**Manual de utilização do TortoiseSVN**

O TortoiseSVN é um software de código-fonte aberto, que serve de interface cliente para o sistema de controle de versão Subversion. Ambos são desenvolvidos sob a GNU General Public License (GPL). O TortoiseSVN acrescenta funcionalidades ao Windows Explorer tornando a interação com o repositório central do servidor de controle de versões Subversion mais amigável.

**1. Procedimento diário**

Após o repositório ter sido criado e o TortoiseSVN for ser utilizado pela primeira vez deve-se obter a estrutura de diretórios do repositório, para tanto, utiliza-se o comando “Obter” (checkout) (vide capítulo 3).

Uma vez criada a cópia de trabalho com os artefatos é necessário toda vez que iniciar o trabalho atualizar o repositório usando o comando “Atualizar” (*update*), pois o usuário pode ter alterado o artefato no repositório.

Quando terminar o dia utilize o comando “Submeter” (*Commit*) para atualizar os artefatos no repositório.

**2. SVN Importar (import)**

O comando “importar” (*import*), do TortoiseSVN, mostrado na Figura 1, objetiva criar uma hierarquia de diretórios no repositório de forma completa em uma única ação. Embora o comando execute o trabalho de forma rápida e econômica ele tem as seguintes desvantagens:

         não existe forma de selecionar arquivo e pastas a serem incluídas. A sub-árvore local selecionada é importada com seu conteúdo sem a possibilidade de excluir ou incluir arquivos;

         o diretório importado não se torna automaticamente a cópia de trabalho local. Você precisará executar o comando “obter” (*checkout*) para obter uma cópia de trabalho devidamente ligada com o repositório.

         é (muito) fácil importar em um nível de diretório errado no repositório.



**Figura 1** – Menu contextual mostrando o comando “*Import*” NÃO RECOMENDADO PARA EXECUÇÂO

*Por isso o manual da ferramenta recomenda que o comando “Import”****NÃO DEVE SER UTILIZADO****.*

Além disso a estrutura de diretório está formalmente definida pelo Plano de Gerência de Configuração. Este documento é preenchido na atividade “Planejar a Gerência de Configuração” na Fase de Planejamento do projeto.

Portanto para iniciar a população do repositório com arquivos de trabalho você deve primeiramente executar um “*Checkout*” (próximo tópico deste roteiro) para obter a estrutura de diretórios criada pela área de suporte da empresa conforme padrão definido no **Plano de Gerência de Configuração**.

Por fim execute o comando “*Add*”, mostrado na Figura 2, para incluir os arquivos locais na próxima sincronização (“*Commit*”) com o repositório. 

**Figura 2** – Opção “*Add*” do menu contextual da ferramenta TortoiseSVN.

Por fim execute o comando “*Commit*” (veja tópico mais a frente neste roteiro) para confirmar o *upload* e sincronização da cópia local com o repositório.

**3. SVN Obter (checkout)**

Objetiva baixar um conjunto de arquivos (diretório) do repositório de controle de versões para uma pasta no computador local.

**3.1 Realizando o obter (checkout)**

Em uma pasta local, onde se deseja baixar o conteúdo do repositório, clique com o botão direito do mouse e selecione a opção “SVN Obter” ou “SVN Checkout”.



**Figura 3** - Selecionando a opção Checkout (Obter)



**Figura 4** - Informando o diretório do projeto que se deseja obter uma copia dos arquivos

Preencher o campo “URL of repository” acima com o endereço do projeto:

**http://cri.servehttp.com/crirepository/projetos/[projeto]/tags/**

Este campo é utilizado para indicar o caminho onde se encontra o diretório dos arquivos do projeto sob controle de versões do SVN.

Clique em “**OK**”.



**Figura 5** - Janela de autenticação do usuário do repositório SVN

Preencher o campo acima com sua conta e senha.

Aguarde a operação e clique em “**Ok**”.



Após ter feito o “Checkout” de um repositório para uma pasta, sempre que for acessado o menu de contexto dentro da pasta aparecerão todas as operações que poderemos realizar no repositório. As duas principais funções são: SVN Commit que grava todos os dados da pasta local para o repositório e SVN Update que traz os documentos do repositório para a pasta local.

**4. SVN Atualizar (update)**

O objetivo é atualizar a cópia local com os arquivos mais atualizados existentes no repositório. Este comando afeta somente os arquivos na sua maquina local e, portanto, não tem efeito nenhum sobre o repositório.

**4.1 Realizando o atualizar (update)**

Selecione a pasta local , clique com o botão direito do mouse e selecione a opção “SVN Atualizar” ou “SVN Update”.







Preencher os campos com sua conta e senha.

Aguarde a operação e clique em “Ok”.

**4.2 Resolvendo conflitos**

Ao tentar atualizar o repositório com o comando “Submeter” (Commit) pode ocorrer erro, veja na tela abaixo:



Ao utilizar o comando “Atualizar” (Update) no repositório, vão ser criadas quatro cópias do arquivo.



< nome\_do\_arquivo > .mine (arquivo com conteúdo que se quer atualizar)

< nome\_do\_arquivo > .r -1 (revisão anterior)

< nome\_do\_arquivo > .r (revisão atual)

< nome\_do\_arquivo > .extensão (arquivo com conteúdo mesclado da revisão atual no repositório com o conteúdo que foi alterado)

Neste ultimo arquivo são colocados as alterações cujo conteúdo se quer atualizar (< nome\_do\_arquivo > .mine) mesclado com o concluído presente no repositório (< nome\_do\_arquivo > .r).

Deve-se realizar as mudanças necessárias no arquivo para acrescentar as modificações que se quer, excluindo as linhas desnecessárias, vide exemplo abaixo:



Após modificação o conteúdo ficará como no exemplo abaixo:



Para resolver o conflito vá em tortoiseSVN > resolvido.



Será exibida a seguinte mensagem abaixo e apenas o conflito que se quer resolver deve ser selecionado.



Ao selecionar o comando “Ok” serão excluídos os três arquivos abaixo:

< nome\_do\_arquivo > .mine (arquivo com conteúdo que se quer atualizar)

< nome\_do\_arquivo > .r -1 (revisão anterior)

< nome\_do\_arquivo > .r (revisão atual)

E será exibida a seguinte informação, relatando que o conflito foi resolvido.



Após solucionado o conflito o repositório poderá ser atualizado.

**5. SVN Submeter (commit)**

O objetivo é atualizar os arquivos do repositório com as versões alteradas de sua cópia local.

NOTA – Também conhecido como checking in.

**5.1 Realizando o submeter (commit)**

Selecione a pasta local, clique com o botão direito do mouse e selecione a opção “SVN Submeter…” ou “SVN Commit…”, a tela “Enter Log Message – [nome da pasta]” , será exibida conforme abaixo:



SEMPRE PREENCHA O CAMPO Message COM INFORMAÇÕES RELEVANTES SOBRE A TAREFA DE ATUALIZAÇÃO DO REPOSITÓRIO QUE ESTÁ SENDO EXECUTADA.

• Clique em “Ok”. Abrirá a tela ” \* - submeter – TortoiseSVN Concluído!” ou “\* TortoiseSVN Commit…..Finished! ”



* Clique em “**Ok**”.

**6. SVN Verificar Alterações (check of modifications)**

Objetivo é identificar as modificações realizadas nos arquivos locais em relação a versão armazenada no repositório.

**6.1 Realizando a verificação das alterações (check of modifications)**

Selecione a pasta local, clique com o botão direito do mouse e selecione a opção “TortoiseSVN → Verificar Alterações” ou “TortoiseSVN - > Check for Modifications”, a tela “Cópia de trabalho” ou “Working copy” será exibida conforme abaixo:



* Clique em “Verificar repositório” ou “*Check repository*”.
* Aparecerá todos arquivos que sofreram alteração no servidor para que se possa analisar, conforme abaixo:



**7. Solução de problema de renomeação de arquivos no repositorio**

É fato documentado no TortoiseSVN que existe um problema no Windows. O sistema operacional Windows é reconhecidamente CASE-INSENSITIVE, isto é, não faz distinção entre maiúsculas e minúsculas nos nomes dos arquivos. Esta característica deve ter sua razão para guardar compatibilidade com as versões anteriores (MS-DOS).

Portanto quando se tenta utilizar a função *Rename* do menu contextual do TortoiseSVN alterando-se apenas a caixa das letras do nome do arquivo obtemos o aviso (warning) que pode ser visto abaixo:



A diretriz de utilizar apenas letras minúsculas é tentadora mas apresenta pelo menos dois problemas imediatos:

**a.** O Plano de gerência de configuração define nomes que utilizam letras maiúsculas e a sua alteração prejudicaria muito o entendimento alem de incluir varios caracteres sublinhados nos nomes.

**b.** Os nomes dos arquivos fontes de classes em Java são case-sensitive e a utilização somente letras minúsculas é contraria ao padrão de fato na programação Orientada por Objetos.

Portanto o TortoiseSVN recomenda uma solução de contorno para o problema do Windows da seguinte forma:

1. Utilize a opção ”*Rename*” do TortoiseSVN e altere o nome do arquivo acrescentando um caracter sublinhado ao final do nome.
2. Faça o Commit da alteração.
3. Faça um Update.
4. Utilize a opção ”*Rename*” do TortoiseSVN e altere a caixa das letras que deseja mudar e também retire o caractere sublinhado incluído anteriormente.
5. Execute o Commit.
6. Execute um Update.

**8. Geração de baseline (Rótulo ou Tag)**

Uma das características dos sistemas de controle de versão é a capacidade de se criar uma cópia de uma versão específica no repositório (Por exemplo: criar uma versão chamada 1.5), assim você pode a qualquer momento, recriar e construir um certo ambiente. Esse processo é conhecido como baseline (Tag).

Se você importou seu projeto com a estrutura do diretório recomendado, a criação de uma baseline (Tag) é muito simples:



Selecione a pasta na sua cópia de trabalho que você deseja copiar para uma tag, em seguida, selecione o comando TortoiseSVN → *Branch / Tag.*

A URL padrão de destino será a URL de origem no qual sua cópia de trabalho se baseia. Você precisará editar a URL para o novo caminho para a sua Tag. Assim em vez de[http://servehttp.com/repository/processo](http://cri.servehttp.com/crirepository/processo) você poderá usar algo como <http://servehttp.com/repository/processo/tags/1.0.0.1>.

Se você não se lembra da convenção de nomes que você usou na última vez, clique no botão à direita para abrir o navegador repositório para que você possa visualizar a estrutura de repositório de dados.Agora você tem que selecionar a cópia da fonte. Aqui você tem três opções:

1) Revisão geral no repositório

O novo branch é copiado diretamente no repositório da revisão geral. O dados não precisam ser transferido de sua cópia de trabalho, e o branch é criado rapidamente.

2) Revisão específica no repositório

O novo branch é copiado diretamente no repositório mas você pode escolher uma revisão mais antiga. Isso é útil se você se esqueceu de fazer uma marca quando você lançou o projeto na semana passada. Se você não consegue lembrar o número de revisão, clique no botão à direita para mostrar o log de revisão, e selecione o número de revisão de lá. Novamente nenhum dado é transferido de sua cópia de trabalho, e o branch é criado rapidamente.

3) Cópia de trabalho

O novo branch é uma cópia idêntica de sua cópia de trabalho local. Se você atualizou alguns arquivos para uma revisão mais antiga em sua cópia de trabalho, ou se você tiver feito alterações locais, que é exatamente o que vai para a cópia. Naturalmente, este tipo de tag complexo pode envolver a transferência de dados a partir da sua cópia de trabalho de volta para o repositório, e se não existe passará a existir.

Se você quiser que sua cópia de trabalho seja transferida para o recém-criado branch automaticamente, use a opção cópia de trabalho e marque branch/tag. Mas se você fizer isto, primeiro certifique-se que sua cópia de trabalho não contém modificações. Se isso acontecer, essas alterações serão mescladas no branch da cópia de trabalho quando você mudar.

Pressione OK para confirmar a nova cópia para o repositório. Não se esqueça de fornecer uma mensagem de log. Note-se que a cópia é criada dentro do repositório.

Note que a menos que você optou por mudar a sua cópia de trabalho para o branch recém-criado, com a criação de um branch ou Tag isso não afetará a sua cópia de trabalho. Mesmo se você criar um novo branch da sua cópia de trabalho, essas alterações serão geradas como um novo branch, não para o trunk, para que a sua cópia de trabalho ainda possa ser marcada como modificada em relação ao trunk.

**9. Mesclando arquivos (Merge)**

Quando os *branches* são usados para manter linhas separadas de desenvolvimento, em algum momento você vai querer mesclar as alterações feitas em um *branch* de volta para o *trunk*, ou vice-versa.

É importante compreender como *branching and merging* funcionam no Subversion antes de começar a usá-lo, pois ele pode se tornar bastante complexo.

O próximo ponto a observar é que o *merging* ocorre sempre dentro de uma cópia de trabalho. Se você deseja mesclar as alterações em um *branch*, você tem que ter uma cópia de trabalho para esse *branch*fazendo um *check-out*, e chamar o assistente direto de uma cópia de trabalho usando o TortoiseSVN → Merge…

Em geral é uma boa idéia realizar um *merge* em uma cópia de trabalho não modificado. Se você fez outras mudanças em seu documento e faz o *commit* primeiro e o *merge* não foi como esperado, você pode querer reverter as mudanças, e o comando *Revert* descartará todas as mudanças, incluindo algumas que você fez antes do *merge*.