



SteelEye Protection Suite for Windows

v7.4.2

Technical Documentation

August 2012

This document and the information herein is the property of SIOS Technology Corp. (previously known as SteelEye® Technology, Inc.) and all unauthorized use and reproduction is prohibited. SIOS Technology Corp. makes no warranties with respect to the contents of this document and reserves the right to revise this publication and make changes to the products described herein without prior notification. It is the policy of SIOS Technology Corp. to improve products as new technology, components and software become available. SIOS Technology Corp., therefore, reserves the right to change specifications without prior notice.

LifeKeeper, SteelEye and SteelEye DataKeeper are registered trademarks of SIOS Technology Corp.

Other brand and product names used herein are for identification purposes only and may be trademarks of their respective companies.

To maintain the quality of our publications, we welcome your comments on the accuracy, clarity, organization, and value of this document.

Address correspondence to:
ip@us.sios.com

Copyright © 2012
By SIOS Technology Corp.
San Mateo, CA U.S.A.
All rights reserved

目次

Chapter 1: SteelEye Protection Suite for Windows について	1
SPS for Windowsが統合するコンポーネント	1
保護対象リソース	1
SPS Core ソフトウェア	2
SPS Core リカバリキット	3
SPS Microsoft SQL Server Recovery Kit	4
コミュニケーションパスの概要	4
コミュニケーションパスの種類	5
共有ディスクコミュニケーションパスの詳細	5
SPS ハートビート	6
ハートビート間隔	6
セーフティチェック	6
Chapter 2: インストールガイド	9
Chapter 3: 構成	10
SPS の設定手順	10
アクティブ/アクティブ構成	10
アクティブ/スタンバイ構成	11
インテリジェントスイッチバックと自動スイッチバック	12
SPS の設定	13
共通ハードウェアコンポーネント	13
システムのグループ化	14
Chapter 4: SPS 管理の概要	16
GUIによる管理作業	16
サーバプロパティの編集	16
サーバのシャットダウン方法の設定	17

サーバプロパティ	17
[コミュニケーションパス] タブ	20
[リソース] タブ	22
自動フェイルオーバを無効にする	22
コミュニケーションパスの作成	23
構成に関する注意事項	23
コミュニケーションパスの作成	24
コミュニケーションパスの確認	25
コミュニケーションパスの削除	25
リソース階層に関連する作業	26
リソース階層の作成	26
SPS のアプリケーションリソース階層	26
Microsoft SQL Server Recovery Kit	27
ファイル共有リソース階層の作成	27
ファイル共有リソースの条件	27
ファイル共有リソースの作成	27
Generic Application リソース階層の作成	28
リソース階層を作成する前に	28
リソース階層の作成	29
LAN Manager リソース階層の作成	30
ボリュームリソース階層の作成	31
ボリュームリソース作成後の作業	32
IP リソース階層の作成	32
IP ローカルリカバリの要件	33
IP ローカルリカバリのシナリオ	35
リソース優先度の編集	35
[上へ] ボタンと[下へ] ボタンの使用	36
優先度の編集	37
変更内容の適用	37
不完全なリソース優先度の変更	37

階層の一貫性を復元する	37
リソースプロパティの編集	38
リソース階層の拡張	38
DNS リソース階層の拡張	39
ファイル共有リソース階層の拡張	39
Generic Application リソース階層の拡張	39
LAN Manager リソース階層の拡張	39
ボリュームリソース階層の拡張	39
IP リソース階層の拡張	40
リソース階層の拡張解除	42
リソース依存関係の追加	42
リソース依存関係の削除	43
全サーバのリソース階層の削除	43
マニュアルページ	43
LCD - その他のLCD プログラム	43
構文	43
説明	44
終了コード	44
lcdrp	44
lcdrecover	44
lcdremexec	46
lcdsync	46
lkstart	47
lkstop	47
LCDI アプリケーション	47
構文	47
説明	48
終了コード	48
app_create	48
app_list	48

app_remove	49
LCDI インスタンス	49
構文	49
説明	49
終了コード	50
ins_list	50
初期化の方法	52
初期状態	52
ins_create	52
ins_gettag	53
ins_remove	53
ins_setas	54
ins_setchkint	54
ins_setin	54
ins_setit	55
ins_setlocalrecover	55
ins_setst	55
LCDI 関係	56
構文	56
説明	56
終了コード	56
dep_create	57
dep_list	57
dep_remove	58
eqv_create	58
eqv_list	58
eqv_remove	59
LCDI リソースタイプ	59
構文	59
説明	59

終了コード	60
typ_create	60
typ_list	60
typ_remove	61
LCDI システム	61
構文	61
説明	61
終了コード	61
sys_create	62
sys_getds	62
sys_getst	62
sys_list	62
sys_remove	62
LifeKeeper のフラグ	63
flg_create	63
flg_list	63
flg_remove	63
flg_test	64
LCDI フラグ	64
構文	64
説明	64
lk_chg_value	66
名前	66
構文	66
説明	66
例	66
終了コード	67
注記	67
ファイル	67
lk_err	68

構文	68
説明	68
perform_action	68
構文	68
説明	69
例	71
終了コード	71
sendevent	71
構文	71
説明	72
出力	73
終了コード	74
volume	74
構文	74
説明	74
例	75
終了コード	75
IP ローカルリカバリ	75
SNMP による SPS イベント転送の概要	76
SPS イベントテーブル	76
Chapter 5: ユーザガイド	80
LifeKeeper GUI	81
LifeKeeper グラフィカルユーザインターフェース	81
GUI の概要	81
GUI サーバ	81
GUI クライアント	81
LifeKeeper GUI クライアントの起動	82
Web クライアントの起動	82
アプリケーションクライアントの起動	82
GUI クライアントの終了	82

ステータステーブル	82
プロパティパネル	83
アウトプットパネル	83
メッセージバー	83
ツールバー	84
SteelEye Protection Suite for Windows ツールバー	84
Global Toolbar	84
リソースコンテキストツールバー	85
サーバコンテキストツールバー	86
メニュー	87
SteelEye Protection Suite for Windows メニュー	87
リソースコンテキストメニュー	87
サーバコンテキストメニュー	88
[ファイル] メニュー	89
[編集] メニュー - リソース	89
[編集] メニュー - サーバ	90
[表示] メニュー	91
ヘルプ] メニュー	93
LifeKeeper GUI サーバコンポーネントおよびクライアントコンポーネント	93
SPS クライアントの実行	94
ブラウザのセキュリティレベルの設定	95
Internet Explorer	96
Mozilla Firefox	96
Netscape Navigator と Netscape Communicator	96
SPS サーバ上の GUI アプリケーションの実行	96
LifeKeeper GUI ユーザアカウント	97
ログイン	97
GUI ユーザの設定	97
共通タスク	98
クラスタへの接続	98

クラスタからの切断	99
接続先 サーバの表示	99
サーバステータスの表示	100
サーバログファイルの表示	100
サーバプロパティの表示	100
リソースタグとID の表示	101
リソースステータスの表示	101
サーバリソースのステータス	101
グローバルリソースのステータス	102
リソースプロパティの表示	102
メッセージ履歴の表示	102
メッセージ履歴の読み方	103
リソース階層ツリーの展開と縮小	103
オペレータタスク	104
リソースを起動す	104
リソースを停止する	104
ボリュームリソースの起動と停止	105
ボリューム削除時のプログラムとサービスの停止と再起動	105
ボリューム復元時のプログラムとサービスの停止と再起動	107
ボリュームシャド ウコピー (VSS)	108
DataKeeper/SPS ボリュームでボリュームシャド ウコピー (VSS) を使用する	108
共有 SCSI ボリュームのボリュームロック	108
詳細項目	109
LifeKeeper 構成 データベース (LCD)	109
LCD のディレクトリ構造	109
LCD ディレクトリの構造図	110
LCD 構成データ	110
依存関係に関する情報	111
リソースステータスに関する情報	111
サーバ間のイクイバレンスに関する情報	111

LCD リソースタイプ	111
リソースサブディレクトリ	111
リソース操作	112
LCDI コマンド	113
LifeKeeper 通信 マネージャ (LCM)	113
通信 ステータス情報	113
データレプリケーション	113
レプリケーションボリュームリソースの監視	114
レプリケーションの設定	116
レプリケーションボリュームの操作	117
スプリットブレインとは	119
スプリットブレインリカバリ	120
データリワインドの概要	123
リワインドの設定	124
レプリケーションボリュームリソースのリワインド操作	125
Chapter 6: DataKeeper	128
Chapter 7: トラブルシューティング	129
アプレットのトラブルシューティング	129
説明	129
LifeKeeper を停止するときに Win2008 で CYGWIN がクラッシュする場合がある	129
現象	129
解決策	130
コマンドプロンプトから SPS コマンドを実行しようとしたときにエラーが発生する	130
現象	130
解決策	130
ファイアウォール	130
現象	130
解決策	131
GUI エラーメッセージ	131
説明	131

GUI ネットワーク関連 - サーバへの初期接続の失敗 (エラー 117)	132
現象	132
解決策	133
GUI ネットワーク関連 - Windows プラットフォーム上の長期接続遅延	133
現象	133
解決策	133
サンマイクロシステムズ社の FAQ より	133
GUI ネットワーク関連 - 接続試行時に生成される NoRouteToHostException メッセージ ..	133
現象	133
解決策	134
GUI ネットワーク関連 - 接続試行時に生成される不明なホスト例外メッセージ	134
現象	134
解決策	134
Windows の場合	135
Linux の場合	135
GUI サーバのトラブルシューティング	136
現象	136
解決策	136
ヘルスチェックタイムアウト	137
不完全なリソースの作成	138
説明	138
インストール - アクセス拒否	138
現象	138
解決策	138
IP リソース作成問題	139
現象	139
ワークアラウンド	139
Java の署名があるコードとないコードの混在の警告	139
現象	139
解決策	140

参照リストに LANMAN 名が2回表示される場合がある	140
現象	140
解決策	140
ライセンス-ライセンスされたリカバリキットリソースが起動しない	141
現象	141
解決策	141
ライセンス-ライセンスキーが見つからない	141
現象	141
解決策	141
LifeKeeper Web クライアントがロックされる場合がある	141
現象	141
解決策	141
SPS Web クライアントがロックされる場合がある-既存の階層が複数変更された	142
現象	142
解決策	142
新しい評価ライセンスキーのエラー	142
現象	142
解決策	142
1 x 1 構成における SPS のサーバ障害からの復旧	142
推奨されるアクション:	142
サービス休止中の階層のリカバリ	150
説明	150
リカバリキットのアンインストール時に Remove がハングする	150
現象	150
解決策	150
複製ボリュームのスイッチオーバーの失敗	151
説明	151
リソースタグ名の制限	152
タグ名の長さ	152
有効な "特殊" 文字	152

無効な文字	152
リストアおよびヘルスチェックアカウントの障害	152
現象	152
解決策	152
SQL 2005 および SQL 2008	152
現象	152
解決策	153
SQL Server Reporting Services (MSSQLSERVER)	153
現象	153
解決策	153
2 サーバクラスタの問題	153
現象	153
解決策	153
不明なユーザ名または誤ったパスワード	154
アクセス拒否: 不明なユーザ名または誤ったパスワード	154
現象	154
解決策	154
Web クライアントのトラブルシューティング	154
背景	154
回答	154
Win2008 - IIS リソース階層作成エラー	155
現象	155
解決策	155
Index	156

Chapter 1: SteelEye Protection Suite for Windows について

SteelEye Protection Suite (SPS) for Windows は、高可用性クラスタリングおよびデータ複製機能を統合し、ミッションクリティカルなデータおよびアプリケーションを保護します。さらに SPS は SQL サーバ、オラクル、Exchange のためのアプリケーション・アウェア・エージェント機能を提供する SIOS のアプリケーション・リカバリ・キットによって、より正確な情報に基づいた復旧の決断をもたらします。

SPS for Windowsが統合するコンポーネント

LifeKeeper for Windows は、データ、アプリケーション、通信リソースの高可用性を実現する耐障害ソフトウェアです。LifeKeeper では専用の耐障害ハードウェアは不要です。複数のシステムをグループ化して、サイト構成情報を定義するだけで動作します。クラスタを構築すると、LifeKeeper は障害を自動的に検出して回復します。

障害が発生すると、LifeKeeper は保護対象リソースをバックアップシステムに移動します。フェイルオーバーの際、システムはわずかに中断しますが、フェイルオーバーリカバリが完了すると、LifeKeeper はバックアップシステムでリソースを自動的に復元します。

SteelEye DataKeeper は、最適化されたホストベースのレプリケーションソリューションで、ネットワーク経由でソースサーバーから一つ以上のターゲットサーバーへ可能な限りすばやく、そして効率的にデータのレプリケーションを行います。

オプションの Microsoft SQL Server Recovery Kit は SQL Server に対する granular visibility を提供します。システムがただ起動しているかどうかだけでなく、適切な動作をしているかを確認することができます。

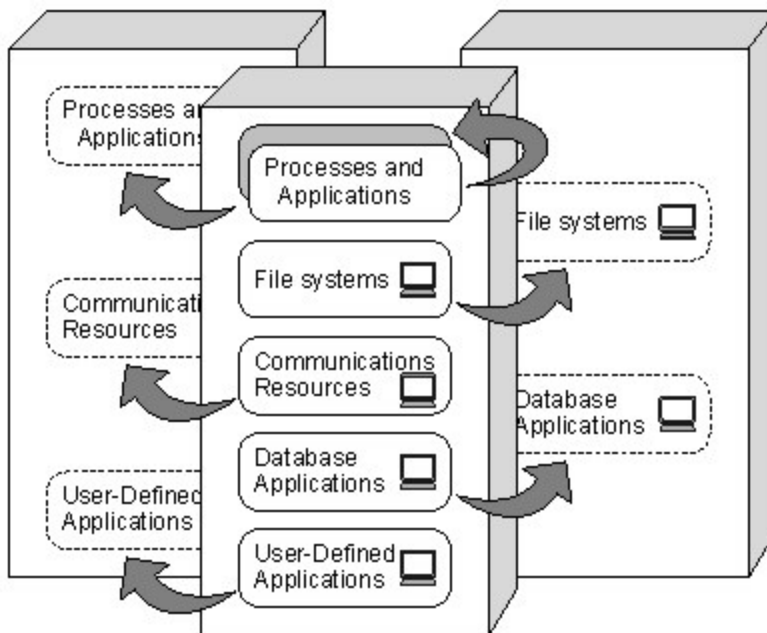
保護対象リソース

SPS 製品ファミリには、多様なシステムリソースを保護するソフトウェアが用意されています。SPS がサポートするアプリケーションとリソースは次のとおりです。:

- **ボリューム**。ボリュームリソースタイプを使用すると、作成したリソースを格納したディスクボリュームの LifeKeeper 階層にリソースを作成して、共有 SCSI 周辺機器または複製ボリューム上でデータとアプリケーションを保護できます。
- **ファイル共有**。LAN Manager Recovery Kit は、共有ドライブ上のフォルダとディレクトリを保護します。
- **コンピュータの別名**。LAN Manager Recovery Kit は、NetBEUI を介してサーバと通信するアプリケーションのコンピュータ別名を自動的にフェイルオーバーします。
- **通信リソース**。

- IP Recovery Kit は、IP アドレスを自動的に切り替えるリソースを作成します。
- DNS Recovery Kit は、DNS の A レコードと PTR レコードを更新します。
- **データベースアプリケーション**。SPS は、Microsoft SQL Server に対してオプションのリカバリキットを提供します。
- **Generic applications**。Generic Application リカバリキットは、専用のリカバリキットがないアプリケーションのリソースを作成します。

SPS は、さまざまなリソースタイプについて、N 方向のリカバリをサポートします。N 方向のリカバリでは、複数のリソースを複数のクラスターサーバにフェイルオーバーできます。



SteelEye Protection Suite のコンポーネントについては [SPS Core ソフトウェア](#) を参照してください。

SPS Core ソフトウェア

SteelEye Protection Suite for Windows Core は、LifeKeeper と基本的な LifeKeeper ソフトウェアパッケージ、DataKeeper とリカバリキットを含んでいます。

- **LifeKeeper** - LifeKeeper for Windows core は基本的な LifeKeeper ソフトウェアパッケージとリカバリキットを含んでいます。
 - **LifeKeeper 構成データベース (LCD)** LifeKeeper が保護するリソースに関する情報が格納されています。リソースインスタンス、依存関係、共有イクイバレンシ、リカバリ方向、LifeKeeper 操作フラグなどの情報が格納されています。LCD データはシステムの共有メモリにキャッシュされた後、ファイルに格納されるため、システムを再起動しても、設定データは保持されます。
 - **LCD インターフェース (LCDI)** - 構成データベース (LCD) にクエリを発行し、LCD

内のデータを取得して変更できます。LCDI はアプリケーションリカバリキット (ARK) で起動し、リソース状態や各種情報を取得します。

- **LifeKeeper 通信マネージャ(LCM)** - クラスタサーバのステータスを確認します。LifeKeeper によるプロセス間通信 (ローカルとリモート) にも使用できます。クラスタサーバのコミュニケーションパスで LCM 通信が完全に停止すると、LifeKeeper はサーバに障害が発生したと判断します。
- **LifeKeeper アラームインターフェース** - イベントをトリガする基盤を提供します。LifeKeeper で保護するリソースに障害が発生すると、アプリケーションのデーモンは、sendevent プログラムを呼び出します。sendevent プログラムは LCD と通信を行い、リカバリに使用できるスクリプトがあるかどうかを判断します。
- **LifeKeeper リカバリアクション制御インターフェース(LRACI)** - リソースに実行する適切なリカバリスクリプトを決定し、適切な restore/remove スクリプトを起動します。
- **LifeKeeper GUI** - クライアント/サーバアプリケーションです。LifeKeeper と設定データを操作するグラフィカルな管理インターフェースを提供します。LifeKeeper GUI クライアントには、スタンドアロンのアプリケーションと Web ブラウザから起動する Java アプレットがあります。
- **DataKeeper** - DataKeeper パッケージは、複数サーバおよびデータセンターを経由して Windows サーバのデータの同期を保ちながらリアルタイムのデータ複製機能を提供します。

SPS Core リカバリキット

Core リカバリキットは、障害検出とリカバリのあらゆるしくみを提供します。また、ボリューム(ドライブ名)、ファイル共有、通信リソース、およびマイクロソフトの Internet Information Server を SPS で保護するために必要な定義を提供します。

- **Volume Recovery Kit** - リソースを作成して、共有ドライブまたはミラー化したドライブ全体 (例: K: ドライブ) を保護します。
- **LAN Manager Recovery Kit** - コンピュータの別名と Windows ファイル共有リストを自動的にリストアップします。LAN Manager リソースでは、NetBEUI 経由でサーバと通信するアプリケーションの「切り替え可能」なコンピュータ名 やシステム名を作成できます。また、ファイル共有リカバリコンポーネントを含みます。
- **IP Recovery Kit** - 障害が発生したプライマリサーバからバックアップサーバに「切り替え可能」な IP アドレスを移動します。切り替え可能な IP アドレスとは、サーバ間で切り替える仮想 IP アドレスを指します。ネットワークインターフェースカードの実アドレスとは異なります。SPS で保護するアプリケーションには、切り替え可能な IP アドレスを登録します。プライマリサーバに障害が発生すると、この IP アドレスがバックアップサーバに移動します。IP Recovery Kit は IP リソースのローカルリカバリ機能も提供します。
- **DNS Recovery Kit** - DNS プライマリサーバまたは別名の A レコードと PTR レコードを更新します。DNS リソースを使用した場合、フェイルオーバーや切り替えが発生すると、プライマリサーバまたは別名の A レコードと PTR レコード (存在する場合) が、バックアップサーバの IP アドレスで更新されます。

- **Microsoft IIS Recovery Kit** - Microsoft Internet Information Services (IIS) の Web、FTP、SMTP サービスを保護します。Web サーバの稼働状態を継続的に監視し、問題を検出すると、障害が発生した Web サーバをバックアップサーバに自動的にフェイルオーバーします。
- **Generic Application Recovery Kit** - 専用のリカバリキットがないアプリケーションのリソースを作成します。

SPS Microsoft SQL Server Recovery Kit

SPS Microsoft SQL Server アプリケーションリカバリキットは、SPS が Microsoft SQL ベースのデータベースアプリケーションを管理、制御するためのツールとユーティリティを含んでいます。インストールされると、SPS はアプリケーションの状態を監視したり、アプリケーションに障害が発生した場合自動的に復旧させることが可能になります。

このキットはSPS で保護するための特別な制約やアプリケーションに対する変更を必要としません。

このキットはMicrosoft SQL Server インスタンスの自動スイッチオーバーに対してリソースのメニュー選択方式を提供します。簡単にリソース階層を作成できるので、リカバリ処理はSQL Serverが使用する名前付きパイプもしくはIP ソケットリソースのようなデータベースへのアクセスに使用するすべてのディスクリソースを組み込むことができます。

詳細情報については Microsoft SQL Server Recovery Kit Technical Documentationを参照してください。

注記: 製品要件およびFAQ 情報についてはOS指定のリリースノートで確認することができます。

コミュニケーションパスの概要

SPS はリソースの利用状況を監視し、サーバに設定したコミュニケーションパスを使用してフェイルオーバーを実行します。SPS の障害検出機能とリソースリカバリ機能を利用するには、サーバ間の通信をアクティブに保つことが重要です。そのためには、異なるプロトコルを使用して複数のコミュニケーションパスを定義し、通信回線の障害に起因するシステムのフェイルオーバーを回避する必要があります。

SPS でリソースやリソース階層を定義する前に、各サーバにコミュニケーションパスを定義します。SPS は定義したコミュニケーションパスを介して、ノード間のリソース定義と状態を調整し、障害の検出とリカバリを実行します。

[コミュニケーションパスの種類](#)では、コミュニケーションパスに関する以下の項目について説明します。

- **コミュニケーションパスの種類:** 2 台のサーバで構成するクラスターでは、2 種類のコミュニケーションパス (TCP/IP、共有ディスク) を使用できます。3 台以上のサーバで構成するクラスターでは、TCP/IP コミュニケーションパスだけを使用できます。
- **SPS ハートビート:** コミュニケーションパスではハートビートが重要な役割を果たします。ハートビートは、接続先サーバの死活を監視するために、一定の間隔で送信される信号です。
- **セーフティチェック:** すべてのコミュニケーションパスが停止すると、SPS はセーフティチェックを実行してシステム障害を検査した後、フェイルオーバーを実行します。

コミュニケーションパスの種類

SPSには3種類のコミュニケーションパスがあり、それぞれ異なるプロトコルを使用して、コミュニケーションパスを冗長化できます。同じ媒体に同じ種類のコミュニケーションパスを複数定義しても意味がありませんが、異なるプロトコルまたは異なる媒体を使用してパスを冗長化すると、コミュニケーションパス障害に起因するフェイルオーバを回避できます。

コミュニケーションパスには優先順位を設定します。SPSは、優先順位を参照して、ノード間の通信を行います。ハートビート信号はアクティブなコミュニケーションパスすべてに送信されます。SPSがサポートするコミュニケーションパス、およびコミュニケーションパスに割り当てたデフォルトの優先順位の範囲は次のとおりです。

- **TCP/IP (ソケット):** LAN コミュニケーションパスは最も高い優先順位を持ちます。ソケットコミュニケーションパスのデフォルトの優先順位の範囲は1～30です。サーバペアに異なるアダプタを使用して、複数のLAN コミュニケーションパスを定義すると、フェイルオーバの誤動作を回避できます。

注記: 2本のTCP/IP コミュニケーションパスを設定した場合、SPSは優先順位の高いコミュニケーションパスを使用して、リソースのステータス情報を交換します。

- **共有ディスク:** SPSでは、クラスタを構成するサーバペアが通信ロケーションとして使用する共有ディスク上にraw ディスクパーティションを定義できます。共有ディスクコミュニケーションパスは、2台のサーバが同じドライブ名を使用し、ドライブ名は同じディスクパーティションを指す必要があります。通常の場合、ディスクパーティションは1MB程度で十分です。共有ディスクコミュニケーションパスのデフォルトの優先順位は61～99です。共有ディスクコミュニケーションパスは、2サーバ構成だけで使用できます。

注記: 共有ディスクコミュニケーションパスは、クラスタサーバの死活状態を監視します。そのため、TCP/IP コミュニケーションパスがダウンすると、共有ディスクコミュニケーションパス、2本目のTCP/IP コミュニケーションパスがアクティブでも、GUIには、通信先サーバの階層がUNKNOWN(不明)と表示されます。

共有ディスクコミュニケーションパスの詳細

共有ディスクコミュニケーションパスは、他のコミュニケーションパスがすべて停止した場合の最後の手段となります。共有ディスクコミュニケーションパスが停止すると、少なくとも1台のサーバがストレージのサブシステムにアクセスできなくなるため、2台のサーバが同じディスクリソースに同時にアクセスする「スプリットブレイン」を回避できます。

注意事項

- クラスタの2つのサーバ間で共有ディスクコミュニケーションパスは必須ではありません。
- JBOD またはホストベースのRAID で共有ディスクコミュニケーションパスを使用する場合は、クラスタメンバがシャットダウンするか、サービスを停止した際のコミュニケーションパスの信頼性を事前にテストしてください。JBOD またはホストベースのRAID を使用する構成では、クラスタメンバがダウンすると、コミュニケーションパスが切断されて、フェイルオーバできないことがあります。

SPS ハートビート

SPS は、ハートビートを使用して、主要な障害を検出します。ハートビートとは、サーバペアのコミュニケーションパスに一定間隔で送信される信号を指します。サーバは、ハートビートを通じて、通信先サーバが稼働していることを確認します。コミュニケーションパスには、[ハートビート間隔](#)を秒単位で定義します。また、ハートビート信号が連続して欠落した場合、何回目かでコミュニケーションパスの停止と判断するか、その回数を指定します。

コミュニケーションパスが停止すると、SPS は優先順位が次に高いコミュニケーションパスを使用して、ノード間通信を開始します。すべてのコミュニケーションパスでハートビート信号を受信できない場合、SPS は [セーフティチェック](#) を実行して、フェイルオーバーカバリを実行するか判断します。

ハートビート間隔

SPS のハートビート間隔とは、サーバの死活状態を確認するハートビート信号を送信する間隔 (秒単位) を指します。デフォルト値の6秒を推奨します。

- ハートビート間隔を最小の4秒に設定する場合は、コミュニケーションパスをプライベートネットワーク上で設定した後、十分にテストしてください。ハートビート間隔を5秒未満にすると、ネットワークの中断によって、フェイルオーバーが誤動作することがあります。
- ハートビート間隔はハートビート最大欠落数と関係します。ハートビート最大欠落数は、デフォルトの5回を推奨します。ハートビート最大欠落数を最小値 (3または4) に設定すると、フェイルオーバーが誤動作することがあります。本番環境で十分にテストしてください。
- ハートビート間隔とハートビート最大欠落数を大きな値に設定すると、SPS の障害検出機能が低下します。

セーフティチェック

すべてのコミュニケーションパスが停止すると、SPS は、クラスタサーバが完全に停止したと判断し、フェイルオーバーを実行します。ただし、その際、SPS は障害がコミュニケーションパスではなく、サーバに発生したことを確認するセーフティチェックを実行します。

セーフティチェックでは、LAN Manager を使用して、サーバが稼働しているかどうかを確認します。その結果に応じて、次のいずれかのイベントが発生します。

- **システムが稼働している場合**：セーフティチェックにより、サーバがネットワークに接続していることを確認すると、LifeKeeper はフェイルオーバーを中止して、次のメッセージを LifeKeeper イベントログに出力します。

```
SAFETY CHECK FAILED: COMM_DOWN ABORTED
```

- **システムが停止している場合**：サーバが所定のタイムアウト (デフォルトは8秒) 以内に応答を返さないと、LifeKeeper は、サーバが停止していると判断して、フェイルオーバーを実行します。

SPS は、すべてのコミュニケーションパスが停止すると、セーフティチェックを実行します。セーフティチェックでシステムの稼働を確認すると、SPS はフェイルオーバーを中止します。また、LifeKeeper が中断したフェイルオーバーを再開するのは、次のすべてのイベント (1、2、3) が順番に発生したときになります。

1. 少なくとも 1 本のコミュニケーションパスが**ALIVE**に戻る。
2. すべてのコミュニケーションパスが再び**全断**する。
3. セーフティチェックが実行され、対向ノードの稼働が**確認**できない。

Chapter 2: インストールガイド

SteelEye Protection Suite for Windows のインストールに関する情報は、SIOSのテクニカルドキュメンテーションサイトの ***SPS for Windows Installation Guide*** を参照してください。:

<http://jpdocs.us.sios.com/#SPS4W>

Chapter 3: 構成

LifeKeeper 環境をインストールしたら、クラスタ内のサーバごとに LifeKeeper ソフトウェアを構成できます。このセクションでは構成の方法について説明します。

SPS の設定手順

次の手順に従って設定します。リンクをクリックすると詳細な説明が表示されます。以下の作業は、クラスタ内の各サーバに実行します。

1. **[コントロールパネル]**を開き、**[管理ツール]**の**[サービス]**で LifeKeeper の各種サービスが実行されていることを確認します。LifeKeeper サービスと LifeKeeper External Interfaces サービスの両方が稼働していることを確認します。2つのサービスが稼働していない場合は、必要なサービスを起動してください。

詳細については、[LifeKeeperの起動と停止](#)を参照してください。

2. SPSサーバの管理者権限を持つユーザは、サーバ上でアプリケーションクライアントを起動できます。**[スタート]**メニューをクリックし、**[プログラム]** -> **[LifeKeeper]** -> **[LifeKeeper (Admin Only)]**を選択してください。

アプリケーションを読み込むと、LifeKeeper GUI が開き、**[クラスタへの接続]**ダイアログボックスが開きます。接続先サーバの名前を**[サーバ名]**に入力し、**ログイン名**と**パスワード**を入力してください。

3. [コミュニケーションパスを作成します](#)。SPS による保護をアクティブにする前に、LifeKeeper のコミュニケーションパス(ハートビート)を定義してください。
4. [サーバーのシャットダウン方法](#)を設定します。システムをシャットダウンする際、リソースを切り替えるか指定します。
5. 以上の操作を実行すると、SPS でアプリケーションを保護する準備が完了します。次の手順は、SPSリカバリキットの種類によって異なります。
 - Core Recovery Kit を使用する場合は、[ボリュームリソース階層の作成](#)、[DNS リソース階層の作成](#)、[IP リソース階層の作成](#)、[ファイル共有リソース階層の作成](#)、[LAN Manager リソース階層の作成](#)、または [Generic Application リソース階層の作成](#)を参照してください。
 - オプションのMicrosoft SQL Server Recovery Kit,を使用する場合は、Microsoft SQL Server Recovery Kit 管理ガイドセクションのリソース階層の作成と拡張に関する説明を参照してください。

アクティブ/アクティブ構成

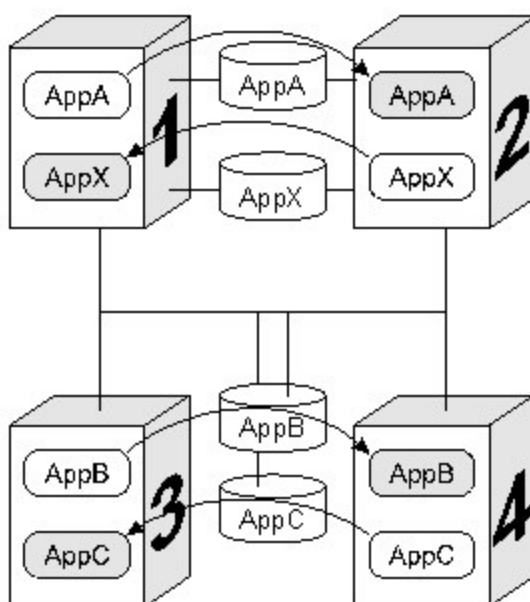
アクティブ/アクティブ構成では、各サーバがサービスを起動します。また、各サーバは、他のノードのバック

アップとしても動作します。

次の例では、4台のサーバが2組のアクティブ/アクティブペアを構成しています。Server 1はAppAを処理し、Server 2で稼働するAppXもバックアップしています。Server 2も同様です。Server 2はAppXを処理し、Server 1で稼働するAppAもバックアップしています。Server 3と4もアクティブ/アクティブ構成です。

Server 1と2およびServer 3と4はよく似た構成ですが、決定的な違いがあります。AppAアプリケーションとAppXアプリケーションは、Server 1とServer 2だけがサービスを提供しています。そのため、それ以外のサーバは、共有リソースにアクセスできません。

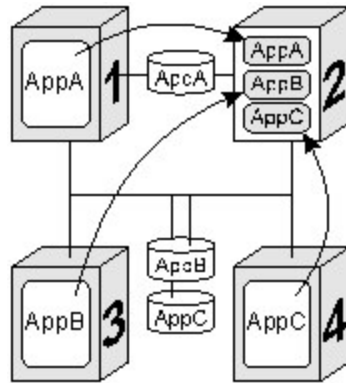
一方、AppBとAppCについては、4台のサーバがAppBとAppC共有リソースにアクセスできるため、複数の構成が可能です。例えば、Server 1やServer 2を3番目と4番目のバックアップとして、AppBとAppCをフェイルオーバーできます。



注記: LifeKeeperはボリュームレベルでロックをかけるため、AppBとAppCディスクリソースに接続しているサーバ4台のうち、実際にディスクリソースにアクセスできるのは1台だけです。この例では、Server 3がAppBを処理するため、Server 1、2、4は、ディスクリソースに物理的に接続していても、実際にはアクセスできません。

アクティブ/スタンバイ構成

アクティブ/スタンバイ構成では、プライマリサーバが処理を実行します。バックアップサーバは、プライマリサーバに障害が発生した場合に備えて待機します。待機系には、処理能力が低い小規模なシステムを使用できますが、稼働系に障害が発生した場合に備えて、リソースを継続的に運用する処理能力が必要になります。



スタンバイサーバは、複数のアクティブサーバをバックアップできます。例えば、上図の場合、Server2は3組のアクティブ/スタンバイペアをバックアップしています。リソース定義により、このアクティブ/スタンバイ構成は、次のように動作します。

- Server1で稼働するAppAは、Server2にフェイルオーバーします。
- Server3で稼働するAppBは、Server2にフェイルオーバーします。
- Server4で稼働するAppCは、Server2にフェイルオーバーします。

複数のアクティブ/スタンバイ構成を採用する場合は、次の点に注意してください。

- **ディスクの所有権**: アクティブアプリケーションが異なる場合、同じボリューム上のディスクスライスは使用できません。LifeKeeperはボリューム単位でロックをかけます。SCSIをロックした場合、共有SCSIバス上の1つのシステムだけがディスク装置のボリュームにアクセスできます。上の例では、Server3がAppBディスクリソースを所有しており、Server4がAppCリソースを所有しています。
- **処理能力**: Server1、Server3、Server4が同時に故障することは希ですが、1台のスタンバイサーバで複数のリソースをバックアップする場合は、複数サーバに障害が発生した場合を想定して、1台のサーバで重要な処理を継続できる能力が必要になります。
- **LifeKeeperの管理**: 上の例では、Server2がサーバ3台をバックアップしています。LifeKeeperデータベースを複数の論理グループで同時に管理することは望ましくありません。最初に1組の稼働系と待機系にリソースを作成し、次にその待機系と他の稼働系にリソースを作成します。

インテリジェントスイッチバックと自動スイッチバック

リソースのスイッチバック方法は、デフォルトで **[INTELLIGENT]** に設定されます。インテリジェントスイッチバックを指定した場合、リソースがサーバAからサーバBにフェイルオーバーされた後、サーバBに障害が発生するか、管理者がリソースを手動で別のサーバに切り替えるまで、リソースはサーバBにとどまります。そのため、サーバAが復旧しても、リソースはサーバBで稼働します。サーバAはバックアップとなります。

しかし、状況によっては、プライマリサーバが障害から復旧した場合、リソースを自動的に元に戻すことが望ましいこともあります。SPSでは、デフォルトのインテリジェントスイッチバックを自動スイッチバックに変更できます。自動スイッチバックはリソース階層ごとに指定できます。リソース階層を稼働しているサーバに自動スイッチバックを指定した場合、そのサーバに障害が発生すると、リソース階層はバックアップシス

テムにフェイルオーバーされますが、プライマリサーバが障害から復旧すると、リソース階層は自動的に元に戻ります。

注記

- 複製データ (DataKeeper) を使用する場合は、**インテリジェントスイッチバック**を選択してください。複製データは**自動スイッチバック**に対応していません。
- スイッチバックの確認は、LifeKeeper を起動するか、サーバをクラスタに追加すると実行されます。通常の運用時にはスイッチバックは確認されません。
- SPS は、優先度の高いサーバから低いサーバには**自動スイッチバック**を実行しません。
- 2つのリソースに依存関係があり、各リソースに異なるスイッチバック方法が指定されている場合、親リソースの設定が優先します。

SPS の設定

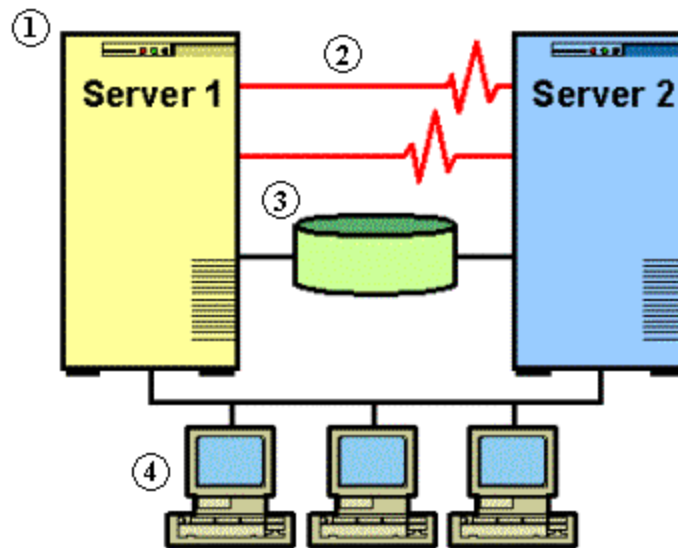
SPS は、複数のサーバに定義されたリソース階層に基づいて動作します。SPS によるフェイルオーバーの背景説明については、次の項目を参照してください。

- [共通ハードウェアコンポーネント](#)
- [システムのグループ化](#)
- [リソース階層](#)

共通ハードウェアコンポーネント

SPS 構成は、次図に示すように、共通コンポーネントを共有します。

- サーバグループ:** SPS は、Windows 2003 サーバをクラスタ化して障害回復機能を提供します。サーバ(クラスタノード)のプラットフォームは、同じでなくてもかまいません。
- コミュニケーションパス (ハートビート):** コミュニケーションパスは最低 1 本を設定する必要がありますが、2 本のコミュニケーションパスを作成することを強く推奨します。通信障害による不要なフェイルオーバーを回避するために、TCP/IP (ソケット) など、プロトコルの異なるコミュニケーションパスを作成して冗長化します。SPS は、コミュニケーションパスを使用して、障害検出用のハートビートを一定間隔で送信します。サーバの死活状態を監視し、必要に応じて、リソースを切り替えます ([コミュニケーションパスの概要](#)を参照)。
- 共有データリソース:** SPS は、共有データ、アプリケーション、通信リソースをリストアップします。そして、共有ストレージデバイスの上のリソースに対して、ボリューム (ドライブ名) 単位でアクセスを制御します。サーバに障害が発生すると、SPS は保護対象リソースをアクティブサーバに自動的に切り替えます。複数のシステムが共有する周辺装置は、外部のキャビネットに一括して収容してください。共有ストレージの構成方法については、[ストレージの構成](#)を参照してください。
- ユーザ接続用の共有通信リソース:** SPS は、IP アドレス、コンピュータの別名、ファイル共有リストなどのユーザ通信リソースを自動的に管理して切り替えます。SPS が通信リソースを自動的に切り替えることにより、ユーザは通常のパスで接続を継続できます。



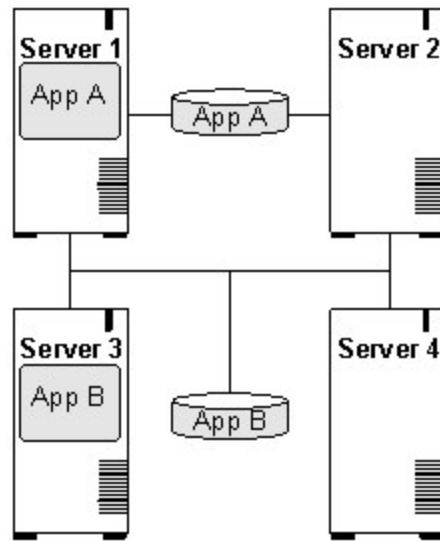
システムのグループ化

リソース階層はSPS サーバのクラスターに定義します。階層を構成するサーバには、優先度を指定します。優先度が最も高いサーバは、優先度 1 が付きます。最高の優先度が付いたプライマリサーバは、リソースを操作するコンピュータです。2 番目に高い優先度を持つサーバは、プライマリサーバに障害が発生した際、リソースを切り替えるバックアップサーバです。

[アクティブ/アクティブ構成](#)では、すべてのサーバがリソースを起動し、同時に他のサーバをバックアップします。[アクティブ/スタンバイ構成](#)では、プライマリサーバが処理を実行します。バックアップサーバは、プライマリサーバに障害が発生した場合に備えて待機します。待機系には、処理能力が低い小規模なシステムを使用できますが、稼働系に障害が発生した場合に備えて、リソースを継続的に運用する処理能力が必要になります。

共有リソースへの物理的な接続とアクセス方法により、グループ化のオプションが決まります。グループ化を実行するには、ハートビートを送信するコミュニケーションパスを設定し、全サーバが共有 SCSI インターフェイスまたはファイバチャネルインターフェイスを介してディスクリソースにアクセスできるようにします。例えば、次図では、Server 1 のリソース AppA に関して、1 つのグループを設定できます。この構成では、AppA データベースに共有アクセスするサーバは Server 2 だけです。

一方、Server 3 で稼働する AppB リソースは、3 台のサーバを任意に構成できます。共有 SCSI バスがサーバ 4 台に AppB データベースへのアクセスを提供しているためです。



Chapter 4: SPS 管理の概要

SPS には、次の 2 種類の管理インターフェースがあります。:

- LifeKeeper GUI
- LifeKeeper コマンドラインインターフェース

LifeKeeper 管理インターフェースは次のタスクに使用します。ここでは、SPS を設定する順番に従って、各タスクを説明します。

- **コミュニケーションパスの定義**。コミュニケーションパスを定義するには、[編集] メニューまたは GUI ツールバーの [コミュニケーションパスの作成] を使用します。ソースインスタンスとリソース階層を定義する前に、コミュニケーションパスを定義します。
- **リソースの定義**。リカバリキットをインストールすると、そのリカバリキットがサポートするリソースタイプが [リソース階層の作成] ダイアログボックスに表示されます。大半のリカバリキットは、依存関係を自動的に作成します。
- **監視**。LifeKeeper GUI のステータスには、接続先サーバの保護対象リソースのステータスがアイコンで表示されます。SPS が記録するログも GUI に表示されます。
- **手動による介入**。SPS を保守管理する場合は、サーバとリソースを停止します。LifeKeeper GUI には、特定のリソースを起動/停止するメニューがあります。SPS でアプリケーションを保護している場合、サービスの起動と停止は、SPS だけで実行します。

SPS の初期設定については [SPS の設定手順](#) を参照してください。

GUI インターフェースを使用した SPS の管理、設定、保守の手順については、[GUI による管理作業とメンテナンス作業](#) を参照してください。

注記: SPS は、各サーバのローカルシステムアカウントで SPS サービスを実行するように設定されています。この設定を変更して、他のユーザアカウントで SPS を実行しないでください。

GUI による管理作業

サーバプロパティの編集

1. サーバプロパティを編集するには、最初に [サーバプロパティを表示](#) します。
2. 適切なアクセス権でサーバにログインすると、次の項目を編集できます。
 - シャットダウン方式
 - 自動フェイルオーバーの設定

サーバのシャットダウン方法の設定

- サーバ設定 (特殊な構成を設定したサーバのみ)
3. 編集が完了すると、**[適用]** ボタンが有効になります。このボタンをクリックすると、変更内容が反映されます。ただし、ウィンドウは閉じません。
 4. 作業が完了したら、**[OK]** をクリックして、変更内容を保存してウィンドウを閉じるか、**[キャンセル]** をクリックして、変更を適用せずに、ウィンドウを閉じます。

サーバのシャットダウン方法の設定

シャットダウン方法とは、サーバをシャットダウンする際、リソースをバックアップサーバに切り替える設定オプションを指します。次のオプションを指定できます。

リソースを切り替えない(デフォルト)	通常のシャットダウンではリソース階層を切り替えません。
リソースを切り替える	通常のシャットダウン時にすべてのリソース階層を切り替えます。

制限: シャットダウン時のスイッチオーバーの設定は SteelEye DataKeeper リソースではサポートされません。

シャットダウン方法はデフォルトで **[リソースを切り替えない]** に設定されています。各サーバにシャットダウン方法を指定してください。必要に応じて、**[リソースを切り替える]** に変更します。

次の手順に従って、クラスタサーバのシャットダウン方法を変更してください。

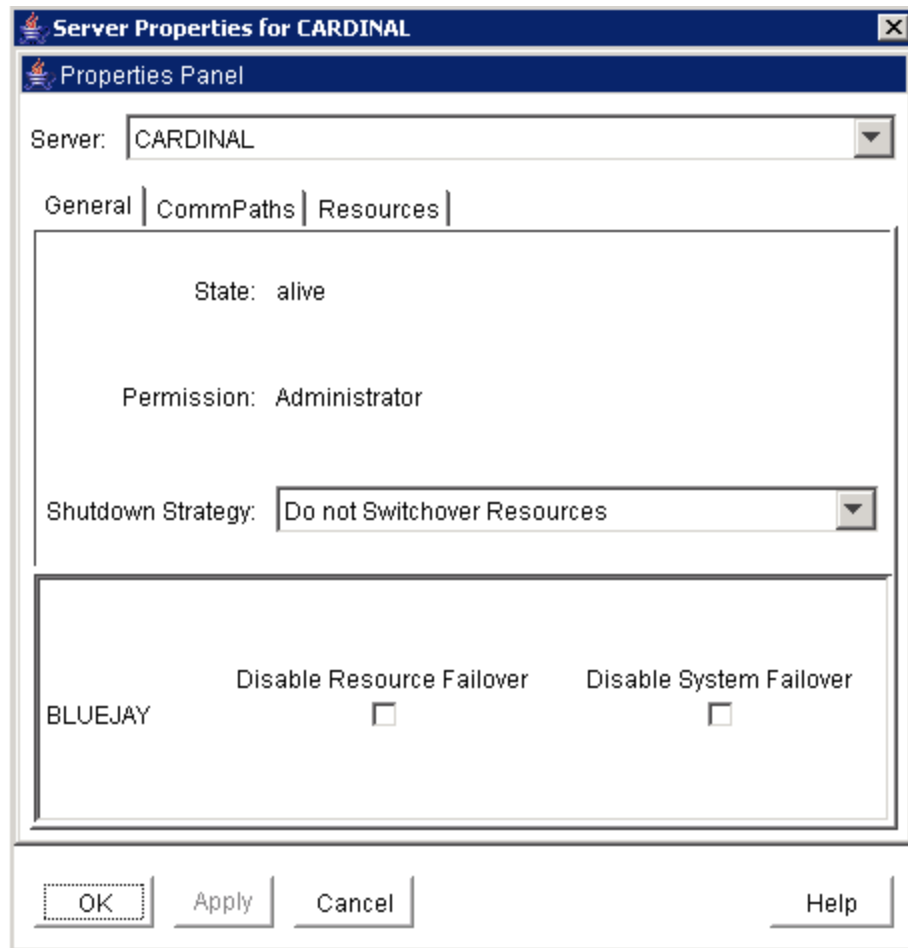
1. [サーバプロパティの表示](#) で説明した手順に従って、**[サーバプロパティ]** ダイアログボックスを開きます。
2. **サーバプロパティ** ダイアログボックスの **[一般]** タブでシャットダウン方法を選択します。

注記: ここで指定したシャットダウン方法は、通常のシャットダウン時に、SPS プロセスが稼働していないと正常に機能しません。SPS が稼働していないか、リソースサービスが停止していると、リソースは切り替えられません。

サーバプロパティ

サーバプロパティ ダイアログボックスは、[編集メニュー](#) またはサーバのポップアップメニューから使用できます。このダイアログボックスには、サーバのプロパティが表示されます。[編集メニュー](#) を使用して、サーバを選択します。選択したサーバを変更すると **サーバプロパティ** ダイアログボックスも自動的に更新されます。

[OK] をクリックすると、変更内容が適用されてウィンドウが閉じます。**[適用]** をクリックすると、変更が反映されます。**[キャンセル]** をクリックすると、**[適用]** をクリックした後に変更した内容を保存せずにウィンドウが閉じます。

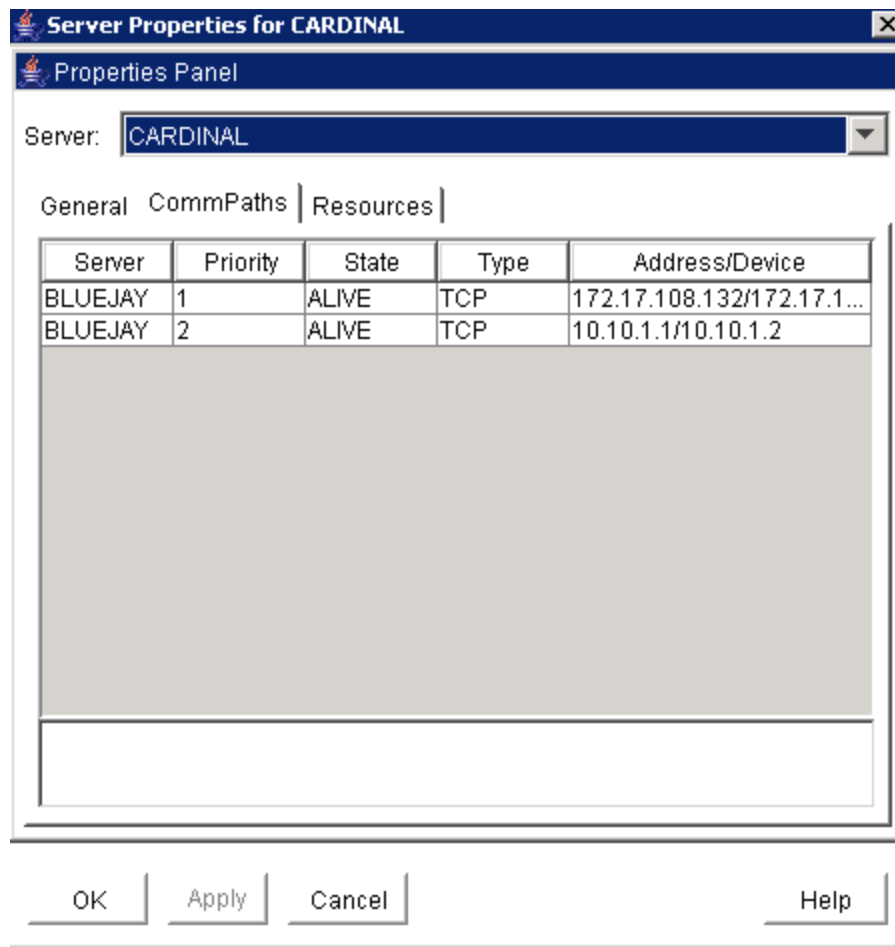


- **名前:** 選択したサーバの名前
- **権限:** サーバにログインするユーザに付与する権限です。次の権限を指定できます。
 - **管理者** - SPS に関するすべてのタスクを実行できます。
 - **オペレータ** - SPS のリソースとサーバの状態を監視できます。リソースを起動/停止できます。
 - **ゲスト** - SPS のリソースとサーバの状態を監視できます。
- **状態:** サーバの現在の状態。サーバは次のいずれかの状態になります。
 - **稼働中** - サーバは使用可能です。
 - **停止中** - サーバは使用できません。
 - **不明** - 状態を特定できません。GUI サーバは使用できない可能性があります。
- **シャットダウン時の動作:** (編集可能) サーバをシャットダウンする際、稼働中のリソースをバックアップサーバに切り替える方法を制御します。[リソースを切り替える]を設定すると、リソースはクラスタ内のバックアップサーバに移動します。[リソースを切り替えない]を設定すると、リソースはクラ

スタ内の別のサーバに移動しません。

- **サーバ名**: クラスタ内でローカルサーバからバックアップサーバに自動的にフェイルオーバーする機能は、このタブで設定します。休止中のサーバはリストに表示されません。そのため、クラスタを構成しているサーバすべてが運用可能です。つまり、少なくとも、SPS コミュニケーションパス 1 本がアクティブになっています。リストには、ローカルサーバを除き、クラスタでアクティブなサーバの名前が一覧表示されます。各サーバには、2 種類のフェイルオーバー機能を設定できます。すべてのフェイルオーバー機能は、デフォルトで有効に設定されます。
 - **リソースフェイルオーバー不可** - ローカルサーバのリソース階層に障害が発生した場合、バックアップとして使用しないリモートサーバを選択します。バックアップ機能を無効にしたサーバは、ローカルリソースに障害が発生しても、フェイルオーバーサイトとして使用されません。フェイルオーバー機能は、設定を解除すると再び有効になります。
 - **システムフェイルオーバー不可** - ローカルサーバが完全に停止した場合、バックアップとして使用しないリモートサーバを選択します。無効に設定したサーバは、ローカルサーバが完全に停止しても、フェイルオーバーサイトとして機能しません。フェイルオーバー機能は、設定を解除すると再び有効になります。

注記: すべてのリモートサーバでリソースフェイルオーバーを無効にすると、障害が発生したリソースに「Failed」マークが付きます。このマークが付いたリソースは、クイックチェックとディープチェックの監視外となります。ただし、障害が発生したリソースまたは階層内の依存リソースは、サービスから削除されませんが、フェイルオーバーされません。

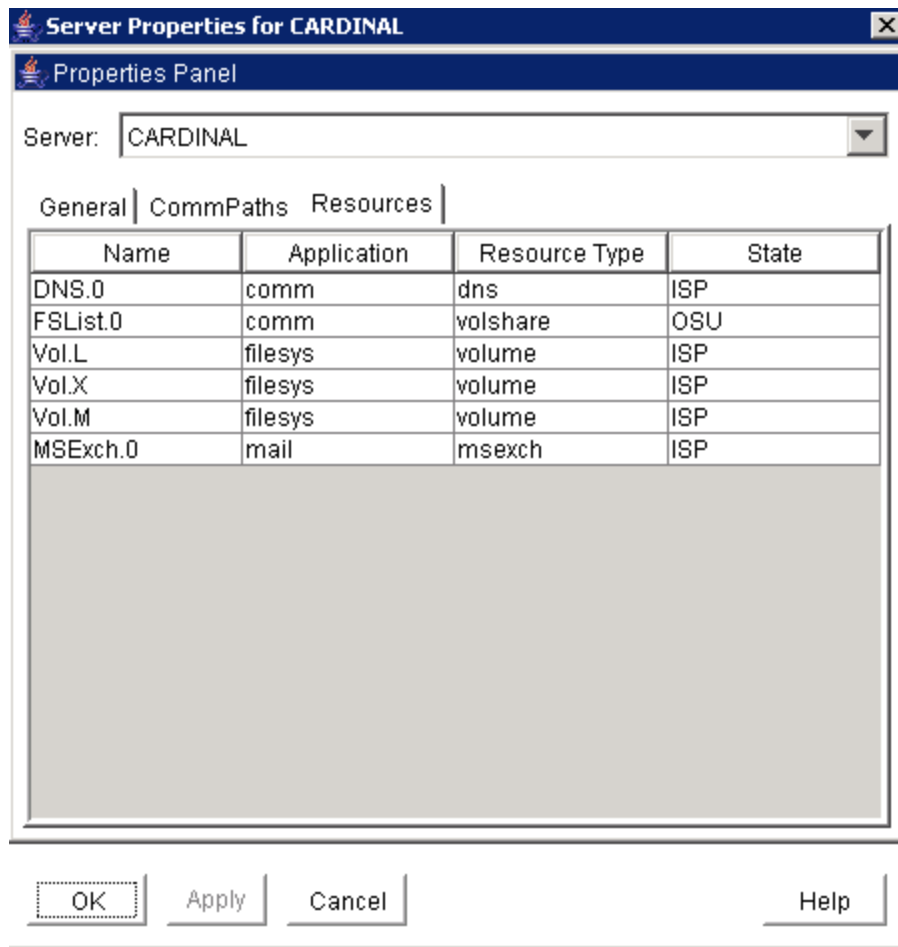


[コミュニケーションパス] タブ

- **サーバ:** SPS クラスタでコミュニケーションパスが接続されているその他のサーバの名前。
- **タイプ:** リストのサーバと[サーバ]フィールドで指定されているサーバ間のコミュニケーションパスの種類 (TCP/IP、共有ディスク)。
- **状態:** LifeKeeper 構成データベース (LCD) のコミュニケーションパスの状態。コミュニケーションパスの状態を表す値は、次のとおりです。
 - **稼働中** - 正常に機能しています。
 - **停止中** - 正常に機能していません。
 - **不明** - 状態を特定できません。LifeKeeper GUI サーバは使用できない可能性があります。
- **アドレス/デバイス:** このコミュニケーションパスが使用する IP アドレスまたはデバイス名。
- **コミュニケーションパスステータス:** LifeKeeper 構成データベース (LCD) のコミュニケーションパスの

状態に基づき、GUI が決定したコミュニケーションパスステータスのサマリ。下側のパネルには、コミュニケーションパスのステータスが次のように表示されます。

- **正常** - すべてのコミュニケーションパスが正常に機能しています。
- **障害** - 特定のサーバへのすべてのコミュニケーションパスが機能していません。
- **不明** - コミュニケーションパスの状態を特定できません。LifeKeeper GUI サーバは使用できない可能性があります。
- **警告** - 特定のサーバへの1つ以上のコミュニケーションパスが停止しているか、コミュニケーションパス1本だけが稼働しています。
- **縮退** - 特定のサーバに接続した冗長コミュニケーションパスが機能していません。
- **定義なし** - コミュニケーションパスが定義されていません。



[リソース] タブ

- **名前:** 選択したサーバのリソースインスタンスのタグ名。
- **アプリケーション:** リソースタイプのアプリケーション名 (gen、scsi など)。
- **リソースタイプ:** リソースタイプ。つまり、サービスを提供しているハードウェア、ソフトウェア、またはシステムエンティティのクラス (ボリュームや TCP/IP、SQL など)。
- **状態:** リソースインスタンスの現在の状態。
 - **ISP** - ローカルで稼働しており、LifeKeeper で保護されています。
 - **ISU** - ローカルで稼働していますが、ローカルリカバリは試行されません。
 - **OSF** - サービス休止中で、障害が発生しています。
 - **OSU** - サービスを休止していますが、障害は発生していません。
 - **不正な状態** - LifeKeeper の起動シーケンスとして実行されるリソースの初期化プロセスがリソースステータスを適切に初期化していません。この状態のリソースは、LifeKeeper で保護されません。
 - **不明** - リソースの状態を特定できません。GUI サーバは使用できない可能性があります。

自動フェイルオーバーを無効にする

プライマリサーバがローカルリソースのリカバリに失敗するか、サーバが完全に停止した場合、LifeKeeper は障害が発生したリソースをバックアップサーバに復元します。これが LifeKeeper のデフォルトの動作です。しかし、LifeKeeper で保護するリソースをリカバリサイトで自動的に起動しない方が良いこともあります。例えば、LifeKeeper を WAN 環境にインストールしても、サーバ間のネットワーク接続が不安定な場合は、障害回復に不適切です。

自動フェイルオーバー機能は、デフォルトで保護対象リソースに設定されます。保護対象リソースの自動フェイルオーバーを無効にするか、バックアップサーバへの自動フェイルオーバーを無効にするには、[\[サーバプロパティ\]](#) ダイアログボックスを開き、**[一般]** タブの **[フェイルオーバー]** タブをクリックしてください。

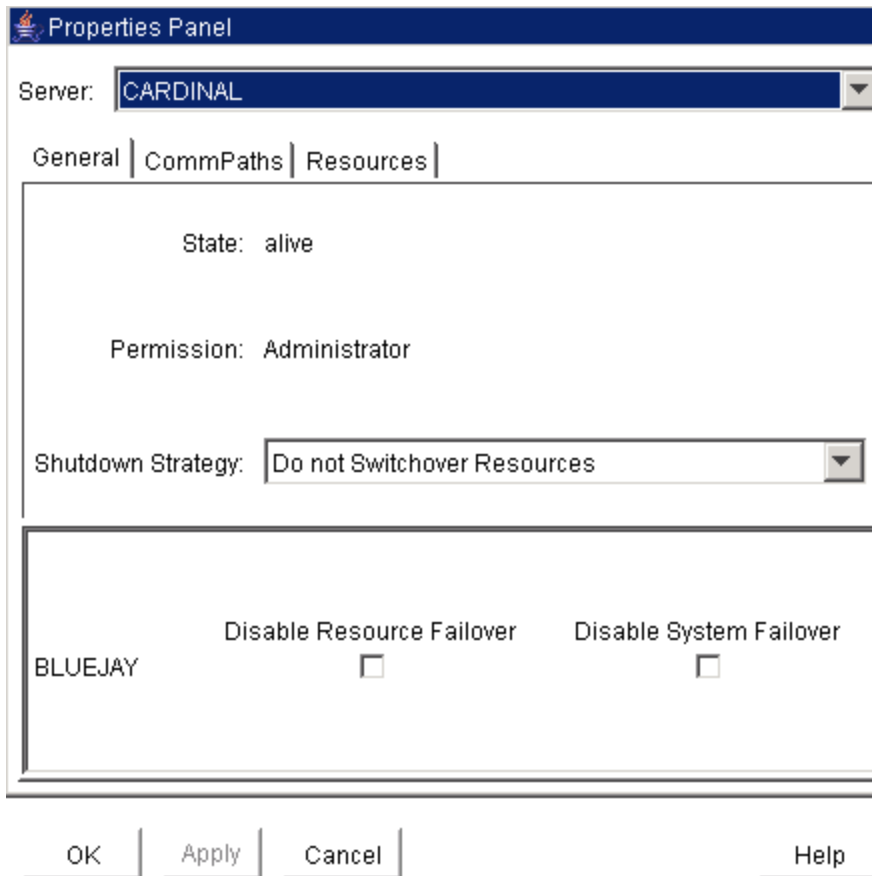
次の手順に従って、クラスタサーバのシャットダウン方法を変更してください。

1. [サーバプロパティの表示](#) で説明した手順に従って、**[サーバプロパティ]** ダイアログボックスを開きます。
2. **[一般]** タブをクリックします。**[サーバプロパティ]** ダイアログボックスの **[フェイルオーバー]** グループに移動し、システムとリソースのフェイルオーバー機能を無効にするサーバを確認します。SPS のフェイルオーバー機能はデフォルトで有効に設定されています。

ローカルサーバが完全に停止した際、バックアップ機能を無効にするサーバを **[システムフェイルオーバー不可]** 列から選択してください。

ローカルサーバのリソース階層に障害が発生した際、バックアップ機能を無効にするサーバを **[リソースフェイルオーバー不可]** 列から選択してください。リソースのフェイルオーバー機能は、システムのフェイルオーバー機能が無効でない、無効に設定できません。

[適用]をクリックして、設定を反映してください。



コミュニケーションパスの作成

SPS コミュニケーションパスをサーバ間に設定する前に、ハードウェアとソフトウェアが適切にセットアップされていることを確認してください。要件については、[構成](#) セクションを参照してください。

構成に関する注意事項

- サーバ間には、共有ディスクコミュニケーションパスを1本ずつ設定します。
- 共有ディスクコミュニケーションパスは、2サーバ構成のクラスタだけに対応しています。
- 3台以上のサーバでクラスタを構成する場合は、複数のTCP/IP コミュニケーションパスを使用して、ハートビートを冗長化します。TCP/IP パスには優先度を指定します。SPS は優先度を参照して、リモートサーバへのTCP/IP パスを使用する順番を決定します。
- **重要:** 通信回線1本に障害が発生した場合、不要なフェイルオーバーが発生しないように、コミュニケーションパスを冗長化しておきます。コミュニケーションパスを1本だけ使用した場合、そのコミュニケーションパスに障害が発生すると、リソース階層が複数サーバで同時にサービスを起動

することがあります。これを「スプリットブレイン」と呼びます。また、TCP/IP コミュニケーションパスでネットワークトラフィックが増加すると、偽のフェイルオーバーが発生したり、SPS を正常に初期化できなくなったりするなど、予期せぬ問題が生じることがあります。

コミュニケーションパスの作成

1. 1 台のサーバを選択し、[サーバコンテキストメニュー](#)または[サーバコンテキストツールバー](#)の【コミュニケーションパスの作成】をクリックします。
2. リストボックスから 1 台以上のリモートサーバを選択します。リモートサーバをクラスタに接続していないため、サーバがリストに表示されない場合は、【サーバ追加】をクリックして、リモートサーバを追加してください。DNS を使用するか、`/etc/hosts` ファイルに IP アドレスを登録し、リモートとローカルのサーバの IP アドレスが解決できることを確認してください。【次へ】をクリックしてください。
3. デバイスタイプとして、TCP、TTY、または DISK を選択し、【次へ】をクリックします。
4. デバイスタイプ設定に必要な情報を選択し、各手順が完了するたびに、【次へ】をクリックします。それぞれの設定においての詳細情報は、下記の表を参照してください。



フィールド	説明
TCP/IP コミュニケーションパス	
ハートビート間隔	ハートビート間隔 には 4 ~ 15 の値を入力して、サーバが稼働していることを確認するハートビート信号を送信する間隔を秒単位で指定します。デフォルト値は 6 です。
ハートビート最大欠落数	有効値は 3 ~ 99 です。ハートビート最大欠落数は、ハートビート信号が連続して欠落した場合、何回目でコミュニケーションパス障害と判断するかを指定します。デフォルト値は 5 です。
ローカル IP アドレス	コミュニケーションパスを使用するローカルサーバの IP アドレスを入力します。
優先度	ローカルサーバにおけるコミュニケーションパスの優先度を指定します。優先度は、サーバ間のコミュニケーションパスの優先順位を決定するために使用されます。優先度は 1 が最高、99 が最低です。
リモート IP アドレス	コミュニケーションパスを使用するリモートサーバの IP アドレスを入力します。
TCP ポート番号	TCP/IP サービスが使用する一意のポート番号を入力します。有効値は 1500 ~ 10000 です。ポート番号にはデフォルト値が表示されますが、この値は変更できません。
共有ディスクコミュニケーションパス	

フィールド	説明
ハートビート間隔	ハートビート間隔 には4～15の値を入力して、サーバが稼働していることを確認するハートビート信号を送信する間隔を秒単位で指定します。デフォルト値は6です。
ハートビート最大欠落数	有効値は3～99です。ハートビート最大欠落数は、ハートビート信号が連続して欠落した場合、何回目でコミュニケーションパス障害と判断するかを指定します。デフォルト値は5です。
優先度	ローカルサーバにおけるコミュニケーションパスの優先度を指定します。優先度は、サーバ間のコミュニケーションパスの優先順位を決定するために使用されます。優先度は1が最高、99が最低です。
ドライブ名	共有ディスクコミュニケーションパスに使用する共有ボリュームのドライブ名を指定します。ドライブ名は両方のサーバで統一します。

5. **[作成]** をクリックします。ネットワーク接続が確立したことを伝えるメッセージが表示されます。アウトプットパネルが有効な場合、メッセージはアウトプットパネルに表示されます。**[次へ]** をクリックしてください。
6. 複数のローカルIPアドレスまたはリモートサーバを選択し、**デバイスタイプ**に**TCP**を指定した場合は、手順4に戻り、次のコミュニケーションパスを作成します。複数のリモートサーバを選択し、**デバイスタイプ**を**DISK**に設定した場合は、手順3に戻り、次のコミュニケーションパスを設定します。
7. コミュニケーションパスの作成が完了したことを伝えるメッセージが表示されたら、**[完了]** をクリックします。

コミュニケーションパスの確認

コミュニケーションパスを確認するには、[\[サーバプロパティ\]](#) ダイアログボックスを使用します。ステータスが**[起動中]**であることを確認してください。

次に、GUIの右ペインにあるサーバのアイコンを確認してください。最初のコミュニケーションパスを作成した後であれば、サーバのアイコンに黄色のハートビートが表示されます。これはコミュニケーションパス1本が稼働中であり、冗長なコミュニケーションパスが存在しないことを示しています。	
2つのコミュニケーションパスを作成すると、サーバのアイコンが緑色のハートビートに変わります。	

コミュニケーションパスを作成してから数分が経過しても、稼働中にならない場合は、リモートサーバの名前が正しいことを確認してください。

コミュニケーションパスの削除

1. 1台のサーバを選択し、[サーバコンテキストメニュー](#)または[サーバコンテキストツールバー](#)の**[コミュニ**

ケーションパスの削除]をクリックします。

2. 削除するコミュニケーションパスを選択し、[コミュニケーションパスの削除]をクリックします。
3. [アウトプットパネル](#)が有効な場合、ダイアログボックスが閉じて、コマンドの実行結果がアウトプットパネルに表示されます。すべての結果が表示されたら、[完了]をクリックしてダイアログボックスを閉じます。

リソース階層に関連する作業

このセクションでは、すべてのリソース階層に共通するタスクについて説明します。各タスクは、Core Recovery Kit とオプションのリカバリキットで共通しています。

オプションの SPS Recovery Kits のドキュメントは SPS for Windows テクニカルドキュメンテーションから入手可能です。

リソース階層の作成

1. サーバを選択します。次に、[サーバコンテキストメニュー](#)または[サーバコンテキストツールバー](#)の[リソース階層の作成]をクリックします。
2. [リソース階層の作成]ウィザードが起動し、クラスタにインストールされているリカバリキットの一覧が表示されます。リソース階層を構築して、アプリケーションを保護するリカバリキットを選択し、[次へ]をクリックします。
3. 引き続きダイアログを続け、作成するリソース階層のタイプに必要なデータを入力します。
4. リソース階層が正常に作成されたことを伝えるメッセージが表示されたら、[次へ]をクリックして操作を続けます。問題が発生すると情報ボックスにエラーが表示され、部分的に作成されたリソース階層が削除されて、[次へ]ボタンが無効になります。その場合は、[キャンセル]をクリックして、ウィザードを終了してください。

SPS のアプリケーションリソース階層

オプションのリカバリキットをインストールせずに、SPS だけをインストールした場合、[保護するアプリケーション]の一覧には、オプションとして、DNS、共有ファイルリスト、Generic Application、IIS、IP アドレス、LAN Manager、ボリュームが表示されます。[汎用アプリケーション] オプションは、リカバリキットに対応していないアプリケーションを保護するときに使用してください。

選択可能なオプションについては、次の項目を参照してください。

- [DNS リソース階層の作成](#)
- [ファイル共有リソース階層の作成](#)
- [Generic Application リソース階層の作成](#)
- [IP リソース階層の作成](#)

- [LAN Manager リソース階層の作成](#)
- [ボリュームリソース階層の作成](#)

Microsoft SQL Server Recovery Kit

Microsoft SQL Server Recovery Kitをインストールすると、保護するアプリケーションの一覧にエントリが追加されます。必要なリソース階層の作成方法につきましてはMicrosoft SQL Server Recovery Kit 管理ガイドを参照してください。

ファイル共有リソース階層の作成

ファイル共有はWindows版のファイルマネージャで定義します。SPS LAN Manager リカバリキットをインストールすると、ファイル共有を含めたファイル共有リソースを作成できます。

ファイル共有リソースの条件

LifeKeeperは、すべてのファイル共有をサポートしていません。次の説明を参考にして、利用可能なファイル共有を判断してください。

- 共有名はサーバが共有するボリュームに設定します。
- 2台のサーバにファイル共有リソースを作成すると、共有ボリュームは自動的に保護されます。しかし、3台目以降のサーバは、ファイル共有階層を拡張しないと、ファイルを共有できません。
- 2台目のサーバに共有名をすでに設定している場合、共有名は同じディレクトリを指す必要があります。
- いずれか一方のサーバで共有名が保護されている場合、その共有名は使用できません。
- 管理者は、作成した共有名がディレクトリを指すことを確認します。ディレクトリに共有名を作成した後でも、そのディレクトリを削除できます。ディレクトリを削除した場合は、共有名を削除したことを確認してください。

注記: バックアップサーバでファイル共有リソースを起動すると、共有が有効になります。階層を削除しても、ファイル共有は削除されません。

ファイル共有リソースの作成

ファイル共有リソースを作成するには、次の手順に従ってください。

1. サーバを選択します。次に、[サーバコンテキストメニュー](#)または[サーバコンテキストツールバー](#)の[リソース階層の作成]をクリックします。
2. [保護するアプリケーションを作成] ウィンドウが起動し、プライマリサーバおよびバックアップサーバが表示されます。まだ選択していない場合は、設定のための適切なシステムを選択してください。
3. クラスタにインストールされているすべてのリカバリキットの一覧が表示されます。[ファイル共有リスト]を選択し、[次へ]をクリックします。

4. 設定ウィザードで、以下の情報を入力します。ダイアログボックス内で **[戻る]** ボタンがアクティブになっているときは、直前のダイアログボックスに戻ることができます。これは特に、前に入力した情報の訂正が必要なエラーが発生した場合に役立ちます。階層作成手順の間に **[キャンセル]** をクリックすると、LifeKeeper は作成プロセス全体を取り消します。

フィールド	説明
共有ファイルとパス名	保護する共有ファイルを選択します。複数のファイルを選択できます。ファイルが表示されない場合は、ファイル共有を指定したボリュームがSPSで保護されているか確認してください。
共有ファイルリソースタグ	作成している共有ファイルのリソースインスタンスに割り当てて一意のタグ名を選択するか入力します。このフィールドには、デフォルトのタグ名 FSList.x が自動的に表示されます。x には、開始番号が自動的に入ります。タグ名は変更できます。

5. データの入力を完了すると、**[次へ]** ボタンが表示されます。**[次へ]** をクリックすると、リソース階層の作成と検証が実行されます。
6. リソース階層が正常に作成されたことを伝えるメッセージが表示されたら、**[次へ]** をクリックして操作を続けます。問題が発生すると情報ボックスにエラーが表示され、部分的に作成されたリソース階層が削除されて、**[次へ]** ボタンが無効になります。その場合は、**[キャンセル]** をクリックして、ウィザードを終了してください。

Generic Application リソース階層の作成

このオプションでは、対応するリカバリキットのアプリケーションを保護します。

リソース階層を作成する前に

- 最初に、SPS の 5 つの基本的な動作を制御するスクリプトを用意します。
 - Restore (起動)
 - Remove (停止)
 - Quick Check (監視)
 - Deep Check (監視)
 - ローカルリカバリ

各スクリプトの Perl 版と VB 版のテンプレートが `$LKROOT\admin\kit\appl\templates` に格納されています。テンプレートをカスタマイズしてテストするときは、`$LKROOT` と同じボリューム上の別のディレクトリにコピーしてください。

注記: オプションの **Create**、**Extend**、**Delete** スクリプトを使用する場合は、他のスクリプトと一緒に前記のフォルダにコピーしてください。スクリプト選択ウィザードは、追加されたスクリプトの名前と拡張子を調べ、スクリプトを自動的に登録します。

2. ボリュームやIP アドレスなど、他のリソースに依存するアプリケーションでは、リソースを個別に作成した後、Generic Ark でリソース階層を作成します。依存関係を後から作成するには、[\[依存関係の追加\]](#) ダイアログボックスを使用してください。

リソース階層の作成

基本スクリプトを用意したら、Generic Ark を使用して、アプリケーションのリソース階層を作成してください。

1. サーバを選択します。次に、[サーバコンテキストメニュー](#)または[サーバコンテキストツールバー](#)の[リソース階層の作成]をクリックします。
2. **[保護するアプリケーションを作成]** ウィンドウが起動し、**プライマリサーバ**および**バックアップサーバ**が表示されます。まだ選択していない場合は、設定のための適切なシステムを選択してください。**[次へ]**をクリックします。
3. クラスタにインストールされているすべてのリカバリキットの一覧が表示されます。**[Generic Application]** を選択し、**[次へ]**をクリックします。
4. **設定ウィザード**で、以下の情報を入力します。**注記:** ダイアログボックス内で**[戻る]** ボタンがアクティブになっているときは、直前のダイアログボックスに戻ることができます。これは特に、前に入力した情報の訂正が必要なエラーが発生した場合に役立ちます。階層作成手順の間に**[キャンセル]**をクリックすると、SPS は作成プロセス全体を取り消します。

フィールド	説明
Restore スクリプト	アプリケーションに使用する restore スクリプト のパス名とファイル名を入力します。このスクリプトはアプリケーションを起動します。restore スクリプトのテンプレートは、templates ディレクトリにあります。restore スクリプトは、稼働中のアプリケーションに影響を与えてはなりません。
スクリプトの削除	アプリケーションに使用する remove スクリプト のパス名とファイル名を入力します。このスクリプトはアプリケーションを停止します。remove スクリプトのテンプレートは、templates ディレクトリにあります。
クイックチェックスクリプト [オプション]	アプリケーションに使用する quickCheck スクリプト のパス名を入力します。このスクリプトはアプリケーションを監視します。quickCheck スクリプトのテンプレートは、templates ディレクトリにあります。
ディープチェックスクリプト [オプション]	アプリケーションに使用する deep check スクリプト のパス名を入力します。deepCheck スクリプトは、quickCheck スクリプトより詳細に保護対象アプリケーションを監視します。deep check スクリプトのテンプレートは、templates ディレクトリにあります。
ローカルリカバリ [オプション]	アプリケーションで使用する ローカルリカバリスクリプト のパス名を入力します。このスクリプトは、障害が発生したアプリケーションをローカルサーバに復旧します。recover スクリプトのテンプレートは、templates ディレクトリにあります。

フィールド	説明
アプリケーション情報 (オプション)	任意のアプリケーション情報を入力します。restore、remove、recover、quickCheck スクリプトに必要なアプリケーションに関する情報をオプションとして入力します。
リソースタグ名	このフィールドには、デフォルトのタグ名 <i>App.x</i> が自動的に表示されます。x は SPS が割り当てる値で、0 から始まります。

- データの入力を完了すると、[インスタンスの作成] ボタンが表示されます。[インスタンスの作成] をクリックすると、リソース階層の作成と検証が実行されます。
- リソース階層が正常に作成されたことを伝えるメッセージが表示されたら、[次へ] をクリックして操作を続けます。問題が発生すると情報ボックスにエラーが表示され、部分的に作成されたリソース階層が削除されて、[次へ] ボタンが無効になります。その場合は、[キャンセル] をクリックして、ウィザードを終了してください。

LAN Manager リソース階層の作成

LAN Manager Recovery Kit は、共有ファイルを持つコンピュータに別名を作成します。コンピュータの別名とは、「切り替え可能」なコンピュータ名を指します。別名に関連付けた共有ファイルは、SPS の LAN Manager 階層が起動しているシステムで使用できます。IP アドレスも LAN Manager 階層としてコンピュータの別名に関連付けることができます。

- サーバを選択します。次に、[サーバコンテキストメニュー](#)または[サーバコンテキストツールバー](#)の [リソース階層の作成] をクリックします。
- [保護するアプリケーションを作成] ウィンドウが起動し、プライマリサーバおよびバックアップサーバが表示されます。まだ選択していない場合は、設定のための適切なシステムを選択してください。[次へ] をクリックします。
- クラスタにインストールされているすべてのリカバリキットの一覧が表示されます。[LanManager] をクリックし、[次へ] をクリックします。
- 設定ウィザードで、以下の情報を入力します。ダイアログボックス内で [戻る] ボタンがアクティブになっているときは、直前のダイアログボックスに戻ることができます。これは特に、前に入力した情報の訂正が必要なエラーが発生した場合に役立ちます。階層作成手順の間に [キャンセル] をクリックすると、SPS は作成プロセス全体を取り消します。

フィールド	説明
コンピュータの別名	コンピュータの別名を入力するか、デフォルト名を使用します。

フィールド	説明
LAN Manager リソース タグ	作成している LAN Manager リソースインスタンスの一意のタグ名を選択するか入力します。このフィールドには、前記の操作で入力したコンピュータの別名がデフォルトとして表示されます。別名は変更できます。

5. データの入力を完了すると、**[次へ]** ボタンが表示されます。**[次へ]** をクリックすると、リソース階層の作成と検証が実行されます。
6. リソース階層が正常に作成されたことを伝えるメッセージが表示されたら、**[次へ]** をクリックして操作を続けます。問題が発生すると情報ボックスにエラーが表示され、部分的に作成されたリソース階層が削除されて、**[次へ]** ボタンが無効になります。その場合は、**[キャンセル]** をクリックして、**ウィザード**を終了してください。

ボリュームリソース階層の作成

共有 SCSI ディスク上のリソースを保護するには、Windows のディスク管理ツールを使用して、共有ディスクを論理ボリュームに分割します。ボリュームリソースインスタンスを定義すると、LifeKeeper で共有ボリュームを保護できます。インスタンスにはドライブレターを指定します (例: G:)。

SPS は、ボリュームリソースインスタンスをプライマリサーバで起動した後、ソフトウェアロックをかけます。その結果、ボリュームがプライマリサーバでアクティブな限り、バックアップサーバはボリュームにアクセスできません。プライマリサーバに障害が発生すると、SPS はボリュームリソースをバックアップサーバで起動し、修復中のプライマリサーバがボリュームリソースにアクセスできないようにロックします。

SPS は、プライマリとバックアップを入れ替えるため、障害が発生したサーバは、ボリュームリソースにアクセスできなくなります。そのため、障害が発生したサーバを修理している間は、不適切なアクセスからリソースを保護できます。

SPS のインテリジェントスイッチバック機能は、プライマリサーバとバックアップサーバを動的に再定義します。そのために、修理が完了したら、適切なタイミングを指定して、修復したサーバでリソースを起動できます。

ボリュームリソースを作成するには、次の手順に従ってください。SPS はボリュームロックを保持するため、ボリュームリソースを作成した後は、SPS を停止しないでください。SPS を停止すると、ボリュームロックが無効になります。

注記: ミラーボリュームリソースを作成して拡張する前に、SPS クラスタシステムに接続している DataKeeper GUI プロセスを必ず終了してください。

1. サーバを選択します。次に、[サーバコンテキストメニュー](#)または[サーバコンテキストツールバー](#)の**[リソース階層の作成]** をクリックします。
2. **[保護するアプリケーションを作成]** ウィンドウが起動し、**プライマリサーバ**および**バックアップサーバ**が表示されます。まだ選択していない場合は、設定のための適切なシステムを選択してください。
3. クラスタにインストールされているすべてのリカバリキットの一覧が表示されます。ダイアログボックス内で**[戻る]** ボタンがアクティブになっているときは、直前のダイアログボックスに戻ることができます。

す。これは特に、前に入力した情報の訂正が必要なエラーが発生した場合に役立ちます。階層作成手順の間に **[キャンセル]** をクリックすると、SPS は作成プロセス全体を取り消します。

フィールド	説明
ボリューム	保護対象のボリュームを選択します。「見つかりません」と表示される場合は、SPS で保護されるボリュームを確認してください。
ボリュームタグ	ボリュームタグはリソース識別子を指定します。SPS は、デフォルトのボリュームタグ名「Volume.X」を割り当てます。X は、ドライブレターです。タグ名は変更できますが、一意でなければなりません。

4. データの入力を完了すると、**[次へ]** ボタンが表示されます。**[次へ]** をクリックすると、リソース階層の作成と検証が実行されます。
5. リソース階層が正常に作成されたことを伝えるメッセージが表示されたら、**[次へ]** をクリックして操作を続けます。**[ボリュームリソースの拡張 ウィンドウ]** が表示されます。この手順を完了させるための詳細情報については、ヘルプトピックの [ボリュームリソースの拡張](#) を参照してください。
6. 問題が発生すると情報ボックスにエラーが表示され、部分的に作成されたリソース階層が削除されて、**[次へ]** ボタンが無効になります。その場合は、**[キャンセル]** をクリックして、**ウィザード** を終了してください。

ボリュームリソース作成後の作業

SPS ボリュームを作成するか削除した場合は、SPS で保護するボリュームに次のコマンドを実行します。

```
chkntfs /x <vol_1> <vol_2> ... <vol_n>
```

この Windows のコマンドは、システムの起動時に chkdsk のチェック対象となるボリュームのリストから、指定したボリュームを除外します。SPS で保護するボリュームは、SPS が起動する前に、バックアップシステムがアクセスできないようにしておきます。chkntfs /d コマンドは、SPS ボリュームがないと、Windows のデフォルト設定を復元します。

注意: chkntfs /x コマンドは前回実行対象となったボリュームを記憶していないため、ユーザがこのコマンドを実行すると SPS の設定が無効になります (結果的に SPS がユーザ設定を上書きしてしまう可能性があります)。システム起動時に SPS 以外のボリュームをチェック対象から外す場合は、すべての SPS ボリュームを chkntfs /x コマンドに指定してください。

IP リソース階層の作成

SPS は切り替え可能なローカル IP アドレスを監視し、障害を検出すると、同じシステム上の別のネットワークアダプタに IP アドレスを移動します。切り替え可能な IP アドレスを使用すると、リソース階層全体をバックアップサーバにフェイルオーバーすることを回避できます。

IP ローカルリカバリ機能には、システム設定に関する次の要件と制限があります。

IP ローカルリカバリの要件

IP ローカルリカバリでは、サーバ上で SPS が保護する各 IP アドレスに 1 つのバックアップネットワークインターフェースを指定できます。バックアップインターフェースを正常に機能させるには、プライマリインターフェースと同じ物理ネットワークに接続する必要があります。システム管理者は有効なインターフェースを選択する必要があります。バックアップインターフェースは 1 台のサーバに指定します。他のクラスタサーバに指定しても意味がありません。つまり、1 台のサーバでバックアップインターフェースを選択しても、他のサーバのバックアップインターフェースの選択には影響しません。

バックアップアダプタはローカルリカバリアダプタとも呼ばれ、プライミアダプタに障害が発生すると、切り替え可能なアドレスがアクティブになります。バックアップアダプタを設定するには、次の手順に従ってください。

- 2 つのアダプタを同じ物理サブネットに接続します。
- ローカルリカバリアダプタの全アドレスは、適切にルーティングできるように、プライミアダプタの実アドレスと異なる論理サブネット上に置く必要があります。また、プライミアダプタに設定して SPS で保護する仮想 IP アドレスとは異なる論理サブネットに置く必要があります。
- IP ローカルリカバリ機能を使用するには、ネットワークにゲートウェイを設置します。具体的には、システムの TCP/IP 設定の [デフォルトゲートウェイ] フィールドに、ネットワークゲートウェイのアドレスを指定します。ローカルリカバリアダプタにも同じネットワークゲートウェイを設定します。
- ケーブリングとネットワーク経路は、論理的なサブネットから ping を発行し、プライマリネットワークカードまたはローカルリカバリネットワークカードを配置したサブネットと仮想 IP アドレスに到達するように設定します。ネットワーク設定が有効なことを確認するには、論理的なサブネット上で ping を発行します。ping が失敗すれば、ネットワーク経路の設定に問題があります。
- IP ローカルリカバリ機能は、IP リソースを作成する際に有効にすることができます。IP リソースを作成した後、リソース属性を変更しても、ローカルリカバリ機能を IP リソースに追加できません。
- IP リソースの IP ローカルリカバリ機能を無効にするには、「ins_setlocalrecovery」コマンドラインユーティリティを使用してください。このユーティリティは、SPS の \bin ディレクトリ (デフォルトで C:\LK\bin) に格納されています。このユーティリティの使用方法和オプションの説明を参照するには、コマンドプロンプトから「ins_setlocalrecovery」を実行してください。

SPS 階層に IP アドレスリソースを作成して使用する場合は、[ネットワーク構成の確認](#)の説明に従って、ネットワークを設定してテストしてください。

設定が完了したら、ping を打ち、仮想 IP アドレスが一意なことを確認します。仮想 IP アドレスは必須ではありませんが、IP アドレス階層を作成すると、自動的に作成されます。

IP アドレスのリソース階層をプライマリサーバに作成するには、次の手順に従ってください。

1. サーバを選択します。次に、[サーバコンテキストメニュー](#)または[サーバコンテキストツールバー](#)の [リソース階層の作成] をクリックします。
2. [保護するアプリケーションを作成] ダイアログが起動し、クラスタにインストールされているリカバリキットの一覧が表示されます。[IP アドレス] をクリックし、[次へ] をクリックします。
3. ウィザードの指示に従って、以下の情報を入力します。ダイアログボックス内で [戻る] ボタンがア

クティブになっているときは、直前のダイアログボックスに戻ることができます。これは特に、前に入力した情報の訂正が必要なエラーが発生した場合に役立ちます。階層作成手順の間に[キャンセル]をクリックすると、SPS は作成プロセス全体を取り消します。

フ イ ー ド	説 明
IP ア ド レ ス	SPS がリソースに使用する仮想 IP アドレスまたはホスト名を指定します。クライアントアプリケーションは、この IP リソースを使用して、特定のネットワークインターフェースを介して親アプリケーションにログインします。ホスト名を使用する場合は、ホスト名がローカルコンピュータの %windir%\system32\etc\drivers\hosts ファイルに記述されているか、DNS を介してアクセス可能でなければなりません。別名とドメイン名は、上記の条件を満たしていれば使用できます。このフィールドには、デフォルトは表示されません。 注記: ホスト名を指定すると、IP リソースを拡張する際、実 IP アドレスが「TCP/IP リソース」として表示されます。
サブ ネッ ト マ スク	TCP/IP リソースがターゲットサーバで使用する IP サブネットマスクを指定します。TCP/IP リソースのアドレスクラスに対応した標準的なネットマスクを指定できます。 注記: TCP/IP リソースが使用するサブネットは、ここで選択したサブネットマスクと前記の IP アドレスによって決まるため、ネットワーク設定を統一する必要があります。
IP リ ソ ー ス タ グ 	作成している IP リソースインスタンスの一意の IP リソースタグ名を選択するか入力します。このフィールドには、リソース名または IP アドレスに一致するデフォルトのタグ名が自動的に設定されます。タグ名は変更できます。
ネッ ト ワ ー ク 接 続	物理的なイーサネットカードの IP アドレスを指定します。既存のネットワーク設定、および TCP/IP リソースのアドレスやネットマスクの値をもとに、適切なカードを選択します。デフォルト値には、前のダイアログボックスで選択したアドレスとネットマスクに最も近い IP アドレスが使用されます。
ロー カル リカ バリ のネ ット ワ ー ク 接 続	ローカルリカバリを有効にした場合、バックアップインターフェースとして使用するネットワーク接続を選択する必要があります。バックアップサーバに NIC があれば、それを指定します。バックアップサーバに NIC がなければ、プライマリサーバの NIC を指定します。

4. データの入力が完了すると、[次へ] ボタンが表示されます。[次へ] をクリックすると、リソース階層の作成と検証が実行されます。

5. リソース階層が正常に作成されたことを伝えるメッセージが表示されたら、**[次へ]**をクリックして操作を続けます。問題が発生すると情報ボックスにエラーが表示され、部分的に作成されたリソース階層が削除されて、**[次へ]** ボタンが無効になります。その場合は、**[キャンセル]**をクリックして、**ウィザード**を終了してください。

注記: チューニングソフトウェアを使用するか、仮想 IP リソースを作成した後、ネットワークカードを変更すると、仮想 IP リソースがいったん削除され、ネットワークカードのインデックス番号を反映した IP リソースが再作成されます。

IP ローカルリカバリのシナリオ

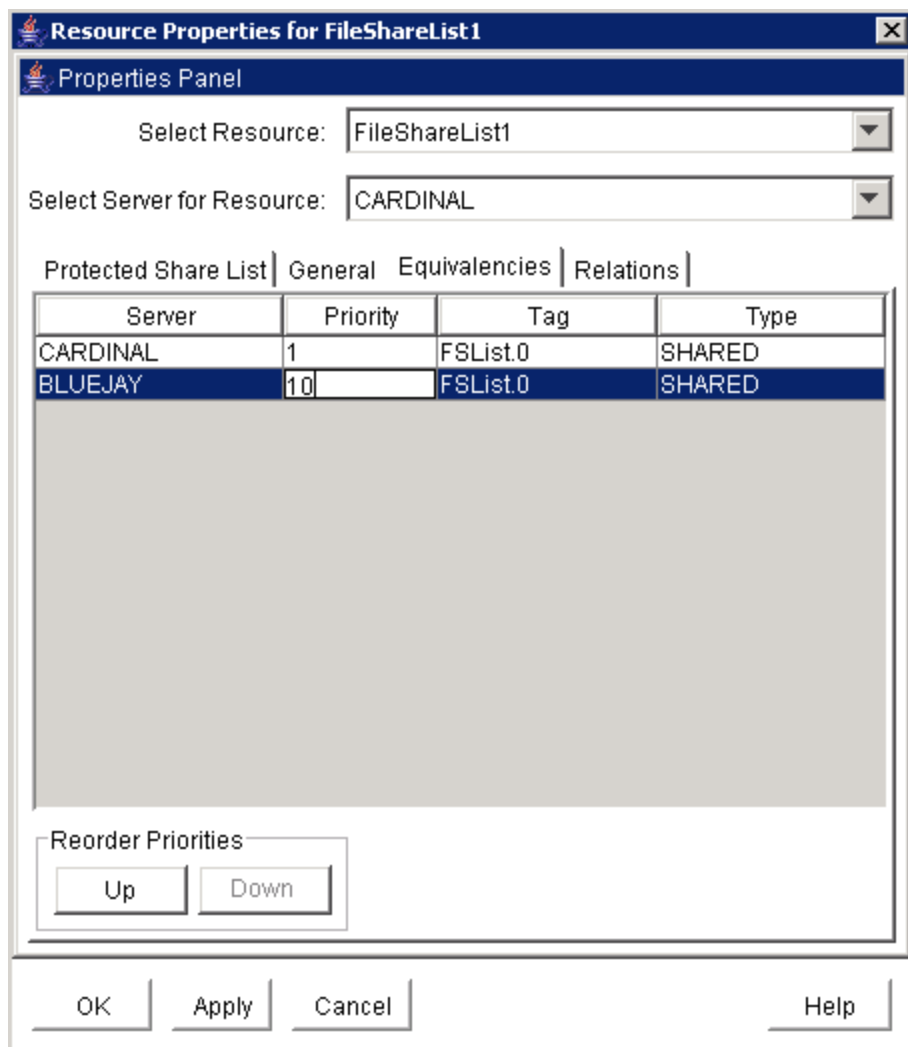
IP ローカルリカバリ機能を有効にした場合、定期的に行う IP リソースのディープチェックが失敗すると、SPS は次の処理を実行します。

- 現在のネットワークインターフェースで IP アドレスを起動します。
- 前記の処理に失敗すると、リソースインスタンスをチェックし、バックアップインターフェースが使用できるか調べます。バックアップインターフェースが使用できる場合は、IP アドレスをバックアップインターフェースに移動します。
- ローカルリカバリが完全に失敗すると、SPS は、IP アドレスと依存リソースをバックアップサーバにフェイルオーバーします。

ローカルリカバリ機能は、バックアップアダプタがなくても有効にすることができます。ローカルリカバリ機能を有効にすると、SPS はプライマリネットワークのインターフェースを再起動しますが、それに失敗すると、リソースをバックアップサーバにフェイルオーバーします。

リソース優先度の編集

リソース階層を定義したサーバの優先度は、編集したり並び替えたりすることができます。最初に、[リソースプロパティの表示](#)で説明した手順に従って、**[リソースのプロパティ]** ダイアログボックスを開いてください。**[リソースのプロパティ]** ダイアログボックスが開き、選択したサーバのリソースの優先度が**[イクイバレンス]** タブに表示されます (次図)。



優先度は次の方法で変更できます。

- [上へ] または [下へ] をクリックしてイクイバレンシを移動し、優先度を並び替える。
- 優先度を直接編集する。

[上へ] ボタンと[下へ] ボタンの使用

1. イクイバレンシ表の行をクリックして、イクイバレンシを選択します。選択したイクイバレンシに応じて、[上へ] ボタンまたは[下へ] ボタンが有効になります。[上へ] は優先度が最高のサーバ以外を選択すると有効になります。[下へ] は優先度が最低のサーバ以外を選択すると有効になります。
2. [上へ] または [下へ] をクリックし、イクイバレンシを優先順位リスト内で移動します。

優先度自体は変わりませんが、イクイバレンシが表内を上下に移動します。

優先度の編集

1. **イクイバレンシ表**の**[優先度]**列にある優先度をクリックして優先度を選択します。優先度の近くにボックスが表示され、値がハイライト表示されます。
2. 優先度を入力し、**Enter** キーを押します。

注記: 優先度の有効値は **1 ~ 999** です。

優先度の編集が完了すると、**イクイバレンシ表**が並び替えられます。

変更内容の適用

イクイバレンシ表の優先度を並び替えたら、**[適用]**または**[OK]**をクリックして、変更内容を反映してください。**[適用]**をクリックすると、変更が反映されます。**[OK]**をクリックすると、変更内容が適用されてウィンドウが閉じます。**[キャンセル]**をクリックすると、**[適用]**をクリックした後に変更した内容を保存せずにウィンドウが閉じます。

不完全なリソース優先度の変更

SPSの階層は、リソースに親子関係を割り当てて定義します。リソースが複数の親を持つ場合、階層の全ルートを表示しても、親リソースを簡単に見つけられません。階層の整合性を保持するために、SPSは優先度の変更を全クラスタサーバに反映します。GUIは、**[OK]**または**[適用]**がクリックされると、選択された階層の全ルートを表示して、優先度を統一することを要求します。この時点では、すべてのルートを受け入れるか、操作を取り消します。ルートの一覧を受け入れると、新しい優先度が階層内の全リソースに適用されます。

[リソースプロパティ]ダイアログボックスが表示されている間に、階層を変更しなかったことを確認してください。**[リソースプロパティ]**ダイアログボックスで優先度を編集する前に、SPSに加えた変更は、このダイアログボックスに動的に反映されます。SPSで優先度を変更しても、ダイアログボックスの値は変更されません。**[適用]**または**[OK]**をクリックすると、優先度を変更したことを伝えるメッセージが表示されて、操作が一時的に止まります。

複数の優先度を一括して変更し、回復不能なエラーが発生しないように、SPSは1回の操作で、サーバ1台に1件ずつ優先度を変更します。また、優先度が競合しないように、ユーザが指定した変更を暫定値として扱います。暫定値として最大値999を超える値を許容し、ユーザが優先度を変更している間、暫定値をGUIに表示します。優先度の変更操作が完了すると、暫定値はユーザが要求した値に置換されます。エラーが発生して、優先度がロールバックできない場合、暫定値が残ることがあります。このようなエラーが発生した場合は、次の手順に従って、階層を修復してください。

階層の一貫性を復元する

優先度の変更中にエラーが発生し、操作を完了できなかった場合、優先度に矛盾が生じます。エラーは、システムやコミュニケーションパスの障害など、さまざまな原因で発生します。優先度の変更操作を開始してから、操作を完了する前にエラーが発生し、前の状態にロールバックできない場合、その旨を通知するメッセージが表示されます。こうした状態が発生した場合は、次の手順に従って、階層の一貫性を回復してください。

1. 可能な限り、問題の原因を突き止めてください。システムやコミュニケーションパスに障害が発生していないか調べてください。優先度管理プログラムの実行中に、別の操作を同時に実行しなかったことを確認してください。
2. 操作を続ける前に、問題を可能な限り解決してください。例えば、階層を復元する前に、障害が発生したシステムやコミュニケーションパスを修復します。
3. [リソースプロパティ] ダイアログボックスを使用して、優先度の変更を再試行してください。
4. [リソースプロパティ] ダイアログボックスで優先度を変更できない場合は、`hry_setpri` をコマンドラインから実行して、階層を修復してください。`hry_setpri` スクリプトを使用すると、GUI を介さずに、サーバごとに優先度を変更できます。
5. 修復作業を完了したら、すべてのクラスタサーバに `eqv_list` コマンドを実行し、各サーバ上の構成データベースが整合していることを確認し、各サーバが返す優先度を調べてください。
6. 以上の操作でも階層を修復できない場合は、最後の手段として、階層を削除して再作成してください。

リソースプロパティの編集

1. リソースのプロパティを編集するには、[リソースプロパティの表示](#)と同じ手順で [リソースのプロパティ] ダイアログボックスを開きます。
2. 適切なアクセス権でサーバにログインすると、次の項目を編集できます。
 - スイッチバックタイプ
 - リソース設定 (特殊な構成を設定したリソースのみ)
 - リソースのプロパティ
3. 編集が完了すると、[適用] ボタンが有効になります。このボタンをクリックすると、変更内容が反映されます。
4. 作業が完了したら、[OK] をクリックして、変更内容を保存してウィンドウを閉じるか、[キャンセル] をクリックして、変更を適用せずに、ウィンドウを閉じます。

リソース階層の拡張

SPS の [Extend Resource Hierarchy] オプションを使用すると、既存のリソース階層を別のサーバにコピーし、同じリソース階層をコピー先のサーバに作成できます。リソース階層を他のサーバに拡張すると、カスケーディングフェイルオーバーが可能になります。コピー元のリソース階層を持つサーバをテンプレートサーバ、リソース階層を拡張するサーバをターゲットサーバと呼びます。

ターゲットサーバは拡張した階層を起動し、アクティブなコミュニケーションパスを介して、他のリモートサーバ上のイクイバレンスと通信できなければなりません。つまり、既存のリソース階層に関連付けたりカバリキットは、ターゲットサーバを含めて階層が常駐する全サーバにインストールする必要があります。

注記: 新しいリソース階層を作成すると、階層を拡張するか確認するメッセージが表示されます。

1. 既存のリソース階層を拡張する場合は、拡張するリソース階層を選択し、[リソースコンテキストメニュー](#)または[リソースコンテキストツールバー](#)の [Extend Resource Hierarchy] をクリックします。

2. **バックアップサーバ**を選択し、**次へ**をクリックします。
3. SPS による拡張事前検査の結果がダイアログボックスに表示されます。検査に成功した場合は、拡張事前検査スクリプトが正常に実行されたことを示すメッセージが表示されます。各項目が適切なことを確認し、**次へ**をクリックし、拡張するリソース階層に必要なデータを指定します。

注記: 複数のルートを持つ階層は、一括して拡張する必要があります。複数ルートの階層は、単一のルート階層としては拡張できません。

DNS リソース階層の拡張

このダイアログボックスは、[DNS リソース階層の作成](#)を完了するか、既存の DNS リソース階層を拡張すると、自動的に開きます ([リソース階層の拡張](#)を参照)。DNS リソース階層を拡張するには、次の情報を入力してください。

フィールド	説明
IP アドレス	保護するプライマリサーバまたは別名に関連付けられた A レコードの IP アドレスを入力します。DNS リソースがこのサーバで起動した場合、レコードはこの IP アドレスで更新されます。
DNS サーバ名 (完全修飾ドメイン名)	リソースレコードを変更する DNS サーバの完全修飾ドメイン名 (FQDN) を <DNS ServerName>.<mydomain>.com 形式で入力します。DNS サーバは、可能であれば、同じサイトのバックアップサーバからアクセスできるようにします。

ファイル共有リソース階層の拡張

このダイアログボックスは、[ファイル共有リソース階層の作成](#)を完了すると自動的に開きます。既存のファイル共有リソース階層からも開けます ([リソース階層の拡張](#)を参照)。ファイル共有リソース階層を拡張する場合、データを追加する必要はありません。

Generic Application リソース階層の拡張

このダイアログボックスは、[Generic Application リソース階層の作成](#)を完了すると自動的に開きます。既存の Generic Application 階層からも開けます ([リソース階層の拡張](#)を参照)。Generic Application で作成したリソース階層を拡張する場合、データを追加する必要はありません。

LAN Manager リソース階層の拡張

このダイアログボックスは、[LAN Manager リソース階層の作成](#)を完了すると自動的に開きます。既存の LAN Manager リソース階層からも開けます ([リソース階層の拡張](#)を参照)。LAN Manager リソース階層を拡張する場合、データを追加する必要はありません。

ボリュームリソース階層の拡張

この操作は、[ボリュームリソース階層の作成](#)を完了するか、既存のボリュームリソース階層を拡張する

と、自動的に開きます ([リソース階層の拡張](#)を参照)。ボリュームリソース階層を拡張するには、次のデータを追加してください。

フィールド	説明
ボリュームタイプ	共有ストレージを使用する場合は [共有ディスク] を選択します。ミラーが存在しない状態で SteelEye DataKeeper を使用する場合は [ミラーの作成] を選択します。ミラーが存在している状態で SteelEye DataKeeper を使用する場合は、 [既存のミラー] を選択します。
ネットワークのエンドポイント (ターゲット/ソース)	ボリュームタイプとして [ミラーの作成] または [既存のミラー] を選択した場合は、ミラーのエンドポイントを選択します。エンドポイントには IP アドレスを指定します。
モード	ボリュームタイプとして [ミラーの作成] を選択した場合は、ミラーのモードを選択します。 非同期ミラー: ソースへの書き込みは、キューに入ってからターゲットに転送されます。制御は即時に戻ります。同期ミラーに比べて、信頼性は低下しますが、ソースへの書き込みは高速になります。 同期ミラー: ソースボリュームへの書き込みは、ターゲットボリュームで即時にコミットされます。信頼性は高くなりますが、パフォーマンスが低下します。
ボリュームリソースをクラスタ内の 3 つ目のシステムに拡張する場合は、クラスタ内のイクイバレンシシステムにボリュームタイプを指定する必要があります。	
ボリュームタイプ (共有または SteelEye DataKeeper)	イクイバレンシシステム間のミラーリングには、 [共有ディスク] またはネットワークのエンドポイントを選択します。

注記: ボリュームリソース階層を削除すると、LifeKeeper GUI で作成したミラーも削除されます。ミラー削除を回避するには、[LifeKeeper ミラー削除フラグ](#)を **False** に設定してください。

IP リソース階層の拡張

このダイアログボックスは、[IP アドレスリソース階層の作成](#)を完了するか、既存の DNS リソース階層を拡張すると、自動的に開きます ([リソース階層の拡張](#)を参照)。IP アドレスリソース階層を拡張する場合は、次の情報を入力してください。

フィールド	説明
サブネットワークマスク	ターゲットサーバの IP リソースに使用するサブネットマスクを入力します。テンプレートサーバのサブネットマスクがデフォルトとして表示されます。

フィールド	説明
ネット ワーク 接続	ターゲットサーバで使用するネットワーク接続を選択します。
ター ゲット の リス トア モード	<p>IP アドレスのリストア機能は、同じサブネットにある LAN ノード 2 台と異なるサブネット上にある WAN ノード 1 台で構成する 3 ノードクラスターに適用されます。IP リソースのリストアモードは、LAN ノードでは有効、WAN ノードでは無効にそれぞれ設定されます。</p> <p>ターゲットサーバの IP リソースに適用するリストアモードを選択します。状況によっては、保護対象の IP アドレスがターゲットシステムで使用できないことがあります。例えば、リモートのターゲットシステムがクラスターと異なるサブネットに接続されている場合、保護対象の IP アドレスは使用できません。その場合、リストアモード「無効」で拡張することができます。リストアモード「無効」を指定すると、SPS は、リソースを起動する際、ターゲットシステムの IP アドレスを設定せずに、IP アドレスが無効なことを監視します。このようなケースでは、ネットワーク上のリダイレクトは、SPS DNS リソースなどで実装します。ターゲットシステムの [IP リソースのプロパティ] ページを使用すると、設定を後で変更できます。詳細については、IP リソース管理を参照してください。</p>
ター ゲット の ロー カル リカ バリ	ターゲットサーバの IP ローカルリカバリ を有効にする場合は、 [はい] をクリックします。有効にしない場合は、 [いいえ] をクリックします。
ター ゲット の ロー カル リカ バリの ネ ット ワー ク接 続	ローカルリカバ리를有効にした場合、バックアップインターフェースとして使用するネットワーク接続を選択する必要があります。バックアップサーバに NIC があれば、それを指定します。バックアップサーバに NIC がなければ、プライマリサーバの NIC を指定します。

注記: 無効な IP リソースは別のシステムに拡張できません。IP リソースを拡張するときは、別のシステム上で ping コマンドを発行して、IP リソースに到達可能なことを確認してください。無効な IP リソースは、起動することは可能ですが、ping コマンドには応答しません。

リソース階層の拡張解除

[リソース階層の拡張解除] オプションは、すべてのリソースを対象として単一のサーバからすべての階層を削除します。すべてのサーバから階層を削除する [\[リソース階層の削除\]](#) オプションとは、動作が異なります。

[リソース階層の拡張解除] を使用する場合、既存の階層を削除するサーバがターゲットサーバとなります。

[リソース階層の拡張解除] は、ターゲットサーバとのコミュニケーションパスがアクティブな SPS サーバで使用できます。

1. 拡張を解除するサーバ固有のリソースインスタンスを選択し、[リソースコンテキストメニュー](#)または[リソースコンテキストツールバー](#)の**[リソース階層の拡張解除]**をクリックします。
2. 拡張を解除するサーバとリソース階層を確認するメッセージが表示されます。**[拡張解除]**をクリックして拡張を解除します。
3. [アウトプットパネル](#)が有効な場合は、ダイアログボックスが閉じて、リソース階層の拡張を解除した結果がアウトプットパネルに表示されます。すべての結果が表示されたら、**[完了]**をクリックしてダイアログボックスを閉じます。

リソース依存関係の追加

ほとんどの場合はリカバリキットで元のリソース階層の作成作業を行うときに依存関係が作成されますが、特定の状況では、新しいリソース依存関係を作成したり、既存の依存関係を削除したりする必要がある場合があります。例えば、既存のIP依存関係を別のIPアドレスに変更する場合などです。リソース階層全体を削除してから新しく作成する代わりに、既存のIP依存関係を削除して、別のIPアドレスで新しい依存関係を作成することができます。

1. 子の依存関係を追加する親として、サーバ固有のリソースインスタンスを選択します。次に、[リソースコンテキストメニュー](#)または[リソースコンテキストツールバー](#)から**[依存関係の作成...]**を選択します。
2. ドロップダウンボックスから**[親リソース IP アドレス]**を選択します。**[次へ]**をクリックします。
3. サーバ上の既存の有効なリソースのドロップダウンボックスから、**[子リソースタグ]**を選択します。以下のリソースを除いて、サーバ上で使用できるリソースがすべてダイアログに表示されます。
 - 親リソース、その祖先および子。
 - 親リソースと同じサーバに拡張されていないリソース。
 - 親リソースと同じ相対優先順位ではないリソース。
 - 親リソースがサービス中の場合、親と同じサーバでサービス中ではないリソース。

[次へ]をクリックして、次のダイアログに進みます。

4. ダイアログには、依存関係を作成するために適切な親子のリソースタグを選択したかどうかの確認が表示されます。**[依存関係の追加]**をクリックして、親が拡張されたクラスタ内のすべてのサーバ上に依存関係を作成してください。

5. [アウトプットパネル](#)が有効な場合、ダイアログボックスが閉じて、依存関係作成コマンドの実行結果がアウトプットパネルに表示されます。すべての結果が表示されたら、**[完了]**をクリックしてダイアログボックスを閉じます。

リソース依存関係の削除

1. 子の依存関係を削除する親として、サーバ固有のリソースインスタンスを選択します。次に、[リソースコンテキストメニュー](#)または[リソースコンテキストツールバー](#)から**[依存関係の削除]**を選択します。
2. ドロップダウンボックスから**[子リソース]**を選択します。これは削除する依存関係の子の名前です。**[次へ]**をクリックして、次のダイアログボックスに進みます。
3. ダイアログには、依存関係を削除するために適切な親子のリソースタグを選択したかどうかの確認が表示されます。**[依存関係の削除]**をクリックすると、クラスタ内の全サーバの依存関係が削除されます。
4. [アウトプットパネル](#)が有効な場合、ダイアログボックスが閉じて、依存関係削除コマンドの実行結果がアウトプットパネルに表示されます。すべての結果が表示されたら、**[完了]**をクリックしてダイアログボックスを閉じます。

全サーバのリソース階層の削除

1. 削除するサーバ固有のリソースインスタンスを選択し、[リソースコンテキストメニュー](#)または[リソースコンテキストツールバー](#)の**[リソース階層の削除]**をクリックします。
2. 削除するリソース階層を確認するメッセージが表示されます。**[削除]**をクリックして、リソース階層を削除します。最初に選択したサーバから順番にリソース階層が削除されます。
3. [アウトプットパネル](#)が有効な場合、ダイアログボックスが閉じて、階層削除コマンドの実行結果がアウトプットパネルに表示されます。すべての結果が表示されたら、**[完了]**をクリックしてダイアログボックスを閉じます。

マニュアルページ

LCD - その他の LCD プログラム

構文

[lcdremexec](#) [-e] -d destname -- cmd [arg1 arg2 ... argn]

[lcdsync](#) [-d destname]

[lcdrecover](#) -g {remote|restore|delete} -- [arg1 arg2 ... argn] | -G {remote|restore|delete} -- [arg1 arg2 ... argn] | -p primarytest /| [-o resource]

`lcdrcp file1 file2 file3 ...{dest:ofile | dest:odir}`

`lkstart [-w waitperiod]`

`lkstop [-f or -r|-n]]`

説明

上記のプログラムは、アプリケーション開発者がさまざまな目的で使用できます。各プログラムは %LKROOT%\bin ディレクトリに格納されています。

終了コード

前記のコマンドは、次の終了コードを返します。

0	処理に成功した。
1	システムコールまたはライブラリコールが内部エラーを返した。
2	ユーザが指定した構文にエラーがある。
3	LifeKeeper の内部エラー。
4	既存のオブジェクトに対する処理を要求した。
5	不正な引数を渡した。
6	インデックスが有効な範囲を超えている。
7	存在しないオブジェクトに要求を発行した。
8	リソースインスタンスの削除を要求したが、そのリソースインスタンスには、削除していないリソースインスタンスが依存している。
9	相手のシステムと通信できない。

lcdrcp

`lcdrcp file1 file2 file3 ... {dest:ofile | dest:odir}`

lcdrcp は、LifeKeeper コミュニケーションパスを使用して ASCII ファイル *file1 file2 file3 ...* を別のシステムに転送する汎用プログラムです。**lcdrcp** を使用してバイナリファイルをコピーすることはできません。

ファイルは *dest* のディレクトリ *odir* に転送されます。送信するファイルが1つだけの場合には、システム *dest* 上の位置 *ofile* に送信先のファイル名を含めた形式を使用できます。送信先の引数に Windows のドライブ名 (D: など) を使用する場合は特に注意してください。送信先名がない場合に誤って送信先の名前として解釈される可能性があります。ただし、送信先システム名を指定した場合、ドライブ名は正常に解釈されます。

lcdrecover

`lcdrecover -g {remove|restore|delete} -- [arg1 arg2 ... argn] | -G`

```
{remote|restore|delete} -- [arg1 arg2 ... argn] | -p primarytest | [-o resource]
```

`-g` オプションは、`remove`、`restore`、`delete` のうちいずれかの引数を取ります。このオプションによって、指定した引数に対するグローバル前処理スクリプトが実行されます。グローバル前処理スクリプトは、特定のイベントの前に実行するようにアプリケーションによって登録されます。例えば、このオプションに `restore` 引数を指定すると、LifeKeeper によって登録された `restore` 前処理スクリプトが実行されてから、すべてのアプリケーションによって登録された `restore` 前処理スクリプトが実行されます。通常、`perform_action` ([LRACI perform_action](#) を参照) によって `restore` 前処理スクリプトが自動的に実行されます。ただし、`perform_action` に `-G` オプションが指定された場合を除きます。

`perform_action` の `-G` オプションを使用すると、最初に `lcdrecovery -g restore` を使用した `perform_action` の実行の前に 1 回だけグローバル前処理スクリプトを実行して、複数の `perform_action` コマンドを実行できます。アプリケーションは、以下のパスにスクリプトをインストールして、グローバル前処理スクリプトを登録できます。

```
%LKROOT%\ subsys\<<appname>\actions\prerestore.ksh
```

`arg1`, `arg2`, ... `argn` は実行時にグローバル前処理スクリプトに渡される引数です。

`remove` 引数についても同様のスクリプト (`remove` 前処理スクリプト) が存在し、`perform_action -G -a remove` が実行される前に実行できます。実行されるのは、`lcdrecovery -g remove` の実行時です。

`delete` 前処理スクリプトも同様ですが、`ins_remove` の `-G` が省略された場合を除いて、`ins_remove -G ...` プログラムの前に実行されます ([LCDI インスタンス](#) を参照)。

`lcdrecovery` の `-G` オプションは `-g` に類似していますが、グローバル後処理スクリプトを実行するように指定する点異なります。`-G` オプションは、その前の `lcdrecovery -g arg` を実行せずに使用しないでください。また、すべての `perform_action` プログラムまたは `ins_remove` プログラムが実行された後に実行する必要があります。`getlocks` によって保護された領域内で `-G` オプションを実行する場合 (`getlocks` の後、`rislocks` の前)、`arg1` に `-m` を設定して、`getlocks` の 2 つ目のインスタンスが実行されないようにしてください。2 つ目のインスタンスが実行されると、処理がハングします。

以下の例では、複数の `perform_action` コマンドを実行します。グローバル前処理スクリプトとグローバル後処理スクリプトは 1 回だけ実行されます。

```
lcdrecovery -g restore

# run "preglobal" restore scripts

perform_action -G -a restore -t tagname

# neither scripts are run

perform_action -G -a restore -t tagname2

# neither scripts are run

lcdrecovery -G restore -- -m

# run "postglobal" restore scripts

# use -m arg when in getlocks protected region of code
```

この例では、複数の `restore` 前処理スクリプトと `restore` 後処理スクリプトが実行されます。

```
perform_action -a restore -t tagname

# all scripts once

perform_action -a restore -t tagname2

# all scripts again
```

lcdrecover の **-p** オプションは、特定のリソースがプライマリシステムまたはセカンダリシステムのどちらのリソース階層にあるかを判定するために使用されます。リソースタグ名に **primary test** を指定すると、リソースがプライマリ階層に存在する場合は文字列 **primary** が、存在しない場合は **secondary** が標準出力に出力されます。

-o オプションを使用すると、指定したリソースタグに関連付けられたリモートシステムを取得できます。

lcdremexec

```
lcdremexec [-e] -d destname -- cmd [arg1 arg2 arg3 ... argn]
```

このプログラムは、LifeKeeper コミュニケーションパスを介してシステム *destname* にリモート要求を送信し、引数 *arg1 arg2 arg3 ...* を用いてコマンド **cmd** をリモートで実行します。また、リモートコマンドの標準出力および標準エラー出力を **lcdremexec** コマンドの標準出力に返します。リモートコマンドの終了コードは **lcdremexec** によって返されます。

注記: *destname* がカレントシステムである場合、メッセージは送信されず、**lcdremexec** によってローカルで実行されます。

-e オプションを指定すると、リモートコマンドの標準出力と標準エラー出力が分割され、最初にリモートコマンドの標準出力が **lcdremexec** の標準出力に出力されてから、リモートコマンドの標準エラー出力が **lcdremexec** コマンドの標準エラー出力に出力されます。このオプションはローカルコマンドには効果がなく、標準出力と標準エラー出力は変わりません。

cmd には、Korn シェルスクリプトまたは Win32 実行可能ファイルのいずれかを指定できます。*destname* の **%LKROOT%** を現在の作業ディレクトリとして実行されるので、**%LKROOT%** に対する相対パス名を指定できます。

実行前に、*destname* の **PATH** 変数の先頭には常にディレクトリ **%LKROOT%\BIN** が追加されます。*destname* がダウンしているか実行中にダウンした場合には、**lcdremexec** は 0 以外の終了コードを返します。

lcdsync

```
lcdsync [-d destname]
```

共有メモリに保存された LifeKeeper リソース階層構成とコミュニケーションパスのステータスデータが変更されているかどうかをチェックします。異なる場合、データは「同期を取って」ディスクに書き込まれます。そのため、このプログラムから復帰したとき、ディスク上に正常なデータがあることが保証されます。*destname* を指定しないと、カレントシステムが対象となります。

注記: リソース階層構成またはコミュニケーションパスの変更を使用されるコマンド (例: **ins_create**、**dep_create**、**ins_setit**、**eqv_remove**) は、共有メモリセグメントのみを変更し、**lcdsync** プログラムを実行するまで、LifeKeeper の恒久的なファイルストレージには反映されません。

lkstart

```
lkstart [-w waitperiod]
```

このプログラムは、LifeKeeper が実行されていない場合にカレントシステム上で LifeKeeper を起動します。lkstart を実行すると、LifeKeeper デーモンが停止した場合に再起動されるように、LifeKeeper デーモンに属する %LKROOT%\etc\LKinit.config ファイルのエントリが修正されます。

-w オプションを使用すると、タイムアウト間隔を変更できます。waitperiod には、秒数を指定します。起動前の待機時間を指定するには -w 引数を使用します。

LifeKeeper サービスは、管理ツールにある Services mmc を使用して起動することも、「sc start LifeKeeper」または「net start LifeKeeper」のいずれかを使用してコマンドプロンプトから起動することもできます。

注記: このプログラムはコンソールから実行する必要があります。

lkstop

```
lkstop [-n] [-f] [-r]
```

このスクリプトは、LifeKeeper が実行されている場合にシステム上の LifeKeeper をシャットダウンします。LifeKeeper はシステム起動時に自動的に再起動されます。

lkstop の各オプションを入力したときの LifeKeeper の動作を以下の表に示します。

コマンドライン	動作
lkstop	サービス中のリソースはサービスから削除されますが、バックアップサーバにはスイッチオーバーされません。
lkstop -n	オプションを指定しない lkstop と同じです。
lkstop -f	サービス中のリソースはサービスから削除されません。
lkstop -r	-f と同じです。

LifeKeeper のサービスは、Windows の [コントロールパネル] にある [管理] タスクの [サービス] ツールを使用して停止することもできます。

LCDI アプリケーション

構文

```
app\_create [-d destsys] -a appname
```

`app_remove [-d destsys] -a appname`

`app_list [-d destsys]`

説明

LifeKeeper アプリケーションは、関連するリソースタイプのグループです。アプリケーションが削除されると、その下にインストールされたリソースタイプもすべて削除されます。

これらのプログラムは、構成データベースに新しいアプリケーションを生成したり、既存のアプリケーションを削除したりするためのインターフェースを提供します。どのコマンドも、正常に実行された場合は 0 を返します。失敗した場合は 0 以外のコード（「終了コード」を参照）を返し、エラーメッセージを標準エラーに出力します。

終了コード

前記のコマンドは、次の終了コードを返します。

0	処理に成功した。
1	システムコールまたはライブラリコールが内部エラーを返した。
2	ユーザが指定した構文にエラーがある。
3	LifeKeeper の内部エラー。
4	既存のオブジェクトに対する処理を要求した。
5	不正な引数を渡した。
6	インデックスが有効な範囲を超えている。
7	存在しないオブジェクトに要求を発行した。
8	リソースインスタンスの削除を要求したが、そのリソースインスタンスには、削除していないリソースインスタンスが依存している。
9	相手のシステムと通信できない。

app_create

`app_create [-d destsys] -a appname`

新しいアプリケーションを作成します。アプリケーションは、*appname* という名前を使用して、システム *destsys* に関連付けられます。*destsys* を指定しない場合、アプリケーションはローカルに作成されます。

app_list

`app_list [-d destsys]`

app_remove

システム `destsys` 上で LifeKeeper と連携するようにソフトウェアをインストールしたアプリケーションのリストを、標準出力に表示します。 `destsys` を指定しないと、カレントシステムが対象となります。

app_remove

```
app_remove [-d destsys] -a appname
```

既知のアプリケーションの構成データベースセットから、指定したアプリケーションを削除します。このアプリケーションに関連するリソースタイプ、リソースインスタンス、およびイクイバレンシも、すべて削除されます。構成データベースでアプリケーションが既知ではないためにエラーが発生する場合があります。

LCDI インスタンス

構文

```
ins\_gettag [-d destsys] -i id
```

```
ins\_create [-d destsys] -a appname -r restyp [-I{AUTORES_ISP|INIT_ISP|INIT_OSU}] [-v info] -t tag  
-i id [-Q quickChkInt] [-DdeepChkInt] [-l localRecover{Y/N}] [-s AUTOMATIC/INTELLIGENT]
```

```
ins\_remove [-d destsys] [-R roottag] [-a appname] [-r restyp] [-ttag] [-i id] [-v] [-I] [-N] [-G]
```

```
ins\_setin [-d destsys] -t tag [-v info]
```

```
ins\_setit [-d destsys] -t tag -I {AUTORES_ISP|INIT_ISP|INIT_OSU}
```

```
ins\_setst [-d destsys] -t tag -S {ISP|SU|OSU} [-R reason] [-A]
```

```
ins\_list [-d destsys] [-fC] [-R top] [-a appname] [-r typ] [-t tag] [-i id]
```

```
ins\_setchkint [-d destsys] -t tag -c {q=quick|d=deep} -vinterval
```

```
ins\_setlocalrecover [-d destsys] -t tag -I {Y=enable|N=disable}
```

```
ins\_setas [-d destsys] -t tag -s {INTELLIGENT|AUTOMATIC}
```

説明

リソースは、システムが認識しているボリューム、アプリケーション、またはシステムオブジェクトを表すために LifeKeeper によって使用されます。リソースタイプはリソースの分類です。リソースインスタンスはリソースタイプの実際のインスタンスです。例えば、リソースタイプには、ファイルシステムボリューム、ファイル共有、IP アドレス、LAN Manager 名、および SQLServer などの各種サーバが含まれます。一般のユーザ定義可能タイプを使用すると、ユーザはカスタム耐障害セットアップを構築できます。あるリソースタイプに対して、複数のインスタンスが存在する場合があります。

リソースインスタンスはいくつかの状態 で存在します。これらの状態は以下の値と意味を持ちます。

ISP	リソースはサービス中であり、保護されています。ISP はプライマリノード上のリソースの標準状態です。
OSU	サービスを停止していますが、障害は発生していません。remove スクリプトを実行してサービスが停止されたため、このシステムではリソースを使用できません。OSU 状態は、OSF 状態または OSU 状態の子に依存関係を持つオブジェクトに対して、あるいはバックアップマシン上の同等オブジェクトが ISP 状態または ISU 状態にある場合にも使用されます。OSU はセカンダリノード上のリソースの標準状態です。
OSF	障害によりサービスを停止しています。オブジェクトを復旧しようとして障害が発生したため、このシステムではリソースを使用できません。

終了コード

どのコマンドも、正常に実行された場合は 0 を返します。失敗した場合は 0 以外のコードを返し、標準エラーに出力します。前記のコマンドは、次の終了コードを返します。

0	処理に成功した。
1	システムコールまたはライブラリコールが内部エラーを返した。
2	ユーザが指定した構文にエラーがある。
3	LifeKeeper の内部エラー。
4	既存のオブジェクトに対する処理を要求した。
5	不正な引数を渡した。
6	インデックスが有効な範囲を超えている。
7	存在しないオブジェクトに要求を発行した。
8	リソースインスタンスの削除を要求したが、そのリソースインスタンスには、削除していないリソースインスタンスが依存している。
9	相手のシステムと通信できない。

ins_list

```
ins_list [-d destsys] [-fC] [-R top] [-a appname] [-r typ] [-t tag] [-i id]
```

このコマンドは、リソースインスタンスのセットに関する行を標準出力に表示します。各行には、特定のリソースインスタンスに関する現在の情報がすべて含まれています。行の例を以下に示します。

```
LKSYS1-filesys-volume-ISSUTILS-K:--ISP-restore action hassucceeded-
AUTORES_ISP--0-0-180-300-0
```

各行のフィールドは区切り文字で区切られています。デフォルトの区切り文字は ^A (\001) です。-fC オプションを指定した場合、区切り文字は指定した文字に変更されます。上記の例では、区切り文字としてダッシュ(-)が表示されています。上記のコマンド例のフィールドは以下のとおりです。

LKSYS1	リソースインスタンスが存在するシステムの名前。
filesystem	リソースタイプのアプリケーション名。
volume	リソースタイプ名。
ISSUTILS	ユーザ定義のリソースインスタンスタグ識別子。
K:	リソースインスタンスに対する LifeKeeper の内部識別子。
--	このフィールドが(例のように)空ではない場合、タイプに応じて、追加のインスタンス情報が表示されます。
ISP	リソースインスタンスの現在の状態。ISP、ISU、OSU、または OSF と表示されます。
restore action has succeeded	状態が最後に変更された理由。
AUTORES_ISP	使用可能なリソース初期化オプションは、AUTORES_ISP、INIT_ISP、INIT_OSU です。
--	このフィールドが(例のように)空ではない場合、リソースが現在、以下の理由で予約されていることを示します。 RESTORE: リソースをサービス開始にする REMOVE: リソースをサービスから削除する RECOVER: リソースのローカルリカバリを実行する
0	リソースを予約したプロセスのプロセス ID。
0	予約済み。
180	クイックチェック間隔 (秒単位)。
300	ディープチェック間隔 (秒単位)。
0	ローカルリカバリ。0 = 無効、1 = 有効。

その他の引数を指定すると、リストに含まれるリソースインスタンスの数が制限されます。引数を使用しない場合は、`destsys` のすべてのリソースが列挙されます。以下は制限するための引数です。

`destsys`: `destsys` を指定しないと、カレントシステムが対象となります。指定した場合は、リモートシステムのデータが表示されます。

`top`: `top` が空白文字列 "" の場合は、ルートリソースのみが表示されます。`top` に空白以外が指定された場合は、`top` リソースおよびその下のすべての子リソースが再帰的に表示されます。

`appname`: `appname` を指定した場合は、このアプリケーションで定義されたすべてのリソースタイプに関連付けられたすべてのリソースインスタンスが表示されます。`appname` を指定しない場合は、システム上で定義されたすべてのアプリケーションのすべてのリソースインスタンスが表示されます。

`typ`: `typ` を指定した場合は、アプリケーション `appname` のタイプ `typ` のリソースインスタンスがすべて表示されます。

`tag` または `id`: `tag` または `id` を指定した場合は、その `tag` または `id` に関連付けられたリソースインスタンスが表示されます。

初期化の方法

デフォルトの「Auto ISP 初期化方法」を使用することを推奨します。LifeKeeper が起動時に実行する初期化を次表に示します。

Autores ISP	ペアノードでリソースが起動していないと、リソースを自動的に起動します。
Init ISP	リソースを常に ISP 状態に初期化します。
Init OSU	リソースを常に OSU 状態に初期化します。

初期状態

状態とは、リソースの現在の処理ステータスを指します。例えば、プライマリシステムのリソースは、LifeKeeper で保護された状態「**ISP** (サービス中)」が標準の状態です。一方、セカンダリシステムのリソースは、サービスを休止した「**OSU**」が標準の状態です。

デフォルトの初期状態を使用することを推奨します。初期状態を **OSU** に設定した場合は、リソースを手動で起動する必要があります。

ins_create

```
ins_create [-d destsys] -a appname -r restyp [-I {AUTORES_ISP|INIT_ISP|INIT_OSU}] [-v info] -t tag -i id [-Q quickChkInt] [-D deepChkInt] [-l localRecover{Y|N}] [-s AUTOMATIC|INTELLIGENT]
```

destsys システムの構成データベースに新しいリソースインスタンスを定義します。リソースインスタンスは指定した引数で記述されます。*destsys* を指定しないと、カレントシステムが対象となります。このコマンドには、次の文字列タグオプションを指定できます。

- *-a* オプションと *-r* オプションは、新しいインスタンスに関連付ける既存のアプリケーションおよびソースタイプを指定します。
- *-I* オプションに指定する初期化タイプフィールドは、LifeKeeper が再起動する場合 (起動時など) にリソースインスタンスを初期化する方法を示します。
- *-v* オプションに指定する *info* 文字列には、リソースタイプに固有の追加情報を記述します。リソースタイプごとに一意である必要はありません。
- *-t* オプションに指定する文字列タグは、システム上で一意なリソースインスタンスの名前です。LifeKeeper の外部でもわかりやすい文字列を指定してください。
- *-i* オプションに指定する文字列 ID もシステムごとに一意ですが、LifeKeeper の内部でのみ意味を持ちます。
- *-Q* オプションに指定するクイックチェック間隔は秒単位です。リソースに対して **quickchk.ksh** スクリプトが存在しない場合には、この値に 0 を指定してください。LifeKeeper は、**quickchk.ksh** スクリプトを続けて実行するまでにこの期間だけ待機します。有効な値の範囲は 0 ~ 604800 です。

- `-D` オプションに指定するディープチェック間隔は秒単位です。リソースに対して `deepchk.ksh` スクリプトが存在しない場合には、この値に0を指定してください。LifeKeeperは、`deepchk.ksh` スクリプトを続けて実行するまでにこの期間だけ待機します。有効な値の範囲は0～604800です。
- ローカルリカバリオプションには、`recover.ksh` スクリプトによるリソースのリカバリを実行するかどうかを指定します。リソースに対して `recover.ksh` スクリプトが存在しない場合には、このオプションに「N」を指定してください。

ins_gettag

```
ins_gettag [-d destsys] -i id
```

名前が `destsys` のシステムで、内部識別子 `id` に対応するタグ名を標準出力に表示します。`destsys` を指定しないと、カレントシステムが対象となります。

注記: リソースのタグ名と `id` 名はシステム上で一意ですが、異なるシステム上の別のリソースインスタンスに再利用することができます。

リソースタグを使用して人間が読める名前 (例: `user-partition`) を付けると、リソースインスタンスの処理がわかりやすくなります。一方、`id` は内部記述子です。リソース名 `id` は、リソースを一意に記述するために、リソースに関連付けられたアプリケーションソフトウェアで使用されます。

ins_remove

```
ins_remove [-d destsys] [-R roottag] [-a appname] [-r restyp] [-t tag] [-i id] [-v] [-I] [-N] [-G]
```

`destsys` システムの構成データベースからリソースインスタンスを削除します。このコマンドを実行すると、リソースの依存関係とイクイバレンシも削除されます。`destsys` を指定しないと、カレントシステムが対象となります。

注記: `ins_remove` は、指定されたリソースに直接的または間接的に依存するリソースを削除してから、指定されたリソースを削除します。

リソースインスタンスを削除する場合、削除するインスタンスのリソースタイプに削除処理が定義されていると、その削除処理はインスタンスを削除する前に実行されます。

このコマンドには、次のオプションを指定できます。

<code>-R</code>	下位の階層全体と、それに依存するリソースをすべて削除します。 <code>roottag</code> 文字列に複数のリソースタグ名を区切り文字 <code>^A</code> で区切って記述すると、指定したリソースとその下位リソースを再帰的に削除できます。削除処理は、削除対象でないリソースが依存するリソースが現れるまで続きます。
<code>-a</code>	指定したアプリケーションのリソースだけを削除します。
<code>-r</code>	指定したリソースタイプのリソースをすべて削除します。
<code>-t</code> (または <code>-l</code>)	<code>tag</code> タグ、または <code>id</code> ID と一致するインスタンスとその依存リソースを削除します。

-v	削除したリソースインスタンスすべてのタグ名を標準出力に表示します (詳細モード)。
-l	リソース階層を初期化し、 ins_remove の適切な動作を保証します。このオプションは、 ins_remove を最上位レベルで再帰的に呼び出す場合に使用します。例えば delete スクリプトで ins_remove を再帰的に呼び出す場合、このオプションは不要です。delete スクリプトから ins_remove を再帰的に呼び出す場合は、 -l オプションを使用しないでください。
-N	-N オプションを指定すると、 ins_remove はリソース階層を初期化しません。上位レベルで ins_remove を呼び出すと、 -l オプションが自動的に付くためです。delete スクリプトは親である ins_remove から呼び出され、 -N オプションは階層サイクルを発生しないため、 -N オプションは delete スクリプト内部で使用する必要があります。
-G	-G オプションは、 ins_remove を呼び出す際、delete 前処理スクリプトと delete 後処理スクリプトを実行しないように指示します (LCD を参照)。 ins_remove を最上位で複数回呼び出し、 ins_remove を呼び出す前後に LCD の <code>lcdrecover-g delete</code> を使用して delete 前処理スクリプトと delete 後処理スクリプトを手動で実行する場合に、このオプションが有用です。最上位で呼び出した ins_remove は、delete 前処理スクリプトと delete 後処理スクリプトを実行するため、delete スクリプトには -G オプションを指定してください。

ins_setas

```
ins_setas [-d destsys] -t tag -s {INTELLIGENT|AUTOMATIC}
```

`destsys` システムにある `tag` タグ名が付いたルートリソースのスイッチバックタイプを、`-s` オプションで指定された方法に設定します。必ずルートリソースで使用して、ルートリソースおよびそのすべての依存リソースのスイッチバックタイプを変更してください。

ins_setchkint

```
ins_setchkint [-d destsys] -t tag -c {q=quick|d=deep} -v interval
```

`-ttag` タグ名で指定されたリソースのクイックチェック間隔またはディープチェック間隔を変更します。チェック間隔は、秒単位で入力する必要があります。

例

ファイル共有リソース `FSList.0` のクイックチェック間隔を2分間に変更する場合は、`$LKROOT\bin` から以下のコマンドを実行してください。

```
ins_setchkint -t FSList.0 -c quick -v 120
```

ファイル共有リソース `FSList.0` のディープチェックを無効にする場合は、`$LKROOT\bin` から以下のコマンドを実行してください。

```
ins_setchkinst -t FSList.0 -c deep -v0
```

ins_setin

```
ins_setin [-d destsys] -t tag [-v info]
```

`-v` オプションに指定する *info* 文字列には、リソースタイプに固有の追加情報を記述します。リソースタイプごとに一意である必要はありません。例えば、ファイル共有タイプのリソースインスタンスには、このインスタンスが管理するすべての共有名を *info* に指定します。

ins_setit

```
ins_setit [-d destsys] -t tag -I {AUTORES_ISP | INIT_ISP | INIT_OSU}
```

LifeKeeper 自体を初期化する際 (システム起動時など) にリソースの状態を初期化する方法を LifeKeeper に指示します。このオプションを設定しない場合、LifeKeeper は初期化状態をデフォルトオプションに設定します。

指定できるリストアオプションを以下に示します。

AUTORES_ISP: リソースの初期化を *AUTORES_ISP* に設定した場合は、リソースを OSU 状態に設定してから *restore* 処理が実行されます。処理が正常に実行されると、リソースは ISP 状態になります。 *restore* 処理に失敗した場合、リソースは OSF 状態になります。

INIT_ISP: *INIT_ISP* に設定した場合、LifeKeeper は他の方法でリソースを初期化するものとみなし、リソースを ISP 状態にします。

INIT_OSU: *INIT_OSU* に設定した場合、LifeKeeper は、リソースが初期化時に起動せず、システム管理者が LifeKeeper グラフィカルユーザインターフェース (GUI) アプリケーションを使用して、手動でリソースを起動するものとみなします。

ins_setlocalrecover

```
ins_setlocalrecover [-d destsys] -t tag -l {Y=enable|N=disable}
```

`-t tag` タグ名で指定されたリソースのローカルリカバリ設定を変更します。

例

ファイル共有リソース *FSList.0* のローカルリカバリを無効にする場合は、`$LKROOT/bin` から以下のコマンドを実行してください。

```
ins_setlocalrecover -t FSList.0 -l N
```

ins_setst

```
ins_setst [-d destsys] -t tag -S {ISP|ISU|OSU} [-R reason] [-A]
```

destsys システムにある *tag* タグ名が付いたリソースを `-S` オプションで指定された状態に設定します。 *destsys* を指定しないと、カレントシステムが対象となります。リソースの状態は、*remove* や *restore* などの処理スクリプトで変更されるため、このコマンドは慎重に使用してください。呼び出し側は、現在の状態がアプリケーションの実際の状態を反映していることを確認する必要があります。

`-R` オプションを指定すると、リソースの状態が変化した理由を説明するテキストが表示されます。 `-A` オプションは、指定したリソースとその依存リソースの状態を階層を遡って再帰的に変更します。

LCDI 関係

構文

[dep_create](#) [-d destsys] -p partag -c chdtag

[dep_remove](#) [-d destsys] [-p partag] [-c chdtag]

[dep_list](#) [-d destsys] [-fC] [-C allchild | -P allparent | -c ofparenttag | -p ofchildtag] [-r typ] [-a app]

[eqv_create](#) [-d destsys] [-s sys] -t tag [-p sysPriority] [-Sothersys] -o othertag [-r othersysPriority] -e SHARED

[eqv_remove](#) [-d destsys] [-s sys] -t tag [-S othersys] -o othertag -e SHARED

[eqv_list](#) [-d destsys] [-s sys] [-t tag] [-fC]

説明

LifeKeeper リソースは、互いに対する関係の中で存在しています。2つのリソースは、無関係である場合も依存関係がある場合もあります。階層内のリソースは、複数のリソースから依存されたり、複数のリソースに依存したりしています。各リソースは、共有イクイバレンシのペアシステム上で同様なリソースとも関連しています。この共有イクイバレンシによって、リソースは1度に1つのシステムでのみアクティブになることが保証されます。イクイバレンシオブジェクトは、リソースに対するシステムの優先順位も示しています。この優先順位の値によって、カスケードリングフェイルオーバーの順序が決まります。リソースのリカバリ時には、優先順位の高いシステムが優先順位の低いシステムよりも優先されます。優先順位の値は1が最高です。数値が大きいほど優先順位は低くなります。1つのリソースについて、2つのシステムに同じ優先順位を割り当てることはできません。有効な範囲は1～1024です。

終了コード

どのコマンドも、正常に実行された場合は0を返します。失敗した場合は0以外のコードを返し、標準エラーに出力します。前記のコマンドは、次の終了コードを返します。

0	処理に成功した。
1	システムコールまたはライブラリコールが内部エラーを返した。
2	ユーザが指定した構文にエラーがある。
3	LifeKeeper の内部エラー。
4	既存のオブジェクトに対する処理を要求した。
5	不正な引数を渡した。
6	インデックスが有効な範囲を超えている。
7	存在しないオブジェクトに要求を発行した。

8	リソースインスタンスの削除を要求したが、そのリソースインスタンスには、削除していないリソースインスタンスが依存している。
9	相手のシステムと通信できない。

dep_create

```
dep_create [-d destsys] -p parent -c child
```

parent タグと *child* タグのリソースインスタンスの間に依存関係を作成します。両方のリソースが同一の *destsys* システム上にある必要があります。*destsys* を指定しないと、カレントシステムが対象となります。これは、*parent* リソースが正常に機能するには *child* リソースが必要であることを意味します。正常に機能するには、両方のリソースインスタンスが同じ状態 (ISP または OSU) で存在していなければなりません。

dep_list

```
dep_list [-d destsys] [-fC] [-C allchild | -P allparent | -c ofparenttag | -p ofchildtag] [-r typ] [-a appname]
```

リソースインスタンス間の依存関係を記述した文字列を標準出力に表示します。*destsys* を指定しないと、カレントシステムが対象となります。各文字列の形式は以下のとおりです。

```
LK0-LKSYS:135.66.249.201
```

```
LK0-LKSYSA:FSLIST.0
```

```
FSLIST0:fi.vo.0
```

各文字列には2つのフィールドがあり、区切り文字で区切られています。デフォルトの区切り文字は ^A (\001) です。*-fC* オプションを指定した場合、区切り文字は指定した文字に変更されます。上記の例では、区切り文字としてコロン (:) が表示されています。先頭のフィールドには関係の親のタグ名、右側のフィールドには子のタグ名が表示されます。

オプションを使用すると、リストの表示内容を絞り込むことができます。オプションを使用しない場合は、すべての依存関係が表示されます。このコマンドには、次のオプションを指定できます。

-C	-C オプションを指定した場合、 <i>allchild</i> で指定したリソースの直接および間接的な子の依存関係がすべて表示されます。
-P	-P オプションを指定した場合、 <i>allparent</i> で指定したリソースの直接および間接的な親の依存関係がすべて表示されます。
-c	-c オプションを指定した場合、 <i>ofparenttag</i> で指定したリソースの直接的な子の依存関係のみが表示されます。
-p	-p オプションを指定した場合、 <i>ofchildtag</i> で指定したリソースの直接的な親の依存関係のみが表示されます。
-r	-r オプションを指定すると、子 <i>typ</i> の依存関係がすべて表示されます。
-a	-a オプションを指定すると、アプリケーション <i>appname</i> の依存関係がすべて表示されます。

dep_remove

```
dep_remove [-d destsys] [-p parent] [-c child]
```

destsys システムのデータベースから依存関係を削除します。*destsys* を指定しないと、カレントシステムが対象となります。*child* を指定しない場合、*parent* の依存関係がすべて削除されます。*parent* を指定しない場合、*child* タグの依存関係がすべて削除されます。両方とも指定しない場合は、すべての依存関係が削除されます。

eqv_create

```
eqv_create [-d destsys] [-s sys] -t tag [-p sysPriority] [-S othersys] -o  
othertag [-r othersysPriority] -e SHARED
```

destsys システムの構成データベースにイクイバレンシを作成します。*destsys* を指定しないと、ローカルシステムが対象となります。

sys または *othersys* を指定した場合は、LifeKeeper によって自動的にリモートシステムに *SHARED* イクイバレンシが追加されます。*sys* システムのタグとして指定されたリソースは、通常の状態で作動する「プライマリ」リソースであるとみなされます。*othersys* システムの *othertag* として指定されたリソースは、ペアシステム上の「セカンダリ」リソースとなります。LifeKeeper の初期化時に、リソース初期化設定に応じてプライマリリソースが設定されます ([LCDI インスタンス](#)を参照)。待機システムが起動すると、待機システム上の LifeKeeper によって、プライマリシステムが機能しているかどうかと、プライマリリソースが ISP 状態にあるかどうかをチェックされます。両方とも該当する場合、セカンダリリソースは OSU 状態になります (リソース初期化は無視されます)。いずれかが該当しない場合は、「リソース初期化」の設定に従ってセカンダリリソースが初期化されます。*-p* オプションで指定された値は、リソース *tag* についての *sys* システムの優先順位です。*-r* オプションで指定された値は、リソース *othertag* についての *othersys* システムの優先順位です。

eqv_list

```
eqv_list [-d destsys] [-s sys] [-t tag] [-e SHARED] [-fC]
```

リソースインスタンス間のイクイバレンシ関係を記述した文字列を標準出力に表示します。*destsys* を指定しないと、カレントシステムが対象となります。各行のフィールドは区切り文字で区切られています。デフォルトの区切り文字は ^A (\001) です。*-fC* オプションを指定した場合、区切り文字は C に変更されます。

下記の例では、区切り文字としてコロン (:) が表示されています。

```
LKSYSYA:135.66.249.201:LKSYSB:135.66.249.201:SHARED
```

```
LKSYSYA:FSLIST.0:LKSYSB;FSLIST.0:SHARED
```

```
LKSYSYA:LK0-LKSYSYA:LKSYSB:LK0-LKSYSYA:SHARED
```

LKSYSYA:fi.vo.0:LKSYSB;fi.vo.0:SHARED を例にして、各フィールドについて説明します。

LKSYSYA	リソースがあるプライマリシステムの名前。
fi.vo.0	プライマリシステムのボリュームリソースタグ。
LKSYSB	リソースイクイバレンシがあるセカンダリシステムのシステム名。
fi.vo.0	セカンダリシステム上のイクイバレンシリソースのボリュームリソースタグ。
SHARED	イクイバレンシタイプ。

残りの引数を指定すると、出力される情報が以下のように絞り込まれます。

-s *sys*: 引数 *sys* で指定されたシステムに関連付けられたイクイバレンシのみが出力されます。

-t *tag*: 引数 *tag* で指定されたタグに関連付けられたイクイバレンシのみが出力されます。

-e *SHARED*: すべての SHARED イクイバレンシ情報が表示されます。

eqv_remove

```
eqv_remove [-d destsys] [-s sys] -t tag [-S othersys] -o othertag [-e SHARED]
```

-e オプションで指定したイクイバレンシタイプを *destsys* システムの構成データベースから削除します。destsys を指定しないと、ローカルシステムが対象となります。tag リソースと othertag リソースの間にあるイクイバレンシタイプを sys システムと othersys システムから再帰的に削除します。sys または othersys を指定しないと、カレントシステムが対象となります。

LCDI リソースタイプ

構文

[typ_create](#) [-d destsys] -a appname -r restyp

[typ_remove](#) [-d destsys] -a appname -r restyp

[typ_list](#) [-d destsys] [-fC] [-a appname]

説明

リソースは、システムが認識しているボリューム、アプリケーション、またはその他のオブジェクトを表すために LifeKeeper によって使用されます。リソースタイプはリソースの分類であり、すべてのインスタンスに適用できるリカバリ手順の共通セットによって区別されます。リソースタイプの例を以下に示します。

- ファイルシステムボリューム (例: K:)
- ファイル共有 (例: UTIL_SHARE)
- IP アドレス (例: 153.66.232.21)

`typ_create` コマンドおよび `typ_remove` コマンドは、構成データベースに新しいタイプを生成するためのインターフェースを提供します。`typ_list` コマンドは、特定のシステムに存在するすべてのリソースタイプを一覧表示するための構成データベースへのインターフェースを提供します。

終了コード

どのコマンドも、正常に実行された場合は0を返します。失敗した場合は0以外のコード（「終了コード」を参照）を返し、標準エラーに出力します。前記のコマンドは、次の終了コードを返します。

0	処理に成功した。
1	システムコールまたはライブラリコールが内部エラーを返した。
2	ユーザが指定した構文にエラーがある。
3	LifeKeeper の内部エラー。
4	既存のオブジェクトに対する処理を要求した。
5	不正な引数を渡した。
6	インデックスが有効な範囲を超えている。
7	存在しないオブジェクトに要求を発行した。
8	リソースインスタンスの削除を要求したが、そのリソースインスタンスには、削除していないリソースインスタンスが依存している。
9	相手のシステムと通信できない。

typ_create

```
typ_create [-d destsys] -a appname -r restyp
```

`destsys` システムの構成データベースに新しいリソースタイプを作成します。`destsys` を指定しないと、ローカルシステムが対象となります。リソースタイプは、`restyp` という名前で、既存のアプリケーション `appname` の下にインストールされます。システムまたはアプリケーションが不明の場合、あるいはリソースタイプがすでに存在している場合には失敗します。

typ_list

```
typ_list [-d destsys] [-fC] [-a appname]
```

`destsys` システムにインストールされたアプリケーション `appname` で定義されているリソースタイプのリストを、標準出力に表示します。`destsys` を指定しないと、ローカルシステムが対象となります。`appname` を指定しない場合は、すべてのアプリケーションのすべてのリソースタイプが以下の形式で表示されます。

```
filesys:volume
comm:ip
database:informix
```

typ_remove

区切り文字の左側はアプリケーション名、右側はリソースタイプ名です。各行のフィールドは区切り文字で区切られています。デフォルトの区切り文字は ^A (\001) です。-fC オプションを指定した場合、区切り文字は指定した文字に変更されます。上記の例では、区切り文字としてコロン (:) が表示されています。

typ_remove

```
typ_remove [-d destsys] -a appname -r restyp
```

destsys システムの既知のリソースタイプの構成データベースセットから、指定したリソースタイプを削除します。destsys を指定しないと、ローカルシステムが対象となります。このタイプに関連付けられたリソースインスタンス、依存関係、およびイクイバレンシも、すべて削除されます。指定したリソースタイプが構成データベースで不明な場合は失敗します。

LCDI システム

構文

[sys_create](#) [-d destsys] -s sys

[sys_remove](#) [-d destsys] -s sys

[sys_getds](#) [-d destsys] -s sys

[sys_getst](#) [-d destsys] -s sys

[sys_list](#) [-d destsys]

説明

LifeKeeper 構成データベースは関連システムを認識します。リソースとリソースタイプはそれらが配置されているシステム固有のものなので、構成データベースインターフェースにシステムの概念を含める必要があります。

LCDI システムコマンドは、データベースに対して、情報を返したり、作成したり、削除したりします。

終了コード

どのコマンドも、正常に実行された場合は 0 を返します。失敗した場合は 0 以外のコードを返し、標準エラーに出力します。前記のコマンドは、次の終了コードを返します。

0	処理に成功した。
1	システムコールまたはライブラリコールが内部エラーを返した。
2	ユーザが指定した構文にエラーがある。
3	LifeKeeper の内部エラー。
4	既存のオブジェクトに対する処理を要求した。

5	不正な引数を渡した。
6	インデックスが有効な範囲を超えている。
7	存在しないオブジェクトに要求を発行した。
8	リソースインスタンスの削除を要求したが、そのリソースインスタンスには、削除していないリソースインスタンスが依存している。
9	相手のシステムと通信できない。

sys_create

```
sys_create [-d destsys] -s sys
```

destsys システムの構成データベースに新しいシステム定義を作成します。*destsys* を指定しないと、ローカルシステムが対象となります。*-s sys* オプションは、システム名を割り当てるシステムを特定するために必要です。

sys_getds

```
sys_getds [-d destsys] -s sys
```

destsys システムのデータベースから、システムが現在の状態になった理由を説明したオプションテキストを取得し、標準出力に表示します。*destsys* を指定しないと、ローカルシステムが対象となります。

sys_getst

```
sys_getst -s sys
```

システムの状態を、次のいずれかの文字列で、標準出力に表示します。

DEAD (停止)	システムは使用できないと思われます。
ALIVE (稼働中)	システムは使用できると思われます。
UNKNOWN (不明)	システムの状態は不明です。

sys_list

```
sys_list [-d destsys]
```

destsys システムのデータベースから、LifeKeeper が認識しているシステムのリストを取得し、標準出力に表示します。*destsys* を指定しないと、ローカルシステムが対象となります。

sys_remove

```
sys_remove [-d destsys] -s sys
```

destsys システムの構成データベースからシステム定義を削除します。destsys を指定しないと、ローカルシステムが対象となります。-s sys オプションは、システム名を割り当てたシステムを特定するために必要です。

LifeKeeper のフラグ

LifeKeeper GUI では、システムに設定されたフラグの一覧がステータス表示の最後に表示されます。ロック LCD フラグは、システム共通のフラグであり、ロックをかけたプロセスが完了するまで、他のプロセスに待機を要求します。LCD ロックの標準的な形式を次に示します。

```
!action!processID!time!machine:id
```

LCD ロックフラグの具体例を次に示します。

- **!action!02833!701236710!<servername>:Restore_hierarchy** - ファイルシステム階層を作成すると、この形式のロックフラグがステータスウィンドウに表示されます。filesys には、ディスクパーティション階層を持つ appdp アプリケーション、またはファイルシステム階層を持つ appfs アプリケーションを指定できます。
- その他の一般的なフラグには、!nofailover!machine と shutdown_switchover があります。!nofailover!machine フラグは、サーバのフェイルオーバーを制御するために、LifeKeeper が作成または削除する一時的な内部フラグです。shutdown_switchover フラグは、このサーバのシャットダウン方法が switchover に設定されており、サーバをシャットダウンすると、切り替えが発生することを示します。フラグの詳細については、[LCDI フラグ](#)を参照してください。

flg_create

```
flg_create [-d destsys] -f flag
```

destsys システムにフラグ flag を作成します。

注記: この処理では、LifeKeeper 構成データベースの「共有メモリ」セグメントのみが変更されます。

共有メモリの変更内容を恒久的なストレージのディスクファイルに確実に反映するには、このコマンドの後に LifeKeeper **lcdsync** コマンドを実行してください。

flg_list

```
flg_list [-d destsys]
```

flg_list は、このシステム (destsys が指定されている場合を除く) で現在定義されているすべてのフラグについて、1 行に 1 つの短いリストを標準出力に表示します。リストの順序は不定です。

flg_remove

```
flg_remove [-d destsys] -f flag
```

destsys システムのフラグ flag を削除します。

注記: この処理では、LifeKeeper 構成データベースの共有メモリセグメントのみが変更されます。

共有メモリの変更内容を恒久的なストレージのディスクファイルに確実に反映するには、このコマンドの後に LifeKeeper **lcdsync** コマンドを実行してください。

flg_test

```
flg_test [-d destsys] -f flag
```

destsys システムにフラグ *flag* が存在するかどうかをチェックします。0 または 7 を返します。

LCDI フラグ

構文

```
flg\_create[-d destsys] -f flag  
flg\_remove[-d destsys] -f flag  
flg\_test[-d destsys] -f flag  
flg\_list[-d destsys]
```

説明

LifeKeeper には、さまざまなタスクを実行するためにフラグを動的に設定する機能があります。以下の特殊用途のフラグが存在する場合があります。

Inofailover!uname

このフラグが存在する場合、名前が *uname* のシステム上のリソースはフェイルオーバーが抑制されます。フラグを持つシステムはバックアップシステムとして定義されます。注記：これは一時的なフラグであり、*uname* システムが起動中であることを LifeKeeper が検出すると、フラグは自動的に削除されます。

laction!procid!timestamp!uname:identifier

これは「admin lock flag」の例です (getlocks を参照)。「admin lock flag」は、LifeKeeper 構成内のいずれかのシステムで他の処理が同時に実行されないようにする必要がある処理に対して使用されます。例えば、あるシステムで階層を作成しながら、別のシステムでも階層を作成することはできません。「admin lock flag」を使用すると、これらの「global」処理の 1 つについて、現在実行中の処理が完了するまで実行されないことが保証されます。

実行中の処理の種類は、「admin lock flag」の識別子フィールドによって識別されます。「admin lock flag」を要求したプロセスが実行されていたシステムは *uname* で指定されます。フラグは、1970 年 1 月 1 日以降の秒数のタイムスタンプで、getlocks を呼び出したプロセス ID が *procid* のプロセスによって作成されたものです (getlocks を参照)。

このようなフラグの例を次に示します。

laction!01525!701120147!cindy:Create_Hierarchy

このフラグは、処理 *Create_Hierarchy* を実行中であり、階層を作成していることを示します。プロセス ID 1525 のプロセスが、*cindy* システムで時刻 701120147 に「admin lock flag」を要求しました。

lrestore

このフラグは、restore 前処理スクリプト ([LCD](#) を参照) の実行時に LifeKeeper によって設定され、restore 後処理スクリプトを実行する必要があることを示します。通常、これは一時的な状態であり、restore 後処理スクリプト ([LCD](#) を参照) の実行時に LifeKeeper によって自動的に修正されます。唯一の例外は、以下のコマンドを使用して、明示的に restore 後処理スクリプトを実行する場合があります。

```
%LKROOT%\bin\lcdrecover -G restore
```

lrestore!uname

このフラグを設定すると、uname システムで restore 後処理スクリプト ([LCD](#) を参照) をリモートで実行する必要があることを示します。このシステムで restore 後処理スクリプトが実行されると、LifeKeeper は restore 後処理スクリプトを実行するように uname システムにリモート要求を送信します。通常、これは一時的な状態であり、LifeKeeper によって自動的に修正されます。唯一の例外は、`%LKROOT%\bin\lcdrecover -G restore` コマンドを使用して、明示的に restore 後処理スクリプトを実行する場合があります。

lremove

このフラグは、remove 前処理スクリプト ([LCD](#) を参照) の実行時に LifeKeeper によって設定され、後から remove 後処理スクリプトを実行する必要があることを示します。通常、これは一時的な状態であり、後から remove 後処理スクリプト ([LCD](#) を参照) を実行すると LifeKeeper によって自動的に修正されます。唯一の例外は、以下のコマンドを使用して、明示的に remove 後処理スクリプトを実行する場合があります。

```
%LKROOT%\bin\lcdrecover -G remove
```

lremove!uname

このフラグを設定すると、uname システムで remove 後処理スクリプト ([LCD](#) を参照) をリモートで実行する必要があることを示します。このシステムで remove 後処理スクリプトが実行されると、remove 後処理スクリプトを実行するように uname システムにリモート要求が送信されます。通常、これは一時的な状態であり、LifeKeeper によって自動的に修正されます。唯一の例外は、`%LKROOT%\bin\lcdrecover -G remove` コマンドを使用して、明示的に remove 後処理スクリプトを実行する場合があります。

ldelete

このフラグは、delete 前処理スクリプト ([LCD](#) を参照) の実行時に LifeKeeper によって設定され、後から delete 後処理スクリプトを実行する必要があることを示します。通常、これは一時的な状態であり、delete 後処理スクリプト ([LCD](#) (1M) を参照) の実行時に LifeKeeper によって自動的に修正されます。唯一の例外は、以下のコマンドを使用して、明示的に delete 後処理スクリプトを実行する場合があります。

```
%LKROOT%\bin\lcdrecover -G delete
```

ldelete!uname

このフラグを設定すると、uname システムで delete 後処理スクリプト ([LCD](#) を参照) をリモートで実行する必要があることを示します。このシステムで delete 後処理スクリプトが実行されると、delete 後処理スクリプトを実行するように uname システムにリモート要求が送信されます。通常、これは一時的な状態であり、LifeKeeper によって自動的に修正されます。唯一の例外は、以下のコマンドを使用して、明示的に delete 後処理スクリプトを実行する場合があります。

```
%LKROOT%\bin\lcdrecover -G delete
```

lk_chg_value

名前

lk_chg_value.ksh — ローカルにある LifeKeeper 構成 データベースファイル内の 指定された値を変更します。

構文

```
lk_chg_value.ksh {-o old_value -n new_value | -f filename} [-vFIMT]
```

説明

ローカルの LifeKeeper 構成 データベースファイルに登録されている任意の値 (LifeKeeper のシステム名、コミュニケーションパスのアドレス、リソースタグ名など) を変更します。lk_chg_value.ksh は、LifeKeeper が稼働していないクラスタノードに管理者権限でログインして実行します。また、LifeKeeper 付属のシェル (sh.exe) を使用し、前記の構文に従って、スクリプトを呼び出します。このコマンドは、システム名やネットワークインターフェースを変更しません。LifeKeeper のシステム名やコミュニケーションパスの IP アドレスを変更する場合は、システムユーティリティを使用して、システム名とネットワークインターフェースを変更してから、このコマンドを実行します。LifeKeeper を適切に更新するには、このコマンドをすべてのクラスタサーバに実行します。

値の変更は、-o と -n オプションを指定してコマンドラインから実行することも、-f オプションを指定してファイルを指定することもできます。ファイルを指定する場合は、置換する値を old_value=new_value 形式で 1 行ずつ記述します。

lk_chg_value.ksh が変更する値を変更前に確認するには、-M オプションを指定します。lk_chg_value.ksh のバージョン情報を確認するには、-v オプションを指定します。タグ名を変更しない場合は、-T オプションを指定します。リソース ID を変更しない場合は、-I オプションを指定します。

リソース ID は構造化された情報を含むため、lk_chg_value.ksh は ID フィールドの完全な置換を許可しません。この動作をオーバーライドするには、-F オプションを指定します。

例

システム A、B、C は、LifeKeeper クラスタを構成しています。システム A と B はデータベースリソースを管理し、システム A はシステム C とのコミュニケーションパスを管理しています。システム A のシステム名とコミュニケーションパスのアドレスを変更するには、次の手順に従います。

1. 各システムに lkstop コマンドを実行して、LifeKeeper を停止します。ただし、管理対象のリソースが稼働している場合は、lkstop に -f オプションを指定します。
2. システム A のシステム名を「X」、ネットワークアドレスを「new_address」にそれぞれ変更します。置換用のファイル (/tmp/lksubs) を作成し、置換前と置換後の文字列を次の形式で記述します。

```
A=X old_address=new_address
```

終了コード

管理者でシステム A にログインし、次のコマンドを実行します。

```
set LKROOT=<LKROOT> (例: set LKROOT=C:\LK)
```

```
<LKROOT>\bin\sh.exe lk_chg_value.ksh -vf /tmp/lksubs
```

システム A にある LifeKeeper Core とリカバリキットに登録されている「A」と「old_address」すべてがシステム「X」と「new_address」に置換されます。

3. 置換ファイルをシステム A からシステム B と C にコピーします。管理者でシステム B と C にログインし、次のコマンドを実行します。

```
set LKROOT=<LKROOT> (例: set LKROOT=C:\LK)
```

```
<LKROOT>\bin\sh.exe lk_chg_value.ksh -vf /tmp/lksubs
```

システム B と C の構成データベースに登録されている「A」と「old_address」が「X」と「new_address」にそれぞれ置換されます。

終了コード

- 0 処理に成功しました。
- 1 割り込みが発生しました。ファイルをリストアしています。
- 2 不正な引数が渡りました。
- 3 LifeKeeper プロセスは実行中です。
- 4 コマンドは管理者権限で実行してください。
- 5 ID フィールドの変更が要求されました。-I オプションを使用しないと、ID は変更できません。
- 6 LKROOT 環境変数が設定されていません。
- 7 一致するものが見つかりません。

注記

lk_chg_value.ksh ユーティリティは、<LKROOT>\bin フォルダにあります。

lk_chg_value.ksh ユーティリティでは、大文字と小文字が区別されます。

前述したように、**lk_chg_value.ksh** スクリプトを起動するには、LifeKeeper 付属のシェル (sh.exe) を使用します。

<LKROOT> は LifeKeeper のホームディレクトリです。デフォルトのホームディレクトリは C:\LK ですが、このディレクトリは、LifeKeeper をインストールするときに変更できます。

ファイル

```
<LKROOT>\bin\lk_chg_value.ksh
```

lk_err

構文

```
lk_err -c Category -n Error number -p Process Name [-d {TO_LOG | TO_STDERR}] "Message"
```

説明

リカバリスクリプト内で実行して、Windows のイベントログにエラーを出力します。stderr (標準出力) にもメッセージを表示します。

次の引数を指定できます。

Category: LifeKeeper メッセージカテゴリとイベントログの分類は、次のとおりです。

LK カテゴリ	イベントカテゴリ	イベントタイプ
FRS_MES	汎用	情報
FRS_WARN	汎用	警告
FRS_ERR	汎用	エラー

Error Number: 正数を指定します。

Process Name: lk_err を呼び出すスクリプトの名前を指定します。

Destination: 省略可能です。lk_err が生成するイベントは、デフォルトで Windows Event Log (TO_LOG) とシステムコンソール stderr のメッセージストリーム (TO_STDERR) の両方に出力されます。-d オプションを指定すると、いずれか一方にエラー情報を出力できます。

ただし、LifeKeeper Core は、プログラムやスクリプトを対話形式ではなく、バックグラウンドで実行するため、LifeKeeper Core が実行するプログラムやスクリプトでメッセージを stderr (TO_STDERR) にリダイレクトすると、メッセージはシステムコンソールに出力されません。手作業でスクリプトをテストしてデバッグする場合だけ、メッセージを stderr に出力してください。

Message: テキストは " " で囲みます。

perform_action

構文

```
perform_action [-G] [-s] [-b] [-n] -t tag-name -a action-name [- - arg1 arg2 ... argn]
```

説明

LRACI プログラム **perform_action** は、次の順序でプロセスを実行します。

- *-t* オプションの引数 *tag-name* で指定されるリソースを探します。
- *-a* オプションの引数 *action-name* で指定される処理スクリプトを探します。
- *tag-name* リソースインスタンス上で処理スクリプトを実行します。

– 引数の後の引数はそのまま処理スクリプトに渡されます。これらの引数は、処理の開発者が必要に応じて要求します。

perform_action プログラムは次のアルゴリズムで処理スクリプトを探します。まず、以下のように、*tag-name* で指定されるリソースインスタンスのリソースタイプについて、actions ディレクトリを検索します。

```
%LKROOT%\subsys\appname\resources\restypname\actions\action-
name.ksh
```

見つからない場合は、以下のように、リソースインスタンスが属するアプリケーションの actions ディレクトリをチェックします。

```
%LKROOT%\subsys\appname\actions\action-name.ksh
```

restore 処理と remove 処理は LRACI の特例です。restore 処理は、リモートシステムで稼働している可能性があるアプリケーション階層をローカルシステムに移動します。restore 処理では、LRACI は最初にリソースインスタンス *tag-name* が ISP 状態ではないことを確認します。ISP 状態でなければ、このリソースが依存しているすべてのリソースインスタンスが再帰的にチェックされます。このチェックは、リソースが別のリソースに依存していないか、依存しているリソースがすべて ISP 状態になっていることが判明するまで継続されます。リソースが ISU 状態だった場合は、ISP 状態になります。

リソースが OSU 状態または OSF 状態だった場合は、リモートシステム上の影響を受けるリソースに対して remove スクリプトが実行されます。これが完了すると、LRACI は上記のアルゴリズムを使用して restore 処理を検索して実行します。スクリプトが失敗した場合、リソースは OSF 状態になり、LRACI は停止します。正常に実行された場合、LRACI は *tag-name* で指定されたリソースがリストアされるまで、ツリーを上に向かってリソースを再帰的に「リストア」します。次に、LRACI は関連するルートリソースインスタンスがすべてリストアされるまで、同様の方法で再帰的に親リソースインスタンスのチェックと「リストア」を行います。どの場合でも、LRACI は上記のアルゴリズムにより、LRACI が現在スキャンしているリソースのリソースアプリケーションおよびリソースタイプを使用して適切な restore スクリプトを検索します。*tag-name* リソースのリソースアプリケーションとリソースタイプではありません。

remove 処理では、リソースが逆方向に再帰的に移動されます。LRACI は、*tag-name* リソースに直接的または間接的に依存しているルートリソースから下に向かって、ISP 状態または ISU 状態のすべてのリソースの remove スクリプトを呼び出します。*tag-name* リソースも含まれます。ISP 状態でも ISU 状態でもないリソースは無視されます。いずれかの remove スクリプトが失敗した場合、LRACI は失敗したリソースを OSF 状態にして停止します。どの場合でも、LRACI は上記のアルゴリズムにより、LRACI が現在スキャンしているリソースのリソースアプリケーションおよびリソースタイプを使用して適切な remove スクリプトを検索します。*tag-name* リソースのリソースアプリケーションとリソースタイプではありません。

remove 処理および restore 処理では、引数 *-t tag-name* および *-i ident-field* が、対象のリソースインスタンスに対応する引数リストに自動的に追加されます。

以下に、**perform_action** に指定できる引数について説明します。

-G	<p>action-name が remove、restore、delete の場合のみ使用します。このオプションを指定しない場合、処理を実行する前後にグローバル前処理スクリプトとグローバル後処理スクリプトが実行されます (LCD の lcdrecover を参照)。このオプションを指定した場合、グローバル前処理スクリプトとグローバル後処理スクリプトは実行されません。</p> <p>このオプションは、perform_action を 2 回以上実行する必要があるが、グローバル前処理スクリプトとグローバル後処理スクリプトは 1 回だけ実行したいときに役立ちます。また、リソース階層を作成しながら perform_action を実行する必要がある場合にも役立ちます。階層作成スクリプトは「admin lock flag」を取得するように設定されますが (LCDI フラグ を参照)、restore 後処理スクリプトも「admin lock flag」を要求し、これによって競合の問題が発生します。そのため、階層作成時には perform_action によってグローバル前処理スクリプトとグローバル後処理スクリプトが実行されないようにする必要があります。</p>
-s	<p>restore 処理のデフォルトの動作は、指定したタグの上および下のすべてのオブジェクトを起動します。また、remove 処理のデフォルトの動作は、指定したタグの上のすべてのオブジェクトを停止します。-s オプションは、処理の範囲を指定したフラグだけに制限します。</p>
-b	<p>restore 処理のデフォルトの動作は、指定したタグの上および下のすべてのオブジェクトを起動します。-b オプションは、この動作を、指定したタグの下のオブジェクトだけに変更します。このオプションは remove 処理には影響しません。</p>
-n	<p>action-name が remove または restore の場合のみ使用します。このオプションを指定した場合、リソースの予約はチェックされず、リソースが予約されているかどうかにかかわらず処理が実行されます。</p> <p>警告: このオプションの使用時は、特に注意してください。</p> <p>このオプションを指定しない場合、remove スクリプトまたは restore スクリプトを実行する前に、LRACI はいずれかの処理を実行するリソースが現在別のプロセスによって予約されているかどうかを確認します。リソースの「サービスからの削除」、「サービスのリストア」、または「リカバリ」の処理を実行している間リソースを予約できません。</p> <p>いずれかのリソースが予約されている場合、プロセスがリソースの予約を解除するまで、LRACI は指定された期間だけ待機します。この期間が経過すると、LRACI は予約を解除します。いずれの場合でも、LRACI はすべてのリソースを予約してから、指定されたアルゴリズムに従って処理を実行します。</p>
-t tag-name	<p>最後に処理が実行されるリソースインスタンス。</p>
-a action-name	<p>実行されるリソース処理。</p>

例

<code>--arg1 arg2... argn</code>	リソース処理開発者が必要に応じて定義して、処理スクリプトに渡すことができる引数。getlocks によって保護された領域内で -G オプションを使用せずに perform_action を実行する場合は、 <i>arg1</i> に -m を設定して、getlocks の 2 つ目のインスタンスが実行されないようにしてください。2 つ目のインスタンスが実行されると、処理がハングします。
----------------------------------	---

例

例として、以下の処理を呼び出します。

```
perform_action -t SCSI-USR-DISK -a reset-heads -- -h 7
```

LRACI プログラム **perform_action** は、reset-heads に対応する処理を検索して、以下の引数で実行します。

```
reset-heads -t SCSI-USR-DISK -h 7
```

終了コード

LRACI は、次の終了コードを返します。

0	処理に成功した。
1	システムコールまたはライブラリコールが内部エラーを返した。
2	ユーザが指定した構文にエラーがある。
3	LifeKeeper の内部エラー。
4	既存のオブジェクトに対する処理を要求した。
5	不正な引数を渡した。
6	インデックスが有効な範囲を超えている。
7	存在しないオブジェクトに要求を発行した。
8	リソースインスタンスの削除を要求したが、そのリソースインスタンスには、削除していないリソースインスタンスが依存している。
9	相手のシステムと通信できない。

sendevent

構文

```
%LKROOT%\bin\sendevent -C class-of-event -E event -m monitor-name -  
nname-of-obj-inst [-s severity]
```

説明

イベント通知機能は、イベント通知機構 (`%LKROOT%\bin\sendevent`) とアプリケーション登録環境で構成されています。イベント通知機能を使用するアプリケーションは、イベントまたはアラームを選択するか、すべてのイベントとアラーム通知を受信することを登録します。

sendevent コマンドは、モニタが監視対象のオブジェクトにイベント (障害またはリカバリ) を検出した際、デーモン監視プロセスから起動されるプログラムです。このコマンドは、一般ユーザやシステム管理者がシェル上で使用するのではなく、デーモンプロセスや別のコマンドによって起動されます。

sendevent コマンドは、イベント発生を登録済みアプリケーションに通知します。例えば、アプリケーションは、システムのシャットダウンを事前に通知されると、ファイルとデータを保存します。また、クライアントサーバ環境では、アプリケーションは別のサービスプロバイダを使用するように自身を再設定します。アプリケーションには、イベントを処理する適切なコマンドを実装する必要があります。

sendevent コマンドは、アプリケーションのイベント応答コマンドにすべてのオプションを渡します。

アプリケーションは、イベント応答コマンドを専用の登録ディレクトリ `%LKROOT%\events` にインストールして、イベントやアラーム通知を受信します。この作業は、アプリケーションのインストール時に実行します。`%LKROOT%\events` に格納されるイベントは、イベントクラスに分類されます。アプリケーション側では、`genclass` (汎用イベント用) と `allclass` サブディレクトリを作成して、イベント発生通知の受信を登録します。

注記: アプリケーション応答コマンドを特定のイベントロケーションとすべてのロケーションの両方に配置すると、イベントが発生した際、両方のスクリプトが実行されます。

クラスディレクトリには、そのクラスに属するイベントに対応するサブディレクトリがあります。イベントを監視し、イベント通知機構を使用するアドオンパッケージは、監視サポート対象のイベントを記録します。

アプリケーションオブジェクトモニタパッケージは、各サブディレクトリに配置した ACTIVE ファイルを保持する必要があります。ACTIVE ファイルが存在する場合、モニタが起動し、特定のイベントのオブジェクトを監視していることをアプリケーションに通知します。モニタプログラムを含むパッケージを削除すると、パッケージの削除スクリプトが監視対象イベントの ACTIVE ファイルを削除します。イベントを監視しないことをアプリケーションに通知します。パッケージを削除しても、イベント応答コマンドとイベントディレクトリは削除されません。

アプリケーションが別のアプリケーションの標準的なコマンドに依存する場合、アプリケーション登録環境は、アプリケーション用の別のアプリケーション固有のディレクトリ `%LKROOT%\subsys\application-name\actions` を提供し、共有可能なアクションコマンドを配置します。例えば、アプリケーション X がアプリケーション Y に依存しており、イベントリカバリ後に起動して実行するとします。この順番に従ってアプリケーションが起動しない場合、アプリケーション X は `LKROOT%\subsys\Y\actions\start` ディレクトリからアプリケーション Y の `start` コマンドを呼び出します。アプリケーション間の依存関係は、アプリケーション開発者が判断して指定します。

`-C` (イベントのクラス)、`-E` (イベント)、`-m` (モニタ名)、`-n` (オブジェクトインスタンス名) オプションは必須です。`-s` (重大度) オプションを指定しないと、`sendevent` にはデフォルトで MAJOR アラームが設定されます。

監視プロセスが **sendevent** コマンドを起動すると、**sendevent** は、`-C` オプションと `-E` オプションを参照して、イベントが発生したクラスを識別します。**sendevent** コマンドは、クラス/イベントに対応する登録ディレクトリにあるイベント応答コマンドとすべてのディレクトリに登録されたコマンドを完全に実行するまで、バックグラウンドで動作します。

指定可能なオプションは次のとおりです。

出力

-C class-of-event

イベントはクラス別にグループ化されます。イベントが属するクラスを指定します。これは必須オプションです。

-E event

クラス内でトリガするイベントを指定します。必須オプションです。

-m monitor-name

アラーム/イベントを送信するアプリケーションオブジェクトモニタの名前は、次の形式で記述します。

OM-product-name:OM-component-name

OM-product-name には、最大 8 文字の ASCII 文字列を記述します。これは、アラームまたはイベントをトリガするオブジェクトを監視する製品の略称です。OM-component-name には、最大 16 文字の ASCII 文字列を記述します。これは、アラームまたはイベントを検出するオブジェクトモニタのコンポーネントを識別するオブジェクトモニタ名です。

モニタ名は、1 つのオブジェクトを監視する複数の製品を識別します。

-n name-of-obj-inst

アプリケーションオブジェクトのインスタンス名です。最大 64 文字の ASCII 文字列を記述します。例えば、D: はボリュームアプリケーションオブジェクトの名前を指すのに対して、1234 はプロセスオブジェクトを指すのに使用されます。

-s severity

各アラームとイベントには、報告する問題の重大度を指定します。このオプションを指定しないと、**sendevent** はデフォルトの重大度 (MAJOR アラーム) を使用します。重大度は、次表に示すように、整数 (ASCII 文字) で指定します。

0	CLEARED アラーム: 「id-of-alarm/event」が検出したアラーム
1	INFORMATIONAL アラーム: INFO メッセージまたは cmn_err () NOTICE メッセージ
2	WARNING アラーム: 警告メッセージ
3	MINOR アラーム: 重大度の低いマイナーメッセージ
4	MAJOR アラーム: メジャーメッセージまたはエラーメッセージ (デフォルト)
5	CRITICAL アラーム: 深刻なエラーを伝えるメッセージ (cmn_err () PANIC または HALT メッセージ)

出力

sendevent コマンドは、次のいずれかの条件で出力を生成します。

- エラーメッセージが標準エラーに出力されるか、0 以外の終了コードが返った場合
- sendevent を呼び出した際に、id-of-alarm/event アラーム/イベントの識別子が標準出力に出力された場合

終了コード

sendevent は、次の終了コードを返します。

0	sendevent コマンドはエラーを出力せずに、処理を正常に完了した。
1	引数リストに構文エラーがある。
2	-C オプションで指定されたクラスが %LKROOT%\events ディレクトリにない。
3	-E オプションで指定した文字列に対応するイベントが %LKROOT%\events\ <class> td="" ディレクトリにない。<=""> </class>>
4	-A オプションは、内部的に生成されており、直接指定されていない。
5	-s 0 (重大度: CLEARED) オプションを使用する場合は、 -i オプションを指定する必要がある。

volume

構文

```
volume [ -d | -D ] [-l | -u | -p | -U volume_letter ]
```

説明

Windows サーバのボリュームをロックまたはロック解除します。また、LifeKeeper サービスにボリュームを登録します。LifeKeeper にボリュームを登録した場合、LifeKeeper は起動時に保護 (ロック) するボリュームを決定します。ロックは、ボリュームに排他制御を適用するため、他のプロセスはボリュームにアクセスできなくなります。

このコマンドを使用するには、LifeKeeper が稼働している必要があります。このコマンドは、LifeKeeper サービスと通信して、ロック機構を提供します。

volume コマンドに指定できるオプションは、次のとおりです。volume_letter には、ロック/ロック解除または保護/保護解除するドライブレター (C ~ Z) を指定します。

-d	ロックしているボリュームを表示します。
-D	LifeKeeper に登録されているボリュームを表示します。-p オプションで追加されたボリュームを表示します。-D を指定すると、-d オプションとは異なるリストが表示されます。
-l	ボリュームを排他的アクセスとしてロックします。リモートユーザがボリュームを開くか、ローカルアプリケーションが書き込み用にボリュームを開くと、ロックは失敗します。
-u	ボリュームの排他アクセスを解除します。
-p	LifeKeeper にボリュームを登録します。LifeKeeper を再起動すると、ボリュームは自動的にロックされます。
-U	LifeKeeper に登録したボリュームを解除します。LifeKeeper を起動しても、ボリュームは自動的にロックされません。

例

例

volume コマンドの使用例を次に示します。

```
#  
# Register drive volume e: to be locked by LifeKeeper  
#  
ret=`volume -p E`  
if [ $ret -gt 0 ]  
then  
    #Report error that it wasn't protected  
fi  
#  
# Lock volume e: for exclusive access  
#  
ret=`volume -l E`  
if [ $ret -gt 0 ]  
then  
    #Report error that it wasn't locked  
fi
```

終了コード

このコマンドは、次の終了コードを返します。

0	処理に成功した。
正数	操作に失敗した。エラーメッセージは標準出力に表示される。

IP ローカルリカバリ

IP ローカルリカバリ機能を有効にした場合、定期的に行う IP リソースのディープチェックが失敗すると、SPS は次の処理を実行します。

- 現在のネットワークインターフェースで IP アドレスを起動します。
- 前記の処理に失敗すると、リソースインスタンスをチェックし、バックアップインターフェースが使用できるか調べます。バックアップインターフェースが使用できる場合は、IP アドレスをバックアップイン

ターフェースに移動します。

- ローカルリカバリが完全に失敗すると、SPS は、IP アドレスと依存リソースをバックアップサーバにフェイルオーバーします。

ローカルリカバリ機能は、バックアップアダプタがなくても有効にすることができます。ローカルリカバリ機能を有効にすると、SPS はプライマリネットワークのインターフェースを再起動しますが、それに失敗すると、リソースをバックアップサーバにフェイルオーバーします。

SNMP による SPS イベント転送の概要

SNMP (Simple Network Management Protocol) は、デバイスから独立してネットワークを管理するフレームワークを提供します。ネットワーク上のデバイスは、製造メーカーが提供する MIB (Management Information Base) 変数で定義されます。ネットワーク上の各ノードで稼働する SNMP エージェントは、Network Manager ノードと対話します。Network Manager はエージェントにクエリを発行し、MIB 変数値を取得または設定して、ノードを監視制御します。エージェントはトラップと呼ばれるメッセージを非同期的に生成して、例外イベントをマネージャに通知します。SNMP を使用してネットワークを監視管理するアプリケーションは多数あります。

SPS には、特定のイベントやアラームの送信を希望するアプリケーションを登録するイベント通知機構があります。詳細については、[sendevent](#) マニュアルページを参照してください。SPS では、主要な SPS イベントの SNMP トラップをサードパーティ製ネットワーク管理コンソールに簡単に送信できます。SPS は MIB ファイルを SPS トラップ定義を記述した `%LKROOT%\include\LifeKeeper-MIB.txt` の下にインストールします。

SNMP トラップを受信するリモートの管理コンソールは、そのシステムの管理ソフトウェアであらかじめ設定しておきます。SPS は外部の SNMP 設定を提供していません。リモート管理サーバは通常、SPS クラスタ外にあり、SPS ノードではありません。

SPS イベントテーブル

SPS イベントとトラップ番号を次表に示します。オブジェクト ID (OID) は、プレフィックスとトラップ番号から成り、次の形式で表示されます。

prefix.0.specific trap number

プレフィックスは、**.1.3.6.1.4.1.7359** です。MIB ツリー内で展開すると、**iso.org.dod.internet.private.enterprises.7359** となります。7359 は SIOS 社の製造メーカー番号であり、LifeKeeper を表す 1 が続きます。例えば、LifeKeeper Startup Complete イベントは、OID **.1.3.6.1.4.1.7359.1.0.100** を生成します。

SPS イベントの説明	トラップ番号	オブジェクト ID
LifeKeeper Startup Complete (LifeKeeper の起動完了) LifeKeeper を起動したノードが送信	100	.1.3.6.1.4.1.7359.1.0.100

SPS イベントの説明	トラップ 番号	オブジェクト ID
LifeKeeper Shutdown Initiated (LifeKeeper シャットダウンの開始) シャットダウンを開始したノードが送信	101	.1.3.6.1.4.1.7359.1.0.101
LifeKeeper Shutdown Complete (シャットダウン完了) LifeKeeper のシャットダウンを完了したノードが送信	102	.1.3.6.1.4.1.7359.1.0.102
LifeKeeper Manual Switchover Initiated on Server (LifeKeeper の手動切り替えをサーバ上で開始) 手動切り替えを要求したノードが送信	110	.1.3.6.1.4.1.7359.1.0.110
LifeKeeper Manual Switchover Complete recovered list (LifeKeeper の手動切り替え完了 - リカバリリスト) 手動切り替えを完了したノードが送信	111	.1.3.6.1.4.1.7359.1.0.111
LifeKeeper Manual Switchover Complete failed list (LifeKeeper の手動切り替え完了 - 失敗リスト) 手動切り替えを完了したノードが送信	112	.1.3.6.1.4.1.7359.1.0.112
LifeKeeper Node Failure Detected (LifeKeeper ノードの失敗検出) 失敗を検出したクラスタノードが送信	120	.1.3.6.1.4.1.7359.1.0.120
LifeKeeper Node Recovery Complete recovered list (LifeKeeper ノードのリカバリ完了 - リカバリリスト) 失敗したノードのリソースをリカバリしたクラスタノードが送信	121	.1.3.6.1.4.1.7359.1.0.121
LifeKeeper Node Recovery Complete failed list (LifeKeeper ノードのリカバリ完了 - 失敗リスト) 失敗したノードからのリソースのリカバリに失敗したクラスタノードが送信	122	.1.3.6.1.4.1.7359.1.0.122
LifeKeeper Resource Recovery Initiated (LifeKeeper リソースのリカバリ開始) リソースをリカバリするノードが送信。直後に 131 または 132 トラップが続き、リカバリの完了または失敗を通知	130	.1.3.6.1.4.1.7359.1.0.130
LifeKeeper Resource Recovery Failed (LifeKeeper リソースのリカバリ失敗) リソースの起動に失敗したノード (トラップ 130) が送信	131*	.1.3.6.1.4.1.7359.1.0.131

SPS イベントの説明	トラップ番号	オブジェクト ID
LifeKeeper Resource Recovery Complete (LifeKeeper リソースのリカバリ完了) リソースのリカバ리를完了したノード (トラップ 130) が送信	132	.1.3.6.1.4.1.7359.1.0.132
Mirror State Change (ミラー状態の変更) ミラーのソースであり、ミラーの状態が変更されたノードが送信。ターゲットノードのボリュームレター、ミラーの状態、および IP アドレスを表示 有効なミラーの状態は次のとおり -1: 無効な状態 0: ミラーなし 1: ミラーリング 2: ミラーは再同期中 3: ミラーは中断 4: ミラーは一時停止 5: 再同期は待ち状態	150	.1.3.6.1.4.1.7359.1.0.150
LifeKeeper replicated volume Split Brain detected (LifeKeeper によって複製されているボリュームのスプリットブレイン検出) 両側でミラーがソースであることが LifeKeeper によって検出されたノードが送信。ターゲットノードのボリュームレターおよび IP アドレスを表示	160	.1.3.6.1.4.1.7359.1.0.160
トラップ PDU の追加情報を通知する変数は次のとおり		
トラップメッセージ	すべて	.1.3.6.1.4.1.7359.1.1
リソースタグ	130	.1.3.6.1.4.1.7359.1.2
リソースタグ	131	.1.3.6.1.4.1.7359.1.2
リソースタグ	132	.1.3.6.1.4.1.7359.1.2
リカバリ済みリソースのリスト	111	.1.3.6.1.4.1.7359.1.3
リカバリ済みリソースのリスト	121	.1.3.6.1.4.1.7359.1.3
失敗したリソースのリスト	112	.1.3.6.1.4.1.7359.1.4
失敗したリソースのリスト	122	.1.3.6.1.4.1.7359.1.4

* リカバリが複数のバックアップサーバ上で失敗した場合、このトラップは連続して表示されることがあります。

Chapter 5: ユーザガイド

ユーザガイドは、LifeKeeper GUI で実行できる多数の作業に関する詳細な情報が含まれた、総合的な検索可能リソースです。このドキュメントにアクセスするには、「ユーザガイド」をクリックしてください。

LifeKeeper GUI で実行できる作業は、次の3つに大別されます。

共通タスク - ユーザが実行する基本的な作業です。クラスタへの接続、サーバやリソースのプロパティ表示、ログファイル表示、GUI 設定の変更などです。

オペレータタスク - リソースサービスの起動と停止など、より高度なタスクです。オペレータタスクを実行するには、オペレータ権限が必要です。

管理者タスク - 管理者権限が必要です。サーバを対象とするタスク(サーバプロパティの編集、リソース作成、コミュニケーションパスの作成と削除など)とリソースを対象とするタスク(リソースの編集、拡張、削除)があります。

3種類のユーザ権限で実行できるタスクを次表に示します。リソースタイプによっては、さらに別のタスクも実行できます。詳細については、個々のリソースキットに付属するドキュメントを参照してください。

タスク	権限		
	ゲスト	オペレータ	管理者
サーバとリソースの表示	X	X	X
サーバの接続と切断	X	X	X
サーバプロパティとログの表示	X	X	X
サーバプロパティの変更			X
リソース階層の作成			X
コミュニケーションパスの作成と削除			X
リソースプロパティの表示	X	X	X
リソースプロパティの変更			X
リソースのサービス起動と停止		X	X
リソース階層の拡張と拡張解除			X
リソース依存関係の作成と削除			X
リソース階層の削除			X

LifeKeeper GUI

LifeKeeper グラフィカルユーザインターフェース

SPS Core をインストールすると GUI コンポーネントもインストールされます。

LifeKeeper GUI は、Java 技術を使用して、SPS およびその構成データに対するグラフィカルユーザインターフェースを提供します。LifeKeeper GUI はクライアント/サーバアプリケーションなので、ユーザはクライアントコンポーネントを実行して SPS を監視、管理します。クライアントコンポーネントとサーバコンポーネントは、同一システム上で実行される場合も、異なるシステムで実行される場合もあります。

GUI の概要

LifeKeeper GUI では、適切なアクセス権を行使して、クラスタを構成する任意のサーバ上で他のサーバやリソースを管理、運用、監視できます。詳細については、[GUI ユーザの設定](#)を参照してください。以降に、GUI サーバとクライアントの概要を説明します。

GUI サーバ

GUI サーバは、システムを起動すると、個々の SPS サーバ上で初期化されます。GUI サーバは、HTTP (Hypertext Transfer Protocol) と RMI (Remote Method Invocation) プロトコルを使用して、GUI クライアントと通信します。

GUI クライアント

GUI クライアントは、Java 対応システム上の Web クライアント、SPS サーバ上のアプリケーションとして動作します。

クライアントは次のコンポーネントで構成されています。

- [ステータステーブル](#) (左上): 接続先サーバとリソースの詳細な状態を表示します。
- [プロパティパネル](#) (右上): ステータステーブルで最後に選択したオブジェクトに関する詳細な情報を表示します。
- [アウトプットパネル](#) (下): コマンドの出力を表示します。
- [メッセージバー](#) (ウィンドウの下): 処理状態に関するメッセージを表示します。
- コンテキスト (プロパティパネル) と [ツールバー](#): 使用頻度の高いタスクに素早くアクセスできます。
- コンテキスト (ポップアップ) とグローバル [メニュー](#): すべてのタスクにアクセスできます。

LifeKeeper GUI クライアントの起動

Web クライアントの起動

SPS サーバ上の Web クライアントを起動するには、[スタート] -> [すべてのプログラム] -> [SteelEye] -> [LifeKeeper] -> [LifeKeeper] の順に選択してください。Web ブラウザが起動し、ローカルの GUI サーバ (<http://localhost:81>) に接続します。

SPS クラスタ外のシステムを使用する場合は、Web ブラウザを開き、URL に <http://<server name>:81> を指定してください。ここで、<server name> には、SPS サーバの名前を入力します。GUI サーバから Web クライアントがロードされます。

Web クライアントのロードが完了すると、[クラスタへの接続] ダイアログボックスが表示されます。Web クライアントを任意の GUI サーバに接続します。

注記: Web クライアントを実行した場合、適切な Java プラグインがシステムにインストールされていないと、プラグインのダウンロードサイトに自動的にジャンプします。[ブラウザのセキュリティを設定し、Java を有効にする](#)の必要もあります。

以上の手順を実行しても、クライアントがロードされない場合は、[Web クライアントのトラブルシューティング](#)を参照してください。

アプリケーションクライアントの起動

SPS サーバの管理者権限を持つユーザは、サーバ上でアプリケーションクライアントを起動できます。[スタート] メニューをクリックし、[プログラム] -> [LifeKeeper] -> [LifeKeeper GUI] の順に選択してください。

以上の操作を実行しても、クライアントをロードできない場合は、[ネットワーク関連のトラブルシューティング](#)を参照してください。

GUI クライアントの終了

[ファイル] メニューの [終了] をクリックし、すべてのサーバとの接続を切断し、GUI ウィンドウを閉じます。

ステータステーブル

ステータステーブルには、接続先サーバとリソースの状態がグラフィカルに表示されます。表示される項目は、次のとおりです。

- 最上位の行: サーバの状態
- 左側の列: グローバル(サーバ間)の状態とリソースの親子関係
- 残りのセル: 各サーバにあるリソースの状態

サーバとリソースの状態は、グラフィックスとテキストがさまざまな色で示されます。サーバの下にあるテーブルセルが空の場合は、そのサーバにリソースが定義されていないことを示します。

ステータステーブルでサーバまたはリソースを選択すると、詳細な状態情報とコンテキストツールバーが [プロパティパネル](#) に表示されます。セルを右クリックすると、[サーバコンテキストメニュー](#) または [リソースコンテキストメニュー](#) が表示されます。

ステータステーブルは、2つのエリアに分かれています。左右のパネルのサイズは、デバイダを使用して変更できます。ステータステーブルは、折りたたんで最上位の項目だけを表示できます。[リソース階層ツリーを展開/縮小](#)すると、表内の階層も展開/縮小します。

プロパティパネル

プロパティパネルには、[ステータステーブル](#)で選択したサーバやリソースのプロパティが表示されます。プロパティパネルの機能は、[\[サーバプロパティ\] ダイアログボックス](#)、[\[リソースプロパティ\] ダイアログボックス](#)、コンテキストツールバーと同じであり、使用頻度の高いコマンドを簡単に使用できるようになっています。プロパティパネルのタイトルバーには、サーバを選択すると `server_name`、リソースを選択すると `server_name: resource_name` がそれぞれ表示されます。

プロパティパネルのコンテキストツールバーは、[サーバコンテキストツールバー](#)と[リソースコンテキストツールバー](#)で構成されています。コンテキストツールバーはカスタマイズできます。

プロパティパネルの下にあるボタンは、次の機能を備えています。

- **適用**: パネル内で編集可能なプロパティに加えた変更を反映します。このボタンは、編集可能なプロパティを変更すると有効になります。
- **更新**: すべてのプロパティの値をサーバに問い合わせ、古い変更をクリアします。このボタンは常に有効になっています。

プロパティパネルのサイズを変更するには、パネルの左側にあるセパレータを左右にスライドしてください。プロパティパネルを表示/非表示にするには、[\[表示\] メニュー](#)の [プロパティパネル](#) チェックボックスを使用してください。

アウトプットパネル

アウトプットパネルは、GUI クライアントが発行したコマンドの出力を収集します。コマンドを実行すると、タイムスタンプがアウトプットパネルにラベルとして追加され、コマンドの出力がその下に追加されます。複数のサーバからコマンドを発行し、複数のコマンドを同時に実行した場合、各コマンドの出力は対応するセクションに送信されるため、結果を簡単に確認できます。

アウトプットパネルのサイズを変更するには、パネルの左側にあるセパレータを上下にスライドしてください。アウトプットパネルを表示/非表示にするには、[\[表示\] メニュー](#)の [アウトプットパネル](#) チェックボックスを使用してください。アウトプットパネルを閉じて、コマンドが起動したダイアログボックスは、そのまま表示されます。出力結果は、ユーザがクリアするまで表示されます。しかし、ダイアログボックスを閉じると、コマンドの出力は表示されなくなります。アウトプットパネルを再び開くと、GUI はデフォルトの動作に戻ります。

メッセージバー

メッセージバーはステータスウィンドウの下に表示されます。メッセージバーには、1行のメッセージが表示されます。例えば、「Connecting to Server X」、「Failure to connect to Server X」などのメッセージが表示されます。

- メッセージバーを隠すには、[\[表示\] メニュー](#)の [\[メッセージバー\]](#) チェックボックスをオフにします。
- メッセージバーを表示するには、[\[表示\] メニュー](#)の [\[メッセージバー\]](#) チェックボックスをオンにします。

- メッセージバーに表示されたメッセージの履歴を表示する方法については、[メッセージ履歴の表示](#)を参照してください。







ツールバー

SteelEye Protection Suite for Windows ツールバー

Global Toolbar

ツールバーは、デフォルトの[サーバコンテキストツールバー](#)と[リソースコンテキストツールバー](#)で構成されており、[プロパティパネル](#)に表示されます。ツールバーを使用するには、アクションを実行するサーバやリソースを選択する必要があります。



	接続 : クラスタに接続します。
	切断 : クラスタから切断します。
	更新: GUI を更新します。
	ログの表示 : ログメッセージを表示します。
	リソース階層の作成 : リソース階層を作成します。
> 	リソース階層の削除 : 全サーバのリソース階層を削除します。

	コミュニケーションパスの作成 : サーバ間にコミュニケーションパスを作成します。
	コミュニケーションパスの削除 : サーバのコミュニケーションパスを削除します。
	サービス開始 : リソース階層をサービス中にします。
	サービス停止 : リソース階層をサービス停止にします。
	Extend Resource Hierarchy : フェイルオーバーを実行できるように、リソース階層を別のサーバにコピーします。
	リソース階層の拡張解除 : 拡張したリソース階層を1台のサーバから削除します。
	依存関係の追加 : 2つのリソースに親子関係を作成します。
	依存関係の削除 : 2つのリソースの親子関係を削除します。

リソースコンテキスト ツールバー

リソースコンテキストツールバーは、[ステータステーブル](#)でサーバ固有のリソースを選択すると、[プロパティパネル](#)に表示されます。このページでは、デフォルトのリソースコンテキストツールバーについて説明しますが、リソースコンテキストツールバーはリソースタイプ単位でカスタマイズできます。カスタマイズの詳細については、リカバリキットに付属しているドキュメントを参照してください。

ツールバーをクリックして、所定のアクションをリソースとサーバに実行します。選択したリソースに適用できないアクションは淡色で表示されます。









	サービス開始 : リソース階層を In Service にします。
	サービス停止 : リソース階層を Out of Service にします。
	Extend Resource Hierarchy : フェイルオーバーを実行できるように、リソース階層を別のサーバにコピーします。
	リソース階層の拡張解除 : 拡張したリソース階層を1台のサーバから削除します。
	依存関係の追加 : 2つのリソースに親子関係を作成します。
	依存関係の削除 : 2つのリソースの親子関係を削除します。
	リソース階層の削除 : 全サーバのリソース階層を削除します。

サーバコンテキストツールバー

サーバコンテキストツールバーは、[ステータステーブル](#)でサーバを選択すると、[プロパティパネル](#)に表示されます。ツールバーを選択して、所定のアクションをサーバに実行します。



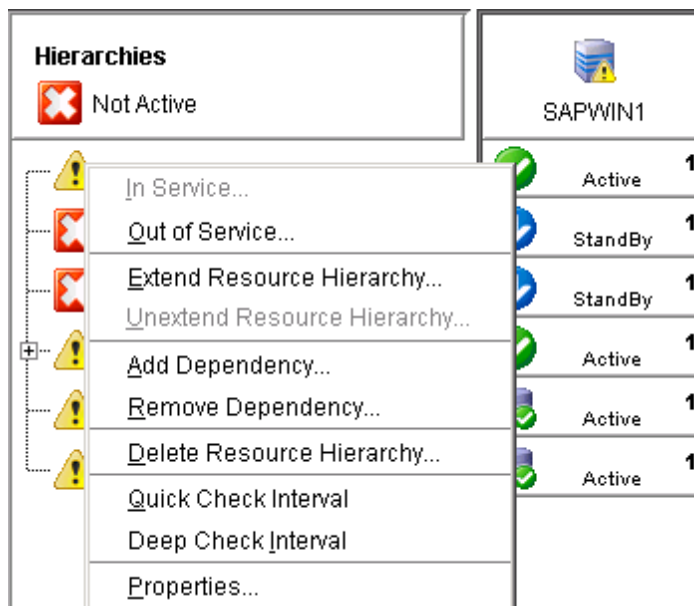
	切断 : クラスタから切断します。
	更新 : GUIを更新します。

	ログの表示 : ログメッセージを表示します。
	リソース階層の作成 : リソース階層を作成します。
	コミュニケーションパスの作成 : サーバ間にコミュニケーションパスを作成します。
	コミュニケーションパスの削除 : サーバのコミュニケーションパスを削除します。

メニュー

SteelEye Protection Suite for Windows メニュー

リソースコンテキストメニュー



リソースコンテキストメニューは、クラスタ全体に共通するグローバルリソース(上図)またはサーバ固有のリソースインスタンス(下図)を[ステータステーブル](#)で右クリックすると表示されます。このページでは、デフォ

ルトのリソースコンテキストメニューについて説明しますが、リソースコンテキストメニューはリソースタイプ単位でカスタマイズできます。カスタマイズの詳細については、リカバリキットに付属しているドキュメントを参照してください。

メニューを選択して、リソースに所定のアクションを実行します。特定のサーバのリソースインスタンスを選択して、アクションを実行します。また、クラスタ全体に共通するグローバルリソースを選択する場合は、アクションを実行するサーバを選択します。

サービス開始: リソース階層を In Service にします。

サービス停止: リソース階層を Out of Service にします。

Extend Resource Hierarchy: フェイルオーバーを実行できるように、リソース階層を別のサーバにコピーします。

リソース階層の拡張解除: 拡張したリソース階層を1台のサーバから削除します。

依存関係の追加: 2つのリソースに親子関係を作成します。

依存関係の削除: 親子関係を削除します。

リソース階層の削除: 全サーバのリソース階層を削除します。

ローカルリカバリ: リソースに対するローカルリカバリを有効にするには、**[はい]**を選択してください。ファイル共有のローカルリカバリは、フォルダにアクセスできなくなった場合、SPS は、フォルダの再作成を行うということを意味します。

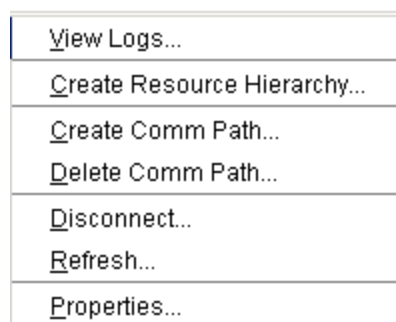
クイックチェック間隔: リソースの基本的な状態確認の間隔を分単位で入力してください。それぞれのシステムに異なる値を指定できます。デフォルト値は3分です。設定値の範囲は0から10080です。間隔を0にするとクイックチェックは無効となります。

ディープチェック間隔: リソースの広範囲にわたる状態確認の間隔を分単位で入力してください。ディープチェックの場合は、ディープチェックとともにクイックチェックも行います。それぞれのシステムに異なる値を指定できます。デフォルト値は5分です。設定値の範囲は0から10080です。間隔を0にするとディープチェックは無効となります。

プロパティ: [リソースプロパティ] ダイアログを表示します。

サーバコンテキストメニュー

サーバコンテキストメニューは、**ステータステーブル**でサーバを右クリックすると表示されます。メニューを選択して、所定のアクションをサーバに実行します。



[ファイル] メニュー

[ログの表示](#): SPS ログメッセージを表示します。

[リソース階層の作成](#): リソース階層を作成します。

[コミュニケーションパスの作成](#): サーバ間にコミュニケーションパスを作成します。

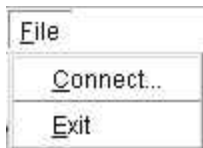
[コミュニケーションパスの削除](#): コミュニケーションパスを削除します。

[切断](#): クラスタから切断します。

更新: GUI を更新します。

[プロパティ](#): [\[サーバプロパティ\] ダイアログ](#)を表示します。

[ファイル] メニュー

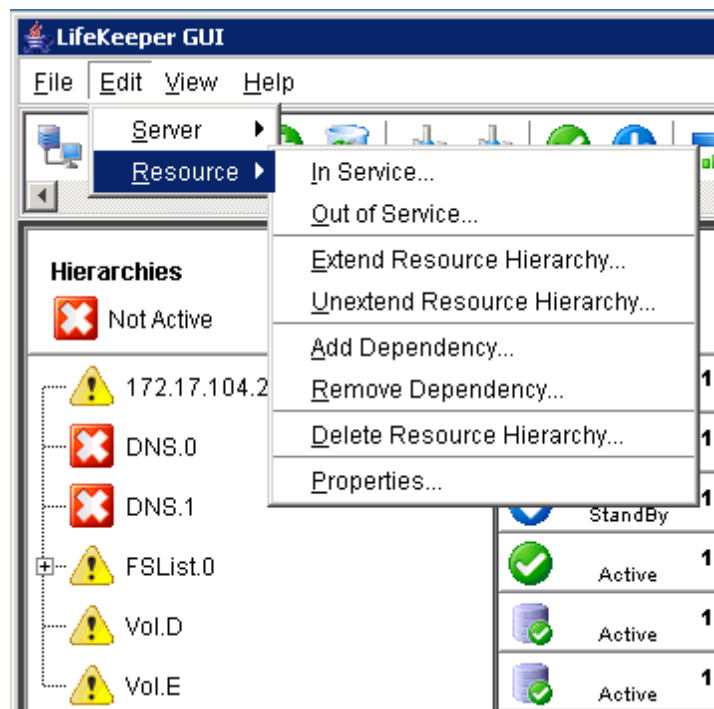


[接続](#): SPS クラスタに接続します。サーバごとにログイン認証が必要です。

終了: すべてのサーバから切断し、GUI ウィンドウを閉じます。

[編集] メニュー - リソース

このメニューは、デフォルトの[リソースコンテキストメニュー](#)と基本的に同じ動作ですが、アクションを実行するリソースとサーバを選択する必要があります。**[編集]**メニューの**[リソース]**コマンドは、カスタマイズできません。



サービス開始: リソース階層をサービス中にします。

サービス停止: リソース階層をサービス停止にします。

Extend Resource Hierarchy: フェイルオーバを実行できるように、リソース階層を別のサーバにコピーします。

リソース階層の拡張解除: 拡張したリソース階層を1台のサーバから削除します。

依存関係の追加: 2つのリソースに親子関係を作成します。

依存関係の削除: 親子関係を削除します。

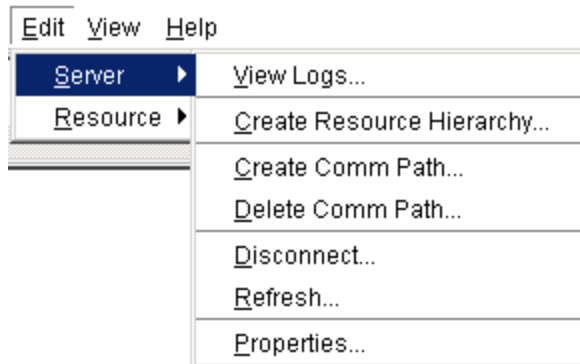
リソース階層の削除: 全サーバのリソース階層を削除します。

プロパティ: [\[リソースプロパティ\]ダイアログ](#)を表示します。

[編集]メニュー - サーバ

このメニューは、デフォルトのサーバコンテキストメニューと基本的に同じ動作ですが、アクションを実行するサーバを選択する必要があります。**[編集]**メニューの**[サーバ]**コマンドは、カスタマイズできません。

[表示]メニュー



[ログの表示](#): SPS ログメッセージを表示します。

[リソース階層の作成](#): リソース階層を作成します。

[コミュニケーションパスの作成](#): サーバ間にコミュニケーションパスを作成します。

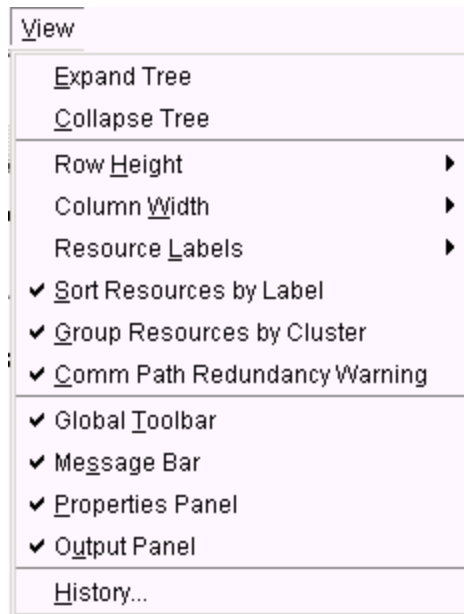
[コミュニケーションパスの削除](#): コミュニケーションパスを削除します。

[切断](#): クラスタから切断します。

[更新](#): GUIを更新します。

[プロパティ](#): [\[サーバプロパティ\]ダイアログ](#)を表示します。

[表示]メニュー



[ツリーを展開](#): ステータステーブルを展開して、すべてのリソース階層を表示します。

ツリーを折りたたむ: ステータステーブルを折りたたみ、各階層の最上位にあるリソースだけを表示します。

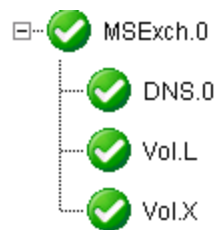
行の高さ: リソース階層ツリーおよびリソーステーブルの行の高さを修正します。表示されたリソースの数により大、中、小を選択してください。

列の幅: リソース階層ツリーおよびリソーステーブルの列幅を修正します。表示されたリソースにより自動、大、中、小を選択してください。

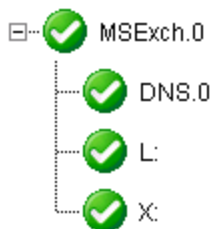
リソースラベル

このオプショングループは、リソースが、タグ名とID のどちらでリソース階層ツリー内に表示されるか指定することができます。

タグ名



ID



ラベルでソートする: リソースラベルのみでリソースを分類します。

クラスタ単位でリソースをグループ: クラスタサーバおよびリソースラベルによって分類します。同じクラスタ内に属するリソースが同じグループになります。

コミュニケーションパスが冗長化されていない場合に警告する: サーバステータスグラフィックでコミュニケーションパスの状態についての説明を明記します。

- 選択した場合は、一組のサーバ間のコミュニケーションパスが冗長化されていない場合、サーバ警告が表示されます。
- 選択しない場合は、一組のサーバ間のコミュニケーションパスが冗長化されていない場合は無視しますが、コミュニケーションパスが切れた場合には、サーバ警告が表示されます。

ツールバー: チェックボックスをオンにするとコンポーネントが表示されます。

ヘルプ]メニュー

[メッセージバー](#): チェックボックスをオンにするとコンポーネントが表示されます。

[プロパティパネル](#): チェックボックスをオンにするとコンポーネントが表示されます。

[アウトプットパネル](#): チェックボックスをオンにするとコンポーネントが表示されます。

[履歴](#): [メッセージ履歴] ダイアログボックスのメッセージバーに最新のメッセージが表示されます。

ヘルプ]メニュー



ヘルプメニューは、リリースノート および **テクニカルドキュメンテーション** をリンクします。:

リリースノート: リリースノートは、新機能だけではなく製品の最終テスト時に明らかになったパッケージのバージョン、指示や手順に関する最終段階での変更点、トラブルシューティングセクションへのリンク、製品の制限、トラブル解決のヒントなどが記載されています。このドキュメントには、インストール時とその前後に留意すべき最新情報が記載されているのでソフトウェアのインストールおよび設定前に必ず確認してください。

テクニカルドキュメンテーション: オンラインドキュメントは、利用しやすいフォーマットで Steel Eye 製品に関する最新情報や詳細情報などを提供できるようデザインされています。SIOS Technology Corp. はこのサイトでサポート対象の SteelEye 製品のドキュメントを管理しています。旧バージョンのドキュメントについては support@us.sios.com までご依頼ください。

About: LifeKeeper GUI のバージョン情報を表示します。

LifeKeeper GUI サーバコンポーネントおよびクライアントコンポーネント

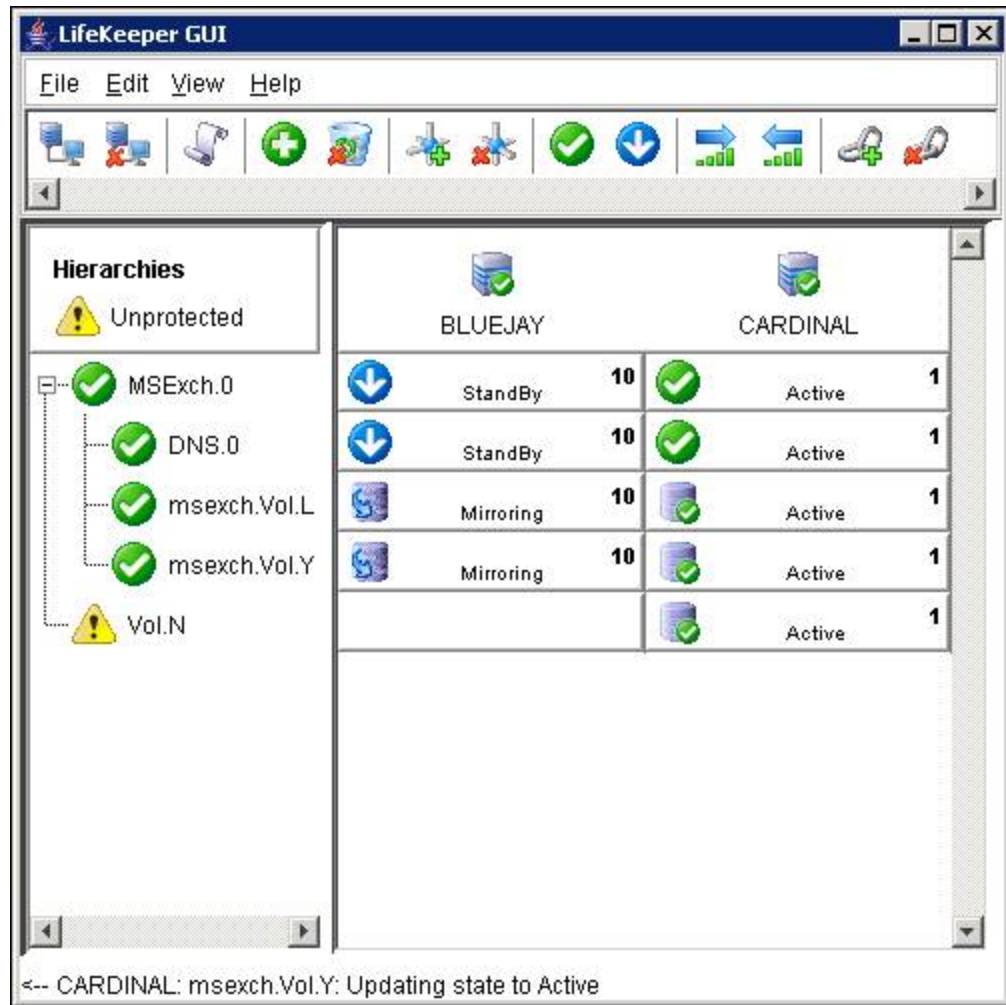
LifeKeeper GUI サーバは、システムを起動すると、個々の SPS サーバ上で初期化されます。LifeKeeper GUI サーバは、HTTP (Hypertext Transfer Protocol) と RMI (Remote Method Invocation) プロトコルを使用して、LifeKeeper GUI クライアントと通信します。

LifeKeeper GUI サーバに接続できるのは、クラスタ内のすべてのサーバのポート 81 および 82 に接続できる任意のシステムから実行できる Web クライアント、または SPS に付属し、クラスタ内のサーバ上で実行するように設計されたアプリケーションクライアントです。

両方の SPS クライアントに同じグラフィカルコンポーネントが含まれています。

- ポップアップのサーバコンテキストメニューおよびリソースコンテキストメニュー: サーバ関連およびリソース関連の操作にアクセスできます。
- メニューバー: LifeKeeper GUI のすべての操作にアクセスできます。

- ツールバー: 多数の SPS の操作に素早くアクセスできます。
- ステータスウィンドウ: クラスタ内で接続されたサーバ、リソース階層、およびリソースとサーバのステータスがグラフィカルに表示されます。
- メッセージバー (ウィンドウの下): 処理状態が表示されます。



SPS クライアントの実行

クラスタの外部のシステムから SPS を管理する場合は、Web クライアントを使用する必要があります。これは、どのオペレーティングシステムを実行するリモートシステムからでも可能です。SPS for Windows は、単一セッションで Linux サーバと Windows サーバの両方を管理することはできませんが、OS に関わらず、Windows システムまたは Linux システムのいずれかを管理できます。最初に接続するサーバ OS のタイプによって、そのセッションで管理できる OS のタイプが決まります。Linux サーバと Windows サーバの両方を同時に管理する必要がある場合は、それぞれに 1 つずつ、2 つのブラウザウィンドウを開く必要があります。

ブラウザのセキュリティレベルの設定

リモートシステムのブラウザは、JRE 1.4 以降をサポートする必要があります。SPS Web クライアントがサポートされるプラットフォームとブラウザについては、SPS for Windows リリースノートを参照してください。以降では、リモートシステムで Web ブラウザを設定するための手順を説明します。

以下の手順に従って、SPS Web クライアントを実行してください。

1. SPS Web ページの URL `http://<server name>.81` (<server name> は SPS サーバの名前) を開きます。Web ページには SPS スプラッシュ画面とアプレットが含まれています。

Internet Explorer または Netscape を使用して Web クライアントを初めて実行した場合、適切な Java プラグインがシステムにインストールされていないと、プラグインのダウンロードサイトに自動的にジャンプします。

注記

- プラグインをインストールした場合およびプラグインのプロパティを変更した場合は、ブラウザをいったん閉じてから再起動してください。上記の SPS サーバの URL を再入力する必要があります。

Web ページが開くと、以下の処理が実行されます。

- スプラッシュ画面の表示
- アプレットの読み込み
- Java 仮想マシンの起動
- サーバファイルのダウンロード
- アプレットの初期化

ネットワークとシステムの構成によっては、これらの処理に最大 20 秒かかる場合があります。通常、アプレットの読み込み時や初期化時にはブラウザに最小限のステータスが表示されます。

注記: 「証明書を確認できません。コードは未署名として扱われます」という Java プラグインのセキュリティ警告が表示される場合があります。[OK] をクリックしてください。

次に、スプラッシュ画面の下のアプレット領域に [起動] ボタンが表示されます。スプラッシュ画面に [起動] ボタンが表示されない場合や、アプレットの読み込みと初期化に失敗したと思われる場合は、このガイドの [GUI ネットワーク関連のトラブルシューティング](#) を参照してください。

2. **[起動]** をクリックします。SPS Web クライアントが起動し、[クラスタへの接続] ダイアログボックスが自動的に表示されます。接続先サーバの名前を **[サーバ名]** に入力し、**ログイン名** と **パスワード** を入力します。サーバ名を入力してクラスタへの接続が確立されると、GUI ウィンドウが表示されます。

注記: 一部のブラウザでは、アプレットによって作成されるウィンドウとダイアログに「警告: アプレットウィンドウ」が追加されます。これは正常なので、無視してください。

ブラウザのセキュリティレベルの設定

SPS Web クライアントを実行するには、ブラウザのセキュリティ設定の変更が必要になる場合があります。以下の手順に従ってください。

Internet Explorer

Internet Explorer では、通常、SPS サーバは自動的にローカルイントラネットゾーンに配置されます。配置されていない場合、以下のように、すべての SPS サーバをローカルイントラネットゾーンに手動で追加してください。

1. [ツール] メニューの **[インターネットオプション]** をクリックします。
2. **[セキュリティ]** タブをクリックします。
3. **[ローカルイントラネット]** を選択します。
4. **[サイト]** をクリックします。
5. **[詳細設定]** をクリックします。
6. 接続する SPS サーバのサーバ名とポート番号 (例: `http://server1:81`) を入力します。複数ある場合は、入力するたびに **[追加]** をクリックします。
7. 完了したら、**[OK]** をクリックします。

Mozilla Firefox

1. [ツール] メニューの **[オプション]** をクリックします。
2. [オプション] ダイアログボックスの **[コンテンツカテゴリ]** をクリックします。
3. **[Java を有効にする]** と **[Java スクリプトを有効にする]** オプションをオンにします。
4. **[OK]** をクリックします。

Netscape Navigator と Netscape Communicator

1. [編集] メニューの **[設定]** をクリックします。
2. [設定] ダイアログボックスの **[詳細]** をダブルクリックします。
3. **[Java を有効にする]** と **[Java スクリプトを有効にする]** オプションをオンにします。
4. **[OK]** をクリックします。

警告: セキュリティを低く設定した場合は、他のサイトへのアクセスに注意してください。

SPS サーバ上の GUI アプリケーションの実行

LifeKeeper GUI を SPS サーバ上のアプリケーションとして実行することもできます。こうすると、事実上、GUI クライアントとサーバを同一システム上で実行できます。SPS サーバで管理者権限を持つユーザのみが SPS アプリケーションの実行を許可されます。

1. LifeKeeper GUI を起動するには、**[スタート]** -> **[プログラム]** -> **[SteelEye]** -> **[LifeKeeper]** -> **[LifeKeeper GUI]** の順にクリックします。
2. アプリケーションを読み込むと、LifeKeeper GUI が開き、**[クラスタへの接続]** ダイアログボックスが表示されます。接続先サーバの名前を **[サーバ名]** に入力し、ログイン名とパスワードを入力しま

す。ログインの詳細は、[LifeKeeper GUI ユーザアカウント](#)を参照してください。

3. クラスタへの接続が確立されると、GUI ウィンドウが表示されます。

Web クライアントを使用してSPS サーバ上で LifeKeeper GUI を実行するには、[スタート] -> [プログラム] -> [SteelEye] -> [LifeKeeper] -> [LifeKeeper] の順にクリックしてください。Web ブラウザが起動し、LifeKeeper (*localhost:81*) に接続します。

LifeKeeper GUI ユーザアカウント

すべての LifeKeeper GUI ユーザは SPS セキュリティグループに属する必要があります。クラスタの SPS 管理者は、各サーバでローカルグループとユーザアカウントを使用できます。あるいは、ドメイングループとローカルログオン権限を持ったユーザを設定できます。

ログイン

SPS アカウントがクラスタ内の各サーバで同じ (同じログイン名とパスワード) である場合は、クラスタ内の 1 台のサーバにログインすると、追加のログイン手順を要求されずに他のサーバにアクセスできます。ユーザ名とともにドメイン名を入力する必要がある場合もあります (例: Southdomain\john)。

SPS アカウントがクラスタ内の各サーバで異なる (ログイン名またはパスワードの少なくとも一方が異なる) 場合は、クラスタ内の最初のサーバにログインすると、クラスタ内の次のサーバにログインしようとしたときに以下のメッセージが表示されます。

```
Access denied: invalid user name or bad password. Only
users with local privileges can use LifeKeeper. Would you
like to re-enter the authentication data?
```

次のサーバにログインするためのプロンプトを表示するには、[はい]をクリックしてください。

GUI ユーザの設定

GUI ユーザには次の 3 種類があり、それぞれ異なるアクセス権が付与されます。

1. 管理者: LifeKeeper GUI を使用して、クラスタ全体を管理する権限を付与します。
2. オペレータ: 特定のサーバを対象として、構成情報とステータス情報の参照、リソースの起動と停止を実行する権限を付与します。
3. ゲスト: 特定のサーバの構成情報とステータス情報を参照する権限を付与します。

アクセス権の付与は、クラスタ全体を対象とすることを推奨します。アクセス権をサーバ単位で設定すると、ユーザが混乱し、管理作業が繁雑になります。

ユーザ管理では、各サーバのローカルユーザグループにユーザを割り当てます。ローカルの Administrators グループに割り当てたユーザは、管理者権限を持ちます。ローカルの `LK_OPERATOR` グループに割り当てたユーザは、オペレータ権限を持ちます。ローカルの `LK_GUEST` グループに割り当てたユーザは、ゲスト権限を持ちます。ローカルの Administrators グループは、Windows OS にあらかじめ設定されていますが、残り 2 つのローカルグループは、LifeKeeper 管理者が作成します。

英語版の Windows でグループ名を設定するには、`Server_RB_en.properties` ファイルのエントリを編集します。このファイルは、`$LKROOT/htdoc/com/steeleye/LifeKeeper/locale` に格納されています。グループ名をローカライズするには、同じフォルダに `Server_RB_xx.properties` ファイルを作成し、エントリを編集してください。ここで、ファイル名の「xx」には、ローカルコードを記述します。

ドメインコントローラで管理するローカルグループとユーザが未定義の場合は、信頼できるグローバルセキュリティグループとして、`LK_OPERATOR` と `LK_GUEST` グループを作成してください。次に、グループのセキュリティポリシーを作成し、ローカルからのログオンを各グループに許可してください。

ローカルの Windows サーバへのログオンを許可するには、次の手順に従ってください。

1. ローカルの管理者権限でサーバにログインします。
2. Windows の管理ツールにあるローカルセキュリティポリシー MMC を起動します。
3. [ローカルポリシー] から [ユーザ権限の割り当て] までスクロールします。
4. 詳細ペインに移動し、[ローカルのログオンを許可する] ポリシー (Windows 2003) をダブルクリックします。
5. [ユーザとグループの追加...] をクリックし、前記の手順で作成した `LK_OPERATOR` と `LK_GUEST` ドメイングループを [ローカルログイン権限] に追加します。

重要: ドメイン GPO (グループポリシーオブジェクト) によりローカルポリシーの変更が上書きされていないことを確認してください。

Windows Server 2003 の場合は、最後に `SECEDIT /REFRESHPOLICY USER_POLICY gpupdate` コマンド (詳細は <http://support.microsoft.com/?kbid=227302> を参照) を実行して、これらの変更を有効にする必要があります。一度このコマンドを実行すると、SPS は、それらのグループを認識して適切なアクセス権を割り当てることができます。

注記: ローカルサーバにグループとユーザを作成した場合、アクセス権の割り当ては、そのサーバの GUI 権限だけに適用されます。ローカルサーバだけを設定した場合は、同じ操作をクラスタサーバ全体に繰り返してください。このような手順を踏むと、作業量は増えますが、ドメインコントローラにアクセスする必要がないため、クラスタがより堅牢になります。

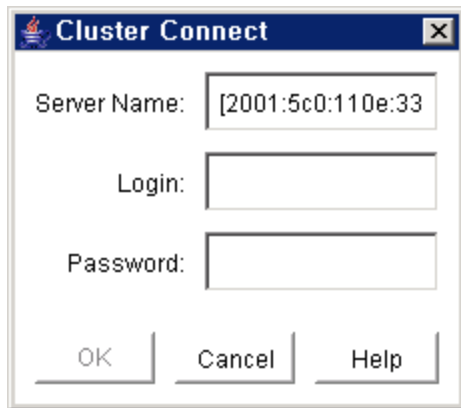
共通タスク

このセクションでは、すべてのユーザが実行できる基本的なタスクについて説明します。

クラスタへの接続

1. [ファイル] メニューまたは [ツールバー](#) から [接続] をクリックします。

注記: IPv6 アドレスを使用する場合は、そのアドレスを括弧 [] で囲む必要があります。その結果 IPv6 アドレスを通して接続が確立されます。あるいはそのアドレスに名前を割り当てると、その名前を接続に使用することができます。



2. の **[サーバ名]** フィールドに、**[クラスタへの接続]** ダイアログボックスクラスタ内の接続先サーバの名前を入力します。
3. 接続先サーバのSPSを管理するユーザのログイン名とパスワードを **[ログイン]** と **[パスワード]** フィールドに入力します。
4. **[OK]** をクリックします。

指定したサーバに正常に接続すると、GUIは新しいサーバが検出されなくなるまでクラスタを構成する既知のサーバすべてに接続し、接続したサーバをステータス表示に追加します。

注記: クラスタ内の接続先サーバで最初に入力したログイン名とパスワードによるクライアント認証に失敗した場合、ユーザはそのサーバで利用できる別のログイン名とパスワードの入力を求められます。の **[キャンセル]** をクリックすると、**[パスワード]** ダイアログボックス指定したサーバへの接続は中止されますが、クラスタ内の他のサーバへの接続処理は続行されます。

クラスタからの切断

クラスタを構成する全サーバからクライアントを切断します。

1. 切断するサーバを選択し、[サーバコンテキストメニュー](#)または[サーバコンテキストツールバー](#)の **[切断]** をクリックします。
2. クラスタを構成する全サーバが **[確認]** ダイアログボックスに表示されます。**[確認]** ダイアログボックスの **[OK]** をクリックし、クラスタ内の全サーバから切断することを確認します。

クラスタを切断すると、クラスタを構成する全サーバが [ステータステーブル](#) から削除されます。

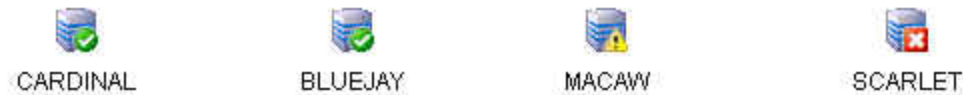
接続先サーバの表示

サーバの状態は、GUI画面にアイコンで表示されます(次図)。アイコンの意味については、[サーバステータスの表示](#) を参照してください。



サーバステータスの表示

サーバステータスは、次図に示すように、テーブルヘッダにアイコンで表示されます。



サーバの状態	アイコン	表示状態
ALIVE (稼働中)		クライアントはサーバに接続しています。 サービスを提供しているリモートサーバとのコミュニケーションパスは稼働しています。 停止中のサーバは独自のアイコンで表示されるため、停止中のコミュニケーションパスとサーバは無視されます。
ALIVE (稼働中)		クライアントはサーバに接続しています。 リモートサーバ間のコミュニケーションパスは、1本以上が停止しています。 リモートサーバとの間に冗長なコミュニケーションパスがありません。
DEAD (停止)		他のクラスタサーバから停止が報告されています。
UNKNOWN (不明)		ネットワークが切断されました。「不明」になる前は、「稼働中」でした。

サーバログファイルの表示

サーバログファイルを表示するには、以下の操作を実行してください。

1. サーバを選択し、[サーバコンテキストメニュー](#)または[サーバコンテキストツールバー](#)の[ログの表示]をクリックします。[\[ログビューア\]ダイアログボックス](#)が開きます。
2. ログを閲覧したら、[OK]をクリックして、ダイアログボックスを閉じます。

サーバプロパティの表示

サーバプロパティを表示するには、以下の操作を実行してください。

- [プロパティパネル](#)が有効な場合、[ステータステーブル](#)からサーバを選択すると、[プロパティパネル](#)にサーバのプロパティが表示されます。

- プロパティパネルが無効な場合は、サーバを選択し、[サーバコンテキストメニュー](#)の[プロパティ]をクリックしてください。

リソースタグとIDの表示

リソースのタグとIDを表示するには、ステータスウィンドウのリソースアイコン上にマウスカーソルを移動し、マウスを左クリック(シングルクリック)してください。優先順位が最も低いサーバのタグとIDがメッセージバーに表示されます。サーバを指定して、リソースタグとIDを表示するには、テーブルでリソースインスタンスのセルをシングルクリックしてください。

次のようなメッセージがメッセージバーに表示されます。

Resource Tag = F-Drive, Resource ID = F:

状況によっては、GUIがリソースIDを特定できないことがあります。その場合は、リソースタグだけがメッセージバーに表示されます。

リソースステータスの表示





リソースのステータス(状態)は次の2種類の形式で表示されます。**グローバルリソースステータス**(すべてのサーバのリソース)と**サーバリソースステータス**(サーバ単体のリソース)です。グローバルリソースステータスは、ステータスウィンドウの左側のペインにある**リソース階層ツリー**に表示されます。サーバリソースステータスは、表内でリソース行とサーバ列が交差するセルに表示されます。

サーバリソースのステータス

サーバリソースの状態	アイコン	表示状態
アクティブ		このサーバでは、リソースが稼働しており、LifeKeeperで保護されています(ISP)。
縮退		リソースは稼働していますが、バックアップリソースで保護されていません(ISU)。
スタンバイ		バックアップリソースは、アクティブなリソースをフェイルオーバーできる状態にあります(OSU)。
障害		障害が発生しています。リソースの起動に失敗したことが考えられます(OSF)。
不明		リソースが初期化されていないか(ILLSTATE)、LifeKeeperが稼働していません。

サーバリソースの状態	アイコン	表示状態
	(表示なし)	リソースが定義されていません。

グローバルリソースのステータス

説明	アイコン	意味/原因
正常		リソースがアクティブであり (ISP)、バックアップもすべてアクティブです。
警告		リソースはアクティブです (ISP)。1 つ以上のバックアップが不明であるか、障害が発生しています (OSF)。
障害 いずれかのサーバでリソースがアクティブではありません (OSF)		リソースは正常な理由によりサービスが停止されています。 リソースは異常な方法で停止されました。 リカバリは完了していないか、失敗しています。
不明 現在のデータだけではリソースの状態を判定できません		複数のサーバでリソースが起動しています。 サーバとの接続が切断されています。 すべてのサーバリソースインスタンスが不明な状態です。

リソースプロパティの表示

リソースプロパティを表示するには、以下の操作を実行してください。

- [プロパティパネル](#)が有効な場合、サーバ固有のリソースインスタンスを[ステータステーブル](#)から選択すると、[プロパティパネル](#)にプロパティが表示されます。
- [プロパティパネル](#)が無効な場合は、サーバ固有のリソースインスタンスを選択し、[リソースコンテキストメニュー](#)の[\[プロパティ\]](#)をクリックしてください。

メッセージ履歴の表示

1. [\[表示\]](#)メニューの[\[履歴\]](#)をクリックします。[\[メッセージ履歴\]](#)ダイアログボックスが開きます (次図)。
2. 履歴を完全にクリアするには、[\[クリア\]](#)をクリックします。
3. [\[OK\]](#)をクリックしてダイアログボックスを閉じます。

メッセージ履歴の読み方

[メッセージ履歴] ダイアログボックスには、メッセージバーに表示された最新のメッセージが表示されます。履歴には 1,000 行まで表示できます。新しいメッセージが到着し、メッセージが 1,000 行を超えると、古いメッセージから削除されます。

メッセージは、クライアントとサーバ間のアクションだけを表示します。リストでは最新のメッセージが一番上に表示されます。

メッセージ履歴の読み方

<--: サーバが送信したメッセージです。次の形式で表示されます。

```
<--"server name":"action"  
  
<--"server name":"app res": "action"  
  
<--"server name":"res instance":"action"
```

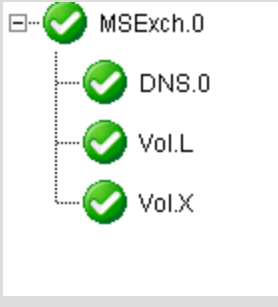


-->: クライアントが送信したメッセージです。次の形式で表示されます。

```
-->"server name":"action"  
  
-->"server name":"app res": "action"  
  
-->"server name":"res instance":"action"
```



[クリア] をクリックすると、履歴がクリアされますが、ダイアログボックスは閉じません。

[OK] をクリックすると、履歴をクリアせずに、ダイアログボックスが閉じます。

リソース階層ツリーの展開と縮小

	<p>MSExch.0 リソースは折りたたまれています。</p> <p>MSExch.0 リソースは展開されています。</p> <p> は、折りたたまれている場合に表示されます。</p> <p> は、ツリーを展開すると、リソースアイコンの左側に表示されます。</p>
---	---

リソース階層を展開するには、以下のいずれかの操作を行います。



-  をクリックする。
-  の右側にあるリソースアイコンをダブルクリックする。

リソース階層をすべて展開するには、以下のいずれかの操作を行います。

- [\[表示\]メニュー](#)の **[ツリーを展開]** をクリックする。
- [ステータステーブル](#)の左上隅にある **[階層]** をダブルクリックする。

注記: リソース階層に表示されるリソースタグ(またはID)は、優先度が最も低いサーバのもので、サーバを指定して、リソースタグやIDを表示するには、表内のリソースインスタンスのセルを左クリックします。指定したサーバのタグとIDがメッセージバーに表示されます。

リソース階層を縮小するには、以下のいずれかの操作を行います。

-  をクリックする。
-  の右側にあるリソースアイコンをダブルクリックする。

リソース階層をすべて縮小するには、以下のいずれかの操作を行います。

- [\[表示\]メニュー](#)の[ツリーを折りたたむ]をクリックする。
- [ステータステーブル](#)の左上隅にある[階層]をダブルクリックする。

オペレータタスク

このセクションでは、リソースサービスの起動と停止など、より高度なタスクについて説明します。オペレータタスクを実行するには、オペレータ権限が必要です。

リソースを起動す

リソースを起動するには、以下の操作を実行してください。

1. サービスを起動するサーバ固有のリソースインスタンスを選択し、[リソースコンテキストメニュー](#)または[リソースコンテキストツールバー](#)の[サービス開始]をクリックします。
2. サービスを開始するサーバとリソースを確認するダイアログボックスが表示されます。親リソースを起動せずに、依存する子リソースだけを起動すると、ダイアログボックスに警告メッセージが表示されます。[サービス開始]をクリックし、依存する子リソースとともに、リソースを起動します。
3. [アウトプットパネル](#)が有効な場合は、ダイアログボックスが閉じて、リソースを起動した結果がアウトプットパネルに表示されます。すべての結果が表示されたら、[完了]をクリックしてダイアログボックスを閉じます。
4. リソースを起動する際に発生したエラーは、LifeKeeper ログおよびリソースを起動したサーバのGUI ログに記録されます。

リソースを停止する

リソースを停止するには、以下の操作を実行してください。

1. サービスを停止するサーバ固有のリソースインスタンスを選択し、[リソースコンテキストメニュー](#)または[リソースコンテキストツールバー](#)の[サービス停止]をクリックします。
2. 停止するリソースを確認するダイアログボックスが表示されます。親リソースを停止せずに、依存する子リソースを停止すると、ダイアログボックスに警告メッセージが表示されます。[サービス停止]をクリックし、リソースを停止します。
3. [アウトプットパネル](#)が有効な場合は、ダイアログボックスが閉じて、リソースを停止した結果がアウトプットパネルに表示されます。アウトプットパネルが無効な場合は、ダイアログボックスに結果が

表示されます。すべての結果が表示されたら、**[完了]**をクリックしてダイアログボックスを閉じます。

4. リソースを停止する際に発生したエラーは、LifeKeeper ログ、およびリソースを停止するサーバの GUI ログに記録されます。

ボリュームリソースの起動と停止

ウイルススキャナやWindows サービスなど、バックグラウンドで動作するプロセスは、共有ボリュームに書き込み権限を要求します。こうしたプロセスはアプリケーションの実行に不可欠であり、長期的に停止させられません。

バックグラウンドで動作するアプリケーションは通常、SPS ボリュームの手動切り替えに影響しません。しかし、手動切り替え時に次のエラーメッセージが表示される場合、実際のフェイルオーバーには影響が出ませんが、後述するボリューム削除時の停止と再起動機能で修正します。

```
*ERROR* [No. 12035] Unable to lock volume <volume ID> on <system name> machine at this time as it may be in use by some application. Please free this volume and try again.
```

書き込み権限のないユーザがボリュームにアクセスし、リモートリンクやローカルオープンを実行している場合、ボリュームの削除は、他のシステムへのボリュームのリストアと同様に成功します。既存ユーザのオープン操作は、当然ながら、無効となります。ただし、元のシステムにボリュームをリストアする操作は、手動切り替えでも、自動的なフェイルオーバーでも失敗します。こうした操作を実行すると、次のエラーメッセージが表示されます。

```
*ERROR* [No. 12046] LifeKeeper RESTORE VOLUME <volume ID> FAILED(err=<error number>).
```

ローカルユーザ(またはリモートユーザ)がボリュームにアクセスすると、ボリュームの削除とリストア、バックアップサーバへのボリュームの切り替え、ボリュームリソースを含む階層の切り替えは失敗します。

また、システムのPATH変数には、SPSで保護するボリュームにファイル共有を指定できません。PATH変数にファイル共有を指定すると、ボリューム操作が失敗することがあります。ボリュームリソースを作成する前に、ファイル共有を削除してください。PATH変数を変更するには、**[コントロールパネル]**を開き、**[システム]-[環境]**を選択します。

ボリューム削除時のプログラムとサービスの停止と再起動

LifeKeeperでは、レジストリキーを編集すると、ボリューム削除時に停止して再起動するプログラムとサービスを指定できます。SPSは、ユーザがボリュームを削除してサービスを停止した場合、そのボリュームにオープンハンドルがあれば、指定されたプログラムとサービスを停止します。オープンハンドルがないと、プログラムとサービスを停止しません。ボリュームを削除するときに、プログラムとサービスを停止するには、次の手順に従います。

1. 削除するボリュームにアクセスし、フェイルオーバーをブロックしているプログラムとサービスを探します。
2. 以下の各プログラムのサブキーを追加し、停止して再起動するプログラムを指定します。

- 32ビット: HKEY_LOCAL_

`MACHINE\SOFTWARE\SteelEye\LifeKeeper\VolumeRemoveStopPrograms*`

- **64 ビット:** `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\SteelEye\LifeKeeper\VolumeRemoveStopPrograms*`
 例えば、「myapp.exe」というプログラムを停止するには、次のサブキーを追加します。
`HKEY_LOCAL_MACHINE\SOFTWARE\SteelEye\LifeKeeper\VolumeRemoveStopPrograms\myapp.exe\`

3. 各プログラムのサブキーに次の値 (REG_SZ) を追加します。

ProgramName	プログラム名のみ (注記: 手順 2 で指定したサブキー名を使用します。)
ProgramPath	プログラム名 (絶対パス)
Restart	プログラムを再起動しない場合は、0 を指定します。再起動する場合は、1 を指定します。
StartCmdLine	プログラム起動時に渡すコマンドラインオプションを指定します。
WasRunning	プログラム停止前に稼働していたインスタンス数を記憶します。0 に初期化します。

例えば、`myapp.exe /a /t /p` を停止して再起動する場合は、次のようにレジストリを編集します。

ProgramName	myapp.exe
ProgramPath	C:\mydir\myapp.exe
Restart	1
StartCmdLine	/a /t /p
WasRunning	0

注記: SPS はデフォルトで `perfmon.exe` を停止するサブキーをレジストリに書き込むため、再起動オプションは無効になります。

1. 以下の各サービスにサブキーを追加し、停止して再起動するサービスを指定します。

- **32 ビット:** `HKEY_LOCAL_MACHINE\SOFTWARE\SteelEye\LifeKeeper\VolumeRemoveStopServices\`
- **64 ビット:** `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\SteelEye\LifeKeeper\VolumeRemoveStopServices\`
 例えば、「mysvc」というサービスを停止するには、次のサブキーを追加します。
`HKEY_LOCAL_MACHINE\SOFTWARE\SteelEye\LifeKeeper\VolumeRemoveStopServices\mysvc\`

2. 各サービスのサブキーに次の値 (REG_SZ) を追加します。

ボリュームシャドウコピー (VSS)

DataKeeper/SPS ボリュームでボリュームシャドウコピー (VSS) を使用する

Windows 2003 および 2003 R2 では、SPS によって保護される (共有または複製) ボリュームで VSS シャドウコピーを有効にすることはできません。保護対象ボリュームのスナップショットを設定すると、スナップショットが異なるボリュームに保存される場合でも、SPS および DataKeeper はボリュームをロックできません。このため、ボリューム上のデータを保護することができません。

Windows 2008 および 2008 R2 では、SPS によって保護される (共有または複製) ボリュームで VSS シャドウコピーを有効にすることができます。ただし、以下のガイドラインが適用されます。

- SPS によって保護されるボリュームには VSS スナップショットイメージを保存できません。SPS によって保護されるボリュームに VSS スナップショットを保存すると、SPS はボリュームをロックして別のノードにスイッチオーバーすることができません。
- SPS によって保護されるボリュームがスイッチオーバーまたはフェイルオーバーした場合、SPS によって保護されるボリュームから取得された以前のスナップショットは破棄され、再利用することができません。
- VSS スナップショットスケジュールは SPS サーバ間でコピーされません。プライマリサーバで 1 日 2 回スナップショットを取得するように予定されているときにスイッチオーバーが発生した場合、このスケジュールはバックアップサーバには渡されず、バックアップサーバで再定義する必要があります。
- スナップショットが以前有効になっていたサーバに切り替えられた場合、動作にわずかな違いがあります。
- ボリュームが共有ボリュームである場合は、VSS スナップショットを再度有効にする必要があります。
- ボリュームが複製ボリュームである場合は、VSS スナップショットは自動的に有効になります。

共有 SCSI ボリュームのボリュームロック

共有 SCSI ディスク上のリソースを保護するには、**Windows のディスク管理** ツールを使用して、共有ディスクを論理ボリュームに分割します。ボリュームリソースインスタンスを定義すると、SPS で共有ボリュームを保護できます。インスタンスにはドライブレターを指定します (例: G:)。

SPS は、ボリュームリソースインスタンスをプライマリサーバで起動した後、ソフトウェアロックをかけます。その結果、ボリュームがプライマリサーバでアクティブな限り、バックアップサーバはボリュームにアクセスできません。プライマリサーバに障害が発生すると、SPS はボリュームリソースをバックアップサーバで起動し、修復中のプライマリサーバがボリュームリソースにアクセスできないようにロックします。

SPS は、プライマリとバックアップを入れ替えるため、障害が発生したサーバは、ボリュームリソースにアクセスできなくなります。そのため、障害が発生したサーバを修理している間は、不適切なアクセスからリソースを保護できます。

SPS のインテリジェントスイッチバック機能は、プライマリサーバとバックアップサーバを動的に再定義します。そのために、修理が完了したら、適当なタイミングを指定して、修復したサーバでリソースを起動できます。

ボリュームがロックされているときは、SPS を停止しないでください。SPS を停止するとロックが無効になります。

詳細項目

このセクションでは、LifeKeeper 構成データベースとLifeKeeper の通信について説明します。

LifeKeeper 構成データベース (LCD)

LifeKeeper 構成データベース (LCD) は、オブジェクト指向のリソース階層情報を保持しています。SPS が保護するリソースタイプのリカバリ方向に関する情報も格納しています。LCD データはシステムの共有メモリにキャッシュされてから、ファイルに格納されるため、システムを再起動しても構成データは保持されます。LCD にはリカバリに必要なリソースインスタンスに関する状態情報や詳細なデータも格納されています。

LCD のディレクトリ構造、格納するデータの種類、利用可能なリソースタイプ、アプリケーションスクリプトの使用方法などについては、次の項目を参照してください。

- [LCD のディレクトリ構造](#)
- [LCD ディレクトリの構造図 \(\\$lkroot/LifeKeeper\)](#)
- [LCD 構成データ](#)

LCD のディレクトリ構造

`$LKROOT` (デフォルトは `c:\LK`) には、次のサブディレクトリがあります。

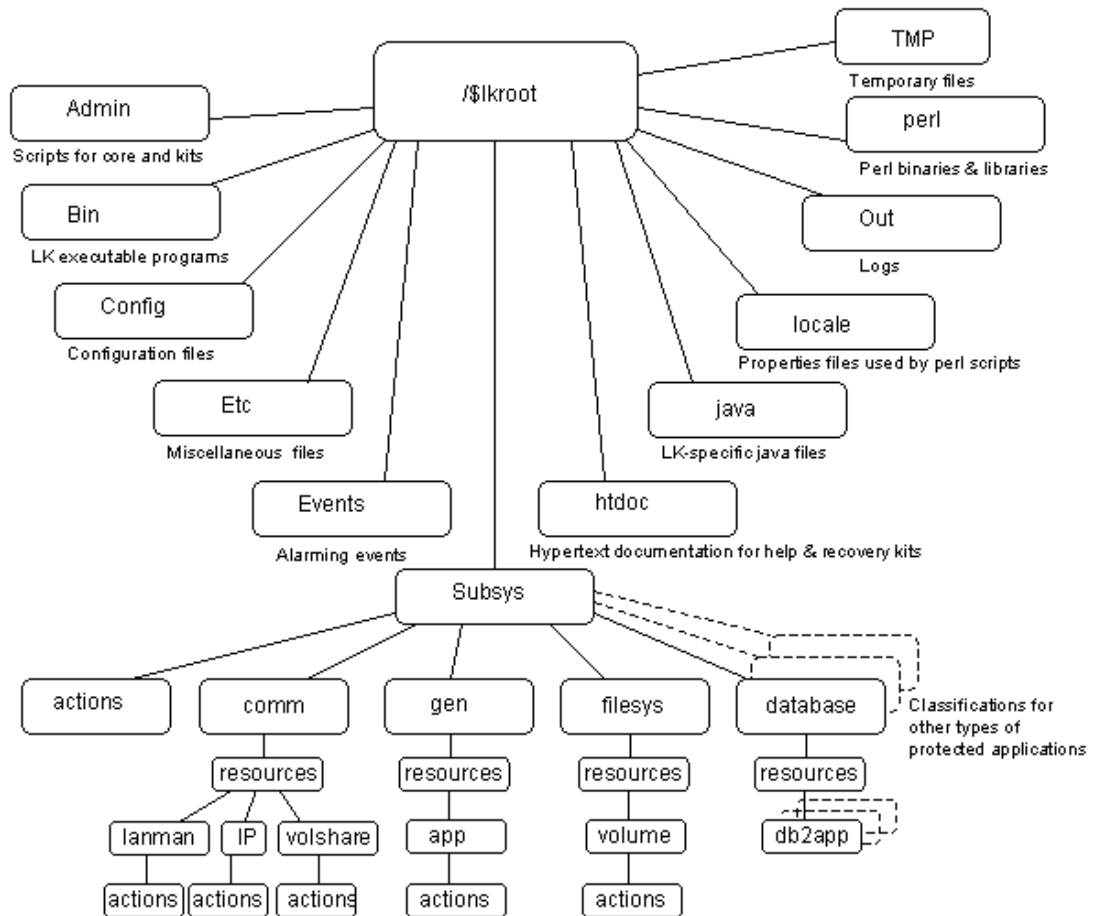
- **Admin:** SPS Core とリカバリキットのスクリプトが格納されています。
- **Config:** 共有イクイバレンシなどの SPS 構成ファイルが格納されています。
- **Bin:** SPS の実行ファイルが格納されています。
- **Subsys:** リソースとタイプに関する情報が格納されています。SPS は Subsys サブディレクトリの下にリソースとタイプの定義を格納します。例えば、通信リソースは `comm`、ボリュームリソースは `filesys`、Generic Application のリソースは `gen` にそれぞれ格納されます。オプションのリカバリキットは、固有のリソースタイプを別のディレクトリに作成します。例えば、データベースアプリケーションのリソースは `database` ディレクトリに格納されます。
- **Events:** イベントアラームが格納されています。
- **Out:** LifeKeeper のログが格納されています。SPS は、エラーとメッセージの種類に応じて、ログをディレクトリに送ります。
- **perl:** Perl の実行可能プログラム (バイナリ) とライブラリが格納されています。

`$LKROOT` 以下の LCD ディレクトリ構造については、[LCD ディレクトリの構造図](#)を参照してください。

注記: サブディレクトリの場所は、環境変数 `LKROOT` の値を編集すると変更できます。

LCD ディレクトリの構造図

\$lkroot のディレクトリ構造を次図に示します。



LCD 構成データ

LCD には次のデータが格納されています。

- 依存関係に関する情報
- リソースステータスに関する情報
- サーバ間のイクイバレンシに関する情報

依存関係に関する情報

SPS は、依存元と依存先のリストを定義済みリソースごとに保持しています。他のリソースに依存するリソースを依存元と呼びます。詳細については、マニュアルページの [LCDI 関係](#) と [LCDI インスタンス](#) を参照してください。

リソースステータスに関する情報

LCD にはリソースインスタンスのステータス情報が格納されています。LCD には、[リソースの状態](#) として、**ISP**、**OSF**、**OSU**、**ILLSTATE** が記録されています。リソースの状態は、システムイベントが発生するか、管理者が特定の作業を実行すると変化します。リソースの状態が変化すると、その情報がローカルサーバとバックアップサーバの LCD に反映されます。

サーバ間のイクイバレンシに関する情報

複数のサーバが保持するリソースには、一定の関係があります。[共有イクイバレンシ](#)とは、同じ物理エンティティである複数のサーバが保持する2つのリソースの関係を指します。2台のサーバが共有イクイバレンシ関係を持つリソースを保持する場合、SPS は一方のサーバが保持するリソースインスタンスだけを「サービス中 (ISP)」にして、スプリットブレインを回避します。2台のサーバのリソースインスタンスは、両方ともサービス休止状態 (OSU または OSF) にできますが、リソースをサービス中 (ISP) にする場合は、一方のサーバだけでリソースを起動して、データの完全性を保証します。

イクイバレンシリソースには、SCSI バスで接続したディスクなどがあります。

LifeKeeper は、階層内に依存関係を設定して、共有ファイルなど、ボリュームに依存する全リソースが一方のサーバだけで稼働することを保証します。

LCD リソースタイプ

LifeKeeper 構成 データベース (LCD) は、共有メモリと `$LKROOT` ディレクトリで保持されています。LifeKeeper 構成 データベース (LCD) の [ディレクトリ構造図](#) に示したように、`subsys` にはアプリケーションインターフェースを定義する次のアプリケーションリソースセットが格納されています。

- `filesys` - ボリュームなどのファイルシステム関連リソース
- `comm` - IP、共有ボリューム (共有ファイル)、LAN Manager などの通信関連リソース
- `database` - Oracle などのデータベースリソース

サブディレクトリの詳細については、[リソースサブディレクトリ](#) を参照してください。

リソースサブディレクトリ

`filesys`、`comm`、`WebServer`、`database`、`mail appsuite` の各ディレクトリには、`resources` サブディレクトリがあります。`resources` ディレクトリには、に定義して管理するリソースタイプのリストが格納されています。

- **ファイルシステムリソースタイプ:** `$LKROOT\LifeKeeper\subsys\filesystem\resources` ディレクトリに格納されています。
 - **ボリューム:** ディスクパーティションまたは仮想ディスクデバイスです。
- **通信リソースタイプ:** `/$LKROOT/LifeKeeper/subsys/comm/resources` ディレクトリに格納されています。
 - **IP:** IP リカバリキットで作成するリソースです。
 - **DNS:** DNS リカバリキットで作成するリソースです。
 - **volshare:** LAN Manager リカバリキットが作成するファイル共有リソースです。
 - **lanman:** LAN Manager リカバリキットが作成するコンピュータの別名です。
- **WebServer リソースタイプ:** `$LKROOT\LifeKeeper\subsys\WebServer\resources` ディレクトリに格納されています。
 - **IIS:** IIS リカバリキットで作成するリソースです。
- **データベースリソースタイプ:** `$LKROOT\LifeKeeper\subsys\database\resources` ディレクトリに格納されています。
 - **Oracle**
 - **Microsoft SQL Server**
- **メールリソースタイプ:** `$LKROOT\LifeKeeper\subsys\mail\resources` ディレクトリに格納されています。

各リソースタイプディレクトリには、次のサブディレクトリがあります。

- **instances:** LifeKeeper 構成データベース (LCD) に格納されたリソースインスタンスに関する恒久的な情報を保存するファイルです。このリソースタイプに関連するリソースインスタンスの説明が記述されています。
- **actions:** 特定のリソースタイプのリソースインスタンスで動作する一連のリカバリアクションプログラムが格納されています。アプリケーションのすべてのリソースタイプに適用するアクションは、リソースタイプディレクトリではなく、アプリケーションディレクトリの `actions` サブディレクトリに配置します。リカバリディレクションソフトウェアは、リソースインスタンスを変更または修復します。

リソースタイプの `actions` ディレクトリには、**remove** と **restore** スクリプトをセットで格納する必要があります。

リソース操作

リソースタイプの `actions` サブディレクトリには、特定のアプリケーション機能を記述するプログラム (通常はシェルスクリプト) があります。リソースタイプには `restore` と `remove` スクリプトが必要です。

`remove` と `restore` スクリプトは、互いに逆の機能を実行し、一方が他方の処理を取り消します。スクリプトは手動で実行しないでください。スクリプトは、LifeKeeper リカバリ制御インターフェース (LRACI) の `perform_action` シェルプログラムから実行します。詳細については、[LRACI-perform action](#) マニュアルページを参照してください。

LCDI コマンド

SPS では、次の2種類の方法でアプリケーションのリソース階層を定義できます。

- LifeKeeper GUI
- LifeKeeper 構成 データベースインターフェース (LCDI) のコマンド

LCDI は、SPS が提供する一連のインターフェースコマンドです。LCDI を使用すると、ユーザのアプリケーション要件に従って、リソース階層を作成してカスタマイズできます。例えば、アプリケーションが複数のリソース(ファイルシステムなど)に依存する場合は、コマンドインターフェースを使用します。

コマンドの詳細については、[LCDI のマニュアルページ](#)を参照してください。ここでは、GUI とコマンドの両方を使用して、リソース階層を作成する開発シナリオについて説明します。

LifeKeeper 通信 マネージャ (LCM)

LifeKeeper 通信 マネージャ (LCM) は、クラスタサーバに高信頼性のプロセス間通信を提供します。コミュニケーションパスを冗長化すると、1本のコミュニケーションパスに障害が発生しても、SPS と保護対象リソースは停止しません。LCM は、RS-232 (TTY)、TCP/IP、共有ディスク接続をサポートしています。

LCM は次の機能を提供します。

- **SPS ハートビート:** クラスタサーバに定期的に信号を送信し、他のシステムの死活状態を確認します。SPS は、ハートビート信号が停止すると、システム全体に障害が発生したと解釈します。この機能は他の手段では実現できません。
- **管理 サービス:** SPS は、LCM を使用して、クラスタサーバをリモートから管理します。LCM を使用すると、クラスタサーバの一元管理、クラスタ構成の検証、管理アクションの健全性検査を実行できます。
- **クラスタ構成とステータス通信:** LifeKeeper 構成 データベース (LCD) は、LCM を介して、リソースのステータス、可用性、クラスタ構成を追跡し、LCD におけるプライマリサーバとバックアップサーバ間でリソース情報の整合性を保証します。
- **フェイルオーバーカバリ:** システム上のリソースに障害が発生すると、LCM は、リソースをバックアップシステムで起動するように SPS に通知します。

通信ステータス情報

[\[サーバプロパティ\] ダイアログボックス](#)の **[通信]** タブには、SPS に登録したサーバの一覧、カレントステータス、コミュニケーションパスに関する情報が表示されます。

データレプリケーション

このセクションでは、SPS とともにデータレプリケーションを使用する方法について説明します。

レプリケーションボリュームリソースの監視






SPS で保護するレプリケーションボリュームリソースの状態は、LifeKeeper GUI に表示されます。ミラー状態の詳細については、SteelEye DataKeeper のトピック[ミラー状態の定義](#)を参照してください。

次図の例では、レプリケーションボリュームリソースのミラー状態は、Vol.L が再同期、Vol.Y がミラーリングになっています。

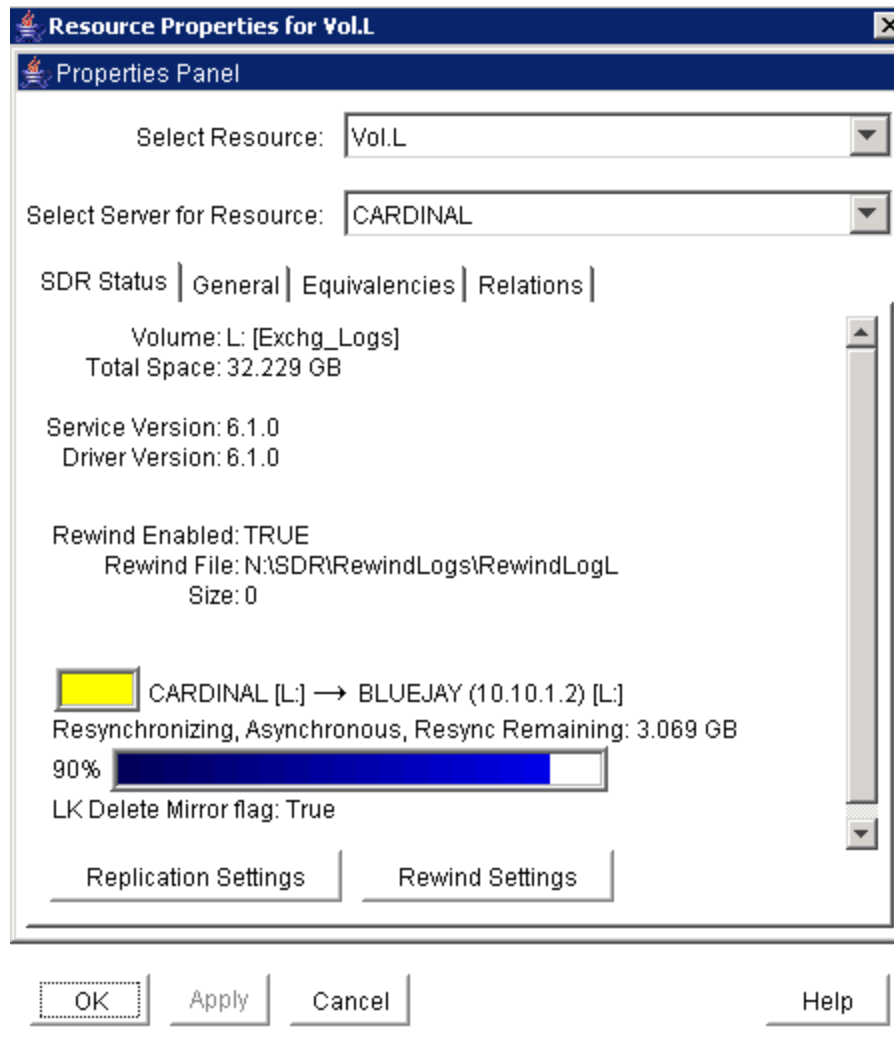
Hierarchies		CARDINAL		BLUEJAY	
Active Protected					
MSEch.0		Active 1	StandBy 10		
172.17.108.130		Active 1	StandBy 10		
Vol.L		In Service 1	Resync 10		
Vol.Y		In Service 1	Mirroring 10		

レプリケーションボリュームリソースの状態と意味を次表に示します。

リソースの状態	アイコン	表示状態
アクティブ		プライマリサーバで稼働しており、保護されています (ISP)。
縮退		プライマリサーバで稼働していますが、バックアップサーバでは保護されていません (ISU)。
不明		リソースが初期化されていないか (ILLSTATE)、LifeKeeper が稼働していません。
障害		障害が発生しています。リソースの稼働に失敗した場合 (OSF)。
オフライン		リソースが稼働していません。ボリュームの読み書き操作はできません。

リソースの状態	アイコン	表示状態
再同期 (ペンディング)		バックアップサーバ上のリソースステータスは、 再同期 (ペンディング) です。
ミラーリング		バックアップサーバ上のリソースステータスは、 ミラーリング です。
一時停止		バックアップサーバ上のリソースステータスは、 一時停止 です。
再同期		バックアップサーバ上のリソースステータスは、 再同期 です。
中断		バックアップサーバ上のリソースステータスは、 中断 です。

レプリケーションボリュームリソースの設定状態を LifeKeeper GUI に表示するには、ボリュームリソースを右クリックし、**[プロパティ]**をクリックし、**[SDR ステータス]**タブをクリックしてください。次図の例では、Vol.E がLK031 上のソースであり、ターゲットサーバ(192.168.0.2) が1 台あり、ターゲットサーバLK032 で再同期が実行されています。



レプリケーションの設定

[ボリュームリソースのプロパティ] ページを開き、[レプリケーション設定] をクリックすると、複製ボリュームの圧縮レベル、ネットワーク帯域制限、SPS 削除ミラーフラグを設定できます。








フィールド	説明
ターゲットの選択	操作を実行するターゲットサーバを選択します。

フィールド	説明
圧縮レベルの設定	<p>レプリケーションボリュームの圧縮レベルを指定します。</p> <p>有効値は0～9です。レベル0は「圧縮なし」です。1から9の値で圧縮のCPUの集中レベルを増分的に指定します。圧縮レベル1は高速圧縮です。データを圧縮するCPU時間は最短ですが、パケットサイズが最大になります。レベル9は圧縮時間が最長ですが、圧縮率は最大になります。つまり、CPU時間は長くなりますが、パケットサイズは最小になります。ご使用のシステム、ネットワーク、作業負荷を考慮して、CPU使用率とネットワーク転送速度のバランスを取れるように、1～9までの値を選択してください。</p> <p>デフォルト: 0</p>
ネットワーク帯域制限の設定	<p>ネットワーク帯域設定では、レプリケーションボリュームの再同期と通常のボリューム書き込みを使用するネットワーク帯域をKB単位で指定します。</p> <p>デフォルト: 0</p>
SPS デリートミラーフラグの設定	<p>SPS デリートミラーフラグは、レプリケーションボリュームのSPS リソースを削除する動作を制御します。このフラグを [True] に設定した場合、SPS ボリュームリソースを削除すると、ミラーが削除されます。[True] を指定しないと、ミラーは削除されません。</p> <p>ボリュームリソースの拡張を解除するか、SPS から削除する場合は、[True] を選択します。</p> <p>ミラーを削除しない場合は、[False] を選択します。</p> <p>LifeKeeper GUI でミラーを作成している場合は、[True] が既定値となります。 LifeKeeper GUI を使用せずにミラーを作成している場合、既定値は [False] になります。</p>

レプリケーションボリュームの操作

LifeKeeper GUI を使用して、レプリケーションボリュームリソースを操作するには、レプリケーションボリュームリソースを右クリックし、実行する操作をコンテキストメニューから選択してください。[表示]メニューの[プロパティパネル]をクリックして、**プロパティパネル**を有効にした場合、ボリュームを選択すると、リソースツールバーが表示されます。

操作	アイコン	説明
ターゲットの選択		操作を実行するターゲットシステムを選択します。

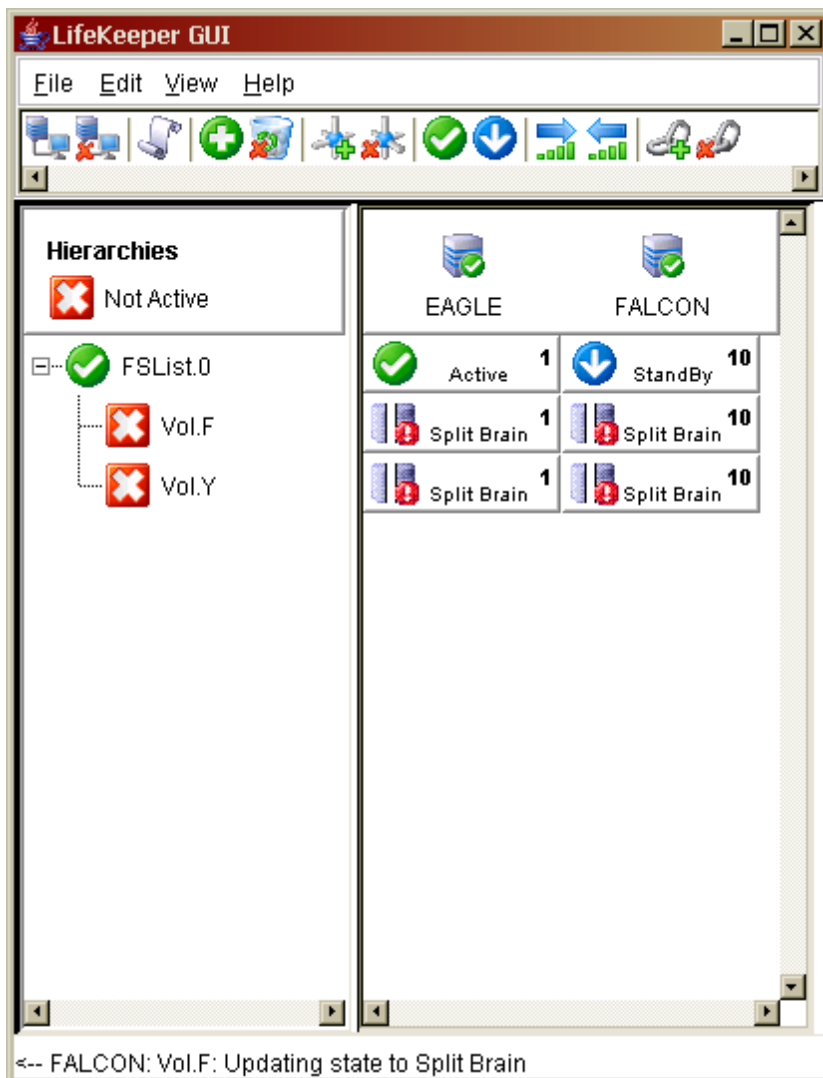
操作	アイコン	説明
ミラー一時停止		[ミラー一時停止] をクリックすると、データのミラー化が一時停止します。 [続行] をクリックしてミラーリングを再開すると、部分再同期が実行されます。ミラーリングを一時停止した後、 [ターゲットのロック解除] をクリックすると、ターゲットボリュームのロックを解除できます。
ミラー再開/ターゲットロック		ミラーリングを一時停止した後に再開するには、 [続行/ターゲットのロック] をクリックします。このボタンをクリックすると、一時停止が解除されます。ターゲットボリュームがロックされている場合、ロックが解除してミラーリングが再開されます。 ミラーリングを一時停止した場合は、その間の書き込みデータが記録されます。ミラーリングを再開すると、記録されたデータがターゲットボリュームに送信され、部分再同期が自動的に実行されます。
ターゲットのロック解除		ミラーのターゲットボリュームのロックを解除するには、 [ターゲットのロック解除] をクリックします。このボタンをクリックした場合、ミラーリングが一時停止して、ターゲットシステム上のミラーボリュームのロックが解除されます。その結果、ボリューム上のデータを読み書きできるようになります。 [続行] をクリックすると、ターゲットボリュームが再びロックされます。部分再同期の実行後にミラーリングが再開されます。 警告: ミラーリングのロック解除中には、ターゲットボリュームにデータを書き込まないでください。ロック解除時に書き込んだデータは、ミラーの再同期時に失われます。
ミラー中断/ターゲットロック解除		ミラーリングを中断すると、選択したボリュームのミラーリングが中断され、ターゲットボリュームのロックが解除されます。ただし、ミラーはボリュームリストから削除されません。 [ミラー中断/ターゲットロック解除] を選択した場合は、その後で完全な再同期を実行して、ミラーリングを再確立する必要があります。 警告: ミラーリングの中断中には、ターゲットボリュームにデータを書き込まないでください。中断時に書き込んだデータは、ミラーの再同期時に失われます。
ミラー再同期/ターゲットロック		中断したミラーリングを再確立するには、 [ミラー再同期/ターゲットロック] をクリックします。完全な再同期が実行されます。
ターゲットボリュームリワインド		このウィザードは、ターゲット上のデータを良好なデータセットまで戻し、リカバリプロセスを続行します。
リワインドログブックマーク追加		ブックマークを表示して、リワインドログにブックマークを追加できます。ブックマークは、リワインド操作に備えて、アップグレードなどの重要なイベントを追跡する際に使用します。リワインドを実行すると、データリワインドポイントとして、ブックマークのログが表示されます。

スプリットブレインとは

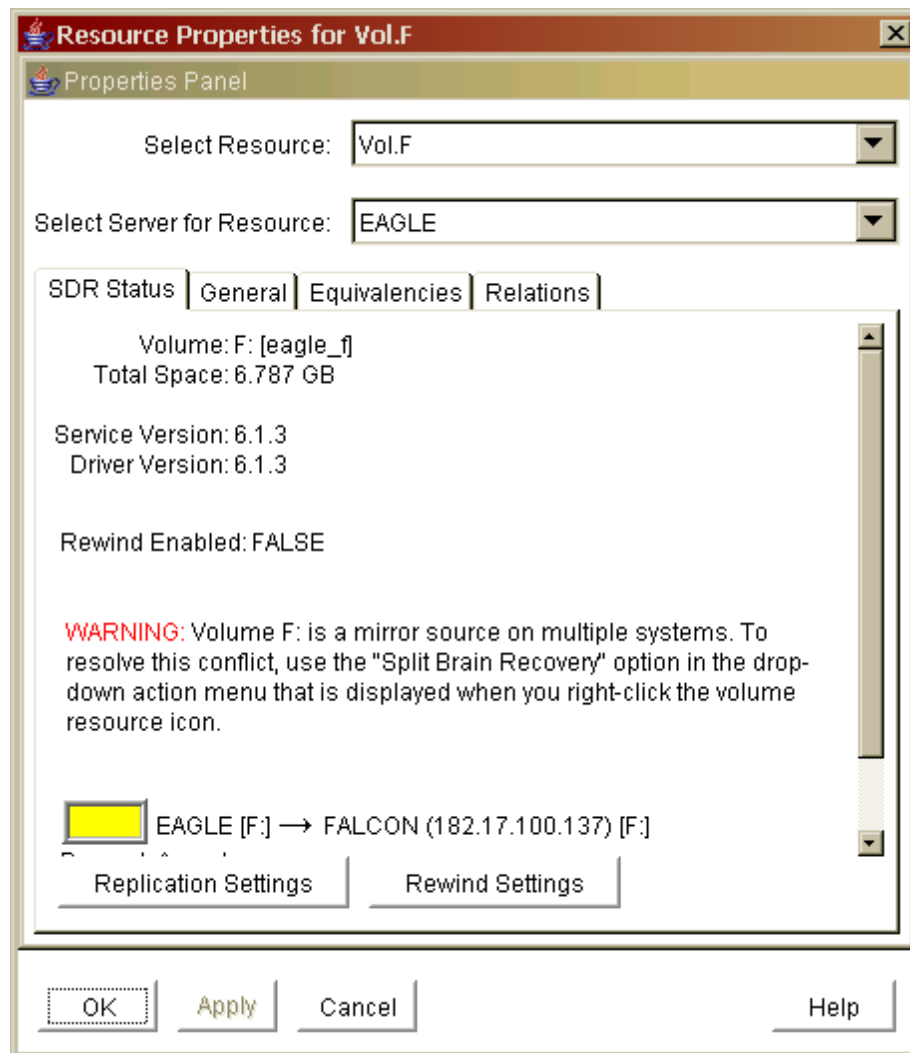
SPS のすべてのコミュニケーションパスが切断された場合、**自動ノードフェイルオーバー**が有効になっていると、SPS のそれぞれの側がもう一方を停止したものと見なし、すべてのリソースを起動しようとする。SteelEye DataKeeper リソースの場合、両側がミラーソースになり、ボリュームにデータを書き込めるようになります。この状態は「スプリットブレイン状態」と定義され、LifeKeeper GUI では以下のアイコンが表示されます。



この状態を解決するために必要な手順については、[スプリットブレインリカバリ](#)を参照してください。



選択したボリュームのプロパティパネルに、スプリットブレイン状態に関する追加情報と、この問題を解決するための手順が表示されます。



スプリットブレインリカバリ

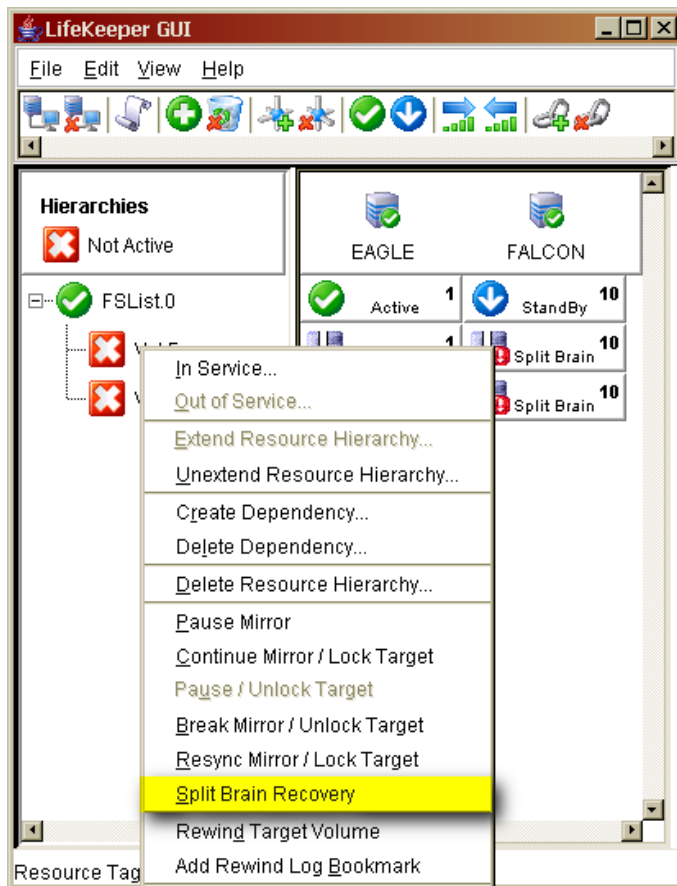
システムのコミュニケーションパスが復旧した後にボリュームがスプリットブレイン状態にあることが検出された場合、以下のスプリットブレインリカバリ手順を実行する必要があります。

注記: 複数のボリュームが異なるリソース階層で検出された場合は、それぞれのボリュームでスプリットブレインリカバリ手順を実行する必要があります。スプリットブレイン状態のボリュームが同じ階層にある場合は、リカバリが同時に実行されます。

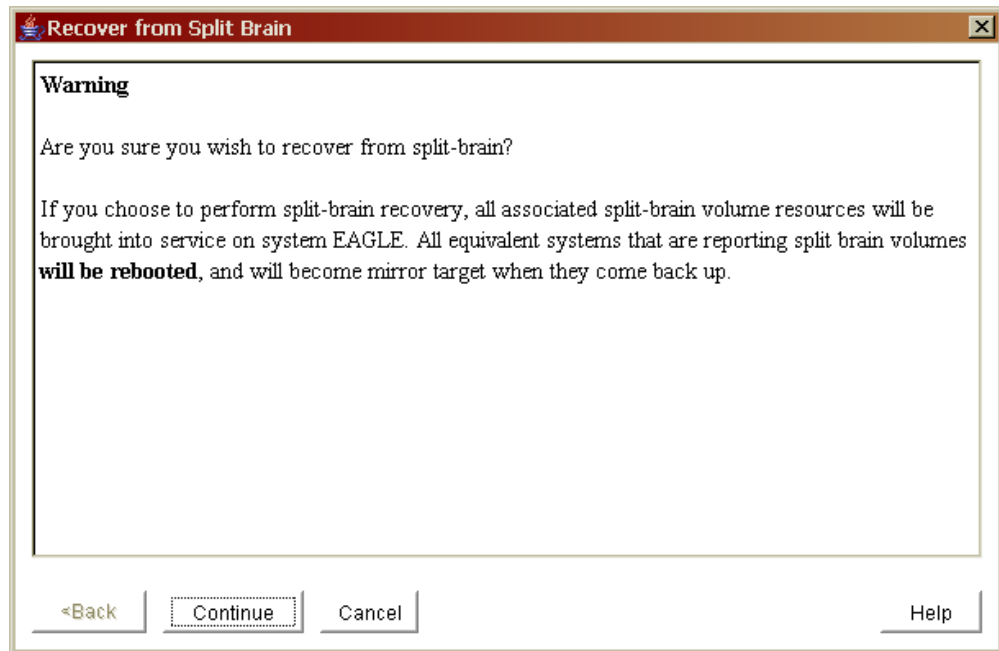
1. ソースにするシステムの下 のボリュームインスタンスアイコンを右クリックします。リソースコンテキストメニューが表示されます。一番左のパネルのリソース階層リストのボリュームインスタンスアイコンを右クリックすることもできます。メニューから [スプリットブレインリカバリ] を選択すると、ミラーソースと

するサーバを選択するよう求められます。

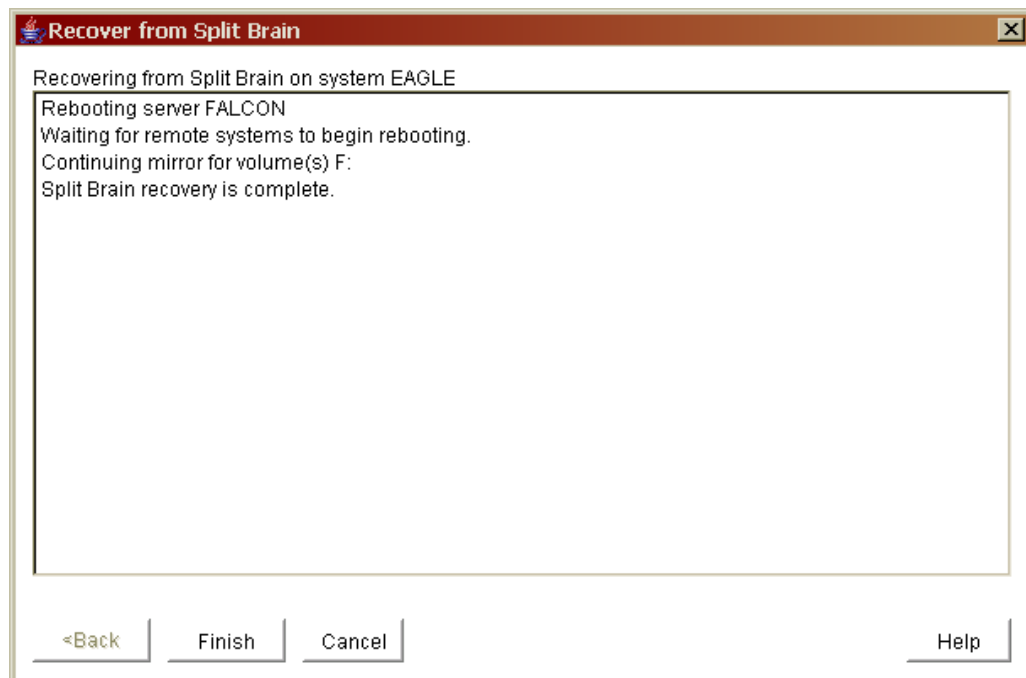
2. メニューから [スプリットブレインリカバリ] を選択します。



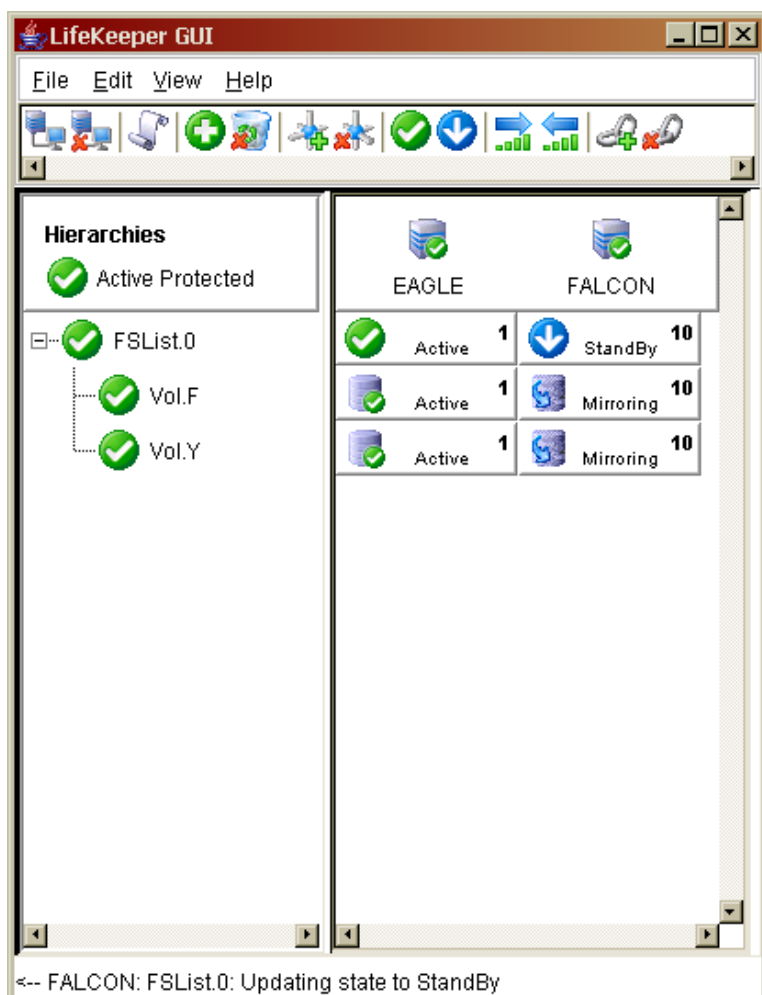
3. 以下の警告メッセージが表示されます。[続行] ボタンをクリックすると、スプリットブレインリカバリプロセスが完了します。注記: この手順では、スプリットブレイン状態にある他のすべてのシステムが再起動され、再起動プロセスが終了したときにミラーターゲットになります。



4. スプリットブレインリカバリプロセスの進行に合わせて、以下のメッセージが表示されます。リモートシステムが再起動している間に遅延が発生します。**[完了]**をクリックします。



5. リカバリが完了すると、復旧したリソースは GUI に以下のように表示されます。ターゲットシステムが再起動し、ミラーが再同期すると、スプリットブレインリカバリプロセスは完了します。



データリワインドの概要

データリワインド機能は、任意のデータレプリケーションポイントまで戻るか進めて、レプリケーションボリュームのデータを変更します。この機能は、ボリューム上のデータ変更を記録するリワインドログを使用します。リワインドログにタイムスタンプが記録されていれば、指定した日時までボリュームを戻せます。また、ポイントをさらに戻すことも、進めることもできます。

リワインドログは、レプリケーション先のターゲットシステムに作成されて保持されます。レプリケーション元（プライマリサーバ）への影響を最小限に抑えるために、リワインド操作はターゲットサーバに対して実行されます。すべてのレプリケーションボリュームでは、リワインド機能がデフォルトで無効に設定されています。レプリケーションボリュームリソースを選択して、リワインド機能を有効にするには、**[ボリュームリソースのプロパティ]** ページを開き、**[リワインド設定]** をクリックします。

最適なパフォーマンスを得るには、リワインドログファイルを別の物理ディスクに作成します。ログとレプリケーションデータを配置する物理ディスクは、**[キャッシュへの書き込み]** を有効にすることをお勧めします。**[キャッシュへの書き込み]** を有効にするには、**[マイコンピュータ]** をクリックし、**[ボリューム]** を右クリックし、**[プロパティ]** をクリックします。**[ボリュームのプロパティ]** ダイアログボックスが開いたら、**[ハードウェア]** タブ

の[プロパティ]をクリックします。次に、[ポリシー] タブをクリックし、[ディスクの書き込みキャッシュを有効にする]をオンにします。

リワインド ログファイルをデフォルトと異なる場所に置く場合は、レプリケーションボリュームリソースのリワインド機能を設定する前に、ファイルロケーションを変更します。[ボリュームリソースのプロパティ] ページを開き、[リワインド設定] をクリックし、選択したレプリケーションボリュームリソースのリワインドログの場所を変更します。

リワインドログのディスク容量を節約するには、リワインドログファイルのNTFS 圧縮機能を有効にします。NTFS 圧縮機能は、選択したレプリケーションボリュームリソースのリワインド機能を有効にする前に設定する必要があります。リワインドログファイルのNTFS 圧縮を有効にするには、[ボリュームリソースのプロパティ] ページの [リワインド設定] をクリックします。

リワインド操作を実行し、適切なデータを見つけた後に使用できるリカバリオプションが1つあります。

- ターゲット (バックアップ) サーバからソース (プライマリ) サーバにデータを手動でコピーします。コピーが完了したら、ターゲットとの部分再同期を実行します。


リワインドの設定

[ボリュームリソースのプロパティ] ページを開き、[リワインドの設定] をクリックすると、リワインド機能の有効/無効の設定、リワインドログのロケーション変更、同時にリワインドするボリューム (単一のエンティティ) の指定、ログファイルの有効期限の設定、選択したレプリケーションボリュームリワインドのNTFS 圧縮の使用を設定できます。

フィールド	説明
リワインド機能の有効/無効の設定	<p>選択したボリュームリソースのリワインド機能を有効/無効に設定します。</p> <p>リワインド機能はデフォルトで無効に設定されています。</p>
リワインドログファイルのロケーション	<p>選択したボリュームのリワインドログファイルを格納するフォルダを指定します。ミラーボリューム、またはで保護するボリューム上のフォルダは指定できません。ロケーションには、書き込み可能なボリュームを選択します。</p> <p>デフォルト: %extmirrbase%\RewindLogs</p> <p>注記: ログファイルのロケーションは、リワインド機能を有効にする前に変更してください。</p>
ログファイルの圧縮	<p>NTFS 圧縮を使用して、リワインドログファイルを圧縮します。</p> <p>ログファイルを圧縮する場合は、[True] を選択します。</p> <p>デフォルト: False</p> <p>注記: ログファイルの圧縮オプションは、リワインド機能を有効にする前に設定してください。</p>

フィールド	説明
関連ボリューム	<p>同時にリwindするボリュームを単一エンティティとして指定します。関連ボリュームは、コンマ(,)で区切って指定します。例えば、2つのアプリケーションを使用しており、一方はD:とE: ボリューム、他方はP:とQ: ボリュームを使用している場合、「DE,PQ」と記述します。</p> <p>注記: このオプションは通常、変更する必要はありません。アプリケーションが使用するボリュームリソースは、SPS 階層に定義されているため、リwindを実行すると、関連ボリュームも自動的にリwindの対象となります。リソース階層を検索しても、自動的に検出できない特殊なボリュームを使用する場合は、デフォルトの動作を変更します。</p>
ログファイルの最大サイズ	<p>ログファイルの最大サイズをMB単位で指定します。0を指定すると、ログファイルは無制限となります。</p> <p>デフォルト: 0</p>
ログファイルの有効期限	<p>リwindログの有効期限を分単位で指定します。0を指定すると、無制限となります。</p> <p>デフォルト: 0</p>
ログファイルの最小空き容量	<p>空き容量の最小サイズをMB単位で指定します。ここで指定する値は、リwindログファイルを格納するボリュームが満杯になり、空き容量が完全に消費される前の予備の容量となります。</p> <p>デフォルト: 100 MB</p>

レプリケーションボリュームリソースのリwind操作

LifeKeeper GUI を使用してレプリケーションボリュームリソースをリwindするには、レプリケーションボリュームリソースを右クリックし、**コンテキストメニュー**の**[ターゲットボリュームリwind]**をクリックしてください。**[表示]**メニューの**[プロパティパネル]**をクリックして、**プロパティパネル**を表示した場合は、リwindするレプリケーションボリュームリソースのターゲットを選択し、リソースツールバーのアイコンをクリックしてください。リwind操作が始まります。

警告メッセージが表示され、リwind可能なボリュームが表示されます。ターゲットボリュームは一時停止して、ロックが解除されます。ミラーリングが一時停止している場合、ターゲットサーバはボリュームを回復できません。データをターゲットサーバにリwindしている間は、変更内容が追跡されます。**[ミラーリングの継続]**を実行すると、変更内容がターゲットに書き込まれます。**[継続]**をクリックしてリwind操作を継続するか、**[キャンセル]**をクリックしてリwindを終了します。リwind操作で指定できるオプションを次表に示します。

フィールド	説明
ボリューム <driveletter(s)> を使用している <Source> 上の アプリケーション を停止します か?	SPS 階層を構成する <driveletter(s)> ボリュームを使用しているプライマリサーバ <Source> で稼働している全アプリケーションを停止する場合は、 [はい] を選択します。プライマリサーバ<Source> 上のアプリケーションを停止しない場合は、 [いいえ] を選択します。 デフォルト: [いいえ]
<target> の <volume tag> にデータを復元 する準備をしま すか?	<volume tag> に指定したボリュームの準備が完了したら、 [次へ] をクリックしてリwind操作を続けます。
データリwind ポイントの指定	データリwindポイントを一覧から選択するか、開始時間を入力します。データリwindポイントには、次の日付形式を指定できます。 5/4/07 08:27 April 29 23:00 last Saturday 13:21 yesterday 08:23:17 入力した文字列 (タイムスタンプ) は、事前に検証されます。ダイアログボックスの下にあるプログレスバーにリwindログのデータの識別状態が表示されます。リwindプロセスを開始した時点では、すべてのデータが「不明」と表示されます。リwind処理が進み、データレプリケーションポイントが前に移動すると、各種データセットによるテスト結果に基づき、データセクションに「不良」または「良好」マークが付きます。
リwind実行	ターゲットのデータが前記の手順で指定したデータレプリケーションポイントに戻ります。 [次へ] をクリックし、リwind操作を続けます。
テスト結果に関 するコメントを入 力してください。 コメントは省略 できます。	テスト結果を評価して、コメントを記述します。コメントは保存されます。保存されたコメントは、 データレプリケーションポイント の一覧に表示されます。
データは有効で すか?	ボリューム上のデータが有効なことを指示します。ここで入力した回答は、リwindを実行する際、タイムスタンプ一覧の作成に使用されます。SPS は、ユーザの回答に基づき、タイムスタンプの開始時刻と終了時刻を絞り込み、直近の良好なデータを示します。 [はい] を選択した場合、選択したタイムスタンプより前のポイントには戻れません。 [いいえ] を選択した場合、選択したタイムスタンプより先のポイントには進めません。データの有効性が不明な場合は、 [不明] を選択し、開始時刻と終了時刻を選択しません。
リwind処理 の結果は次のと おりです。	評価とコメントを保存し、データレプリケーションポイントの間隔を調整します。 [次へ] をクリックし、リwind操作を続けます。

フィールド	説明
リカバリを続けるか、別のデータセットを選択してください。 [次の操作] をクリックします。	別のポイントに移動するか、リwind操作を停止して、現在の状態でボリュームをリカバリします。ダイアログボックスの下にあるプログレスバーにリwindログのデータの識別状態が表示されます。リwindプロセスを開始した時点では、すべてのデータが「不明」と表示されます。リwind処理が進み、データレプリケーションポイントが前に移動すると、各種データセットによるテスト結果に基づき、データセクションに「不良」または「良好」マークが付きます。
データリカバリ	失われたか破損したデータを復旧する場合、ソース(プライマリ)サーバで実行されているアプリケーションをそのままにすることも、必要に応じて停止することもできます。見つからないか破損したデータを、リwind済みターゲット(バックアップ)サーバからソース(プライマリ)サーバに手動でコピーします。コピーが完了したら、 [次へ] をクリックします。リwind済みボリュームが再びロックされ、ミラーリングが続行されます。これによって、リwind操作およびデータリカバリ操作が完了します。

Chapter 6: DataKeeper

SteelEye DataKeeper for Windows に関する情報は、SIOS テクニカルドキュメンテーションサイトの SPS for Windows オンラインドキュメンテーション内にある DataKeeper for Windows テクニカルドキュメンテーションを参照してください。:

- <http://docs.us.sios.com/#SPS4W>

Chapter 7: トラブルシューティング

このセクションでは、既知の問題および制限事項に関する重要な情報とそれらに対するワークアラウンドおよびまた解決方法について説明します。

アプレットのトラブルシューティング

説明

Web クライアントに [クラスタへの接続] ダイアログボックスが表示されない場合は、以下を実行してください。

1. アプレットが起動に失敗したか確認してください。通常は、ブラウザのウィンドウ内にアプレットの状態を示すメッセージが表示されます。Netscape と Internet Explorer では、状態を表すメッセージのほか、アプレットの代わりにアイコンが表示されることがあります。このアイコンをクリックすると、失敗に関する説明が表示されます。
2. Java コンソールを開いてください。
 - Firefox、Netscape、旧バージョンの Internet Explorer を使用している場合は、**コントロールパネル**を開き、**[Java プラグイン]** アイコンをクリックしてください。コンソール表示を有効にした後、Web ブラウザを再起動してください。
 - 新しいバージョンの Internet Explorer を使用している場合は、**[ツール]** メニューの **[Sun の Java コンソール]** をオンにしてください。[Sun の Java コンソール] メニューが表示されない場合は、**[ツール]** メニューの **[アドオンの管理]** をクリックし、コンソール表示を有効にしてから、ブラウザを再起動すると、Java コンソールが表示されます。
 - Mozilla を使用している場合は、**[ツール]** メニューの **[Web 開発]** をクリックし、**[Sun の Java コンソール]** をクリックしてください。
3. Web クライアントが開かない場合は、URL (`http://<server name>:81`) を指定して、ブラウザを再起動してください。
4. コンソールのメッセージを確認してください。エラーメッセージが表示された場合は、それを参考にして問題を解決してください。問題がネットワークに関係している場合は、「ネットワーク関連のトラブルシューティング」を参照してください。

LifeKeeper を停止するときに Win2008 で CYGWIN がクラッシュする場合がある

現象

解決策

LifeKeeper を停止するときに、Win2008 で CYGWIN がクラッシュする場合があります (例: GREP セグメンテーションの失敗)。

アプリケーションに対して Terminal Service および Data Execution Prevention が有効になっている場合、Win2008 では tsappcmp.dll に不具合があるため、クラッシュが発生することがあります。

解決策

tsappcmp.dll によって保護レベルが変更されるメモリページがあり、何らかの理由でそのページの「実行」権限が解除されます。このページには実行可能コードが含まれており、実行可能にしておく必要がありますが、権限が変更されたため、例外が発生してプログラムがクラッシュします。

詳細および回避策については、以下の記事を参照してください。

<http://www.eggheadcafe.com/software/aspnet/33380656/corinnayou-wrote-that-th.aspx>
<http://www.mail-archive.com/cygwin@cygwin.com/msg91569.html>

コマンドプロンプトから SPS コマンドを実行しようとしたときにエラーが発生する

現象

コマンドプロンプトから SPS コマンドを実行しようとしたときに、以下のエラーが発生します。

```
[File:lock.CLine:1610] Win32 Error:2
*CRITICAL* (No. 472) Can't run this application without LCD Daemon
running.
```

解決策

SPS コマンドを実行するには、「コンソール」権限が必要です。Server 2003 で SPS Core を実行しているときにリモートデスクトップを使用している場合は、「/console」スイッチを付けてリモートデスクトップを起動してください。Server 2008 で SPS Core を実行している場合は、「/admin」スイッチを付けてリモートデスクトップ接続を起動してください。

例: %SystemRoot%\system32\mstsc.exe/console

SPS システム自体のコマンドプロンプトから SPS コマンドを実行することもできます。

ファイアウォール

現象

インストール時にファイアウォールを無効にしましたが、今は有効にしています。Windows ファイアウォールの規則は、どのように追加するのですか。

解決策

LifeKeeper ディレクトリにはファイアウォールの規則を有効にすることができるスクリプトがあります。このスクリプトは次の場所にあります。

```
<LifeKeeper Root Directory>\support\firewallsetup.bat
```

コマンドプロンプトを開いて `firewallsetup.bat <LifeKeeper Root directory>` を実行すると、規則を追加できます。規則が追加されていた場合、このスクリプトによって重複した規則が追加されることはありません。

Windows ファイアウォール (`wf.msc`) を開くと、プレフィックスに LifeKeeper ラベルが付いた受信の規則が表示されます。

注記: SPS に必要なポートを無効にする規則を作成していた場合、インストールプログラムによってその規則が無効にされますが、規則が削除されることはありません。

GUI エラーメッセージ

説明

エラー 101: 不正な引数が渡されました。

エラー 102: このプログラムを正しく実行するには、バージョン 1.5 以上の Java 仮想マシンが必要です。[LifeKeeper GUI](#) のドキュメントを参照し、ブラウザの設定を確認してください。

エラー 103: LifeKeeper GUI のロックアンドフィールを設定できませんでした。

エラー 104: <filename> イメージをロードできませんでした。

エラー 106: RMI 経由でデータを取得しようとしたときにエラーが発生しました。処理を完了できませんでした。

エラー 107: グローバルリソースインスタンスの作成に失敗しました。

エラー 108: グローバルリソースの作成に失敗しました。

エラー 109: ダイアログでサーバを選択する必要があります。

エラー 112: リソースインスタンスをグローバルイクイバレンスに一致させられませんでした。

エラー 114: <server name> セキュリティ例外が発生したため、このサーバへの接続に失敗しました。そのため、クラスタリソースのレンダリングが縮退することがあります。Java ポリシーファイルが正しくインストールされていることを確認してください。[SPS Web クライアントの実行](#)を参照してください。

エラー 115: <server name> このサーバの名前を解決できなかったため、接続に失敗しました。そのため、クラスタリソースのレンダリングが縮退することがあります。[LifeKeeper GUI](#) のドキュメントを参照し、ネットワーク命名規則を確認してください。[不明なホスト例外](#)を参照してください。

エラー 116: <server name> このサーバがこのクライアントホストの名前を解決できなかったため、接続に失敗しました。そのため、クラスタリソースのレンダリングが縮退することがあります。[LifeKeeper GUI](#) のドキュメントを参照し、ネットワーク命名規則を確認してください。[不明なホスト例外](#)を参照してください。

GUI ネットワーク関連 - サーバへの初期接続の失敗 (エラー 117)

エラー 117: サーバとの初期接続に失敗しました。LifeKeeper GUI は接続を再試行します。[LifeKeeper GUI](#) のドキュメントを参照し、SPS と LifeKeeper GUI サーバがこのシステムで稼働していることを確認してください。[Java RMI のバインドに関する問題](#) を参照してください。

エラー 118: クライアントパッケージとサーバパッケージに互換性がないため、接続に失敗しました。ターゲットサーバとクライアント起動元サーバのバージョンを調べて、互換性があることを確認してください。

エラー 119: リモートオブジェクトをエクスポートできませんでした。

エラー 120: リモートメソッドの呼び出しで例外が発生しました。

エラー 121: 管理 JavaBean を初期化できませんでした。

エラー 122: 管理 JavaBean にロードできるダイアログクラスがありません。管理 JavaBean を記述したプロパティファイルに「list」プロパティがありません。

エラー 123: 管理 JavaBean を記述したプロパティファイルに次のプロパティがありません。

エラー 124: プロパティバンドルが見つかりませんでした。

エラー 125: URLClassLoader を作成しようとしたときにセキュリティ例外が発生しました。java.policy ファイルで正しいパーミッションが付与されていることを確認してください。通常は java.policy ファイルをホームディレクトリに作成します。java.policy ファイルの内容には大文字と小文字の区別があります。そのため、LifeKeeper GUI パッケージ付属のサンプルファイルをコピーするのが最善の方法です。

Java のブラウザプラグインを使用している場合は、Java コンソールを有効にして表示される最初の数行を調べれば、Java 環境用に使用されているユーザホームディレクトリをチェックできます。[GUI クライアントの設定](#) の詳細については、LifeKeeper GUI のドキュメントを参照してください。

エラー 126: サーバにリソースが見つかりませんでした。

エラー 127: このキットの拡張プロパティファイルが見つかりませんでした。

エラー 128: 内部プロパティファイルエラーです。

エラー 129: サーバへの RMI 接続を確立できません。サーバ上で LifeKeeper GUI サーバが稼働していることを確認してください。

エラー 130: 入力したタグは別のリソースで使用されています。別のタグを入力してください。

エラー 131: ユーザインターフェースの更新のために invokeAndWait メソッドを呼び出しているときに例外が発生しました。

エラー 132: 管理 JavaBean の呼び出し中に例外が発生しました。

エラー 133: イクイバレンシの優先順位として入力した値が無効です。優先順位の値は 1 から 999 の範囲になければなりません。

エラー 134: イクイバレンシの優先順位の値がテーブル内の別の優先順位と競合しています。イクイバレンシの優先順位の値はそれぞれテーブル内で一意になければなりません。

GUI ネットワーク関連 - サーバへの初期接続の失敗 (エラー 117)

現象

サーバとの初期接続に失敗しました (エラー 117)。

複数のネットワークインターフェースカード (NIC) を持つサーバに接続する場合、最初の NIC の IP アドレスが到達不能な場合、Java RMI のバインドに関する問題が発生します。最初の NIC とは、ipconfig コマンドが出力する最初の IP アドレスを指します。

解決策

SPS サーバのネットワークサービスが使用するプロトコルバインディングの順序を変更してください。SPS サーバ上で [ネットワークとダイヤルアップ接続] を開き、[詳細設定] メニューの [詳細設定] をクリックしてください。ダイアログボックス上部のリストボックスに NIC が表示されます。これが現在の順番です。矢印ボタンを使用して、到達可能な NIC をリストの最上位に移動してください。このように NIC の順番を変更すると、Java RMI は、クライアントによるサーバへの接続を許可します。サーバをリブートして設定を反映してください。

GUI ネットワーク関連 - Windows プラットフォーム上の長期接続遅延

現象

Windows プラットフォーム上で、接続が長時間遅延します。

解決策

サンマイクロシステムズ社の FAQ より

「通常の場合、ホストのネットワーク設定が不適切なことが原因です。RMI は JavaAPI のネットワークのクラス (java.net.InetAddress) を使用して、両方のホストの TCP/IP ホスト名を参照し、IP アドレスとホスト名をマッピングします。Windows プラットフォームでは、この参照はネイティブの Windows ソケットライブラリが実行するため、遅延は RMI ではなく、Windows ライブラリで発生します。ホストで DNS を使用する場合、DNS サーバは通信に参加するホストを認識しないため、DNS 参照がタイムアウトになることが問題となります。その場合は、ホスト名/アドレスをローカルの *local file\winnt\system32\drivers\etc\hosts* ファイルまたは *\windows\hosts* ファイルに記述します。ホストファイルは次の書式で記述します。

```
IPAddress Server Name
```

```
e.g.: 208.2.84.61 homer.somecompany.com
```

以上のように記述すると、最初の参照時間を短縮できます。」

また、サブネットマスクとゲートウェイアドレスの設定を間違えると、接続遅延や障害の原因となります。DNS の設定が正しいことをネットワーク管理者に確認してください。

GUI ネットワーク関連 - 接続試行時に生成される NoRouteToHostException メッセージ

現象

解決策

接続試行時に `NoRouteToHostException` というメッセージが生成されます。

ホストとの接続を確立できないため、ソケットをリモートホストに接続できません。

解決策

ローカルサーバとリモートサーバを結ぶネットワークの一部のリンクが停止しているか、リモートサーバがファイアウォールで保護されています。

GUI ネットワーク関連 - 接続試行時に生成される不明なホスト例外メッセージ

現象

接続試行時に不明なホスト例外メッセージが生成されます。

LifeKeeper GUI のクライアントとサーバは、Java RMI (Remote Method Invocation) 技術を使用して通信します。RMI が適切に動作するには、クライアントとサーバは、解決可能なホスト名または IP アドレスを使用する必要があります。解決不能な名前 (WINS 名) または不適切な DHCP 名を使用すると、Java は `UnknownHostException` をスローします。

このエラーメッセージは、次の場合にも表示されます。

- サーバ名が見つからない。サーバ名のスペルを確認してください。
- DHCP サーバの設定が不適切な場合、RMI サーバの完全修飾ドメイン名が、RMI サーバの実際に存在するドメインではなく、リゾルバドメインのドメイン名に設定されることがあります。その場合、ドメイン名が不正なため、サーバの DHCP ドメインの外側にある RMI クライアントはサーバに接続できません。
- サーバが Windows Internet Naming Service (WINS) を使用するネットワーク上にある。WINS に登録されたホストは、DNS だけを使用するホストから接続できないことがあります。
- RMI クライアントとサーバがファイアウォールを挟んだ両側にある。RMI クライアントがファイアウォールの外側にあり、サーバがその内側にある場合、クライアントはリモートコールをサーバに送信できません。

解決策

LifeKeeper GUI を使用するときは、クライアントが使用するホスト名は、サーバが解決できなければなりません。また、クライアントはサーバのホスト名を解決できなければなりません。LifeKeeper GUI はこの例外を捕捉してユーザに警告します。クライアントがサーバのホスト名を解決できない場合は、この例外がスローされて、エラーメッセージ 115 が表示されます。サーバがクライアントのホスト名を解決できない場合は、この例外がスローされて、エラーメッセージ 116 が表示されます。2つのメッセージは、ホストが見つからないことを通知する Java の例外です。

ホスト名が適切に解決されていることをテストして確認するには、以降に示す手順に従ってください。

Windows の場合

1. SPS サーバとの通信が確立していることを確認します。プロンプトから、ホスト名を指定して ping を打ってください。

```
ping<TARGET_NAME>
```

以下に例を示します。

```
ping homer
```

ターゲットのホスト名とIP アドレスが返ります。

2. 設定が適切なことを確認します。
 - a. DNS の設定を確認するか、DNS サーバをネットワークにインストールしてください。
 - b. DNS の設定を確認するには、**[コントロールパネル] -> [ネットワーク] -> [プロトコル] -> [TCP/IP]**の順に選択してください。DNS の設定が正しいことをネットワーク管理者に確認してください。[DNS] タブのホスト名がローカルのネームサーバと同じであることを確認してください。また、この名前が GUI エラーメッセージに表示されたホスト名であることを確認してください。
 - c. ローカルホストと接続先の SPS サーバを hosts ファイルに記述してください。

Windows 2003/2008 の hosts ファイルは、次の場所にあります。

```
%SystemRoot%\system32\drivers\etc\HOSTS (e.g.  
C:\windows\system32\drivers\etc\HOSTS)
```

注記: Windows 2003/2008 では、hosts ファイルの最終エントリの後に改行記号 (CR/LF) がないと、hosts ファイルは参照されません。

例えば、ユーザのシステムが HOSTCLIENT.MYDOMAIN.COM であり、IP アドレスが 153.66.140.1 である場合、hosts ファイルに次のエントリを追加します。

```
153.66.140.1 HOSTCLIENT.MYDOMAIN.COM
```

3. GUI クライアントが使用するホスト名のプロパティを設定します。ホスト名のプロパティを設定するには、Java Plug-in をインストールした Web ブラウザを開き、**Java プラグインコントロールパネル**を表示して、次の 1 行を [Java Run Time Parameters] に追加してください。

```
-Djava.rmi.server.hostname=<MY_HOST>
```

4. www.microsoft.com にアクセスして、Microsoft が提供するネットワーク関連のパッチを確認します。

Linux の場合

1. ホスト名または IP アドレスを指定して ping を打ち、ターゲットサーバとの接続が確立していることを確認します。

```
ping -s<TARGET_NAME>
```

以下に例を示します。

```
ping -s homer
```

ターゲットのホスト名が返ります。

2. ホスト名または IP アドレスを指定して ping を打ち、クスタサーバが *localhost* 名を解決していることを確認します。DNS が実装されていない場合は、*/etc/hosts* ファイルを編集し、*localhost* 名を追加してください。ローカルサーバの IP アドレス、またはデフォルトエントリ (127.0.0.1) を指定できます。
3. DNS が NIS の前に記述されていることを確認します。DNS は、*/etc/nsswitch.conf* のホスト行で NIS の前に記述します。また、*/etc/resolv.conf* には適切に設定された DNS サーバを指定します。
4. DNS が実装されていないか、他の名前解決プロトコルが機能していない場合は、ホスト名を */etc/hosts* ファイルに記述します。
5. GUI クライアントが使用するホスト名のプロパティを設定します。これは、管理者ごとに変更する必要があります。

ホスト名のプロパティを設定するには、Java Plug-in をインストールした Web ブラウザを開き、**Java プラグインコントロールパネル**を表示して、次の 1 行を **[Java RunTime Parameters]** に追加してください。

```
-Djava.rmi.server.hostname=<MY_HOST>
```

この作業を HotJava ブラウザから実行するには、次の 1 行を hotjava コマンドラインに追加してください。

```
-Djava.rmi.server.hostname=<MY_HOST>
```

例

```
-Djava.rmi.server.hostname=153.66.140.1
```

```
-Djava.rmi.server.hostname= homer.somecompany.com
```

GUI サーバのトラブルシューティング

現象

LifeKeeper GUI は、管理 Web サーバと Java リモートオブジェクトレジストリに、各サーバのポート 81 および 82 を使用します。別のアプリケーションが同じポートを使用している場合、LifeKeeper GUI は正常に機能しません。

解決策

値を変更するには、以下のレジストリエントリを編集してください。

```
GUI_WEB_PORT=81
```

```
GUI_RMI_PORT=82
```

これらのエントリは、以下のレジストリキーにあります。

```
HKEY_LOCAL_MACHINE\SOFTWARE\SteelEye\LifeKeeper\JavaGUI\Server
```

注記: ポートの値は GUI サーバの起動時に初期化されます。変更した場合は、GUI サーバを停止して再起動する必要があります。これらの値は、接続するクラスタすべてで同じでなければなりません。

ヘルスチェックタイムアウト

現象: 時として SPS は正常に動作しないシステムにインストールされることがあります。そこには SPS が検知することが可能な、ヘルスチェックが正常に起動または終了しないといったような前兆現象があります。チェック処理のタイムアウトを引き起こすもっとも一般的な問題は、データベースまたはメールサーバに対するシステムメモリの最適化が適切でない場合です。常時ヘルスチェックプロセスを起動し、フェールオーバーを実施するためには十分なメモリが必要です。

解決策: SPS リリースノートは、SPS のシステムメモリ要件を定義するためのガイドラインを記載しています。使用するシステムにおいて、SPS のようなアプリケーションに対する十分なメモリがあるかを確認するための Windows Performance Monitors の使用方法もまた説明しています。この状態を確認する手助けとして、チェック処理のタイムアウトが発生した際に、異常なふるまいを記録する2つのリソース監視オプションと、是正操作を行う3つのオプションがあります。5つすべてのオプションは SPS のレジストリ設定で有効または無効にし、ユーザーにあった要件やパフォーマンスを満たすことができます。:

HKEY_LOCAL_MACHINE\SOFTWARE\SteelEye\LifeKeeper\General\ResMon_RecordTimeout

HKEY_LOCAL_MACHINE\SOFTWARE\SteelEye\LifeKeeper\General\ResMon_RecordMemory

HKEY_LOCAL_MACHINE\SOFTWARE\SteelEye\LifeKeeper\General\ResMon_ResFail

HKEY_LOCAL_MACHINE\SOFTWARE\SteelEye\LifeKeeper\General\ResMon_RebootWaitInSec

HKEY_LOCAL_MACHINE\SOFTWARE\SteelEye\LifeKeeper\General\ResMon_ResFailMaxWaitInMin

最初の4つのオプション (RecordTimeout, RecordMemory, ResFail, と RebootWaitInSec) は、期待される時間内に完了しない保護対象のリソースを監視している SPS Quick Check または Deep Check プロセスによって引き起こされます。

ResMon_RecordTimeout – このオプションはタイムアウトするクイックチェックもしくはディープチェック処理を記録します。この情報は <LifeKeeper Root Folder>\Out\ResMonTimeout.log ファイルに記録されます。このオプションはデフォルトで有効 (=1) になっています。無効にする場合は値を 0 に設定してください。

ResMon_RecordMemory – このオプションは、クイックチェックもしくはディープチェックがタイムアウトするとアクティブなプロセスに対するシステムメモリの使用量とプロセスメモリの使用量を記録します。メモリの使用率は <LifeKeeper Root Folder>\Out\ResMonTimeout.log file に記録されます。このオプションはデフォルトで有効 (=1) になっています。無効にする場合は値を 0 に設定してください。

ResMon_ResFail – このオプションはクイックチェックもしくはディープチェックがタイムアウトしたときリソース階層のフェールオーバーを発生させます。影響を受ける階層の従属リソースがフェールオーバーします。このオプションはデフォルトで無効 (=0) になっています。有効にする場合は値を 1 にしてください。

不完全なリソースの作成

ResMon_RebootWaitInSec – このオプションはクイックチェックもしくはディープチェックがタイムアウトしたときシステムの再起動を発生させます。このオプションはデフォルトで無効 (=0) になっています。有効にする場合はレジストリ設定で0以外を入力してください。この数は、システムがこの機能により再起動されるときにシステムコンソール上に表示されるカウントダウンになります。再起動のシーケンスは自動およびシステム操作を必要としないデザインになっています。一旦カウントダウンが開始されると、再起動は停止することができません。

ResMon_ResFailMaxWaitInMin – このオプションは、フェールオーバー発生時にSPSのフェールオーバー処理を監視します。レジストリ設定の値は、SPSがリソースのフェールオーバー完了までに待つ時間(分)になります。フェールオーバー処理が開始されない場合、もしくはフェールオーバー処理が指定された時間内に完了しない場合は、SPSはシステムの再起動を試みます。リソースがフェールオーバーされなかった場合は、再起動されたサーバと同じ場所でin serviceされます。このオプションはデフォルトで無効 (=0) になっています。

リソース監視オプション はすべての保護対象のリソースに対して適用され、いつでもそれらを変更することが可能です。変更された設定は、次回クイックチェックもしくはディープチェックプロセスが開始された時に反映されます。

不完全なリソースの作成

説明

リソースの設定プロセスが途中で中断し、インスタンスが不完全な状態で作成された場合は、リソースを再作成する前にクリーンアップを手動で実行してください。部分的に作成されたリソースは、LifeKeeperのGUIを使用して削除します。詳細については、[全サーバのリソース階層の削除](#)を参照してください。不完全なリソースが階層の一覧に表示されない場合は、`ins_remove` と `dep_remove` コマンドでクリーンアップしてください。

インストール - アクセス拒否

現象

SPSのアップグレード時または再インストール時に、「Access is denied」というエラーメッセージが表示されます。

解決策

SPSサービスが完全に停止していません。これは、セットアッププログラムがSPSサービスを停止することができなかった場合に発生する可能性があります。コマンドウィンドウを開いて `$LKROOT\bin\lkstop` と入力して、すべてのSPSサービスを停止してください。その後、「LIFEKEEPER NOW STOPPED」と表示されるまで待機してからセットアップを実行してください。

IP リソース作成問題

現象

新しいIPアドレスが既存のIPアドレスと部分的に一致した場合IPリソースの**Pre-Extend Wizard**に失敗します。

ワークアラウンド

現在有効なワークアラウンドはありません。この問題については将来のリリースにおいて対応されます。

Java の署名があるコードとないコードの混在の警告

現象

リモートシステムから LifeKeeper Java GUI クライアントアプレットを読み込むとき、以下のセキュリティ警告が表示される場合があります。



[実行]を選択すると、以下のダイアログが表示されます。



ブロックについては [いいえ] を選択すると、LifeKeeper GUI の実行が許可されます。

解決策

以下の2つの方法でセキュリティ警告の数を減らすことができます。

1. この発行者からのコンテンツを常に信頼します チェックボックスをオンにして、**実行** を選択してください。次に LifeKeeper GUI Java クライアントを読み込むときは警告メッセージが表示されません。

または

2. ブロックに関する2つ目のダイアログが表示されないようにするには、Java の「**deployment.properties**」ファイルに以下のエントリを追加してください。Java クライアントを読み込むときにセキュリティ警告は表示されますが、アプレットはブロックされず、ブロックの確認ダイアログは表示されません。この設定はすべての Java アプレットに適用されることに注意してください。

```
deployment.security.mixcode=HIDE_RUN
```

両方のメッセージを回避するには、1 および 2 を実行してください。

参照リストに LANMAN 名が 2 回表示される場合がある

現象

LANMAN リソースを作成した後、参照リストに LANMAN 名が 2 回表示される場合があります。

解決策

エントリの 1 つは使用できないワークステーションレコードです。このエントリを無視し、もう 1 つの LANMAN サーバ名を使用してください。

ライセンス - ライセンスされたリカバリキットリソースが起動しない

現象

ライセンスされたリカバリキットをアップグレードした後、リソースが起動せず、SPSによって以下のエラーがアプリケーションイベントログに記録されます。

```
"Process: lcdmachfail(3176) *ERROR* (No. 1001) resource <tag name> requires a license (for Kit <recovery kit type>) but none is installed."
```

解決策

SPS ライセンスユーティリティを使用して、リカバリキットのライセンスキーをインストールしてください。ライセンスのインストールについては、SPS for Windows インストールガイドのライセンスの取得とインストールを参照してください。

ライセンス - ライセンスキーが見つからない

現象

ライセンスされたリカバリキットをインストールした後、SPS によって以下のエラーがアプリケーションイベントログに記録されます。

```
"Process: Lkinit:(1832) *ERROR* (No. 20042) SPS Recovery Kit <licensed recovery kit> license key NOT FOUND".
```

解決策

SPS ライセンスユーティリティを使用して、リカバリキットのライセンスキーをインストールしてください。ライセンスのインストールについては SPS for Windows インストールガイドのライセンスの取得とインストールを参照してください。

LifeKeeper Web クライアントがロックされる場合がある

現象

サーバマシンから使用しているとき、そのサーバについての SPS リソースの初期描画時に「-」（リソースの高さを減らす）または「+」（リソースの高さを増やす）アクセラレータキーを押した場合、SPS Web クライアントがロックされることがあります。

解決策

復旧するには、Windows タスクマネージャを開き、「LifeKeeper - <web browser (例: Microsoft Internet Explorer)>」アプリケーションに対して **タスクの終了** を選択してください。プロセスを終了するのに最大で 1 分かかる場合があります。スタート-> [プログラム->SteelEye ショートカットから SPS Web ク

SPS Web クライアントがロックされる場合がある - 既存の階層が複数変更された

クライアントを再起動し、初期画面の描画が完了するまで待機してからアクセラレータキーを使用してください。

SPS Web クライアントがロックされる場合がある - 既存の階層が複数変更された

現象

サーバマシンから使用しているとき、既存の階層に対して複数の変更（つまり、依存関係の作成/削除）が行われ、その変更の間に SPS クライアントを閉じてから開き直した場合、SPS Web クライアントがロックされることがあります。

解決策

復旧するには、**Windows タスクマネージャ**を開き、**LifeKeeper - <web browser (例: Microsoft Internet Explorer)>**アプリケーションに対して **[タスクの終了]**を選択してください。プロセスを終了するのに最大で 1 分かかる場合があります。サーバの階層管理には LifeKeeper GUI を使用してください。ショートカットは、**スタート -> プログラム -> SteelEye -> LifeKeeper -> LifeKeeper GUI** です。

新しい評価ライセンスキーのエラー

現象

SPS for Windows v7.2 の評価時、新しい評価ライセンスキーを使用していないとエラーが発生する場合があります。古い評価用ライセンスはこのリリースでは機能しません。

解決策

新しい評価ライセンスキーを取得する必要があります。**ライセンスキーマネージャ**を再起動して、新しいライセンスキーを適切な形式で入力してください。

1 x 1 構成における SPS のサーバ障害からの復旧

SPS クラスタにおいてサーバにオペレーティングシステムの再インストール(SPSも同じく)が必要となるような障害が発生した場合、クラスタの各サーバからリソース階層を再拡張することが必要となります。クラスタのサーバが再インストールされるサーバとイクイバレンス関係を共有している場合でも SPS では、再インストールされるサーバへ既存のリソース階層を拡張することができません。実際には階層が再インストールされたサーバ上に存在しないため再インストールされるサーバから階層を拡張解除することもできません。

推奨されるアクション:

オペレーティングシステムおよび SteelEye Protection Suite と同様にすべての関連するパッチを再インストールした後、以下の手順に従って復旧を行ってください。(注意: 下記で説明する推奨されるアクションの例において "BENHOGAN" および "GPLAYER" というテストシステム名を使用します。):

1. リソース階層が構成されている各サーバで、[eqv_list](#) コマンドを使用して、すべての共有イクイバレンシの一覧を取得してください。

```
eqv_list [-d destsys] [-s sys] [-t tag] [-e SHARED]
[-fc]
```

この機能は、リソースインスタンス間のイクイバレンシ関係を記述している一般的なアウトプットを表示します。

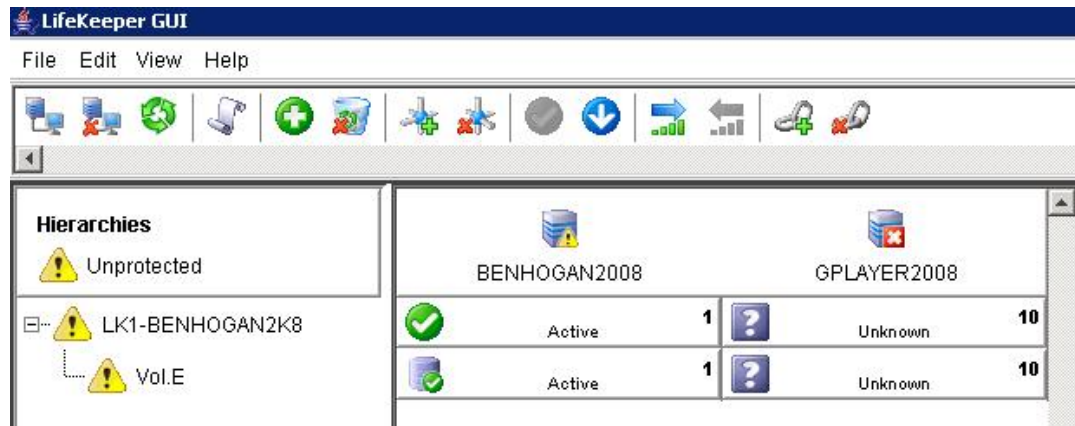
例:

```
c:\LK\Bin>eqv_list

BENHOGAN2008 LK1-BENHOGAN2K8 GPLAYER2008 LK1-
BENHOGAN2K8 SHARED 1 10

BENHOGAN2008 Vol.E GPLAYER2008 Vol.E SHARED 1 10
```

下記の LifeKeeper GUI の階層やリソースなどを参照してください。



2. リソース階層が構成されているサーバで[eqv_remove](#)を使用して手動で階層の各リソースのイクイバレンシ関係を削除してください。

システムsys と othersys それぞれに存在するリソースタグ名とその他タグ間で、-e オプションを指定し、イクイバレンシタイプのシステムdestsys(指定されない場合はローカル)上のコンフィギュレーションデータベースからイクイバレンシを削除します。

```
eqv_remove [-d destsys] [-s sys] -t tag [-S
othersys]-o othertag [-e SHARED]
```

```
eqv_remove -s {this system} -t {TAGNAME} -S
{othersys that has gone away} [-e SHARED]
```

例:

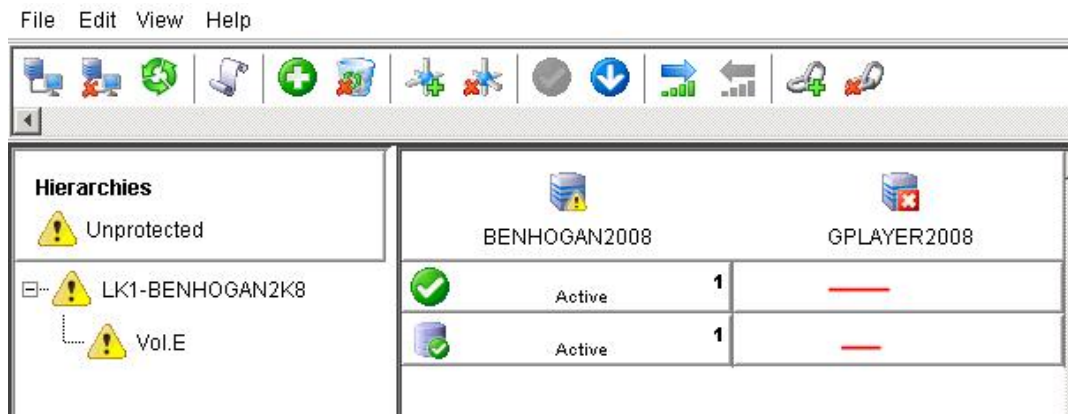
```
c:\LK\Bin>eqv_remove -s BENHOGAN2008 -t LK1-
BENHOGAN2K8 -S GPLAYER2008 -e SHARED

c:\LK\Bin>
```

推奨されるアクション:

3. [eqv_list](#) コマンドを再度実行してください。共有イクイバレンシの一覧は表示されません。

ターゲットリソースどのように削除されているかを下記の LifeKeeper GUI で確認してください。:



4. クラスタにおいてDatakeeper でミラーされているボリュームが構成されている場合は、そのボリュームに対する LKDRInfo ファイルをクリーンアップしてください。

SPS は、各ミラー対象のボリュームリソースに対し

て `%LKROOT%\subsys\filesys\resources\volume` に `LKDRInfo.<volume>` を作成します。このファイルは削除されます。

```
C:\LK\BIN> cd
%LKROOT%\subsys\filesys\resources\volume

C:\LK\subsys\filesys\resources\volume> dir
LKDRInfo.E

05/23/2012 01:02 PM 39 LKDRInfo.E

C:\LK\subsys\filesys\resources\volume> del
LKDRInfo.E
```

5. クラスタにおいてDatakeeper でミラーされているボリュームが構成されている場合は、ミラーをクリーンアップし、ミラーを含む DataKeeper ジョブを削除してください。

クラスタが再インストールされるサーバに再拡張する際に DataKeeper のミラーとジョブを再作成する必要があります。そのため、構成されるミラーのローカルエンドを削除し、それらのミラーとともに構成されるジョブを削除する必要があります。

```
C:\LK\subsys\filesys\resources\volume> cd
%ExtMirrBase%

C:\Program Files\SteelEye\DataKeeper> emcmd .
getjobinfoforvol E

ID = e829700c-27b0-447f-b852-1a3135da31a7
```

```

Name = E Vol

Description =

MirrorEndpoints =
BENHOGAN2008;E;10.200.8.25;GPLAYER2008;E
;10.200.8.26;A

C:\Program Files\SteelEye\DataKeeper> reg /delete
HKLM\System\CurrentControlSet\Services\ExtMirr\Para-
meters\Jobs\e829700c-27b0-447f-b852-1a3135da31a7

Permanently delete the registry key HKEY_LOCAL_
MACHINE\System\CurrentControlSet\Services\ExtMirr\P-
arameters\Jobs\1b5e8715-a488-4030-8166-45c9232bc04e
(Yes/No)? y

The operation completed successfully.

C:\Program Files\SteelEye\DataKeeper> emcmd .
getjobinfoforvol E

C:\Program Files\ SteelEye\DataKeeper> emcmd .
deletelocalmirroronly E

Status = 0

C:\Program Files\ SteelEye\DataKeeper> emcmd .
clearswitchover E

Status = 0

```

これらの手順をすべてのミラー対象のボリュームに対して繰り返してください。このジョブはレジストリディレクトリからの削除が必要です。;emcmd . deletejobを使用して削除しようするとDataKeeperが全ノードでのジョブの削除を試みるので操作に失敗します。これらのノードのひとつがもう存在しておらず、ジョブがそのまま残るからです。

6. 古いシステムを削除する準備が整いました。
 - a. SPS クラスタのすべてのシステムを表示させます。:

```
sys_list [-d destsys]
```

例:

```

c:\LK\Bin>sys_list

BENHOGAN2008

GPLAYER2008

```

- b. すべての古いターゲット/システムを削除します。

推奨されるアクション:

```
sys\_remove [-d destsys] -s sys
```

例:

```
c:\LK\Bin>sys_remove -s GPLAYER2008
```

SPS クラスタのすべてのシステムで実行してください。

注意: このコマンドをローカルソースシステムに対して実行しようとする、以下のエラーが表示されます。:

```
c:\LK\Bin>sys_remove -s BENHOGAN2008

(null)Process: sys_remove(2728)

[File:sys.C Line:81]

*ERROR* (No. 424) can't remove entry for
local system "BENHOGAN2008"
```

7. コミュニケーションパスを削除してください。

- a. コミュニケーションパスが存在することを確認してください。:

```
net_list
```

- b. これらのコミュニケーションパスを削除してください。:

```
net_remove
```

8. 変更を加えてください。

```
lcdsync [-d destname]
```

例:

```
c:\LK\Bin>lcdsync
```

この機能は共有メモリに保存されているSPSリソース階層の構成とコミュニケーションパスの状態が修正されているかどうかをチェックします。もし相違があったら、データはディスクに対して"同期"書き込みを行います。これによって、このプログラムが終了すると、正しいデータがディスク上で保障されます。

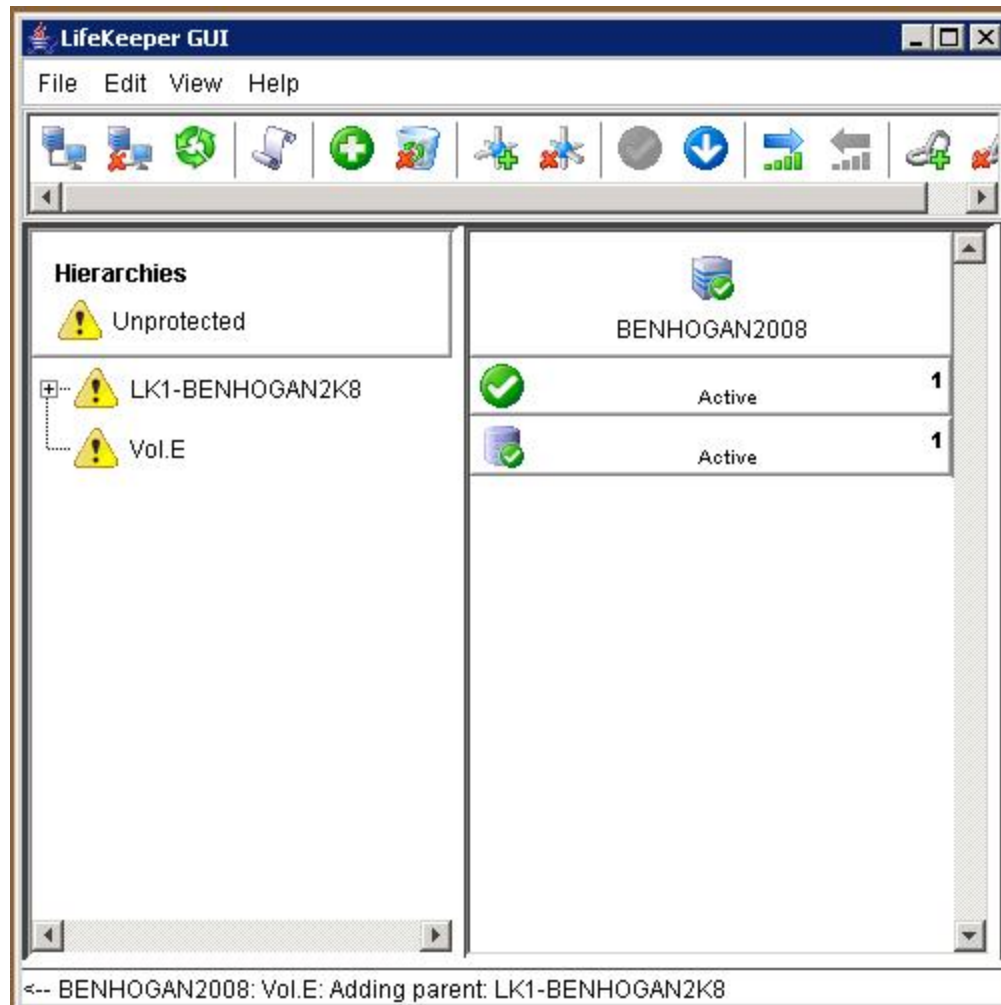
追加情報: すべてのリソースを完全に削除する場合は、まずはじめにすべてのリソースインスタンスの一覧を生成してください。:

```
ins\_list [-d destsys] [-fC] [-R top] [-a appname] [-r typ] [-t tag] [-i id]
```

その後すべてのインスタンスを削除してください。:

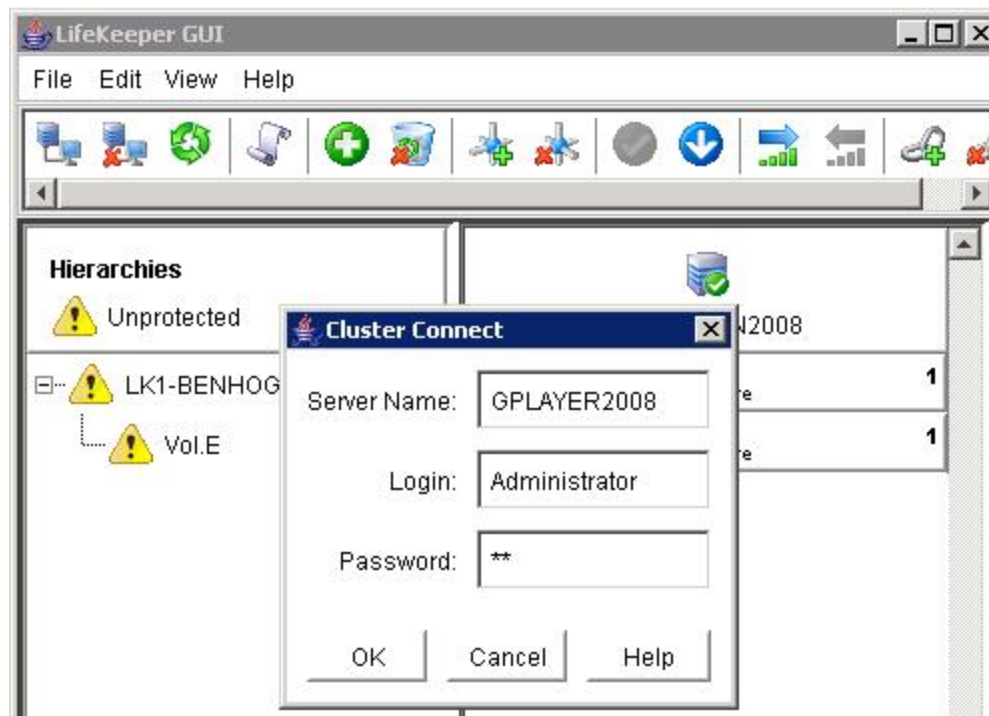
```
ins\_remove [-d destsys] [-R roottag] [-a appname] [-r restyp] [-t tag] [-i id] [-v] [-I] [-N] [-G]
```

9. LifeKeeper GUI を閉じて、再度起動してください。

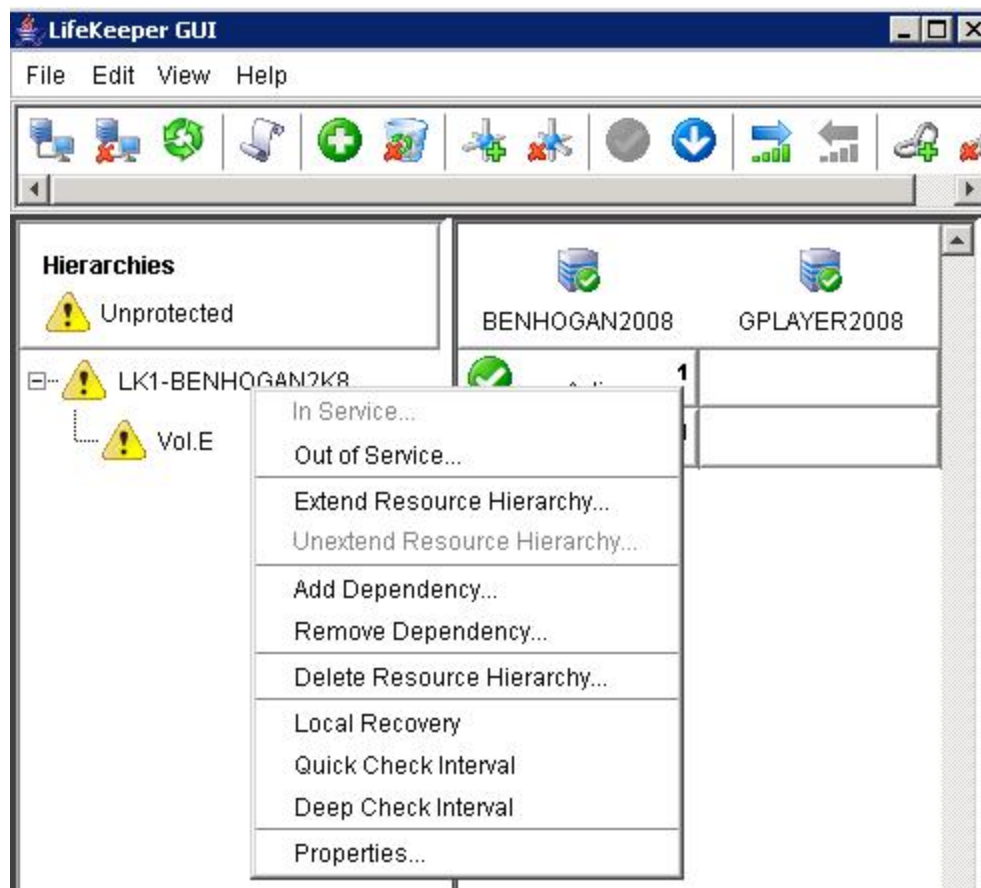


10. GUIを使用して、各リソース階層をリソース階層がin serviceになっているサーバから再インストールされたサーバへ拡張してください。
 - a. 再インストールされたサーバターゲットに接続してください。注意:コミュニケーションパスの再作成もまた必要になります。

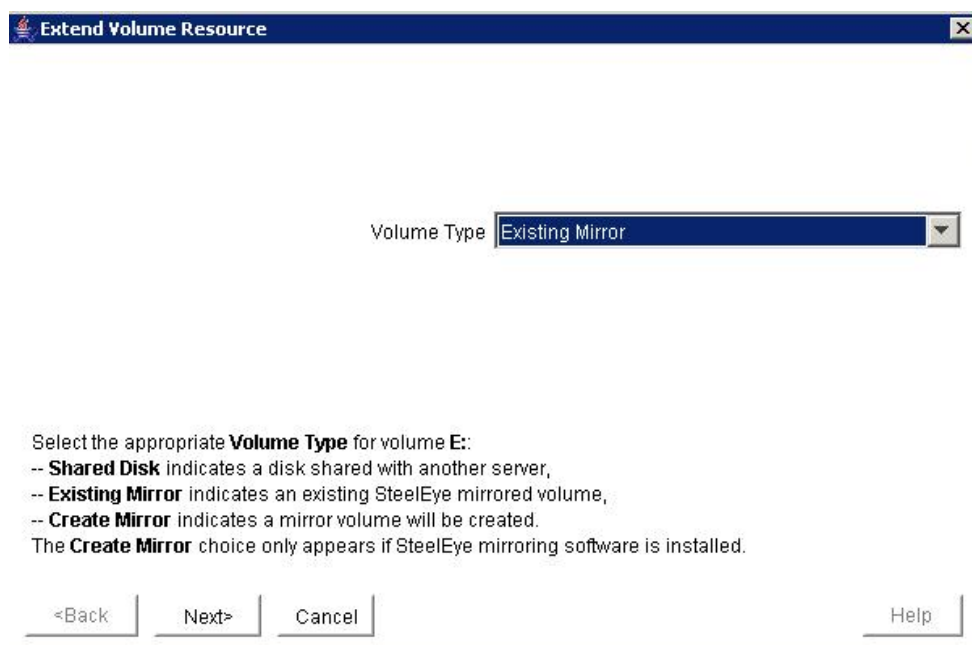
推奨されるアクション:



b. リソース階層を拡張してください。



c. ミラーリソースを追加/拡張してください。



完了 – すべての保護対象のリソースが復旧し、再度保護対象となります。

サービス休止中の階層のリカバリ

説明

障害が発生したサーバ上のリソース階層は、障害発生時に他のサーバ上で起動されていなくても、リカバリ処理の一環として、障害発生時に優先度が最高のサーバにリカバリされます。このリカバリ操作は、サービスを休止しているサーバが最後にサービスを稼働していたサーバ(障害が発生したサーバ)、バックアップサーバ、他のクラスタサーバでも、同様に処理されます。

リカバリキットのアンインストール時に Remove がハングする

現象

サービス中のリカバリキットのリソース階層が存在するときにそのリカバリキットをアンインストールした場合、**Remove** がハングします。

これを回避するために、必ずリカバリキットのリソース階層のサービスを停止し、削除してから、リカバリキットソフトウェアをアンインストールすることを推奨します。

解決策

この状況では多数の関連プロセスがハングしており、すべてをクリアするのは難しいので、ほとんどの場合はシステムを再起動する必要があります。

複製ボリュームのスイッチオーバーの失敗

説明

複製ボリュームがスイッチオーバーする際、DELETEMIRROR コマンドが失敗すると、以下のエラーメッセージがアウトプットパネルに表示されます。

```
*WARNING* (No. 12104) Delete mirror action for volume R:
failed with status 51 (command=C:/SDR/EMCmd10.10.1.2
DELETEMIRROR R: 10.10.1.1)
```

エラーステータス番号、説明、および推奨される処置に関する以下の表を参照してください。また、以下のコマンドを使用することでも、エラーステータスについての情報を得ることができます。

コマンド: net helpmsg {status}

例: net helpmsg 51

エラーステータス	説明	推奨される処置
5	現在のソース上で、ミラーが削除されることを許可していません。	両システムでミラーボリュームのアクセスからローカルシステムアカウントを除外しているかチェックしてください。
46	システム間でユーザ/パスワードが適合していません。コマンドプロンプトからの DELETEMIRROR が作動している場合にのみ発生します。	ドメインアカウントを使用するか、あなたが登録したローカルユーザアカウントと同じパスワードを使用してください。
51	Windows はネットワークパスを見つけることができません。	両システムで、すべてのネットワークカードが、Microsoft ネットワーク用ファイルとプリンタ共有にチェックされていることを確認してください。
53	指定された IP アドレスにアクセスできません。	ネットワーク設定を確認してください (HOSTS ファイルや DNS は一貫して IP アドレスを決定します)。
207	ring 2 stack が使用中です。	両システムで、すべてのネットワークカードが Microsoft ネットワーク用クライアントにチェックされていることを確認してください。

リソースタグ名の制限

タグ名の長さ

SPS に含まれる全てのタグ名は256文字の制限を超えてはいけません。

有効な "特殊" 文字

- _ . /

しかしながら、タグの先頭の文字に "." もしくは "/" を含めることができません。

無効な文字

+ ; : ! @ # \$ * = "space"

リストアおよびヘルスチェックアカウントの障害

現象

一部のリカバリキットは、ユーザやクライアントの動作をシミュレートするクエリ操作を実行することで、保護対象リソースを監視しています。これによって、SPS は保護対象のアプリケーションやサービスに関する正確なステータス情報を得ることができます。また、リソースオブジェクト作成時には、ログイン権限を持った有効なユーザアカウント ID およびパスワードが必要になります。ユーザアカウントが特定のシステム上でログイン権限を持っていない場合、**Windows アプリケーションイベントログ**に以下のエラーメッセージが記録されます。

```
Error Number 1385 - "Logon failure: the user has not been granted the requested logon type at this computer."
```

解決策

ドメイン管理者からユーザアカウントにログイン権限を付与してもらってください。また、ID とパスワードが必要なリカバリキットのほとんどにリソースプロパティまたは構成タブがあり、管理者はリソースに対するアカウント情報を変更できます。リソースオブジェクトを右クリックし、該当するプロパティまたは構成タブを選択してください。リソースにアカウント更新機能がない場合は、リソースオブジェクトを削除してから、アカウント情報を更新して新しく作成する必要があります。

SQL 2005 および SQL 2008

現象

SQL 2005 および SQL 2008 を使用する場合、スイッチオーバーまたはフェイルオーバーの後に変数 @@servername がプライマリシステムを参照しています。

解決策

変数 @@servernameを使用する代わりに、select SERVERPROPERTY('ServerName')を使用してください。このクエリは、スイッチオーバーまたはフェイルオーバーの後も正しいマシン名を返します。

または

1. 新しいバックアップサーバで以下のコマンドを実行してください。

```
sp_dropserver @server='sys-A'
sp_addserver @server='sys-B', @local='LOCAL'
```

2. サービスを再起動してください。

SQL Server Reporting Services (MSSQLSERVER)

現象

SQL Server 2008 R2 サービスを保護する場合、オプションの保護対象サービスとして、「SQL Server Reporting Services (MSSQLSERVER)」が選択されることがあります。しかしながら、このサービスの起動に必要な時間が、デフォルトのWindows サービスタイムアウトを超える場合、エラー 1053 が発生してサービスの起動に失敗することがあります。SPS リソースのサービス中処理は失敗します。

解決策

この問題は、システムのパフォーマンスおよび構成の問題に関連している場合があります。このサービスは保護しないことを推奨します。しかしながら、このサービスを保護する必要がある場合は、次のレジストリ設定によって、サービスが起動するまでの時間を延長することができます。HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\ControlレジストリキーにDWORDのServicesPipeTimeout値を追加し、値に60000(十進数、=60秒)を設定してください。

2 サーバクラスタの問題

現象

2サーバクラスタでは、プライマリサーバに障害が発生したか、シャットダウンされて、バックアップサーバへのリソース階層のフェイルオーバーが行われたとき、バックアップサーバへの階層のフェイルオーバーが完全に実行される前に、バックアップサーバにも障害が発生したかシャットダウンされた場合、以下の動作が検出されます。

両方のサーバを再起動したとき、階層内の一部のリソースは一方のサーバで起動し、一部はもう一方のサーバで起動します。上位の親リソースがいずれのサーバでも起動していない場合があります。

解決策

両方のサーバを再起動し、SPSの初期化が完了した後、階層管理インターフェースから起動しなかった階層内の親リソースを選択し、手動で起動してください。すべての階層が起動するまで、この作業を繰り返してください。

不明なユーザ名または誤ったパスワード

アクセス拒否: 不明なユーザ名または誤ったパスワード

現象

SPS クライアントがシャットダウン中のサーバと通信しようとした場合、サーバはクライアントのログオンを拒否して、認証プロセスを中止することがあります。その場合、クライアントには次のメッセージが表示されます。「Access Denied: unknown user name or bad password. Only members of the SPS-authorized security groups can use SPS. Would you like to re-enter the authentication data?」

解決策

新しい証明書を入力するために **[はい]** をクリックしてから、**[キャンセル]** をクリックするか、証明書を再入力して **[OK]** をクリックしてください。注記: 最初に **[いいえ]** をクリックした場合、LifeKeeper GUI はそのサーバから切断され、自動再接続は行われません。

Web クライアントのトラブルシューティング

背景

Web クライアントに が表示されません。

回答

Web クライアントに が表示されない場合は、以下を実行してください。

1. アプレットが起動に失敗したか確認してください。通常は、ブラウザのウィンドウ内にアプレットの状態を示すメッセージが表示されます。Netscape と Internet Explorer では、状態を表すメッセージのほか、アプレットの代替のアイコンが表示されることがあります。このアイコンをクリックすると、失敗に関する説明が表示されます。
2. **Java コンソール**を開いてください。
 - **Firefox、Netscape、旧バージョンの Internet Explorer** を使用している場合は、コントロールパネルを開き、**[Java プラグイン]** アイコンをクリックして、コンソール表示を有効にしてください。ブラウザを再起動してください。
 - **新しいバージョンの Internet Explorer** を使用している場合は、**[ツール]** メニューの **[Sun の Java コンソール]** をオンにしてください。[Sun の Java コンソール] メニューが表示されない場合は、**[ツール]** メニューの **[アドオンの管理]** をクリックし、コンソール表示を有効にしてから、ブラウザを再起動すると、Java コンソールが表示されます。
 - **Mozilla** を使用している場合は、**[ツール]** メニューの **[Web 開発]** をクリックし、**[Sun の Java コンソール]** をクリックしてください。

Web クライアントが開かない場合は、URL (`http://<server name>:81,`) を指定して、ブラウザを再起動してください。

コンソールのメッセージを確認してください。エラーメッセージが表示された場合は、それを参考にして問題を解決してください。問題がネットワークに関係している場合は、[ネットワーク関連のトラブルシューティング](#)を参照してください。

Win2008 - IIS リソース階層作成エラー

現象

On a Win2008 system when creating an IIS Resource Hierarchy, you receive the following message:

```
No qualified sites were found...
```

and the create fails.

背景トラブルシューティング:

以下のコマンドを実行してください。:

```
C:\LK\Admin\kit\webapp\bin>enumiis query all
```

以下のエラーメッセージを探してください。

```
ERROR:CoCreateInstance Failed!Error:-2147221164 (80040154)
```

```
ERROR:W3Service Com Object Failed to initialize
```

解決策

Win2008 では、SPS IIS キットに IIS 6 Management Compatibility (役割サービス) が必要です。すべてのオプション (Metabase Compatibility、WMI Compatibility、Scripting Tools、Management Console) をインストールしてください。

Index

C

Core ソフトウェア 2

CYGWIN 129

F

Failover

自動フェイルオーバーを無効にする 22

G

GUI

GUI による管理作業 16

GUI ユーザの設定 97

LifeKeeper GUI サーバコンポーネントおよびクライアントコンポーネント 93

LifeKeeper GUI ユーザアカウント 97

SPS サーバ上の GUI アプリケーションの実行 96

トラブルシューティング

GUI エラーメッセージ 131

GUI サーバのトラブルシューティング 136

GUI ネットワーク関連

サーバへの初期接続の失敗 (エラー 117) 132

接続試行時に生成される NoRouteToHostException メッセージ 133

接続試行時に生成される不明なホスト例外メッセージ 134

H

Hardware Components 13

Heartbeat 6

Interval 6

Hierarchies

Creating 26

File Share 27

Generic Application 28

IP Address 32

LAN Manager 30

Troubleshooting

 Incomplete Resource Creation 138

 Win2008 - IIS Resource Hierarchy Creation Error 155

Volume 31

Deleting 43

Extending 38

 DNS 39

 File Share 39

 Generic Application 39

 IP Address 40

 LAN Manager 39

 Volume Resource 39

Status

 Status Table 82

 Viewing the Status of Resources 101

Unextending 42

Working With 26

I

In Service 104

 Taking Volume Resources In and Out Of Service 105

Intelligent 12

Invalid Characters 152

L

LCD 109

 Diagram of LCD Directory 110

 LCD Configuration Data 110

LCD Directory Structure 109

LCD Resource Types 111

LCM 113

License

Troubleshooting

License Key Not Found 141

Licensed Recovery Kit Resource Fails to Come In Service 141

LifeKeeper Communications Manager 113

LifeKeeper Configuration Database 109

Diagram of LCD Directory 110

LCD Configuration Data 110

LCD Directory Structure 109

LCD Resource Types 111

Resources Subdirectories 111

M

Menus 87

Edit Menu - Resource 89

Edit Menu - Server 90

File 89

Help 93

Resource Context 87

Server Context 88

View 91

Message Bar 83

O

Out of Service

Taking Volume Resources In and Out Of Service 105

Troubleshooting 150

Output Panel 83

P

Properties 83

Resource

Editing Resource Properties 38

Viewing Resource Properties 102

Server 17

Editing Server Properties 16

Viewing Server Properties 100

Protected Resources 1

R

Recovering After Server Failure 142

Recovery Kits 4

Core 2

DNS 3

Generic Application 4

IP 3

LAN Manager 3

Microsoft IIS 4

Volume 3

Microsoft SQL Server 4

Replication 113

Performing Actions on Replicated Volumes 117

Settings 116

Requirements

IP Local Recovery 33

Rewind 123

Performing Rewind on Replicated Volume Resources 125

Settings 124

S

Server Failure 142

Shutdown Strategy 17

Split-Brain 119

Recovery 120

SQL

SQL 2005 and SQL 2008 152

SQL Server Reporting Services (MSSQLSERVER) 153

Status Table 82

T

Tag Name 152

Restrictions 152

Valid Characters 152

Toolbars 84

Global 84

Resource Context 85

Server Context 86

Troubleshooting 129

V

Valid Characters 152

Viewing

Connected Servers 99

Message History 102

Resource Properties 102

Resource Tags and IDs 101

Server Properties 100

Status of a Server 100

Status of Resources 101

Volume Locking 108

Volume Shadow Copy 108

VSS 108

W

Web Client 94

Troubleshooting 154

LifeKeeper Web Client May Lock Up 141

LifeKeeper Web Client May Lock Up - Multiple Changes Made to Existing Hierarchy 142

ア

アクティブ/アクティブ 10

アクティブ/スタンバイ 11

ク

クラスタからの切断 99

クラスタへの接続 98

コ

コミュニケーションパス 4

Component 13

Creating 23

Deleting 25

Types 5

フ

ファイアウォール

Troubleshooting 130

リ

リソース階層ツリーの展開と縮小 103

依

依存関係

Adding 42

Removing 43

構**構成**

LifeKeeper Configurations 13

Steps 10

管**管理 16****自****自動スイッチバック 12**

