



Sun Cluster 3.1 Data Service for Apache Tomcat ガイド

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 817-4846-10
2003 年 10 月, Revision A

Copyright 2003 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

本製品およびそれに関連する文書は著作権法により保護されており、その使用、複製、頒布および逆コンパイルを制限するライセンスのもとにおいて頒布されます。サン・マイクロシステムズ株式会社の書面による事前の許可なく、本製品および関連する文書のいかなる部分も、いかなる方法によっても複製することが禁じられます。

本製品の一部は、カリフォルニア大学からライセンスされている Berkeley BSD システムに基づいていることがあります。UNIX は、X/Open Company, Ltd. が独占的にライセンスしている米国ならびに他の国における登録商標です。フォント技術を含む第三者のソフトウェアは、著作権により保護されており、提供者からライセンスを受けているものです。

Federal Acquisitions: Commercial Software—Government Users Subject to Standard License Terms and Conditions.

本製品に含まれる HG-MinchoL、HG-MinchoL-Sun、HG-PMinchoL-Sun、HG-GothicB、HG-GothicB-Sun、および HG-PGothicB-Sun は、株式会社リコーがリコービイマジクス株式会社からライセンス供与されたタイプフェイスマスタをもとに作成されたものです。HeiseiMin-W3H は、株式会社リコーが財団法人日本規格協会からライセンス供与されたタイプフェイスマスタをもとに作成されたものです。フォントとして無断複製することは禁止されています。

Sun、Sun Microsystems、docs.sun.com、AnswerBook、AnswerBook2 は、米国およびその他の国における米国 Sun Microsystems, Inc. (以下、米国 Sun Microsystems 社とします) の商標もしくは登録商標です。

サンのロゴマークおよび Solaris は、米国 Sun Microsystems 社の登録商標です。

すべての SPARC 商標は、米国 SPARC International, Inc. のライセンスを受けて使用している同社の米国およびその他の国における商標または登録商標です。SPARC 商標が付いた製品は、米国 Sun Microsystems 社が開発したアーキテクチャに基づくものです。

OPENLOOK、OpenBoot、JLE は、サン・マイクロシステムズ株式会社の登録商標です。

Wnn は、京都大学、株式会社アステック、オムロン株式会社で共同開発されたソフトウェアです。

Wnn6 は、オムロン株式会社、オムロンソフトウェア株式会社で共同開発されたソフトウェアです。© Copyright OMRON Co., Ltd. 1995-2000. All Rights Reserved. © Copyright OMRON SOFTWARE Co., Ltd. 1995-2002 All Rights Reserved.

「ATOK」は、株式会社ジャストシステムの登録商標です。

「ATOK Server/ATOK12」は、株式会社ジャストシステムの著作物であり、「ATOK Server/ATOK12」にかかる著作権その他の権利は、株式会社ジャストシステムおよび各権利者に帰属します。

本製品に含まれる郵便番号辞書 (7 桁/5 桁) は郵政事業庁が公開したデータを元に制作された物です (一部データの加工を行なっています)。

本製品に含まれるフェイスマーク辞書は、株式会社ビレッジセンターの許諾のもと、同社が発行する『インターネット・パソコン通信フェイスマークガイド '98』に添付のものを使用しています。© 1997 ビレッジセンター

Unicode は、Unicode, Inc. の商標です。

本書で参照されている製品やサービスに関しては、該当する会社または組織に直接お問い合わせください。

OPEN LOOK および Sun Graphical User Interface は、米国 Sun Microsystems 社が自社のユーザおよびライセンス実施権者向けに開発しました。米国 Sun Microsystems 社は、コンピュータ産業用のビジュアルまたはグラフィカル・ユーザインタフェースの概念の研究開発における米国 Xerox 社の先駆者としての成果を認めるものです。米国 Sun Microsystems 社は米国 Xerox 社から Xerox Graphical User Interface の非独占的ライセンスを取得しており、このライセンスは米国 Sun Microsystems 社のライセンス実施権者にも適用されます。

DiComboBox ウィジェットと DtSpinBox ウィジェットのプログラムおよびドキュメントは、Interleaf, Inc. から提供されたものです。(© 1993 Interleaf, Inc.)

本書は、「現状のまま」をベースとして提供され、商品性、特定目的への適合性または第三者の権利の非侵害の黙示の保証を含みそれに限定されず、明示的であるか黙示的であるかを問わない、なんらの保証も行われぬものとします。

本製品が、外国為替および外国貿易管理法 (外為法) に定められる戦略物資等 (貨物または役務) に該当する場合、本製品を輸出または日本国外へ持ち出す際には、サン・マイクロシステムズ株式会社の事前の書面による承諾を得ることのほか、外為法および関連法規に基づく輸出手続き、また場合によっては、米国商務省または米国所轄官庁の許可を得ることが必要です。

原典: Sun Cluster 3.1 Data Service for Apache Tomcat Guide

Part No: 817-3243-10

Revision A



040413@8606



目次

Sun Cluster HA for Apache Tomcat のインストールと構成	5
Sun のオンラインマニュアル	5
表記上の規則	5
Sun Cluster HA for Apache Tomcat のインストールと構成	7
Sun Cluster HA for Apache Tomcat	8
Sun Cluster HA for Apache Tomcat のインストールと構成の計画	8
水平のスケーラビリティ	8
構成に関する制限事項	10
構成に関する要件	11
Apache Tomcat のインストールと構成	14
Apache Tomcat のインストールと構成方法	14
Apache Tomcat のインストールと構成の確認	17
▼ Apache Tomcat のベースインストールと構成の確認方法	17
▼ Apache Tomcat アプリケーションの配備方法	18
Sun Cluster HA for Apache Tomcat パッケージのインストール	19
Web Start プログラムを使用して Sun Cluster HA for Apache Tomcat パッケージをインストールする方法	20
▼ scinstall ユーティリティを使用して Sun Cluster HA for Apache Tomcat パッケージをインストールする方法	21
Sun Cluster HA for Apache Tomcat の登録と構成	22
▼ Sun Cluster HA for Apache Tomcat をフェイルオーバーデータサービスとして登録して構成する方法	22
▼ Sun Cluster HA for Apache Tomcat をスケーラブルデータサービスまたはマルチマスターデータサービスとして登録して構成する方法	25
Sun Cluster HA for Apache Tomcat のインストールと構成の確認	29
▼ Sun Cluster HA for Apache Tomcat のインストールと構成を確認する方法	30

Sun Cluster Apache Tomcat HA パラメタファイルの概要	30
Apache Tomcat HA パラメタファイルの構造	30
TestCmd 変数と ReturnString 変数の選択方法	32
Sun Cluster HA for Apache Tomcat の障害モニターの概要	32
リソースプロパティ	33
検証アルゴリズムと機能	33
Sun Cluster HA for Apache Tomcat のデバッグ方法	34
▼ Sun Cluster HA for Apache Tomcat のデバッグを有効にする方法	34
索引	37

Sun Cluster HA for Apache Tomcat のインストールと構成

Sun のオンラインマニュアル

docs.sun.com では、Sun が提供しているオンラインマニュアルを参照することができます。マニュアルのタイトルや特定の主題などをキーワードとして、検索を行うこともできます。URL は、<http://docs.sun.com> です。

表記上の規則

このマニュアルでは、次のような字体や記号を特別な意味を持つものとして使用します。

表 1-1 表記上の規則

字体または記号	意味	例
AaBbCc123	コマンド名、ファイル名、ディレクトリ名、画面上のコンピュータ出力、コード例を示します。	.login ファイルを編集します。 ls -a を使用してすべてのファイルを表示します。 system%
AaBbCc123	ユーザーが入力する文字を、画面上のコンピュータ出力と区別して示します。	system% su password:
<i>AaBbCc123</i>	変数を示します。実際に使用する特定の名前または値で置き換えます。	ファイルを削除するには、rm <i>filename</i> と入力します。

表 1-1 表記上の規則 (続き)

字体または記号	意味	例
『』	参照する書名を示します。	『コードマネージャ・ユーザズガイド』を参照してください。
「」	参照する章、節、ボタンやメニュー名、強調する単語を示します。	第 5 章「衝突の回避」を参照してください。 この操作ができるのは、「スーパーユーザー」だけです。
\	枠で囲まれたコード例で、テキストがページ行幅を超える場合に、継続を示します。	sun% grep `^#define \ XV_VERSION_STRING`

コード例は次のように表示されます。

■ C シェル

```
machine_name% command y|n [filename]
```

■ C シェルのスーパーユーザー

```
machine_name# command y|n [filename]
```

■ Bourne シェルおよび Korn シェル

```
$ command y|n [filename]
```

■ Bourne シェルおよび Korn シェルのスーパーユーザー

```
# command y|n [filename]
```

[] は省略可能な項目を示します。上記の例は、*filename* は省略してもよいことを示しています。

| は区切り文字 (セパレータ) です。この文字で分割されている引数のうち 1 つだけを指定します。

キーボードのキー名は英文で、頭文字を大文字で示します (例: Shift キーを押します)。ただし、キーボードによっては Enter キーが Return キーの動作をします。

ダッシュ (-) は 2 つのキーを同時に押すことを示します。たとえば、Ctrl-D は Control キーを押したまま D キーを押すことを意味します。

Sun Cluster HA for Apache Tomcat のインストールと構成

表 1-2 に、Sun Cluster HA for Apache Tomcat のインストールと構成に必要な作業を示します。指定された順番どおりに、各作業を行なってください。

表 1-2 作業マップ: Sun Cluster HA for Apache Tomcat のインストールと構成

作業	参照先
インストールを計画する	8 ページの「Sun Cluster HA for Apache Tomcat のインストールと構成の計画」
Sun Cluster HA for Apache Tomcat のインストールと構成を行う	14 ページの「Apache Tomcat のインストールと構成方法」
インストールと構成を確認する	17 ページの「Apache Tomcat のベースインストールと構成の確認方法」
Sun Cluster HA for Apache Tomcat パッケージをインストールする	19 ページの「Sun Cluster HA for Apache Tomcat パッケージのインストール」
Sun Cluster HA for Apache Tomcat をフェイルオーバーデータサービスとして登録し、構成する	22 ページの「Sun Cluster HA for Apache Tomcat をフェイルオーバーデータサービスとして登録して構成する方法」
Sun Cluster HA for Apache Tomcat をスケラブルデータサービスとして登録し、構成する	25 ページの「Sun Cluster HA for Apache Tomcat をスケラブルデータサービスまたはマルチマスターデータサービスとして登録して構成する方法」
Sun Cluster HA for Apache Tomcat のインストールと構成を確認する	30 ページの「Sun Cluster HA for Apache Tomcat のインストールと構成を確認する方法」
Sun Cluster Apache Tomcat HA パラメタファイルについて理解する	30 ページの「Sun Cluster Apache Tomcat HA パラメタファイルの概要」
Sun Cluster HA for Apache Tomcat の障害モニターについて理解する	32 ページの「Sun Cluster HA for Apache Tomcat の障害モニターの概要」
Sun Cluster HA for Apache Tomcat のデバッグ方法	34 ページの「Sun Cluster HA for Apache Tomcat のデバッグを有効にする方法」

Sun Cluster HA for Apache Tomcat

Apache Tomcat は、Apache Web サーバーの背後でサーブレットエンジンとして、あるいはサーブレットエンジンを含むスタンドアロンの Web サーバーとして構成できます。

Apache Tomcat は Apache Software License のもとで無料配布されており、<http://jakarta.apache.org> からダウンロードできます。

Sun Cluster HA for Apache Tomcat データサービスは、Apache Tomcat サービスの順序正しい起動と停止、障害監視、自動フェイルオーバーなどのメカニズムを提供するものです。

表 1-3 コンポーネントの保護

コンポーネント	保護するデータサービス
Apache Tomcat	Sun Cluster HA for Apache Tomcat

Sun Cluster HA for Apache Tomcat のインストールと構成の計画

この節では、Sun Cluster HA for Apache Tomcat のインストールと構成の計画に必要な情報を示します。

水平のスケーラビリティ

Sun Cluster には、スケーラブルサービスと呼ばれるデータサービスを対象とした水平のスケーラビリティという概念があります。この概念には、IP ベースの負荷均衡アルゴリズムが含まれます。このため、ユーザーはハードウェアのロードバランサーを使用することなく水平に拡張できます。このスケーラブルサービスの詳細は、『*Sun Cluster の概念*』を参照してください。

スケーラブル構成で Sun Cluster HA for Apache Tomcat を使用する場合は、あらかじめクラスタとクライアントの設備をよく確認する必要があります。

クライアントがプロキシを介してアプリケーションにアクセスする場合は、使用されているプロキシがセッションコンテキストにおいて同一のままであることを確認する必要があります。これはイントラネットでは一般に保証されますが、確認の調査が必要です。

プロキシがセッションコンテキストで変化する場合は、それは負荷均衡の観点から見ると、ソース IP アドレスが変化することを意味します。この場合、（ハードウェア負荷均衡であるかソフトウェア負荷均衡であるかにかかわらず）IP ベースの負荷均衡が失敗します。

特に、アプリケーションがインターネットで使用されている場合、セッションコンテキストにおいて同一のソース IP アドレスが維持されることは保証されません。

この問題を解決して Apache Tomcat から水平方向のスケーラビリティを得るオプションはいくつかあります。

Sun Cluster HA for Apache Tomcat には、水平方向のスケーラビリティを得る手段として 2 つの構成方法が用意されています。

1. Sun Cluster HA for Apache Tomcat を純粋なスケーラブル構成で使用する。
2. Sun Cluster HA for Apache Tomcat をマルチマスター構成で使用する。

マルチマスター構成とスケーラブル構成は、どちらもスケーラブルデータサービスとして構成されます。この 2 つの構成の違いは、ノードのアクセス方法にあります。

- クライアントは、共有アドレスを使用してスケーラブル構成にアクセスします。この場合、Sun Cluster は負荷均衡を行います。
- クライアントは、各ノードの物理アドレスを介してマルチマスター構成にアクセスします。負荷均衡は、どこか別の場所で行う必要があります。

スケーラブルデータサービスの詳細は、『Sun Cluster の概念』を参照してください。

- スケーラブル構成としての **Sun Cluster HA for Apache Tomcat**

同じソース IP アドレスが（たとえばイントラネットなどで）セッションコンテキストにおいて維持されることが保証される場合は、Sun Cluster HA for Apache Tomcat をスケーラブル構成で使用できます。

セッションコンテキストにおいてソース IP アドレスが変化する可能性があるがスケーラブル構成が望ましいという場合は、セッションを複製して Apache Tomcat を構成する必要があります。このような構成は、広域ファイルシステムまたはデータベースを使用したアプリケーションによって行えます。



注意 - この方法では、パフォーマンスの低下が発生します。

この場合、Tomcats インバウンドメモリーセッション複製を使用するほうが高いパフォーマンスを得られます。

- Apache Tomcat 4.x におけるこの処理は、<http://www.theserverside.com/resources/article.jsp?l=Tomcat> の説明に従って行なってください。
- Apache Tomcat 5.x には、インバウンドメモリーセッション複製が組み込み機能として含まれています。

- Apache Tomcat 3.x の場合、インバウンドメモリーセッション複製は設定できません。
- マルチマスター構成としての **Sun Cluster HA for Apache Tomcat**

上記以外の状況では、Sun Cluster HA for Apache Tomcat をマルチマスター構成で使用できます。外部的な負荷均衡が必要であることに注意してください。通常の構成では、スケーラブル Apache Web サーバーをロードバランサーとして使用し、Apache Web サーバーの各インスタンスの背後で Apache Tomcat インスタンスの物理的なホスト名を 1 つ構成します。また、セッションコンテキストを処理する必要があるハードウェアロードバランサーを使用することもできます。

スケーラブル構成とマルチマスター構成の違いは、クライアントがクラスターノードにアクセスする方法だけです。スケーラブル構成ではクライアントは共有アドレスにアクセスしますが、マルチマスター構成ではクライアントは物理的なホスト名にアクセスします。このため、Sun Cluster の観点からはこれらの 2 つの構成の違いはありません。

構成に関する制限事項

Sun Cluster HA for Apache Tomcat のインストールと構成の計画を立てる際には、この節に示された制限事項を考慮してください。この節では、Sun Cluster HA for Apache Tomcat だけに適用されるソフトウェアとハードウェア上の構成制限の一覧を示します。

すべてのデータサービスに適用される制限事項については、『*Sun Cluster* ご使用にあたって』を参照してください。

- **Sun Cluster HA for Apache Tomcat** によるスケーラブル構成は、セッション複製または信頼性のあるソース IP アドレスを達成できる場合だけ採用してください。これ以外の状況でこの構成を採用すると、アプリケーションの動作に予期しない結果が生じることがあります。
- 信頼できるソース IP アドレスを使用するスケーラブル構成で Sun Cluster HA for Apache Tomcat を導入する場合は、リソースパラメタ **Load_balancing_policy** を必ず **LB_STICKY** に設定する必要があります。特に、セッション複製がまったく使用されない場合にはこの設定が必要です。この制限が守られないと、アプリケーションの動作に予期しない結果が生じることがあります。その他のすべてのスケーラブル構成では、キャッシュを有効利用できます。



注意 - これらの制限事項を守らないと、データサービスの構成がサポートされない場合があります。

構成に関する要件

ここで示す要件を参考にして、Apache Tomcat のインストールと構成の計画を行なってください。これらの要件が適用されるのは、Sun Cluster HA for Apache Tomcat だけです。Sun Cluster HA for Apache Tomcat のインストールと構成を進めるには、あらかじめ次の要件を満たす必要があります。



注意 - 次の要件を満たさないと、データサービスの構成がサポートされない場合があります。

- スケーラブル構成またはマルチマスター構成のために **Tomcat Home** ディレクトリをローカル記憶装置上に構成する - Apache Tomcat をスケーラブルリソースグループとしてインストールする場合は、Tomcat Home ディレクトリとその動的なデータをローカル記憶装置上に作成する必要があります。

この作業を行うのは、Apache Tomcat がディレクトリ構造を利用してその構成、ログ、配備されたアプリケーションなどを保存するためです。

ただし、ローカル記憶装置にゆとりがない場合は、共有記憶装置上の広域ファイルシステムを使用することもできます。



注意 - この場合、Tomcat アプリケーションの配備は Apache Tomcat をホスティングするノードごとに行う必要があります。

- **Sun Cluster HA for Apache Tomcat** マルチマスター構成の配備 — Apache Tomcat をマルチマスター構成で配備する場合は、前述のロードバランサーが必要です。
- フェイルオーバー構成のために共有記憶装置上に **Apache Tomcat Home** ディレクトリを構成する - Apache Tomcat をフェイルオーバーリソースグループとしてインストールする場合は、Tomcat Home ディレクトリを共有記憶装置上に作成する必要があります。Tomcat Home ディレクトリは、HAStoragePlus リソースと共に広域ファイルシステム (GFS) に置くことも、フェイルオーバーファイルシステム (FFS) に置くこともできます。望ましいのは FFS 上への配置です。

この作業を行うのは、Apache Tomcat がディレクトリ構造を利用してその構成、ログ、配備されたアプリケーションなどを保存するためです。バイナリをローカル記憶装置に保存して動的なデータ部分を共有記憶装置に保存することも可能ですが、このような方法はお勧めできません。

注 - 広域ファイルシステムをマウントする場合は /global という接頭辞を指定し、フェイルオーバーファイルシステムをマウントする場合は /local という接頭辞を指定するようにしてください。必須ではありませんが、こうするのが最も良い方法とされています。

- **Apache Tomcat** コンポーネントとその依存性 — Sun Cluster HA for Apache Tomcat データサービスは、1 つ以上の Apache Tomcat インスタンスを保護するように設定できます。各インスタンスは、1 つの Apache Tomcat リソースによって対応される必要があります。次に、Apache Tomcat リソースとほかの必須リソース間の依存関係の概要を示します。

表 1-4 フェイルオーバー構成用の Apache Tomcat コンポーネントとその依存関係 (→ は依存先を示す)

コンポーネント	説明
記憶装置リソース	このリソースは、SUNW.HAStoragePlus または SUNW.HAStorage リソースタイプのどちらかです。
Apache Tomcat (必須)	→ SUNW.HAStoragePlus / SUNW.HAStorage リソース 記憶装置リソースには、Apache Tomcat ファイルシステムのマウントポイントを管理し、マウントされるまで Apache Tomcat が起動しないようにする役割があります。
論理ホスト (必須)	Apache Tomcat は一般的なデータサービスをベースとしているため、このコンポーネントはリソースグループの論理ホストに暗黙に依存します。したがって、論理ホストが存在する必要がありますが、明示的な依存は不要です。

各 Apache Tomcat とそのパラメタファイル pfile がルートファイルシステムに保存されているかぎり、スケーラブル構成の特殊な要件のために記憶装置またはアドレスに依存する必要はありません。この条件が満たされない場合は、次の表の内容に従ってください。

表 1-5 スケーラブル構成またはマルチマスター構成用の Apache Tomcat コンポーネントとその依存関係 (→ は依存先を示す)

コンポーネント	説明
記憶装置リソース	このリソースは、SUNW.HAStorage または SUNW.HAStoragePlus リソースタイプのどちらかです。
Apache Tomcat (必須)	→ 記憶装置リソース 記憶装置リソースには、Apache Tomcat ファイルシステムのマウントポイントを管理し、マウントされるまで Apache Tomcat が起動しないようにする役割があります。
共有アドレス (必須)	共有アドレスリソースは、スケーラブルリソースグループが依存する個別のリソースグループ内に置かれます。

注 – Apache Tomcat の詳細は、jakarta.apache.org Web ページを参照してください。

例 1-1 Sun Cluster 用の Apache Tomcat 構成 / 登録ファイル (続き)

```
SCALABLE=false
PFILE=
HAS_RS=
:::::::::::::
util/sctomcat_register
:::::::::::::
#
# Copyright 2003 Sun Microsystems, Inc. All rights reserved.
# Use is subject to license terms.
#

. `dirname $0`/sctomcat_config

scrgadm -a -j $RS -g $RG -t SUNW.gds \
-x Start_command="/opt/SUNWsctomcat/bin/start_sctomcat \
-R $RS -G $RG -N $PFILE " \
-x Stop_command="/opt/SUNWsctomcat/bin/stop_sctomcat \
-R $RS -G $RG -N $PFILE " \
-x Probe_command="/opt/SUNWsctomcat/bin/probe_sctomcat \
-R $RS -G $RG -N $PFILE " \
-y Port_list=$PORT/tcp -y Network_resources_used=$LH \
-x Stop_signal=9 \
-y Scalable=$SCALABLE \
-y Resource_dependencies=$HAS_RS
```

Apache Tomcat のインストールと構成

この節では、Apache Tomcat のインストールと構成に必要な手順について説明します。

Apache Tomcat のインストールと構成方法

Apache Tomcat のインストールと構成は、次の手順で行なってください。

1. **Sun Cluster** での **Apache Tomcat** の配備方法の決定 – Apache Tomcat をどのように配備するかを決定する必要があります。
 - Apache Tomcat をフェイルオーバーデータサービスとスケーラブルデータサービスのどちらとして使用するかを決定します。
スケーラブルデータサービスとフェイルオーバーデータサービスの概念については、『*Sun Cluster の概念*』を参照してください。
 - どのユーザー名で Apache Tomcat を実行するかを決定します。
 - 配備する Apache Tomcat のバージョンとインスタンスの数を決定します。

- 1つのバージョンのインスタンスを複数配備する場合は、それらがバイナリを共有するかどうかを決定します。
 - 各 Apache Tomcat インスタンスで使用するクラスタファイルシステムを決定します。
2. **Apache Tomcat** クラスタファイルシステムのマウント – Sun Cluster での Apache Tomcat の配備方法を決定したあと、クラスタファイルシステムがマウントされていることを確認する必要があります。



注意 – この場合、Tomcat アプリケーションの配備は Apache Tomcat をホスティングするノードごとに行う必要があります。

3. Apache Tomcat が root ユーザー名以外で実行される場合は、次のコマンドを使用して適切なユーザー名とグループを作成します。
- **# groupadd —g 1000 tomcat**
 - **# useradd —u 1000 —g 1000 —d /global/tomcat —s /bin/ksh tomcat**
4. Apache Tomcat をフェイルオーバーデータサービスとして配備する場合は、Sun Cluster 内の共有ファイルシステム上に Apache Tomcat をインストールします。このソフトウェアをローカルファイルシステムにインストールした場合とクラスタファイルシステム上にインストールした場合の利点と問題点については『Sun Cluster データサービスのインストールと構成』の「アプリケーションバイナリの格納先の決定」を参照してください。
- スケーラブル構成を配備する場合は、Apache Tomcat をローカルディスクにインストールします。
- root ユーザーの状態でない場合は、コマンド **# su — user name** を使用して適切なユーザー名に切り替えます。
 - Apache Tomcat をフェイルオーバーデータサービスとして配備する場合は、1台のノード上の共有記憶装置に Apache Tomcat バイナリをインストールします。Apache Tomcat をスケーラブルデータサービスとして配備する場合は、Apache Tomcat データサービスをホスティングするノードごとに Apache Tomcat バイナリをインストールします。

注 – Apache Tomcat は、<http://jakarta.apache.org/tomcat/index.html> に示されている作業指示に従ってインストールしてください。

スケーラブル構成またはマルチマスター構成のスケーラブルデータサービスとして Apache Tomcat を配備する場合は、Apache Tomcat をホスティングするノードごとに次の手順を繰り返します。

- (Apache Tomcat ユーザー名のログインシェルに応じて) korn シェルスクリプトまたは c シェルスクリプトを作成し、Apache Tomcat の環境変数を設定します。環境変数は、ユーザープロファイル内でなくシェルスクリプト内に設定する必要があります。

この方法を使用すると、複数の Apache Tomcat バージョンとインスタンスを 1 つのユーザー名でインストールして実行できます。



注意 - これらのシェルスクリプトは、Apache Tomcat データサービスをホスティングできるすべてのノードで利用できる必要があります。フェイルオーバー構成の場合は、共有記憶装置に保存してください。スケーラブル構成またはマルチマスター構成の場合は、各ノードのローカルファイルシステムに保存するか、共有記憶装置に保存してください。これらのスクリプトはすべてのノードで同一である必要があります。

例 1-2 Apache Tomcat 3.x/3.3 の環境変数を設定する korn シェルスクリプト

```
# more env.ksh
#!/usr/bin/ksh
#
# Environment for Tomcat
# (日本語)
# Tomcat の環境
#
JAVA_HOME=/usr/j2se
export JAVA_HOME
JAKARTA_HOME=/global/mnt1/jakarta-3.3
export JAKARTA_HOME
TOMCAT_HOME=$JAKARTA_HOME
export TOMCAT_HOME
```

例 1-3 Apache Tomcat 3.3/3.3 の環境変数を設定する C シェルスクリプト

```
# more env.csh
#!/usr/bin/csh
#
# Environment for Tomcat
# (日本語)
# Tomcat の環境
#
setenv JAVA_HOME /usr/j2se
setenv JAKARTA_HOME /global/mnt1/jakarta-3.3
setenv TOMCAT_HOME $JAKARTA_HOME
```

例 1-4 Apache Tomcat 4.1.x の環境変数を設定する korn シェルスクリプト

```
# more env.ksh
#!/usr/bin/ksh
#
# Environment for Tomcat
```

例 1-4 Apache Tomcat 4.1.x の環境変数を設定する Korn シェルスクリプト (続き)

```
(日本語)
# Tomcat の環境
#
JAVA_HOME=/usr/j2se
export JAVA_HOME
CATALINA_HOME=/global/mnt1/jakarta-tomcat-4.1.18
export CATALINA_HOME
```

例 1-5 Apache Tomcat 4.1.x の環境変数を設定する C シェルスクリプト

```
# more env.csh
#!/usr/bin/csh
#
# Environment for Tomcat
(日本語)
# Tomcat の環境
#
setenv JAVA_HOME /usr/j2se
setenv CATALINA_HOME /global/mnt1/jakarta-tomcat-4.1.18
```

環境変数は、バージョンと構成によって異なります。

注 – Apache Tomcat の詳細は、jakarta.apache.org Web ページを参照してください。

Apache Tomcat のインストールと構成の確認

ここでは、インストールと構成を確認する手順について説明します。

▼ Apache Tomcat のベースインストールと構成の確認方法

次の手順でインストールと構成を確認します。データサービスをまだインストールしていないため、この手順ではアプリケーションの可用性が高いかどうかを確認することはできません。

1. Apache Tomcat を起動します。

Tomcat ユーザー名 (次の例ではスーパーユーザー) に切り替え、環境スクリプトが置かれているディレクトリに移動します。次の例では、Tomcat のバージョンは 4.1.24 です。

注 - 起動コマンドと停止コマンドの出力メッセージは、バージョンによって大きく異なります。

```
# ./env.ksh
# cd $CATALINA_HOME/bin
# ./startup.sh
Using CATALINA_BASE:   /tomcat/jakarta-tomcat-4.1.24
Using CATALINA_HOME:   /tomcat/jakarta-tomcat-4.1.24
Using CATALINA_TMPDIR: /tomcat/jakarta-tomcat-4.1.24/temp
Using JAVA_HOME:       /usr/j2se
```

2. インストールをチェックします。

Web ブラウザを起動し、<http://nodename:8080> を指定してクラスタノードに接続します。デフォルトの Tomcat ホームページが表示される場合は、すべて正常です。

3. Apache Tomcat を停止します。

```
# ./shutdown.sh
Using CATALINA_BASE:   /tomcat/jakarta-tomcat-4.1.24
Using CATALINA_HOME:   /tomcat/jakarta-tomcat-4.1.24
Using CATALINA_TMPDIR: /tomcat/jakarta-tomcat-4.1.24/temp
Using JAVA_HOME:       /usr/j2se
```

▼ Apache Tomcat アプリケーションの配備方法

Apache Tomcat をフェイルオーバーリソースとして構成する場合は、次の手順を 1 台のノードで実行してください。Apache Tomcat をスケーラブル構成として配備する場合は、次の手順をノードごとに繰り返してください。



注意 - スケーラブル構成の場合は、必ずすべての Tomcat ホストで Apache Tomcat 構成を同一にする必要があります。

1. 構成ファイル **server.xml** と **Tomcat** ユーザー構成を任意に変更します。詳細は、<http://jakarta.apache.org/tomcat/index.html> を参照してください。
2. 前述の方法で **Apache Tomcat** を起動します。
3. アプリケーションを配備します。詳細は、<http://jakarta.apache.org/tomcat/index.html> を参照してください。

4. 適切な方法でアプリケーションのテストを行います。たとえば、適切なポートを使用して **Web** ブラウザをホストに接続し、アプリケーションの起動ページを選択します。
5. 前述の方法で **Apache Tomcat** を停止します。

Sun Cluster Apache Tomcat リソースの構成には、Apache Tomcat がサービスを提供するポートが必要です。

障害監視には、配備されたアプリケーションの起動ページを使用することをお勧めします。この場合、後述の TestCmd には `get /startpage` を使用できます。この方法で、Apache Tomcat と、Apache Tomcat がサービスを提供するアプリケーションを監視できます。

Sun Cluster HA for Apache Tomcat パッケージのインストール

Sun Cluster の初回のインストールで Sun Cluster HA for Apache Tomcat パッケージをインストールしなかった場合は、ここで説明する作業を行なってパッケージをインストールしてください。この作業は、Sun Cluster HA for Apache Tomcat パッケージをインストールするクラスタノードごとに行なってください。この手順を実行するには、Sun Cluster Agents CD-ROM が必要です。

複数のデータサービスを同時にインストールする場合は、『Sun Cluster 3.1 10/03 ソフトウェアのインストール』の「ソフトウェアのインストール」に示されている作業を行なってください。

Sun Cluster HA for Apache Tomcat パッケージは、次のインストールツールのどちらかを使用してインストールします。

- Web Start プログラム
- `scinstall` ユーティリティ

注 - Web Start プログラムは、Sun Cluster 3.1 Data Services 10/03 より前のリリースでは利用できません。

Web Start プログラムを使用して Sun Cluster HA for Apache Tomcat パッケージをインストールする方法

Web Start プログラムは、コマンド行インタフェース (CLI) またはグラフィカルユーザーインタフェース (GUI) を使用して実行できます。CLI と GUI での作業の内容と手順はほとんど同じです。Web Start プログラムの詳細は、`installer(1M)` のマニュアルページを参照してください。

1. **Sun Cluster HA for Apache Tomcat** パッケージをインストールするクラスタノード上で、スーパーユーザーになります。
2. (省略可能) Web Start プログラムを GUI で実行する場合は、必ず **DISPLAY** 環境変数を設定します。

3. **Sun Cluster Agents CD-ROM** を **CD-ROM** ドライブに挿入します。
ボリューム管理デーモン `vold(1M)` が実行されており、CD-ROM デバイスを管理するように構成されている場合は、デーモンによって CD-ROM が自動的に `/cdrom/scdataservices_3_1_vb` ディレクトリにマウントされます。

4. **CD-ROM** の **Sun Cluster HA for Apache Tomcat** コンポーネントディレクトリに移動します。

Sun Cluster HA for Apache Tomcat データサービスの Web Start プログラムは、このディレクトリに入っています。

```
# cd /cdrom/scdataservices_3_1_vb/components/SunCluster_HA_APACHE_TC_3.1
```

5. Web Start プログラムを起動します。

```
# ./installer
```

6. プロンプトが表示されたところで、インストールのタイプを選択します。

- C ロケールだけをインストールする場合は、「Typical」を選択します。
- ほかのロケールをインストールする場合は、「Custom」を選択します。

7. 画面上の指示に従って、ノードに **Sun Cluster HA for Apache Tomcat** パッケージをインストールします。

インストールが終了した時点で、Web Start プログラムによってインストールのサマリーが表示されます。このサマリーを使用して、インストール時に Web Start によって作成されたログを確認できます。これらのログは `/var/sadm/install/logs` ディレクトリに置かれます。

8. Web Start プログラムを終了します。

9. **CD-ROM** ドライブから **Sun Cluster Agents CD-ROM** を取り出します。

- a. **CD-ROM** が使用されないように、**CD-ROM** 上のディレクトリ以外に移動します。

- b. **CD-ROM** を取り出します。

```
# eject cdrom
```

▼ **scinstall** ユーティリティを使用して Sun Cluster HA for Apache Tomcat パッケージをインストールする方法

SC3.1 に Sun Cluster HA for Apache Tomcat パッケージをインストールするには、ここで説明する手順を使用してください。この手順を実行するには、Sun Cluster Agents CD-ROM が必要です。この作業は、Sun Cluster の初回のインストールでこのデータサービスをインストールしていないことを前提としています。

Sun Cluster の初回のインストールの一部として Sun Cluster HA for Apache Tomcat をインストールした場合は、22 ページの「Sun Cluster HA for Apache Tomcat の登録と構成」へ進んでください。

インストールしなかった場合は、ここで説明する手順を使用して Sun Cluster HA for Apache Tomcat パッケージをインストールしてください。この手順は、Sun Cluster HA for Apache Tomcat を実行するすべてのノードで行なってください。

1. **Sun Cluster Agents CD-ROM** を **CD-ROM** ドライブに挿入します。
2. オプションを指定せずに、**scinstall** ユーティリティを実行します。
scinstall ユーティリティにより、ほかの情報を入力するためのプロンプトが表示されます。
3. メニューオプション「新しいデータサービスのサポートをこのクラスタノードに追加」を選択します。
scinstall ユーティリティが対話型モードで起動します。
4. **Sun Cluster Agents CD-ROM** のパス名を指定します。
このユーティリティでは、この CD は “data services cd” と表示されます。
5. メニューオプション「**q**) 完了」を選択します。
6. 「他のデータサービスも表示しますか」という質問に対して「**yes**」と入力します。
このユーティリティには、この CD は “data services cd” と表示されます。
7. インストールするデータサービスを指定します。
選択したデータサービスが scinstall ユーティリティによって示され、この選択内容の確認が求められます。
8. **scinstall** ユーティリティを終了します。
9. **CD-ROM** ドライブから **CD** を取り出します。

Sun Cluster HA for Apache Tomcat の登録と構成

この節では、Sun Cluster HA for Apache Tomcat の構成に必要な手順について説明します。

▼ Sun Cluster HA for Apache Tomcat をフェイルオーバーデータサービスとして登録して構成する方法

この手順は、Sun Cluster HA for Apache Tomcat をフェイルオーバーデータサービスとして構成する場合に行います。この手順は、このデータサービスパッケージがインストールされていることを前提としています。

Sun Cluster HA for Apache Tomcat パッケージをインストールしなかった場合は、19 19 ページの「Sun Cluster HA for Apache Tomcat パッケージのインストール」に戻ってください。

インストールした場合は、次の手順に従って Sun Cluster HA for Apache Tomcat データサービスを構成します。この手順は、Apache Tomcat のファイルシステムがフェイルオーバーファイルシステムとしてマウントされることを想定しています。

1. **Apache Tomcat** のホストとなるクラスタノードの **1** つで、スーパーユーザーになります。

2. **SUNW.gds** というリソースタイプを登録します。

```
# scrgadm -a -t SUNW.gds
```

3. **SUNW.HAStoragePlus** というリソースタイプを登録します。

```
# scrgadm -a -t SUNW.HAStoragePlus
```

4. フェイルオーバーリソースグループを作成します。

```
# scrgadm -a -g Apache-Tomcat-failover-resource-group
```

5. **Apache Tomcat** ディスク記憶装置に対応するリソースを作成します。

```
# scrgadm -a -j Apache-Tomcat-has-resource \  
> -g Apache-Tomcat-failover-resource-group \  
> -t SUNW.HAStoragePlus \  
> -x FilesystemMountPoints=Apache Tomcat- instance-mount-points
```

6. **Apache Tomcat** 論理ホスト名に対応するリソースを作成します。

```
# scrgadm -a -L -j Apache-Tomcat-lh-resource \
> -g Apache-Tomcat-failover-resource-group \
> -l Apache-Tomcat-logical-hostname
```

7. **Apache Tomcat** ディスク記憶装置と論理ホスト名のリソースが組み込まれたフェイルオーバーリソースグループを有効にします。

```
# scswitch -z -g Apache-Tomcat-failover-resource-group
```

8. **Sun Cluster HA for Apache Tomcat** で必要となるパラメタファイルを用意します。

```
# cd /opt/SUNWsctomcat/bin
# cp pfile desired place
```

パラメタファイルコメントに従って、pfile を編集します。

```
#!/usr/bin/ksh
# Set the Apache Tomcat specific environment variables which the start,
# stop and check functions will use
#
# EnvScript      Script to set runtime environment for tomcat
# User           Apache Tomcat user name
# Basepath       Absolute pathname to Tomcat Home directory i.e. 4.x CATALINA_HOME
#               or TOMCAT_HOME for 3.x
# Host           Hostname to test Apache Tomcat
# Port           Port where Apache Tomcat is configured
# TestCmd        Apache Tomcat test command
# Startwait      Sleeping $Startwait seconds after completion of the
#               start command
```

(日本語)

```
#!/usr/bin/ksh
# 起動、停止、およびチェックの各機能が使用する
# Apache Tomcat 固有の環境変数を設定する
#
# EnvScript      Tomcat の実行時環境を設定するスクリプト
# User           Apache Tomcat ユーザー名
# Basepath       Tomcat のホームディレクトリの絶対パス。つまり、4.x CATALINA_HOME
#               または TOMCAT_HOME for 3.x
# Host           Apache Tomcat をテストするホストの名前
# Port           Apache Tomcat が構成されるポート
# TestCmd        Apache Tomcat のテストコマンド
# Startwait      起動コマンドが完了したあと $Startwait 秒の間休眠状態
#               になる
```

```
EnvScript=
User=
Basepath=
Host=
Port=8080
TestCmd="get /index.jsp"
ReturnString="CATALINA"
Startwait=20
```

次に、Apache Tomcat 4.1.24 の場合の例を示します。

```
EnvScript=/global/mnt1/tomtest/env.ksh
User=tomtest
```

```
Basepath=/global/mnt1/tomtest/jakarta-tomcat-4.1.24
Host=tomcat-lh
Port=8080
TestCmd="get /index.jsp"
ReturnString="CATALINA"
Startwait=20
```

これは、デフォルトの起動ページ index.jsp を使用してスタンドアロン Web サーバーとして構成された Apache Tomcat 4.1.24 の例です。この起動ページには、文字列 CATALINA が含まれます。



注意 - パラメタファイルは、Apache Tomcat データサービスをホスティングできるすべてのノードで利用できる必要があります。フェイルオーバー構成の場合は、共有記憶装置に保存します。Apache Tomcat インスタンスのパラメタファイルは、すべてのノードで同一である必要があります。

この手順を、必要なすべての Apache Tomcat インスタンスで繰り返します。

9. 必要な Apache Tomcat コンポーネントを作成して登録します。

```
# cd /opt/SUNWscTomcat/util
# cp sctomcat_config desired place
# cp sctomcat_register desired place
```

コメントに従って sctomcat_config ファイルを編集します。

```
# これらのパラメタは、(key=value) フォームでカスタマイズできます。
#
# Copyright 2003 Sun Microsystems, Inc. All rights reserved.
# Use is subject to license terms.
#
# This file will be sourced in by sctomcat_register and the parameters
# listed below will be used.
#
# These parameters can be customized in (key=value) form
#
# RS - name of the resource for the application
# RG - name of the resource group containing RS
# PORT - name of the port number
# LH - name of the LogicalHostname SC resource
# SCALABLE - true or false for a scalable or failover resource
# PFILE - absolute pathname to the parameter file for the Tomcat resource
# HAS_RS - name of the HAStoragePlus SC resource
#           (it can be a , separated list for the dependencies)
```

(日本語)

```
# このファイルは、sctomcat_register によってソースに取り込まれます。
# パラメタは以下のものが使用されます。
#
# これらのパラメタは、(key=value) フォームを使用してカスタマイズできます。
#
# RS - アプリケーションのリソースの名前
# RG - RS を含むリソースグループの名前
# PORT - ポート番号の名前
```

```

#          LH - LogicalHostname SC リソースの名前
# SCALABLE - スケーラブルリソースかフェイルオーバーリソースかを true または false で指定
#          PFILE - Tomcat リソースのパラメタファイルの絶対パス名
#          HAS_RS - HAStoragePlus SC リソースの名前
#                  (依存関係にもとづいて個別のリストにもできる)
#

```

```

RS=
RG=
PORT=8080
LH=
SCALABLE=false
PFILE=
HAS_RS=

```

次に、Apache Tomcat の場合の例を示します。

```

RS=tomcat-res
RG=tomcat-rg
PORT=8080
LH=tomcat-lh
SCALABLE=false
PFILE=/global/mnt1/pfile
HAS_RS=tomcat-hastplus-res

```

10. `sctomcat_config` の編集後、リソースを登録する必要があります。

```
# ./sctomcat_register
```

必要なすべての Apache Tomcat インスタンスで、最後の 3 つの手順を繰り返します。

11. Apache Tomcat リソースを有効にします。

先の手順で作成したすべての Apache Tomcat リソースで次の手順を繰り返します。

```
# scstat
# scswitch -e -j Apache-Tomcat-resource
```

▼ Sun Cluster HA for Apache Tomcat をスケーラブルデータサービスまたはマルチマスターデータサービスとして登録して構成する方法

この手順は、Sun Cluster HA for Apache Tomcat をスケーラブルデータサービスまたはマルチマスターデータサービスとして構成する場合に使用します。この手順は、Sun Cluster の初回のインストール時にこのデータサービスパッケージをインストールしている場合を想定しています。

この手順は、Apache Tomcat をスケーラブル構成またはマルチマスター構成で配備するために使用します。

Sun Cluster の初回のインストールの一部として Sun Cluster HA for Apache Tomcat パッケージをインストールしなかった場合は、19 ページの「Sun Cluster HA for Apache Tomcat パッケージのインストール」に戻ってください。

インストールした場合は、次の手順に従って Sun Cluster HA for Apache Tomcat データサービスを構成します。

1. **Apache Tomcat** のホストとなるクラスタノードの **1** つで、スーパーユーザーになります。

2. **SUNW.gds** というリソースタイプを登録します。

```
# scrgadm -a -t SUNW.gds
```

3. **SharedAddress** リソースのフェイルオーバーリソースグループを作成します。

```
# scrgadm -a -g Apache-Tomcat-SharedAddress-resource-group
```

4. **SharedAddress** リソースを作成します。

```
# scrgadm -a -s -j Apache-Tomcat-SharedAddress-resource \  
> -g Apache-Tomcat-SharedAddress-resource-group \  
> -l Apache-Tomcat-SharedAddress-hostname
```

5. **SharedAddress** リソースグループを有効にします。

```
# scswitch -z -g Apache-Tomcat-SharedAddress-resource-group
```

6. **Sun Cluster HA for Apache Tomcat** で必要となるパラメタファイルを用意します。

```
# cd /opt/SUNWscTomcat/bin  
cp pfile desired place
```

ファイル内のコメントに従って、パラメタファイル `pfile` を編集します。次に例を示します。

```
#!/usr/bin/ksh  
# Set the Apache Tomcat specific environment variables which the start,  
# stop and check functions will use  
#  
# EnvScript      Script to set runtime environment for tomcat  
# User           Apache Tomcat user name  
# Basepath       Absolute pathname to tomcat Home directory ie 4.x CATALINA_HOME  
#               or TOMCAT_HOME for 3.x  
# Host           Hostname to test Apache Tomcat  
# Port           Port where Apache Tomcat is configured  
# TestCmd        Apache Tomcat test command  
# Startwait      Sleeping $Startwait seconds after completion of the  
#               start command  
(日本語)  
#!/usr/bin/ksh  
# 起動、停止、およびチェックの各機能が使用する  
# Apache Tomcat 固有の環境変数を設定する  
#  
# EnvScript      Tomcat の実行時環境を設定するスクリプト
```

```
# User          Apache Tomcat ユーザー名
# Basepath      Tomcat のホームディレクトリの絶対パス。つまり、4.x CATALINA_HOME
#              または TOMCAT_HOME for 3.x
# Host          Apache Tomcat をテストするホストの名前
# Port          Apache Tomcat が構成されるポート
# TestCmd       Apache Tomcat のテストコマンド
# Startwait     起動コマンドが完了したあと $Startwait 秒の間休眠状態
#              になる
EnvScript=
User=
Basepath=
Host=
Port=8080
TestCmd="get /index.jsp"
ReturnString="CATALINA"
Startwait=20
```

次に、Apache Tomcat 4.1.24 の場合の例を示します。

```
EnvScript=/tomcat/env.ksh
User=tomcat
Basepath=/tomcat/jakarta-tomca-4.1.24
Host=localhost
Port=8080
TestCmd="get /index.jsp"
ReturnString="CATALINA"
Startwait=20
```

これは、デフォルトの起動ページ `index.jsp` を使用してスタンドアロン Web サーバーとして構成された Apache Tomcat 4.1.24 の例です。この起動ページには、文字列 `CATALINA` が含まれます。

Host パラメタは、`pfile` の場所によって異なります。

- `pfile` が共有記憶装置に保存されている場合は、ローカルホストを使用する必要があります。
- `pfile` がローカル記憶装置に保存されている場合は、ローカルホストまたはノードホスト名のどちらかになります。

ここでは共有アドレスは使用しないでください。



注意 – パラメタファイルは、Apache Tomcat データサービスをホスティングできるすべてのノードで利用できる必要があります。スケーラブル構成またはマルチマスター構成の場合は、各ノードのローカルファイルシステムに保存するか、共有記憶装置に保存します。Apache Tomcat インスタンスのパラメタファイルは、すべてのノードで同一である必要があります。

必要なすべての Apache Tomcat インスタンスとノードで、この手順を繰り返します。

7. スケーラブルサービスのリソースグループを作成します。

```
# scrgadm -a -g Apache-Tomcat-Scalable-resource-group \
> -y Maximum primaries=2 \
> -y Desired primaries=2 \
> -y RG_dependencies=Apache-Tomcat-SharedAddress-resource-group
```

ノードを増やす必要がある場合は、Maximum_primaries と Desired_primaries を適切な値に調整してください。

8. **Apache Tomcat** ディスク記憶装置のリソースを作成します (ルートファイルシステム内に存在しない場合)。その必要がない場合は、手順 10 へ進みます。

```
# scrgadm -a -j Apache-Tomcat-has-resource \
> -g Apache-Tomcat-failover-resource-group \
> -t SUNW.HAStoragePlus \
> -x FilesystemMountPoints=Apache Tomcat- instance-mount-points
```

9. **Apache-Tomcat-has-resource** を有効にします。

```
# scstat
# scswitch -e -j Apache-Tomcat-has-resource
```

10. 必要な **Apache Tomcat** コンポーネントを作成して登録します。

```
# cd /opt/SUNWsctomcat/util
# cp sctomcat_config desired place
# cp sctomcat_register desired place
```

コメントに従って sctomcat_config ファイルを編集します。

```
# これらのパラメータは、(key=value) フォームを使用してカスタマイズできます。
#
# Copyright 2003 Sun Microsystems, Inc. All rights reserved.
# Use is subject to license terms.
#
# This file will be sourced in by sctomcat_register and the parameters
# listed below will be used.
#
# These parameters can be customized in (key=value) form
#
# RS - name of the resource for the application
# RG - name of the resource group containing RS
# PORT - name of the port number
# LH - name of the LogicalHostname SC resource
# SCALABLE - true or false for a scalable or failover resource
# PFILE - absolute pathname to the parameter file for the Tomcat resource
# HAS_RS - name of the HAStoragePlus SC resource
# (it can be a , separated list for the dependencies)
(日本語)
# このファイルは、sctomcat_register によってソースに取り込まれます。
# パラメータは以下のものが使用されます。
#
# これらのパラメータは、(key=value) フォームを使用してカスタマイズできます。
#
# RS - アプリケーションのリソースの名前
# RG - RS を含むリソースグループの名前
# PORT - ポート番号の名前
```

```

#           LH - LogicalHostname SC リソースの名前
# SCALABLE - スケーラブルリソースかフェイルオーバーリソースかを true または false で指定
# PFILE - Tomcat リソースのパラメタファイルの絶対パス名
# HAS_RS - HASStoragePlus SC リソースの名前
#           (依存関係にもとづいて個別のリストにもできる)
#

```

```

RS=
RG=
PORT=8080
LH=
SCALABLE=false
PFILE=
HAS_RS=

```

次に、Apache Tomcat の場合の例を示します。

```

RS=tomcat-res
RG=tomcat-rg
PORT=8080
LH=tomcat-lh
SCALABLE=true
PFILE=/tomcat/pfile
HAS_RS=Apache-Tomcat-has-resource

```

パラメタ LH には、共有アドレス名を設定する必要があります。パラメタ HAS_RS needs を設定する必要があるのは、ほかのファイルシステム (「グローバルファイルシステム」GFS またはローカルファイルシステム) が使用されている場合で、適切な Apache-Tomcat-has-resource が作成されているときだけです。

11. `sctomcat_config` の編集後、リソースを登録する必要があります。

```
# ./sctomcat_register
```

必要なすべての Apache Tomcat インスタンスで、最後の 3 つの手順を繰り返します。

12. Apache Tomcat リソースを有効にします。

先の手順で作成したすべての Apache Tomcat リソースで次の手順を繰り返します。

```
# scstat
# scswitch -e -j Apache-Tomcat-resource

```

Sun Cluster HA for Apache Tomcat のインストールと構成の確認

ここでは、データサービスが正しくインストールされ構成されているかどうかを確認する手順について説明します。

▼ Sun Cluster HA for Apache Tomcat のインストールと構成を確認する方法

この手順は、Sun Cluster HA for Apache Tomcat のインストールと構成が正しく行われたことを確認する場合に実行します。

1. **Apache Tomcat** のホストとなるクラスタノードの **1** つで、スーパーユーザーになります。
2. **scstat** を使用して、すべての **Apache Tomcat** リソースがオンライン状態であることを確認します。

```
# scstat
```

オンライン状態でない Apache Tomcat リソースで、次のように **scswitch** コマンドを実行します。

```
# scswitch -e -j Apache Tomcat- resource
```

3. フェイルオーバーデータサービスを配備した場合は、**scswitch** コマンドを次のように実行して **Apache Tomcat** リソースグループをほかのクラスタノード (*node2* など) に切り替えます。

```
# scswitch -z -g Apache-Tomcat-failover-resource-group -h node2
```

Sun Cluster Apache Tomcat HA パラメタファイルの概要

この節の情報は、Sun Cluster Apache Tomcat HA パラメタファイルの内容について理解するために使用してください。ここでは、Sun Cluster Apache Tomcat HA パラメタファイルの構造と内容について紹介するとともに、このファイルの変数を選択する方法について説明します。

Apache Tomcat HA パラメタファイルの構造

Apache Tomcat HA は、パラメタファイルを使用して起動、停止、および検証を行う各コマンドにパラメタを渡します。このパラメタファイルは、いくつかの変数を設定する有効な korn シェルスクリプトとして作成する必要があります。このファイルの構造は次のとおりです。パラメタファイルの例は、22 ページの「Sun Cluster HA for Apache Tomcat の登録と構成」を参照してください。

表 1-6 Apache Tomcat HA パラメタファイルの構造

変数	意味
EnvScript	これは、Apache Tomcat を所有するユーザー名のログインシェルによって決まる、ksh スクリプトまたは csh スクリプトです。このスクリプトは、Apache Tomcat インスタンスの起動と停止を行うために必要な Apache Tomcat 固有の環境変数を設定するために使用されます。
User	Apache Tomcat インスタンスの所有者。
Basepath	Basepath は、Tomcat bin ディレクトリが置かれているディレクトリの絶対パスです。通常は TOMCAT_HOME (3.x) または CATALINA_HOME (4.x) です。
Host	変数 Host は、Apache Tomcat サーバプロセスの健全性を調べるホストを表します。このテストは、Host:Port に対する接続を介して行われます。
Port	Apache Tomcat がサービスを行うポート。この Port と、Host を併用して、Apache Tomcat サーバプロセスの健全性を調べます。
TestCmd	この変数は、Apache Tomcat サーバプロセスの健全性を調べるためにそのプロセスに渡されるコマンドを示します。
ReturnString	変数 ReturnString は、TestCmd に対する応答内に存在する必要がある文字列を示します。「Connection refused」という文字列は、Apache Tomcat サーバプロセスが稼動していない場合に応答内で使用されるので、この文字列は指定できません。
Startwait	この変数は、Apache Tomcat の起動コマンドが完了したあと待機する時間を秒数で指定するために使用します。Apache Tomcat の起動コマンドが完了したあと Apache Tomcat サーバプロセスが完全に稼動可能な状態になるまで、多少時間がかかります。実際の秒数は、ハードウェアの速度と負荷によって異なります。通常は、10 秒から 20 秒の間を指定することをお勧めします。

上記のすべてのパラメタは、任意の時点で変更できます。唯一の違いは、どの時点で変更が適用されるかです。

Apache Tomcat HA パラメタファイルの次のパラメタは、Apache Tomcat の起動と停止に使用されます。これらのパラメタに対する変更は、Apache Tomcat リソースの再起動または無効にして有効にする操作によって適用されます。

- EnvScript
- User
- Basepath
- Startwait

Apache Tomcat HA パラメタファイルの次のパラメタは、障害モニター内で使用されます。これらのパラメタに対する変更は、`Thorough_probe_interval` ごとに適用されます。

- Host
- Port
- TestCmd
- ReturnString

TestCmd 変数と ReturnString 変数の選択方法

ReturnString はいくつかの方法で選択できます。

1. アプリケーションの起動ページを利用し、TestCmd に `get /startpage` を設定します。この場合は、起動ページに含まれる文字列を ReturnString に設定します。この方法で、Apache Tomcat がユーザーのアプリケーションに対してサービスを提供していることを確認できます。
2. Apache Tomcat のデフォルトの起動ページを使用し、TestCmd に `get /default-startpage` を設定します。この場合は、起動ページに含まれる文字列を ReturnString に設定します。この文字列は、配備された Apache Tomcat バージョンによって異なります。3.x の場合は *Tomcat home* であり、4.x の場合は *CATALINA* です。この方法で、Apache Tomcat がデフォルトのアプリケーションに対してサービスを提供していることを確認できます。
3. Apache Tomcat にテスト用のアプリケーションを配備します。(Sun Cluster Apache Tomcat エージェントには付属していません) `get /startpage-of-the-application` を TestCmd に設定します。この場合は、起動ページに含まれる文字列を ReturnString に設定します。この方法で、Apache Tomcat がユーザーのテストアプリケーションに対してサービスを提供していることを確認できます。
4. 上記のどの方法も適切でない場合は、`get /a-page-which-does-not-exists` を TestCmd に設定します。この場合は、Error Page に含まれる文字列を ReturnString に設定します。この方法では、存在しないページが配布されたことがわかるため、Apache Tomcat が稼動していることを確認できます。

ブラウザで `hostname:port` を指定して接続すれば、異なるページを評価できます。

Sun Cluster HA for Apache Tomcat の障害モニターの概要

この節の情報は、Sun Cluster HA for Apache Tomcat の障害モニターについて理解するために使用してください。ここでは、Sun Cluster HA for Apache Tomcat の障害モニターの検証アルゴリズムと機能について説明し、検証エラーに関連する条件、メッセージ、および回復アクションを示します。

障害モニターの概念については、『*Sun Cluster 3.1 の概念*』を参照してください。

リソースプロパティ

Sun Cluster HA for Apache Tomcat の障害モニターは、リソースタイプ `SUNW.gds` と同じようなリソースプロパティを使用します。使用するリソースプロパティの全リストについては、`SUNW.gds (5)` のマニュアルページを参照してください。

検証アルゴリズムと機能

Apache Tomcat の健全性を監視するために次の作業が行われます。

1. `Thorough_probe_interval` の間、休止します。
2. Apache Tomcat パラメタファイル内に構成されている *Host* を ping コマンドで確認します。
3. *Host* と *Port* を介して Apache Tomcat に接続します。接続が正常に行われると、*TestCmd* が送信され、*ReturnString* が戻るかどうかのテストが行われます。接続が失敗すると、5 秒後に再スケジュールされます。再び失敗すると、検証プログラムによって Apache Tomcat が再起動します。



注意 - 「**Connection refused**」は接続が不可能な場合に返される文字列であるため、`ReturnString` にこの文字列を指定することはできません。

4. Apache Tomcat プロセスが停止した場合には、`pmf` が検証プログラムに割り込み、Apache Tomcat をただちに再起動します。
5. Apache Tomcat が繰り返し再起動され、最終的に `Retry_interval` の `Retry_count` がゼロになると、別のノードでそのリソースグループのフェイルオーバーが開始されます。これは、リソースプロパティ `Failover_enabled` に `TRUE` が設定されている場合の動作です。

Sun Cluster HA for Apache Tomcat のデバッグ方法

▼ Sun Cluster HA for Apache Tomcat のデバッグを有効にする方法

この節の情報は、Sun Cluster HA for Apache Tomcat のデバッグを有効にする場合に使用してください。

Sun Cluster HA for Apache Tomcat は、複数の Apache Tomcat インスタンスで使用できます。デバッグは、すべての Apache Tomcat インスタンスに対して有効にすることも、特定の Apache Tomcat インスタンスに対して有効にすることもできます。

Apache Tomcat コンポーネントの DEBUG ファイルは /opt/SUNWsctomcat/etc にあります。

このファイルを使用して、すべての Apache Tomcat インスタンスについてデバッグを有効にすることも、特定の Sun Cluster ノード上の特定の Apache Tomcat インスタンスについてデバッグを有効にすることもできます。Sun Cluster 全体に対して Sun Cluster HA for Apache Tomcat のデバッグを有効にする必要がある場合は、Sun Cluster 内の全ノードで次の手順を繰り返す必要があります。

次の手順を、Sun Cluster の各ノード上の、デバッグ出力が必要な Apache Tomcat に対して行います。

1. /etc/syslog.conf を編集します。

/etc/syslog.conf を編集し、daemon.notice を daemon.debug に変更します。

```
# grep daemon /etc/syslog.conf
*.err;kern.debug;daemon.notice;mail.crit      /var/adm/messages
*.alert;kern.err;daemon.err                    operator
#
```

daemon.notice を daemon.debug に変更し、syslogd を再起動します。次に示す grep daemon /etc/syslog.conf コマンドの出力例では、daemon.debug が設定されています。

```
# grep daemon /etc/syslog.conf
*.err;kern.debug;daemon.debug;mail.crit      /var/adm/messages
*.alert;kern.err;daemon.err                    operator
#
# pkill -1 syslogd
```

2. /opt/SUNWsctomcat/etc/config を編集します。

/opt/SUNWsctomcat/etc/config を編集し、DEBUG= を DEBUG=ALL または
DEBUG= resource に変更します。

```
# cat /opt/SUNWsctomcat/etc/config
#
# Copyright 2003 Sun Microsystems, Inc. All rights reserved.
# Use is subject to license terms.
#
# 使用方法:
#     DEBUG=<RESOURCE_NAME> または ALL
#
DEBUG=ALL
#
```

注 - デバッグを無効にするには、この手順を逆に実行して下さい。

索引

C

C ロケール, 20

L

load_balancing_policy, 10

P

pfile, 「パラメタファイル」を参照

T

Tomcat リソースの有効化, 25, 28, 29

V

/var/sadm/install/logs ディレクトリ, 20

W

Web Start プログラム, 20

い

依存関係, 12

インストール, 14, 21

Sun Cluster HA for Apache Tomcat
Web Start プログラムによる, 20
作成されるログファイル, 20

か

確認, 17, 30
環境スクリプト, 16

け

検証, 33

こ

構成
スケーラブル構成, 9
フェイルオーバー構成, 11
マルチマスター構成, 9, 10

コマンド

groupadd, 15
scinstall, 21
scrgadm, 22, 26, 27
scstat, 25, 30
scswitch, 23, 25, 26, 30
shutdown.sh, 18
startup.sh, 18
su, 15
useradd, 15

コンポーネント
スケーラブル構成, 12
フェイルオーバー構成, 12
マルチマスター構成, 12
コンポーネントの保護, 8

し
障害モニター, 32

す
スケーラビリティ, 8

せ
制限事項, 10

て
デバッグ, Sun Cluster HA for Apache Tomcat の
デバッグ, 34

と
登録, 22, 24, 25, 28
登録スクリプト, 13, 24, 28
登録ファイル, 13

は
配備, 18
パラメタファイル, 23, 26, 30

ふ
ファイル, インストールログ, 20

ほ
方法, 32

よ
要件, 11

り
リソースグループの有効化, 23, 26

ろ
ログファイル, インストール, 20
ロケール, 20