

Guia do Usuario

Requisitos

O svntool utiliza alguns recursos externos para seu funcionamento. A lista segue abaixo:

No Windows 32bits:

- Python, instalado e no path do Sistema Operacional.
- Apache Ant, instalado e no path do Sistema Operacional.
- Robocopy (ou SyncToy), instalado e no path do Sistema Operacional.
- Subversion, instalado e no path do Sistema Operacional.

No Windows 64bits:

- Python, instalado e no path do Sistema Operacional.
- Apache Ant, instalado e no path do Sistema Operacional.
- SyncToy, instalado e no path do Sistema Operacional.
- Subversion, instalado e no path do Sistema Operacional.

No Linux 32 ou 64 bits:

- Python, instalado e no path do Sistema Operacional.
- Apache Ant, instalado e no path do Sistema Operacional.
- RSync, instalado e no path do Sistema Operacional.
- Subversion, instalado e no path do Sistema Operacional.

Ainda, algumas dependências extras existem, como o uso do SvnStat, StatSvn, Ant-Contrib e SvnAnt. Porém, essas bibliotecas já estão incorporadas dentro do diretório /lib da instalação.

NOTA: Assim como o svntool, todas as suas dependências não possuem custo de licenciamento.

Instalação

1. Baixe a ultima versao compactada do svntool em TODO.
2. Descompacte o svntool em um diretorio, como /usr/local/svntool ou c:\tools\svntool. Esse diretorio sera chamado simplesmente de svntool_home a partir de agora.
3. Edite o arquivo svntool_home/build.properties, ajustando as entradas para o seu ambiente operacional. O arquivo é auto-explicativo.
4. Para testar, acesse svntool_home e digite "ant -p". A saida deve ser algo como abaixo:

```
Buildfile: build.xml
```

```
Main targets:
```

```
admin.create CORE: Cria um repositorio em uma area especifica
admin.help CORE: Help do subcomando 'admin'.
help CORE: Help da ferramenta 'svntool'.
report.generate CORE: Gera relatorios estatisticos a partir de um repositorio
report.generate.checkout AUX.: Efetua checkout de um repositorio
report.help CORE: Help do subcomando 'report'.
svntool.dist CORE: Cria uma distribuicao do svntool no arquivo 'svntool.zip'.
sync.help CORE: Help do subcomando 'sync'.
sync.run CORE: Inicializa e mantem sincronizados repositorios
sync.run.linux AUX.: Efetua a sincronizacao no Linux.
sync.run.windows AUX.: Efetua a sincronizacao no Windows.
sync.run.windows.robocopy AUX.: Efetua a sincronizacao via utilitario robocopy
sync.run.windows.synctoy AUX.: Efetua a sincronizacao via utilitario synctoy
Default target: help
```

As tarefas marcadas como "CORE" podem ser invocadas pelo usuário. As tarefas marcadas como "AUX" são internas da ferramenta e não devem ser invocadas pelo usuário, pois podem incorrer em mau funcionamento e/ou perda de dados.

Para obter ajuda sobre uma determinada tarefa, execute "ant tarefa.help", como abaixo:

```
D:\TRABALHO\Eclipse\3layer-public\incubator\svntool>ant admin.help
Buildfile: build.xml
```

```
admin.help:
\[\echo\] Parametros disponiveis: \-Darea=dir \-Drepo=repositorio
\[\echo\] onde:
\[\echo\] dir = Diretorio onde deve ser criado o repositorio.
\[\echo\] repositorio = Nome do repositorio a ser criado.
\[\echo\]
\[\echo\] EXEMPLO 1: Cria o repositorio 'incubator' na area 'jboss' com autorizacao por area.
\[\echo\]
\[\echo\] ant admin.create \-Darea=jboss \-Drepo=incubator
\[\echo\]
```

```
BUILD SUCCESSFUL
Total time: 0 seconds
D:\TRABALHO\Eclipse\3layer-public\incubator\svntool>
```

Objetivos e Funcionamento

O svntool possui três grandes objetivos, sendo os dois últimos focados na automatização através de ferramentas de agendamento do Sistema Operacional :

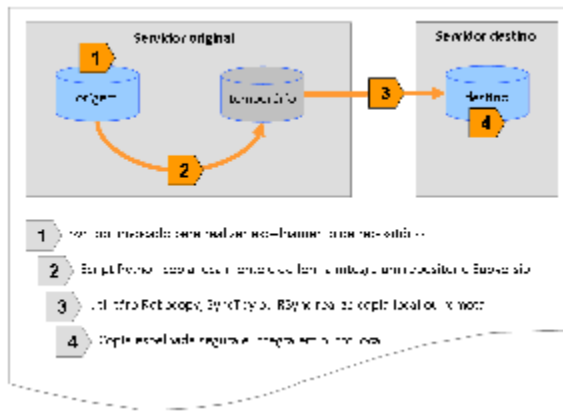
1. **Criação de repositórios:** Repositórios Subversion são criados com base em um repositório padrão (template), que pode ser configurado conforme a necessidade do projeto. Por padrão, o repositório template é um repositório vazio com a estrutura /branches, /tags e /trunk, típica para repositórios Subversion. Ainda, como forma de facilitar a estruturação, os repositórios criados pelo svntool seguem uma organização dividida por áreas, ou seja, diretórios separados. A figura abaixo facilita a compreensão:



Confor mostra a figura, cada área possui um arquivo de autorização em separado, o que permite um equilíbrio na configuração de segurança para vários repositórios que possuem a mesma semântica (objetivos), como "projetos públicos", "projetos privados", "projetos do cliente X", etc.

2. **Geração de relatórios:** Através dos utilitários SvnStat e StaSvn, diversos relatórios estatísticos em formato HTML são produzidos através de uma única chamada. Para um exemplo desses relatórios, acesse <http://3layer.com.br/svnreport>.
3. **Espelhamento de repositórios:** Repositórios podem ser espelhados (replicados) de um ponto para outro, tanto local quanto remotamente. Isso facilita muito o trabalho de ter múltiplas cópias de um repositório, e também funciona como forma de backup segura. Em ambientes Windows a replicação é feita com ou uso do ótimo utilitário Robocopy (somente para Win32) ou então com o SyncToy (Win32 ou Win64) e no Linux, isso é feito através do utilitário RSync. Nota-se que, devido a alta sensibilidade da estrutura dos repositórios Subversion, é necessário o uso de scripts

Python (do próprio site do Subversion) para evitar problemas durante o espelhamento. Em outras palavras, o processo de espelhamento é feito em duas partes, como mostra a figura abaixo:



Todas essas funcionalidades operam em ambientes Windows e Linux de 32 ou 64 bits de forma transparente.

Criacao de repositorios

A criação de repositórios geralmente é uma das primeiras tarefas a serem realizadas. No svntool, essa tarefa é facilitada, onde o administrador do sistema simplesmente invoca o comando "ant admin.create" com os parâmetros necessários.

Como fazer?

1. Invoca-se o comando "ant admin.create -Darea=dir -Drepo=nome", substituindo os valores de dir e nome pela área e o nome do repositório, respectivamente.

Exemplo

Criar o repositório teste, dentro a área de repositórios públicos.

```
ant admin.create \-Darea=public \-Drepo=teste
```

Como resultado, o repositório chamado "teste" será criado dentro do diretório "public" localizado dentro do caminho especificado pela variável "\$svn.root", localizada no arquivo svntool_home/build.properties. Em outras palavras, o caminho do repositório será "\$svn.root/public/teste".

Ao executar um comando como "svnlook info D:\DATA\SVN\public\teste", você verá uma saída como:

```
admin
2008-12-20 05:37:55 \-0200 (sáb, 20 dez 2008)
32
Estrutura inicial do repositorio
```

Ou seja, a estrutura criada para o repositório "teste" é baseada no repositório template do svntool, localizado dentro da pasta /resources.

Se você acha que precisa uma estrutura diferente para o repositório template, conecte neste repositório template e altere-o conforme sua necessidade.

Caso o repositório a ser criado já exista, nenhuma alteração é feita sobre ele.

Caso deseje não utilizar áreas para os repositórios, utilize o parâmetro de área como vazio, como -Darea="".

Geracao de relatorios

O SVNTool utiliza duas bibliotecas para produzir relatórios, incluindo elementos gráficos. São o SvnStat e StatSvn. Porém, o SVNTool abstrai a complexidade de uso desses utilitários, e para o usuário final a geração de relatórios se resume a execução de comandos, como:

```
/opt/lm2/apache-ant-1.7.1/bin/ant -f /opt/lm2/svntool/build.xml report.generate -Doutput=public/3pup -Drepo=file:///opt/lm2/data/svn/public/3pup -Drev=HEAD -Duser=user.admin -Dpass=xxx
```

Neste exemplo, o SVNTool é chamado através da linha de comando com a opção "report.generate", com o parâmetro "-Doutput" apontando para um diretório de saída, onde serão gerados os relatórios para o repositório especificado pelo parâmetro "-Drepo", usando a última revisão do repositório (opção "-Drev") com o usuário e senha especificados por "-Duser" e "-Dpass", respectivamente.

Um exemplo de saída desses relatórios pode ser conferido em <http://3layer.org/svnreport/public/incubator/svntool>.

Geracao continua de relatorios

Estando em ambiente Windows ou Linux, eh possivel utilizar as ferramentas nativas do Sistema Operacional para produzir continuamente os relatorios do SVNTool.

NO Windows, a linha de comando acima pode ser adicionada a uma entrada no servico Task Scheduler. No Linux o mesmo pode ser feito com o servico Crontab.

Abaixo, um exemplo de configuracao no Crontab no linux, que produz relatorios para diversos repositorios diariamente as 06:00 AM:

```
[root@jekyll ~]# crontab -e

# - -
# Relatorios diarios do SVN, diariamente as 06:00 AM
00 6 * * * /opt/lm2/svntool/cron.sh 1>> /var/log/cron_log 2>> /var/log/cron_error

[root@jekyll ~]# vim /opt/lm2/svntool/cron.sh

#!/bin/bash -x

# Define o locale para o SVN funcionar. Vide tarefa JBAG2A-20
LC_ALL=en_US.UTF-8
export LC_ALL

# - -
# public/3pup
/opt/lm2/apache-ant-1.7.1/bin/ant -f /opt/lm2/svntool/build.xml report.generate -Doutput=public/3pup -
Drepo=file:///opt/lm2/data/svn/public/3pup -Drev=HEAD -Duser=user.admin -Dpass=xxx
# - -
# public/magoo
/opt/lm2/apache-ant-1.7.1/bin/ant -f /opt/lm2/svntool/build.xml report.generate -Doutput=public/magoo -
Drepo=file:///opt/lm2/data/svn/public/magoo -Drev=HEAD -Duser=user.admin -Dpass=xxx

# - -
# public/merlin
/opt/lm2/apache-ant-1.7.1/bin/ant -f /opt/lm2/svntool/build.xml report.generate -Doutput=public/merlin -
Drepo=file:///opt/lm2/data/svn/public/merlin -Drev=HEAD -Duser=user.admin -Dpass=xxx

# - -
# public/mammon
/opt/lm2/apache-ant-1.7.1/bin/ant -f /opt/lm2/svntool/build.xml report.generate -Doutput=public/mammon -
Drepo=file:///opt/lm2/data/svn/public/mammon -Drev=HEAD -Duser=user.admin -Dpass=xxx
```

Espelhamento de repositorios



IMPORTANTE

O svntool suporta apenas repositorios do tipo FileSystem (FSFS), que sao os mais atuais e o o padrao recomendado pelo Subversion. Se voce utiliza repositorios baseados em BerkleyDB, nao utilize o svntool para fazer espelhamento, pois os resultados sao imprevisiveis.





IMPORTANTE

O svntool foi projetado para suportar uma estrutura de pastas de dois níveis, ou seja, um diretório superior que indicam as "áreas de repositórios" (parâmetro -Darea) e, dentro de cada uma dessas áreas, os repositórios em si (parâmetro -Drepo). Ao utilizar uma estrutura com níveis mais profundos de diretórios, o comportamento do svntool pode não ser o esperado. Então, muita atenção ao utilizar estruturas de diretórios diferentes dessa:

```
---Raiz
+---Area1
+---Repo1
+---Repo2
+---Repo3
+---RepoN
+---Area2
+---Repo11
+---Repo21
+---Repo31
+---RepoN
+---AreaN
```

No Windows

TODO

No Linux

No Linux, tem-se duas situações para o espelhamento, descritas abaixo:

Local-Local

É a forma mais simples de espelhamento, onde a cópia resultante é armazenada no próprio FileSystem do repositório de origem. Por ser local, não necessita o uso de autenticação remota e, portanto, o próprio usuário do processo chamador do svntool realiza a cópia.

Como fazer?

1. Invoque o comando "ant sync.run -Dsource=dir1 -Ddest=dir2", substituindo dir1 pelo caminho do repositório a ser espelhado e dir2 pelo local onde o repositório espelhado deve ser salvo.

Exemplo

Copiar o repositório /opt/3layer/data/svn/public/incubator para o caminho local /opt/3layer/backup/svn/public/incubator:

```
ant syn.run \-Dsource=/opt/3layer/data/svn/public/incubator \-Ddest=/opt/3layer/backup/svn/public/incubator
```

Não se preocupe em criar o diretório de destino, pois o próprio svntool se encarga disso.

Local-Remoto

Cópias que envolvem servidores remotos (na origem ou no destino) exigem autenticação no servidor remoto. O svntool, visando transparência para este processo, vale-se do RSync através de uma conexão SSH com o uso de chaves públicas e privadas sem senha para conexão.

É importante salientar que mesmo utilizando chaves sem senhas, o processo de conexão e tráfego de dados é seguro e criptografado. Leia mais sobre isso em TODO.

Como fazer?

PARTE 1 - Preparação da autenticação remota via SSH

1. Crie um par de chave pública/privada sem senha para o usuário que vai ser utilizado para se conectar no(s) servidor(es) remoto(s) e salve os arquivos gerados na pasta /resources do svntool com o nome svntool.key (par ao arquivo de chave privada) e svntool.key.pub (para o arquivo de chave pública). Veja abaixo uma sequência de comandos para isso:

```
[root@jekyll resources]\# pwd
/opt/lm2/svntool/resources
[root@jekyll resources]\# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa): /opt/lm2/svntool/resources/svntool.key
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /opt/lm2/svntool/resources/svntool.key.
Your public key has been saved in /opt/lm2/svntool/resources/svntool.key.pub.
The key fingerprint is:
18:df:10:96:3d:ec:79:77:4d:b3:b5:c7:09:70:52:6d root@jekyll.lm2.com.br
[root@jekyll resources]\#
```

2. Copie o arquivo gerado svntool_home/resources/svntool.key.pub para dentro do(s) servidor(s) remoto(s). Geralmente, usa-se a pasta /home/usuario/.ssh para salvar este arquivo. Observe que o usuario que vai ser usado para conexao precisa ter de leitura e escrita nos diretorios utilizados. Abaixo, um exemplo:

```
\[root@jekyll resources]\# scp svntool.key.pub marcelo.mrack@hermes:/home/marcelo.mrack/.ssh
marcelo.mrack@hermes's password:
svntool.key.pub 100% 404 0.4KB/s 00:00
```

3. Proteja a chave privada gerada para que outros usuarios nao possam utiliza-la indevidamente. Se isso nao for feito, é possível que o servidor remoto recuse a chave privada. Veja como fazer isso abaixo:

```
chmod 700 /opt/lm2/svntool/resources/svntool.key
```

4. Conecte-se no servidor remoto, e adicione o conteudo da chave publica copiada ao arquivo de chaves autorizadas para conexao remota. Geralmente, este arquivo é /home/usuario/.ssh/authorized_keys ou entao /home/usuario/.ssh/authorized_keys2. Veja abaixo uma sequencia para isso:

```
\[root@jekyll resources]\# ssh marcelo.mrack@hermes
marcelo.mrack@hermes's password:
Last login: Thu Jan 29 17:46:41 2009 from jekyll.lm2.com.br
\[marcelo.mrack@hermes ~]\$ cd /home/marcelo.mrack/.ssh/
\[marcelo.mrack@hermes .ssh]\$ ls
authorized_keys id_rsa.pub known_hosts myPublicKey-ssh2.key svntool.key.pub
\[marcelo.mrack@hermes .ssh]\$ cat svntool.key.pub > authorized_keys
\[marcelo.mrack@hermes .ssh]\$
```

5. Teste sua conexao SSH para o servidor remoto utilizando sua chave publica/privada e verifique se possui acesso de escrita no caminho onde o repositorio espelho vai ser armazenado. Veja abaixo um exemplo onde o usuario marcelo.mrack se conecta com a chave privada e navega ate o diretorio onde o espelhamento vai ser feito e cria um diretorio para salvar o espelhamento:

```
\[root@jekyll resources]\# ssh -i /opt/lm2/svntool/resources/svntool.key marcelo.mrack@hermes
Last login: Thu Jan 29 17:47:55 2009 from jekyll.lm2.com.br
\[marcelo.mrack@hermes ~]\$ cd /opt/backup
\[marcelo.mrack@hermes backup]\$ mkdir svn
\[marcelo.mrack@hermes backup]\$ pwd
/opt/backup
\[marcelo.mrack@hermes backup]\$ ls -la
total 16
drwxrwxr-x 3 root Domain Users 4096 Jan 29 17:56 .
drwxr-xr-x 6 root root 4096 Jan 29 17:55 ..
drwxr-xr-x 2 marcelo.mrack Domain Users 4096 Jan 29 17:56 svn
\[marcelo.mrack@hermes backup]\$
```

6. Pronto. Nesse momento é possível realizar a cópia remota de arquivos e pastas de forma segura através do SSH. Isso é a base para a replicação remota que o svntool utiliza com base no Rsync.

PARTE 2 - Executando a replicação com o svntool

1. No servidor de origem, execute o svntool informando o nome do repositorio a ser copiado e o destino remoto, como abaixo:

```
ant sync.run \-Dsource=lm2/admin \-Ddest=marcelo.mrack@hermes:/opt/backup
```

2. Pronto. No caminho hermes:/opt/backup deve existir uma estrutura idêntica ao repositório de origem, incluindo o nome da área especificada, caso houver. Em outras palavras, a cópia espelho estará em hermes:/opt/backup/lm2/admin.