warning e futuramente bloqueando a edição.

Uso de Temp-tables

A quantidade de campos utilizados em uma temp-table e a quantidade de registros retornados por uma consulta podem ocasionar problemas e risco de travar o APP Server, Flex ou até mesmo a JVM. Para tabelas de movimento as consultas

devem sempre ser feita sobre agrupamentos de dados (sumários) de forma a evitar um uso excessivo da memória do servidor.

Identação

Para melhorar legibilidade todo código deve ser indentado, ou seja, statements dentro de um bloco recomenda-se que sejam usados três espaços para indentar.

⊖ Atenção!

Se usar qualquer outro editor evite usar TAB — ou o configure para trocar o caracter especial por espaços, pois o editor do metadados trabalha com espaços.

Outros Materias

[Expand all](#) [Collapse all](#)

- [Regras](#)
- [Nomenclatura](#)
- [Práticas de Programação](#)
- [Dicas de Performance](#)
- [Regras de Interface](#)
 - [Mensagens em Tela](#)
 - [Validações](#)
 - [Parametrização de Formulários](#)
 - [Chamadas ao Servidor](#)
 - [Processos Demorados](#)
 - [Dicas de Interface](#)

Informações sobre este Documento

Versão	Estado	Compatibilidade	Criador	Criação	Último Revisor	Última Revisão
5	incubado	SDK 1.0.0	Poliane Ketterine Valdrich	Fev 08, 2012 16:42	Poliane Ketterine Valdrich	Fev 14, 2012 09:17



Your Rating:

Results:

0 rates

[O que é RSS?](#)

[Add Comment](#)