

# Guia de Início Rápido do OPENBUS 2.0.0

Tecgraf

6 de março de 2014

## Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
<b>2</b>	<b>Como instalar e levantar o barramento:</b>	<b>1</b>
<b>3</b>	<b>Como compilar e executar um demo:</b>	<b>2</b>

## 1 Introdução

Este documento visa prover os passos necessários para instalar e levantar um barramento OPENBUS [3] da versão 2.0.0 e executar um demo. Caso tenha interesse em entender melhor o que é um barramento OPENBUS e os detalhes de instalação, consulte o manual de referência [5] e o manual de instalação [4]. Os procedimentos a seguir descrevem a instalação em plataforma Linux (Ubuntu 12.04) utilizando comandos do Bash shell (interpretador de linguagem de comandos) [2] [1]. O demo é em C++, e é executado na mesma máquina do barramento. Para a execução em outras plataformas, de demos em outras linguagens, e utilizando configurações diferentes, é sugerido que se consulte os manuais acima referidos.

## 2 Como instalar e levantar o barramento:

1. Faça o download da versão estável do core do OpenBus para um diretório do Linux. Selecione o pacote mais apropriado para a sua plataforma:

<https://jira.tecgraf.puc-rio.br/confluence/display/OPENBUS020/Core>

2. Descompacte o pacote:

```
$ tar xvzf openbus-core-2.0.0.1-Linux26g4.tar.gz
```

3. Para facilitar o processo, é recomendado que se inclua as seguintes variáveis de ambiente no arquivo de inicialização. Não esqueça de sair e entrar novamente no shell para que as variáveis de ambiente sejam definidas para a sessão corrente.

```
$ cd ~
$ vi .bashrc
export OPENBUS_HOME=<caminho do openbus-core>
export PATH="${OPENBUS_HOME}/bin:.${PATH}"
export LD_LIBRARY_PATH="/usr/lib:/lib/i386-linux-gnu:${OPENBUS_HOME}/lib: \
    ${LD_LIBRARY_PATH}"
export OPENSsl_HOME="${OPENBUS_HOME}/openssl"
export LUA_PATH="${OPENBUS_HOME}/?.lua;${LUA_PATH}"
```

4. Gere um arquivo de chave privada (arquivo com extensão ".key") e um certificado (arquivo com extensão ".cert") utilizando o comando `openssl` que acompanha o pacote. Os arquivos podem ter um nome diferente do sugerido abaixo, desde que o nome do arquivo de chave privada seja passado no parâmetro `-privatekey` do servidor do barramento (`busservices`).

```
$ cd $OPENBUS_HOME
$ ${OPENBUS_HOME}/bin/openssl genrsa -out tmp_openssl.key 2048
$ ${OPENBUS_HOME}/bin/openssl pkcs8 -topk8 -nocrypt \
  -in tmp_openssl.key -out openbus.key -outform DER
$ rm -f tmp_openssl.key
$ ${OPENBUS_HOME}/bin/openssl req -config $OPENSSL_HOME/openssl.cnf -new -x509 \
  -key openbus.key -keyform DER \
  -out openbus.crt -outform DER
```

5. Crie um script de validação em LUA para verificar se o nome do usuário e senha utilizados para acessar o barramento são válidos. No nosso caso, vamos utilizar um script de validação que verifica apenas se o nome do usuário é igual à senha:

```
$ cd $OPENBUS_HOME
$ vi validador.lua
local function validator(name, password)
  if name == password then
    return true
  end
end
return function(configs) return validator end
```

6. Levante o barramento. O nome do script de validação é passado no parâmetro `-validator` sem a extensão ".lua". A variável de ambiente `$LUA_PATH` contém o caminho onde o servidor do barramento fará a busca pelo script de validação. Caso o parâmetro `-privatekey` não seja passado, o servidor do barramento utilizará o nome do arquivo de chave privada padrão (`openbus.key`). Como não foram passados os parâmetros `-host` e `-port`, o servidor do barramento será levantando utilizando a máscara de host e a porta padrão (`host="*" port=2089`).

```
$ cd $OPENBUS_HOME
$ ./bin/busservices -privatekey openbus.key -admin admin -validator validador
```

### 3 Como compilar e executar um demo:

1. Faça o download de uma versão estável de um SDK (ex: C++ - ORB Mico). Selecione o SDK mais apropriado para a sua plataforma:

<https://jira.tecgraf.puc-rio.br/confluence/display/OPENBUS020/SDK>

2. Descompacte o pacote:

```
$ tar xvzf openbus-cpp-mico-2.0.0.0-Linux26g4.tar.gz
```

3. Instale também as seguintes bibliotecas no Linux (Ubuntu 12.04):

```
$ sudo apt-get install g++
$ sudo apt-get install libstdc++6
```

4. Para facilitar o processo, é recomendado que se inclua as seguintes variáveis de ambiente no arquivo de inicialização. Não esqueça de sair e entrar novamente no shell para que as variáveis de ambiente sejam definidas para a sessão corrente.

```
$ cd ~
$ vi .bashrc
export PATH="/<caminho do openbus-cpp-mico>/bin:.${PATH}"
export LD_LIBRARY_PATH="/<caminho do openbus-cpp-mico>/lib:${LD_LIBRARY_PATH}"
export BOOST_ROOT="/<caminho do openbus-cpp-mico>/boost-build"

alias mico-mt='export LD_LIBRARY_PATH="/<caminho do openbus-cpp-mico>/lib/mico-mt: \
${LD_LIBRARY_PATH}";export PATH="/<caminho do openbus-cpp-mico>/bin/mico-mt:${PATH}''

alias mico-mt-d='export LD_LIBRARY_PATH="/<caminho do openbus-cpp-mico>/lib/mico-mt-d: \
${LD_LIBRARY_PATH}";export PATH="/<caminho do openbus-cpp-mico>/bin/mico-mt-d:${PATH}''

alias mico-st='export LD_LIBRARY_PATH="/<caminho do openbus-cpp-mico>/lib/mico-st: \
${LD_LIBRARY_PATH}";export PATH="/<caminho do openbus-cpp-mico>/bin/mico-st:${PATH}''

alias mico-st-d='export LD_LIBRARY_PATH="/<caminho do openbus-cpp-mico>/lib/mico-st-d: \
${LD_LIBRARY_PATH}";export PATH="/<caminho do openbus-cpp-mico>/bin/mico-st-d:${PATH}''
```

5. Chame um dos aliases acima de acordo com o tipo de compilação desejado (mt-multithreading/st-single threading, d-debug):

```
$ mico-mt-d
```

6. Entre no diretório de demo que deseja compilar:

```
$ cd /<caminho do openbus-cpp-mico>/demo/hello
```

7. Compile o demo selecionado utilizando os parâmetros correspondente ao tipo de compilação:

```
$ b2 --layout=tagged threading=multi
```

8. Execute o script LUA com um lote de comandos para cadastrar o servidor do demo no barramento. Caso o parâmetro `--password` não seja passado, a senha de acesso será pedido pelo administrador do barramento (busadmin).

```
$ cd /<caminho do openbus-cpp-mico>/demo/hello/admin
$ busadmin --login=admin --password=admin --script=busadmin.adm
[INFO] Categoria 'DemoCppHelloCategory' cadastrada com sucesso
[INFO] Entidade 'demo' cadastrada com sucesso
[INFO] Certificado da entidade 'demo' cadastrada com sucesso
[INFO] Interface 'IDL:tecgraf/openbus/interop/simple/Hello:1.0' cadastrada com sucesso.
[INFO] Autoriza??o concedida a 'demo': IDL:tecgraf/openbus/interop/simple/Hello:1.0
```

9. Para listar as informações cadastradas do servidor do demo, utilize os seguintes comandos:

```
$ busadmin --login=admin --password=admin --list-entity
-----+-----+-----+-----
| ID ENTIDADE | ID CATEGORIA | NOME
```

```

-----+-----+-----+-----+
001 | demo          | DemoCppHelloCategory | Entidade gerada pelo script
-----+-----+-----+-----+

$ busadmin --login=admin --password=admin --list-category
-----+-----+-----+-----+
    | ID CATEGORIA      | NOME
-----+-----+-----+-----+
001 | DemoCppHelloCategory | Categoria gerada pelo script
-----+-----+-----+-----+

$ busadmin --login=admin --password=admin --list-interface
-----+-----+-----+-----+
    | INTERFACE
-----+-----+-----+-----+
001 | IDL:tecgraf/openbus/interop/simple/Hello:1.0
-----+-----+-----+-----+

$ busadmin --login=admin --password=admin --list-authorization
-----+-----+-----+-----+
    | ID ENTIDADE | INTERFACES
-----+-----+-----+-----+
001 | demo        | IDL:tecgraf/openbus/interop/simple/Hello:1.0
-----+-----+-----+-----+

```

10. Execute o servidor do demo. O caminho do binário segue o tipo de compilação escolhido:

```

$ cd /<caminho do openbus-cpp-mico>/demo
$ ./bin/hello/gcc-4.6/debug/threading-multi/hello_server --private-key \
  hello/admin/DemoCppHello.key

```

11. Em um shell diferente, chame o alias correspondente ao tipo de compilação desejado (mt-multithreading/st-single threading, d-debug):

```
$ mico-mt-d
```

12. Execute o cliente do demo. O caminho do binário segue o tipo de compilação escolhido:

```

$ cd /<caminho do openbus-cpp-mico>/demo
$ ./bin/hello/gcc-4.6/debug/threading-multi/hello_client

```

13. Liste as ofertas de serviços disponíveis no barramento:

```

$ busadmin --login=admin --password=admin --list-offer
-----+-----+-----+-----+-----+
    | ID ENTIDADE | INTERFACES                                     | DATA      | HORA
-----+-----+-----+-----+-----+
001 | demo        | IDL:scs/core/IComponent:1.0                   | 06/06/2013 | 17:48:07
    |              | IDL:scs/core/IMetaInterface:1.0               |             |
    |              | IDL:scs/core/IReceptacles:1.0                 |             |
    |              | IDL:tecgraf/openbus/interop/simple/Hello:1.0 |             |
-----+-----+-----+-----+-----+

```

14. Se o servidor do demo foi parado e reiniciado, os serviços serão oferecidos novamente. Remova todos os serviços oferecidos e reinicie o servidor do demo.

```
$ busadmin --login=admin --password=admin --list-offer
```

	ID ENTIDADE	INTERFACES	DATA	HORA
001	demo	IDL:scs/core/IComponent:1.0	06/06/2013	17:48:07
		IDL:scs/core/IMetaInterface:1.0		
		IDL:scs/core/IReceptacles:1.0		
		IDL:tecgraf/openbus/interop/simple/Hello:1.0		
002	demo	IDL:scs/core/IComponent:1.0	06/06/2013	17:53:52
		IDL:scs/core/IMetaInterface:1.0		
		IDL:scs/core/IReceptacles:1.0		
		IDL:tecgraf/openbus/interop/simple/Hello:1.0		

```
$ ./bin/hello/gcc-4.6/debug/threading-multi/hello_client
```

Existe mais de um servico Hello no barramento. Tentaremos encontrar uma funcional.  
Erro de comunicacao. Verifique se o sistema se encontra ainda disponivel ou se sua  
conexao com o mesmo foi interrompida

```
$ busadmin --login=admin --password=admin --del-offer --entity=demo
```

	ID ENTIDADE	INTERFACES	DATA	HORA
001	demo	IDL:scs/core/IComponent:1.0	06/06/2013	17:48:07
		IDL:scs/core/IMetaInterface:1.0		
		IDL:scs/core/IReceptacles:1.0		
		IDL:tecgraf/openbus/interop/simple/Hello:1.0		
002	demo	IDL:scs/core/IComponent:1.0	06/06/2013	17:53:52
		IDL:scs/core/IMetaInterface:1.0		
		IDL:scs/core/IReceptacles:1.0		
		IDL:tecgraf/openbus/interop/simple/Hello:1.0		

Informe o Indice da oferta que deseja remover:

002

[INFO] Oferta '2' removida com sucesso.

```
$ busadmin --login=admin --password=admin --del-offer --entity=demo
```

	ID ENTIDADE	INTERFACES	DATA	HORA
001	demo	IDL:scs/core/IComponent:1.0	06/06/2013	17:48:07
		IDL:scs/core/IMetaInterface:1.0		
		IDL:scs/core/IReceptacles:1.0		
		IDL:tecgraf/openbus/interop/simple/Hello:1.0		

Informe o Indice da oferta que deseja remover:

001

[INFO] Oferta '1' removida com sucesso.

```
$ cd /<caminho do openbus-cpp-mico>/demo
```

```
$ ./bin/hello/gcc-4.6/debug/threading-multi/hello_server --private-key \
hello/admin/DemoCppHello.key
```

## Referências

- [1] Mendel Cooper. Advanced bash-scripting guidel. <http://tldp.org/LDP/abs/html/>, 2012.
- [2] GNU. Bash reference manual. <http://www.gnu.org/software/bash/manual/bashref.html>, 2010.
- [3] TecGraf. OpenBus - Enterprise Integration Application Middleware. <http://www.tecgraf.puc-rio.br/openbus>, 2006.
- [4] TecGraf. *Manual de instalação do OpenBus 2.0.0*. TecGraf, 2012.
- [5] TecGraf. *Manual do OpenBus 2.0.0*. TecGraf, 2012.