

JBoss Application Server 入門

～ Seasar2を動かして見よう！～

サイオステクノロジー株式会社 武田正英

の前に
CMです



CM中

JavaのWebアプリケーションフレームワーク
「**Wicket**」の解説書が出版されます。
翻訳ではなく、本日の13時から
「**Wicket**とシステム開発の現場」で
お話された矢野勉さんの書き下ろしになります。
私もレビューに参加してます。



渾身の作になるので読んで
文句があったら言って来い
との事です。

是非読んで下さい><



CM中

JavaのWebアプリケーションフレームワーク
「Wicket」の解説書が出版されます。
翻訳ではなく、本日の**13時から**
「Wicketとシステム開発の現場」で
お話された矢野勉さんの書き下ろしになります。



3/13発売 つまり昨日です。
新鮮ピチピチです。

是非読んで下さい＞＜

CM終了

sponsored by 矢野勉

矢野勉曰く、
おまいら Wicketだろjk WicketとS2の組み合わせもいいと思うよ。



Contents



JBossとは



Seasar2対応状況



JBoss5.0.0.GAでS2を動かして見よう



Appendix



自己紹介



名前: 武田正英 id:bose999

所属: サイオステクノロジー(株) OSSテクノロジーセンター

**情報システム部門、運用・構築、開発を経験し、
プロジェクトマネージャー兼全工程要員(プレイングマネージャー)
として経験を積んできました。**

**最近は過去の経験を生かして開発業務のコンサルタントとして
全国のシステム開発・運用の現場を支援して回ってます。**

ブログ: <http://bose.techie.jp>

技術wiki: <http://www26.atwiki.jp/memotech/>



自己紹介



名前: 武田正英 id: buse999

所属: サイオステクノロジー(株) OSSテクノロジーセンター

今日も九州のお客様先から
飛行機でとんぼ返りして
なんとか間に合いました ><

ブログ: <http://bose.techie.jp>

技術wiki: <http://www26.atwiki.jp/memotech/>



皆さんが使っているAPサーバは？

Tomcat ?



WebLogic ?



WebSphere ?



Glass Fish ?



Gernonimo ?



etc....



私が主に使っているAPサーバ

武田はJavaEE準拠のアプリケーションサーバ
JBossを主に使用してます！！

JBoss
a division of
Red Hat



JBossとは

開発 開始時

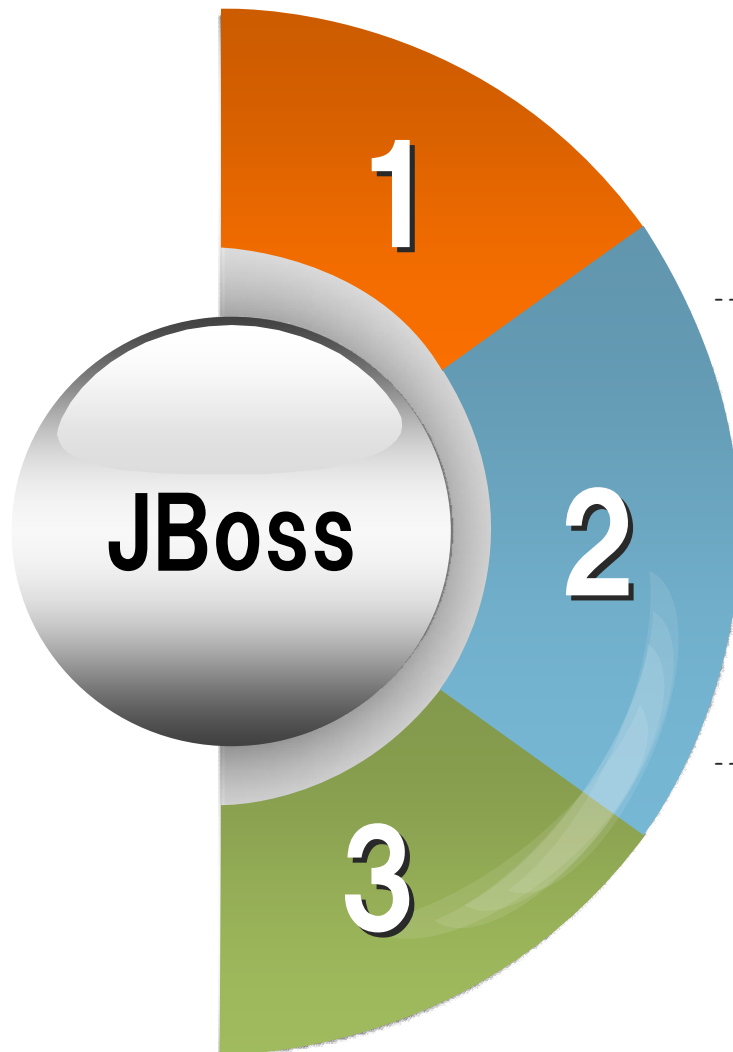
EJBOSSとしてEJBコンテナとして
開発がスタート。
後にEを取り除きJBossとして改名。

現在

現在はApplication Server、ESB、BPM
Engine、Rule Engine、開発フレームワーク、
開発IDEと多岐にわたりOSSとして
ミドルウェアが提供されている。



JBoss org版とJBoss EAPについて



1. OSS版と有償サポート版の関係

LinuxにおけるFedoraとRHELの関係と同様に現在はOSS版と商用サポート版の開発体制が取られている。

2. 有償サポート版の提供方法

OSSでのある時の最適な組み合わせをJBoss EAP (Enterprise Application Platform)としてスタック化して提供。

3. 有償サポートって？

パッチの提供、問い合わせの権等が提供される。
サイオステクノロジー社等から購入が可能。



JBossを使用する理由

例えばTomcatにJavaEEに存在する機能が
欲しくなったらどうします？



あれ？
この機能がない！

- ❖ マイクロコンテナアーキテクチャを採用していて、簡単に有効な機能のカスタマイズが可能。最初はTomcat相当の機能で動かして必要に応じて機能追加もJBossなら簡単。
- ❖ JavaEEサーバなら当たり前に持っている機能使わなきゃ損です。それとも作りますか？
- ❖ WebコンテナにTomcatベースのものを使用しているので、Apacheとの連携等のTomcatの知識が生かせます。



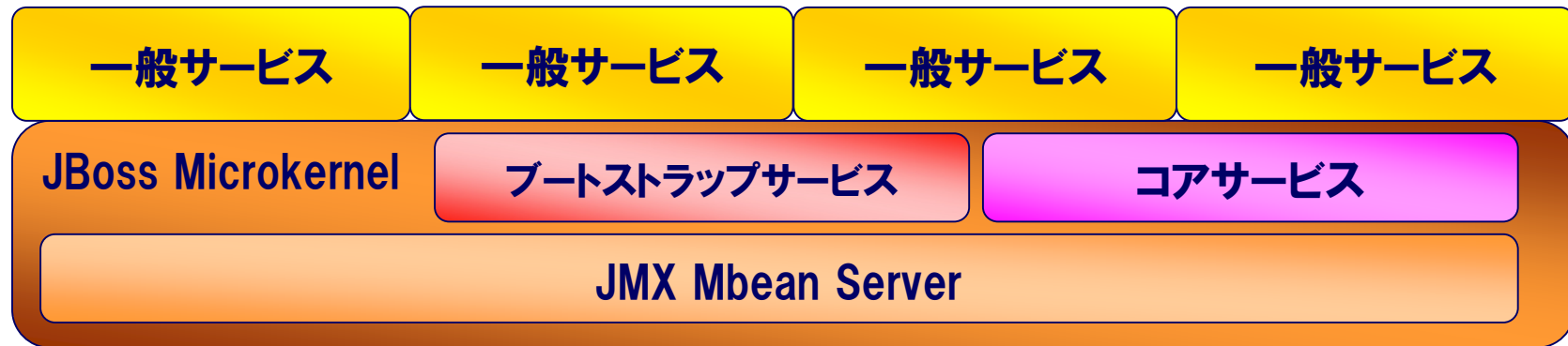
マイクロカーネルアーキテクチャ

**JBoss Application Serverは
マイクロカーネルアーキテクチャを採用している。**

**その為、必要最小限の起動サービスに対して、
必要なサービスを追加していきサーバを構成し、
カスタマイズするのが容易になっている。**



マイクロカーネルアーキテクチャ



❖ ブートストラップサービス

- 他のサービスをブートする為の最小限のサービス

❖ コアサービス

- ブートストラップサービス初期化後にデプロイされる
ホットデプロイ不可能なサービス(JNDI、トランザクション等)

❖ JMX MBean Server

- サービスをデプロイするサーバ

❖ 一般サービス

- コアサービス初期化後にデプロイされる。ホットデプロイが可能。
- この一つ一つがTomcatをベースにしたWebコンテナだったり、JMSサーバだったりします。



JBoss Application Server4

Mbean形式
の
一般サービス

Web
コンテナ
Tomcat
ベース

EJB
コンテナ

Java
Messaging
Service

Webサービス
JPA JSF
etc.

・ J2EE1.4準拠

JBoss
Application
Server
4.2.X

J2EE1.4準拠の機能を
内包している。

org版 JBoss Application Server 4.2.Xは
J2EE1.4に準拠した各種機能をアプリケーションサーバとして提供



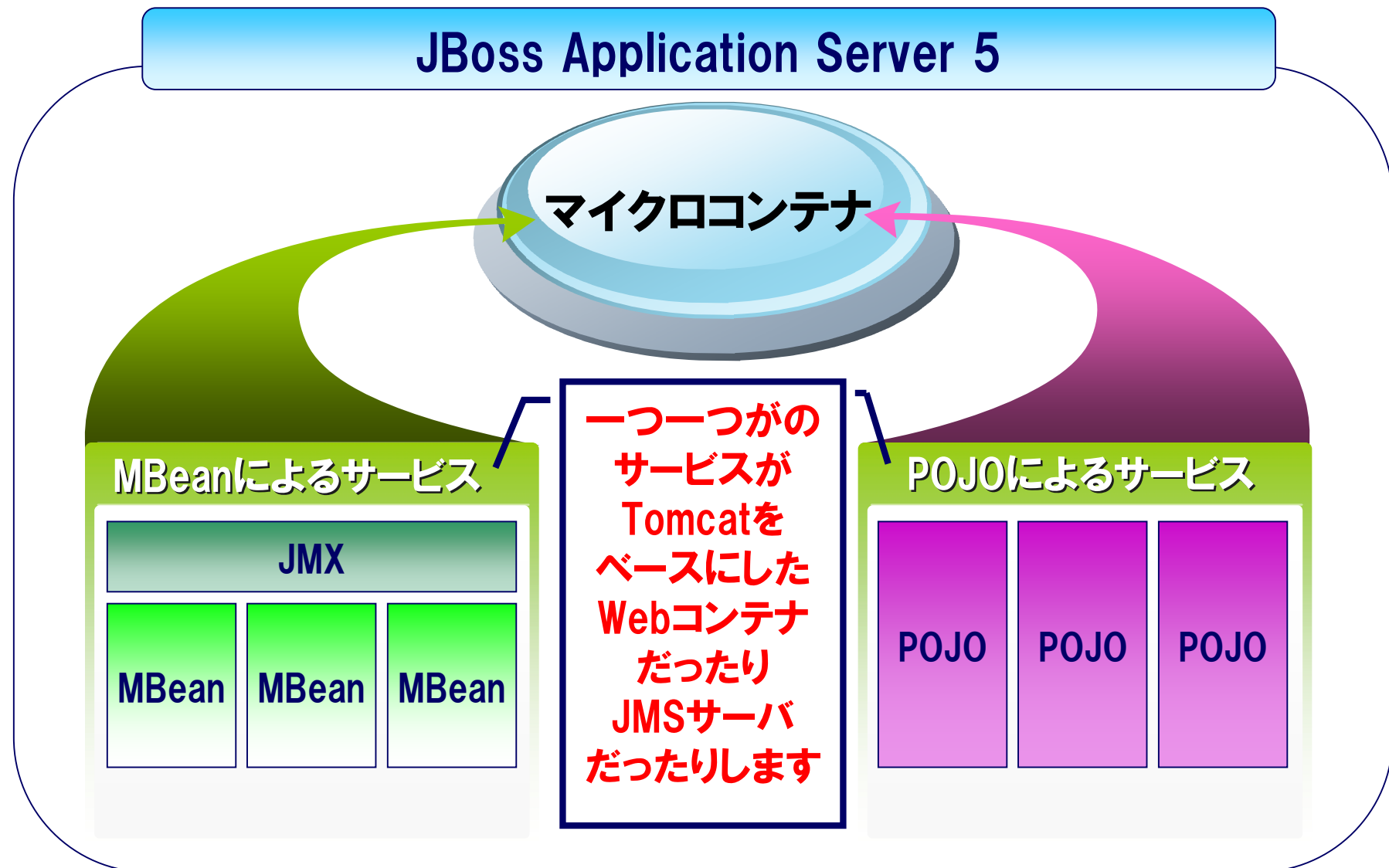
マイクロコンテナアーキテクチャー

JBoss Application Server 5からはマイクロコンテナを採用。

- JBossAS4まではマイクロカーネルだった。
Mbean形式でサービスを作成する必要があったが、
5からはPOJOでサービスが作成可能になった。**
- マイクロコンテナ上に色々なコンテナを構築できる。
例: Spring、Guice、JMX、OSGi**



マイクロコンテナアーキテクチャ





JBoss Application Server5

Mbean形式
または
POJO形式の
一般サービス

Web
コンテナ
Tomcat
ベース

EJB
コンテナ

Java
Messageing
Service

Webサービス
JPA JSF
etc.

・ JavaEE5準拠

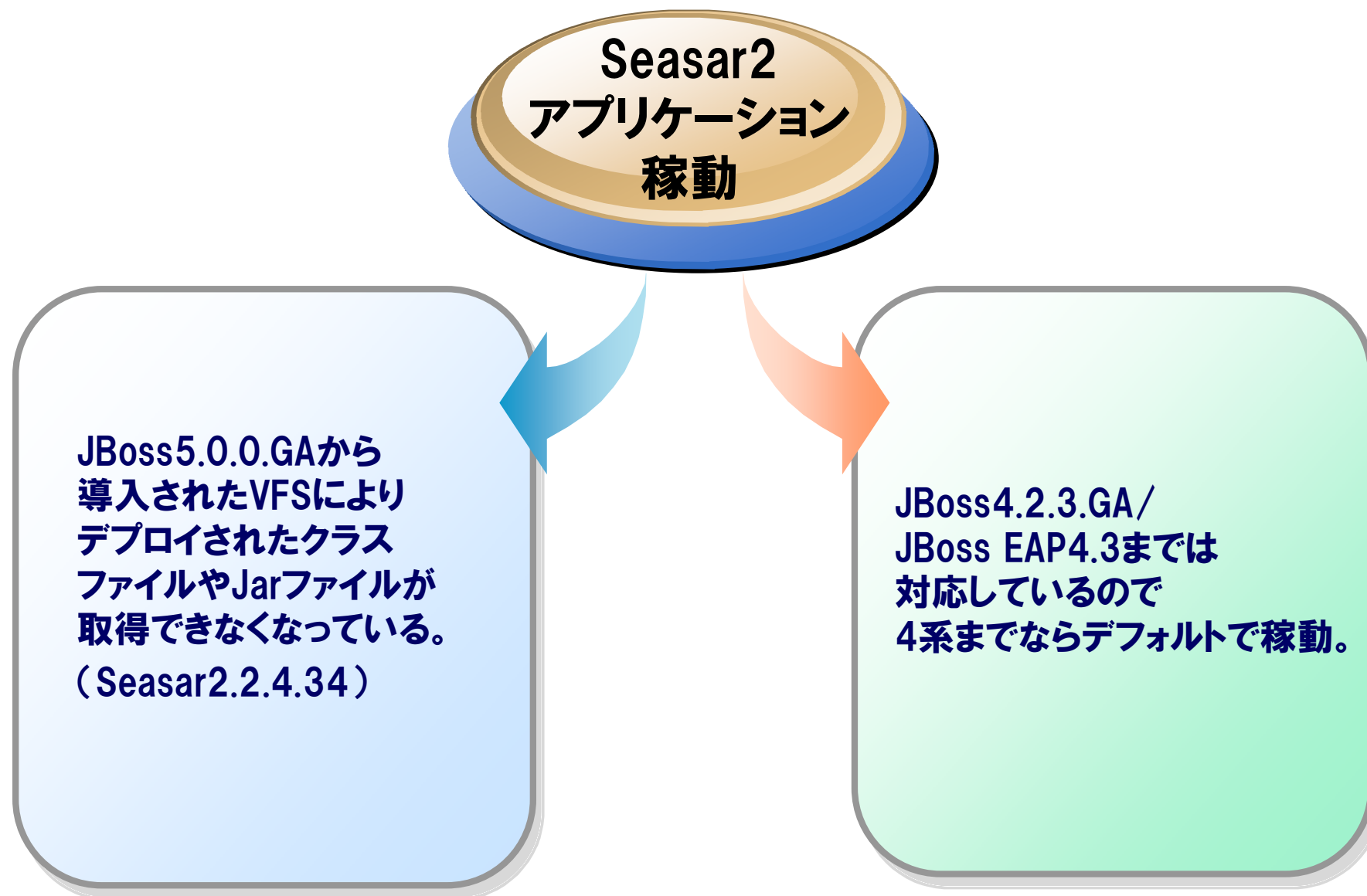
JBoss
Application
Server
5.0.X

JavaEE5準拠の機能を
内包している。

org版 JBoss Application Server 5.0.XはJavaEE5に準拠した
各種機能をアプリケーションサーバとして提供している。



Seasar2対応状況



Seasar2を
JBossで
動かすのは
JBoss4.2.Xまでで
終了のお知らせ

否

動かめなら
動かしてみせよう
JBoss5



ということで

現在、Seasarとしては正式にJBoss5の対応を進めております。特に設定もなく稼働させる対応を目指して作業を進めております。私自身も作業に参加をしています。正式リリースはもう少しお待ちください。

対応を進めて下さってる id:koichikさん ありがとうございます！！
そして、私自身がS2VFSとして対応を進めようとしていた時に
助けて下さいましたT2チームの米林さん、大谷さん、片山さんに
この場を借りてお礼を言いたいと思います。

対応のログ

[operation:2295] [申請] S2VFS
<http://ml.seasar.org/archives/operation/2009-February/004124.html>
[seasar-dev:1067] [S2Container] JBossAS5 対応
<http://ml.seasar.org/archives/seasar-dev/2009-March/001066.html>

私のブログ

Seasar2とJBoss5.0.0.GA VFSの問題
<http://bose.techie.jp/?eid=703330>
Seasar2とJBoss5 VFSの問題解決に向けて
<http://bose.techie.jp/?eid=707358>



ということで

現在、Seasarとしては正式にJBoss5の対応を

進めてまいります。特に認容メカニズム



皆さんに使って頂けるよう
当たり前の事をしただけです。
Enjoy Seasar ! Enjoy T2 !
Enjoy OSS!
By 米林

<http://bose.techie.jp/?eid=703330>

Seasar2とJBoss5 VFSの問題解決に向けて

<http://bose.techie.jp/?eid=707358>



現状の対応状況

❖ JBoss5.0.0.GA

- war形式に関してWindowsで動作確認済
 - ・ 展開したwar 及び warファイルのまま 両方動作確認済
 - ・ 各種環境において来週検証予定
 - － 検証環境例
 - » 64bit RHEL OpenJDK,Sun JDK
 - » 64bit SUSE Enterprise Linux IBM JDK,Sun JDK

❖ JBoss5.0.1.GA

- 展開war形式に関してWindowsで動作確認済み
 - ・ Warファイルのままは動かない状態 JBoss5.0.2.GAの動向を見極めて対応する予定



JBoss5.0.0.GAで開発中の対応版を動かして見よう

1

ダウンロードでJBossゲット

2

JBossフォルダ構成理解

3

SAStrutsを動かしてみよう！



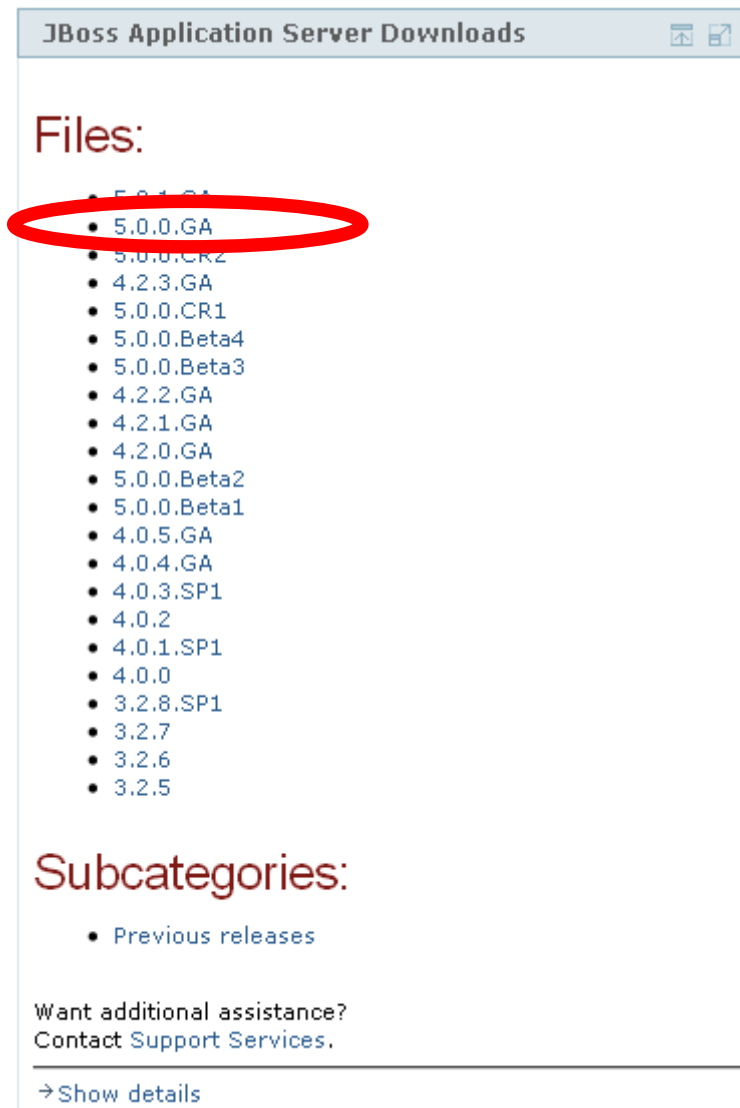
ダウンロード

<http://www.jboss.org> を開いて
Projects → Servers → JBoss Application Server を選択

The screenshot shows the JBoss Community website. The top navigation bar includes links for Home, Members, Projects, and Products. The Projects dropdown menu is open, showing a list of categories: Project Matrix, Web Interface, Programming Model, Services, Servers, Management, Tools, Other, and Archive. The Servers category is further expanded, showing a list of servers: Mobicents, Teiid, JBoss Microcontainer, JBoss Web, and JBoss Application Server. The JBoss Application Server link is circled in red. Below the navigation bar, there is a large 'Welcome to the community' banner. To the left of the banner, there is an announcement: 'ANNOUNCEMENT: JBoss Community gets a new look'. Below the banner, there is a large graphic for JBoss Application Server 5.0.0 GA, stating 'Over 100,000 Downloads! Released to the community.' To the right of the banner, there are two smaller promotional banners: 'A Look Inside JBoss Rails & Cloud with Bob McWhirter Check it out on DZone' and 'Management for Java Middleware projects just got a lot easier. Jopr'.



ダウンロード



左図の5.0.0.GAを
クリックする！



ダウンロード

使用するJDKが6ならjboss-5.0.0.GA-jdk6.zip、
JDKが5ならjboss-5.0.0.GA.zip をクリックして
ダウンロードする。解凍後、任意の場所に設置する。

SOURCEFORGE.NET [Log in](#) [Create account](#)

[Summary](#) [Tracker](#) [Mailing Lists](#) [Services](#) [Download](#) [Documentation](#) [Tas](#)

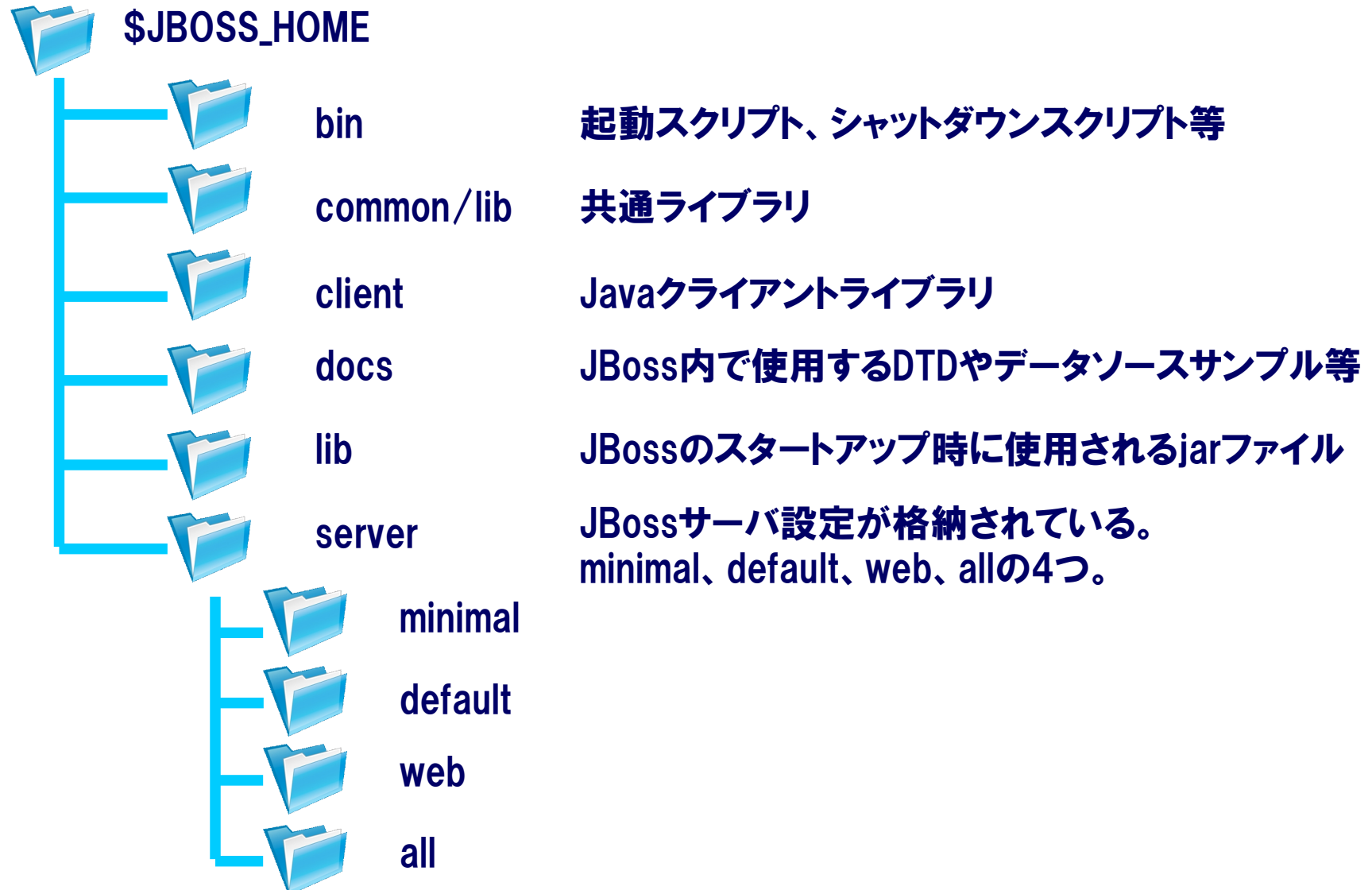
You have selected the JBoss-5.0.0.GA release.
Please choose the file that best matches your architecture or operating system from the list of files contained in this release.

[Options](#)

Package	Release	Filename	Size	Architecture
JBoss				
Latest	JBoss-5.0.0.GA (2008-12-05 05:07)			
		jboss-5.0.0.GA-jdk6.zip	108961947	Platform-Independent
		jboss-5.0.0.GA-jdk6.zip.MD5	32	Platform-Independent
		jboss-5.0.0.GA-jdk6.zip.SHA-256	64	Platform-Independent
		jboss-5.0.0.GA-src.tar.gz	138107113	Platform-Independent
		jboss-5.0.0.GA-src.tar.gz.MD5	32	Platform-Independent
		jboss-5.0.0.GA-src.tar.gz.SHA-256	64	Platform-Independent
		jboss-5.0.0.GA.zip	108810087	Platform-Independent
		jboss-5.0.0.GA.zip.MD5	32	Platform-Independent
		jboss-5.0.0.GA.zip.SHA-256	64	Platform-Independent
Totals:	1	9	355879435	



JBoss5.0.0.GAフォルダ構成理解





サーバ設定概要

minimal

JBossを稼動させる最小限の設定。
マイクロカーネルを起動させてログイン、
JNDIサーバ、URLデプロイメントスキャナーが起動。

default

JavaEEアプリケーションをデプロイする
標準の基本JavaEE5構成

all

クラスタリング、IIOP等を含む
完全JavaEE5構成

Web

JavaEE6 Webプロファイル相当
Webコンテナを軸とした軽量設定



SAStrutsを動かしてみよう！

**今回はdefaultのサーバ設定を利用して
\$JBOSS_HOME/server/default 上で、
SAStruts+S2JDBCを動かしてみる！**



Hudsonから最新のビルドを手に入れる

- ❖ <https://www.seasar.org/hudson/job/s2container-2.4-seasar2/ws/seasar2/s2-framework/target/>
から **s2-framework-2.4.35-SNAPSHOT.jar** を入手
- ❖ <https://www.seasar.org/hudson/job/s2container-2.4-seasar2/ws/seasar2/s2-extension/target/>
から **s2-extension-2.4.35-SNAPSHOT.jar** を入手
- ❖ <https://www.seasar.org/hudson/job/s2container-2.4-s2tiger/ws/s2-tiger/target/>
から **s2-tiger-2.4.35-SNAPSHOT.jar** を入手



SAStrutsのPJのjarの入替

- ❖ Hudsonから手に入れた最新のjarファイルを既存のアプリケーションのjarファイルと差し替える。

**これだけで
JBoss5で動かす準備が
出来ちゃいます。
基本的には特別な設定をしなくても
いいように対応中です。**



J2EEサーバ上のSeasar2 データソース&JTA

でも、jar差し替えて動かすだけではTomcatのようなJTAの実装の無い環境の為にS2JTA & 独自のデータソースをSeasarが実装して用意したものを使ってる状態です。

今回はJ2EEサーバとしてデータソースもJTAも備えているJBossを使用するので、Seasarのデータソース&JTAではなくてJBossのデータソース&JTAを使う設定を紹介します。

参考URL:

<http://s2container.seasar.org/2.4/ja/transaction.html>

<http://d.hatena.ne.jp/koichik/20081204#1228395633>

※S2JTAは厳密な2PCじゃないので厳密な2PCが必要な方はJ2EEサーバのJTA使用を要検討です。大抵のケースはS2JTAで問題はないですが。。

※今回の設定はJDBC2層ドライバを設定しないので2PCの設定ではありません。通常のJDBCドライバの設定になります。



J2EEサーバ上のSeasar2 データソース&JTA

でも、jar差し替えて動かすだけではTomcatのようなJTAの実装の無い環境の為にS2JTA & 独自のデータソースをSeasarが実装して

用

今

使

J

考

h

h

個人的な感想としては、
独自にJTAを実装している
Seasarはすごい
と思います。

※S2JTAは厳密な2PCじゃないので厳密な2PCが必要な方はJ2EEサーバのJTA使用を要検討です。大抵のケースはS2JTAで問題はないですが。。

※今回の設定はJDBC2層ドライバを設定しないので2PCの設定ではありません。
通常のJDBCドライバの設定になります。



JBossでは必要ないjarファイルを除く

❖絶対に抜いておくファイル

- WEB-INF/lib内のgeronimo*.jarは全て除く。
 - JTA等の部分でJBossのライブラリとぶつかりますので絶対にはずして下さい。

**これしないとS2JTAのjarが
先に読み込まれて
JBossのライブラリが
読まれません ><**



環境構築 データソース設定

JBossサーバのデータソース機能を利用するには、
*-ds.xmlという命名規則でファイル名を
設定してdeployフォルダに設置する。



結果、JNDIに登録されてDataSourceが取得可能に！
APサーバのデータソース&JTAが使用可能に！！

DB2、SQL Server、Oracle、MySQL、
PostgreSQL等のサンプルファイルが
\$JBOSS_HOME/docs/examples/
jca以下に存在する。

このファイルをカスタマイズして
設定するのが簡単な方法。

RDBMSのJDBCドライバは
\$JBOSS_HOME/server/default/
libに置く。



PostgreSQLのデータソース設定例

```
<?xml version="1.0" encoding="UTF-8"?>
<datasources>
  <local-tx-datasource>
    <jndi-name>PostgreSQLDS</jndi-name>
    <connection-url>
      jdbc:postgresql://[servername]:[port]/[database name]
    </connection-url>
    <driver-class>org.postgresql.Driver</driver-class>
    <user-name>XXXX</user-name>
    <password>XXXX</password>
    <new-connection-sql>select 1</new-connection-sql>
    <check-valid-connection-sql>select 1</check-valid-connection-sql>
    <metadata>
      <type-mapping>PostgreSQL 8.0</type-mapping>
    </metadata>
  </local-tx-datasource>
</datasources>
```

JNDIに登録する名前
※java:JNDIに登録する名前で
取得できるようになる

JDBC接続情報

新しい接続をする時に使用するSQL

新しい接続をする時に
使用するSQL

PostgreSQL、PostgreSQL 7.2、PostgreSQL 8.0から選択
\$JBoss_HOME/server/JBossCONF/conf/
standardjbosscmp-jdbc.xmlの定義を参照



環境構築 S2JDBC設定

app,diconにs2jdbc.dicon
ファイルをインクルードする

jta-jboss42.diconを用意

s2container.diconにてjta.diconと
jta-jboss42.diconの差し替えを設定

jdbc.diconファイルを
JNDIを利用して
データソースを
取得するように設定

s2jdbc.diconファイルを
使用するRDBMSに
あわせて設定



app.dicon例

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE components PUBLIC "-//SEASAR//DTD S2Container 2.4//EN"
"http://www.seasar.org/dtd/components24.dtd">
<components>
  <include path="convention.dicon"/>
  <include path="aop.dicon"/>
  <include path="j2ee.dicon"/>
  <include path="s2jdbc.dicon"/>
  <component name="actionMessagesThrowsInterceptor"
    class="org.seasar.struts.interceptor.ActionMessagesThrowsInterceptor"/>
</components>
```



jta-jboss42.dicon 例

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE components PUBLIC "-//SEASAR//DTD S2Container 2.4//EN"
"http://www.seasar.org/dtd/components24.dtd">
<components namespace="jta">
  <component name="UserTransaction" class="javax.transaction.UserTransaction">
    @org.seasar.extension.j2ee.JndiResourceLocator@lookup ("java:comp/UserTransaction")
  </component>

  <component name="TransactionManager" class="javax.transaction.TransactionManager">
    @org.seasar.extension.j2ee.JndiResourceLocator@lookup ("java:/TransactionManager")
  </component>

  <component name="TransactionSynchronizationRegistry"
    class="javax.transaction.TransactionSynchronizationRegistry">
    @org.seasar.extension.j2ee.JndiResourceLocator@lookup ("java:comp/TransactionSynchronizationRegistry")
  </component>

  <component class="org.seasar.extension.tx.adapter.JTATransactionManagerAdapter"/>
</components>
```

Seasarのサイトに記述されているものと同じでOK
<http://s2container.seasar.org/2.4/ja/transaction.html>



s2container.dicon 例

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE components PUBLIC "-//SEASAR//DTD S2Container 2.4//EN"
"http://www.seasar.org/dtd/components24.dtd">
<components>
  <include condition="#ENV == 'ut'" path="warmdeploy.dicon"/>
  <include condition="#ENV == 'ct'" path="hotdeploy.dicon"/>
  <include condition="#ENV != 'ut' and #ENV != 'ct'" path="cooldeploy.dicon"/>

  <component class="org.seasar.framework.container.factory.SimplePathResolver">
    <initMethod name="addRealPath">
      <arg>"jta.dicon"</arg>
      <arg>"jta-jboss42.dicon"</arg>
    </initMethod>
  </component>

</components>
```

JBoss用のJTAの設定ファイル
この部分を実ファイル名にする事でS2JTAと切替になる



jdbc.dicon 例

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE components PUBLIC "-//SEASAR2.1//DTD S2Container//EN"
    "http://www.seasar.org/dtd/components21.dtd">
<components namespace="jdbc">
  <include path="jta.dicon"/>

  <!-- from JNDI -->
  <component name="DataSource" class="javax.sql.DataSource">
    @org.seasar.extension.j2ee.JndiResourceLocator@lookup ("java:/PostgreSQLDS")
  </component>

</components>
```

JNDIに登録した名前
※java:JNDIに登録する名前を取得



s2jdbc.dicon 例

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE components PUBLIC "-//SEASAR//DTD S2Container 2.4//EN"
    "http://www.seasar.org/dtd/components24.dtd">
<components>
  <include path="jdbc.dicon"/>
  <include path="s2jdbc-internal.dicon"/>
  <component name="jdbcManager"
    class="org.seasar.extension.jdbc.manager.JdbcManagerImpl">
    <property name="maxRows">0</property>
    <property name="fetchSize">0</property>
    <property name="queryTimeout">0</property>
    <property name="dialect">postgreDialect</property>
  </component>
</components>
```

JNDIに登録したRDBMSに
あうものを記述する



SAStrutsを動かしてみよう！ 環境構築 デプロイ

あとは\$JBOSS_HOME/server/default/deploy
フォルダにデータソース設定ファイルと
出来上がったwarをおくだけ、
更新時も上書きするだけ！

**JavaEE APサーバにありがちな
複雑なDeploy手順はありません！！**



SAStrutsを動かしてみよう！ 環境構築 起動

❖ **\$JBOSS_HOME/bin/run.batを起動**

※ **\$JBOSS_HOME/bin/run.bat -b xxx.xxx.xxx.xxx**
とする事で任意のIPアドレスで起動が可能
デフォルトではlocalhostにバインドされて起動する

**他にもオプションがありますが、
詳細なものは別の機会に！！**



Appendix

1

TeedaをJBossで動かす際の注意点

2

Apacheとの連携 (mod_jk) について

3

mod_jkのコンパイル方法について



TeedaをJBossで動かす際の注意点



Teedaを動かす際の注意点

- ❖ Teedaで作られたアプリケーションをデプロイする時には既にJBossが内包しているJSFとコンフリクトします。
- ❖ web.xmlに下記を追加してJBossが内包しているJSFを使わないようにします。
- ❖ JSFはJavaEEに含まれるのでJBossがJSFを持っています。

```
<context-param>  
  <param-name>org.jboss.jbossfaces.WAR_BUNDLES_JSF_IMPL</param-name>  
  <param-value>true</param-value>  
</context-param>
```



Apacheとの連携 (mod_jk) について



Apache連携設定ファイル例

❖ /etc/httpd/conf.d/mod_jk.conf

```
LoadModule jk_module modules/mod_jk.so  
JkWorkersFile conf.d/workers.properties  
JkLogFile /var/log/www/mod_jk.log  
JkLogLevel warn  
JkMount /sample/* blancer
```

mod_jk設定ファイル

mod_jkログファイル

mod_jkログ出力レベル

mod_jk連携先設定



Apache連携設定ファイル例

❖ workers.properties

```
worker.list=blancer  
worker.blancer.type=lb  
worker.blancer.balance_workers=jboss01,jboss02  
worker.blancer.sticky_session=True  
worker.blancer.method=Session
```

StickySessionを有効にし、
Session毎にアクセスを振り分ける。

```
worker.list=jboss01  
worker.jboss01.port=8009  
worker.jboss01.host=192.168.0.2  
worker.jboss01.type=ajp13  
worker.jboss01.route=jboss01
```

JBossアプリケーションサーバの
IPアドレス、ポート、
連携プロトコル ajp13を指定

```
worker.list=jboss02  
worker.jboss02.port=8009  
worker.jboss02.host=192.168.0.3  
worker.jboss02.type=ajp13  
worker.jboss02.route=jboss02
```

\$JBoss_HOME/server/
\$JBossCONF/deploy/
jbossweb.sar/server.xmlの
Engine要素のjvmRouteに
設定されたユニークな名前を指定

参考URL:<http://tomcat.apache.org/connectors-doc/reference/workers.html>



mod_jkのコンパイル方法について



RHEL5 上でのmod_jkコンパイル例

❖ Apacheの下記のモジュール相当がインストールされている事

- httpd-2.2.3-11.el5_2.4.x86_64
- httpd-devel-2.2.3-11.el5_2.4.x86_64

❖ Apacheのサイトからtomcat-connectors-1.2.xx-src.tar.gzというソースをダウンロード

ダウンロードURL:<http://tomcat.apache.org/download-connectors.cgi>

❖ rootユーザ権限で下記を実行

- ```
tar zxvf tomcat-connectors-1.2.xx-src.tar.gz
cd ./tomcat-connectors-1.2.xx-src
./configure --with-apxs=/usr/sbin/apxs
make install
```





## おまけのおまけ

私のブログとか技術wikiに  
たどり着きたい場合は  
Google先生に  
**bose999** って  
聞いてみてください。



## 終了後の質問

❖ **\$JBOSS\_HOME/server/default/lib にライブラリのjarファイルを置いてますがその方法でJBoss5でS2動きますか？**

- **このパターンの検証がまだされていません。このパターンの検証も進めますので正式回答はお待ち下さい。**