



**SIOS Protection Suite for Linux  
Apache Web Server Recovery Kit  
v9.0.1**

**管理ガイド**

**2015年12月**

本書およびその内容は SIOS Technology Corp. (旧称 SteelEye® Technology, Inc.) の所有物であり、許可なく使用および複製は禁止されています。SIOS Technology Corp. は本書の内容に関していかなる保証も行いません。また、事前の通知なく本書を改訂し、本書に記載された製品に変更を加える権利を保有しています。SIOS Technology Corp. は、新しい技術、コンポーネント、およびソフトウェアが利用可能になるのに合わせて製品を改善することを方針としています。そのため、SIOS Technology Corp. は事前の通知なく仕様を変更する権利を保留します。

LifeKeeper、SteelEye、および SteelEye DataKeeper は SIOS Technology Corp. の登録商標です。

本書で使用されるその他のブランド名および製品名は、識別のみを目的として使用されており、各社の商標が含まれています。

出版物の品質を維持するために、弊社は本書の正確性、明瞭性、構成、および価値に関するお客様のご意見を歓迎いたします。

以下の宛先に電子メールを送信してください。

[ip@us.sios.com](mailto:ip@us.sios.com)

Copyright © 2015

By SIOS Technology Corp.

San Mateo, CA U.S.A.

All rights reserved

# 目次

---

<b>Chapter 1: はじめに</b> .....	<b>3</b>
SteelEye Protection Suite ドキュメンテーション .....	3
参考ドキュメント .....	3
<b>Chapter 2: 要件</b> .....	<b>5</b>
キットのハードウェア/ソフトウェア要件 .....	5
<b>Chapter 3: LifeKeeper における Apache Web Server の設定</b> .....	<b>7</b>
設定の定義と例 .....	7
ローカル設定 .....	7
共有設定 .....	9
アクティブ/スタンバイ設定とアクティブ/アクティブ設定 .....	11
Apache Web Server の設定に関する考慮事項 .....	11
<b>Chapter 4: LifeKeeper の設定作業</b> .....	<b>15</b>
Apache Web Server リソース階層の作成 .....	16
リソース階層の拡張 .....	17
リソース階層の拡張解除 .....	20
リソース階層の削除 .....	20
リソース階層のテスト .....	21
GUI からの手動スイッチオーバーの実行 .....	21
リカバリ動作 .....	21
<b>Chapter 5: トラブルシューティング</b> .....	<b>23</b>
階層作成時のエラー .....	23
Web Server バイナリの位置検証時 .....	23
Web Server のルートディレクトリの検証時 .....	24
Apache リソース階層の作成時 .....	25
階層拡張時のエラー .....	26

---

Web Server バイナリの位置検証時 .....	26
ターゲットシステムの Apache 設定ファイルの検証時 .....	26
ターゲットサーバでの Apache リソース階層の作成時 .....	27
階層リソースのリストア、削除、リカバリ時のメッセージとエラー .....	27
Apache リソースを In Service にするとき (リストア) .....	28
Apache リソースを Out of Service にするとき (削除) .....	29
Apache リソースを In Service に戻すとき (リカバリ) .....	30

## Chapter 1: はじめに

SIOS Protection Suite (SPS) for Linux Apache Web Server Recovery Kit は、SPS 環境で Apache Web Server ソフトウェアの障害回復能力を向上します。

このガイドでは次の事項について説明します。

- [SPS ドキュメンテーション](#): SPS for Linux のドキュメンテーションの全リスト、および情報が入手可能な場所。
- [要件](#): リカバリソフトウェアのインストールおよびセットアップを行うには、サーバが一定のハードウェア/ソフトウェア要件を満たしている必要があります。SPS Apache Recovery Kit のインストール/アンインストール方法については、SPS インストールガイドを参照してください。
- [Recovery Kit の設定](#): 必要とする保護と柔軟性が得られるように SPS を設定するには、設定ルールを把握する必要があります。設定を適切にプランニングするには、ネットワーク設定、インターフェースの選択、ユーザシステムのセットアップ、階層オプション、Apache 設定作業について理解することが必要です。このセクションでは、設定のプランニングだけでなく、設定の例を示し、Recovery Kit を設定するために必要な特定の作業について説明します。
- [トラブルシューティング](#): このセクションには、情報メッセージとエラーメッセージ、および推奨される対処方法の一覧を示します。

### SteelEye Protection Suite ドキュメンテーション

以下に、SIOS Technology Corp が発行している SPS 関連情報の一覧を示します。

- SPS for Linux リリースノート
- SPS for Linux テクニカルドキュメンテーション (LifeKeeper GUI 内の [Help] メニューから参照できます)。
- SPS for Linux インストールガイド

このドキュメンテーション、および他の SPS Recovery Kit に関連するドキュメンテーションは、以下のサイトからオンラインで使用できます。

<http://docs.us.sios.com>

### 参考ドキュメント

以下に、Apache Web Server アプリケーションおよび SPS Apache Recovery Kit に関連する参考ドキュメントの一覧を示します。

- Apache のオンラインドキュメンテーション
- Apache: The Definitive Guide, 2<sup>nd</sup> Edition, Ben Laurie and Peter Laurie, O'Reilly & Associates, Inc. 1999



## Chapter 2: 要件

Apache Recovery Kit のインストールやアンインストールを試みる前に、このパッケージのハードウェア/ソフトウェア要件と、インストールとアンインストールの手順を理解しておく必要があります。

### キットのハードウェア/ソフトウェア要件

LifeKeeper Apache Recovery Kit をインストールして設定する前に、使用している設定が以下の要件を満たしていることを確認してください。

- サーバ。Recovery Kit を使用するには、SPSfor Linux テクニカルドキュメンテーションとSPS リリースノートに記載されている LifeKeeper の要件に従って設定した、サポートされるコンピュータが 2 台以上必要です。これらのドキュメントは、SIOS のテクニカルドキュメンテーションサイト ([docs.us.sios.com](http://docs.us.sios.com)) にあります。
- LifeKeeper ソフトウェア。各サーバに同じバージョンの LifeKeeper ソフトウェアとパッチをインストールする必要があります。特定の LifeKeeper の要件については、SPS リリースノートおよび SPS for Linux テクニカルドキュメンテーションを参照してください。
- LifeKeeper IP Recovery Kit。同じバージョンの Recovery Kit を各サーバにインストールする必要があります。
- IP ネットワークインターフェース。各サーバは、イーサネット TCP/IP をサポートするネットワークインターフェースを 1 つ以上必要とします。IP スイッチオーバーが正しく動作するには、ローカルネットワークに接続されているユーザシステムが標準の TCP/IP 仕様に準拠している必要があります。  
注記: 各サーバが必要とするネットワークインターフェースが 1 つだけであっても、異種メディア要件、スループット要件、単一障害点の排除、ネットワークのセグメンテーションといった多くの理由のために、複数のインターフェースを使用すべきです。
- TCP/IP ソフトウェア。各サーバは TCP/IP ソフトウェアも必要とします。
- Apache ソフトウェア。LifeKeeper および LifeKeeper Apache Web Server Recovery Kit を設定する前に、各サーバに Apache Web Server、および使用する DSO (動的共有モジュール) をインストールし、設定しておく必要があります。Web サーバソフトウェアのすべてのパッケージについて、各サーバに同じバージョンをインストールする必要があります。最新リリースの互換性と発注情報については、SPS リリースノートを参照するか、営業担当者にお問い合わせください。

LifeKeeper Apache Recovery Kit のインストール/アンインストール方法については、SPS インストールガイドを参照してください。



## Chapter 3: LifeKeeper における Apache Web Server の設定

本セクションでは、LifeKeeper における典型的な Apache Web Server の設定の定義および例と、Apache Web Server の設定を開始する前に考慮すべき情報について説明します。

LifeKeeper Core のリソース階層を設定する手順については、SPS for Linux テクニカルドキュメンテーションを参照してください。

### 設定の定義と例

Apache Web Server は、httpd デーモンの複数インスタンスの同時実行をサポートしています。LifeKeeper の各 Apache Web Server 階層は、固有の「サーバルート」ディレクトリを持つ個別の Apache インスタンスに対応します。各インスタンスは、「仮想ホスト」を使用するように設定されているかどうかにより、1 つ以上の Web サイトをサポートできます。

主に、サーバルートディレクトリは 1 つの Apache Web Server インスタンスを定義します。これは、Web インスタンスの設定方法を指定する `conf/httpd.conf` 設定ファイルがこのディレクトリに入れられるからです。このファイル内にある Apache 設定ディレクティブにより、このインスタンスのログファイル、Web ドキュメント、その他の設定ファイルなどの場所、および使用する IP/ドメイン名のアドレスが指定されます。

LifeKeeper ファイルシステム (共有ストレージを使用) を使用するかどうかに基づいて、Apache Web Server の設定を分類すると便利です。1 つの共有ファイルシステムを、サーバルートディレクトリ (設定ファイル `conf/httpd.conf` とともに使用) やドキュメントルートディレクトリ (さらにオプションで httpd 実行可能ファイル自体) に使用できます。特定の Apache インスタンスについて、ローカル設定と共有設定のいずれを使用するかは、2 つの主要な要因によって決まります。その 1 つは、設定ファイルや Web サイトのドキュメントの複数の同一コピーを個別に管理することの困難さであり、もう 1 つは、2 台以上のサーバで共有 (またはミラー化) できるストレージの可用性とアクセス容易性です。ただし、1 台のサーバ上に、ローカルの Apache インスタンスと共有の Apache インスタンスの両方を設定できることに注意してください。

以降のセクションでは、LifeKeeper 環境での [ローカル](#) と [共有](#) の Apache Web Server 設定の例を示し、それぞれの主な特徴の概要を説明します。

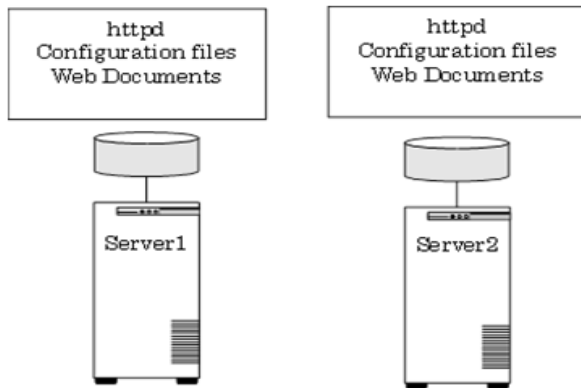
### ローカル設定

一般的なローカル設定では、複数のサーバ間で共有されるものではありません。Apache Web Server の設定ファイル、Web ドキュメント、DSO モジュール (およびその設定ファイル)、および httpd 実行可能ファイルのそれぞれについて、同一コピーが各サーバの正確に同じ位置に存在します。各サーバに Apache コンポーネントの同一コピーを維持するのは、Apache 管理者の責任です。

設定ファイルにより、各 Web サイトに IP アドレス、または特定の IP アドレスをマップするドメインアドレスが割り当てられます。それぞれのアドレスについて LifeKeeper の IP アドレスが作成され、Apache リソース階層に追加されます。あるサーバから別のサーバに Apache 階層がスイッチオーバーすると、元のサーバではこの特定の httpd イン

タンスが停止し、IP アドレスが非アクティブになります。次に、スイッチオーバー先のサーバで、IP アドレスがアクティブになり、インスタンスが開始されます。その後、TCP/IP 経路でクライアントがスイッチオーバー先のサーバ上にある同一の Web サイトに自動的に接続されます。

図 1: ローカル設定



設定に関する注記:

- 図 1 は、共有ファイルシステムに何も配置しないローカル設定の例です。
- 各サーバの同じ位置 (通常は /usr/bin/httpd) に、同一バージョンの Apache Web Server 実行可能ファイルがあります。
- 各サーバは同一のサーバルートディレクトリを持ち、そこには各インスタンスの設定ファイルの同一コピーが配置されます。
- 各サーバは同一のドキュメントルートディレクトリを持ち、そこには各インスタンスの Web ドキュメントの同一コピーが配置されます。
- DSO モジュールを使用する場合は、各サーバの同じ場所にその同一コピーがあります。

Server 1 での Apache Web Server リソース階層の作成:

サーバ:	Server1
Web Server バイナリの位置:	/usr/sbin/httpd
Web Server のルートディレクトリ:	/home/www/examples/instance1/
ルートタグ	apache-www.examples.instance1

Apache Web Server リソース階層の Server 2 への拡張:

テンプレートサーバ:	Server1
拡張するタグ:	apache-www.examples.instance1
ターゲットサーバ:	Server2
ターゲットの優先順位:	10

Apache リソース階層を 1 台以上の追加サーバに拡張するときには、ローカル設定と共有設定のいずれの場合でも、すべてのサーバで同一の Web Server バイナリの位置、および同一の Web Server のルートディレクトリを使用する必要があります。詳細については、前述の説明、および[Apache Web Server の設定に関する考慮事項](#)のセクションを参照してください。また、階層の拡張時に、Apache リソース階層の一部を構成する依存リソースがすべて、LifeKeeper により拡張されます。

## 共有設定

一般的な共有設定では、サーバルートディレクトリとドキュメントルートディレクトリはすべて、同一の共有ファイルシステムに存在します。同一の設定ファイルと Web ドキュメントが複数のサーバで共有されるので、各サーバに同一コピーを維持する必要はありません。DSO モジュールを使用する場合は、必要となる設定ファイルやリソースとともに、同じ共有ファイルシステムに配置できます。

Web ドキュメントのみを共有ファイルシステムに配置できるようにしてください。この方法では、サーバルートディレクトリがローカルなので一般的なローカル設定に似ていますが、階層には共有ファイルも含まれます。

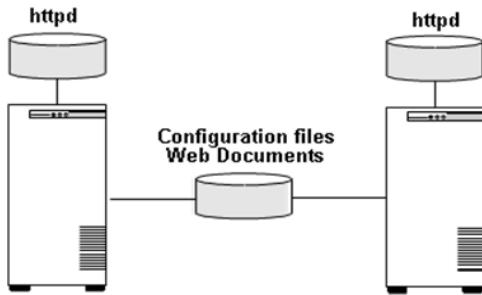
この Apache リソース階層で Apache 実行可能ファイルの特定バージョンまたは個別のコピーを使用する場合は、この実行可能ファイルも共有ファイルシステムに配置できます。この実行可能ファイルは、このインスタンスにのみ使用可能になります。この場合は、Web Server バイナリの位置の入力を要求されたときに、共有ファイルシステム上にある httpd 実行可能ファイルのフルパスを入力します。

使用できる共有ファイルシステムは 1 つのみであることに注意してください。これにより、確実に、共有ストレージにあるすべての必須コンポーネントが同時に利用可能になります。共有ファイルシステム上の Web Server バイナリの位置を使用する場合は、同じ共有ファイルシステム上にある Web Server のルートディレクトリも選択する必要があります。また、このサーバルート用に設定した DocumentRoot ディレクトリがすべて、同じ共有ファイルシステム上に存在する必要があります。同様に、共有ファイルシステム上の Web Server のルートディレクトリを選択するときには、DocumentRoot ディレクトリがすべて、同じ共有ファイルシステム上に存在する必要があります。バイナリとサーバルートが共有ファイルシステムに存在しないが、いずれかの DocumentRoot ディレクトリが共有される場合、すべての DocumentRoot ディレクトリを同じファイルシステム上で共有する必要があります。

これらの規則は、以下のように要約できます。

- Apache 実行可能ファイルを共有する場合は、サーバルートディレクトリを共有する必要があります。
- サーバルートディレクトリを共有する場合は、すべての DocumentRoot ディレクトリを共有する必要があります。
- DocumentRoot ディレクトリを 1 つでも共有する場合は、すべての DocumentRoot ディレクトリを共有する必要があります。
- 個々の Apache リソース階層について、共有ファイルシステムを 1 つのみ使用できます。

図 2: 共有設定



設定に関する注記:

- 図 2 は、設定ファイルと Web ドキュメントを共有する共有設定の例です。
- Web ドキュメントのみを共有ファイルシステムに配置できます。この方法は一般的なローカル設定に似ていますが、階層には共有ファイルシステムも含まれる点が異なります。
- DSO モジュールを使用する場合は、必要な設定ファイルやリソースとともに、共有ファイルシステムに配置できます。

Server 1 での Apache Web Server リソース階層の作成:

サーバ:	Server1
Web Server バイナリの位置:	/usr/sbin/httpd または /shared/example/instance2/bin/httpd
Web Server のルートディレクトリ:	/shared/example/instance2
ルートタグ	apache-shared.example.instance2

Apache Web Server リソース階層の Server 2 への拡張:

テンプレートサーバ:	Server1
拡張するタグ:	apache-shared.example.instance2

ターゲット サーバ	Server2
ターゲットの優先順位:	10

## アクティブ / スタンバイ設定とアクティブ / アクティブ設定

Apache Web Server は、LifeKeeper ではアクティブ / アクティブアプリケーションと呼ばれます。これは、いつでも複数の Apache インスタンスをサーバ上で実行できるということです。例えば、2 台のサーバが Apache インスタンスを 1 つずつ実行していて、1 台のサーバに障害が発生した場合、このサーバの Apache インスタンスがもう 1 台のサーバにフェイルオーバーでき、さらにもともと実行していたインスタンスの実行も続行できます。一部のアプリケーションではこの機能をサポートしていないため、アプリケーションのインスタンスごとにサーバを使用可能にしておく必要があります。そのようなアプリケーションは、アクティブ / スタンバイアプリケーションと呼ばれます。いずれの設定も使用できるアプリケーションもあります。

Apache を「アクティブ / スタンバイ」モードで実行したい状況が発生することがあります。特に、例えばサーバの 1 台を Apache の実行専用にする場合です。この特殊な場合では、バックアップサーバで何も実行しないように、標準的な Apache のデフォルトインストールの自動起動を無効にする必要があります。

1 台以上の特定サーバで Apache インスタンスを手動で in service にすることで、任意に負荷を分散できます。さらに、各インスタンスのサーバ優先順位を調整することで、障害が発生したときに、Apache インスタンスが最終手段として特定サーバにフェイルオーバーするように設定することも、複数のサーバにフェイルオーバーして負荷を分散するように設定することもできます。

クラスタ内のすべてのサーバで Apache の自動起動を無効にした場合、このインスタンスで LifeKeeper の IP アドレスを使用するように設定することで、デフォルトサーバのルートディレクトリ「/etc/http」を LifeKeeper の Apache リソース履歴に使用できます。また、ドキュメントのルートディレクトリ用に共有ファイルシステムを使用できます。ただし、これはアクティブ / スタンバイ設定 (前述) なので、通常の方法ではデフォルトのインスタンスを起動できなくなります。さらに、サーバのルートは一意である必要があるため、デフォルトサーバのルートディレクトリを複数の履歴に使用することはできません。

## Apache Web Server の設定に関する考慮事項

Apache リソース階層を作成する前に、Apache Web Server アプリケーションについて以下の設定を完了したことを確認する必要があります。

1. 標準デフォルトの Apache インスタンスは、修正を行わない限り、LifeKeeper が保護するインスタンスと競合します。Apache パッケージのインストール時に、デフォルトの Apache インスタンスが、システムのセットアップ時に /etc/rc.d ディレクトリにある runlevel (rc) スクリプトにより起動するように自動設定されます。デフォルトの Apache インスタンスに関連付けられている httpd.conf ファイルには、BindAddress ディレクティブ、またはすべての IP アドレスのリسن機能 (BindAddress) に相当する Listen ディレクティブがありません。  
\*

LifeKeeper が保護する Apache インスタンスに加えて、標準デフォルトの Apache インスタンスを使用する場合は、BindAddress または Listen のディレクティブを使用して、LifeKeeper が保護しない 1 つ以上の特定の IP アドレスをリسنするように設定ファイル (/etc/httpd/conf/httpd.conf)\* を変更する必要があります。変更内容が検出されるように、httpd デーモンを停止し、再起動する必要があります。デーモンの停止と再起動には、以下のコマンドを使用できます (SuSE の場合のスクリプト名は /etc/init.d/apache)。

```
/etc/rc.d/init.d/httpd stop
```

```
/etc/rc.d/init.d/httpd start
```

設定を変更しない場合は、デフォルトの Apache インスタンスの自動起動を無効にする必要があります。無効にしない場合、このインスタンスが、LifeKeeper の Apache インスタンスの動作を妨害します。デフォルトインスタンスの自動起動を無効にする手順については、Apache Web Server のドキュメンテーションを参照してください。

**\* SuSE で Apache を使用する場合:** デフォルト設定を使用して Apache を SuSE にインストールする場合、httpd.conf 設定ファイルは ServerRoot のサブディレクトリ conf に配置されません。この場合は、設定ファイルを /etc/httpd/conf ディレクトリに再配置する必要があります。

- LifeKeeper の個々の Apache Web Server 階層について、個別のルートディレクトリを作成する必要があります。この「サーバルート」ディレクトリは、Apache の「ServerRoot」設定とコマンドラインパラメータに対応します。LifeKeeper の各 Apache Web Server 階層は、一意の Apache インスタンスおよび関連付けられたサーバルートディレクトリに対応します。サーバルートディレクトリは、すべてのサーバ上で同一であり、特定の Apache 階層用に設定する必要があることに注意してください。Web サイトのすべての設定ファイルの情報は、LifeKeeper ソフトウェアから検出とアクセスができるように、サーバルートからの標準の場所 (conf/httpd.conf) に配置する必要があります。
- また、すべての設定情報を 1 つの httpd.conf ファイルに入れるという Apache の推奨事項にも従う必要があります。Apache Web Server に付属する標準の srm.conf ファイルと access.conf ファイルにはディレクティブが含まれているので、LifeKeeper の設定に干渉しないようにこれらのファイルは無効にする必要があります (デフォルトでは、使用するサーバルートディレクトリには無関係に、Apache はこれらのファイルを /etc/httpd/conf から探します)。これら 2 つの設定ファイルの使用を無効にするには、以下のディレクティブを httpd.conf の本体に含めます。

```
AccessConfig /dev/null
```

```
ResourceConfig /dev/null
```

- BindAddress または Listen のディレクティブを使用して特定の LifeKeeper の IP アドレスをリスンするように、すべての Web サイト (仮想ホスト) を設定する必要があります。これらの LifeKeeper が保護する IP アドレスは、Apache 階層を作成する場所にあらかじめ作成し、in service 可能にしておく必要があります。LifeKeeper が保護する IP アドレスは、Apache リソース階層に自動的に追加されます。

LifeKeeper の共有ファイルシステムを使用する場合は、Apache 階層を作成する前に、ファイルシステムの作成に必要な準備をすべて行う必要があります。特に、サーバ上の Apache 階層を作成する場所に、ファイルシステムをマウントする必要があります。LifeKeeper のファイルシステム階層をあらかじめ作成していない場合、このファイルシステム階層は Apache 階層とともに自動作成され、その後 Apache リソース階層に結合されます。

仮想ホストの設定の詳細については、Apache Web Server のドキュメンテーションを参照してください。前述したように、すべての Apache インスタンスが、LifeKeeper が保護する特定のアドレスをリスンするように設定する必要があります。例えば、IP ベースの仮想ホストとネームベースの仮想ホストを組み合わせたインスタンスの設定ファイルには、以下のようなディレクティブがあります。

```
User webuser
Group webgroup
ServerName localhost
```

```
AccessConfig /dev/null
ResourceConfig /dev/null
```

```
Listen 172.17.100.55:8000
NameVirtualHost 172.17.100.55:8000
Listen 172.17.100.56:80

<Virtualhost site.name_one:8000>
ServerName site.name_one
DocumentRoot /shared/site/name_one
</VirtualHost>

<VirtualHost site.name_two:8000>
ServerName site.name_two
DocumentRoot /shared/site/name_two
</VirtualHost>

<VirtualHost 172.17.100.56:80>
ServerName site.ip
DocumentRoot /shared/site/ip
</VirtualHost>
```

5. Apache インスタンスに対する SSL サポートを有効にしている場合、LifeKeeper が保護する適切な IP アドレスを使用するように SSL Listen ディレクティブ (多くの場合、個別の ssl.conf ファイル内にある) を設定する必要があります。この設定を行わない場合、IP アドレス 0.0.0.0 が LifeKeeper で保護されていないことを示すエラーが発生して、Apache 階層の作成に失敗します。一部の Linux 配布の Apache 設定ファイルでは、デフォルトで SSL サポートが有効になっていることに注意してください。

例えば、/etc/httpd/conf.d/ssl.conf にあるデフォルトの SSL 設定ファイルで、エントリ

```
Listen 0.0.0.0:443
```

を以下のように変更します。

```
Listen 172.17.100.55:443
```

6. ローカル設定の場合、プライマリサーバとすべてのバックアップサーバの同じ場所に Apache をインストールして設定し、同一の (または同等の) 設定ファイルをすべてのサーバ上の同じサーバルートディレクトリに設定する必要があります。また、すべてのサーバ上にすべてのドキュメントルートディレクトリが存在し、同一のファイルを持つ必要があります (「設定と定義の例」の [ローカル設定](#) を参照)。
7. 共有設定では通常、LifeKeeper の共有ファイルシステム上にサーバルートディレクトリを設定します。使用できる共有ファイルシステムは 1 つのみであることに注意してください。これにより、確実に、共有ストレージにあるすべての必須コンポーネントが同時に利用可能になります。このため、すべてのドキュメントルートディレクトリが、同じ共有ファイルシステムのサブディレクトリである必要があります。ただし、サーバルートディレクトリ自体のサブディレクトリにする必要はありません。Apache 実行可能ファイルも同じ共有ファイルシステムに配置できますが、この実行可能ファイルは、この特定の Apache リソース階層のみが使用できます。

**注記:** 共有ストレージを使用する目的で、サーバルートディレクトリを共有ファイルシステムに配置する必要はありません。設定ファイルなどを入れるローカルのサーバルートディレクトリを選択でき、ドキュメントルートディレクトリのみを共有ファイルシステムに配置できます。ただし、すべてのサーバ上に (前述のローカル設定について)、同一のサーバルートディレクトリ、および同一の (または同

等の) 設定ファイルを設定する必要があります。また、すべてのドキュメントルートディレクトリが同じ共有ファイルシステム上に存在する必要があります(「設定と定義の例」の[共有設定](#)を参照)。

8. 一部の Web サイトの実装では、特定の機能に対する Apache のサポートを拡張するために、DSO (Dynamic Shared Object: 動的共有オブジェクト) モジュールを活用しています。例えば、PHP や Perl の機能を実装するモジュールがあります。実行時に Apache コアから、これらのモジュールのロードとアクセスができます。モジュールを使用する場合は、クラスタ内のすべてのサーバでモジュールを同一設定にする必要があります。Linux プラットフォームでモジュールを使用するように Apache を設定する方法については、モジュールパッケージのドキュメンテーション、およびベンダ提供のドキュメンテーションを参照してください。使用するモジュールおよびそのリソースによっては、ファイルオーバを適切に実行するために、一部のオブジェクトを共有ストレージに配置する必要があります。場合によっては、Generic Application Recovery Kit、またはカスタムリカバリキットを使用して、個別モジュールの保護が必要になることがあります。
9. SSL (Secure Sockets Layer: セキュアソケットレイヤ) モジュールを Apache とともに使用する場合は、サーバをパスワードで保護しないことが重要です。Web サーバがパスワードで保護されている場合、デーモンを起動するたびに管理者がパスワードをプロンプトに入力しなければなりません。この手動操作は、リカバリ時間が重要な HA クラスタ環境とは矛盾するため、LifeKeeper はパスワードで保護されたインスタンスをサポートしません。以下のコマンドを使用してパスワードを削除してください。

```
openssl rsa -in server.key -out unprotected_server.key
```

プロンプトが表示されたら、サーバキーパスワードを入力してください。サイトのセキュリティを維持するために、必ずファイルはルートでのみ読み取り可能にしてください。

```
chmod 400 unprotected_server.key
```

Apache インスタンスの階層作成時に、Recovery Kit により、リソースがパスワードで保護されていないことがチェックされます。リソースがパスワードで保護されている場合、階層の作成に失敗し、エラーメッセージが表示されます。ただし、インスタンスが別のサーバに拡張されている場合、Recovery Kit はバックアップサーバでパスワードの保護をチェックしません。拡張する階層がパスワードで保護されていないことを確認する必要があります。

サーバキーファイル (Apache の設定ファイルの SSLCertificateKeyFile ディレクティブで指定) は、同じ名前を持ち、クラスタ内のすべてのサーバ上で同じ場所にある必要があります。

**注記:** LifeKeeper が使用する httpd プロセスの PID ファイル名は、以下の形式を取ります。

```
「/var/run/httpd.<TAG name>.pid」
```

この PID ファイル名は、OS が使用するデフォルトの PID ファイル名とは異なります。この PID ファイルを参照する必要がある場合 (ログのローテーションなど)、LifeKeeper の PID ファイル名と形式に注意してください。

## Chapter 4: LifeKeeper の設定作業

以下の設定作業は、LifeKeeper GUI を使用して実行できます。以下の4つの作業は、Apache リソースインスタンスに特有のものであり、Recovery Kit ごとに異なるため、本書で説明しています。

- [リソース階層の作成](#): アプリケーションリソース階層を LifeKeeper クラスタに作成します。
- [リソース階層の拡張](#): リソース階層をプライマリサーバからバックアップサーバに拡張します。
- [リソース階層の拡張解除](#): リソース階層を LifeKeeper クラスタ内の1つのサーバから拡張解除 (削除) します。
- [リソース階層の削除](#): リソース階層を LifeKeeper クラスタ内のすべてのサーバから削除します。

以下の作業は、すべての Recovery Kit で共通の作業であり、手順が同一なので、SPS テクニカルドキュメンテーションの「GUIの管理」セクションで説明しています。

- リソース依存関係の作成: 既存のリソース階層と別のリソースインスタンスとの間に親子の依存関係を作成し、クラスタ内のすべての対象サーバに依存関係の変更を反映します。
- リソース依存関係の削除: リソースの依存関係を削除して、クラスタ内のすべての対象サーバに依存関係の変更を反映します。
- In Service: リソース階層を特定のサーバで In Service にします。
- Out of Service: リソース階層を特定のサーバで Out of Service にします。
- プロパティの表示 / 編集: 特定のサーバでリソース階層のプロパティを表示または編集します。

**注記:** 本セクションの残りの部分では、LifeKeeper GUI の [Edit] メニューから特定の作業を選択して Recovery Kit を設定する方法について説明します。設定作業はツールバーから選択することもできます。状況表示ウィンドウのリソース階層ツリー (左側のペイン) のグローバルリソースを右クリックして、[Edit] メニューと同じドロップダウンメニューの選択項目を表示することもできます。言うまでもなく、これは階層がすでに存在している場合にだけ可能な方法です。

サーバおよび特定リソースの状況に応じて、状況表示ウィンドウのリソース階層表 (右側のペイン) のリソースインスタンスを右クリックして、リソース階層の作成を除くすべての設定作業を実行することもできます。

## Apache Web Server リソース階層の作成

### 重要:

Web Server リソース階層を作成する前に、Apache の設定ファイルに LifeKeeper が保護する既存の IP リソースが含まれていることを確認する必要があります。

Webドキュメントまたは設定ファイルが共有ディスク上にある共有環境では、共有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。また、重要事項として、リソースをバックアップサーバに拡張するには、稼働しているコミュニケーションパス(すなわちハートビート)が必須であることを念頭においてください。

プライマリサーバからリソースインスタンスを作成するには、以下の手順を実行します。

1. LifeKeeper GUI メニューから [Edit] を選択し、次に [Server] を選択してください。メニューから、[Create Resource Hierarchy] を選択してください。

リソースを作成するときは、Apache Web Server が停止している必要があります。ただし、システム設定ファイルにリスン変数を設定している場合は、デフォルトのデーモンの動作を許可できます。

[Create Resource Wizard] ダイアログボックスが表示され、クラスタ内にインストールされて認識済みの Recovery Kit がすべて、ドロップダウンリストボックスに表示されます。

2. [Apache Web Server] を選択し、**[Next]** をクリックしてください。
3. 以下の情報を入力するようにプロンプトが表示されます。ダイアログボックスで [Back] ボタンが有効な場合は、前のダイアログボックスに戻ることができます。これは、エラーが発生して、前に入力した情報を修正する必要がある場合に便利な機能です。いつでも [Cancel] をクリックして、作成処理全体を取り消すことができます。

フィールド	ヒント
Switchback Type	[intelligent] または [automatic] を選択してください。この設定は、バックアップサーバへのフェイルオーバー後にサーバが復帰したときに、このサーバに Apache インスタンスをスイッチバックする方法を指定します。スイッチバックタイプは、必要な場合、[Resource Properties] ダイアログボックスの [General] タブで後から変更できます。
Server	Apache Web Server を配置するサーバ(通常これをプライマリサーバまたはテンプレートサーバと呼ぶ)を選択してください。クラスタ内のすべてのサーバがドロップダウンリストボックスに表示されます。
Web Server Binary Location	httpd Apache Web Server デーモンのフルパス名 (ファイル名を含む) を選択するか、入力してください。デフォルトは、/usr/sbin/httpd です。

フィールド	ヒント
Web Server Root Directory	<p>Web Server のルートディレクトリのフルパスを指定する必要があります。相対パスやシンボリックリンクを使用することはできません。Apache Web Server の設定ファイルは、サーバルートの conf/httpd.conf にあります。</p> <p>注記: この時点で、保護される IP リソースが利用可能かどうか LifeKeeper によってチェックされます。また、Apache Web Server リソース階層の作成に使用する入力データが妥当かどうかチェックされます。この2つのチェックのどちらかで問題が検出された場合、[ERROR] ボックスが表示されます。Web Server のルートディレクトリのパスは妥当であるが、Apache の設定自体にエラーがある場合は、そのエラーを修正してから、階層の作成を続けることができます。ここで必要な LifeKeeper IP リソースを作成することもできます。</p>
Root Tag	<p>Web Server 階層に付けるタグ名を選択するか、入力してください。デフォルトの apache&lt;root directory&gt;を選択するか、独自のタグ名を入力することができます。</p>

4. [Create] をクリックしてください。[Create Resource Wizard] によって、Apache リソース階層が作成されます。LifeKeeper によって入力データが検査されます。問題が検出された場合、情報ボックスにエラーメッセージが表示されます。
5. 情報ボックスが表示され、Apache リソース階層が正常に作成されたこと、およびフェイルオーバーによる保護を実現するにはその階層をクラスタ内の別のサーバに拡張する必要があることが示されます。[Next] をクリックしてください。

**注記:** 新しい Apache インスタンスの起動に失敗したというエラーメッセージが表示されることがあります。新しい Apache 階層を別のシステムに拡張するには、起動する(In Service にする)必要があることに注意してください。この時点で、表示されたエラーメッセージに基づいて問題を修正し、新しい階層を In Service にしてから、階層の拡張を続行できます。

6. [Continue] をクリックしてください。[Pre-Extend Wizard] が起動されます。リソース階層を別のサーバに拡張する方法の詳細については、「Apache リソース階層の拡張」(後述)の手順 2 を参照してください。

[Cancel] をクリックすると、別のダイアログボックスが表示され、Apache リソース階層を LifeKeeper の保護下に置くには、いずれかの時点でここに戻って Apache リソース階層を別のサーバに拡張する必要があることが警告されます。

## リソース階層の拡張

この操作は [Edit] メニューから開始できます。または [Create Resource Hierarchy] オプションの動作が完了すると自動的に開始されます。その場合は、手順 2 を参照してください。

1. [Edit] メニューの [Resource] から [Extend Resource Hierarchy] を選択してください。[Pre-Extend Wizard] が表示されます。拡張操作に慣れていない場合は、[Next] をクリックしてください。LifeKeeper のリソース階層の拡張作業のデフォルト値が分かっている、入力と確認を省略する場合は [Accept Defaults] をクリックしてください。

## リソース階層の拡張

2. [Pre-Extend Wizard] に以下の情報を入力してください。

**注記:** 最初の2つのフィールドは、[Edit] メニューの [Extend] から操作を開始した場合にだけ表示されません。

フィールド	ヒント
Template Server	Apache リソースが現在 in service のサーバを入力してください。ここで選択するテンプレートサーバと次のダイアログボックスで選択する [Tag to Extend] によって、in service のリソース階層が表されることを認識しておくことが重要です。選択したテンプレートサーバで out of service のリソースタグを選択すると、エラーメッセージが表示されます。このダイアログのドロップダウンボックスに、クラスタ内の全サーバの名前が表示されます。
Tag to Extend	テンプレートサーバからターゲットサーバに拡張する Web Server インスタンスの名前を選択してください。ウィザードのドロップダウンリストボックスに、テンプレートサーバ上に作成した全リソースのうち、前のダイアログボックスで選択したリソースの一覧が表示されます。
Target Server	Web Server リソース階層を拡張するターゲットサーバを選択してください。ドロップダウンボックスには、選択した階層に含まれないクラスタ内のサーバ名の一覧が表示されます。
Switchback Type	[intelligent] または [automatic] を選択してください。ここでは、バックアップサーバへのフェイルオーバーの後、Web Server インスタンスが in service に戻ったときに、このサーバにどのようにスイッチバックされるのかを設定します。[intelligent] の場合、インスタンスをプライマリ/元のサーバにスイッチバックするときに管理者の操作が必要になります。[automatic] の場合、プライマリサーバがオンラインに戻り、LifeKeeper コミュニケーションパスを再確立した直後に自動的にスイッチバックが行われます。
Template Priority	テンプレート階層の優先順位を選択するか、入力してください。優先順位は、1 ~ 999 の範囲で未使用の値が有効で、小さい数字ほど優先順位が高くなります (数字 1 が最高の優先順位に相当)。拡張処理時に、別のシステムですでに使用中の優先順位をこの階層に対して指定することはできません。デフォルト値を使用することを推奨します。注記: このフィールドは、階層をはじめて拡張するときにだけ表示されます。
Target Priority	拡張する Web Server 階層のターゲットの優先順位を選択するか、入力してください。この優先順位は 1 ~ 999 の数値で表され、リソースのカスケードリングフェイルオーバーシナリオにおけるサーバの優先順位を示します。階層の優先順位は数字でソートされ、小さい数字ほど優先順位が高くなります (数字 1 が最高の優先順位に相当)。LifeKeeper では、階層が作成されたサーバに「1」が自動的に割り当てられることに注意してください。優先順位は連続している必要はありませんが、特定のリソースについて 2 つのサーバに同じ優先順位を割り当てることはできません。
	<p>拡張前のチェックが正常に終了したというメッセージが表示された後、[Next] をクリックしてください。</p> <p>拡張する階層に応じて、拡張されるリソースタグを示す一連の情報ボックス (編集不可) が表示されます。[Extend] をクリックしてください。</p>

## リソース階層の拡張

フィールド	ヒント
Network Interface	ネットワークインターフェースを選択するか、入力してください。これは、IP リソースがターゲットサーバで使用するネットワークインターフェース(つまりイーサネットカード)の名前です。
Backup Interface	IP リソースを拡張するサーバで IP ローカルリカバリ機能を使用する場合は、バックアップインターフェースを選択してください。デフォルト値は [none] です。ただし、このサーバ上で他のネットワークインターフェースカードを設定している場合は、そのネットワークインターフェースカードがドロップダウンリストに表示されます。
IP Resource Tag	IP リソースタグを選択するか、入力してください。これは、ターゲットサーバに拡張される IP リソースが使用するリソースタグ名です。
Root Tag	ルートタグを選択するか、入力してください。これは、Web Server 階層に付けたタグ名です。デフォルトでは、テンプレートサーバとターゲットサーバの両方でルートタグ名は同一です。
Mount Point	Web Server のルートディレクトリが共有ファイルシステム上に置かれている場合にのみ表示されます。  Web Server のルートディレクトリが置かれている共有ファイルシステムのマウントポイントを選択するか、入力してください。テンプレートサーバとターゲットサーバで、共有 Web Server のルートディレクトリのマウントポイントが同一である必要があります。多くの場合、ダイアログボックスに表示されたデフォルトのマウントポイントを選択する必要があります。
Root Tag	Web Server のルートディレクトリが共有ファイルシステム上に置かれている場合にのみ表示されます。  ルートタグを選択するか、入力してください。これは、共有ファイルシステムのタグ名です。

3. 拡張が実行中であることを示す情報ボックスが表示されます。同じ Apache リソースインスタンスをクラスタ内の別のサーバに拡張する場合は、[Next Server] をクリックしてください。その場合は、リソース階層を拡張する操作を繰り返します。

[Finish] をクリックした場合、Web Server リソースの拡張処理が正常に完了したかどうかを検証されます。

4. 最後のダイアログボックスで [Done] をクリックして、[Extend Resources Hierarchy] メニューの選択を終了してください。

**注記:** 必ず両方のサーバで新しいインスタンスの機能をテストしてください。

## リソース階層の拡張解除

1. [Edit] メニューの [Resource] から [Unextend Resource Hierarchy] を選択してください。
2. 拡張解除する Web Server リソースが存在するターゲットサーバを選択してください。Web Server が現在 in service になっているサーバは選択できません (このダイアログボックスは、右側のペインでリソースインスタンスを右クリックして [Unextend] 作業を選択した場合には表示されません)。  
[Next] をクリックしてください。
3. 拡張解除する Web Server 階層を選択し、[Next] をクリックしてください (このダイアログボックスは、いずれかのペインでリソースインスタンスを右クリックして [Unextend] 作業を選択した場合には表示されません)。
4. 拡張解除の対象として選択したターゲットサーバと Web Server リソース階層を確認する情報ボックスが表示されます。[Unextend] をクリックしてください。
5. Web Server リソースが正常に拡張解除されたことを示す別の情報ボックスが表示されます。
6. [Done] をクリックして終了してください。

## リソース階層の削除

階層を out of service にする前に階層を削除した場合、リソース階層は LifeKeeper の保護から除外されますが、Apache インスタンスは現在アクティブなサーバで動作を続行します (ただし、手動で停止した場合やシステムを再起動した場合を除く)。別の IP アドレスを持つ同じ Apache 階層を再作成したり、以前使用した IP アドレスを使用して新しい Apache 階層を作成 (ただし別のサーバレートをを使用) しようとする、同じアドレスで動作中の Apache インスタンスとの競合が発生します。

LifeKeeper 環境内のすべてのサーバからリソース階層を削除するには、以下の手順を実行してください。

1. LifeKeeper GUI メニューから [Edit] を選択し、次に [Resource] を選択してください。ドロップダウンメニューから、[Delete Resource Hierarchy] を選択してください。
2. 削除する Web Server リソース階層が存在するターゲットサーバの名前を選択してください。[Next] をクリックして次のダイアログボックスに進んでください。

**注記:** 右側のペインで個々のリソースインスタンスを右クリックするか、またはリソースが1台のサーバ上のみにあるときに左側のペインでグローバルリソースを右クリックして [Delete Resource] 作業を選択した場合、ターゲットサーバのダイアログボックスは表示されません。

3. [Hierarchy to Delete] を選択してください (このダイアログボックスは、左側または右側のペインでリソースインスタンスを右クリックして、[Delete Resource] 作業を選択した場合には、表示されません。) このリストボックスには、ターゲットサーバ上にある、in service と out of service の両方の階層がすべて表示されます。Apache インスタンスを停止してリソース階層を LifeKeeper の保護から削除する場合は、削除前に、選択した階層を必ず out of service にする必要があります。

[Next] をクリックしてください。

4. 選択したターゲットサーバと、削除の対象として選択した階層を確認する情報ボックスが表示されます。[Next] をしてください。

5. Web Server リソースが正常に削除されたことを確認する別の情報ボックスが表示されます。
6. [Done] をクリックして終了してください。

## リソース階層のテスト

手動スイッチオーバーを開始することによって、Apache リソース階層をテストできます。このテストでは、プライマリサーバからバックアップサーバへのリソースインスタンスのフェイルオーバーをシミュレートします。

### GUI からの手動スイッチオーバーの実行

スイッチオーバーを手動で開始するには、LifeKeeper GUI から [Edit] の [Resource] を選択し、ドロップダウンメニューから [In Service] を選択します。例えば、バックアップサーバで in service 要求を実行すると、アプリケーション階層がバックアップサーバで in service になり、プライマリサーバでは out of service になります。この時点で、元のバックアップサーバがプライマリサーバに、元のプライマリサーバがバックアップサーバに変わります。

Out of Service 要求を実行すると、アプリケーションは他のサーバで in service にならずに、out of service になります。

### リカバリ動作

プライマリサーバに障害が発生すると、Apache Recovery Kit ソフトウェアは以下の作業を実行します。

- プライマリサーバの1つ以上の物理ネットワークインターフェースで IP アドレスを in service にすることにより、Apache をバックアップサーバで in service にします。
- ファイルシステムを使用する場合は、そのサーバの共有ディスク上にファイルシステムをマウントします。
- Apache に関連するデーモンプロセスを開始します。

リカバリ後に、ブラウザの [Reload] / [Refresh] ボタンを使用して、再接続できます。



## Chapter 5: トラブルシューティング

本セクションでは、LifeKeeper Apache Web Server リソース階層の作成 / 拡張中、およびリソースの削除 / リストア中に表示される可能性のあるメッセージの一覧を示します。また、必要に応じて、エラーの原因や、エラー状態を解決するために必要な処置について補足説明します。他のSPSコンポーネントからメッセージが送出されることもあります。その場合は、メッセージカタログ(弊社のテクニカルドキュメンテーションサイトの「エラーコードの検索」内)を参照してください。メッセージカタログには、操作、管理、GUIなど、SIOS Protection Suite for Linuxを使用しているときに表示される可能性がある、すべてのエラーコードの一覧を示します。また、エラーコードの原因に関する補足説明や、問題解決のために必要な処置についても、必要に応じて記載します。この一覧から、受信したエラーコードを検索できます。また、該当するSPSコンポーネントの個別のメッセージカタログに直接アクセスすることもできます。

本セクションのメッセージは、以下のトピックに分類されています。

- [階層の作成](#)
- [階層の拡張](#)
- [階層の削除、リストア、リカバリ](#)

### 階層作成時のエラー

Apache階層の作成時に表示されるエラーメッセージを、説明とともに以下に示します。LifeKeeper Core および他のRecovery Kitが表示するエラーメッセージはこのガイドには記載していません。ここに記載されている問題を解決するために、階層の作成を中断できます。そして中断した時点から、Apacheの設定に必要な新しいLifeKeeperリソースの作成も含めて、階層の作成を続行できます。

### Web Server バイナリの位置検証時

```
"Error: valid_httpd_path: Must specify absolute path to httpd executable."
```

有効な Apache httpd 実行可能ファイルの絶対パスを省略せずに入力してください。

```
"Error: valid_httpd_path: File does not exist at path specified."
```

指定した場所に、有効な Apache httpd 実行可能ファイルが存在しません。

```
"Error: valid_httpd_path: Httpd failed to display Server version."
```

指定した場所にある httpd 実行可能ファイルが、Apache 標準の「サーバのバージョン」を表示しません。

```
"Error: valid_httpd_path: Incorrect version $MAJOR.$MINOR.$POINT of
Apache at $HTTPD_PATH.
```

指定した場所にある Apache httpd 実行可能ファイルが表示する「サーバのバージョン」が正しくありません。

## Web Server のルートディレクトリの検証時

```
"Error: valid_http_root: Cannot find Apache configuration file at $CONF_FILE."
```

指定したサーバルートディレクトリの conf/httpd.conf に、有効な Apache 設定ファイルが必要です。デフォルト設定を使用して Apache を SuSE にインストールする場合、httpd.conf 設定ファイルは ServerRoot のサブディレクトリ conf に配置されません。この場合は、設定ファイルを /etc/httpd/conf ディレクトリに配置し直す必要があります。

```
"Error: valid_http_root: Must specify absolute path to Apache server root directory."
```

サーバルートディレクトリの絶対パスを省略せずに入力してください。

```
"Error: valid_http_root: Apache instance at $HTTP_ROOT is already under LifeKeeper protection."
```

各インスタンスには、一意のサーバルートディレクトリが必要であり、設定ファイルが conf/httpd.conf に必要です。指定されたサーバルートディレクトリは、すでに他の Apache インスタンスが使用しています。

```
"Syntax error on line <line number> of <configuration file path>, etc..."
```

Apache 設定ファイルに構文エラーがあります。これらのエラーメッセージは、httpd -T コマンドを使用して \$CONF\_FILE の構文をチェックするときに表示されます。詳細については、表示されたエラーメッセージを確認してください。

```
"Error: valid_http_root: Since $HTTPD_PATH is shareable on $HTTPD_PATH_SHARED, $HTTP_ROOT
must be also."
```

httpd 実行可能ファイルが共有 / 共有可能ストレージに存在する場合は、サーバルートとすべての DocumentRoot ディレクトリも共有 / 共有可能ストレージに存在する必要があります。

```
"Error: valid_http_root: Since $HTTP_ROOT is shareable on $HTTP_ROOT_SHARED, all document root
directories must shareable on this same filesystem."
```

## Apache リソース階層の作成時

サーバのルートディレクトリが共有 / 共有可能ストレージに存在する場合は、すべての DocumentRoot ディレクトリも共有 / 共有可能ストレージに存在する必要があります。

"Error: http\_docs\_shared: Since one/more Apache document root directories are shareable on \$docs\_shared, \$curr\_root must be also.

いずれかの DocumentRoot ディレクトリが共有 / 共有可能ファイルシステムに存在する場合、すべての DocumentRoot ディレクトリが同じファイルシステムに存在する必要があります。

"Error: valid\_http\_root: Must include BindAddress or Listen directives for each Apache instance. Check the Apache configuration file at \$CONF\_FILE."

複数の Apache インスタンスを実行するには、各設定ファイルに BindAddress または Listen ディレクティブが必要です。詳細については、前のセクションにある[Apache Web Server の設定に関する考慮事項](#)を参照してください。

"Error: valid\_http\_root: Default IP address \* not allowed for LifeKeeper protection. Check the Apache configuration file at \$CONF\_FILE."

個々の Apache インスタンスについて、LifeKeeper が保護する IP アドレスを 1 つ以上指定する必要があります。

"Error: valid\_http\_root: A Listen directive is being used which specifies an IP address but no port. Check the Apache configuration file at \$CONF\_FILE."

Listen ディレクティブの正しい構文は、「Listen [IPaddress:] port number」です。これは、Apache では構文エラーとは見なされませんが、間違って解釈されます (IP アドレスの最初の数字がポート番号の指定と見なされる)。

"Error: valid\_http\_root: IP address \$ip is not LifeKeeper protected."

Apache の設定ファイルが、LifeKeeper の保護下に設定されていない IP アドレスまたはドメイン名を参照しています。これらの LifeKeeper IP アドレスのリソースをあらかじめ作成しておく必要があります。

## Apache リソース階層の作成時

"Error: Could not find IP resource for \$IP\_ADD on machine \$MACH."

このリソースを作成してから、Apache リソースの作成を続行する必要があります。

"Error: Create Apache file system hierarchy failure for filesystem \$FSNAME used by server root \$HTTP\_ROOT."

"Error: Failure bringing Apache Resource \$TAG into service on machine \$MACH."

## 階層拡張時のエラー

メッセージについては、Apache のエラーログを確認してください (デフォルトの位置は /var/log/httpd/error\_log ですが、他のログがリストされることがあります)。

この問題の原因の多くは、Apache の設定ファイルのエラーです。問題を修正した後、このリソースを手動で in service にすることができます。

```
"LifeKeeper: RESTORE: *ERROR* Apache: The instance is Password Protected."
```

LifeKeeper Apache Web Server Recovery Kit は、SSL 対応 Web サーバに使用される、パスワードで保護された秘密キーファイルをサポートすることができません。これは、Apache を起動するたびに手動操作が必要になり、再起動とフェイルオーバーの自動実行を妨げるからです。本書の「Apache Web Server の設定に関する考慮事項」セクションに、秘密キーファイルのパスワード保護を解除する方法 (SSLCertificateKeyFile ディレクティブで指定) が記載されています。このメッセージは、Apache とともに SSL モジュールを使用する環境にのみ適用されます。

## 階層拡張時のエラー

Apache 階層の拡張時に表示されるエラーメッセージを、説明とともに以下に示します。これらのエラーメッセージは、GUI が「Executing the pre-extend script...」というメッセージを表示して、新しいシステムに階層を拡張する前の階層の検証時に表示されることに注意してください。

各メッセージの前には、

```
「Error - canextend(template_server, tag, app_type/resource_type, target_server) -」のようなエラーメッセージが表示されます。
```

各メッセージの後には、

```
「Error - extmgr(template_server, tag, target_tag, target_server) -」のようなエラーメッセージが表示されます。
```

## Web Server バイナリの位置検証時

階層作成時のエラーの下にある、Web Server バイナリの位置検証時のエラーリストを参照してください。

## ターゲットシステムの Apache 設定ファイルの検証時

```
"Cannot find Apache configuration file at $CONF_FILE on $TARGET_SYS."
```

指定したサーバのルートディレクトリの conf/httpd.conf に、有効な Apache の設定ファイルが必要です。

```
"DocumentRoot directory "$doc" in $CONF_FILE on $TARGET_SYS was not found in the configuration file on $TEMPLATE_SYS."
```

または

```
"DocumentRoot directory "$doc" in $CONF_FILE on $TEMPLATE_SYS was not found in the configuration file on $TARGET_SYS."
```

ターゲットサーバとテンプレートサーバの設定ファイルを比較するときに、これらのいずれとも一致しない DocumentRoot ディレクトリが1つ以上見つかりました。表示されたエラーメッセージの詳細を確認して、2つの設定ファイルの違いを調べてください。DocumentRoot ディレクトリのパスを正しく入力しなかった場合、それぞれ

の設定ファイルには他方の設定ファイルに存在しないエントリがあるように見えるので、通常はこれらのエラーメッセージの両方が表示されます。

```
"IP:port combination "$ipp" in $CONF_FILE on $TARGET_SYS was not found in the configuration file on $TEMPLATE_SYS."
```

または

```
"IP:port combination "$ipp" in $CONF_FILE on $TEMPLATE_SYS was not found in the configuration file on $TARGET_SYS."
```

ターゲットサーバとテンプレートサーバの設定ファイルを比較するとき、一方のサーバで使用されるように設定されているが、他方のサーバでは設定されていない IP/ポートの組み合わせが1つ以上見つかりました。使用される IP/ポートの組み合わせは、Apache のさまざまな設定ディレクティブを使用して、IP アドレス、ポート、およびドメイン名で指定されます。比較されるのは、実際に使用される IP/ポートの組み合わせであり、IP/ポートの指定に使用されるディレクティブではありません。表示されたエラーメッセージの詳細を確認して、2つの設定ファイルの違いを調べてください。

```
"SSLCertificateKeyFile "$file" in $CONF_FILE on $SYS1 was not found in the configuration file on $SYS2."
```

ターゲットシステムの Apache の設定ファイル内で SSLCertificateKeyFile として指定されたファイル名が、テンプレートシステムで指定されているファイル名と一致しません。これらの設定は同一である必要があります。このメッセージは、Apache とともに SSL モジュールを使用する環境にのみ適用されます。

```
"Apache SSLCertificateKeyFile exists on $SYS1 but not on $SYS2."
```

Apache の設定ファイル内に指定された SSLCertificateKeyFile が、一方のシステムには存在しますが、他方のシステムには存在しません。このファイルは、両方のノードに存在する必要があります。このメッセージは、Apache とともに SSL モジュールを使用する環境にのみ適用されます。

```
"WARNING: PHP configuration file $PHP_CONFIG appears to be different on $SYS1 and $SYS2."
```

ターゲットシステムの PHP モジュールの設定ファイルが、テンプレートシステムの PHP モジュールの設定ファイルと異なります。両方のサーバの設定を確認して、同一にする必要があります。このメッセージは、Apache とともに PHP モジュールを使用する環境にのみ適用されます。

## ターゲットサーバでの Apache リソース階層の作成時

階層作成時のエラーの下にある、Apache リソース階層作成のエラーリストを参照してください。

## 階層リソースのリストア、削除、リカバリ時のメッセージとエラー

以下のトピックに含まれるエラーメッセージは、LifeKeeper のエラーログに出力されます。

このエラーログは、「lk\_log log」と入力すると表示できます。

## Apache リソースを In Service にするとき (リストア)

"LifeKeeper: RESTORE: APACHE: RESTORING \$TAG TO SERVICE START AT: <date>"

情報メッセージです。リストアの開始時刻を記録します。各リストアの開始時に記録されます。

"LifeKeeper: RESTORE APACHE RESOURCE \$TAG END err=\$err AT: <date>"

情報メッセージです。リストアの完了時刻を記録します。各リストアの完了時に記録されます。リストア時にエラーが発生した場合は、追加のメッセージがこれら2つのメッセージの間に記録され、err=\$errに表示される値がゼロ以外の値になります。

"Apache: No instance information found for Tag=\$TAG."

エラー: restore スクリプトに渡されたタグ値を持つインスタンスが定義されていないことを示します。LifeKeeper が既知のタグのみが In Service と Out of Service の動作の選択肢として使用できるため、GUI で発生する可能性は低いです。

"LifeKeeper: RESTORE: Apache: Tag=\$TAG already running."

情報メッセージです。表示されたインスタンスがすでに起動して動作中であることを示します。

"LifeKeeper: RESTORE: Apache: Existing processes terminated for ID=\$ID."

情報メッセージです。このインスタンスIDについて、既存のhttpd プロセスが実行中であることが検出されましたが、PidFile が見つからないか、無効です。このため、実行中のプロセスが強制終了されました。

"LifeKeeper: RESTORE: Apache: Invalid PidFile=\$PIDFILE has been deleted."

情報メッセージです。このインスタンスについて既存のPidFile が見つかりましたが、その内容が無効です。このため、PidFile が削除されました。

"LifeKeeper: RESTORE: Apache: Tag=\$TAG is being restarted."

情報メッセージです。表示されたインスタンスが部分的に動作中であるが、再起動が必要であることを示します。PidFile (親 httpd プロセスのプロセスIDを持つ) がまだ存在している場合、インスタンスはHUP信号を使用して再起動されます。PidFile が見つからない場合、インスタンスは完全に停止されてから再起動されます。

"LifeKeeper: RESTORE: \*ERROR\* Apache: Error in web server configuration file \$CONF\_FILE for instance \$ID."

"LifeKeeper: RESTORE: \*ERROR\* Apache: Execute the following command to check the syntax of this file:"

"LifeKeeper: RESTORE: \*ERROR\* Apache: \$HTTPD\_PATH -t -d \$SERVER\_ROOT -f \$CONF\_FILE."

## Apache リソースを Out of Service にするとき (削除)

インスタンスの起動前に、`httpd -t` オプションを使用して、設定ファイルの構文がチェックされます。`-d` オプションは、`ServerRoot` ディレクトリをチェックします。モジュールを使用するように Apache を設定した場合、モジュールに関するその他のオプションも表示されることがあります。階層作成時に検出された構文エラーは LifeKeeper GUI に表示されますが、その後には持ち込まれた構文エラーは、GUI や LifeKeeper のログには表示されません。以下のコマンドを手動で実行して、設定の不具合を特定する必要があります (該当する場合は、モジュールのオプションを追加する)。

```
$HTTPD_PATH -t -d $SERVER_ROOT -f $CONF_FILE
```

```
"LifeKeeper: RESTORE: *ERROR* Apache: Error starting web server instance $INSTANCE."
```

```
"LifeKeeper: RESTORE: *ERROR* Apache: Restore of tag $TAG failed."
```

```
"LifeKeeper: RESTORE: *ERROR* Apache: Examine the Apache error log at $ERROR_LOG"
```

```
"LifeKeeper: RESTORE: *ERROR* Apache: to determine the cause of the problem."
```

パラメータを指定した `httpd` デーモンの実行時にエラーが発生しました。使用している `httpd` 実行可能ファイル、設定ファイル、および一般設定に問題がないかどうかをチェックしてください。

```
"LifeKeeper: RESTORE: *ERROR* Apache: Web server instance $ID did not start correctly."
```

```
"LifeKeeper: RESTORE: *ERROR* Apache: Restore of tag $TAG failed."
```

```
"LifeKeeper: RESTORE: *ERROR* Apache: Examine the Apache error log at $ERROR_LOG"
```

```
"LifeKeeper: RESTORE: *ERROR* Apache: to determine the cause of the problem."
```

多くの場合、Web サイトが予測どおりに応答しなくとも、`httpd` デーモンが起動したように見えることがあることに注意してください。`restore` スクリプトは、使用する IP/ポートの組み合わせをすべてチェックし、設定されたすべてのサイトがフルに機能することを確認します。設定されたすべてのサイトがフルに機能しない場合、このメッセージが出力され、リストアは失敗します。

サイトが Out of Service 状態になっていても、1 つ以上の `httpd` プロセスが動作し続けている場合があることに注意してください (1 つ以上の Web サイトが動作している可能性があり、それらの Web サイトを強制終了したくないため、意図的にプロセスの動作を継続させています)。できるだけ早く問題を解決し、インスタンスを In Service にする必要があります。問題が解決されない場合、LifeKeeper が最終的にインスタンスのリカバリおよびリストアを自動的に試行します。LifeKeeper によるリカバリとリストアができない場合は、階層が別のサーバにフェイルオーバーします。

## Apache リソースを Out of Service にするとき (削除)

```
"LifeKeeper: REMOVE: APACHE: REMOVE $TAG FROM SERVICE START AT: <date>"
```

情報メッセージです。削除の開始時刻を記録します。各削除の開始時に記録されます。

```
"LifeKeeper: REMOVE APACHE RESOURCE $TAG END err=$err AT: <date>"
```

情報メッセージです。削除の完了時刻を記録します。各削除の完了時に記録されます。

削除時にエラーが発生した場合は、追加のメッセージがこれら2つのメッセージの間に記録され、err=\$errに表示される値がゼロ以外の値になります。

```
"LifeKeeper: REMOVE: *WARNING* APACHE: Error attempting to kill parent process for
INSTANCE=$INSTANCE."
```

親 httpd プロセスを強制終了しようとするエラーがありました (親プロセスのプロセス ID は Pidfile に格納される)。

```
"LifeKeeper: REMOVE: *ERROR* APACHE: Error attempting to kill all processes for
INSTANCE=$INSTANCE."
```

親 httpd プロセスが正常に強制終了されたように見えますが、このインスタンスの1つ以上のプロセスがまだ動作中です。通常、削除動作では、このインスタンスのすべてのプロセスを強制終了できます。強制終了できない場合、このメッセージが出力され、削除は失敗します。

## Apache リソースを In Service に戻すとき (リカバリ)

LifeKeeper Core は、Apache の quickCheck スクリプトを実行して、ローカルサーバで In Service 状態にあるすべての Apache インスタンスの健全性を定期的にチェックします。Apache の quickCheck スクリプトは、リストア時と削除時にインスタンスの状態をチェックするものと同じスクリプトを使用して、Web サイトをチェックします。インスタンスが完全に動作していない場合、recover スクリプトが実行され、インスタンスの再起動が試行されます。これは、以下に示す最初のメッセージを記録し、restore を起動します。restore スクリプトの結果に応じて、最後のエラーメッセージまたは成功メッセージを出力し、restore と同じ結果を返します。restore/recover が失敗した場合、このインスタンスは別のサーバにフェイルオーバーされます。

```
"LifeKeeper: RECOVER: APACHE: Invoking restore for Apache instance "$ID" at: <date>"
```

```
"LifeKeeper: RECOVER: APACHE: Restore for Apache instance $ID returned error $RET at: <date>"
```

```
"LifeKeeper: RECOVER: APACHE: Restore for Apache instance $ID successful at: <date>"
```