



Tutorial de Preparação de um Instalador do CSGrid

Tecgraf/PUC-Rio
csgrid@tecgraf.puc-rio.br

Resumo

Esse documento é destinado a equipe de desenvolvimento e explica como gerar um instalador do CSGrid para ser distribuído. Assume-se que, para gerar o instalador, o desenvolvedor está trabalhando em uma máquina Linux do ambiente Tecgraf.



Sumário

1	Diretório de trabalho	4
1.1	Criar o diretório	4
1.2	Fazer check-out do Subversion	5
1.3	Fazer o build	5
2	Criação de um brach	7
2.1	Ir para o diretório do cmake	7
2.2	Executar o comando de criação de branch	7
3	Criação de um tag	8
3.1	Ir para o diretório do cmake	8
3.2	Executar o comando de criação de tag	8
4	Instalador do CSGrid	10
4.1	Identificar plataformas de instalação	10
4.2	Compilar os módulos dependentes de plataforma	11
4.3	Criar o diretório do Tomcat	11
4.4	Gerar uma chave digital para assinar o CSGrid	12
4.5	Fazer o deploy do cliente	14
4.6	Gerar o executável de instalação	14



5	Empacotamento para Distribuição	15
5.1	Criar o arquivo de validação	15
5.2	Criar o arquivo de configuração de SGAs	15
5.3	Criar o arquivo Readme	16
5.4	Criar o arquivo de distribuição	17

1 Diretório de trabalho

Essa etapa é básica para qualquer uma das tarefas envolvidas na preparação de uma instalação. Supõe-se que o desenvolvedor crie um diretório na sua área de trabalho para efetuar o check-out dos fontes do SVN, alterar o necessário e fazer commit das suas mudanças. Esse diretório de trabalho sempre reflete um determinado **ramo** do subversion.

Especificamente na preparação de uma instalação a ser feita, temos várias áreas de trabalho envolvidas, cada uma referente a uma etapa diferente do processo. Normalmente, são essas as áreas de trabalho que mencionamos no decorrer desse documento.

- **trunk_dir**: é o diretório que possui os fontes que serão usados para criar um branch (conforme Seção 2) de versão de instalação. Normalmente essa área possui os fontes provenientes do trunk.
- **branch_dir**: é o diretório que possui os fontes que serão usados para criar um tag (conforme Seção 3) de versão de instalação. Normalmente essa área possui os fontes provenientes do branch correspondente a versão.
- **tag_dir**: é o diretório que possui os fontes que serão usados para geração do instalador (conforme Seção 4.6). Normalmente essa área possui os fontes provenientes do tag correspondente a versão a ser instalada. Não se deve fazer alterações nessa área de trabalho, já que representa uma versão “congelada”. Correções devem ser feitas sempre no **branch_dir** (com posterior merge para o trunk) ou no **trunk_dir**.

Repare que os nomes **trunk_dir**, **branch_dir** e **tag_dir** são apenas exemplos. Cada desenvolvedor deve criar os diretórios com os nomes que melhor refletem o seu propósito.

Para criar um diretório de trabalho a partir do subversion, você deve seguir os passos seguintes.

1.1 Criar o diretório

Para fazer o check-out do CSGrid, o primeiro passo é definir um diretório de trabalho onde os módulos do CSGrid serão colocados e compilados. Usamos o nome **WORKING_DIR** como exemplo. Esse nome, obviamente, deve ser substituído pelo nome do diretório que o desenvolvedor está criando.



```
[sepetiba:~/workspace] mkdir WORKING_DIR
```

onde **WORKING_DIR** é o nome do diretório.

1.2 Fazer check-out do Subversion

No diretório de trabalho que você criou na etapa anterior, faça check-out do ramo do SVN dos fontes e recursos do CSGrid. Repare que, apesar do CSGrid ser parte do CSBase (é um subdiretório do CSBase no SVN), o check-out é feito apenas do CSGrid. Por exemplo, para o caso de um check-out a partir do trunk, você deve fazer:

```
[sepetiba:~/workspace] cd WORKING_DIR
[sepetiba:~/workspace/WORKING_DIR] svn co https://subversion.
tecgraf.puc-rio.br/engdist/csbase/trunk/csgrid csgrid
```

onde <https://subversion.tecgraf.puc-rio.br/engdist/csbase/trunk/csgrid> é a URL no SVN.

1.3 Fazer o build

Como o check-out do passo anterior foi feito apenas sobre o CSGrid, é necessário recuperar os outros módulos do CSBase para que o processo de build possa ser feito. O CSGrid usa o **csmake** para recuperar os módulos que o compõe. O arquivo `csmake.cfg` define quais módulos fazem parte do CSGrid, ou seja, aquilo que é recuperado do CSBase e o que é parte do próprio CSGrid.

Atualmente o CSGrid é composto dos seguintes módulos, de acordo com a árvore definida no `csmake.cfg`:

```
installTree {
  { rep = repositories.csbase, module = "csbase" },
  { rep = repositories.csbase, module = "csgrid", main = true },
  { rep = repositories.csbase, module = "install" },
  { rep = repositories.csbase, module = "init" },
  { rep = repositories.csbase, module = "sgad" },
```



```
{ rep = repositories.csbase, module = "sgab" },  
  { rep = repositories.csbase, module = "validations" },  
  { rep = repositories.csbase, module = "validations_sys" },  
}
```

A execução do csmake é sempre feita a partir do diretório csgrid:

```
[sepetiba:~/workspace/WORKING_DIR] cd csgrid  
[sepetiba:~/workspace/WORKING_DIR/csgrid] csmake rebuild
```

Você pode confirmar todas as opções requisitadas, teclando ENTER. Esse passo pode demorar um pouco, em função do tamanho do CSBase.

2 Criação de um branch

Essa é a primeira etapa na preparação de uma versão do CSGrid para instalação. Ao criar um branch, você assegura que uma “cópia” do código fonte envolvido na versão que será gerada, isolando esse novo ramo no ciclo de desenvolvimento. Normalmente um branch é criado a partir do trunk.

Para criar um branch do CSGrid, você já deve estar com uma área de trabalho preparada conforme o descrito na Seção 1. Normalmente, essa área de trabalho contém o código proveniente do trunk.

Para criar um branch, você deve seguir os passos seguintes.

2.1 Ir para o diretório do cmake

```
[sepetiba:~/workspace/WORKING_DIR] cd csgrid
```

2.2 Executar o comando de criação de branch

```
[sepetiba:~/workspace/WORKING_DIR/csgrid] cmake branch
```

O script solicita o nome do branch. Esse nome é criado no SVN dentro do diretório **branches**. Você pode confirmar o nome sugerido, digitando ENTER, ou pode entrar com um novo nome. O trecho a seguir é um exemplo do que aparece na console como resultado da execução do comando.

```
Entre com o identificador do branch [CG_v1_01_2009_10_22]
```

```
[INFO ] aplicando branch CG_v1_01_2009_10_22 em >>> csbase
```

```
Committed revision 97937.
```

Após criado, testes e correções de bugs podem ser feitos nesse branch, dando continuidade assim aos preparativos para fechamento de versão. Lembre-se que, mudanças feitas no branch devem ser posteriormente levadas (**merge**) para o trunk.

3 Criação de um tag

Após criado um branch e estabilizada a versão para instalação, deve ser criada uma cópia “congelada” desse branch. Essa cópia, no Subversion, não deve sofrer modificações. Deve-se evitar ao máximo fazer *commits* em uma ramo de tag.

Essa é a primeira etapa na preparação de uma versão do CSGrid para instalação. Ao criar um branch, você assegura que uma “cópia” do código fonte envolvido na versão que será gerada, isolando esse novo ramo no ciclo de desenvolvimento. Normalmente um branch é criado a partir do trunk.

Para criar um tag do CSGrid, você já deve estar com uma área de trabalho preparada conforme o descrito em 1. Essa área de trabalho normalmente contém o código proveniente de um branch.

Para criar um tag, você deve seguir os passos seguintes.

3.1 Ir para o diretório do csmake

```
[sepetiba:~/workspace/WORKING_DIR] cd csgrid
```

3.2 Executar o comando de criação de tag

```
[sepetiba:~/workspace/WORKING_DIR/csgrid] csmake tag
```

O script solicita o nome do tag. Esse nome é criado no SVN dentro do diretório **tags**. Costumamos adotar o seguinte padrão de nome de tag: **CG_v1_01_00_2009_10_22**, onde:

- **CG**: abreviação de 2 letras para o CSGrid
- **v1_01_00**: código da versão no formato **major**, **minor** e **patch**.
- **2009_10_22**: data de criação do tag, no formato **ano**, **mês** e **dia**.

Você pode confirmar o nome sugerido, digitando ENTER, ou pode entrar com um novo nome. O trecho a seguir é um exemplo do que aparece na console como resultado da execução do comando.



Entre com o identificador da tag [CG_v1_01_00_2009_10_22]

```
[INFO ] criando tag CG_v1_01_00_2009_10_22 em >>> csbase
```

```
Committed revision 97962.
```

```
A resources/version.properties
```

```
Checked out revision 97962.
```

```
Sending version.properties
```

```
Transmitting file data .
```

```
Committed revision 97963.
```

```
A resources/version.properties
```

```
Checked out revision 97963.
```

```
Sending version.properties
```

```
Transmitting file data .
```

```
Committed revision 97964.
```

Após criado, esse código copiado para o ramo da tag, será usado para gerar o instalador do CSGrid.

4 Instalador do CSGrid

Essa etapa se refere a geração do binário de instalação do CSGrid, compatível com a versão que está sendo criada. Normalmente, essa etapa é feita usando um diretório de trabalho construído a partir de um tag previamente criado. Consulte a Seção 3 sobre como criar um tag de uma versão e a Seção 1 sobre como preparar uma área de trabalho a partir de um tag do CSGrid.

Para gerar o instalador do CSGrid você deve seguir os passos seguintes.

4.1 Identificar plataformas de instalação

Antes de iniciar a geração do instalador, é fundamental que você identifique em quais plataformas serão instalados o servidor CSGrid e os servidores de execução de algoritmos (SGA). A geração do binário de instalação do CSGrid depende da versão do sistema, da versão do compilador gcc instalado nas máquinas do ambiente e de sua arquitetura.

Uma forma mais rápida de obter essa informação é fornecendo o script da Figura 4.1¹ e o valor da variável de ambiente TEC_UNAME seja informado à equipe de desenvolvimento.

```
setenv TEC_SYSNAME `uname -s`
setenv TEC_SYSVERSION `uname -r|cut -f1 -d.`
setenv TEC_SYSMINOR `uname -r|cut -f2 -d.`
setenv TEC_UNAME "${TEC_SYSNAME}${TEC_SYSVERSION}${TEC_SYSMINOR}"

set GCCVER=`/usr/bin/gcc -dumpversion|cut -f1 -d.`
if ($GCCVER == '4') then
    setenv TEC_UNAME $TEC_UNAME'g4'
endif
unset GCCVER

setenv TEC_SYSARCH `uname -m`
if ($TEC_SYSARCH == 'x86_64') then
    setenv TEC_UNAME $TEC_UNAME'_64'
endif
echo $TEC_UNAME
```

Figura 1: Definição da variável de ambiente TEC_UNAME.

¹Código desenvolvido para o shell csh.

Na Seção 2 do Manual de Administração do CSGrid você encontra mais informações sobre os requisitos de hw/sw e configuração necessária no ambiente de execução do CSGrid.

4.2 Compilar os módulos dependentes de plataforma

Compilar módulos SGAs específicos de cada plataforma. Esses módulos devem ser compilados na plataforma correta. No Tecgraf, existem máquinas com os nomes das principais plataformas: linux26g4, linux26g4_64 e outras. Você pode usar essas máquinas para compilar os módulos dos SGAs. Por exemplo, para compilar um SGA para Linux26g4_64, você deve fazer o seguinte:

- Fazer login na máquina com a plataforma (por exemplo, linux26g4).

```
[sepetiba:~/workspace/csgrid_build] ssh linux26g4_64
```

- Executar o comando de compilação do binário do SGA

```
[linux26g4-64:~/workspace/csgrid_build/csgrid] ./csmake build sgad
```

Para gerar o executável do SGA para Windows, ir para uma máquina Windows no Tecgraf (tipicamente a Pero), ir para o diretório `sgad/src` que possui os fontes do SGA e rodar o script `build.bat`.

O diretório `sgad/bin` passa a ter um sub-diretório de cada plataforma para a qual um SGA foi compilado. Por exemplo:

```
[sepetiba:~/workspace/csgrid_build/sgad/bin] ls  
Linux26g4/ Linux26g4_64/ sga-daemon@ sgad-cnfvavc.lua@ sgad-cnf.lua@
```

4.3 Criar o diretório do Tomcat

Colocar o tomcat no diretório de trabalho. Normalmente basta expandir o arquivo mais atual que fica em `/softwares/linux/bin/tomcat`. Em seguida, fazer um link com o nome **tomcat** para o diretório criado.



```
[sepetiba:~/workspace/WORKING_DIR/csgrid] cd ..  
[sepetiba:~/workspace/WORKING_DIR] tar -xvzf /softwares/linux/  
bin/tomcat/apache-tomcat-6.0.20.tar.gz  
[sepetiba:~/workspace/WORKING_DIR] ln -s apache-tomcat-6.0.20/ tomcat
```

4.4 Gerar uma chave digital para assinar o CSGrid

Antes de executar o **deploy** do CSgrid, é necessário criar uma chave digital para fazer a assinatura dos arquivos jar. Isso porque o CSGrid usa alguns recursos da máquina cliente (p.ex. impressão, acesso ao filesystem local) que exigem que os JARs sejam assinados digitalmente para serem “confiáveis”.

A rigor esta assinatura deveria ser obtida por um fornecedor oficial como VeriSign² ou Thawte³. Por enquanto, alguns sistemas serão assinados com um certificado "auto-assinado", i.e. gerado pelo próprio Tecgraf. Este certificado tecnicamente não oferece garantia de segurança ao usuário, e tem validade de apenas 6 meses, mas é a única alternativa no momento.

O arquivo gerado com a chave, no CSGrid, possui o nome **CSGKeystore** e deve estar no diretório **install**. Para se gerar o arquivo com a chave digital para o CSGrid deve-se executar o seguinte comando:

```
[sepetiba:~/workspace/WORKING_DIR] cd install  
[sepetiba:~/workspace/WORKING_DIR/install] keytool -genkey  
-keystore CSGKeystore -alias CSGrid
```

O resultado a seguir é exibido na console. O texto marcado com vermelho deve ser digitado por você.

```
Enter keystore password: tecgraf  
Re-enter new password: tecgraf  
What is your first and last name?  
[Unknown]: CSGrid  
What is the name of your organizational unit?  
[Unknown]: Tecgraf
```

²<http://www.verisign.com>

³<http://www.thawte.com>



What is the name of your organization?

[Unknown]: PUC-Rio

What is the name of your City or Locality?

[Unknown]: Rio de Janeiro

What is the name of your State or Province?

[Unknown]: Rio de Janeiro

What is the two-letter country code for this unit?

[Unknown]: BR

Is CN=CSGrid, OU=Tecgraf, O=PUC-Rio, L=Rio de Janeiro, ST=Rio de Janeiro, C=BR correct?

[no]: yes

Enter key password for <csgrid>

(RETURN if same as keystore password): <ENTER>

Re-enter new password:

Após a geração da chave, o certificado auto-assinado é gerado via comando:

```
[sepetiba:~/workspace/WORKING_DIR/install] keytool -selfcert  
-alias CSGrid -keystore CSGKeystore
```

O resultado a seguir é exibido na console. O texto marcado com vermelho deve ser digitado por você.

Enter keystore password: tecgraf

Repare que a senha “tecgraf” deve ser a mesma usada no script **install** que executa o comando de assinatura do jar. No caso de querer usar outra senha, não se esqueça de alterar o argumento **storepass** do seguinte comando desse script:

```
jarsigner -storepass tecgraf -keystore CSGKeystore FILE csgrid
```

Também é importante lembrar que certificados auto-assinados têm validade de apenas 6 meses.



4.5 Fazer o deploy do cliente

Nessa etapa, os arquivos jar e outros *resources* (por exemplo, imagens, etc) necessários para execução do cliente, são copiados para a pasta de aplicações do Tomcat. Nesse momento, a chave digital criada na Seção 4.4 é usada para fazer a assinatura dos jars e, portanto, já deve ter sido criado.

O comando que faz o deploy é executado pelo **csmake** e, assim como os outros, deve ser executado no diretório **csgrid**.

```
[sepetiba:~/workspace/WORKING_DIR/csgrid] csmake webdeploy
```

4.6 Gerar o executável de instalação

Essa etapa se refere a geração do instalador, ou seja, do binário executável de instalação. Após executar o comando a seguir, um arquivo com extensão `.bin` será criado na raiz do diretório de trabalho.

```
[sepetiba:~/workspace/WORKING_DIR/csgrid] csmake pack
```

5 Empacotamento para Distribuição

Após ter feito a validação da instalação, deve ser criado um arquivo **tar** com o binário do instalador e outros arquivos necessários. No CSGrid, costumamos chamar o arquivo de distribuição de **csgridInstal.tgz** e ele é composto de:

- o binário do instalador gerado na Seção 4.6,
- o arquivo `Readme`, com informações sobre os procedimentos básicos de instalação,
- o arquivo `sgad-cnflua`, com um exemplo de configuração de SGAs para as plataformas disponibilizadas no binário do instalador
- o arquivo `Validations.properties` com o arquivo de propriedades usado pelo binário de instalação.

5.1 Criar o arquivo de validação

Esse arquivo de propriedades é passado como parâmetro para o binário de instalação e usado para verificar condições no ambiente onde o CSGrid está sendo instalado como, por exemplo, no caso de já haver uma outra instalação anterior cujos dados precisem ser migrados para a nova instalação.

O arquivo de validação é uma cópia do arquivo `validations_sys/System.properties.default`. Para criá-lo basta executar o seguinte comando de cópia:

```
[sepetiba:~/workspace/WORKING_DIR] cp validations_sys/System.  
properties.default Validations.properties
```

5.2 Criar o arquivo de configuração de SGAs

Para facilitar, costumamos enviar no arquivo de distribuição, a configuração dos SGAs que serão instalados. Você pode usar como base o arquivo `sgad-cnflua` que está disponível em `sgad/src` e alterá-lo de acordo com as plataformas nas quais o SGA será instalado.



```
[sepetiba:~/workspace/WORKING_DIR] cp sgad/src/sgad-cnf.lua .
```

Um exemplo de um nó para a execução do SGA em Linux26g4 é a seguinte:

```
Node{
  name = "Linux26g4",
  num_processors = 2,
  platform_id = "Linux26g4",
  byte_order = LITTLE_ENDIAN,
}
```

5.3 Criar o arquivo Readme

O arquivo Readme deve conter orientações básicas que não façam parte do Manual de Administração e sejam específicas do ambiente onde o CSGrid será instalado.

Segue um exemplo de um exemplo de Readme que usamos como base para a distribuição de uma versão.

Versão do CSGrid: CG-v1_01_00-2009_10_22

Conteúdo do arquivo:

- . Readme
- . CG-v1_01_00-2009_10_22.bin: instalador
- . Validations.properties: parâmetro a ser passado para o instalador
- . sgad-cnf.lua: exemplo de configuração de SGAs para o ambiente a ser instalado

Esse instalador foi preparado para fazer a instalação do CSGrid e o SGA em uma máquina Linux26g4.

O manual de instalação do CSGrid é o ManualAdminCSGrid.pdf. Nesse manual vocês encontram as orientações com relação aos requisitos do ambiente, instalação e execução do servidor CSGrid.



5.4 Criar o arquivo de distribuição

Crie um arquivo compactado contendo todos os arquivos que fazem parte da distribuição do CSGrid.

```
[sepetiba:~/workspace/WORKING_DIR] tar -czf csgridInstall.tgz  
CG-v1_01_00-2009_10_22.bin ReadMe sgad-cnf.lua Validations.properties
```