


FAQs - Suporte técnico

Central de Atendimento TOTVS (<https://centraldeatendimento.totvs.com/hc/pt-br>) > TOTVS Agro (<https://centraldeatendimento.totvs.com/hc/pt-br/categories/1500000346841-TOTVS-Agro>)

Pesquisa

Pesquisa 

Manual de Instalação - CCT Server

Atualizado em:

26 de dezembro de 2019 09:22

Seguir

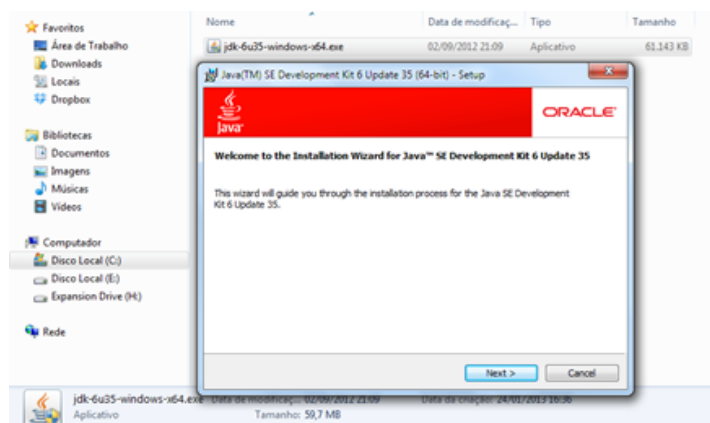
1.1 Objetivo

Este manual tem como objetivo fornecer os passos necessários para instalação e configuração do aplicativo CCTServer. As instruções aqui apresentadas contemplarão desde a instalação dos softwares necessários, como o SDK Java e Servidor de Aplicações JBoss 6.1.0, bem como a implantação da aplicação nestes ambientes, em plataforma Windows.

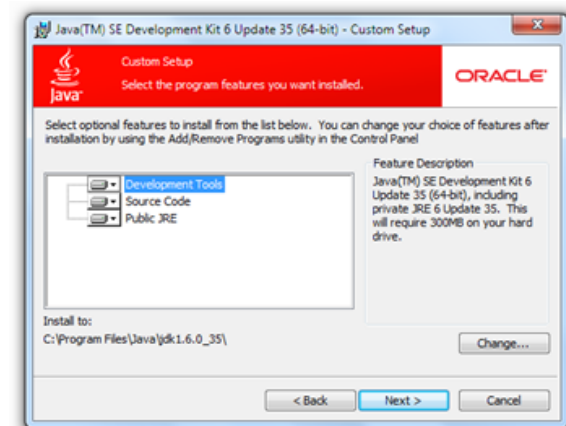
1.2 JDK

a. Instalação

Com a versão mais recente (atualizada) do instalador à disposição, proceda com a execução do mesmo:

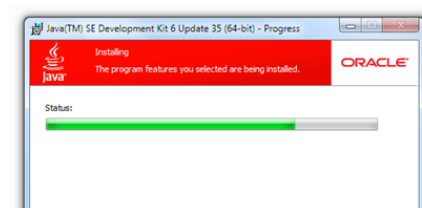


No próximo passo configuraremos os diretórios de instalação:



Clicando em change, alteraremos o caminho de instalação para: C:\totvs\jdk

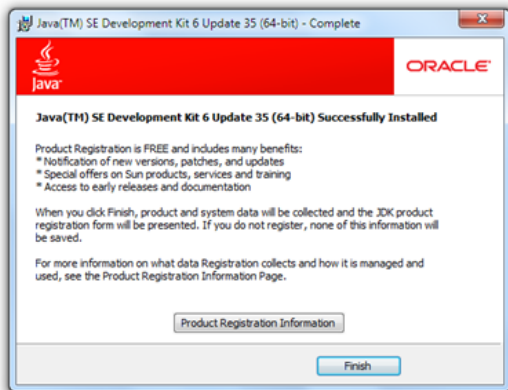
Após confirmar e prosseguir, será iniciado o processo de instalação:



Na sequência será solicitado o local de instalação do JRE (Java Runtime Environment). A instalação deste Runtime não é necessário para o funcionamento do CCTServer, mas recomenda-se sua instalação (caso o sistema já não tenha uma versão instalada) pois os consoles de administração do servidor de aplicações se baseiam em Applets, que exigem um JRE.

Como caminho para o JRE informe C:/totvs/jre

Ao clicar em Finish encerramos a instalação. Não é necessário registrar o produto (opcional).



b. Variáveis de Ambiente

Definir as seguintes variáveis de ambiente abaixo como sendo do sistema:

JAVA_HOME=C:\totvs\jdk

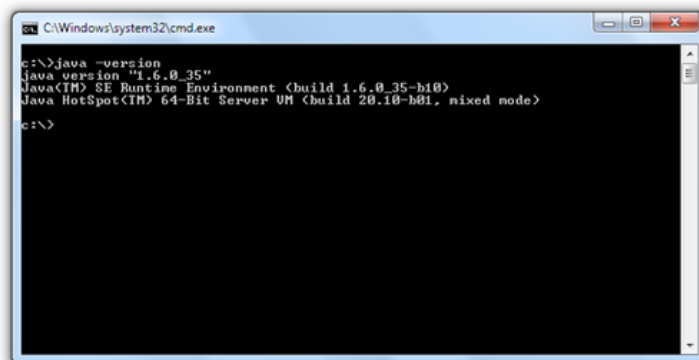
CLASSPATH=.

PATH=%JAVA_HOME%\bin;(restante do path)

c. Teste da instalação

Ao acessar o terminal o comando abaixo pode ser executado para verificar se a instalação e configuração foram realizadas com sucesso. Após a execução do comando deve ser possível visualizar as informações da versão instalada.

C:\> java -version



d. Problemas na instalação

Caso ocorra algum problema na instalação, consultar o manual Online citado nas referências para resolução, e em segundo nível o suporte da TOTVS.

1.3 Instalação do Servidor de Aplicações JBoss 6.1

Para a implantação do CCTServer a instalação de um Servidor de Aplicações é exigida. O único servidor de aplicações homologado para a execução do

exigida. O único servidor de aplicações homologado para a execução do CCTServer, atualmente é o JBoss Application Server, em sua versão 6.1.0.

O software JBoss AS é open-source e multiplataforma, o que significa que seu instalador pode ser utilizado para diversos sistemas operacionais e plataformas (32 bits e 64 bits) que suportem a plataforma Java.

Apesar de estar gratuitamente disponível para download no site do fornecedor, obtenção do instalador do JBoss para o CCTServer deve ser feita através do FTP disponibilizado pela TOTVS, pois o mesmo já estará pré-configurado com as bibliotecas necessárias para o funcionamento do CCTServer.

Portanto, para a obtenção do instalador, é necessário entrar em contato com o suporte da TOTVS para obter os dados de acesso ao FTP.

a. Instalação

I. Requisitos de espaço em disco

Para a instalação do JBoss AS 6.1.0, no mínimo 10GB devem estar disponíveis no disco de instalação.

II. Requisitos de Software

É necessário que se tenha o JDK Oracle devidamente instalado e configurado no ambiente.

III. Início da Instalação

Uma vez obtido o pacote, a instalação consiste basicamente em descompactar o mesmo. Portanto descompacte-o no diretório de instalação. A recomendação é o diretório C:\totvs.

A instalação está concluída! Nas seções seguintes serão apresentadas algumas configurações e uma breve apresentação dos principais recursos do servidor.

IV. Variáveis de ambiente

Uma vez instalado o JBoss é necessário definir a variável de ambiente JBOSS_HOME (apresentada abaixo) como sendo do sistema, onde a variável deve ter como valor o caminho para o diretório de instalação do JBoss AS:

Nome da variável: JBOSS_HOME

Valor da variável: C:\totvs\jboss-6.1.0.Final

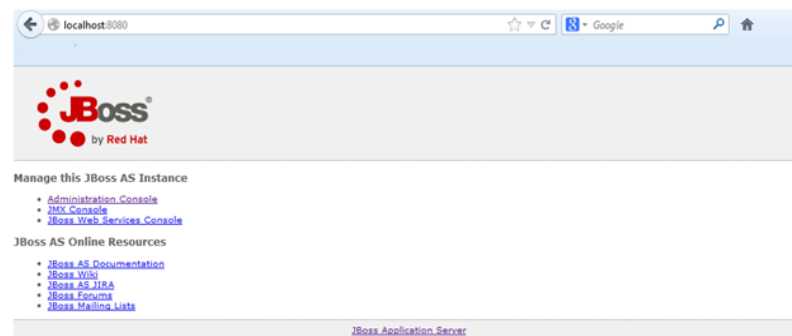
Acessando o diretório bin da instalação do JBoss, será possível encontrar o arquivo **run.bat**. Ele é responsável pela execução do servidor.

V. Teste da instalação

Para testar a instalação execute o arquivo run.bat. Isso irá inicializar o servidor em modo stand-alone.

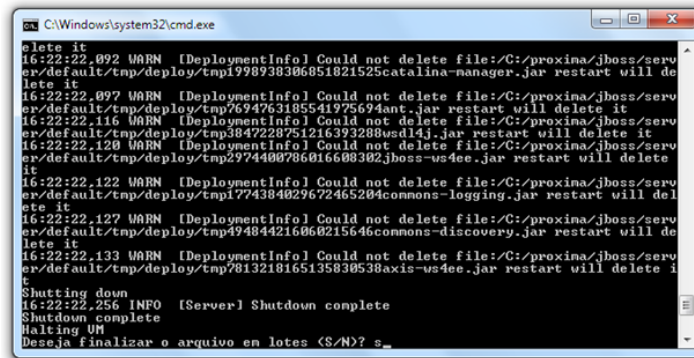
Ao final do log é possível constatar o tempo de inicialização do servidor.

Para comprovar que o servidor está realmente sendo executado, acesso através de um navegador o seguinte endereço: <http://localhost:8080/> (<http://localhost:8080/>).



VI. Encerrando o modo stand-alone

Basta clicar CTRL+C ou fechar a janela do JBOSS em stand-alone.



```
C:\Windows\system32\cmd.exe
delete it
16:22:22.092 WARN [DeploymentInfo] Could not delete file:/C:/proxima/jboss/serve
er/default/tmp/deploy/tmp199893836851821525catalina-manager.jar restart will de
lete it
16:22:22.097 WARN [DeploymentInfo] Could not delete file:/C:/proxima/jboss/serve
er/default/tmp/deploy/tmp7694763185541975694ant.jar restart will delete it
16:22:22.116 WARN [DeploymentInfo] Could not delete file:/C:/proxima/jboss/serve
er/default/tmp/deploy/tmp3847228751216393288wsdl4j.jar restart will delete it
16:22:22.120 WARN [DeploymentInfo] Could not delete file:/C:/proxima/jboss/serve
er/default/tmp/deploy/tmp2974408786016608302jboss-us4ee.jar restart will delete
it
16:22:22.122 WARN [DeploymentInfo] Could not delete file:/C:/proxima/jboss/serve
er/default/tmp/deploy/tmp1774384029672465204commons-logging.jar restart will de
lete it
16:22:22.127 WARN [DeploymentInfo] Could not delete file:/C:/proxima/jboss/serve
er/default/tmp/deploy/tmp494844216868215646commons-discovery.jar restart will de
lete it
16:22:22.133 WARN [DeploymentInfo] Could not delete file:/C:/proxima/jboss/serve
er/default/tmp/deploy/tmp7813218165135838538axis-us4ee.jar restart will delete i
t
Shutting down
16:22:22.256 INFO [Server] Shutdown complete
Shutdown complete
Waiting VM
Desaja finalizar o arquivo em lotes <S/N>? s_
```

VII. Ajustando Parâmetros de Memória

Se for necessário alterar a quantidade de memória disponível para a JVM (Java Virtual Machine) utilizada pelo JBoss, devemos editar o arquivo run.conf.bat, que está no diretório bin, da instalação do JBoss.

Neste arquivo, procure por uma linha como a seguinte:

```
set "JAVA_OPTS=-Xms768M -Xmx768M -XX:MaxPermSize=256M"
```

Os parâmetros -Xms e -Xmx determinam, respectivamente, a quantidade inicial e a quantidade máxima de memória que pode ser utilizada pelo servidor, e devem ser configuradas para o mesmo valor, por questões de desempenho. Neste caso (do exemplo) a configuração determina 768MB de memória, que pode ser aumentado de acordo com a necessidade e disponibilidade de recursos do servidor.

Para maiores informações sobre os possíveis parâmetros e seus significados, acessar a documentação <http://www.oracle.com/technetwork/java/javase/gc-tuning-6-140523.html> (<http://www.oracle.com/technetwork/java/javase/gc-tuning-6-140523.html>)

b. Configurando a Instância Padrão do PIMSCS no JBoss

Um dos parâmetros esperados pela aplicação Server é a instância padrão que deve ser utilizada para consultas na base de dados do PIMSCS.

Esta configuração deve ser feita no arquivo run.conf.bat (para ambientes Windows) ou run.conf (para ambientes Linux).

Nestes arquivos, procure a parte que é responsável pela passagem de parâmetros de memória para a JVM, como a seguir:

```
set "JAVA_OPTS=-server -Xms1024m -Xmx1024m -Xss1024k -Xnoclassgc -
XX:MaxPermSize=256m -XX:+UseParNewGC -Duser.language=pt -
Duser.country=BR -Duser.timezone=GMT-3 -Dinstancia_default=ALBER -
Djms.servers.ips=localhost -Djms.servers.port=1099 -Dversao.jboss=JB6"
```

O parâmetro -Dinstancia_default=ALBER neste exemplo está com valor ALBER, e deve ser substituído pelo nome da instância em questão. Um ponto de atenção é que o nome da instância deve ser escrito em caixa alta (letras maiúsculas) pois é assim que são configurados na base de dados do PIMSCS. O parâmetro -Djms.servers.ips deve ser configurado com o IP do host.

NOTA: O nome da instância não tem nenhuma relação com os nomes dos diretórios de configuração do JBoss.

É bom salientar que, caso mais de uma instância do JBoss seja inicializada como serviço na mesma instalação, será necessário ter um arquivo run.bat e run.conf.bat (run.sh e run.conf, em caso de plataformas Linux) para cada instância de serviço do JBoss. Mais detalhes na seção 3.3.10.

c. Configurando o JBOSS para executar como serviço do Windows

Dentro do diretório bin da instalação do JBoss é possível encontrar os arquivos jbosssvc.exe e service.bat. Estes dois arquivos são utilizados para a execução e configuração do serviço, respectivamente.

O arquivo service.bat contém as configurações para a execução do serviço. Dentre as configurações encontradas neste arquivo existe uma que define o nome pelo qual o serviço do JBoss será identificado. Esta configuração pode ser vista no trecho abaixo:

```
if "x%SVCNAME%" == "x" (
    set "SVCNAME=JBAS61SVC"
)
```

set SVCDESC=JBoss Application Server 6.1

set SVCDESC=JBoss Application Server 6.1.0 Final/Platform: Windows
%PROCESSOR_ARCHITECTURE%

set NOPAUSE=Y

Se for necessário, as descrições (em vermelho) podem ser modificadas para outro nome.

NOTA: Quando houver a necessidade de instalação de múltiplos serviços do JBoss a variável SVCNAME deverá ser obrigatoriamente diferente para cada instalação.

O arquivo service.bat utiliza os arquivos de inicialização e finalização do JBoss (run.bat e shutdown.bat). Se for necessário informar parâmetros de inicialização ou finalização ao JBoss (como quando há a necessidade de inicializar uma instância com uma configuração em específico) isto pode ser feito nas seções apresentadas abaixo. Os parâmetros apresentados a seguir são apenas exemplos de como se passariam parâmetros à execução se houvesse a necessidade, e não precisam ser informadas em uma configuração padrão.

Para passar parâmetros adicionais na inicialização:

```
jbosssvc.exe -p 1 "Starting %SVCDESC%" > run.log
```

```
call run.bat -c producao -b 192.168.0.111 < .lock >> run.log 2>&1
```

```
jbosssvc.exe -p 1 "Shutdown %SVCDESC% service" >> run.log
```

Para passar parâmetros adicionais na finalização:

```
jbosssvc.exe -p 1 "Shutting down %SVCDESC%" > shutdown.log
```

```
call shutdown -S --server=192.168.0.111:1099 < .lock >> shutdown.log 2>&1
```

```
jbosssvc.exe -p 1 "Shutdown %SVCDESC% service" >> shutdown.log
```

Nos exemplos acima, foram passados parâmetros para inicializar uma instância com a configuração chamada "produção":

-c producao

Outro parâmetro indica que a instância deve ser executada no IP 192.168.0.111:

-b 192.168.0.111

Já na finalização, foi indicado qual o IP do servidor que deve ser parado:

--server=192.168.0.111:1099

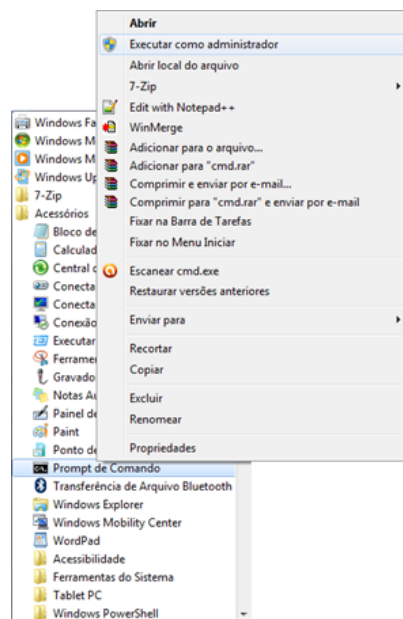
Para realizar a instalação do serviço, basta executar no prompt de comando (como administrador), e a partir do diretório JBOSS_HOME/bin, o seguinte comando:

JBOSS_HOME/bin> service.bat install

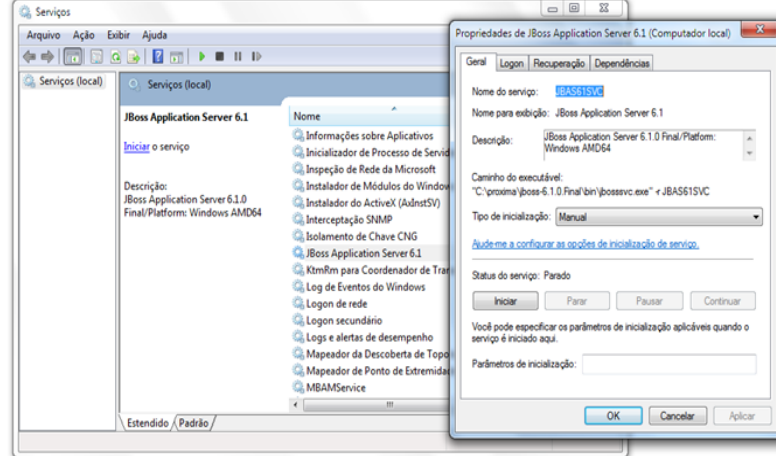
Este comando precisa ser realizado apenas uma vez. Para realizar a desinstalação o procedimento é parecido:

JBOSS_HOME/bin> service.bat uninstall

ATENÇÃO: caso a instalação do serviço falhe, tenha certeza de ter aberto o prompt de comando com permissão de Administrador, para a execução dos comandos:



Após a configuração é possível ver o serviço disponível como qualquer outro serviço do Windows. Configure-o para inicialização automática e reinicie o servidor, para que o serviço possa ser iniciado (isto pode ser necessário caso o usuário em execução não tenha permissão para inicialização de serviços).



d. Configurando Várias Instâncias do JBOSS para executar como serviço do Windows

Quando houver a necessidade de instalar várias instâncias do JBoss como serviço do Windows será necessário, primeiramente, realizar uma cópia do diretório default e, mantendo a cópia no mesmo diretório do diretório default, renomear esta cópia para o nome da instância a ser utilizada. No exemplo da imagem abaixo a nova instância criada tem o nome nova instância:

all	14/06/2013 14:24
default	14/06/2013 15:28
novainstancia	14/06/2013 15:40

Após este procedimento, será necessário criar uma cópia dos arquivos run.bat, run.conf.bat e service.bat, que estão no diretório Bin do JBoss, e renomeá-los para identificar a instância que será utilizada. Por exemplo: run_novainstancia.bat, run_novainstancia.conf.bat e service_novainstancia.bat.

Isto é necessário para que as configurações de memória, IP e porta possam ser configurados individualmente para cada instância.

Estes arquivos citados referenciam-se uns aos outros e, portanto, é necessário alterar os nomes nestes locais de acordo com o que for definido. Por exemplo, no arquivo run_novainstancia.bat o local que referencia o arquivo run.conf.bat deve passar a referenciar o arquivo run_novainstancia.conf.bat. No arquivo service_novainstancia.bat os locais que referenciam os arquivos

run.conf.bat, run.bat e service.bat devem referenciar os novos nomes dos arquivos. Também é necessário renomear os arquivos de lock (s.lock e r.lock) e log (run.log e shutdown.log) referenciados no arquivo, adicionando a eles um sufixo que identifique a nota instância. Veja um exemplo nas imagens abaixo:

```
rem Read an optional configuration file.
if "%RUN_CONF%" == "x" (
    set "RUN_CONF=%DIRNAME%\run_novainstancia.conf.bat"
)
if exist "%RUN_CONF%" (
    call "%RUN_CONF%" %*
)

:cmdInstall
jbosservice.exe -imwcd %SVCSNAME% "%DIRNAME%" "%SVCDISP%" "%SVCDISC%" service_novainstancia.bat
if not errorlevel 0 goto errExplain
echo Service %SVCDISP% installed
goto cmdEnd

jbosservice.exe -p 1 "Starting %SVCDISP%" > run_novainstancia.log
call run_novainstancia.bat < .r_novainstancia.lock >> run_novainstancia.log 2>&1
jbosservice.exe -p 1 "Shutdown %SVCDISP% service" >> run_novainstancia.log
del .r_novainstancia.lock
goto cmdEnd

:cmdStop
REM Executed on service stop
echo Y > .s_novainstancia.lock
jbosservice.exe -p 1 "Shutting down %SVCDISP%" > shutdown_novainstancia.log
call shutdown -S < .s_novainstancia.lock >> shutdown_novainstancia.log 2>&1
jbosservice.exe -p 1 "Shutdown %SVCDISP% service" >> shutdown_novainstancia.log
del .s_novainstancia.lock
goto cmdEnd

:cmdRestart
REM Executed manually from command line
REM Note: We can only stop and start
echo Y > .s_novainstancia.lock
jbosservice.exe -p 1 "Shutting down %SVCDISP%" >> shutdown_novainstancia.log
call shutdown -S < .s_novainstancia.lock >> shutdown_novainstancia.log 2>&1
del .s_novainstancia.lock
```

Outro ponto importante é modificar o nome do serviço a ser instalado nos pontos destacados abaixo:

```

if "%SVCNAME%" == "x" (
    set "SVCNAME=JBAS61SVC-NOVAINSTANCIA"
)
set SVCDISP=JBoss Application Server 6.1-NOVAINSTANCIA
set SVCDESC=JBoss Application Server 6.1.0 Final/Platform: Windows %PROCESSOR_ARCHITECTURE%
set NOPAUSE=Y

```

Após estes ajustes, o procedimento de instalação do serviço é o mesmo referenciado anteriormente.

e. Configurações de Firewall

A aplicação CCTServer tem como cliente a aplicação CCTDesktop, que é um cliente remoto executado, geralmente, fora da rede onde o servidor será instalado.

Para ambientes que possuem um firewall entre o ambiente externo e o servidor é necessário que sejam liberadas algumas portas de serviços que serão acessados pela aplicação CCTDesktop.

Os serviços acessados pela aplicação cliente, e suas respectivas portas, são:

Conector HTTP

Porta: 8080

Serviço de Nomes e Diretórios (JNDI)

Portas: 1098, 1099

Java Message Service (JMS)

Porta: 5445

Importante notar que estas são as portas acessadas a partir da aplicação cliente a partir da rede externa. No entanto o servidor de aplicações JBoss abre sockets em portas aleatórias para seu processamento interno. Caso haja conflitos pode ser necessário a liberação de um range de portas acima de 4000.

f. Ajustando a Quantidade de Threads no Servidor

Em certas ocasiões se faz necessário o ajuste da quantidade de threads que o servidor mantém para atender às requisições.

Esta configuração pode ser encontrada no arquivo C:\totvs\jboss-6.1.0.Final\server\default\deploy\jbossweb.sar\server.xml. Neste arquivo o trecho responsável por esta configuração está na definição do conector HTTP 1.1, como mostrado abaixo:

```
<!-- A HTTP/1.1 Connector on port 8080 -->
```

```

<Connector protocol="HTTP/1.1"
    port="${jboss.web.http.port}"
    address="${jboss.bind.address}"
    redirectPort="${jboss.web.https.port}"
    maxThreads="150"
    maxHttpHeaderSize="8192"
    acceptCount="100"/>

```

Abaixo segue a descrição das duas propriedades que podem ser ajustadas:

acceptCount: O comprimento máximo da fila de espera para os pedidos de conexão, quando todas as possíveis threads de processamento de solicitação estão em uso. Todos os pedidos recebidos quando a fila está cheia serão recusados. O valor padrão é 10.

maxThreads: O número máximo de threads de processamento de solicitação a serem criadas para este conector, que,

portanto, determina o número máximo de solicitações simultâneas que podem ser atendidos. Se não for especificado, este atributo é definido como 200.

Maiores informações podem ser encontradas na documentação oficial do servidor JBoss: <http://docs.jboss.org/jbossweb/2.1.x/config/http.html>
(<http://docs.jboss.org/jbossweb/2.1.x/config/http.html>)

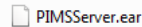
g. Problemas na instalação

Caso ocorra algum problema na instalação, consultar o manual Online citado nas referências para resolução, e em segundo nível o suporte da TOTVS.

1.4 Instalação e Configuração da Aplicação CCT Server

A aplicação CCTServer é desenvolvida em Java, como um componente que deve ser implantado no servidor de aplicações JBoss 6.1.0.

O arquivo da aplicação, disponibilizado para implantação, chama-se PIMSServer.ear.

 PIMSServer.ear

Durante a implantação, ou então quando for necessário realizar uma atualização da aplicação, este arquivo deverá ser copiado para o diretório deploy, cujo caminho está descrito a seguir:

C:\totvs\jboss-6.1.0.Final\server\default\deploy

NOTA: lembre-se que este caminho pode mudar de acordo com o local escolhido para a instalação do servidor de aplicações JBoss.

NOTA: quando for realizar a atualização da aplicação, tenha certeza de ter realizado uma cópia de segurança da versão que está sendo substituída, com uma cópia do arquivo em local adequado antes de sua remoção para a substituição pelo novo arquivo.

a. Datasource para conexões com o banco de dados.

Para que a aplicação funcione adequadamente, é necessário configurar o acesso a alguns servidores de bancos de dados através de arquivos de configuração de datasource. Os arquivos em questão estão no diretório deploy, citado acima. São eles: PIMSCS-oracle-xa-ds.xml, PIMSLT-oracle-xa-ds.xml e PIMSSL-oracle-xa-ds.xml.

Nestes arquivos devem ser informados o IP (ou nome de rede) do servidor onde está configurado o banco de dados, a porta de conexão, o nome da instância, usuário e senha de acesso. O exemplo abaixo, (específico para o Oracle e com dados fictícios) apresenta estas informações em destaque:

```
<datasources>
  <xa-datasource>
    <jndi-name>PIMSCSDS</jndi-name>
    <isSameRM-override-value>false</isSameRM-override-value>
    <xa-datasource-class>oracle.jdbc.xa.client.OracleXADataSource</xa-datasource-class>
    <xa-datasource-property name="URL">jdbc:oracle:thin:@localhost:1521:ORACLE</xa-datasource-property>
    <xa-datasource-property name="User">TESTE</xa-datasource-property>
    <xa-datasource-property name="Password">PIMSCS</xa-datasource-property>
    <exception-sorter-class-name>
      org.jboss.resource.adapter.jdbc.vendor.OracleExceptionSorter
    </exception-sorter-class-name>
    <no-tx-separate-pools/>
    <min-pool-size>1</min-pool-size>
    <max-pool-size>5</max-pool-size>
    <idle-timeout-minutes>5</idle-timeout-minutes>
    <metadata>
      <type-mapping>Oracle9i</type-mapping>
    </metadata>
  </xa-datasource>
</datasources>
```

A sintaxe da propriedade URL do exemplo acima se aplica caso a conexão estiver identificada por SID. Se a conexão estiver identificada por um nome de serviço a sintaxe deve ser:

```
<datasources>
  <xa-datasource>
    <jndi-name>PIMSCSDS</jndi-name>
    <isSameRM-override-value>false</isSameRM-override-value>
    <xa-datasource-class>oracle.jdbc.xa.client.OracleXADataSource</xa-datasource-class>
    <xa-datasource-property name="URL">jdbc:oracle:thin:@//localhost:1521/ORACLE</xa-datasource-property>
    <xa-datasource-property name="User">TESTE</xa-datasource-property>
    <xa-datasource-property name="Password">PIMSCS</xa-datasource-property>
    <exception-sorter-class-name>
      org.jboss.resource.adapter.jdbc.vendor.OracleExceptionSorter
    </exception-sorter-class-name>
    <no-tx-separate-pools/>
    <min-pool-size>1</min-pool-size>
    <max-pool-size>5</max-pool-size>
    <idle-timeout-minutes>5</idle-timeout-minutes>
    <metadata>
      <type-mapping>Oracle9i</type-mapping>
    </metadata>
  </xa-datasource>
</datasources>
```

b. Configuração de datasource para conexões com o banco de dados em RAC.

Quando o servidor de banco de dados estiver configurado em RAC, é necessário

que o cliente do Oracle esteja instalado na máquina, e que seu diretório de instalação esteja referenciado nas variáveis de ambiente, como PATH, por exemplo.

Além disso, existem mudanças na maneira de se configurar o datasource. Tomemos como exemplo a seguinte entrada no tsnames.ora:

```
PROD.NOME.REDE =
(DESCRIPTION =
  (LOAD_BALANCE=on)
  (FAILOVER=on)
  (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = TCP)(HOST = 10.0.0.25)(PORT = 1541))
    (ADDRESS = (PROTOCOL = TCP)(HOST = 10.0.0.26)(PORT = 1541))
    (ADDRESS = (PROTOCOL = TCP)(HOST = 10.0.0.27)(PORT = 1541))
    (ADDRESS = (PROTOCOL = TCP)(HOST = 10.0.0.28)(PORT = 1541))
  )
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = NOME_SERVICO)
    (FAILOVER_MODE= (TYPE=select) (METHOD=basic))
  )
)
```

Exemplo de configuração resultante no datasource:

```
<datasources>
  <xa-datasource>
    <jndi-name>PIMSLTDS</jndi-name>
    <isSameRM-override-value>>false</isSameRM-override-value>
    <xa-datasource-class>oracle.jdbc.xa.client.OracleXADataSource
    </xa-datasource-class>
    <xa-datasource-property name="URL">
      jdbc:oracle:oci:8 (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP) (HOST=10.0.0.25) (PORT=15
      21)) (ADDRESS=(PROTOCOL=TCP) (HOST=10.0.0.26) (PORT=1521)) (ADDRESS=(PROTOCOL=TCP
      ) (HOST=10.0.0.27) (PORT=1521)) (ADDRESS=(PROTOCOL=TCP) (HOST=10.0.0.28) (PORT=152
      1)) (LOAD_BALANCE=yes) (CONNECT_DATA=(SERVER=DEDICATED) (SERVICE_NAME=pimscon) (F
      AILOVER_MODE=(TYPE=SELECT) (METHOD=BASIC) (RETRIES=180) (DELAY=5))))
    </xa-datasource-property>
    <xa-datasource-property name="User">TESTE</xa-datasource-property>
    <xa-datasource-property name="Password">PIMSCS</xa-datasource-property>
    <no-tx-separate-pools/>
    <min-pool-size>1</min-pool-size>
    <max-pool-size>5</max-pool-size>
    <idle-timeout-minutes>5</idle-timeout-minutes>
    <valid-connection-checker-class-name>
      org.jboss.resource.adapter.jdbc.vendor.OracleValidConnectionChecker
    </valid-connection-checker-class-name>
    <exception-sorter-class-name>
      org.jboss.resource.adapter.jdbc.vendor.OracleExceptionSorter
    </exception-sorter-class-name>
    <metadata>
      <type-mapping>Oracle9i</type-mapping>
    </metadata>
  </xa-datasource>
</datasources>
```

c. Criptografia de Senhas no Datasource

É possível criptografar a senha exposta no Datasource para uma maior segurança, principalmente em ambientes de produção.

Para isso é necessário, inicialmente, criar um módulo de segurança no arquivo JBOSS_HOME/server/default/conf/login-config.xml.

A criação do módulo de segurança consiste na inclusão de um trecho de XML, como o que está exemplificado abaixo, no arquivo login-config.xml:

```
<application-policy name="PIMSCSDSRealm">
  <authentication>
    <login-module
      code="org.jboss.resource.security.SecureIdentityLoginModule" flag="required">
      <module-option name="userName">TESTE</module-option>
      <module-option
        name="password">6ee9d08cc488f63</module-option>
```

```

</module-option>
name="managedConnectionFactoryName">jboss.jca:service=XATxCM,name=PIM
SCSDS</module-option>

</login-module>

</authentication>
</application-policy>

```

Na configuração acima destacam-se os seguintes itens:

PIMSCSDSRealm: este é o nome do módulo de segurança que está sendo criado. Este nome é importante pois será referenciado no datasource, posteriormente.

TESTE: este é o nome do usuário de banco de dados a ser utilizado para a conexão.

6ee9d08cc488f63: esta é a senha já criptografada do usuário utilizado para acessar o banco de dados. Para realizar a criptografia da senha acesse o diretório de instalação do JBoss pelo prompt de comando e execute o comando abaixo, substituindo "SENHA" pela senha do usuário. O comando deve ser digitado todo em uma única linha:

```

java -cp client/jboss-logging.jar;lib/jbosssx.jar
org.jboss.resource.security.SecurityIdentityLoginModule SENHA

```

Exemplo de execução do comando:

```

E:\Servidores\jboss-6.1.0.Final>java -cp client/jboss-logging.jar;lib/jbosssx.jar
org.jboss.resource.security.SecurityIdentityLoginModule SENHA
Encoded password: 6ee9d08cc488f63

```

PIMSCSDS: Este deve ser o nome JNDI do datasource configurado para o banco de dados que utilizará o módulo de segurança.

Abaixo está um exemplo do módulo acima configurado dentro do arquivo login-config.xml:

```

<application-policy name="JmsXARealm">
  <authentication>
    <login-module code="org.jboss.resource.security.ConfiguredIdentityLoginModule"
      flag="required">
      <module-option name="principal">guest</module-option>
      <module-option name="userName">guest</module-option>
      <module-option name="password">guest</module-option>
      <module-option name="managedConnectionFactoryName">jboss.jca:service=TxCM,name=JmsXA</module-option>
    </login-module>
  </authentication>
</application-policy>

<application-policy name="PIMSCSDSRealm">
  <authentication>
    <login-module code="org.jboss.resource.security.SecurityIdentityLoginModule" flag="required">
      <module-option name="userName">TESTE</module-option>
      <module-option name="password">6ee9d08cc488f63</module-option>
      <module-option name="managedConnectionFactoryName">jboss.jca:name=PIMSCSDS,service=XATxCM</module-option>
    </login-module>
  </authentication>
</application-policy>

```

```

<!-- A template configuration for the jmx-console web application. This
defaults to the UsersRolesLoginModule the same as other and should be
changed to a stronger authentication mechanism as required.
-->

```

```

<application-policy name="jmx-console">
  <authentication>
    <login-module code="org.jboss.security.auth.spi.UsersRolesLoginModule"
      flag="required">

```

Abaixo estão exemplos das configurações do datasource (conexão direta e RAC) utilizando o módulo de segurança:

```

<!-- ===== -->
<!-- Datasource config for Oracle originally from Steven Coy -->
<!-- ===== -->
<datasources>
  <xa-datasource>
    <jndi-name>PIMSCSDS</jndi-name>
    <isSameRM-override-value>false</isSameRM-override-value>
    <xa-datasource-class>oracle.jdbc.xa.client.OracleXADataSource</xa-datasource-class>
    <xa-datasource-property name="URL">jdbc:oracle:thin:@localhost:1521:ORACLE
    </xa-datasource-property>
    <security-domain>PIMSCSDSRealm</security-domain>
    <exception-sorter-class-name>org.jboss.resource.adapter.jdbc.vendor.OracleExceptionSorter
    </exception-sorter-class-name>
    <no-tx-separate-pools/>
    <min-pool-size>1</min-pool-size>
    <max-pool-size>5</max-pool-size>
    <idle-timeout-minutes>5</idle-timeout-minutes>
    <metadata>
      <type-mapping>Oracle9i</type-mapping>
    </metadata>
  </xa-datasource>
</datasources>

```

d. Limpeza de Diretórios Temporários

Sempre que o servidor for parado, seja para atualização de versão ou por qualquer outro motivo, recomenda-se que alguns arquivos temporários sejam removidos do servidor antes de iniciá-lo novamente.

Os arquivos dos diretórios: log, temp e work.

Se for necessário, pode-se realizar um backup dos arquivos de log antes da exclusão.

NOTA: o diretório data nunca deve ser excluído, pois neste diretório se encontra um banco de dados embutido utilizado pela aplicação.

e. Arquivos de Log e Verbosidade de Logs

Dentro do diretório log encontram-se os logs de execução do aplicativo CCTServer.

Cada módulo terá seu arquivo de log exclusivo, identificado pelo nome. Na imagem apresentada o arquivo CCT.log possui todos os logs gerados pelo módulo CCT.

Além dos arquivos de log específicos a cada módulo, poderão existir ainda, se necessário, arquivos específicos para uma execução em particular.

À quantidade de detalhes dos logs gerados pela aplicação damos o nome de verbosidade. A configuração dos logs pode ser mais verbosa (gerar mais logs, com mais detalhes) ou menos verbosa (gerar menos logs, com menos detalhes).

Em um ambiente de produção recomenda-se uma geração de logs menos verbosa, que registre apenas os erros e as informações mais relevantes dos processos executados.

Há quatro configurações comumente usadas para a verbosidade dos logs: DEBUG, INFO, WARN e ERROR, sendo DEBUG a mais verbosa e ERROR a menos verbosa.

O arquivo onde se pode configurar a verbosidade dos logs é o jboss-logging.xml, que está no diretório deploy.

Neste arquivo podem-se encontrar as seções em destaque na imagem abaixo, em que se pode configurar a verbosidade dos logs de cada arquivo:

```
<logger category="br.agr.proxima.pimsserver.oct">
  <level name="ERROR"/>
  <handlers>
    <handler-ref name="OCT"/>
  </handlers>
</logger>

<logger category="br.agr.proxima.pimsserver.oct.bec">
  <level name="ERROR"/>
  <handlers>
    <handler-ref name="BEC"/>
  </handlers>
</logger>

<logger category="br.agr.proxima.pimsserver.oct.gam">
  <level name="ERROR"/>
  <handlers>
    <handler-ref name="GAM"/>
  </handlers>
</logger>

<logger category="br.agr.proxima.pimsserver.oct.logtra">
  <level name="ERROR"/>
  <handlers>
    <handler-ref name="LOGTRA"/>
  </handlers>
</logger>
```

Nos exemplos a verbosidade está configurada para ERROR, mas em algumas situações, de testes, este valor pode ser alterado para DEBUG. A recomendação é que em ambiente de produção este valor fique configurado para ERROR.

NOTA: os arquivos de logs gerados serão organizados por datas. Recomenda-se uma limpeza periódica dos arquivos mais antigos à medida que o espaço em disco for ficando ocupado por muitos arquivos.

f. Problemas conhecidos

I. Falha na utilização de comandos

paralelos com a utilização do "|" (pipe).

Há relatos de problemas com alguns servidores que, por algum motivo ainda não conhecido, não aceitam comandos paralelos com a utilização do "|" (pipe).

Na execução do serviço, service.bat, existe a linha de comando:

```
del .r.lock 2>&1 | findstr /C:"being used" > nul
```

E para que o serviço possa ser executado corretamente é necessário incluir, no início da linha, a instrução `cmd /c`, conforme mostra a linha abaixo:

```
cmd /c del .r.lock 2>&1 | findstr /C:"being used" > nul
```

Este comando habilita para pegar a saída do primeiro comando e pode ser usada no segundo.

Para saber se um servidor está com o mesmo problema, o resultado do comando abaixo será este erro:

```
C:\>dir | more
```

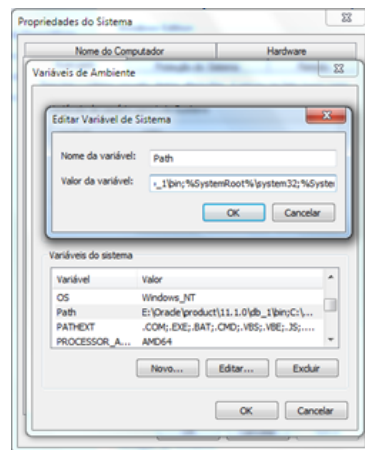
The input line is too long.

II. 'findstr' não é reconhecido como um comando interno ou externo (Windows)

Há alguns casos em que, ao tentar iniciar o serviço do JBoss, ou até mesmo executar o arquivo `run.bat`, a aplicação não é inicializada, e a mensagem a seguir é apresentada:

'findstr' não é reconhecido como um comando interno ou externo, um programa operável, ou um arquivo de lotes.

Para solucionar este problema é preciso garantir que a variável de ambiente `PATH` contenha o caminho para o diretório `System32`, como na imagem abaixo:



Na imagem acima `%SystemRoot%\system32` poderia ser `C:\Windows\System32`, por exemplo.



Esse artigo foi útil?



Usuários que acharam isso útil: 0 de 0

0 Comentários

Por favor, entre (https://centraldeatendimento.totvs.com/hc/pt-br/signin?return_to=https%3A%2F%2Fcentraldeatendimento.totvs.com%2Fhc%2Fpt-br%2Farticles%2F115004056288-Manual-de-Instala%25C3%25A7%25C3%25A3o-CCT-Server) para comentar.

Como podemos ajudar?

Entre em contato e fale conosco!

