

Guia de Configuração

TOTVS Datasul 11



Alerta

Faça backup antes de iniciar a alteração em qualquer arquivo de configuração. Este backup deve ser feito para um diretório fora do Jboss. Ao editar arquivos XML, verifique através do browser se a sintaxe está correta. Arquivos com sintaxe incorreta causarão erros na carga do Jboss.

1.	Prefácio	3
1.1.	Finalidade	3
1.2.	Público Alvo	3
1.3.	Documentações Importantes	3
1.4.	Como utilizar este guia	3
1.5.	Atualização do guia	4
2.	Arquivos de Configuração	4
2.1.	Bancos de Dados	4
2.2.	AppServer	5
2.3.	Log	5
2.4.	License Server	7
2.5.	Configurações do Serviço Jboss	7
2.5.1.	Linux	8
2.6.	Parâmetros de Memória	8
2.7.	Metadados	9
2.8.	Time Out Flex	10
2.9.	Josso	10
2.10.	Portas	10
2.11.	Sessão Progress	11
2.12.	Localização dos arquivos	12

1. Prefácio

1.1. Finalidade

Este guia tem como finalidade descrever as configurações do produto Datasul 11, demonstrando o arquivo onde cada configuração é feita, sem entrar em detalhes sobre a configuração do Jboss. Informações adicionais podem ser localizadas no capítulo “Documentações Importantes”.

1.2. Público Alvo

A complexidade do processo requer profissionais com conhecimento da arquitetura do produto Datasul 11, Progress, Java, Flex, Jboss e ambiente de rede.

1.3. Documentações Importantes

- ✓ Manuais Jboss
 - Installation Guide;
 - Getting Started Guide;
 - Server Configuration Guide.

Estas e outras documentações podem ser obtidas em Jboss AS Documentation e Jboss Wiki.

- ✓ Manuais Totvs
 - Guia de Instalação Datasul 11.5.2 ([link](#));

1.4. Como utilizar este guia

As configurações neste guia estão capituladas por funcionalidade. No decorrer do guia citaremos apenas o nome do arquivo onde a configuração é feita. No capítulo

“Localização dos Arquivos” estão listados os caminhos completos para a localização dos arquivos.

1.5. Atualização do guia

A versão atualizada deste guia encontra-se disponível através do EConsultor, em formato de Kbase. Sempre que for utilizá-lo recomendamos verificar este [Kbase](#) para garantir estar com a versão atualizada.

2. Arquivos de Configuração

2.1. Bancos de Dados

Os bancos de dados “emsdev”, “emsfnd”, “mdmerge”, “mdtfrw”, “finance”, “payroll”, “ems2sor” e “ems2cad” são acessados diretamente pelo Jboss. Para este acesso são criados datasources, através do arquivo progress-ds.xml (ambiente Progress) ou oracle-ds.xml (ambiente Oracle).

Listaremos os atributos mais importantes e suas devidas funções:

jndi-name: Nome do Datasource (Não pode ser alterado);

username: Usuário para conexão ao banco;

password: Senha para conexão ao banco;

connection-url: String de conexão JDBC ao banco de dados. Ex:

jdbc:datadirect:openedge://JAGUARIBE:23618;databaseName=emsdev

min-pool-size: Número mínimo de conexões ativas ao banco de dados;

max-pool-size: Número máximo de conexões ativas ao banco de dados;

Pode existir mais de uma conexão para o mesmo banco de dados, porém, com o JNDI diferente. Se for necessário fazer alguma alteração no arquivo, tome o cuidado de fazê-la no local certo e para todos os datasources necessários.

Para ambiente Progress o username deve ser “PUB”, pois as tabelas estão criadas abaixo deste usuário.

2.2. AppServer

A configuração do AppServer é feita no arquivo `datasul_framework.properties`.

```
progress.server.name=JAGUARIBE
```

```
progress.server.port=5162
```

```
progress.server.application=datasul-1151-progress-8080
```

```
progress.server.maxconnections=5
```

Onde:

`progress.server.name`: Servidor onde está configurado o broker AppServer;

`progress.server.port`: Porta do NameServer;

`progress.server.application`: Nome do broker AppServer;

`progress.server.maxconnections`: Quantidade máxima de agentes que o Jboss poderá utilizar.

Caso seja exibida a mensagem “Erro ao obter conexão” durante a execução de um programa Flex, verifique se o broker AppServer configurado neste arquivo está ativo e configurado corretamente.

2.3. Log

No JBoss a geração de logs é controlada pelo arquivo `jboss-log4j.xml`. Este arquivo define um conjunto de "appenders", que são basicamente as saídas de log. Por padrão, o JBoss está configurado com saídas para o console e para um arquivo de log (`server.log`).

O nível de log a ser configurado depende do tipo de informação que se quer obter. O produto Datasul 11 está configurado com o log para "ERROR", o que significa que somente as mensagens de erro serão exibidas no arquivo. Esta é a configuração adequada para um ambiente de produção, para que mensagens informativas não sejam gravadas em log, o que pode tornar o arquivo muito grande e difícil de ler.

Existem seis níveis de log: TRACE, DEBUG, INFO, WARN, ERROR e FATAL. Quando um desses níveis é definido no arquivo, todas as mensagens acima do nível escolhido também serão registradas. Quer dizer, se for escolhido o nível "INFO", todas as

mensagens iguais e superiores a este nível serão escritas em log (INFO, WARN, ERROR e FATAL).

O exemplo abaixo é de um appender com saída para arquivo, configurado para "INFO":

```
<appender name="FILE" class="org.jboss.logging.appender.DailyRollingFileAppender">
  <errorHandler class="org.jboss.logging.util.OnlyOnceErrorHandler"/>
  <param name="File" value="$jboss.server.log.dir/server.log"/>
  <param name="Append" value="false"/>
  <param name="Threshold" value="INFO"/>
  <param name="DatePattern" value="".yyyy-MM-dd-HH"/>
  <layout class="org.apache.log4j.PatternLayout">
    <param name="ConversionPattern" value="%d %-5p [%c] %m%n"/>
  </layout>
</appender>
```

Para alterar o nível de log, basta alterar o "Threshold" de "INFO" para o valor desejado. Após a alteração não é necessário reiniciar a instância. O jboss-log4j.xml é o único arquivo de configuração do JBoss que não exige restart da instância. A alteração entra em vigor no máximo em 60 segundos após salvar o arquivo.

Na configuração padrão do produto o log rotate será feito a cada hora. Ao início de cada hora, o arquivo corrente de log será renomeado e o log será gerado em outro arquivo, com o mesmo nome. Configure o log conforme sua necessidade, visando não gerar arquivos com tamanho exagerado. Pode-se configurar o log rotate por: mês, semana, dia, metade do dia, hora e minuto. Para fazer esta alteração, mudar o valor da opção "DatePattern", conforme a tabela abaixo:

DatePattern	Período
' . 'yyyy-MM	Rotaciona o log a cada mês
' . 'yyyy-ww	Rotaciona o log a cada semana
' . 'yyyy-MM-dd	Rotaciona o log a cada dia
' . 'yyyy-MM-dd-a	Rotaciona o log a meia noite e meio dia a cada dia
' . 'yyyy-MM-dd-HH	Rotaciona o log a cada hora
' . 'yyyy-MM-dd-HH-mm	Rotaciona o log a cada minuto

Além dos Appenders podem ser criadas categorias. As categorias possuem configurações específicas para determinadas aplicações. Por exemplo, você pode manter o nível de log como "INFO", e configurar uma categoria para as mensagens de determinada aplicação

como “WARN”. O exemplo abaixo é de uma categoria do componente “org.jgroups” (cluster) configurado como “WARN”:

```
<category name="org.jgroups">  
  <priority value="WARN"/>  
</category>
```

Dessa forma, para todas as aplicações serão exibidas apenas as mensagens com o nível “ERROR”, exceto as mensagens da aplicação “org.jgroups”, que possui o nível de log “WARN”.

Informações sobre os níveis de log podem ser obtidas [aqui](#).

2.4. License Server

A configuração do License Server (LS) é feita no arquivo “datasul_framework.properties”, nos parâmetros abaixo:

license.server=calixto

license.port=7777

license.timeout=20

license.showlog=none

Onde:

license.server: Endereço IP ou nome do LS;

license.port: Porta configurada para o LS (default 5555);

license.timeout: Tempo (segundos) para conexão com o LS. Se este tempo for atingido sem receber uma resposta do LS, será solicitada a chave de emergência;

license.showlog: Nível de log para as mensagens do LS. Os tipos permitidos são: none, all, error, debug, info (default: none).

Se não houver comunicação com o LS ou a licença estiver expirada, será solicitada a chave de emergência.

2.5. Configurações do Serviço Jboss

Na instalação padrão do produto (ambiente Windows) é criado pela mídia o serviço “TOTVS Datasul - <ID> - 8x80”, que é responsável pela carga do Jboss. Ao iniciar o serviço é executado o script “service-8x80.bat”, o qual gerencia o serviço, além de efetuar a chamada da linha de execução do Jboss.

No arquivo “run.bat” está configurada a variável “JAVA_OPTS”. Esta é a variável de ambiente padrão para a configuração de parâmetros do Java. Observando os arquivos “service-8x80.bat” e “run.bat” podemos verificar que esta variável é definida várias vezes, geralmente concatenando o valor antigo e incluindo outros parâmetros e valores. Caso exista a necessidade de incluir um novo parâmetro, você pode incluir em qualquer um destes dois arquivos. A diferença básica é que o arquivo “run.bat” é utilizado para todas as instâncias, sendo que os parâmetros definidos neste arquivo serão os mesmos para todas as instâncias. Já o arquivo “service-8x80.bat” é utilizado por instância, nos dando a possibilidade de definir valores diferenciados para cada uma das instâncias.

2.5.1. Linux

O Jboss é instalado através da mídia somente para ambiente Windows. Para ambientes onde o Jboss está configurado em um servidor Linux, a carga do Jboss é feita através do script “run.sh”, sendo que as configurações da variável JAVA_OPTS devem ser feitas no arquivo “run.conf”.

Para a configuração do Jboss em servidor Linux recomendamos a leitura deste [kbase](#).

2.6. Parâmetros de Memória

Para obter a performance ideal na utilização do produto, é necessário configurar corretamente os parâmetros de memória da JVM. Estas configurações estão definidas na variável “JAVA_OPTS”, citada anteriormente. No padrão instalado pela mídia os parâmetros de memória estão definidos no “run.bat”, indicando que os valores ali alterados serão para todas as instâncias. Recomendamos remover esta configuração deste arquivo e colocá-la no “service-8x80.bat” (de acordo com a instância), para que a configuração de memória seja diferente para cada instância.

Segue abaixo exemplo da linha com os parâmetros de memória do Java (padrão da mídia de instalação):

```
set JAVA_OPTS=%JAVA_OPTS% -Xms128m -Xmx768m -Xss128k -  
XX:MaxPermSize=512m -XX:+UseParallelGC -XX:ParallelGCThreads=10
```

Abaixo uma breve descrição de cada um desses parâmetros:

Xms: Parâmetro de memória que define o mínimo de memória para a aplicação;

Xmx: Parâmetro de memória que define o máximo de memória para a aplicação;

Xss: Tamanho da pilha reservada para cada thread;

XX:MaxPermSize: Tamanho máximo da área de objetos permanentes em memória;

-XX:+UseParallelGC: Habilita o Garbage Collector para abrir várias thread e atuar paralelamente;

-XX:ParallelGCThreads: Número de thread que serão abertas para GC. Por default é aberto uma Thread para cada processador.

Em um ambiente de produção o ideal é que o valor do Xms e Xmx estejam iguais. Configurando dessa forma o Garbage Collector será executado com menor frequência. Para obter o valor ideal de memória é necessário um monitoramento da JVM. Para monitorar a utilização da memória, sugerimos a leitura deste [Post](#).

2.7. Metadados

As informações para a carga do Metadados estão localizadas no arquivo “startup-state.xml”. Os atributos estão listados abaixo:

metadatapath: Diretório onde encontram-se as informações para a carga do Metadados.

<generate-database>: Esse atributo indica ao servidor de aplicação (JBOSS) que ele deve iniciar gerando a base de dados do metadados. Lembrando que o servidor gerará os dados na base de acordo com os xmls contidos no atributo metadataPath que se encontra no mesmo arquivo. Pode receber os valores true (realiza a geração) ou false (ignora a geração).

<deploy-applications>: Esse atributo indica ao servidor de aplicação (JBOSS) que ele deve iniciar gerando todos os formulários do produto padrão. Pode receber os valores true (realiza a geração) ou false (ignora a geração).

Ao instalar ou atualizar o produto Datasul 11, na primeira carga do Jboss estes atributos estarão com o valor “true”, o que realizará a geração das informações. Se esta carga for efetuada sem erros, o valor será automaticamente alterado para “false”.

Esse arquivo é de extrema importância, pois além de gerar o start do produto ele pode ser usado para restaurar os padrões de fábrica.

2.8. Time Out Flex

A configuração do Time Out do Flex é feita através do arquivo “datasul_framework.properties”:

```
session.timeout=30
```

```
session.timeout.message=2
```

Onde:

session.timeout: Tempo (minutos) que a sessão Web fica ativa sem nenhuma interação do usuário com o sistema;

session.timeout.message: Indica quanto tempo (minutos) antes do session.timeout será exibida a mensagem ao usuário de que a sessão está prestes a expirar.

2.9. Josso

Josso, ou Java Open Single Sign-On, é uma infra-estrutura opensource que provê uma solução centralizada de autenticação e autorização de usuários. É o componente utilizado para o login dos usuários no produto Datasul 11.

As informações de conexão ao Josso encontram-se no arquivo “josso-agent-config.xml”. Neste arquivo serão encontradas três entradas com o nome do servidor e a porta. Caso seja necessário alterar o nome do servidor, essas entradas deverão ser alteradas.

No arquivo “josso-gateway-config.xml” estão as configurações de autenticação no Ldap. Ao editar o arquivo você encontrará o usuário utilizado para localizar as informações no Ldap, bem como a senha em clear text. Caso a senha seja alterada pode-se editar o arquivo e fazer a alteração manualmente. Outra alternativa é executar o script “Dominio.vbs”, que solicita em tela as informações e gera um arquivo (LogDominio.txt) com as informações necessárias para inserir no arquivo.

2.10. Portas

No arquivo “Jboss-service.xml” é definido o conjunto de portas a ser utilizado pela instância. No exemplo abaixo, o atributo “ServerName” está apontando para o conjunto de portas “ports-8080”:

```
<mbean code="org.jboss.services.binding.ServiceBindingManager"
  name="jboss.system:service=ServiceBindingManager">
  <attribute name="ServerName">ports-8080</attribute>
  <attribute name="StoreURL">${jboss.home.url}/binding/ports-bindings.xml</attribute>
  <attribute name="StoreFactoryClassName">
    org.jboss.services.binding.XMLServicesStoreFactory
  </attribute>
</mbean>
```

O atributo “StoreURL” aponta para o arquivo “ports-bindings.xml”, que é o arquivo onde se encontram os conjuntos de portas a serem utilizados. Pelo padrão instalado pela mídia existem seis conjuntos de portas definidos no arquivo: ports-8080, ports-8180, ports-8280, ports-8380, ports-8480 e ports-8580. A recomendação é alterar estas configurações somente para ambiente Linux (visto que a mídia não configura a instância para Linux) e quando necessitar de mais instâncias do que as portas configuradas no “ports-bindings.xml” permitem.

Se for necessário alterar alguma das portas, verifique nos arquivos “datasul_framework.properties” e “jboss-service.xml” se existe também alguma referência a esta mesma porta, pois que nestes arquivos também existem algumas portas configuradas. A lógica é somar 100 portas para a instância 8180, 200 para a instância 8280, etc.

2.11. Sessão Progress

Na primeira execução de um programa Progress no Datasul 11 em cada estação, a tela “Configuração de Acesso” será apresentada. No campo “Atalho de Execução” é possível selecionar o atalho para a execução dos programas Progress. As informações desta tela são armazenadas no arquivo “config.xml”. Neste arquivo, você pode alterar o atalho padrão ou ainda criar outros atalhos (para execução do clientlog, por exemplo).

Para criar um novo atalho duplique a tag <Shortcut>, localizada dentro da tag <LocalShortcuts>. Altere as informações desejadas na nova linha criada (Description, Ini, Pf e Alias) e salve o arquivo. Após a alteração é necessário reiniciar o Jboss para que a alteração efetuada tenha efeito.

Neste mesmo arquivo estão localizadas as informações para o acesso remoto ao produto (Citrix, GoGlobal ou Terminal Server). Para mais informações sobre esta configuração consulte o Guia de Instalação, capítulo “Acesso Remoto”.

A referência para a localização do arquivo “config.xml” encontra-se no arquivo “di.properties”. Ao abrir este arquivo é localizado o atributo “config.path”, que possui como valor o caminho completo para a localização do arquivo “config.xml”.

2.12. Localização dos arquivos

Ports-bindings.xml	Jboss_home\binding\ports-bindings.xml
Service-8x80.bat	Jboss_home\bin\service-8x80.bat
Run.bat	Jboss_home\bin\run.bat
Run.sh	Jboss_home\bin\run.sh
Run.conf	Jboss_home\bin\run.conf
Jboss-log4j.xml	Jboss_home\server\instance-8x80\conf\datasul\jboss-log4j.xml
Jboss-service.xml	Jboss_home\server\instance-8x80\conf\jboss-service.xml
Josso-agent-config.xml	Jboss_home\server\instance-8x80\conf\josso-agent-config.xml
Josso-gateway-config.xml	Jboss_home\server\instance-8x80\conf\josso-gateway-config.xml
Datasul_framework.properties	Jboss_home\server\instance-8x80\conf\datasul\datasul_framework.properties
Di.properties	Jboss_home\server\instance-8x80\conf\datasul\di.properties
Startup-state.xml	Jboss_home\server\instance-8x80\conf\datasul\startup-state.xml
Progress-ds.xml	Jboss_home\server\instance-8x80\deploy\progress-ds.xml
Oracle-ds.xml	Jboss_home\server\instance-8x80\deploy\oracle-ds.xml

Mail-service.xml	Jboss_home\server\instance-8x80\deploy\mail-service.xml
Config.xml	<Dir instalação>\ERP\scripts-8x80\config.xml