



Sun Cluster データサービスの計画 と管理 (Solaris OS 版)

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

Part No: 817-6429-10
2004 年 4 月, Revision A

Copyright 2004 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

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

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

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

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

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

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

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

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

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

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

「ATOK」は、株式会社ジャストシステムの登録商標です。

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

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

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

Unicode は、Unicode, Inc. の商標です。

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

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

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

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

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

原典: Sun Cluster Data Services Planning and Administration Guide for Solaris OS

Part No: 817-4638-10

Revision A



040602@9061



目次

はじめに	7
1 Sun Cluster データサービスの計画	13
Sun Cluster データサービス構成のガイドライン	13
データサービス固有の要件の確認	14
アプリケーションバイナリの格納先の決定	14
nsswitch.conf ファイルの内容の確認	15
クラスタファイルシステムの構成の計画	15
リソースグループとディスクデバイスグループの関係	15
HAStorage と HAStoragePlus の概要	16
データサービスが HAStorage または HAStoragePlus を必要とするかどうかを確認する方法	17
HAStorage または HAStoragePlus の選択	18
考慮事項	18
ノードリストプロパティ	19
インストールと構成プロセスの概要	19
インストールと構成の作業の流れ	20
SPARC: 例	21
データサービスリソースを管理するためのツール	21
SunPlex Manager グラフィカルユーザーインターフェース (GUI)	22
SPARC: Sun Management Center GUI 用の Sun Cluster モジュール	22
scsetup ユーティリティー	22
scrgadm コマンド	22
データサービスリソースの管理作業	23
Sun Cluster データサービスフォルトモニター	24
フォルトモニターの呼び出し	24

2	データサービスリソースの管理	27
	データサービスリソースの管理	28
	Sun Cluster データサービスの構成と管理	30
	リソースタイプの登録	31
	▼ リソースタイプを登録する	31
	リソースタイプのアップグレード	32
	▼ アップグレードされたリソースタイプをインストールして登録する	32
	▼ 既存のリソースを新バージョンのリソースタイプに移行する	33
	リソースタイプのダウングレード	37
	▼ 古いリソースタイプバージョンにダウングレードする	37
	リソースグループの作成	38
	▼ フェイルオーバーリソースグループを作成する	39
	▼ スケーラブルリソースグループを作成する	40
	リソースグループへのリソースの追加	42
	▼ 論理ホスト名リソースをリソースグループに追加する	42
	▼ 共有アドレスリソースをリソースグループに追加する	44
	▼ フェイルオーバーアプリケーションリソースをリソースグループに追加する	46
	▼ スケーラブルアプリケーションリソースをリソースグループに追加する	48
	リソースグループをオンラインにする	50
	▼ リソースグループをオンラインにする	50
	リソースモニターの無効化と有効化	51
	▼ リソースフォルトモニターを無効にする	51
	▼ リソースフォルトモニターを有効にする	52
	リソースタイプの削除	53
	▼ リソースタイプを削除する	53
	リソースグループの削除	54
	▼ リソースグループを削除する	54
	リソースの削除	55
	▼ リソースを削除する	56
	リソースグループの稼働系の切り替え	56
	▼ リソースグループの稼働系を切り替える	57
	リソースの無効化とリソースグループの UNMANAGED 状態への移行	58
	▼ リソースを無効にしてリソースグループを UNMANAGED 状態に移行する	58
	リソースタイプ、リソースグループ、リソース構成情報の表示	60
	リソースタイプ、リソースグループ、リソース構成情報の表示	60
	リソースタイプ、リソースグループ、リソースプロパティの変更	61

▼ リソースタイププロパティを変更する	61
▼ リソースグループプロパティを変更する	62
▼ リソースプロパティを変更する	63
リソースの STOP_FAILED エラーフラグの消去	64
▼ リソースの STOP_FAILED エラーフラグを消去する	65
登録済みのリソースタイプの再登録	66
▼ 登録済みのリソースタイプを再登録する	66
リソースグループへのノードの追加と削除	67
リソースグループにノードを追加する	68
リソースグループからノードを削除する	70
リソースグループとディスクデバイスグループ間での起動の同期	75
▼ 新しいリソース用に HASTorage リソースタイプを設定する	76
▼ 既存のリソース用に HASTorage リソースタイプを設定する	78
HASTorage から HASToragePlus へのアップグレード	78
デバイスグループまたは CFS を使用している場合に HASTorage から HASToragePlus へアップグレードする	79
CFS による HASTorage からフェイルオーバーファイルシステムによる HASToragePlus へアップグレードする	80
HA ローカルファイルシステムの有効化	81
▼ HASToragePlus リソースタイプを設定する	82
プライオリティが低いリソースグループをオフロードすることによるノードリソースの解放	84
▼ RGOffload リソースを設定する	84
RGOffload 拡張プロパティを構成する	87
フォルトモニター	88
A 標準プロパティ	91
リソースタイププロパティ	91
リソースプロパティ	98
リソースグループプロパティ	111
リソースプロパティの属性	118
B 有効な RGM 名と値	121
有効な RGM 名	121
RGM の値	122

C	データサービス構成のワークシートと記入例	123
	構成のワークシート	123
	リソースタイプのワークシート	124
	ネットワークリソースのワークシート	126
	アプリケーションリソース — フェイルオーバーワークシート	128
	アプリケーションリソース — スケーラブルのワークシート	130
	リソースグループ — フェイルオーバーのワークシート	132
	リソースグループ — スケーラブルのワークシート	134
	索引	137

はじめに

Sun Cluster データサービスの計画と管理 (*Solaris OS* 版) では、SPARC® および x86 ベースのシステムで Sun™ Cluster データサービスをインストールし、構成する方法について解説します。

注 - このマニュアルでは、“x86” とは Intel 32 ビットファミリのマイクロプロセッサ、および AMD 製の互換マイクロプロセッサチップを表わします。

このマニュアルは、Sun のソフトウェアとハードウェアについて幅広い知識を持っている上級システム管理者を対象としています。販売活動のガイドとしては使用しないでください。このマニュアルを読む前に、システムの必要条件を確認し、適切な装置とソフトウェアを購入しておく必要があります。

このマニュアルで説明されている作業手順を行うには、Solaris™ オペレーティング環境に関する知識と、Sun Cluster と共に使用するボリューム管理ソフトウェアに関する専門知識が必要です。

注 - Sun Cluster ソフトウェアは、SPARC および x86 の 2 つのプラットフォーム上で動作します。このマニュアルの情報は、特別な章、節、注意、項目、図、表、または例で示さない限りは、両方のプラットフォームに関するものです。

UNIX コマンド

このマニュアルでは、Sun Cluster データサービスのインストールと構成に固有のコマンドについて説明します。このマニュアルでは、UNIX® の基本的なコマンドや手順 (システムの停止、システムのブート、デバイスの構成など) については扱いません。UNIX の基本的なコマンドや手順については、次の情報源を利用してください。

- Solaris オペレーティングシステムのオンラインドキュメント
- Solaris オペレーティングシステムのマニュアルページ
- システムに付属するその他のソフトウェアマニュアル

表記上の規則

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

表 P-1 表記上の規則

字体または記号	意味	例
AaBbCc123	コマンド名、ファイル名、ディレクトリ名、画面上のコンピュータ出力、コード例を示します。	.login ファイルを編集します。 ls -a を使用してすべてのファイルを表示します。 system%
AaBbCc123	ユーザーが入力する文字を、画面上のコンピュータ出力と区別して示します。	system% su password:
AaBbCc123	変数を示します。実際に使用する特定の名前または値で置き換えます。	ファイルを削除するには、rm <i>filename</i> と入力します。
『 』	参照する書名を示します。	『コードマネージャ・ユーザーズガイド』を参照してください。
「 」	参照する章、節、ボタンやメニュー名、強調する単語を示します。	第 5 章「衝突の回避」を参照してください。 この操作ができるのは、「スーパーユーザー」だけです。
\	枠で囲まれたコード例で、テキストがページ行幅を超える場合に、継続を示します。	sun% grep `^#define \ XV_VERSION_STRING`

コード例は次のように表示されます。

- C シェル

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

- C シェルのスーパーユーザー

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

- Bourne シェルおよび Korn シェル


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

- Bourne シェルおよび Korn シェルのスーパーユーザー

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

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

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

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

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

関連マニュアル

Sun Cluster の関連トピックについては、次の表に記載したマニュアルを参照してください。すべての Sun Cluster マニュアルは、<http://docs.sun.com> から入手できます。

トピック	関連文書
データサービス管理	『Sun Cluster データサービスの計画と管理 (Solaris OS 版)』 各データサービスのガイド
概念	『Sun Cluster の概念 (Solaris OS 版)』
概要	『Sun Cluster の概要 (Solaris OS 版)』
ソフトウェアのインストール	『Sun Cluster ソフトウェアのインストール (Solaris OS 版)』
システム管理	『Sun Cluster のシステム管理 (Solaris OS 版)』
ハードウェア管理	『Sun Cluster 3.x Hardware Administration Manual for Solaris OS』 各ハードウェア管理ガイド
データサービス開発	『Sun Cluster データサービス開発ガイド (Solaris OS 版)』
エラーメッセージ:	『Sun Cluster Error Messages Guide for Solaris OS』

トピック	関連文書
コマンドおよび機能のリファレンス	『Sun Cluster Reference Manual for Solaris OS』

Sun Cluster のマニュアルの完全なリストについては、<http://docs.sun.com> から、お使いの Sun Cluster のリリースのリリースノートを参照してください。

Sun のオンラインマニュアル

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

ヘルプ

Sun Cluster をインストールまたは使用しているときに問題が発生した場合は、ご購入先に連絡し、次の情報をお伝えください。

- 名前と電子メールアドレス (利用している場合)
- 会社名、住所、および電話番号
- ご使用のシステムのモデルとシリアル番号
- Solaris オペレーティングシステムのリリース番号 (例: Solaris 8)
- Sun Cluster のバージョン番号 (例: Sun Cluster 3.0)

サービスプロバイダのために、次のコマンドを使用して、システム上の各ノードに関する情報を収集してください。

コマンド	機能
<code>prtconf -v</code>	システムメモリのサイズと周辺デバイス情報を表示します
<code>psrinfo -v</code>	プロセッサの情報を表示します
<code>showrev -p</code>	インストールされているパッチを報告します
<code>SPARC: prtdiag -v</code>	システム診断情報を表示します

コマンド	機能
<code>scinstall -pv</code>	Sun Cluster のリリースおよびパッケージのバージョン情報を表示します

上記の情報にあわせて、`/var/adm/messages` ファイルの内容もご購入先にお知らせください。

第 1 章

Sun Cluster データサービスの計画

この章では、Sun Cluster データサービスのインストールと構成を計画するにあたってのガイドラインを説明します。この章の内容は次のとおりです。

- 13 ページの「Sun Cluster データサービス構成のガイドライン」
- 15 ページの「リソースグループとディスクデバイスグループの関係」
- 16 ページの「HAStorage と HAStoragePlus の概要」
- 18 ページの「考慮事項」
- 19 ページの「ノードリストプロパティ」
- 19 ページの「インストールと構成プロセスの概要」
- 21 ページの「データサービスリソースを管理するためのツール」
- 24 ページの「Sun Cluster データサービスフォルトモニター」

データサービス、リソースタイプ、リソース、およびリソースグループについての概念的な情報については、『Sun Cluster の概念 (Solaris OS 版)』のマニュアルを参照してください。

Sun Cluster データサービスとして現在提供されていないアプリケーションについては、『Sun Cluster データサービス開発ガイド (Solaris OS 版)』を参照してください。アプリケーションを高可用性データサービスとして構成する方法について説明されています。

Sun Cluster データサービス構成のガイドライン

この節では、Sun Cluster データサービスを構成するためのガイドラインを説明します。

データサービス固有の要件の確認

Solaris と Sun Cluster のインストールを開始する前に、すべてのデータサービスの要件を確認します。計画に不備があった場合、インストールエラーが発生し、Solaris や Sun Cluster ソフトウェアを完全にインストールし直す必要が生じる可能性もあります。

たとえば、Sun Cluster Support for Oracle Parallel Server/Real Application Clusters の Oracle Parallel Fail Safe/Real Application Clusters Guard オプションには、ユーザーがクラスタ内で使用するホスト名に関する特殊な要件があります。Sun Cluster HA for SAP にも特殊な要件があります。Sun Cluster ソフトウェアをインストールした後にホスト名は変更できないため、このような必要条件是 Sun Cluster ソフトウェアをインストールする前に調整しておく必要があります。Sun Cluster Support for Oracle Parallel Server/Real Application Clusters および Sun Cluster HA for SAP は、どちらも x86 ベースのクラスタではサポートされていないので注意してください。

アプリケーションバイナリの格納先の決定

アプリケーションソフトウェアおよびアプリケーション構成ファイルは、次のいずれかの場所にインストールできます。

- 各クラスタノードのローカルディスク - ソフトウェアと構成ファイルを個々のクラスタノードに配置すると、次のようなメリットが得られます。あとでアプリケーションを更新する場合に、サービスを停止することなく実施できます。

ただし、ソフトウェアや構成ファイルの複数のコピーが存在するため、保守や管理をするファイルが増えるという欠点があります。

- クラスタファイルシステム - アプリケーションバイナリをクラスタファイルシステムに格納した場合、保守や管理をするコピーが 1 つだけになります。しかし、アプリケーションソフトウェアをアップグレードするには、クラスタ全体でデータサービスを停止する必要があります。アップグレード時に多少の時間停止できるようであれば、アプリケーションおよび構成ファイルの 1 つのコピーをクラスタファイルシステムに格納することが可能です。

クラスタファイルシステムの作成方法については、『Sun Cluster ソフトウェアのインストール (Solaris OS 版)』の計画の章を参照してください。

- **HA** ローカルファイルシステム - HASToragePlus を使用すると、ローカルファイルシステムを Sun Cluster 環境に統合して、ローカルファイルシステムの可用性を高めることができます。HASToragePlus は、Sun Cluster でローカルファイルシステムのフェイルオーバーを行うための付加的なファイルシステム機能 (チェック、マウント、強制的なマウント解除など) も提供します。フェイルオーバーを行うには、アフィニティスイッチオーバーが有効なグローバルディスクグループ上にローカルファイルシステムが存在していなければなりません。

HASToragePlus リソースタイプを使用する方法については、各データサービスガイド、または 81 ページの「HA ローカルファイルシステムの有効化」を参照してください。

nsswitch.conf ファイルの内容の確認

nsswitch.conf ファイルは、ネームサービスの検索用の構成ファイルです。このファイルは次の情報を指定します。

- ネームサービスの検索に使用する Solaris 環境内のデータベース
- データベースの検索順序

一部のデータサービスについては、「group」検索の対象の先頭を「files」に変更してください。具体的には、nsswitch.conf ファイル内の「group」行を変更し、「files」エントリが最初にリストされるようにします。「group」行を変更するかどうかを判断するには、構成するデータサービスに関する章を参照してください。

Sun Cluster 環境で nsswitch.conf ファイルを構成する方法の追加情報については、『Sun Cluster ソフトウェアのインストール (Solaris OS 版)』の計画の章を参照してください。

クラスタファイルシステムの構成の計画

データサービスによっては、Sun Cluster の要件を満たす必要があります。特別な検討事項が必要かどうかを判断するには、そのデータサービスに関する章を参照してください。

リソースタイプ HASToragePlus を使用すると、フェイルオーバー用に構成された Sun Cluster 環境で HA ローカルファイルシステムを使用できます。HASToragePlus リソースタイプの設定方法については、81 ページの「HA ローカルファイルシステムの有効化」を参照してください。

クラスタファイルシステムの作成方法については、『Sun Cluster ソフトウェアのインストール (Solaris OS 版)』の計画の章を参照してください。

リソースグループとディスクデバイスグループの関係

Sun Cluster は、ディスクデバイスグループとリソースグループに関し、ノードリストという概念を持っています。ノードリストには、ディスクデバイスグループまたはリソースグループの潜在的マスターであるノードが順にリストされています。ダウンしていたノードがクラスタに再結合し、そのノードがノードリストで現在の稼働系より前に設定されているときの挙動を決定するために、Sun Cluster は「フェイルバックポリシー」を使用します。フェイルバックが True に設定されていると、デバイスグループまたはリソースグループが現在の稼働系から、再結合したノードに切り替えられ、このノードが新しい稼働系になります。

フェイルオーバーリソースグループの高可用性を保証するために、そのグループのノードリストと関連するディスクデバイスグループのノードリストとを一致するようにしてください。スケーラブルリソースグループの場合、そのリソースグループのノードリストは必ずしもデバイスグループのノードリストと一致するとは限りません。これは、现阶段では、デバイスグループのノードリストには2つのノードしか含むことができないためです。2ノードを超えるクラスタの場合は、スケーラブルリソースグループのノードリストに、3ノード以上を含むことができます。

たとえば、ノード `phys-schost-1` と `phys-schost-2` が含まれるノードリストを持つディスクデバイスグループ `disk-group-1` があり、フェイルバックポリシーが `Enabled` に設定されているとします。さらに、アプリケーションデータの保持に `disk-group-1` を使用する `resource-group-1` というフェイルオーバーリソースグループも持っているとします。このような場合は、`resource-group-1` を設定するときに、リソースグループのノードリストに `phys-schost-1` と `phys-schost-1` も指定し、フェイルバックポリシーを `True` に設定します。

スケーラブルリソースグループの高可用性を保証するために、そのスケーラブルサービスグループのノードリストをディスクデバイスグループのノードリストのスーパーセットにしてください。スーパーセットにすることで、ディスクに直接接続されるノードは、スケーラブルリソースグループを実行するノードになります。この利点は、データに接続されている少なくとも1つのクラスタノードがクラスタで起動されているときに、スケーラブルサービスも同様に利用できるようにスケーラブルリソースグループが同じノード上で実行できることです。

ディスクデバイスグループの設定方法の詳細は、『*Sun Cluster ソフトウェアのインストール (Solaris OS 版)*』を参照してください。ディスクデバイスグループとリソースグループの関係の詳細については、『*Sun Cluster の概念 (Solaris OS 版)*』のマニュアルを参照してください。

HASStorage と HASStoragePlus の概要

リソースタイプ HASStorage と HASStoragePlus は、次のオプションの設定に使用できます。

- ディスクデバイスリソースが利用可能になるまで、HASStorage または HASStoragePlus のリソースを含む同じリソースグループ内にあるほかのリソースの `START` メソッドを待機させることによって、ディスクデバイスとリソースグループの起動の順番を調整できます。
- `AffinityOn` を `True` に設定することで、リソースグループとディスクデバイスグループを同一ノード上におき、ディスクに負荷がかかることの多いデータサービスのパフォーマンスを向上させます。

HASStoragePlus は、マウントされていない状態であることが確認されたグローバルファイルシステムをマウントすることもできます。詳細については、15 ページの「クラスタファイルシステムの構成の計画」を参照してください。

注 - HASTorage または HASToragePlus リソースがオンラインの間にデバイスグループが別のノードに切り替えられた場合、AffinityOn の設定は影響しないので、リソースグループはデバイスグループと共に別のノードに移行することはありません。一方、リソースグループが別のノードに切り替わった場合、AffinityOn が True に設定されていると、デバイスグループはリソースグループと一緒に新しいノードに移動します。

ディスクデバイスグループとリソースグループ間の関係については、75 ページの「リソースグループとディスクデバイスグループ間での起動の同期」を参照してください。追加情報は、SUNW.HASTorage(5) および SUNW.HASToragePlus(5) のマニュアルページにあります。

VxFS などのファイルシステムをローカルモードでマウントする方法については、81 ページの「HA ローカルファイルシステムの有効化」を参照してください。詳細は、SUNW.HASToragePlus(5) のマニュアルページを参照してください。

データサービスが HASTorage または HASToragePlus を必要とするかどうかを確認する方法

- データサービスリソースグループがノードリストを持っており、その一部のノードがストレージに直接接続されていない場合は、リソースグループ内で HASTorage または HASToragePlus リソースを構成し、ほかのデータサービスリソースの依存性を HASTorage リソースまたは HASToragePlus リソースに設定する必要があります。この条件によって、ストレージとデータサービス間の起動の順番が調整されます。
- ディスクに負荷がかかるデータサービスの場合 (Sun Cluster HA for Oracle や Sun Cluster HA for NFS など)、次の作業を行う必要があります。
 - HASTorage リソースまたは HASToragePlus リソースをデータリソースグループに追加します。
 - HASTorage リソースまたは HASToragePlus リソースをオンラインに切り替えます。
 - データサービスリソースの依存性を HASTorage リソースまたは HASToragePlus リソースに設定します。
 - AffinityOn を True に設定します。これらの作業を行うと、リソースグループとディスクデバイスグループは同じノード上に配置されます。
- フェイルバック設定は、リソースグループとデバイスグループで同一にする必要があります。

- 起動時に全ファイルを読み込むようなデータサービスがディスクにさほど負荷をかけない場合 (Sun Cluster HA for DNS など) は、HASTorage または HASToragePlus リソースタイプの構成を省略することもできます。

HASTorage または HASToragePlus の選択

データサービスリソースグループ内で HASTorage リソースと HASToragePlus リソースのどちらを作成すべきかを決定するには、以下の基準を考慮してください。

- Sun Cluster 3.0 12/01 以前を使用している場合は、HASTorage を使用してください。
- Sun Cluster 3.0 5/02 または Sun Cluster 3.1 を使用する場合は、HASToragePlus を使用してください (フェイルオーバーが構成された Sun Cluster にローカルアクセスを構成するようにファイルシステムを統合するには、Sun Cluster 3.0 5/02 または Sun Cluster 3.1 にアップグレードし、HASToragePlus リソースタイプを使用してください。詳細については、15 ページの「クラスタファイルシステムの構成の計画」を参照してください)。

考慮事項

この節の情報は、データサービスのインストールまたは構成について計画する場合に利用してください。これらの情報に目を通すことで、ユーザーの決定がデータサービスのインストールと構成に及ぼす影響について理解できるでしょう。使用しているデータサービスの具体的な考慮事項については、そのデータサービスについて説明された『Sun Cluster データサービスの計画と管理 (Solaris OS 版)』内の章を参照してください。

- データサービスが入出力中心で、クラスタとして大規模のディスクで構成している場合にディスクに障害が発生すると、入出力サブシステムが再試行するので、アプリケーションは遅延を感じることがあります。入出力サブシステムが再試行し、障害から回復するまで、数分かかることもあります。この遅延によって、最終的にディスクが自分自身で回復したとしても、Sun Cluster がアプリケーションを別のノードにフェイルオーバーすることがあります。このような場合のフェイルオーバーを回避するには、データサービスのデフォルトの検証タイムアウト値を増やしてみてください。データサービスのタイムアウトについての詳細や、タイムアウト値を増やす方法については、ご購入先にお問い合わせください。
- よりよいパフォーマンスを保つために、ストレージに直結されたクラスタノードにデータサービスをインストールし、構成してください。

ノードリストプロパティ

データサービスを構成するときに、次の3つのノードリストを指定できます。

1. `installed_nodes` – データサービスのリソースタイプのプロパティ。このプロパティには、リソースタイプがインストールされ、実行が有効になるクラスタノード名の一覧が含まれます。
2. `nodelist` – リソースグループのプロパティ。優先順位に基づいて、グループをオンラインにできるクラスタノード名の一覧が含まれます。これらのノードは、リソースグループの稼働系もしくは待機系です。フェイルオーバーサービスについては、リソースグループノードリストを1つだけ設定します。スケーラブルサービスの場合は、2つのリソースグループを設定するため、ノードリストも2つ必要になります。一方のリソースグループとそのノードリストは、共有アドレスをホストするノードを識別します。このノードリストは、スケーラブルリソースが依存するフェイルオーバーリソースグループを構成します。もう一方のリソースグループとそのノードリストは、アプリケーションリソースをホストするノードを識別します。アプリケーションリソースは、共有アドレスに依存します。そのため、共有アドレスを含むリソースグループ用のノードリストは、アプリケーションリソース用のノードリストのスーパーセットでなければなりません。
3. `auxnodelist` – 共有アドレスリソースのプロパティ。このプロパティは、クラスタノードを識別する物理ノード ID の一覧です。このクラスタノードは共有アドレスをホストできますが、フェイルオーバー時に稼働系になることはありません。これらのノードは、リソースグループのノードリストで識別されるノードとは、相互に排他的な関係になります。このノードリストは、スケーラブルサービスにのみ適用されます。詳細は、`scrgadm(1M)` のマニュアルページを参照してください。

インストールと構成プロセスの概要

データサービスをインストールして構成するには、次の手順を使用します。

- パッケージが提供されているインストールメディアから、データサービスパッケージをインストールします。
 - Sun Java™ Enterprise System CD
 - Sun Java Enterprise System Accessory CD Volume 3
- アプリケーションがクラスタ環境で実行するようにインストールして構成する。
- データサービスが使用するリソースおよびリソースグループを構成する。データサービスを構成するときは、Resource Group Manager (RGM) によって管理される、リソースタイプ、リソース、リソースグループを指定します。これらの手順は、各データサービスに関する章で説明されています。

データサービスのインストールと構成を開始する前に『*Sun Cluster* ソフトウェアのインストール (*Solaris OS* 版)』を参照してください。このマニュアルには、データサービスソフトウェアパッケージのインストール方法と、ネットワークリソースが使用するインターネットプロトコルネットワークマルチパス (IP ネットワークマルチパス) グループを構成する方法を確認してください。

注 - 以下のデータサービスのインストールと構成には、SunPlex™ Manager を使用できます。Sun Cluster HA for Oracle、Sun Cluster HA for Sun Java System Web Server、Sun Cluster HA for Sun Java System Directory Server、Sun Cluster HA for Apache、Sun Cluster HA for DNS、および Sun Cluster HA for NFS。Sun Cluster HA for Oracle および Sun Cluster HA for Apache は、SPARC ベースのクラスタでのみサポートされているので注意してください。詳細は、SunPlex Manager のオンラインヘルプを参照してください。

インストールと構成の作業の流れ

表 1-1 に、Sun Cluster フェイルオーバーデータサービスのインストールおよび構成作業と、その手順が説明されている参照先を示します。

表 1-1 Task Map: Sun Cluster データサービスのインストールと構成

作業	参照箇所
Solaris と Sun Cluster ソフトウェアのインストール	『 <i>Sun Cluster</i> ソフトウェアのインストール (<i>Solaris OS</i> 版)』
IP ネットワークマルチパス グループの設定	『 <i>Sun Cluster</i> ソフトウェアのインストール (<i>Solaris OS</i> 版)』
多重ホストディスクの設定	『 <i>Sun Cluster</i> ソフトウェアのインストール (<i>Solaris OS</i> 版)』
リソースとリソースグループの計画	付録 C
アプリケーションバイナリの格納先の決定 (nsswitch.conf の構成)	14 ページの「アプリケーションバイナリの格納先の決定」 15 ページの「nsswitch.conf ファイルの内容の確認」
アプリケーションソフトウェアのインストールと構成	該当する Sun Cluster データサービスブック
データサービスソフトウェアパッケージのインストール	『 <i>Sun Cluster</i> ソフトウェアのインストール (<i>Solaris OS</i> 版)』または該当する Sun Cluster データサービスブック
データサービスの登録と構成	該当する Sun Cluster データサービスブック

SPARC: 例

この節では、例として高可用性フェイルオーバーデータサービスとして設定されている Oracle アプリケーション用に、リソースタイプ、リソース、リソースグループを設定する方法を紹介します。

この例とスケーラブルデータサービスの例との主な違いは、ネットワークリソースを含むフェイルオーバーリソースグループに加えて、スケーラブルデータサービスはアプリケーションリソース用に別のリソースグループ(スケーラブルリソースグループのこと)を必要とすることです。

Oracle アプリケーションは、サーバーとリスナーの2つのコンポーネントを持ちます。Sun は Sun Cluster HA for Oracle データサービスを提供しているため、これらのコンポーネントはすでに Sun Cluster リソースタイプに対応付けられています。これら両方のリソースタイプが、リソースとリソースグループに関連付けられます。

この例は、フェイルオーバーデータサービスの例なので、論理ホスト名ネットワークリソースを使用し、稼動系から待機系にフェイルオーバーする IP アドレスを使用します。フェイルオーバーリソースグループに論理ホスト名リソースを入れ、Oracle サーバーリソースとリスナーリソースを同じリソースグループに入れます。この順に入れることで、フェイルオーバーを行うすべてのリソースが1つのグループになります。

Sun Cluster HA for Oracle をクラスタ上で実行するには、次のオブジェクトを定義する必要があります。

- LogicalHostname リソースタイプ – このリソースタイプは組み込まれているため、明示的に登録する必要はありません。
- Oracle リソースタイプ – Sun Cluster HA for Oracle は、2つの Oracle リソースタイプ(データベースサーバーとリスナー)を定義します。
- 論理ホスト名リソース – これらのリソースは、ノードで障害が発生した場合にフェイルオーバーする IP アドレスをホストします。
- Oracle リソース – Sun Cluster HA for Oracle—用に2つのリソースインスタンス(サーバーとリスナー)を指定する必要があります。
- フェイルオーバーリソースグループ – 1つのグループでフェイルオーバーを行う、Oracle サーバーとリスナー、および論理ホスト名リソースで構成されています。

データサービスリソースを管理するためのツール

この節では、インストールや構成の作業に使用するツールについて説明します。

SunPlex Manager グラフィカルユーザーインタフェース (GUI)

SunPlex Manager は、次の作業を実行できる Web ベースのツールです。

- クラスタのインストール
- クラスタの管理
- リソースやリソースグループの作成と構成
- Sun Cluster ソフトウェアを使ったデータサービスの構成

SunPlex Manager を使用し、クラスタソフトウェアをインストールする方法については、『*Sun Cluster* ソフトウェアのインストール (Solaris OS 版)』を参照してください。管理作業については、SunPlex Manager のオンラインヘルプを参照してください。

SPARC: Sun Management Center GUI 用の Sun Cluster モジュール

Sun Cluster モジュールを使用すると、クラスタの監視やリソースおよびリソースグループに対する処理の一部を Sun Management Center GUI から行えます。Sun Cluster モジュールのインストール要件と手順については、『*Sun Cluster* ソフトウェアのインストール (Solaris OS 版)』を参照してください。Sun Management Center の詳細は、<http://docs.sun.com> にある Sun Management Center ソフトウェアのマニュアルを参照してください。

scsetup ユーティリティ

scsetup(1M) ユーティリティは、メニュー主導型のインタフェースで、Sun Cluster の一般的な管理に使用できます。このユーティリティは、さらに、データサービスのリソースやリソースグループの構成にも使用できます。この場合には、scsetup のメインメニューからオプション 2 を選択して、「Resource Group Manager」というサブメニューを起動してください。

scrgadm コマンド

scrgadm コマンドにより、データサービスリソースの登録や構成を行うことができます。データサービスの登録と構成の方法については、データサービスのブックを参照してください。たとえば、Sun Cluster HA for Oracle を使用する場合は、『*Sun Cluster Data Service for Oracle ガイド (Solaris OS 版)*』の手順を参照してください。第 2 章にも、scrgadm コマンドを使ってデータサービスリソースを管理する方法が記載されています。最後に、追加情報については、scrgadm(1M) のマニュアルページを参照してください。

データサービスリソースの管理作業

次の表に、データサービスリソースの管理作業に使用できるツール (コマンド行以外の) を示します。これらの作業の詳細や、関連する手順をコマンド行から行う方法については、第 2 章を参照してください。

表 1-2 データサービスリソースの管理作業に使用できるツール

作業	SunPlex Manager	SPARC: Sun Management Center	scsetup ユーティリティ
リソースタイプを登録する	可	不可	可
リソースグループを作成	可	不可	可
リソースグループへソースを追加する	可	不可	可
リソースグループをオンラインにする	可	可	不可
リソースグループを削除	可	可	不可
リソースを削除する	可	可	不可
リソースグループの現在の稼動系を切り替える	可	不可	不可
リソースを使用不可にする	可	可	不可
無効なリソースのリソースグループを非管理状態にする	可	不可	不可
リソースタイプ、リソースグループ、リソース構成情報を表示する	可	可	不可
リソースプロパティを変更する	可	不可	不可
リソースの STOP_FAILED エラーフラグを消去する	可	不可	不可
ノードをリソースグループに追加する	可	不可	不可

Sun Cluster データサービスフォルトモニター

この節では、データサービスフォルトモニターの一般的な事項について説明します。Sun が提供するデータサービスには、パッケージに組み込まれているフォルトモニターがあります。フォルトモニター(または障害検証機能)は、データサービスの状態を検証するプロセスです。

フォルトモニターの呼び出し

フォルトモニターは、リソースグループとそのリソースをオンラインにしたときに、RGM によって呼び出されます。この呼び出しによって、RGM はそのデータサービスの `MONITOR_START` メソッドの呼び出しを内部で行います。

フォルトモニターは、次の 2 つの機能を実行します。

- データサービスのサーバプロセスの異常終了またはプロセスの監視
- データサービスの健全性の検査

サーバプロセスの異常終了の監視

プロセスモニタファシリティ(PMF)は、データサービスプロセスを監視します。

データサービスの障害検証は、無限ループで実行され、`Thorough_probe_interval` リソースプロパティによって設定された調整可能な期間に休止状態(スリープ)になります。休止している間に、検証機能はプロセスが終了したかどうかについて PMF により検査します。サーバプロセスが終了した場合は、その後、検証機能はデータサービスの状態を「Service daemon not running」で更新し、操作を実行します。実行する操作には、データサービスをローカルで再起動する、またはセカンダリクラスタノードにデータサービスをフェイルオーバーするなどが含まれます。検証機能は、そのデータサービスアプリケーションリソースの `Retry_count` および `Retry_interval` リソースプロパティで設定されている値を調べ、データサービスを再起動するか、フェイルオーバーするかどうかを決定します。

データサービスの健全性の検査

通常、検証機能とデータサービスとの間の通信は、専用のコマンドまたは指定したデータサービスポートとの正常な接続によって行われます。

検証機能は主に以下のことを行います。

1. 休止します (Thorough_probe_interval)。
2. タイムアウトプロパティ Probe_timeout でヘルスチェック (健全性の検査) を実行します。Probe_timeoutは、ユーザーが設定可能な各データサービスのリソース拡張プロパティです。
3. 手順2が成功、すなわちサービスが健全であれば、成功/失敗の履歴を更新します。Retry_interval リソースプロパティに設定されている値よりも古い履歴を消去 (パージ) することで、成功/失敗の履歴を更新します。検証機能は、リソースの状態メッセージを「Service is online」に設定し、手順1に戻ります。

手順2が失敗に終了したならば、サービスの状態に異常があれば、検証機能は失敗履歴を更新します。その後、ヘルスチェックに失敗した総数を計算します。

ヘルスチェックの結果は、致命的な失敗から成功までの範囲があります。結果の判断は、個々のデータサービスに依存します。たとえば、検証機能が正常にサーバーに接続でき、ハンドシェイクメッセージをサーバーに送信できるにも関わらず、この検証機能がタイムアウト前に一部の応答しか受け取ることができない場合を考えてみます。この場合、最も可能性のある原因はシステムの過負荷です。サービスの再起動など、操作を何か実行すると、クライアントはそのサービスに接続するため、さらにシステムの負荷が増大します。このような場合に、データサービスのフォルトモニターが、この「一部」の失敗を致命的なものとして扱わないようにします。代わりに、モニターは、サービスの致命的ではない検証としてこの失敗を追跡します。これらの一部の失敗は、Retry_interval プロパティによって指定された期間、累積されます。

ただし、検証機能がまったくサーバーに接続できない場合は、致命的な失敗であると認識されます。一部の失敗が、断片的な量によって失敗カウンターの増加につながります。致命的な失敗、または一部の失敗の累積のいずれかによって、失敗カウンターが合計カウンターに到達するたびに、検証機能はデータサービスの再起動またはフェイルオーバーによってこの状況を修正しようとします。

4. 手順3 (履歴期間内での異常の数)での計算の結果、Retry_count リソースプロパティの値よりも少ない場合は、検証機能は、状況をローカルで修正しようとします (たとえば、サービスの再起動)。検証機能は、リソースの状態メッセージを「Service is degraded」に設定し、手順1に戻ります。
 5. Retry_interval で指定した期間内で発生した失敗の数が Retry_count の値を超える場合、検証機能は、scha_control を「giveover」オプション付きで呼び出します。このオプションは、サービスのフェイルオーバーを要求します。この要求によって失敗が修正されると、このノードでのフォルトモニターが停止されます。検証機能は、リソースの状態メッセージを「Service has failed」に設定します。
 6. さまざまな理由により、前の手順で発行された scha_control 要求が Sun Cluster によって拒否されることがあります。この理由は、scha_control の戻り値で識別できます。検証機能は、戻り値を調べます。scha_control が拒否される場合、検証機能は失敗/成功履歴をリセットし、新たに開始します。検証機能が履歴をリセットするのは、失敗の数がすでに Retry_count を超えているため、フォルトモニターが各後続の繰り返しで scha_control を発行しようとするためです (ただし、再び拒否されます)。この要求は、システムの負荷をさらに高め、サービス障害がさらに生じる可能性が増大します。
- その後、検証機能は、手順1に戻ります。

第 2 章

データサービスリソースの管理

この章では、`scrgadm(1M)` コマンドを使って、リソースや、リソースグループ、リソースタイプを管理する手順を説明します。手順を実行するその他のツールについては、21 ページの「データサービスリソースを管理するためのツール」を参照してください。

この章では、次の手順について説明します。

- 31 ページの「リソースタイプを登録する」
- 32 ページの「アップグレードされたリソースタイプをインストールして登録する」
- 33 ページの「既存のリソースを新バージョンのリソースタイプに移行する」
- 39 ページの「フェイルオーバーリソースグループを作成する」
- 40 ページの「スケラブルリソースグループを作成する」
- 42 ページの「論理ホスト名リソースをリソースグループに追加する」
- 44 ページの「共有アドレスリソースをリソースグループに追加する」
- 46 ページの「フェイルオーバーアプリケーションリソースをリソースグループに追加する」
- 48 ページの「スケラブルアプリケーションリソースをリソースグループに追加する」
- 50 ページの「リソースグループをオンラインにする」
- 51 ページの「リソースフォルトモニターを無効にする」
- 52 ページの「リソースフォルトモニターを有効にする」
- 53 ページの「リソースタイプを削除する」
- 54 ページの「リソースグループを削除する」
- 56 ページの「リソースを削除する」
- 57 ページの「リソースグループの稼働系を切り替える」
- 58 ページの「リソースを無効にしてリソースグループを UNMANAGED 状態に移行する」
- 61 ページの「リソースタイププロパティを変更する」
- 62 ページの「リソースグループプロパティを変更する」
- 63 ページの「リソースプロパティを変更する」
- 65 ページの「リソースの STOP_FAILED エラーフラグを消去する」
- 66 ページの「登録済みのリソースタイプを再登録する」
- 76 ページの「新しいリソース用に HAStorage リソースタイプを設定する」

- 78 ページの「既存のリソース用に HAStorage リソースタイプを設定する」
- 82 ページの「HAStoragePlus リソースタイプを設定する」
- 84 ページの「RGOffload リソースを設定する」

リソースタイプ、リソースグループ、およびリソースに関する概要情報については、第 1 章 および『Sun Cluster の概念 (Solaris OS 版)』マニュアルを参照してください。

データサービスリソースの管理

表 2-1 に、データサービスリソースの管理作業を説明している節を示します。

表 2-1 Task Map: データサービス管理

作業	参照箇所
リソースタイプを登録する	31 ページの「リソースタイプを登録する」
リソースタイプをアップグレードする	33 ページの「既存のリソースを新バージョンのリソースタイプに移行する」 32 ページの「アップグレードされたリソースタイプをインストールして登録する」
フェイルオーバーリソースグループまたはスケーラブルリソースグループの作成	39 ページの「フェイルオーバーリソースグループを作成する」 40 ページの「スケーラブルリソースグループを作成する」
論理ホスト名または共有アドレス、データサービスリソースをリソースグループに追加する	42 ページの「論理ホスト名リソースをリソースグループに追加する」 44 ページの「共有アドレスリソースをリソースグループに追加する」 46 ページの「フェイルオーバーアプリケーションリソースをリソースグループに追加する」 48 ページの「スケーラブルアプリケーションリソースをリソースグループに追加する」
リソースとリソースモニターを有効にし、リソースグループを管理し、リソースグループおよび関連するリソースをオンラインにする	50 ページの「リソースグループをオンラインにする」

表 2-1 Task Map: データサービス管理 (続き)

作業	参照箇所
リソース自体とは関係なく、リソースモニターだけを無効または有効にする	51 ページの「リソースフォルトモニターを無効にする」 52 ページの「リソースフォルトモニターを有効にする」
クラスタからリソースタイプを削除する	53 ページの「リソースタイプを削除する」
クラスタからリソースグループを削除する	54 ページの「リソースグループを削除する」
リソースグループからリソースを削除する	56 ページの「リソースを削除する」
リソースグループの稼働系を切り替える	57 ページの「リソースグループの稼働系を切り替える」
リソースを無効にし、そのリソースグループをUNMANAGEDに移行する	58 ページの「リソースを無効にしてリソースグループを UNMANAGED 状態に移行する」
リソースタイプ、リソースグループ、リソース構成情報を表示する	60 ページの「リソースタイプ、リソースグループ、リソース構成情報の表示」
リソースタイプ、リソースグループ、リソースプロパティの変更	61 ページの「リソースタイププロパティを変更する」 62 ページの「リソースグループプロパティを変更する」 63 ページの「リソースプロパティを変更する」
失敗した Resource Group Manager (RGM) プロセスのエラーフラグの消去	65 ページの「リソースの STOP_FAILED エラーフラグを消去する」
組み込みリソースタイプ LogicalHostname および SharedAddress の再登録	66 ページの「登録済みのリソースタイプを再登録する」
ネットワークリソースのネットワークインタフェース ID リストの更新と、リソースグループのノードリストの更新	68 ページの「リソースグループにノードを追加する」
リソースグループからノードを削除する	70 ページの「リソースグループからノードを削除する」
リソースグループとディスクデバイスグループ間で起動の同期をとるために、リソースグループの HAStorage または HAStoragePlus を設定する	76 ページの「新しいリソース用に HAStorage リソースタイプを設定する」

表 2-1 Task Map: データサービス管理 (続き)

作業	参照箇所
ディスク入出力負荷が高いフェイルオーバーデータサービスに対応するように、HASStoragePlus を設定してローカルファイルシステムの可用性を高める	82 ページの「HASStoragePlus リソースタイプを設定する」
重要なデータサービスのためにノードを自動的に開放するようにリソースタイプを設定する	84 ページの「RGOffload リソースを設定する」

注 - この章では、`scrgadm(1M)` コマンドを使用し、これらの作業を完了する手順について解説します。これ以外のツールを使ってリソースを管理することもできます。これらの方法については、21 ページの「データサービスリソースを管理するためのツール」を参照してください。

Sun Cluster データサービスの構成と管理

Sun Cluster の構成は、複数の手順から成る単一の作業です。これらの手順により次の作業を実行できます。

- リソースタイプの登録
- リソースタイプのアップグレード
- リソースグループの作成
- リソースグループへのリソースの追加
- リソースをオンラインにする

データサービスの構成を変更するには、初期構成が終わった後で次の各手順を使用します。たとえば、リソースタイプ、リソースグループ、およびリソースプロパティを変更する場合は、61 ページの「リソースタイプ、リソースグループ、リソースプロパティの変更」へ進んでください。

リソースタイプの登録

リソースタイプは、指定されたタイプのすべてのリソースに適用される共通のプロパティとコールバックメソッドの仕様を提供します。リソースタイプは、そのタイプのリソースを作成する前に登録する必要があります。リソースタイプについての詳細は、第1章を参照してください。

▼ リソースタイプを登録する

この手順を実行するには、登録するリソースタイプに、データサービス名の略語で名前をつける必要があります。Sun Cluster に標準添付されているデータサービスのリソースタイプ名の詳細は、Sun Cluster のリリースノートを参照してください。

追加情報については、scrgadm(1M) のマニュアルページを参照してください。

注 - この手順は、任意のクラスタノードから実行します。

1. クラスタメンバー上でスーパーユーザーになります。
2. リソースタイプを登録します。

```
# scrgadm -a -t resource-type
-a                指定したリソースタイプを追加します。
-t resource-type  追加するリソースタイプの名前を指定します。指定する事前定義済みの名前を判別するには、Sun Cluster のリリースノートを参照してください。
```

3. 登録されたリソースタイプを確認します。

```
# scrgadm -pv -t resource-type
```

例 - リソースタイプの登録

次に、Sun Cluster HA for Sun Java System Web Server (内部名 iws) を登録する例を示します。

```
# scrgadm -a -t SUNW.iws
# scrgadm -pv -t SUNW.iws
リソースタイプ 名前:                SUNW.iws :4
(SUNW.iws) リソースタイプ 説明:          None registered
(SUNW.iws) リソースタイプ ベースディレクトリ: /opt/SUNWschtt/bin
(SUNW.iws) リソースタイプ 単一のインスタンス: False
```

(SUNW.iws) リソースタイプ 初期ノード:	All potential masters
(SUNW.iws) リソースタイプ フェイルオーバー:	False
(SUNW.iws) リソースタイプ バージョン:	4
(SUNW.iws) リソースタイプ API バージョン:	2
(SUNW.iws) リソースタイプ ノードにインストールされている:	All
(SUNW.iws) リソースタイプ パッケージ:	SUNWschtt
(SUNW.iws) リソースタイプ システム:	False

次に進む手順

リソースタイプを登録したあと、リソースグループを作成し、リソースをそのリソースグループに追加できます。詳細は、38 ページの「リソースグループの作成」を参照してください。

リソースタイプのアップグレード

新バージョンのリソースタイプがリリースされる際には、そのアップグレードされたリソースタイプをインストールして登録できます。また、既存のリソースを新しいリソースタイプバージョンにアップグレードすることも可能です。この節では、次の 2 つの作業、アップグレードされたリソースタイプをインストールして登録する方法と、既存のリソースを新しいリソースタイプバージョンにアップグレードする方法について説明します。

- 32 ページの「アップグレードされたリソースタイプをインストールして登録する」
- 33 ページの「既存のリソースを新バージョンのリソースタイプに移行する」

▼ アップグレードされたリソースタイプをインストールして登録する

この作業は、`scsetup` の「リソースグループ」オプションを使用しても行えます。`scsetup` の詳細は、`scsetup(1M)` のマニュアルページを参照してください。

1. すべてのクラスタノードに、リソースタイプアップグレードパッケージをインストールします。

注 - リソースタイプパッケージがすべてのノードにインストールされない場合は、作業が別途必要です (手順 3)。

リソースタイプのアップグレードパッケージをインストールする際にノードを非クラスタモードで起動する必要があるかどうかは、アップグレードドキュメントに示されています。ダウンタイムを避けるには、パッケージをインストールするノードを非クラスタモード、残りのノードをクラスタモードに設定した状態で、一度に1台のノードに限定してローリングアップグレード方式で新しいパッケージを追加します。

2. この新しいリソースタイプバージョンを登録します。

```
scrgadm -a -t resource_type -f path_to_new_RTR_file
```

新しいリソースタイプの名前は次の形式をとります。

```
vendor_id.raname:version
```

登録した新しいリソースタイプを表示するには、`scrgadm -p` または `scrgadm -pv` (詳細表示) を使用してください。

3. 新しいリソースタイプをインストールしないノードがある場合は、実際にリソースタイプをインストールしたノードを **Installed_nodes** プロパティに設定します。

```
scrgadm -c -t resource_type -h installed_node_list
```

新バージョンのリソースタイプは、次の点で旧バージョンと異なっている可能性があります。

- リソースタイププロパティの設定
- 標準プロパティ、拡張プロパティを含む宣言済みリソースプロパティ
- リソースプロパティの属性 (default、min、max、arraymin、arraymax、または tunability)
- 宣言済みメソッド
- メソッドやモニターの実装

▼ 既存のリソースを新バージョンのリソースタイプに移行する

この作業は、`scsetup` の「リソースグループ」オプションを使用しても行えます。`scsetup` の詳細は、`scsetup(1M)` のマニュアルページを参照してください。

新しいバージョンタイプに移行する方法は、既存のリソースタイプバージョンと、新バージョンにおける変更によって決まります。移行が可能かどうかは、リソースタイプのアップグレードドキュメントに記載されています。移行がサポートされていない場合は、リソースを削除してアップグレードされた新しいリソースに交換するか、あるいはそのリソースを古いリソースタイプバージョンのままにするかを検討してください。

既存のリソースを移行する場合は、以下の値が変化する可能性があります。

デフォルトのプロパティ値

アップグレードされたリソースタイプバージョンがデフォルトプロパティに新しいデフォルト値を宣言している場合は、既存のリソースはこの新しいデフォルト値を継承します。

既存のプロパティ設定が適切かどうかは、新しいリソースタイプバージョンの VALIDATE メソッドによってチェックされます。この設定が不適切な場合は、プロパティを編集して適切な値に変更してください。プロパティの編集方法は、手順 3 を参照してください。

リソースタイプ名

RTR ファイルには、リソースタイプの完全修飾名の形成に使用される以下のプロパティが含まれます。

- Vendor_id
- Resource_type
- RT_Version

アップグレードされたリソースタイプバージョンは、その登録時に `vendor_id.rtname:version` として保存されず、新バージョンに移行されたリソースには、上記のプロパティから構成される新しい Type プロパティが存在します。

Type_version リソースプロパティ

リソースのタイプの RT_Version プロパティは、標準のリソースプロパティ Type_version に格納されます。Type_Version プロパティは、RTR ファイルには現れません。次のコマンドを使用して Type_Version プロパティを編集してください。

```
scrgadm -c -j resource -y Type_version=new_version
```

1. 既存のリソースを新しいリソースタイプバージョンに移行する前に、新しいリソースタイプに付属しているアップグレードマニュアルに目を通し、移行が可能かどうかを確認してください。

そのマニュアルには、移行を実施すべきタイミングが記されています。

- 任意の時点 (Anytime)
- リソースが監視されていないとき
- リソースがオフラインのとき
- リソースが無効なとき
- リソースグループが管理されていないとき

移行がサポートされていない場合は、リソースを削除してアップグレードされた新しいリソースバージョンに置き換えるか、そのリソースを古いリソースタイプバージョンのままにしておく必要があります。

2. 移行するリソースタイプのリソースごとに、アップグレードマニュアルに記載されている方法でそのリソースグループのリソースの状態を適切な状態に変更してください。次に例を示します。

リソースの監視を解除する必要がある場合:

```
scswitch -M -n -j resource
```

リソースをオフラインにする必要がある場合:

```
scswitch -n -j resource
```

リソースを無効にする必要がある場合:

```
scswitch -n -j resource
```

リソースグループの管理を解除する必要がある場合:

```
scswitch -n -j resource_group
```

```
scswitch -F -g resource_group
```

```
scswitch -u -g resource_group
```

3. 移行するリソースタイプをもつリソースごとに、リソースを編集し、その **Type_version** プロパティを新バージョンに変更します。

```
scrgadm -c -j resource -y Type_version=new_version \  
-x extension_property=new_value -y extension_property=new_value
```

必要に応じ、**-x** または **-y** オプション (あるいはこの両方) を追加して同じコマンドを実行し、同じリソースのほかのプロパティを編集して適切な値に変更します。

4. 手順 2 で入力したコマンドを逆に指定することにより、リソースまたはリソースグループの前の状態に戻します。次に例を示します。

リソースを監視状態に戻す場合:

```
scswitch -M -e -j resource
```

リソースを有効な状態に戻す場合:

```
scswitch -e -j resource
```

リソースグループをオンラインの管理状態に戻す場合:

```
scswitch -o -g resource_group
```

```
scswitch -Z -g resource_group
```

例 1 – 既存のリソースを新しいリソースタイプバージョンに移行する

この例は、既存のリソースを新しいリソースタイプバージョンに移行する方法を示しています。新しいリソースタイプパッケージのメソッドは、新しいパスに配置されています。インストール時にメソッドは上書きされないため、アップグレードされたリソースタイプのインストールが完了するまでリソースを無効にする必要はありません。

この実例では、次のことを前提としています。

- 新しいリソースタイプバージョンは 2.0 である
- 移行を実行すべきタイミングは「when_offline (リソースがオフラインのとき)」である
- リソース名は「myresource」である
- リソースタイプ名は「myrt」である
- 新しい RTR ファイルは /opt/XYZmyrt/etc/XYZ.myrt に配備されている
- 移行の対象となるリソースに依存していない
- 所属しているリソースグループをオンラインの状態にしたまま、移行の対象となるリソースをオフラインに切り替えることができる

(ベンダーのディレクトリに従ってすべてのノード上で新しいパッケージをインストールする)

```
# scrgadm -a -t myrt -f /opt/XYZmyrt/etc/XYZ.myrt
# scswitch -n -j myresource
# scrgadm -c -j myresource -y Type_version=2.0
# scswitch -e -j myresource
```

例 2 – 既存のリソースを新しいリソースタイプバージョンに移行する

この例は、既存のリソースを新しいリソースタイプバージョンに移行する方法を示しています。新しいリソースタイプパッケージには、モニターと RTR ファイルしか含まれていません。モニターはインストール時に上書きされるため、アップグレードされたリソースタイプをインストールする前にリソースを無効にする必要があります。

この実例では、次のことを前提としています。

- 新しいリソースタイプバージョンは 2.0 である
- 移行を実行すべきタイミングは「when_unmonitored (リソースの監視が解除しているとき)」である
- リソース名は「myresource」である
- リソースタイプ名は「myrt」である
- 新しい RTR ファイルは /opt/XYZmyrt/etc/XYZ.myrt に配備されている

```
# scswitch -M -n -j myresource
(ベンダーのディレクトリに従って新しいパッケージをインストールする)
# scrgadm -a -t myrt -f /opt/XYZmyrt/etc/XYZ.myrt
# scrgadm -c -j myresource -y Type_version=2.0
# scswitch -M -e -j myresource
```

リソースタイプのダウングレード

▼ 古いリソースタイプバージョンにダウングレードする

リソースを古いリソースタイプバージョンにダウングレードできます。古いリソースタイプバージョンにダウングレードする場合は、新しいリソースタイプバージョンにアップグレードする場合よりも条件が厳しくなります。まず、リソースグループの管理を解除する必要があります。アップグレードが可能なリソースタイプバージョンにしかダウングレードできないということにも注意してください。アップグレードが可能なバージョンは `scrgadm -p` コマンドを使用して確認できます。アップグレードが可能なバージョンの場合、出力には接尾辞 `:version` が含まれます。

1. ダウングレードしたいリソースを含んでいるリソースグループをオフラインに切り替えます。

```
scswitch -F -g resource_group
```

2. ダウングレードしたいリソースと、そのリソースグループ内のすべてのリソースを無効にします。

```
scswitch -n -j resource_to_downgrade
scswitch -n -j resource1
scswitch -n -j resource2
scswitch -n -j resource3
...
```

注 - リソースの無効化は、依存性の高いもの (アプリケーションリソース) から開始し、もっとも依存性の低いもの (ネットワークアドレスリソース) で終了するように行なってください。

3. リソースグループの管理を解除します。

```
scswitch -u -g resource_group
```

4. ダウングレードしたいリソースタイプの古いバージョンがクラスタ内にまだ登録されているかどうか確認します。

- 登録されている場合は、次の手順に進みます。
- 登録されていない場合は、希望する旧バージョンを登録し直します。

```
scrgadm -a -t resource_type_name
```

5. 旧バージョンを **Type_version** に指定し、リソースをダウングレードします。

```
scrgadm -c -j resource_to_downgrade -y Type_version=old_version
```

必要に応じて、同じコマンドを使って、同じリソースのその他のプロパティに適切な値を設定します。

6. ダウングレードしたリソースを含んでいるリソースグループを管理状態に切り替え、すべてのリソースを有効にしたあと、このグループをオンラインに切り替えます。

```
scswitch -Z -g resource_group
```

リソースグループの作成

リソースグループには、一連のリソースが含まれており、これらすべてのリソースは指定のノードまたはノード群で共にオンラインまたはオフラインになります。リソースを配置する前に、空のリソースグループを作成する必要があります。

リソースグループには、フェイルオーバーとスケラブルの2つの種類があります。フェイルオーバーリソースグループの場合、同時にオンラインにできるのは1つのノードでのみです。一方、スケラブルリソースグループの場合、同時に複数のノードでオンラインにできます。

以下の手順では、`scrgadm(1M)` コマンドを使用し、データサービスを登録、構成する方法について解説します。

リソースグループに関する概念情報については、「第1章」および『*Sun Cluster の概念 (Solaris OS 版)*』マニュアルを参照してください。

▼ フェイルオーバーリソースグループを作成する

フェイルオーバーリソースグループは、ネットワークアドレス (組み込みリソースタイプの LogicalHostname や SharedAddress など) と、フェイルオーバーリソース (フェイルオーバーデータサービスのためのデータサービスアプリケーションリソースなど) を含みます。ネットワークリソースは、データサービスがフェイルオーバーまたはスイッチオーバーする場合に、依存するデータサービスリソースと共に、クラスタノード間を移動します。

追加情報については、scrgadm(1M) のマニュアルページを参照してください。

注 - この手順は、任意のクラスタノードから実行します。

1. クラスタメンバー上でスーパーユーザーになります。
2. フェイルオーバーリソースグループを作成します。

```
# scrgadm -a -g resource-group [-h nodelist]
-a                指定したリソースグループを追加します。
-gresource-group  追加するフェイルオーバーリソースグループの名前を指定し
                  ます。任意の名前の先頭文字は ASCII にする必要がありま
                  ず。
-h nodelist       このリソースグループをマスターできるノードの順位リスト
                  を指定します (省略可能)。このリストを指定しない場合は、
                  デフォルトでクラスタ内のすべてのノードになります。
```

3. リソースグループが作成されていることを確認します。

```
# scrgadm -pv -g resource-group
```

例 - フェイルオーバーリソースグループの作成

次に、2つのノード (phys-schost-1, phys-schost-2) でマスターできるフェイルオーバーリソースグループ (resource-group-1) を追加する例を示します。

```
# scrgadm -a -g resource-group-1 -h phys-schost1,phys-schost-2
# scrgadm -pv -g resource-group-1
リソースグループ 名前:                resource-group-1
(resource-group-1) リソースグループ RG_description:    <NULL>
(resource-group-1) リソースグループ management state:  Unmanaged
(resource-group-1) リソースグループ Failback:         False
(resource-group-1) リソースグループ Nodelist:         phys-schost-1
                                                         phys-schost-2
(resource-group-1) リソースグループ Maximum primaries: 1
(resource-group-1) リソースグループ Desired primaries: 1
```

```
(resource-group-1) リソースグループ RG_dependencies: <NULL>
(resource-group-1) リソースグループ mode: Failover
(resource-group-1) リソースグループ network_dependencies: True
(resource-group-1) リソースグループ Global_resources_used: All
(resource-group-1) リソースグループ Pathprefix:
(resource-group-1) リソースグループ System: False
```

次に進む手順

フェイルオーバーリソースグループを作成した後で、そのリソースグループにアプリケーションリソースを追加できます。手順については、42 ページの「リソースグループへのリソースの追加」を参照してください。

▼ スケーラブルリソースグループを作成する

スケーラブルリソースグループは、スケーラブルサービスと共に使用されます。共有アドレス機能は、スケーラブルサービスの多数のインスタンスを1つのサービスとして扱える Sun Cluster のネットワーキング機能です。まず、スケーラブルリソースが依存する共有アドレスを含むフェイルオーバーリソースグループを作成しなければなりません。次にスケーラブルリソースグループを作成し、そのグループにスケーラブルリソースを追加します。

追加情報については、`scrgadm(1M)` のマニュアルページを参照してください。

注 - この手順は、任意のクラスタノードから実行します。

1. クラスタメンバー上でスーパーユーザーになります。
2. スケーラブルリソースが使用する共有アドレスを保持するフェイルオーバーリソースグループを作成します。
3. スケーラブルリソースグループを作成します。

```
# scrgadm -a -g resource-group \  
-y Maximum primaries=m \  
-y Desired primaries=n \  
-y RG_dependencies=depend-resource-group \  
[-h nodelist]
```

```
-a  
スケーラブルリソースグループを追加します。
```

```
-g resource-group  
追加するスケーラブルリソースグループの名前を指定します。
```



```
-y Maximum primaries [ ] =m
このリソースグループのアクティブな稼働系の最大数を指定します。

-y Desired primaries [ ] =n
リソースグループが起動するアクティブな稼働系の数を指定します。

-y RG_dependencies [ ] =depend-resource-group
作成されるリソースグループが依存する共有アドレスリソースを含むリソース
グループを指定します。

-h nodelist
リソースグループを利用できるノードのリストを指定します(省略可能)。この
リストを指定しない場合は、デフォルトですべてのノードになります。
```

4. スケーラブルリソースグループが作成されていることを確認します。

```
# scrgadm -pv -g resource-group
```

例 – スケーラブルリソースグループの作成

次に、2つのノード(phys-schost-1、phys-schost-2)でホストされるスケラブルリソースグループ(resource-group-1)を追加する例を示します。スケラブルリソースグループは、共有アドレスを含むフェイルオーバーリソースグループ(resource-group-2)に依存します。

```
# scrgadm -a -g resource-group-1 \
-y Maximum primaries=2 \
-y Desired primaries=2 \
-y RG_dependencies=resource-group-2 \
-h phys-schost-1,phys-schost-2
# scrgadm -pv -g resource-group-1
リソースグループ 名前:                resource-group-1
(resource-group-1) リソースグループ RG_description:    <NULL>
(resource-group-1) リソースグループ management state:  Unmanaged
(resource-group-1) リソースグループ Failback:         False
(resource-group-1) リソースグループ Nodelist:         phys-schost-1
                                                    phys-schost-2
(resource-group-1) リソースグループ Maximum primaries: 2
(resource-group-1) リソースグループ Desired primaries: 2
(resource-group-1) リソースグループ RG_dependencies:   resource-group-2
(resource-group-1) リソースグループ mode:             Scalable
(resource-group-1) リソースグループ network dependencies: True
(resource-group-1) リソースグループ Global_resources_used: All
(resource-group-1) リソースグループ Pathprefix:
```

次に進む手順

スケラブルリソースグループを作成したあと、そのリソースグループにスケラブルアプリケーションリソースを追加できます。詳細は、48ページの「スケラブルアプリケーションリソースをリソースグループに追加する」を参照してください。

リソースグループへのリソースの追加

リソースは、リソースタイプをインスタンス化したものです。リソースは、RGMによって管理される前に、リソースグループに追加する必要があります。この節では、3種類のリソースタイプについて説明します。

- 論理ホスト名リソース。
- 共有アドレスリソース。
- データサービス (アプリケーション) リソース。

論理ホスト名リソースと共有アドレスリソースは、常にフェイルオーバーリソースグループに追加してください。フェイルオーバーデータサービス用のデータサービスリソースは、フェイルオーバーリソースグループに追加してください。フェイルオーバーリソースグループは、そのデータサービス用の論理ホスト名リソースとアプリケーションリソースの両方を含みます。スケーラブルリソースグループは、スケーラブルサービス用のアプリケーションリソースだけを含んでいます。スケーラブルサービスが依存する共有アドレスリソースは、別のフェイルオーバーリソースグループに存在する必要があります。データサービスをクラスタノード全体に渡ってスケールするには、スケーラブルアプリケーションリソースと共有アドレスリソース間の依存性を指定する必要があります。

リソースに関する詳細は、『*Sun Cluster の概念 (Solaris OS 版)*』マニュアルおよび第1章を参照してください。

▼ 論理ホスト名リソースをリソースグループに追加する

この手順を実行するには、次の情報が必要になります。

- リソースを追加するフェイルオーバーリソースグループの名前
- リソースグループに追加するホスト名

追加情報については、`scrgadm(1M)` のマニュアルページを参照してください。

注 - この手順は、任意のクラスタノードから実行します。

1. クラスタメンバー上でスーパーユーザーになります。
2. 論理ホスト名リソースをリソースグループに追加します。

```
# scrgadm -a -L [-j resource] -g resource-group -l hostnamelist, ... [-n netiflist]
-a                                論理ホスト名リソースを追加します。
```

-L	論理ホスト名リソースの形式を指定します。
-j <i>resource</i>	リソース名を指定します (省略可能)。このオプションを指定しない場合は、デフォルトで -l オプションで最初に指定したホスト名になります。
-g <i>resource-group</i>	リソースを配置するリソースグループの名前を指定します。
-l <i>hostnamelist, ...</i>	クライアントがリソースグループでサービスと通信する UNIX ホスト名 (論理ホスト名) をコマンドで区切って指定します。
-n <i>netiflist</i>	各ノード上の IP ネットワークマルチパスグループをコマンドで区切って指定します (省略可能)。netiflist の各要素は、netif@node の形式で指定する必要があります。netif は IP ネットワークマルチパスグループ名 (sc_ipmp0 など) として指定できます。ノードは、sc_ipmp0@1、sc_ipmp@phys-schost-1 などのノード名またはノード ID で特定できます。

注 - Sun Cluster では、現在、netif にアダプタ名を使用することはできません。

3. 論理ホスト名リソースが追加されていることを確認します。

```
# scrgadm -pv -j resource
```

リソースを追加すると、Sun Cluster ソフトウェアは、そのリソースの妥当性を検査します。妥当性が確認されると、そのリソースを有効にできるとともに、そのリソースグループを RGM の管理下に置くことが可能です。妥当性の検査に失敗すると、scrgadm コマンドはエラーメッセージを生成して終了します。妥当性の検査に失敗した場合は、エラーメッセージについて各ノード上の syslog を調べてください。メッセージは、妥当性の検査を実施したノードで表示されます。必ずしも scrgadm コマンドを実行したノードで表示されるわけではありません。

例 - 論理ホスト名リソースのリソースグループへの追加

次に、論理ホスト名リソース (resource-1) をリソースグループ (resource-group-1) に追加する例を示します。

```
# scrgadm -a -L -j resource-1 -g resource-group-1 -l schost-1
# scrgadm -pv -j resource-1
(resource-group-1) リソース 名前: resource-1
(resource-group-1:resource-1) リソース R_description:
(resource-group-1:resource-1) リソース リソースタイプ: SUNW.LogicalHostname
(resource-group-1:resource-1) リソース タイプのバージョン: 1.0
(resource-group-1:resource-1) リソース リソースグループ名: resource-group-1
(resource-group-1:resource-1) リソース リソースプロジェクト名: default
```

(resource-group-1:resource-1) リソース 有効: False
(resource-group-1:resource-1) リソース 有効なモニター: True

次に進む手順

論理ホスト名リソースを追加したあと、50 ページの「リソースグループをオンラインにする」の手順に従って、このリソースをオンラインにします。

▼ 共有アドレスリソースをリソースグループに追加する

この手順を実行するには、次の情報が必要になります。

- リソースを追加するリソースグループの名前。このグループは、前の手順で作成したフェイルオーバーリソースグループでなければなりません。
- リソースグループに追加するホスト名。

追加情報については、`scrgadm(1M)` のマニュアルページを参照してください。

注 - この手順は、任意のクラスタノードから実行します。

1. クラスタメンバー上でスーパーユーザーになります。
2. 共有アドレスリソースをリソースグループに追加します。

```
# scrgadm -a -S [-j resource] -g resource-group -l hostnamelist, ... \  
[-X auxnodelist] [-n netiflist]
```

<code>-a</code>	共有アドレスリソースを追加します。
<code>-S</code>	共有アドレスリソースの形式を指定します。
<code>-j resource</code>	リソース名を指定します (省略可能)。このオプションを指定しない場合は、デフォルトで <code>-l</code> オプションで最初に指定したホスト名になります。
<code>-gresource-group</code>	リソースグループの名前を指定します。
<code>-l hostnamelist, ...</code>	共有アドレスホスト名をコンマで区切って指定します。
<code>-X auxnodelist</code>	共有アドレスをホストできるクラスタノード (ただし、フェイルオーバー時に稼働系として使用されない) を識別する物理ノード名または ID をコンマで区切って指定します。これらのノードは、リソースグループのノードリストで潜在的マスターとして識別されるノードと相互に排他的です。

`-n netiflist`

各ノード上の IP ネットワークマルチパスグループをコンマで区切って指定します (省略可能)。 `netiflist` の各要素は、 `netif@node` の形式で指定する必要があります。 `netif` は IP ネットワークマルチパスグループ名 (`sc_ipmp0` など) として指定できます。 ノードは、 `sc_ipmp0@1`、 `sc_ipmp@phys-schost-1` などのノード名またはノード ID で特定できます。

注 – Sun Cluster では、現在、 `netif` にアダプタ名を使用することはできません。

- 共有アドレスリソースが追加され、妥当性が検査されていることを確認します。

```
# scrgadm -pv -j resource
```

リソースを追加すると、Sun Cluster ソフトウェアは、そのリソースの妥当性を検査します。妥当性が確認されると、そのリソースを有効にできるとともに、そのリソースグループを RGM の管理下に置くことが可能です。妥当性の検査に失敗すると、`scrgadm` コマンドはエラーメッセージを生成して終了します。妥当性の検査に失敗した場合は、エラーメッセージについて各ノード上の `syslog` を調べてください。メッセージは、妥当性の検査を実施したノードで表示されます。必ずしも `scrgadm` コマンドを実行したノードで表示されるわけではありません。

例 – 共有アドレスリソースのリソースグループへの追加

次に、共有アドレスリソース (`resource-1`) をリソースグループ (`resource-group-1`) に追加する例を示します。

```
# scrgadm -a -S -j resource-1 -g resource-group-1 -l schost-1
# scrgadm -pv -j resource-1
(resource-group-1) リソース 名前: resource-1
(resource-group-1:resource-1) リソース R_description:
(resource-group-1:resource-1) リソース リソースタイプ: SUNW.SharedAddress
(resource-group-1:resource-1) リソース タイプのバージョン: 1.0
(resource-group-1:resource-1) リソース リソースグループ名: resource-group-1
(resource-group-1:resource-1) リソース 有効: False
(resource-group-1:resource-1) リソース リソースプロジェクト名: default
(resource-group-1:resource-1) リソース 有効なモニター: True
```

次に進む手順

共有リソースを追加したあと、50 ページの「リソースグループをオンラインにする」の手順に従ってリソースを有効にします。

▼ フェイルオーバーアプリケーションリソースをリソースグループに追加する

フェイルオーバーアプリケーションリソースは、以前にフェイルオーバーリソースグループに作成した論理ホスト名を使用するアプリケーションリソースです。

この手順を実行するには、次の情報が必要になります。

- リソースを追加するフェイルオーバーリソースグループの名前
- リソースが属するリソースタイプの名前
- アプリケーションリソースが使用する論理ホスト名リソース。これは、以前に同じリソースグループに含めた論理ホスト名になります。

追加情報については、`scrgadm(1M)` のマニュアルページを参照してください。

注 - この手順は、任意のクラスタノードから実行します。

1. クラスタメンバー上でスーパーユーザーになります。
2. フェイルオーバーアプリケーションリソースをリソースグループに追加します。

```
# scrgadm -a -j resource -g resource-group -t resource-type \  
[-x Extension_property=value, ...] [-y Standard_property=value, ...]
```

<code>-a</code>	リソースを追加します。
<code>-j resource</code>	追加するリソースの名前を指定します。
<code>-g resource-group</code>	以前に作成したフェイルオーバーリソースグループの名前を指定します。
<code>-t resource-type</code>	リソースが属するリソースタイプの名前を指定します。
<code>-x <i>Extension_property [] =value, ...</i></code>	特定のデータサービスに依存する拡張プロパティをコンマで区切って指定します。データサービスがこのプロパティの指定が必要かどうかについては、各データサービスについて説明している章を参照してください。
<code>-y <i>Standard_property [] =value, ...</i></code>	特定のデータサービスに依存する標準プロパティをコンマで区切って指定します。データサービスがこのプロパティの指定が必要かどうかについては、各データサービスについて説明している章と付録 A を参照してください。

注 - 別のプロパティを設定することもできます。詳細は、付録 A とこのマニュアルのフェイルオーバーデータサービスのインストールと構成に関する各章を参照してください。

3. フェイルオーバーアプリケーションリソースが追加され、妥当性が検査されていることを確認します。

```
# scrgadm -pv -j resource
```

リソースを追加すると、Sun Cluster ソフトウェアは、そのリソースの妥当性を検査します。妥当性が確認されると、そのリソースを有効にできるとともに、そのリソースグループを RGM の管理下に置くことができます。妥当性の検査に失敗すると、scrgadm コマンドはエラーメッセージを生成して終了します。妥当性の検査に失敗に失敗した場合は、エラーメッセージについて各ノード上の syslog を調べてください。メッセージは、妥当性の検査を実施したノードで表示されます。必ずしも scrgadm コマンドを実行したノードで表示されるわけではありません。

例 - フェイルオーバーアプリケーションリソースのリソースグループへの追加

次に、リソース (resource-1) をリソースグループ (resource-group-1) に追加する例を示します。リソースは、以前に定義したフェイルオーバーリソースグループと同じリソースグループに存在している論理ホスト名リソース (schost-1、schost-2) に依存しています。

```
# scrgadm -a -j resource-1 -g resource-group-1 -t resource-type-1 \  
-y Network_resources_used=schost-1,schost2 \  
# scrgadm -pv -j resource-1  
(resource-group-1) リソース 名前: resource-1  
  (resource-group-1:resource-1) リソース R_description:  
  (resource-group-1:resource-1) リソース リソースタイプ: resource-type-1  
  (resource-group-1:resource-1) リソース タイプのバージョン: 1.0  
  (resource-group-1:resource-1) リソース リソースグループ名: resource-group-1  
  (resource-group-1:resource-1) リソース リソースプロジェクト名: default  
  (resource-group-1:resource-1) リソース 有効: False  
  (resource-group-1:resource-1) リソース 有効なモニター: True
```

次に進む手順

フェイルオーバーアプリケーションリソースを追加したあと、50 ページの「リソースグループをオンラインにする」の手順に従ってリソースを有効にします。

▼ スケーラブルアプリケーションリソースをリソースグループに追加する

スケーラブルアプリケーションリソースは、フェイルオーバーリソースグループに共有アドレスを使用するアプリケーションリソースです。

この手順を実行するには、次の情報が必要になります。

- リソースを追加するスケーラブルリソースグループの名前
- リソースが属するリソースタイプの名前
- スケーラブルサービスリソースが使用する共有アドレスリソース。これは、以前にフェイルオーバーリソースグループに含めた共有アドレスになります。

追加情報については、`scrgadm(1M)` のマニュアルページを参照してください。

注 - この手順は、任意のクラスタノードから実行します。

1. クラスタメンバー上でスーパーユーザーになります。
2. スケーラブルアプリケーションリソースをリソースグループに追加します。

```
# scrgadm -a -j resource -g resource-group -t resource-type \  
-y Network_resources_used=network-resource[,network-resource...] \  
-y Scalable=True \  
[-x Extension_property=value, ...] [-y Standard_property=value, ...]  
  
-a  
    リソースを追加します。  
  
-j resource  
    追加するリソースの名前を指定します。  
  
-g resource-group  
    以前に作成したスケーラブルサービスリソースグループの名前を指定します。  
  
-t resource-type  
    このリソースが属するリソースタイプの名前を指定します。  
  
-y Network_resources_used [ ] = [ ] network-resource[,network-resource...  
]  
    このリソースが依存するネットワークリソース (共有アドレス) のリストを指定  
    します。  
  
-y Scalable [ ] =True  
    このリソースがスケーラブルであることを指定します。
```


- x *Extension_property* [=] *value*, ...
特定のデータサービスに依存する拡張プロパティをコンマで区切って指定します。データサービスがこのプロパティの指定が必要かどうかについては、各データサービスについて説明している章を参照してください。
- y *Standard_property* [=] *value*, ...
特定のデータサービスに依存する標準プロパティをコンマで区切って指定します。データサービスがこのプロパティの指定が必要かどうかについては、各データサービスについて説明している章と付録 A を参照してください。

注 - 別のプロパティを設定することもできます。構成可能なほかのプロパティについては、付録 A とこのマニュアルのスケラブルデータサービスのインストールと構成に関する各章を参照してください。スケラブルサービスの場合は、通常、*Port_list*、*Load_balancing_weights*、*Load_balancing_policy* プロパティを設定します (付録 A を参照)。

3. スケラブルアプリケーションリソースが追加され、妥当性が検査されていることを確認します。

```
# scrgadm -pv -j resource
```

リソースを追加すると、Sun Cluster ソフトウェアは、そのリソースの妥当性を検査します。妥当性が確認されると、そのリソースを有効にできるとともに、そのリソースグループを RGM の管理下に置くことができます。妥当性の検査に失敗すると、*scrgadm* コマンドはエラーメッセージを生成して終了します。妥当性の検査に失敗した場合、エラーメッセージについて各ノード上の *syslog* を調べてください。メッセージは、妥当性の検査を実施したノードで表示されます。必ずしも *scrgadm* コマンドを実行したノードで表示されるわけではありません。

例 - スケラブルアプリケーションリソースのリソースグループへの追加

次に、リソース (*resource-1*) をリソースグループ (*resource-group-1*) に追加する例を示します。*resource-group-1* は、使用されているネットワークアドレス (以下の例の *schost-1* と *schost-2*) を含むフェイルオーバーリソースグループに依存することに注意してください。リソースは、以前に定義した 1 つまたは複数のフェイルオーバーリソースグループに存在している共有アドレスリソース (*schost-1* と *schost-2*) に依存しています。

```
# scrgadm -a -j resource-1 -g resource-group-1 -t resource-type-1 \
-y Network_resources_used=schost-1,schost-2 \
-y Scalable=True
# scrgadm -pv -j resource-1
(resource-group-1) リソース 名前:                resource-1
(resource-group-1:resource-1) リソース R_description:
(resource-group-1:resource-1) リソース リソースタイプ:    resource-type-1
```

```
(resource-group-1:resource-1) リソース タイプのバージョン: 1.0
(resource-group-1:resource-1) リソース リソースグループ名: resource-group-1
(resource-group-1:resource-1) リソース リソースプロジェクト名: default
(resource-group-1:resource-1) リソース 有効: False
(resource-group-1:resource-1) リソース 有効なモニター: True
```

次に進む手順

スケーラブルアプリケーションリソースを追加したあと、50 ページの「リソースグループをオンラインにする」の手順に従って、リソースを有効にします。

リソースグループをオンラインにする

リソースが HA サービスの提供を開始できるようにするには、リソースグループのリソースおよびリソースモニターを有効にし、リソースグループを管理状態にし、リソースグループをオンラインにする必要があります。これらの作業は個々に実行できますが、次に示すように 1 つの手順で実行することもできます。詳細は、`scswitch (1M)` のマニュアルページを参照してください。

注 - この手順は、任意のクラスタノードから実行します。

▼ リソースグループをオンラインにする

1. クラスタメンバー上でスーパーユーザーになります。
2. リソースを有効にし、リソースグループをオンラインにします。
リソースモニターを無効にしている場合は、これも有効になります。

```
# scswitch -z -g resource-group
```

-z 最初にリソースグループとそのモニターを有効にし、リソースグループをオンラインにします。

-g *resource-group* オンラインにするリソースグループの名前を指定します。既存のリソースグループを指定する必要があります。

3. リソースがオンラインになっていることを確認します。

任意のクラスタノードで次のコマンドを実行し、`Resource Group State` のフィールドを調べ、ノードリストで指定されたノードでリソースグループがオンラインになっていることを確認します。

```
# scstat -g
```

例 – リソースグループをオンラインにする

次に、リソースグループ (resource-group-1) をオンラインにし、その状態を確認する例を示します。

```
# scswitch -Z -g resource-group-1
# scstat -g
```

次に進む手順

リソースグループがオンラインになれば、リソースグループが構成されて使用する準備が整ったこととなります。リソースやノードで障害が発生した場合は、RGM は別のノードでそのリソースグループをオンラインに切り替えることでリソースグループの可用性を維持します。

リソースモニターの無効化と有効化

次の各手順では、リソース自体とは関係なくリソースフォルトモニターだけを無効または有効にします。したがって、フォルトモニターが無効にされても、そのリソース自体は正常に動作を続けます。ただし、フォルトモニターが無効になっていると、データサービスに障害が発生しても、障害回復は自動的に開始されません。

追加情報については、scswitch(1M) のマニュアルページを参照してください。

注 – この手順は任意のノードから実行できます。

▼ リソースフォルトモニターを無効にする

1. クラスタメンバー上でスーパーユーザーになります。
2. リソースフォルトモニターを無効にします。

```
# scswitch -n -M -j resource
-n          リソースまたはリソースモニターを無効にします。
-M          指定されたリソースのフォルトモニターを無効にします。
-j resource リソースの名前
```

3. リソースフォルトモニターが無効になっていることを確認します。

各クラスターノードで次のコマンドを実行し、監視されるフィールド (RS Monitored) を確認します。

```
# scrgadm -pv
```

例-リソースフォルトモニターを無効にする

この例では、リソースフォルトモニターを無効にします。

```
# scswitch -n -M -j resource-1
# scrgadm -pv
...
RS Monitored: no...
```

▼ リソースフォルトモニターを有効にする

1. クラスターメンバー上でスーパーユーザーになります。
2. リソースフォルトモニターを有効にします。

```
# scswitch -e -M -j resource
-e          リソースまたはリソースモニターを有効にします。
-M          指定されたリソースのフォルトモニターを有効にします。
-j resource リソースの名前を指定します。
```

3. リソースフォルトモニターが有効になっていることを確認します。
各クラスターノードで次のコマンドを実行し、監視されるフィールド (RS Monitored) を確認します。

```
# scrgadm -pv
```

例-リソースフォルトモニターを有効にする

この例では、リソースフォルトモニターを有効にします。

```
# scswitch -e -M -j resource-1
# scrgadm -pv
...
RS Monitored: yes...
```

リソースタイプの削除

使用されていないリソースタイプを削除する必要はありませんが、次の手順を使用して削除できます。

追加情報については、`scrgadm(1M)` および `scswitch(1M)` のマニュアルページを参照してください。

注 – この手順は、任意のクラスタノードから実行します。

▼ リソースタイプを削除する

リソースタイプを削除する前に、クラスタ内のすべてのリソースグループにある、そのタイプのリソースをすべて無効にし、削除する必要があります。 `scrgadm -pv` コマンドを使用し、クラスタ内のリソースとリソースグループを確認します。

1. クラスタメンバー上でスーパーユーザーになります。
2. 削除するリソースタイプの各リソースを無効にします。

```
# scswitch -n -j resource
-n                リソースを無効にします。
-j resource       無効にするリソースの名前を指定します。
```

3. 削除するリソースタイプの各リソースを削除します。

```
# scrgadm -r -j resource
-r                指定したリソースを削除します。
-j                削除するリソースの名前を指定します。
```

4. リソースタイプを削除します。

```
# scrgadm -r -t resource-type
-r                指定したリソースタイプを削除します。
-t resource-type  削除するリソースタイプの名前を指定します。
```

5. リソースタイプが削除されていることを確認します。

```
# scrgadm -p
```

例 – リソースタイプの削除

次に、リソースタイプのすべてのリソース (resource-type-1) を無効にして削除したあとで、そのリソースタイプ自体を削除する例を示します。この例では、resource-1 は、リソースタイプ resource-type-1 のリソースです。

```
# scswitch -n -j resource-1
# scrgadm -r -j resource-1
# scrgadm -r -t resource-type-1
```

リソースグループの削除

リソースグループを削除するには、最初にそのリソースグループからすべてのリソースを削除する必要があります。

追加情報については、scrgadm(1M) および scswitch(1M) のマニュアルページを参照してください。

注 – この手順は、任意のクラスタノードから実行します。

▼ リソースグループを削除する

1. クラスタメンバー上でスーパーユーザーになります。
2. 次のコマンドを実行し、リソースグループをオフラインに切り替えます。

```
# scswitch -F -g resource-group
```

-F リソースグループをオフラインに切り替えます。

-g resource-group オフラインにするリソースグループの名前を指定します。

3. リソースグループに含まれているすべてのリソースを無効にします。

scrgadm -pv コマンドを使用し、リソースグループ内のリソースを表示できます。リソースグループ内の削除するすべてのリソースを無効にします。

```
# scswitch -n -j resource
```

-n リソースを無効にします。

-j resource 無効にするリソースの名前を指定します。

依存性のあるデータサービスリソースがリソースグループに存在する場合、そのリソースを無効にするには、依存するすべてのリソースを無効にする必要があります。

す。

- リソースグループからすべてのリソースを削除します。
scrgadm コマンドを使用して次の操作を行います。

- リソースの削除
- リソースグループの削除

```
# scrgadm -r -j resource
# scrgadm -r -g resource-group

-r                指定したリソースやリソースグループを削除します。
-j resource       削除するリソースの名前を指定します。
-g resource-group 削除するリソースグループの名前を指定します。
```

- リソースグループが削除されていることを確認します。

```
# scrgadm -p
```

例 — リソースグループの削除

次に、リソースグループ (resource-group-1) のリソース (resource-1) を削除したあとで、そのリソースグループ自体を削除する例を示します。

```
# scswitch -F -g resource-group-1
# scrgadm -r -j resource-1
# scrgadm -r -g resource-group-1
```

リソースの削除

リソースグループからリソースを削除する前に、そのリソースを無効にします。

追加情報については、scrgadm(1M) および scswitch(1M) のマニュアルページを参照してください。

注 – この手順は、任意のクラスタノードから実行します。

▼ リソースを削除する

1. クラスタメンバー上でスーパーユーザーになります。
2. 削除するリソースを無効にします。

```
# scswitch -n -j resource
```

-n リソースを無効にします。

-j resource 無効にするリソースの名前を指定します。

3. リソースを削除します。

```
# scrgadm -r -j resource
```

-r 指定したリソースを削除します。

-j resource 削除するリソースの名前を指定します。

4. リソースが削除されていることを確認します。

```
# scrgadm -p
```

例 – リソースの削除

次に、リソース resource-1 を無効にして削除する例を示します。

```
# scswitch -n -j resource-1
```

```
# scrgadm -r -j resource-1
```

リソースグループの稼動系の切り替え

以下の手順を使用し、リソースグループの現在の稼動系を別のノードに切り替え (スイッチオーバー)、新しい稼動系にすることができます。

追加情報については、scrgadm(1M) および scswitch(1M) のマニュアルページを参照してください。

注 – この手順は、任意のクラスタノードから実行します。

▼ リソースグループの稼働系を切り替える

この手順を実行するには、次の情報が必要になります。

- スイッチオーバーするリソースグループの名前
- リソースグループをオンラインにする、またはオンラインを維持するノードの名前。スイッチオーバーを行うリソースグループの、待機系として設定されているクラスタノードを指定する必要があります。リソースグループの待機系の一覧を表示するには、`scrgadm -pv` コマンドを使用します。

1. クラスタメンバー上でスーパーユーザーになります。
2. 稼働系を待機系に切り替えます。

```
# scswitch -z -g resource-group -h nodelist
-z                指定したリソースグループをオンラインに切り替えます。
-g resource-group 切り替えるリソースグループの名前を指定します。
-h nodelist       リソースグループをオンラインにする、またはオンラインを
                  維持するノードを指定します。このリソースグループは、こ
                  のノード以外のすべてのノードでオフラインに切り替えられ
                  ます。
```

3. リソースグループが新しい稼働系に切り替えられていることを確認します。
次のコマンドを実行し、スイッチオーバーされたリソースグループの状態に関する出力を調べます。

```
# scstat -g
```

例 – リソースグループを新しい稼働系に切り替える

次に、リソースグループ (`resource-group-1`) を現在の稼働系 (`phys-schost-1`) から、待機系 (`phys-schost-2`) へ切り替える例を示します。まず、リソースグループが `phys-schost-1` でオンラインになっていることを確認します。続いて、切り替えを行います。最後に、そのグループが `phys-schost-2` でオンラインに切り替えられたことを確認します。

```
phys-schost-1# scstat -g
...
Resource Group Name:      resource-group-1
Status
  Node Name:              phys-schost-1
  Status:                 Online

  Node Name:              phys-schost-2
  Status:                 Offline
...
phys-schost-1# scswitch -z -g resource-group-1 -h phys-schost-2
```

```
phys-schost-1# scstat -g
...
Resource Group Name:      resource-group-1
Status
  Node Name:              phys-schost-2
  Status:                 Online

  Node Name:              phys-schost-1
  Status:                 Offline
...
```

リソースの無効化とリソースグループの UNMANAGED 状態への移行

リソースグループは、そのリソースグループに対して管理手順を実施する前に、UNMANAGED 状態に移行する必要があります。リソースグループを UNMANAGED 状態に移行する前に、リソースグループに含まれるすべてのリソースを無効にし、リソースグループをオフラインにする必要があります。

追加情報については、`scrgadm(1M)` および `scswitch(1M)` のマニュアルページを参照してください。

注 - この手順は、任意のクラスタノードから実行します。

▼ リソースを無効にしてリソースグループを UNMANAGED 状態に移行する

この手順を実行するには、次の情報が必要になります。

- 無効にするリソースの名前
- UNMANAGED 状態に移行するリソースグループの名前

この手順に必要なリソースとリソースグループの名前を判断するには、`scrgadm -pv` コマンドを使用します。

1. クラスタメンバー上でスーパーユーザーになります。
2. リソースを無効にします。
この手順を、リソースグループ内のすべてのリソースに対して実行します。

```
# scswitch -n -j resource
```

-n リソースを無効にします。

-j resource 無効にするリソースの名前を指定します。

3. 次のコマンドを実行し、リソースグループをオフラインに切り替えます。

```
# scswitch -F -g resource-group
```

-F リソースグループをオフラインに切り替えます。

-g resource-group オフラインにするリソースグループの名前を指定します。

4. リソースグループを **UNMANAGED** 状態にします。

```
# scswitch -u -g resource-group
```

-u 指定したリソースグループを **UNMANAGED** 状態にします。

-g resource-group **UNMANAGED** 状態にするリソースグループの名前を指定します。

5. リソースが無効になり、リソースグループが **UNMANAGED** 状態になっていることを確認します。

```
# scrgadm -pv -g resource-group
```

例 – リソースの無効化とリソースグループの **UNMANAGED** 状態への移行

次に、リソース (resource-1) を無効にし、リソースグループ (resource-group-1) を **UNMANAGED** 状態に移行する例を示します。

```
# scswitch -n -j resource-1
# scswitch -F -g resource-group-1
# scswitch -u -g resource-group-1
# scrgadm -pv -g resource-group-1
リソースグループ 名前: resource-group-1
(resource-group-1) リソースグループ RG_description: <NULL>
(resource-group-1) リソースグループ management state: Unmanaged
(resource-group-1) リソースグループ Failback: False
(resource-group-1) リソースグループ Nodelist: phys-schost-1
phys-schost-2
(resource-group-1) リソースグループ Maximum primaries: 2
(resource-group-1) リソースグループ Desired primaries: 2
(resource-group-1) リソースグループ RG_dependencies: <NULL>
(resource-group-1) リソースグループ mode: Failover
(resource-group-1) リソースグループ network dependencies: True
(resource-group-1) リソースグループ Global_resources_used: All
(resource-group-1) リソースグループ Pathprefix:

(resource-group-1) リソース 名前: resource-1
```

```
(resource-group-1:resource-1) リソース R_description:
(resource-group-1:resource-1) リソース リソースタイプ:          SUNW.apache
(resource-group-1:resource-1) リソース タイプのバージョン:      1.0
(resource-group-1:resource-1) リソース リソースグループ名:      resource-group-1
(resource-group-1:resource-1) リソース リソースプロジェクト名:  default
(resource-group-1:resource-1) リソース 有効:                    True
(resource-group-1:resource-1) リソース 有効なモニター:          False
```

リソースタイプ、リソースグループ、リソース構成情報の表示

リソース、リソースグループ、リソースタイプで管理手順を実施する前に、この手順を使用し、これらのオブジェクトの現在の構成設定を表示します。

追加情報については、`scrgadm(1M)` および `scswitch(1M)` のマニュアルページを参照してください。

注 – この手順は、任意のクラスタノードから実行します。

リソースタイプ、リソースグループ、リソース構成情報の表示

`scrgadm` コマンドは、構成状態に関する次の3つのレベルの情報を表示します。

- `-p` オプションを指定した場合は、リソースタイプ、リソースグループ、リソースのプロパティ値に関する最小限の情報が表示されます。
- `-pv` オプションを指定した場合は、ほかのリソースタイプ、リソースグループ、リソースプロパティに関する詳細が表示されます。
- `-pvv` オプションを指定した場合は、リソースタイプメソッド、拡張プロパティ、すべてのリソースとリソースグループのプロパティを含む、詳細情報が表示されます。

また、表示したいオブジェクトの名前の後に `-t` (リソースタイプ)、`-g` (リソースグループ)、および `-j` (リソース) オプションを指定することによって、特定のリソースタイプ、リソースグループ、またはリソースのステータス情報を確認できます。たとえば、次のコマンドは、リソース `apache-1` のみについて、特定の情報を表示することを指定します。

```
# scrgadm -p[v[v]] -j apache-1
```

詳細は、scrgadm(1M)のマニュアルページを参照してください。

リソースタイプ、リソースグループ、リソースプロパティの変更

リソースグループとリソースは、変更可能な標準の構成プロパティを持っています。次の各手順では、これらのプロパティの変更方法を説明します。

リソースは、拡張プロパティも持っており、一部の拡張プロパティはデータサービス開発者によってあらかじめ定義されているため、変更することができません。各データサービスの拡張プロパティの一覧については、このマニュアルのデータサービスに関する各章を参照してください。

リソースグループとリソースの標準的な構成プロパティに関する詳細は、scrgadm(1M)のマニュアルページを参照してください。

▼ リソースタイププロパティを変更する

この手順を実行するには、次の情報が必要になります。

- 変更するリソースタイプの名前
- 変更するリソースタイププロパティの名前。リソースタイプの場合、変更できるのは1つのプロパティのみです。つまり、このリソースタイプをインスタンス化できるノードのリストのみです。

注 - この手順は、任意のクラスタノードから実行します。

1. クラスタメンバー上でスーパーユーザーになります。
2. **scrgadm** コマンドを使用し、この手順に必要なリソースタイプの名前を判断します。

```
# scrgadm -pv
```

3. リソースタイププロパティを変更します。
リソースタイプで変更できる唯一のプロパティは、`Installed_node_list` です。

```
# scrgadm -c -t resource-type -h installed-node-list
```

- c 指定したリソースタイププロパティを変更します。
- t *resource-type* リソースタイプの名前を指定します。
- h *installed-node-list* このリソースタイプがインストールされるノードの名前を指定します。

4. リソースタイププロパティが変更されていることを確認します。

```
# scrgadm -pv -t resource-type
```

例 – リソースタイププロパティの変更

次に、SUNW.apache プロパティを変更し、このリソースタイプが2つのノード (phys-schost-1 および phys-schost-2) にインストールされるように定義する例を示します。

```
# scrgadm -c -t SUNW.apache -h phys-schost-1,phys-schost-2
# scrgadm -pv -t SUNW.apache
リソースタイプ 名前: SUNW.apache
(SUNW.apache) リソースタイプ 説明: Apache Resource Type
(SUNW.apache) リソースタイプ ベースディレクトリ: /opt/SUNWscapc/bin
(SUNW.apache) リソースタイプ 単一のインスタンス: False
(SUNW.apache) リソースタイプ 初期ノード: All potential masters
(SUNW.apache) リソースタイプ フェイルオーバー: False
(SUNW.apache) リソースタイプ バージョン: 1.0
(SUNW.apache) リソースタイプ API バージョン: 2
(SUNW.apache) リソースタイプ ノードにインストールされている: phys-schost1 phys-schost-2
(SUNW.apache) リソースタイプ パッケージ: SUNWscapc
(SUNW.apache) リソースタイプ システム: False
```

▼ リソースグループプロパティを変更する

この手順を実行するには、次の情報が必要になります。

- 変更するリソースグループの名前
- 変更するリソースグループプロパティの名前とその新しいプロパティ値

この手順では、リソースグループプロパティの変更方法について説明しています。リソースグループプロパティの一覧については、付録 A を参照してください。

注 – この手順は、任意のクラスタノードから実行します。

1. クラスタメンバー上でスーパーユーザーになります。
2. リソースグループプロパティを変更します。

```
# scrgadm -c -g resource-group -y property=new_value
```

-c 指定したプロパティを変更します。

-g *resource-group* リソースグループの名前を指定します。

-y *property* 変更するプロパティの名前を指定します。

- リソースグループプロパティが変更されていることを確認します。

```
# scrgadm -pv -g resource-group
```

例 – リソースグループプロパティの変更

次に、リソースグループ (*resource-group-1*) の Failback プロパティを変更する例を示します。

```
# scrgadm -c -g resource-group-1 -y Failback=True
```

```
# scrgadm -pv -g resource-group-1
```

▼ リソースプロパティを変更する

この手順を実行するには、次の情報が必要になります。

- 変更するプロパティを持つリソースの名前
- 変更するプロパティの名前

この手順は、リソースプロパティの変更方法について説明しています。リソースグループプロパティの一覧については、付録 A を参照してください。

注 – この手順は、任意のクラスタノードから実行します。

- クラスタメンバー上でスーパーユーザーになります。
- scrgadm -pvv** コマンドを実行し、現在のリソースプロパティ設定を表示します。

```
# scrgadm -pvv -j resource
```

- リソースプロパティを変更します。

```
# scrgadm -c -j resource -y property=new_value | -x extension_property=new_value
```

-c 指定したプロパティを変更します。

-j *resource* リソースの名前を指定します。

<code>-y property =new_value</code>	変更する標準プロパティの名前を指定します。
<code>-x extension_property =new_value</code>	変更する拡張プロパティの名前を指定します。Sun が提供するデータサービスについては、データサービスのインストールと構成に関する各章で説明されている拡張プロパティを参照してください。

- リソースプロパティが変更されていることを確認します。

```
# scrgadm pvv -j resource
```

例 – 標準リソースプロパティの変更

次に、リソース (resource-1) のシステム定義プロパティ (Start_timeout) の変更例を示します。

```
# scrgadm -c -j resource-1 -y start_timeout=30
# scrgadm -pvv -j resource-1
```

例 – 拡張リソースプロパティの変更

次に、リソース (resource-1) の拡張プロパティ (Log_level) の変更例を示します。

```
# scrgadm -c -j resource-1 -x Log_level=3
# scrgadm -pvv -j resource-1
```

リソースの STOP_FAILED エラーフラグの消去

Failover_mode リソースプロパティが NONE または SOFT に設定されているときに、リソースの STOP に失敗した場合は、個々のリソースは STOP_FAILED 状態になり、リソースグループは ERROR_STOP_FAILED 状態になります。この状態のリソースグループは、ノード上でオンラインにできません。また、リソースの作成や削除、リソースグループやリソースプロパティの変更などの編集操作を行うこともできません。

▼ リソースの STOP_FAILED エラーフラグを消去する

この手順を実行するには、次の情報が必要になります。

- リソースが STOP_FAILED であるノードの名前
- STOP_FAILED 状態になっているリソースとリソースグループの名前

追加情報については、scswitch(1M) のマニュアルページを参照してください。

注 – この手順は、任意のクラスタノードから実行します。

1. クラスタメンバー上でスーパーユーザーになります。
2. **STOP_FAILED** 状態のリソースと、どのノードでこの状態なのかを確認します。

```
# scstat -g
```

3. **STOP_FAILED** 状態になっているノード上で、リソースとそのモニターを手作業で停止します。

この手順では、プロセスを強制終了するか、リソースタイプ固有のコマンドまたは別のコマンドを実行する必要があります。

4. リソースを手作業で停止したすべてのノード上で、これらのリソースの状態を手作業で **OFFLINE** に設定します。

```
# scswitch -c -h nodelist -j resource -f STOP_FAILED
```

-c フラグを消去します。

-h *nodelist* リソースが実行されていたノード名を指定します。

-j *resource* オフラインにするリソースの名前を指定します。

-f **STOP_FAILED** フラグ名を指定します。

5. 手順 4 で **STOP_FAILED** フラグを消去したノード上で、リソースグループの状態を調べます。

リソースグループの状態は、OFFLINE または ONLINE になっています。

```
# scstat -g
```

コマンド `scstat -g` は、リソースグループの状態が `ERROR_STOP_FAILED` のままかを示します。リソースグループがまだ `ERROR_STOP_FAILED` 状態の場合は、`scswitch` コマンドを実行し、該当するノード上でリソースグループをオフラインに切り替えます。

```
# scswitch -F -g resource-group
```

- F グループをマスターできるすべてのノード上でリソースグループをオフラインにします。
- gresource-group オフラインに切り替えるリソースグループの名前を指定します。

この状況は STOP メソッドに失敗し、停止に失敗したリソースがリソースグループ内のほかのノードの依存性を持っているときに、リソースグループをオフラインに切り替えた場合に発生します。これ以外の状況では、手順 4 のコマンドをすべての STOP_FAILED リソースで実行することによって、リソースグループは自動的に ONLINE または OFFLINE 状態に戻ります。

これで、リソースグループを ONLINE 状態に切り替えることができます。

登録済みのリソースタイプの再登録

あらかじめ登録されているリソースタイプには、SUNW.LogicalHostname と SUNW.SharedAddress があります。すべての論理ホスト名と共有アドレスリソースがこれらのリソースタイプを使用します。これら 2 つのリソースタイプは、誤って削除した場合を除き、登録する必要はありません。誤ってリソースタイプを削除した場合は、次の手順を使用して再登録してください。

追加情報については、scrgadm(1M) のマニュアルページを参照してください。

注 - この手順は、任意のクラスタノードから実行します。

▼ 登録済みのリソースタイプを再登録する

- リソースタイプを再登録します。

```
# scrgadm -a -t SUNW.resource-type
```

-a リソースタイプを追加します。

-t SUNW.resource-type 追加する (再登録する) リソースタイプを指定します。
リソースタイプは、SUNW.LogicalHostname または
SUNW.SharedAddress のいずれかになります。

例 – 登録済みのリソースタイプの再登録

次に、LogicalHostname リソースタイプを再登録する例を示します。

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

リソースグループへのノードの追加と削除

この節の手順では、次の作業を行います。

- リソースグループの追加のマスターとなるクラスタノードを構成する
- リソースグループからノードを削除する

ノードの追加や削除をフェイルオーバーリソースグループに対して行うのか、スケラブルリソースグループに対して行うのかによって、手順は異なります。

フェイルオーバーリソースグループは、フェイルオーバーとスケラブルの両方のサービスによって使用されるネットワークリソースを含みます。クラスタに接続される各 IP サブネットワークは、指定された独自のネットワークリソースを持ち、フェイルオーバーリソースグループに含まれます。このネットワークリソースは、論理ホスト名または共有アドレスリソースのいずれかになります。各ネットワークリソースは、それが使用する IP ネットワークマルチパスグループのリストを含んでいます。フェイルオーバーリソースグループの場合は、リソースグループ (netiflist リソースプロパティ) に含まれる各ネットワークリソースに対し、IP ネットワークマルチパスグループの完全なリストを更新する必要があります。

スケラブルリソースグループの場合は、スケラブルグループをホストの新しいセット上でマスターされるように変更するほかに、スケラブルリソースによって使用されるネットワークリソースを含むフェイルオーバーグループのための手順も実行する必要があります。

追加情報については、scrgadm(1M)のマニュアルページを参照してください。

注 – 任意のクラスタノードから、以下に説明する手順のいずれかを実行します。

リソースグループにノードを追加する

ノードをリソースグループに追加する手順は、リソースグループがスケラブルリソースグループか、またはフェイルオーバーリソースグループかによって異なります。詳細の手順については、以下の節を参照してください。

- 68 ページの「スケラブルリソースグループにノードを追加する」
- 69 ページの「フェイルオーバーリソースグループにノードを追加する」

この手順を実行するには、次の情報が必要になります。

- すべてのクラスタノードの名前とノード ID
- ノードが追加されるリソースグループの名前
- すべてのノード上のリソースグループによって使用されるネットワークリソースをホストする IP ネットワークマルチパスグループの名前

さらに、新しいノードがすでにクラスタメンバーになっていることも確認してください。

▼ スケラブルリソースグループにノードを追加する

1. リソースグループ内のスケラブルリソースが使用する各ネットワークリソースごとに、そのネットワークリソースが配置されているリソースグループが新しいノードで実行されるようにします。

詳細は、以下の作業の手順 1 から手順 4 を参照してください。

2. スケラブルリソースグループをマスターできるノードのリスト (**nodelist** リソースグループプロパティ) に新しいノードを追加します。

この手順は、**nodelist** の値を上書きするため、リソースグループをマスターできるすべてのノードをここに含める必要があります。

```
# scrgadm -c -g resource-group -h nodelist
```

-c	リソースグループを変更します。
-gresource-group	ノードが追加されるリソースグループの名前を指定します。
-hnodelist	リソースグループをマスターできるノードをコマンドで区切って指定します。

3. (省略可能) スケラブルリソースの **Load_balancing_weights** プロパティを更新し、リソースグループに追加するノードにウェイトを割り当てます。

ウェイトを割り当てない場合は、デフォルトで 1 になります。詳細は、**scrgadm (1M)** のマニュアルページを参照してください。

▼ フェイルオーバーリソースグループにノードを追加する

1. 現在のノードリスト、およびリソースグループ内の各リソース用に構成された IP ネットワークマルチパスグループの現在のリストを表示します。

```
# scrgadm -pvv-gresource-group | grep -i nodelist
# scrgadm -pvv-gresource-group | grep -i netiflist
```

注 -nodelist と netiflist のコマンド行出力では、ノード名でノードが識別されます。ノード ID を識別するには、コマンド `scconf -pv | grep -i node_id` を実行してください。

2. ノードの追加によって影響を受けるネットワークリソースの **netiflist** を更新します。

この手順は、*netiflist* の値を上書きするため、すべての IP ネットワークマルチパスグループをここに含める必要があります。

```
# scrgadm -c -j network-resource -x netiflist=netiflist
```

-c ネットワークリソースを変更します。

-j *network-resource* *netiflist* エントリ上でホストされているネットワークリソースの名前 (論理ホスト名または共有アドレス) を指定します。

-x *netiflist* =*netiflist* 各ノード上の IP ネットワークマルチパスグループをコンマで区切って指定します。 *netiflist* の各要素は、*netif@node* の形式にする必要があります。 *netif* は IP ネットワークマルチパスグループ名 (*sc_ipmp0* など) として指定できます。ノードには、ノード名またはノード ID (*sc_ipmp0@1*、*sc_ipmp@phys-schost-1* など) を指定できます。

3. このリソースグループをマスターできるすべてのノードを含めるように、ノードリストを更新します。

この手順は、*nodelist* の値を上書きするため、リソースグループをマスターできるすべてのノードをここに含める必要があります。

```
# scrgadm -c -g resource-group -h nodelist
```

-c リソースグループを変更します。

-g*resource-group* ノードが追加されるリソースグループの名前を指定します。

-h*nodelist* リソースグループをマスターできるノードをコンマで区切って指定します。

4. 更新された情報を確認します。

```
# scrgadm -pvv -g resource-group | grep -i nodelist
# scrgadm -pvv -g resource-group | grep -i netiflist
```

例 – リソースグループへのノードの追加

次に、リソースグループ (resource-group-1) にノード (phys-schost-2) を追加する例を示します。このリソースグループは、論理ホスト名リソース (schost-2) を含んでいます。

```
# scrgadm -pvv -g resource-group-1 | grep -i nodelist
(resource-group-1) リソースグループ Nodelist:      phys-schost-1 phys-schost-3
# scrgadm -pvv -g resource-group-1 | grep -i netiflist
(resource-group-1:schost-2) リソース property name: NetIfList
(resource-group-1:schost-2:NetIfList) リソース property class: extension
(resource-group-1:schost-2:NetIfList) List of IP Networking Multipathing
interfaces on each node
(resource-group-1:schost-2:NetIfList) リソース property type: stringarray
(resource-group-1:schost-2:NetIfList) リソース property value: sc_ipmp0@1 sc_ipmp0@3
```

(ノード 1 と 3 のみが IP ネットワークマルチパスグループに割り当てられています。ノード 2 用の IP ネットワークマルチパスグループを追加する必要があります。)

```
# scrgadm -c -j schost-2-x netiflist=sc_ipmp0@1,sc_ipmp0@2,sc_ipmp0@3
# scrgadm -c -g resource-group-1 -h phys-schost-1,phys-schost-2,phys-schost-3
# scrgadm -pvv -g resource-group-1 | grep -i nodelist
(resource-group-1) リソースグループ Nodelist:      phys-schost-1 phys-schost-2
                                                    phys-schost-3
# scrgadm -pvv -g resource-group-1 | grep -i netiflist
(resource-group-1:schost-2:NetIfList) リソース property value: sc_ipmp0@1 sc_ipmp0@2
                                                    sc_ipmp0@3
```

リソースグループからノードを削除する

ノードをリソースグループから削除する手順は、リソースグループがスケーラブルリソースグループであるか、またはフェイルオーバーリソースグループであるかによって異なります。詳細の手順については、以下の節を参照してください。

- 71 ページの「スケーラブルリソースグループからノードを削除する」
- 72 ページの「フェイルオーバーリソースグループからノードを削除する」
- 74 ページの「共有アドレスリソースを含むフェイルオーバーリソースグループからノードを削除する」

具体例は、75 ページの「例 – リソースグループからのノードの削除」を参照してください。

この手順を実行するには、次の情報が必要になります。

- すべてのクラスタノードの名前とノード ID

- ```
scconf -pv | grep "Node ID"
```
- ノードが削除されるリソースグループまたはグループの名前
- ```
# scrgadm -pv | grep "Res Group Nodelist"
```
- すべてのノード上のリソースグループによって使用されるネットワークリソースをホストする IP ネットワークマルチパスグループの名前
- ```
scrgadm -pvv | grep "NetIfList.*value"
```

さらに、削除するノード上でリソースグループがマスターされていないことを確認してください。削除するノード上でマスターされている場合は、`scswitch` コマンドを実行し、そのノードでリソースグループをオフラインに切り替えてください。次の `scswitch` コマンドは、指定されたノードからリソースグループをオフラインにします。この場合、`new-masters` にこのノードが含まれてはなりません。

- ```
# scswitch -z -g resource-group -h new-masters
```
- g `resource-group` オフラインに切り替えるリソースグループ (削除するノードでマスターされている) の名前を指定します。
 - h `new-masters` このリソースグループを現在マスターできるノードを指定します。

追加情報については、`scswitch(1M)` のマニュアルページを参照してください。



注意 - すべてのリソースグループからノードを削除する場合で、スケーラブルサービス構成を使用するときは、最初にスケーラブルリソースグループからそのノードを削除してください。続いて、フェイルオーバーグループからそのノードを削除してください。

▼ スケーラブルリソースグループからノードを削除する

スケーラブルサービスは、次に示すように 2 つのリソースグループとして構成されます。

- 1 つは、スケーラブルサービスリソースを含むスケーラブルグループです。
- もう 1 つは、スケーラブルサービスリソースが使用する共有アドレスリソースを含むフェイルオーバーグループです。

スケーラブルリソースグループの `RG_dependencies` プロパティは、フェイルオーバーリソースグループへの依存性を使用してスケーラブルグループを構成するように設定されます。このプロパティの詳細は、付録 A を参照してください。

スケーラブルサービスの構成に関する詳細は、『*Sun Cluster の概念 (Solaris OS 版)*』のマニュアルを参照してください。

スケーラブルリソースグループからノードを削除すると、そのスケーラブルサービスはそのノード上でオンラインにすることができなくなります。スケーラブルリソースグループからノードを削除するには、以下の作業を行なってください。

1. スケーラブルリソースグループをマスターできるノードのリスト (**nodelist** リソースグループプロパティ) からノードを削除します。

```
# scrgadm -c -g scalable-resource-group -h nodelist
```

-c	リソースグループを変更します。
-g <i>scalable-resource-group</i>	ノードが削除されるリソースグループの名前を指定します。
-h <i>nodelist</i>	このリソースグループをマスターできるノードをコマンドで区切って指定します。

2. (省略可能) 共有アドレスリソースが入ったフェイルオーバーリソースグループからノードを削除します。

詳細は、74 ページの「共有アドレスリソースを含むフェイルオーバーリソースグループからノードを削除する」を参照してください。

3. (省略可能) スケーラブルリソースの **Load balancing weights** プロパティを更新し、リソースグループから削除するノードのウェイトを削除します。

詳細は、scrgadm(1M) のマニュアルページを参照してください。

▼ フェイルオーバーリソースグループからノードを削除する

フェイルオーバーリソースグループからノードを削除するには、以下の作業を行なってください。



注意 - すべてのリソースグループからノードを削除する場合で、スケーラブルサービス構成を使用するときは、最初にスケーラブルリソースグループからそのノードを削除してください。続いて、この方法を使用してフェイルオーバーグループからノードを削除してください。

注 - フェイルオーバーリソースグループにスケーラブルサービスが使用する共有アドレスリソースが含まれる場合は、74 ページの「共有アドレスリソースを含むフェイルオーバーリソースグループからノードを削除する」を参照してください。

1. このリソースグループをマスターできるすべてのノードを含めるように、ノードリストを更新します。

この手順はノードを削除してノードリストの値を上書きするため、リソースグループをマスターできるすべてのノードをここに含める必要があります。

```
# scrgadm -c -g failover-resource-group -h nodelist
```

-c リソースグループを変更します。

-g *failover-resource-group* ノードが削除されるリソースグループの名前を指定します。

-h *nodelist* このリソースグループをマスターできるノードをコマンドで区切って指定します。

- リソースグループ内の各リソース用に構成した IP ネットワークマルチパスグループの現在のリストを表示します。

```
# scrgadm -pvv -g failover-resource-group | grep -i netiflist
```

- ノードの削除によって影響を受けるネットワークリソースの **netiflist** を更新します。

この手順は **netiflist** の値を上書きするため、すべての IP ネットワークマルチパスグループをここに含める必要があります。

```
# scrgadm -c -j network-resource -x netiflist=netiflist
```

注 - 上記コマンド行の出力は、ノード名によってノードを識別します。ノード ID を識別するには、コマンド `scconf -pv | grep "ノード ID"` を実行してください。

-c ネットワークリソースを変更します。

-j *network-resource* **netiflist** エントリ上でホストされているネットワークリソースの名前を指定します。

-x **netiflist=netiflist** 各ノード上の IP ネットワークマルチパスグループをコマンドで区切って指定します。**netiflist** 内の各要素は、**netif@node** という形式でなければなりません。**netif** は、IP ネットワークマルチパスグループ名 (**sc_ipmp0** など) で指定できます。ノードは、ノード名またはノード ID (**sc_ipmp0@1**, **sc_ipmp@phys-schost-1** など) で識別できます。

注 - Sun Cluster では、現在、**netif** にアダプタ名を使用することはできません。

- 更新された情報を確認します。

```
# scrgadm -pvv -g failover-resource-group | grep -inodelist
```

```
# scrgadm -pvv -g failover-resource-group | grep -inetiflist
```

▼ 共有アドレスリソースを含むフェイルオーバーリソースグループからノードを削除する

スケーラブルサービスが使用する共有アドレスリソースを含むフェイルオーバーリソースグループでは、ノードは次の場所に現れます。

- フェイルオーバーリソースグループのノードリスト
- 共有アドレスリソースの `auxnodelist`

フェイルオーバーリソースグループのノードリストからノードを削除するには、72 ページの「フェイルオーバーリソースグループからノードを削除する」に示されている作業を行なってください。

共有アドレスリソースの `auxnodelist` を変更するには、共有アドレスリソースを削除して作成し直す必要があります。

フェイルオーバーグループのノードリストからノードを削除すると、そのノード上の共有アドレスリソースを継続して使用し、スケーラブルサービスを提供できます。このためには、共有アドレスリソースの `auxnodelist` にそのノードを追加する必要があります。 `auxnodelist` にノードを追加するには、以下の作業を行なってください。

注 - 以下の作業は、共有アドレスリソースの `auxnodelist` からノードを削除するためにも使用できます。 `auxnodelist` からノードを削除するには、共有アドレスリソースを削除して作成し直す必要があります。

1. スケーラブルサービスリソースをオフラインに切り替えます。
2. フェイルオーバーリソースグループから共有アドレスリソースを削除します。
3. 共有アドレスリソースを作成します。
フェイルオーバーリソースグループから削除したノードのノード ID またはノード名を `auxnodelist` に追加します。

```
# scrgadm -a -s -g failover-resource-group\  
-l shared-address -x new-auxnodelist
```

`failover-resource-group` 共有アドレスリソースを含めるために使用されたフェイルオーバーリソースグループの名前。

`shared-address` 共有アドレスの名前。

`new-auxnodelist` 妥当なノードの追加または削除によって変更された新しい `auxnodelist`。

例 – リソースグループからのノードの削除

次に、リソースグループ (resource-group-1) からノード (phys-schost-3) を削除する例を示します。このリソースグループは、論理ホスト名リソース (schost-1) を含んでいます。

```
# scrgadm -pvv -g resource-group-1 | grep -i nodelist
(resource-group-1) リソースグループ Nodelist:          phys-schost-1 phys-schost-2
                                                phys-schost-3
# scrgadm -c -g resource-group-1 -h phys-schost-1,phys-schost-2
# scrgadm -pvv -g resource-group-1 | grep -i netiflist
(resource-group-1:schost-1) リソース property name: NetIfList
(resource-group-1:schost-1:NetIfList) リソース property class: extension
(resource-group-1:schost-1:NetIfList) List of IP Networking Multipathing
interfaces on each node
(resource-group-1:schost-1:NetIfList) リソース property type: stringarray
(resource-group-1:schost-1:NetIfList) リソース property value: sc_ipmp0@1 sc_ipmp0@2
                                                sc_ipmp0@3

(sc_ipmp0@3 is the IP Networking Multipathing group to be removed.)

# scrgadm -c -jschost-1-xnetiflist=sc_ipmp0@1,sc_ipmp0@2
# scrgadm -pvv -g resource-group-1 | grep -i nodelist
(resource-group-1) リソースグループ Nodelist:          phys-schost-1 phys-schost-2
# scrgadm -pvv -g resource-group-1 | grep -i netiflist
(resource-group-1:schost-1:NetIfList) リソース property value: sc_ipmp0@1 sc_ipmp0@2
```

リソースグループとディスクデバイスグループ間での起動の同期

クラスタが起動した後、あるいは、サービスが別のノードにフェイルオーバーした後、グローバルデバイスとクラスタファイルシステムが利用できるようになるまでには、しばらく時間がかかることがあります。ただし、データサービスは、データサービスが依存するグローバルデバイスとクラスタファイルシステムがオンラインになる前に、START メソッドを実行できます。この例では、START メソッドがタイムアウトするため、データサービスが使用するリソースグループの状態をリセットし、データサービスを手動で再起動する必要があります。リソースタイプ HASTorage と HASToragePlus は、グローバルデバイスとクラスタファイルシステムを監視し、同じリソースグループ内のほかのリソースが利用可能になるまでそれらの START メソッドを待機させます (どのリソースタイプを作成するかを決定するには、18 ページの「HASTorage または HASToragePlus の選択」を参照してください)。このような追加の管理作業を軽減するには、グローバルデバイスやクラスタファイルシステムに依存するデータサービスリソースを持つすべてのリソースグループに、HASTorage または HASToragePlus を設定してください。

HAStorage リソースタイプの作成については、76 ページの「新しいリソース用に HAStorage リソースタイプを設定する」を参照してください。

HAStoragePlus リソースタイプの作成については、82 ページの「HAStoragePlus リソースタイプを設定する」を参照してください。

▼ 新しいリソース用に HAStorage リソースタイプを設定する

HAStorage は、今後の Sun Cluster でサポートされなくなる可能性があります。同等の機能が HAStoragePlus でサポートされています。HAStorage から HAStoragePlus へアップグレードするには、78 ページの「HAStorage から HAStoragePlus へのアップグレード」を参照してください。

次の例では、リソースグループ `resource-group-1` は、次の 3 つのデータサービスを含んでいます。

- Sun Java System Web Server (`/global/resource-group-1` に依存する)
- Oracle (`/dev/global/dsk/d5s2` に依存する)
- NFS (`dsk/d6` に依存する)

新しいリソースに対し、HAStorage リソースの `hastorage-1` を `resource-group-1` に作成するには、75 ページの「リソースグループとディスクデバイスグループ間での起動の同期」を読み、その後次の手順を実行します。

HAStoragePlus リソースタイプを作成するには、81 ページの「HA ローカルファイルシステムの有効化」を参照してください。

1. クラスタメンバー上でスーパーユーザーになります。
2. リソースグループ `resource-group-1` を作成します。

```
# scrgadm -a -g resource-group-1
```

3. リソースタイプが登録されているかどうかを調べます。

次のコマンドは、登録されているリソースタイプのリストを出力します。

```
# scrgadm -p | egrep Type
```

4. 必要であれば、リソースタイプを登録します。

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

5. HAStorage リソースである `hastorage-1` を作成し、サービスパスを定義します。

```
# scrgadm -a -j hastorage-1 -g resource-group-1 -t SUNW.HAStorage \  
-x ServicePaths=/global/resource-group-1,/dev/global/dsk/d5s2,dsk/d6
```

ServicePaths には、次の値を含むことができます。

- グローバルデバイスグループ名 (例:nfs-dg)
- グローバルデバイスのパス (例:/dev/global/dsk/d5s2 または dev/d6)
- クラスタファイルシステムのマウントポイント (例:/global/nfs)

注 - ServicePaths にクラスタファイルシステムパスが含まれる場合、グローバルデバイスグループはそれらに対応するリソースグループと共に使用されない場合があります。

6. **hastorage-1** リソースを有効にします。

```
# scswitch -e -j hastorage-1
```

7. リソース **Sun Java System Web Server**、**Oracle**、**NFS** を **resource-group-1** に追加し、これらの依存性を **hastorage-1** に設定します。

たとえば、Sun Java System Web Server の場合、次のコマンドを実行します。

```
# scrgadm -a -jresource -g resource-group-1 -t SUNW.iws \  
-x Confdir_list=/global/iws/schost-1 -y Scalable=False \  
-y Network_resources_used=schost-1 -y Port_list=80/tcp \  
-y Resource_dependencies=hastorage-1
```

8. リソースの依存性を正しく構成したかを確認します。

```
# scrgadm -pvv -j resource |egrep strong
```

9. **resource-group-1** を **MANAGED** 状態に設定し、オンラインにします。

```
# scswitch -Z -g resource-group-1
```

HASStorage リソースタイプは、別の拡張プロパティ (AffinityOn) を含みます。この拡張プロパティは、HASStorage が ServicePaths で定義されているグローバルデバイスおよびクラスタファイルシステムのアフィニティスイッチオーバーを実行する必要があるかどうかを指定するブール値です。詳細は、SUNW.HASStorage(5) のマニュアルページを参照してください。

注 - リソースグループがスケラブルの場合、HASStorage と HASStoragePlus は AffinityOn が TRUE に設定されることを許可しません。スケラブルリソースグループについては、HASStorage と HASStoragePlus は AffinityOn 値をチェックし、この値を内部的に FALSE に設定し直します。

▼ 既存のリソース用に HAStorage リソースタイプを設定する

HAStorage は、今後の Sun Cluster でサポートされなくなる可能性があります。同等の機能が HAStoragePlus でサポートされています。HAStorage から HAStoragePlus へアップグレードするには、78 ページの「HAStorage から HAStoragePlus へのアップグレード」を参照してください。

既存のリソースのために HAStorage リソースを作成するには、75 ページの「リソースグループとディスクデバイスグループ間での起動の同期」を読み、その後以下の作業を行なってください。

1. リソースタイプが登録されているかどうかを調べます。
次のコマンドは、登録されているリソースタイプのリストを出力します。

```
# scrgadm -p | egrep Type
```

2. 必要であれば、リソースタイプを登録します。

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

3. HAStorage リソースである **hastorage-1** を作成します。

```
# scrgadm -a -g resource-group -j hastorage-1 -t SUNW.HAStorage \  
-x ServicePaths= ... -x AffinityOn=True
```

4. **hastorage-1** リソースを有効にします。

```
# scswitch -e -j hastorage-1
```

5. 必要に応じて既存の各リソースについて依存性を設定します。

```
# scrgadm -c -j resource -y Resource_Dependencies=hastorage-1
```

6. リソースの依存性を正しく構成したかを確認します。

```
# scrgadm -pvv -j resource |egrep strong
```

HAStorage から HAStoragePlus へのアップグレード

HAStorage は、今後の Sun Cluster でサポートされなくなる可能性があります。同等の機能が HAStoragePlus でサポートされています。HAStorage から HAStoragePlus へアップグレードする方法については、以下の節を参照してください。

デバイスグループまたは CFS を使用している場合に HAStorage から HAStoragePlus へアップグレードする

HAStorage は、今後の Sun Cluster でサポートされなくなる可能性があります。同等の機能が HAStoragePlus でサポートされています。デバイスグループまたは CFS を使用している場合に HAStorage から HAStoragePlus にアップグレードするには、以下の作業を行なってください。

この例では、HAStorage で単純な HA-NFS リソースが有効になっています。ServicePaths はディスクグループ nfsdg で、AffinityOn プロパティは TRUE です。さらに、この HA-NFS リソースは Resource_Dependencies を HAStorage リソースに設定しています。

1. HAStorage に対するアプリケーションリソースの依存性を除去します。

```
# scrgadm -c -j nfsserver-rs -y Resource_Dependencies=""
```

2. HAStorage リソースを無効にします。

```
# scswitch -n -j nfs1storage-rs
```

3. アプリケーションリソースグループから HAStorage リソースを削除します。

```
# scrgadm -r -j nfs1storage-rs
```

4. HAStorage リソースタイプの登録を解除します。

```
# scrgadm -r -t SUNW.HAStorage
```

5. HAStoragePlus リソースタイプを登録します。

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

6. HAStoragePlus リソースを作成します。

ファイルシステムのマウントポイントを指定するには、次のテキストを入力してください。

```
# scrgadm -a -j nfs1-hastp-rs -g nfs1-rg -t \  
SUNW.HAStoragePlus -x FilesystemMountPoints=/global/nfsdata -x \  
AffinityOn=True
```

グローバルデバイスパスを指定するには、次のテキストを入力してください。

```
# scrgadm -a -j nfs1-hastp-rs -g nfs1-rg -t \  
SUNW.HAStoragePlus -x GlobalDevicePaths=nfsdg -x AffinityOn=True
```

注 - HAStorage の ServicePaths プロパティではなく、HAStoragePlus の GlobalDevicePaths または FilesystemMountPoints プロパティを使用する必要があります。 FilesystemMountPoints 拡張プロパティは、/etc/vfstab で指定されたシーケンスと一致する必要があります。

7. HAStoragePlus リソースを有効にします。

```
# scswitch -e -j nfs1-hastp-rs
```

8. アプリケーションサーバーとHAStoragePlus との間の依存性を設定します。

```
# scrgadm -c -j nfsserver-rs -y \  
Resource_Dependencies=nfs1=hastp-rs
```

CFS による HAStorage からフェイルオーバー ファイルシステムによる HAStoragePlus へアップ グレードする

HAStorage は、今後の Sun Cluster でサポートされなくなる可能性があります。同等の機能が HAStoragePlus でサポートされています。CFS による HAStorage から Failover Filesystem (FFS) による HAStoragePlus にアップグレードするには、以下の作業を行なってください。

この例では、HAStorage で単純な HA-NFS リソースが有効になっています。ServicePaths はディスクグループ nfsdg で、AffinityOn プロパティは TRUE です。さらに、この HA-NFS リソースは Resource_Dependencies を HAStorage リソースに設定しています。

1. HAStorage リソースに対するアプリケーションリソースの依存性を除去します。

```
# scrgadm -c -j nfsserver-rs -y Resource_Dependencies=""
```

2. HAStorage リソースを無効にします。

```
# scswitch -n -j nfs1storage-rs
```

3. アプリケーションリソースグループから HAStorage リソースを削除します。

```
# scrgadm -r -j nfs1storage-rs
```

4. HAStorage リソースタイプの登録を解除します。

```
# scrgadm -r -t SUNW.HAStorage
```

5. /etc/vfstab を変更してグローバルフラグを削除し、「mount at boot」を「no」に変更します。

6. HAStoragePlus リソースを作成します。

ファイルシステムのマウントポイントを指定するには、次のテキストを入力してください。

```
# scrgadm -a -j nfs1-hastp-rs -g nfs1-rg -t \  
SUNW.HAStoragePlus -x FilesystemMountPoints=/global/nfsdata -x \  
AffinityOn=True
```

グローバルデバイスパスを指定するには、次のテキストを入力してください。

```
# scrgadm -a -j nfs1-hastp-rs -g nfs1-rg -t \  
SUNW.HAStoragePlus -x GlobalDevicePaths=nfsdg -x AffinityOn=True
```

注 - HAStorage の ServicePaths プロパティではなく、HAStoragePlus の GlobalDevicePaths または FilesystemMountPoints プロパティを使用する必要があります。 FilesystemMountPoints 拡張プロパティは、/etc/vfstab で指定されたシーケンスと一致する必要があります。

7. HAStoragePlus リソースを有効にします。

```
# scswitch -e -j nfs1-hastp-rs
```

8. アプリケーションサーバーとHAStoragePlus との間の依存性を設定します。

```
# scrgadm -c -j nfsserver-rs -y \  
Resource_Dependencies=nfs1=hastp-rs
```

HA ローカルファイルシステムの有効化

HAStoragePlus リソースタイプを使用すると、ローカルファイルシステムを Sun Cluster 環境内で高可用性にすることができます。このためには、ローカルファイルシステムのパーティションがグローバルディスクグループに存在し、アフィニティスイッチオーバーが有効であり、Sun Cluster 環境がフェイルオーバー用に構成されている必要があります。これによって、多重ホストディスクに直接接続された任意のホストから、多重ホストディスク上の任意のファイルシステムにアクセスできるようになります。(HAStoragePlus では、ルートファイルシステムを高可用性にすることはできません)。フェイルバック設定は、リソースグループとデバイスグループで同一にする必要があります。

入出力の多いデータサービスの中には、HA ローカルファイルシステムの使用が強く望まれるものがあります。このため、このようなデータサービスの登録作業と構成作業には、HAStoragePlus リソースタイプを構成する方法が追加されています。これらのデータサービスの HAStoragePlus リソースタイプを設定する手順については、以下の節を参照してください。

- 『Sun Cluster Data Service for Oracle ガイド (Solaris OS 版)』の「Sun Cluster HA for Oracle の登録と構成」
- 『Sun Cluster Data Service for Sybase ASE ガイド (Solaris OS 版)』の「Sun Cluster HA for Sybase ASE の登録と構成」

ほかのデータサービスの HASToragePlus リソースタイプを設定する方法については、82 ページの「HASToragePlus リソースタイプを設定する」を参照してください。

▼ HASToragePlus リソースタイプを設定する

HASToragePlus リソースタイプは Sun Cluster 3.0 5/02 で導入されています。この新しいリソースタイプは、HASTorage と同じ機能を果たし、リソースグループとディスクデバイスグループ間で起動を同期します。HASToragePlus リソースタイプには、ローカルファイルシステムを高可用性にするための機能が追加されています。(ローカルファイルシステムの可用性を高めるための背景情報については、81 ページの「HA ローカルファイルシステムの有効化」を参照してください)。これら 2 つの機能を両方とも使用するには、HASToragePlus リソースタイプを設定します。

HASToragePlus を設定するには、ローカルファイルシステムのパーティションがグローバルディスクグループに存在し、アフィニティスイッチオーバーが有効であり、かつ Sun Cluster 環境がフェイルオーバー用に構成されている必要があります。

次の例では、簡単な NFS サービスを使用して、ローカルにマウントされたディレクトリ `/global/local-fs/nfs/export/` からホームディレクトリのデータを共有します。この例では、次の条件を前提にしています。

- マウントポイント `/global/local-fs/nfs` は、UFS ローカルファイルシステムを Sun Cluster グローバルデバイスのパーティションにマウントするために使用されます。
- `/global/local-fs/nfs` ファイルシステムの `/etc/vfstab` エントリには、このファイルシステムがローカルファイルシステムで、マウントブートフラグが「no」であるよう指定されている必要があります。
- PathPrefix ディレクトリ (HA-NFS が管理情報と状態情報を保守するために使用するディレクトリ) は、マウントするファイルシステムのルートディレクトリ (たとえば、`/global/local-fs/nfs`) 上に存在します。

1. クラスタメンバー上でスーパーユーザーになります。
2. リソースタイプが登録されているかどうかを調べます。
次のコマンドは、登録されているリソースタイプのリストを出力します。

```
# scrgadm -p | egrep Type
```

3. 必要であれば、リソースタイプを登録します。

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

4. フェイルオーバーリソースグループである **nfs-r** を作成します。

```
# scrgadm -a -g nfs-rg -y PathPrefix=/global/local-fs/nfs
```

5. タイプ **SUNW.LogicalHostname** の論理ホストリソースを作成します。

```
# scrgadm -a -j nfs-lh-rs -g nfs-rg -L -l log-nfs
```

6. クラスタに **HASStoragePlus** リソースタイプを登録します。

```
# scrgadm -a -tSUNW.HASStoragePlus
```

7. タイプ **HASStoragePlus** のリソース **nfs-hastp-rs** を作成します。

```
# scrgadm -a -j nfs-hastp-rs -g nfs-rg -t SUNW.HASStoragePlus\  
-x FilesystemMountPoints=/global/local-fs/nfs \  
-x AffinityOn=TRUE
```

注 - FilesystemMountPoints 拡張プロパティは、1つ以上のファイルシステムマウントポイントをリストの形式で指定するために使用できます。このリストには、ローカルファイルシステムマウントポイントとグローバルファイルシステムマウントポイントの両方を含めることができます。ブートフラグでのマウントは、グローバルファイルシステムの **HASStoragePlus** によって無視されます。

8. リソースグループ **nfs-rg** をクラスタノード上でオンラインにします。

このノードは、/global/local-fs/nfs ファイルシステムの実際のグローバルデバイスのパーティション用の稼働系になります。次に、ファイルシステム /global/local-fs/nfs は当該ノード上にローカルにマウントされます。

```
# scswitch -Z -g nfs-rg
```

9. **SUNW.nfs** リソースタイプをクラスタに登録します。タイプ **SUNW.nfs** のリソース **nfs-rs** を作成して、リソース **nfs-hastp-rs** へのリソース依存関係を指定します。

dfstab.nfs-rs が /global/local-fs/nfs/SUNW.nfs に作成されます。

```
# scrgadm -a -t SUNW.nfs  
# scrgadm -a -g nfs-rg -j nfs-rs -t SUNW.nfs \  
-y Resource_dependencies=nfs-hastp-rs
```

注 - nfs リソースに依存関係を設定するには、nfs-hastp-rs リソースがオンラインである必要があります。

10. リソース **nfs-rs** をオンラインにします。

```
# scswitch -Z -g nfs-rg
```



注意 - 切り替えは、リソースグループレベルに限定して行なってください。デバイスグループレベルで切り替えを行うと、リソースグループが混乱し、フェイルオーバーが発生します。

これで、サービスを新しいノードに移行するときには常に、/global/local-fs/nfs 用のプライマリ入出力パスはオンラインになり、NFS サーバーに配置されます。ファイルシステム /global/local-fs/nfs は NFS サーバーが起動する前にローカルにマウントされます。

プライオリティが低いリソースグループをオフロードすることによるノードリソースの解放

プライオリティ付きサービス管理 (RGOffload) を使用すると、プライオリティが高いデータサービス用にノードのリソースを自動的に解放できます。RGOffload は、プライオリティが高いフェイルオーバーデータサービスを起動するために、プライオリティが低いスケラブルデータサービスまたはフェイルオーバーデータサービスをオフラインにする必要があるときに使用します。RGOffload は、プライオリティが低いデータサービスを含むリソースグループをオフロードするときに使用します。

注 - プライオリティが高いデータサービスはフェイルオーバー可能でなければなりません。オフロードするデータサービスは、フェイルオーバーデータサービスでもスケラブルデータサービスでもかまいません。

▼ RGOffload リソースを設定する

1. クラスタメンバー上でスーパーユーザーになります。
2. **RGOffload** リソースタイプが登録されているかどうかを調べます。
次のコマンドは、リソースタイプのリストを出力します。

```
# scrgadm -p|egrep SUNW.RGOffload
```

3. 必要であれば、リソースタイプを登録します。

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

4. **RGOffload** リソースでオフロードするリソースグループごとに、**Desired primaries** をゼロに設定します。

```
# scrgadm -c -g offload-rg -y Desired primaries=0
```

5. **RGOffload** リソースをプライオリティが高いフェイルオーバーリソースグループに追加して、拡張プロパティを設定します。

リソースグループを複数のリソースの **rg_to_offload** リストに追加してはいけません。リソースグループを複数の **rg_to_offload** リストに追加すると、リソースグループはオフラインになったあとにオンラインになるという動作を繰り返すこととなります。

拡張プロパティについては、87 ページの「**RGOffload** 拡張プロパティを構成する」を参照してください。

```
# scrgadm -aj rgoffload-resource \  
-t SUNW.RGOffload -g critical-rg \  
-x rg_to_offload=offload-rg-1,offload-rg-2,... \  
-x continue_to_offload=TRUE \  
-x max_offload_retry=15
```

注 - この場合、**rg_to_offload** 以外の拡張プロパティはデフォルト値で表示されます。**rg_to_offload** は、お互いに依存しないリソースグループをコンマで区切ったリストです。このリストには、**RGOffload** リソースを追加するリソースグループを含めることはできません。

6. **RGOffload** リソースを有効にします。

```
# scswitch -ej rgoffload-resource
```

7. プライオリティが高いフェイルオーバーリソースから **RGOffload** への依存関係を設定します。

```
# scrgadm -c -j critical-resource \  
-y Resource_dependencies=rgoffload-resource
```

Resource_dependencies_weak も使用できます。

Resource_dependencies_weak を **RGOffload** リソースタイプに使用すると、**offload-rg** のオフロード中にエラーが発生しても、プライオリティが高いフェイルオーバーリソースを起動できます。

8. オフロードするリソースグループを、オンラインにします。

```
# scswitch -z -g offload-rg,offload-rg-2,... -h [nodelist]
```

リソースグループは、プライオリティが高いリソースグループがオフラインであるすべてのノード上でオンラインのままになります。フォルトモニターは、プライオリティが高いリソースグループがオンラインであるノード上でリソースグループが動作しないようにします。

オフロードするリソースグループの `Desired primaries` はゼロに設定されているので(手順4を参照)、“-Z” オプションを指定しても、このようなリソースグループはオンラインになりません。

9. プライオリティが高いフェイルオーバーリソースグループがオンラインでない場合、オンラインにします。

```
# scswitch -Z -g critical-rg
```

SPARC: 例 – RGOffload リソースを構成する

この例では、RGOffload リソース (`rgof1`)、RGOffload リソースを含むプライオリティが高いリソースグループ (`oracle_rg`)、およびプライオリティが高いリソースグループがオンラインになったときにオフロードされるスケラブルリソースグループ (`IWS-SC`, `IWS-SC-2`) を構成する方法について説明します。この例では、プライオリティが高いリソースは `oracle-server-rs` です。

この例では、`oracle_rg`、`IWS-SC`、および `IWS-SC-2` はクラスタ "triped" の任意のノード、つまり、`phys-triped-1`、`phys-triped-2`、`phys-triped-3` 上でマスターできます。

[`SUNW.RGOffload` リソースタイプが登録されているかどうかを判断する]

```
# scrgadm -p | egrep SUNW.RGOffload
```

[必要に応じて、リソースタイプを登録する]

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

[RGOffload によってオフロードされる各リソースグループで、`Desired primaries` をゼロに設定する]

```
# scrgadm -c -g IWS-SC-2 -y Desired primaries=0
```

```
# scrgadm -c -g IWS-SC -y Desired primaries=0
```

[プライオリティが高いリソースグループに RGOffload リソースを追加し、拡張プロパティを設定する]

```
# scrgadm -aj rgof1 -t SUNW.RGOffload -g oracle_rg \  
-x rg_to_offload=IWS-SC,IWS-SC-2 -x continue_to_offload=TRUE \  
-x max_offload_retry=15
```

[RGOffload リソースを有効にする]

```
# scswitch -ej rgof1
```

[プライオリティが高いフェイルオーバーリソースの RGOffload リソースに対する依存性を設定する]

```
# scrgadm -c -j oracle-server-rs -y Resource_dependencies=rgof1
```

[オフロードされるリソースグループをすべてのノードでオンラインにする]

```
# scswitch -z -g IWS-SC,IWS-SC-2 -h phys-triped-1,phys-triped-2,phys-triped-3
```

[プライオリティが高いフェイルオーバーリソースグループがオンラインでない場合は、それをオンラインにする]

```
# scswitch -Z -g oracle_rg
```

RGOffload 拡張プロパティを構成する

通常、RGOffload リソースを作成するとき、拡張プロパティを構成するには、コマンド行 `scrgadm -x parameter=value` を使用します。Sun Cluster のすべての標準プロパティの詳細は、付録 A を参照してください。

表 2-2 に RGOffload に設定できる拡張プロパティを示します。「調整」の欄は、各プロパティをいつ更新できるかを示しています。

表 2-2 RGOffload 拡張プロパティ

名前/データタイプ	デフォルト値
<code>rg_to_offload</code> (文字列)	<p>プライオリティが高いフェイルオーバーリソースグループがノード上で起動するとき、当該ノード上でオフロードする必要があるリソースグループをコンマで区切ったリスト。このリストには、互いに依存するリソースグループが含まれてはいけません。このプロパティにはデフォルト設定値がないので、必ず設定する必要があります。</p> <p>RGOffload は、<code>rg_to_offload</code> 拡張プロパティに設定されたリソースグループのリストにおける依存関係ループを検査しません。たとえば、リソースグループ RG-B が RG-A に依存する場合、RG-A と RG-B が両方とも <code>rg_to_offload</code> に含まれてはいけません。</p> <p>デフォルト: なし</p> <p>調整: 任意の時点</p>
<code>continue_to_offload</code> (ブール型)	<p>リソースグループのオフロード中にエラーが発生した後に、<code>rg_to_offload</code> リスト内の残りのリソースグループをオフロードし続けるかどうかを示すブール型。</p> <p>このプロパティは START メソッドだけが使用します。</p> <p>デフォルト: True</p> <p>調整: 任意の時点</p>

表 2-2 RGOffload 拡張プロパティ (続き)

名前/データタイプ	デフォルト値
max_offload_retry (整数)	<p>クラスタまたはリソースグループの再構成による障害時の起動中に、リソースグループをオフロードしようとする回数。再試行の間には 10 秒の間隔があります。</p> <p>(オフロードされるリソースグループの数 * max_offload_retry * 10 秒) が</p> <p>RGOffload リソースの Start_timeout よりも小さくなるように</p> <p>max_offload_retry を設定します。この値が Start_timeout の値に近い (あるいは、より大きい) 場合、最大再試行数に到達する前に、RGOffload リソースの START メソッドがタイムアウトする可能性があります。</p> <p>このプロパティは START メソッドだけが使用します。</p> <p>デフォルト: 15</p> <p>調整: 任意の時点</p>

フォルトモニター

RGOffload リソースのフォルトモニター検証は、プライオリティが高いリソースをマスターするノード上で、rg_to_offload 拡張プロパティに指定されたリソースグループをオフラインにし続けるために使用されます。各検証サイクルでフォルトモニターは、プライオリティが高いリソースをマスターするノード上で、オフロードされるリソースグループ (offload-rg) がオフラインであることを確認します。プライオリティが高いリソースをマスターするノード上で offload-rg がオンラインである場合、フォルトモニターは重要なリソースをマスターするノード以外のノード上で offload-rg を起動し、同時に、プライオリティが高いリソースをマスターするノード上では offload-rg をオフラインにしようとします。

offload-rg の desired primaries はゼロに設定されているので、この後で利用可能になったノード上では、オフロードするリソースグループは再起動されません。したがって、RGOffload フォルトモニターは maximum primaries に到達するまで、重要なリソースをマスターするノード上では offload-rg をオフラインにしなから、可能な限りのプライマリ上で offload-rg を起動しようとします。

RGOffload は、MAINTENANCE 状態または UNMANAGED 状態でないかぎり、オフロードされたすべてのリソースを起動しようとします。リソースグループを UNMANAGED 状態にするには、scswitch コマンドを使用します。

```
# scswitch -u -g resourcegroup
```


フォルトモニター検証サイクルは、Thorough_probe_interval が実行されたあとにか
ならず呼び出されます。

付録 A

標準プロパティ

この付録では、標準リソースタイプ、リソースグループ、リソースプロパティについて説明します。また、システム定義プロパティの変更および拡張プロパティの作成に使用するリソースプロパティ属性についても説明します。

この章の内容は次のとおりです。

- 91 ページの「リソースタイププロパティ」
- 98 ページの「リソースプロパティ」
- 111 ページの「リソースグループプロパティ」
- 118 ページの「リソースプロパティの属性」

注 - True や False などのプロパティ値は、大文字と小文字は区別されません。

リソースタイププロパティ

表 A-1 に、Sun Cluster で定義されているリソースタイププロパティの内容を示します。プロパティ値は以下のように分類されます。

- 必須 — Resource Type Registration (RTR) ファイル内に利用値を必要とするプロパティです。値がない場合は、プロパティが属するオブジェクトを作成できません。空白文字または空の文字列を値として指定することはできません。
- 条件付 — プロパティが存在するためには、RTR ファイル内で宣言する必要があります。宣言されていない場合は、Resource Group Manager (RGM) はプロパティを作成しないため、管理ユーティリティで利用できません。空白文字または空の文字列を値として指定できます。プロパティが RTR ファイル内で宣言されており、値が指定されていない場合には、RGM はデフォルト値を使用します。

- 条件付 / 明示 — プロパティが存在するためには、明示的に値を指定して、RTR ファイル内で宣言する必要があります。宣言されていない場合は、RGM はこのプロパティを作成しないため、管理ユーティリティで利用できません。空白文字または空の文字列を値として指定することはできません。
- 任意 — プロパティは RTR ファイル内で宣言できます。宣言されていない場合は、RGM がプロパティを作成し、デフォルト値を適用します。プロパティが RTR ファイル内で宣言されており、値が指定されていない場合は、RGM は、プロパティが RTR ファイル内で宣言されないときのデフォルト値と同じ値を使用します。

リソースタイププロパティは、管理ユーティリティで更新することはできません。ただし、`Installed_nodes` は例外です。`Installed_nodes` は RTR ファイル内に宣言できないため、管理者が設定する必要があります。

表 A-1 リソースタイププロパティ

プロパティ名	説明
API_version (整数)	このリソース型の実装が使用するリソース管理 API のバージョンです。 カテゴリ: 任意 デフォルト: 2 調整: 不可
BOOT (文字列)	任意のコールバックメソッド。ノード上で RGM が起動するプログラムへのパス。このプログラムは、このタイプのリソースがすでに管理状態にあるときに、クラスタの結合または再結合を行います。このメソッドは、INIT メソッドと同様に、このタイプのリソースの初期化を行う必要があります。 カテゴリ: 条件付 / 明示 デフォルト: なし 調整: 不可
Failover (ブール値)	True は、複数のノード上で同時にオンラインになることのできる任意のグループで、このタイプのリソースを構成できないことを示します。 カテゴリ: 任意 デフォルト: False 調整: 不可

表 A-1 リソースタイププロパティ (続き)

プロパティ名	説明
FINI (文字列)	<p>任意のコールバックメソッド。RGM 管理からこのタイプのリソースを削除するときに RGM が起動するプログラムへのパス。</p> <p>カテゴリ: 条件付 / 明示</p> <p>デフォルト: なし</p> <p>調整: 不可</p>
INIT (文字列)	<p>任意のコールバックメソッド。このタイプのリソースが RGM によって管理されるようになったときに、RGM が起動するプログラムへのパス。</p> <p>カテゴリ: 条件付 / 明示</p> <p>デフォルト: なし</p> <p>調整: 不可</p>
Init_nodes (列挙)	<p>RGM が INIT、FINI、BOOT、VALIDATE の各メソッドを呼び出すノードを指定します。値には、RG primaries (リソースをマスターできるノードだけ)、または RT installed_nodes (リソースタイプがインストールされるすべてのノード) を指定できます。</p> <p>カテゴリ: 任意</p> <p>デフォルト: RG primaries</p> <p>調整: 不可</p>
Installed_nodes (文字配列)	<p>リソースタイプを実行することができるクラスタノードのリストです。RGM は、自動的にこのプロパティを作成します。クラスタ管理者は値を設定できます。RTR ファイル内には宣言できません。</p> <p>カテゴリ: クラスタ管理者は構成可能</p> <p>デフォルト: 全てのクラスタノード</p> <p>調整: 任意の時点</p>

表 A-1 リソースタイププロパティ (続き)

プロパティ名	説明
Monitor_check (文字列)	<p>任意のコールバックメソッド。このタイプのリソースのフォルトモニターが要求するフェイルオーバーを行う前に、RGM が起動するプログラム。</p> <p>カテゴリ: 条件付 / 明示</p> <p>デフォルト: なし</p> <p>調整: 不可</p>
Monitor_start (文字列)	<p>任意のコールバックメソッド。このタイプのリソースのフォルトモニターを起動するために、RGM がアクティブにするプログラムへのパス。</p> <p>カテゴリ: 条件付 / 明示</p> <p>デフォルト: なし</p> <p>調整: 不可</p>
Monitor_stop (文字列)	<p>Monitor_start が設定されている場合の、必須のコールバックメソッド。このタイプのリソースのフォルトモニターを停止するために、RGM がアクティブにするプログラムへのパス。</p> <p>カテゴリ: 条件付 / 明示</p> <p>デフォルト: なし</p> <p>調整: 不可</p>
各クラスターノード上の Num_resource_restart (整数)。	<p>このプロパティは、RGM によって、このノード上のこのリソースに対して過去 n 秒間 (n はリソースの Retry_interval プロパティの値) に実行された scha_control RESTART 呼び出しの回数に設定されます。リソースタイプが Retry_interval プロパティを宣言していない場合、この型のリソースは Num_resource_restarts プロパティを使用できません。</p>
Pkglist (文字配列)	<p>リソース型のインストールに含まれている任意のパッケージリストです。</p> <p>カテゴリ: 条件付 / 明示</p> <p>デフォルト: なし</p> <p>調整: 不可</p>

表 A-1 リソースタイププロパティ (続き)

プロパティ名	説明
Postnet_stop (文字列)	<p>任意のコールバックメソッド。このタイプのリソースが依存する任意のネットワークアドレスリソース (Network_resources_used) の STOP メソッドを呼び出したあとで、RGM が起動するプログラムへのパス。このメソッドは、ネットワークインタフェースの停止設定に続いて、必要な Stop アクションを行います。</p> <p>カテゴリ: 条件付 / 明示</p> <p>デフォルト: なし</p> <p>調整: 不可</p>
Prenet_start (文字列)	<p>任意のコールバックメソッド。このタイプのリソースが依存する、任意のネットワークアドレスリソース (Network_resources_used) の START メソッドを呼び出す前に、RGM が起動するプログラムへのパス。ネットワークインタフェースが起動に構成される前に必要な Start アクションを行う必要があります。</p> <p>カテゴリ: 条件付 / 明示</p> <p>デフォルト: なし</p> <p>調整: 不可</p>
RT_basedir (文字列)	<p>コールバックメソッドの相対パスのを補完するディレクトリパスです。リソース型パッケージのインストール先のパスになります。スラッシュ (/) で開始する完全なパスを指定する必要があります。すべてのメソッドパス名が絶対パスの場合は、指定しなくてもかまいません。</p> <p>カテゴリ: 必須 (絶対パスではないメソッドパスがある場合)</p> <p>デフォルト: なし</p> <p>調整: 不可</p>
RT_description (文字列)	<p>リソースタイプの簡単な説明です。</p> <p>カテゴリ: 条件付</p> <p>デフォルト: 空の文字列</p> <p>調整: 不可</p>

表 A-1 リソースタイププロパティ (続き)

プロパティ名	説明
Resource_type (文字列)	<p>リソースタイプの名前。</p> <p>現在登録されているリソースタイプ名を表示するには、次のコマンドを使用します。</p> <pre>scrgadm -p</pre> <p>Sun Cluster 3.1 以降、リソースタイプ名は次の形式をとります。</p> <pre>vendor_id.resource_type:version</pre> <p>リソースタイプ名の3つのコンポーネントは、<i>Vendor_id</i>、<i>Resource_type</i>、および <i>RT_version</i> として RTR ファイル内で指定されています。 <code>scrgadm</code> コマンドはピリオドおよびコロン区切りを挿入します。リソースタイプ名の最後の部分、<i>RT_version</i> には、<i>RT_version</i> プロパティと同じ値が入りません。</p> <p>重複を防ぐため、<i>Vendor_id</i> には、リソース型の作成元の会社のストックシンボルを使用することをお勧めします。</p> <p>Sun Cluster 3.1 以前に作成されたリソースタイプ名は次の形式をとります。</p> <pre>vendor_id.resource_type</pre> <p>カテゴリ: 必須</p> <p>デフォルト: 空の文字列</p> <p>調整: 不可</p>
RT_version (文字列)	<p>Sun Cluster 3.1 以降、このリソースタイプの実装に必要なバージョンを指定します。</p> <p><i>RT_version</i> は、完全なリソースタイプ名の末尾の部分です。</p> <p>カテゴリ: 条件付 / 明示</p> <p>デフォルト: なし</p> <p>調整: 不可</p>

表 A-1 リソースタイププロパティ (続き)

プロパティ名	説明
Single_instance (ブール値)	<p>True の場合は、このタイプのリソースがクラスタ内に 1 つだけ存在できることを指定します。したがって、RGM は、同時に 1 つのこのリソースタイプだけに、クラスタ全体に渡っての実行を許可します。</p> <p>カテゴリ: 任意</p> <p>デフォルト: False</p> <p>調整: 不可</p>
START (文字列)	<p>コールバックメソッド。このタイプのリソースを開始するために RGM がアクティブにするプログラムへのパス。</p> <p>カテゴリ: 必須 (RTR ファイルで PRENET_START メソッドが宣言されていない場合)</p> <p>デフォルト: なし</p> <p>調整: 不可</p>
STOP (文字列)	<p>コールバックメソッド。このタイプのリソースを停止するために RGM がアクティブにするプログラムへのパス。</p> <p>カテゴリ: 必須 (RTR ファイルで POSTNET_STOP メソッドが宣言されていない場合)</p> <p>デフォルト: なし</p> <p>調整: 不可</p>
UPDATE (文字列)	<p>任意のコールバックメソッド。実行中のこのタイプのリソースのプロパティが変更された場合に、RGM が起動するプログラムへのパス。</p> <p>カテゴリ: 条件付 / 明示</p> <p>デフォルト: なし</p> <p>調整: 不可</p>
VALIDATE (文字列)	<p>任意のコールバックメソッド。このタイプのリソースのプロパティ値を検査するために呼び出すプログラムへのパス。</p> <p>カテゴリ: 条件付 / 明示</p> <p>デフォルト: なし</p> <p>調整: 不可</p>

表 A-1 リソースタイププロパティ (続き)

プロパティ名	説明
Vendor_ID (文字列)	Resource_type を参照してください。 カテゴリ: 条件付 デフォルト: なし 調整: 不可

リソースプロパティ

表 A-2 に、Sun Cluster で定義されているリソースタイププロパティの内容を示します。この説明は、データサービスの開発者を対象としたものです。特定のデータサービスの詳細は、そのデータサービスのマニュアルページを参照してください。リソースプロパティ値は、以下のように分類されます。

- 必須 — 管理者は、管理ユーティリティでリソースを作成するときに、必ず値を指定する必要があります。
- 任意 — 管理者がリソースグループの作成時に値を指定しない場合、システムがデフォルト値を提供します。
- 条件付 — プロパティが RGM ファイルで宣言されている場合にのみ、RGM がプロパティを作成します。宣言されていない場合プロパティは存在せず、システム管理者はこれを利用できません。RTR ファイルで宣言されている条件付きのプロパティは、デフォルト値が RTR ファイル内で指定されているかどうかによって、必須または任意になります。詳細については、各条件付きプロパティの説明を参照してください。
- 照会のみ — 管理ツールから直接設定できません。

表 A-2 は、リソースプロパティの調整が可能であるか、および、いつ調整できるかも示しています。

None または False	不可
True または Anytime	任意の時点
At_creation	リソースをクラスタに追加するとき
When_disabled	リソースが無効なとき

表 A-2 リソースプロパティ

プロパティ名	説明
<p>Affinity_timeout (整数)</p>	<p>クライアントが、スティックスケーラブルサービスへの接続をすべて閉じたあと、IP アフィニティ (IP 対応) が持続する秒数を制御します。このタイムアウトが終了するまでは、当該クライアントからの新しい接続はすべて同じクラスタノードに転送されます。値が 0 の場合、クライアントとスケラブルサービスとの間に接続が存在するかぎり、アフィニティは継続します。値が -1 の場合、クライアントがバインドされているクラスタノードに障害が起きるか、または再起動されるまで、アフィニティは持続します。値 -1 は、タイムアウト期間が適用されないことを意味します。</p> <p>このプロパティは Load_balancing_policy が LB_STICKY または LB_STICKYWILD である場合にしか適用されません。</p> <p>カテゴリ: 任意 デフォルト: 0 調整: 任意の時点</p>
<p>Cheap_probe_interval (整数)</p>	<p>リソースの即時障害検証の呼び出しの間隔 (秒数)。このプロパティは、RGM のみが作成でき、RTR ファイル内で宣言されている場合には、管理者はこのプロパティを利用できません。</p> <p>RTR ファイル内でデフォルト値が指定されている場合、このプロパティは任意です。リソースタイプファイル内に Tunable 属性が指定されていない場合、このプロパティの Tunable 値は When_disabled になります。</p> <p>RTR ファイルのプロパティ宣言内に Default 属性が指定されていない場合、このプロパティは必須です。</p> <p>カテゴリ: 条件付 デフォルト: 上記を参照 調整: 無効時</p>

表 A-2 リソースプロパティ (続き)

プロパティ名	説明
拡張プロパティ	<p>開発者は、クラスタ管理者がデータサービスを Sun Cluster に登録するときにデータサービスの初期構成内でリソースタイププロパティを宣言します。拡張プロパティに設定可能な各属性については、表 A-4 を参照してください。表 A-4 を参照してください。</p> <p>カテゴリ: 条件付</p> <p>デフォルト: なし</p> <p>調整: 特定のプロパティに依存</p>
Failover_mode (列挙)	<p>None、Soft、Hard のいずれかの設定が可能です。リソースでの START または STOP メソッドの呼び出しの失敗に対して、RGM がリソースグループを再配置するか、またはノードを異常終了させるかを制御します。None は、RGM が単にリソース状態をメソッド失敗に設定し、オペレータの介入を待つことを示します。Soft は、START メソッドが失敗したときに、RGM がリソースのグループを別のノードに再配置し、また、STOP メソッドが失敗したときに、RGM がリソース状態を設定し、システム管理者の介入を待つことを示します。Hard は、START メソッドが失敗したときに、グループの再配置を行い、STOP メソッドが失敗したときに、クラスタノードを異常終了させることで、リソースの強制的な停止を行うことを示します。</p> <p>カテゴリ: 任意</p> <p>デフォルト: なし</p> <p>調整: 任意の時点</p>

表 A-2 リソースプロパティ (続き)

プロパティ名	説明
Load_balancing_policy (文字列)	<p>使用する負荷均衡ポリシーを定義する文字列。このプロパティは、スケーラブルサービス専用です。RTR ファイルに Scalable プロパティが宣言されている場合、RGM は自動的にこのプロパティを作成します。</p> <p>Load_balancing_policy には次の値を設定できます。</p> <p>Lb_weighted (デフォルト)。 Load_balancing_weights プロパティに設定されている重みにより、さまざまなノードに負荷が分散されます。</p> <p>Lb_sticky。スケーラブルサービスの指定のクライアント (クライアントの IP アドレスで識別される) は、常に同じクラスタノードに送信されます。</p> <p>Lb_sticky_wild。指定のクライアント (クライアントの IP アドレスで識別される) はワイルドカードスティッキサービスの IP アドレスに接続され、送信時に使用されるポート番号とは無関係に、常に同じクラスタノードに送信されます。</p> <p>カテゴリ: 条件付 / 任意 デフォルト: Lb_weighted 調整: 作成時</p>

表 A-2 リソースプロパティ (続き)

プロパティ名	説明
Load_balancing_weights (文字配列)	<p>このプロパティは、スケーラブルサービス専用です。RTR ファイルに Scalable プロパティが宣言されている場合、RGM は自動的にこのプロパティを作成します。形式は、「weight@node,weight@node」になります。ここで、weight は、指定したノード (node) に対する負荷分散の相対的な割り当てを示す整数になります。ノードに分散される負荷の割合は、すべてのウエイトの合計でこのノードのウエイトを割った値になります。たとえば、1@1,3@2 は、ノード 1 に負荷の 1/4 が割り当てられ、ノード 2 に負荷の 3/4 が割り当てられることを意味します。デフォルトの空の文字列 ("") は、一定の分散を指定します。明示的にウエイトを割り当てられていないノードのウエイトは、デフォルトで 1 になります。</p> <p>Tunable 属性がリソースタイプファイルに指定されていない場合は、プロパティの Tunable 値は Anytime (任意の時点) になります。このプロパティを変更すると、新しい接続時のみ分散が変更されます。</p> <p>カテゴリ: 条件付 / 任意</p> <p>デフォルト: 空の文字列</p> <p>調整:任意の時点</p>
各コールバックメソッド用の method_timeout (整数)	<p>RGM がメソッドの呼び出しに失敗したと判断するまでの時間 (秒)。</p> <p>カテゴリ: 条件付 / 任意</p> <p>デフォルト: メソッド自体が RTR ファイルで宣言されている場合、3,600 (1 時間) です。</p> <p>調整:任意の時点</p>

表 A-2 リソースプロパティ (続き)

プロパティ名	説明
Monitored_switch (列挙)	<p>クラスタ管理者が管理ユーティリティを使用してモニターを有効または無効にすると、RGM によって Enabled または Disabled に設定されます。Disabled に設定されると、再び有効に設定されるまで、モニターは START メソッドを呼び出しません。リソースが、モニターのコールバックメソッドを持っていない場合は、このプロパティは存在しません。</p> <p>カテゴリ: 照会のみ</p> <p>デフォルト: Enabled</p> <p>調整: 不可</p>
Network_resources_used (文字配列)	<p>リソースが使用する論理ホスト名または共有アドレスネットワークリソースをコンマで区切ったリストです。スケラブルサービスの場合、このプロパティは別のリソースグループに存在する共有アドレスリソースを参照する必要があります。フェイルオーバーサービスの場合、このプロパティは同じリソースグループに存在する論理ホスト名または共有アドレスを参照します。RTR ファイルに Scalable プロパティが宣言されている場合、RGM は自動的にこのプロパティを作成します。Scalable が RTR ファイルで宣言されていない場合、Network_resources_used は RTR ファイルで明示的に宣言されていない限り使用できません。</p> <p>リソースタイプファイルに Tunable 属性が指定されていない場合、プロパティの Tunable 値は、At_creation になります。</p> <p>カテゴリ: 条件付 / 必須</p> <p>デフォルト: なし</p> <p>調整: 作成時</p>

表 A-2 リソースプロパティ (続き)

プロパティ名	説明
On_off_switch (列挙)	<p>クラスタ管理者が管理ユーティリティを使用してリソースを有効または無効にすると、RGM によって Enabled または Disabled に設定されます。無効に設定されると、再び有効に設定されるまで、リソースはコールバックを呼び出しません。</p> <p>カテゴリ: 照会のみ</p> <p>デフォルト: Disabled</p> <p>調整: 不可</p>
Port_list (文字配列)	<p>サーバーが待機するポート番号をコンマで区切ったリストです。各ポート番号に、そのポートが使用しているプロトコルが追加されます (例: Port_list=80/tcp)。Scalable プロパティが RTR ファイルで宣言されている場合、RGM は自動的に Port_list を作成します。それ以外の場合、このプロパティは RTR ファイルで明示的に宣言されていない限り使用できません。</p> <p>Apache 用にこのプロパティを設定する場合は、『Sun Cluster Data Service for Apache ガイド (Solaris OS 版)』を参照してください。</p> <p>カテゴリ: 条件付 / 必須</p> <p>デフォルト: なし</p> <p>調整: 作成時</p>
R_description (文字列)	<p>リソースの簡単な説明です。</p> <p>カテゴリ: 任意</p> <p>デフォルト: 空の文字列</p> <p>調整: 任意の時点</p>

表 A-2 リソースプロパティ (続き)

プロパティ名	説明
Resource_dependencies (文字配列)	<p>このリソースをオンラインにするために、順にオンラインにする必要のある同じグループ内のリソースをコンマで区切ったリストです。リスト内の任意のリソースの起動に失敗した場合、このリソースは起動されません。グループをオフラインにすると、このリソースを停止してから、リスト内のリソースが停止されます。このリソースが先に無効にならなければ、リスト内のリソースは無効にできません。</p> <p>カテゴリ: 任意 デフォルト: 空のリスト 調整: 任意の時点</p>
Resource_dependencies_weak (文字配列)	<p>グループ内のメソッド呼び出しの順序を決定する同じグループ内のリソースのリスト。RGM は、このリスト内のリソースの Start メソッドを先に呼び出してから、このリソースの Start メソッドを呼び出します。また、停止する場合は、このリソースの Stop メソッドを先に呼び出してから、リスト内のリソースの Stop メソッドを呼び出します。リスト内のリソースが開始に失敗した場合、または無効になっても、リソースはオンラインを維持できます。</p> <p>カテゴリ: 任意 デフォルト: 空のリスト 調整: 任意の時点</p>
Resource_name (文字列)	<p>リソースインスタンスの名前です。クラスタ構成内で一意にする必要があります。リソースが作成された後で変更はできません。</p> <p>カテゴリ: 必須 デフォルト: なし 調整: 不可</p>

表 A-2 リソースプロパティ (続き)

プロパティ名	説明
Resource_project_name (文字列)	<p>リソースに関連する Solaris プロジェクト名。このプロパティは、CPU の共有、クラスタデータサービスのリソースプールといった Solaris のリソース管理機能に適用できます。RGM は、リソースをオンラインにすると、このプロジェクト名を持つ関連プロセスを起動します。このプロパティを指定しないと、プロジェクト名はリソースを含んでいるリソースグループの RG_project_name プロパティから取得されます。どちらのプロパティも指定されなかった場合、RGM は事前定義済みのプロジェクト名 default を使用します。プロジェクトデータベース内に存在するプロジェクト名を指定する必要があります (SRM のマニュアルページを参照)。また、root ユーザーは、このプロジェクトのメンバーとして構成されている必要があります。このプロパティは、Solaris 9 以降のリリースでサポートされます。Solaris プロジェクト名の詳細は、『Solaris のシステム管理 (資源管理とネットワークサービス)』を参照してください。</p> <p>カテゴリ: 任意</p> <p>デフォルト: 上記を参照</p> <p>調整: 任意の時点</p>
Resource_state: 各クラスターノード (列挙)	<p>RGM が判断した各クラスターノード上のリソースの状態。可能な状態は次のとおりです。ONLINE、OFFLINE、STOP_FAILED、START_FAILED、MONITOR_FAILED、ONLINE_NOT_MONITORED。</p> <p>ユーザーは、このプロパティの設定を許可されていません。</p> <p>カテゴリ: 照会のみ</p> <p>デフォルト: なし</p> <p>調整: 不可</p>

表 A-2 リソースプロパティ (続き)

プロパティ名	説明
<p>Retry_count (整数)</p>	<p>リソースの起動に失敗した場合にモニターが再起動を試みる試行回数です。このプロパティは、RGM のみが作成でき、RTR ファイル内で宣言されている場合には、管理者はこのプロパティを利用できます。デフォルト値が RTR ファイルで指定されている場合は、このプロパティは任意です。</p> <p>リソースタイプファイル内に Tunable 属性が指定されていない場合、このプロパティの Tunable 値は When_disabled になります。</p> <p>RTR ファイルのプロパティ宣言内に Default 属性が指定されていない場合、このプロパティは必須です。</p> <p>カテゴリ: 条件付</p> <p>デフォルト: 上記を参照</p> <p>調整: 無効時</p>
<p>Retry_interval (整数)</p>	<p>失敗したリソースを再起動するまでの秒数。リソースモニターは、このプロパティと Retry_count を組み合わせて使用します。このプロパティは、RGM のみが作成でき、RTR ファイルで宣言されている場合は、管理者は利用できます。デフォルト値が RTR ファイルで指定されている場合は、このプロパティは任意です。</p> <p>リソースタイプファイル内に Tunable 属性が指定されていない場合、このプロパティの Tunable 値は When_disabled になります。</p> <p>RTR ファイルのプロパティ宣言内に Default 属性が指定されていない場合、このプロパティは必須です。</p> <p>カテゴリ: 条件付</p> <p>デフォルト: 上記を参照</p> <p>調整: 無効時</p>

表 A-2 リソースプロパティ (続き)

プロパティ名	説明
<p>Scalable (ブール値)</p>	<p>リソースがスケーラブルかどうかを示します。このプロパティが RTR ファイルで宣言されている場合は、そのタイプのリソースに対し、RGM は、次のスケーラブルサービスプロパティを自動的に作成します。</p> <p>Network_resources_used、Port_list、Load_balancing_policy、Load_balancing_weights。これらのプロパティは、RTR ファイルで明示的に宣言されないかぎり、デフォルト値を持ちます。RTR ファイルで宣言されている場合、Scalable のデフォルトは True です。</p> <p>このプロパティが RTR ファイルで宣言されている場合、Tunable 属性は、At_creation (作成時) に設定する必要があります。設定しなければ、リソースの生成に失敗します。</p> <p>RTR ファイルにこのプロパティが宣言されていない場合、リソースはスケーラブルにはなりません。したがって、クラスタ管理者はこのプロパティを調整することができず、RGM はスケーラブルサービスプロパティを設定しません。ただし、必要に応じて、明示的に Network_resources_used および Port_list プロパティを RTR ファイルで宣言できます。これらのプロパティは、スケーラブルサービスだけでなく、非スケーラブルサービスでも有用です。</p> <p>カテゴリ: 任意</p> <p>デフォルト: 上記を参照</p> <p>調整: 作成時</p>
<p>Status: 各クラスターノード (列挙)</p>	<p>リソースモニターによって設定されます。指定可能な値は、OK、degraded、faulted、unknown、および offline です。RGM は、リソースがオンラインになると、値を unknown に設定し、オフラインになると offline に設定します。</p> <p>カテゴリ: 照会のみ</p> <p>デフォルト: なし</p> <p>調整: 不可</p>

表 A-2 リソースプロパティ (続き)

プロパティ名	説明
Status_msg: 各クラスターノード (文字列)	<p>リソースモニターによって、Status プロパティと同時に設定されます。このプロパティは、各ノードのリソースごとに調整できます。RGM は、リソースがオフラインになると、このプロパティに空の文字列を設定します。</p> <p>カテゴリ: 照会のみ</p> <p>デフォルト: なし</p> <p>調整: 不可</p>
Thorough_probe_interval (整数)	<p>高オーバーヘッドのリソース障害検証の呼び出し間隔 (秒)。このプロパティは、RGM のみが作成でき、RTR ファイル内で宣言されている場合には、管理者はこのプロパティを利用できます。デフォルト値が RTR ファイルで指定されている場合は、このプロパティは任意です。</p> <p>リソースタイプファイル内に Tunable 属性が指定されていない場合、このプロパティの Tunable 値は When_disabled になります。</p> <p>RTR ファイルのプロパティ宣言内に Default 属性が指定されていない場合、このプロパティは必須です。</p> <p>カテゴリ: 条件付</p> <p>デフォルト: なし</p> <p>調整: 無効時</p>
Type (文字列)	<p>インスタンスのリソースタイプ。</p> <p>カテゴリ: 必須</p> <p>デフォルト: なし</p> <p>調整: 不可</p>

表 A-2 リソースプロパティ (続き)

プロパティ名	説明
Type_version (文字列)	<p>現在このリソースに関連付けられているリソース型のバージョンを指定します。このプロパティは RTR ファイル内に宣言できません。したがって、RGM によって自動的に作成されます。このプロパティの値は、リソースタイプの RT_version プロパティと等しくなります。リソースの作成時、Type_version プロパティはリソースタイプ名の接尾辞として表示されるだけで、明示的には指定されません。リソースを編集すると、Type_version の値が変更されます。</p> <p>RT_version は、データサービスのバージョンをアップグレードするときに使用します。</p> <p>次の項目から派生:</p> <ul style="list-style-type: none"> ■ 現在のリソースタイプのバージョン ■ *RTR ファイル内の #supgrade_from ディレクティブ
Udp_affinity (ブール値)	<p>TRUE の場合、あるクライアントからの UDP パケットは、TCP 接続と同様の方法で、同じクラスターサーバーノードに転送されます。クライアントの TCP 接続がサーバーノードに「スティック」されている場合、その UDP パケットもそのノードに送信されます。FALSE の場合、この動作は保持されません。あるクライアントからの UDP パケットは、そのクライアントからの TCP 接続とは異なるクラスターサーバーノードに転送されます。</p> <p>このプロパティは Load_balancing_policy が LB_STICKY または LB_STICKYWILD である場合にしか適用されません。</p> <p>カテゴリ: 任意</p> <p>デフォルト: False</p> <p>調整: 無効時</p>

表 A-2 リソースプロパティ (続き)

プロパティ名	説明
<p>Weak_affinity (ブール値)</p>	<p>スティックスケーラブルサービスに IP アフィニティを設定します。ただし、次の場合、このプロパティは中断されます。(1) 障害モニターの再起動、リソースのフェイルオーバー、スイッチオーバーなどが原因でサーバーリスナーが再起動する場合か、あるいはフェイルオーバーのあとでノードがクラスタに再結合する場合。(2) 管理アクションのためにスケーラブルリソースの load_balancing_weights が変更されたとき。メモリー消費とプロセッササイクルの点で、デフォルトの形式よりもオーバーヘッドは低くなります。アフィニティタイムアウト値は使用しません。Weak_affinity は、上記の割り込みが発生しないかぎり持続します。</p> <p>このプロパティは Load_balancing_policy が LB_STICKY または LB_STICKYWILD である場合にしか適用されません。</p> <p>カテゴリ: 任意</p> <p>デフォルト: False</p> <p>調整: 無効時</p>

リソースグループプロパティ

表 A-3 に、Sun Cluster で定義されているリソースグループプロパティの内容を示します。

表 A-3 リソースグループプロパティ

プロパティ名	説明
Auto_start_on_new_cluster (ブール値)	<p>このプロパティは、新しいクラスタの形成時にリソースグループマネージャが自動的にリソースグループを起動するかどうかを制御します。</p> <p>デフォルトは TRUE です。TRUE に設定した場合、クラスタのすべてのノードが同時に再起動すると、リソースグループマネージャはリソースグループを自動的に起動して <code>Desired primaries</code> を取得しようとしません。FALSE に設定されている場合、クラスタのすべてのノードが同時に再起動したとき、Resource Group Manager はリソースグループを自動的に起動しません。</p> <p>カテゴリ: 任意</p> <p>デフォルト: True</p> <p>調整: 任意の時点</p>
Desired primaries (整数)	<p>グループが同時に実行できるノード数として望ましい値。</p> <p>デフォルトは 1 です。RG_mode プロパティが Failover の場合、このプロパティの値を 1 より大きく設定することはできません。RG_mode プロパティが Scalable の場合は、1 より大きな値を設定できます。</p> <p>カテゴリ: 任意</p> <p>デフォルト: 1 (上記を参照)</p> <p>調整: 任意の時点</p>
Failback (ブール値)	<p>クラスタのメンバーシップが変更されたとき、グループがオンラインになっているノードセットを再計算するかどうかを示すブール値です。再計算によって、RGM はグループを優先度の低いノードでオフラインにし、優先度の高いノードでオンラインにします。</p> <p>カテゴリ: 任意</p> <p>デフォルト: False</p> <p>調整: 任意の時点</p>

表 A-3 リソースグループプロパティ (続き)

プロパティ名	説明
Global_resources_used (文字配列)	<p>クラスタファイルシステムがこのリソースグループ内のリソースによって使用されるかどうかを指定します。管理者は、すべてのグローバルリソース (アスタリスク記号 *) またはグローバルリソースなし (空の文字列 "") に指定できます。</p> <p>カテゴリ: 任意</p> <p>デフォルト: すべてのグローバルリソース</p> <p>調整: 任意の時点</p>
Implicit_network_dependencies(ブール値)	<p>True の場合に、グループ内のネットワークアドレスリソースに対し、非ネットワークアドレスリソースの暗黙の強い依存性を RGM が強制することを指定するブール値。つまり、RGM はグループ内のすべてのネットワークアドレスリソースを起動してからほかのすべてのリソースを起動し、ほかのリソースのあとにネットワークアドレスリソースを停止します。ネットワークアドレスリソースには、論理ホスト名と共有アドレスリソース型があります。</p> <p>スケーラブルなリソースグループ内では、このプロパティの影響はありません。これは、スケーラブルなリソースグループにはネットワークアドレスリソースが含まれないからです。</p> <p>カテゴリ: 任意</p> <p>デフォルト: True</p> <p>調整: 無効時</p>
Maximum primaries (整数)	<p>グループを同時にオンラインにできるノードの最大数です。</p> <p>デフォルトは 1 です。RG_mode プロパティが Failover の場合、このプロパティの値を 1 より大きく設定することはできません。RG_mode プロパティが Scalable の場合は、1 より大きな値を設定できます。</p> <p>カテゴリ: 任意</p> <p>デフォルト: 1 (上記を参照)</p> <p>調整: 任意の時点</p>

表 A-3 リソースグループプロパティ (続き)

プロパティ名	説明
Nodelist (文字配列)	<p>グループを優先度順にオンラインにできるクラスタノードをコンマで区切ったリストです。これらのノードは、リソースグループの稼働系もしくは待機系です。</p> <p>カテゴリ: 任意</p> <p>デフォルト: すべてのクラスタノードの順不同のリスト</p> <p>調整: 任意の時点</p>
Pathprefix (文字列)	<p>クラスタファイルシステム内のディレクトリです。グループ内のリソースは、このディレクトリ内に必要な管理ファイルを書き込むことができます。一部のリソースの必須プロパティです。各リソースグループの Pathprefix は、一意にする必要があります。</p> <p>カテゴリ: 任意</p> <p>デフォルト: 空の文字列</p> <p>調整: 任意の時点</p>

表 A-3 リソースグループプロパティ (続き)

プロパティ名	説明
Pingpong_interval (整数)	<p>再構成が生じた場合、あるいは <code>scha_control giveover</code> コマンドまたは関数の実行結果として、どのノードでリソースグループをオンラインにするかを判断するときに RGM が使用する負以外の整数値 (秒)。</p> <p>再構成において、リソースの <code>START</code> または <code>PRENET_START</code> メソッドがゼロ以外の値で終了、またはタイムアウトによって終了したことが原因で、<code>Pingpong_interval</code> で指定した秒数内に、リソースグループをオンラインにするのを 2 回以上失敗した場合、RGM はそのノードはリソースグループのホストとして不適切だと判断し、別のマスターを捜します。</p> <p>リソースの <code>scha_control(1HA)</code> コマンドまたは <code>scha_control(3HA)</code> 機能の呼び出しによって、<code>Pingpong_interval</code> で指定した秒数内に特定のノード上でリソースグループがオフラインになった場合、<code>scha_control</code> 呼び出しの結果、そのノードはリソースグループのホストとして不適切だと判断されます。</p> <p>カテゴリ: 任意</p> <p>デフォルト: 3,600(1 時間)</p> <p>調整:任意の時点</p>
Resource_list (文字配列)	<p>グループ内に含まれるリソースのリストです。管理者はこのプロパティを直接設定しません。このプロパティは、管理者がリソースグループにリソースを追加したり、リソースグループからリソースを削除すると、RGM により自動的に更新されます。</p> <p>カテゴリ: 照会のみ</p> <p>デフォルト: 空のリスト</p> <p>調整:不可</p>

表 A-3 リソースグループプロパティ (続き)

プロパティ名	説明
RG_dependencies (文字配列)	<p>このグループが依存するリソースグループをコンマで区切ったリストです。このリストにより、同じノード上のその他のグループをオンラインまたはオフラインにする優先度が指定されます。別のノードでグループをオンラインにする場合は、このリストは無効です。</p> <p>カテゴリ: 任意</p> <p>デフォルト: 空のリスト</p> <p>調整: 任意の時点</p>
RG_description (文字列)	<p>リソースグループの簡単な説明です。</p> <p>カテゴリ: 任意</p> <p>デフォルト: 空の文字列</p> <p>調整: 任意の時点</p>
RG_mode (列挙)	<p>リソースグループがフェイルオーバーグループかスケラブルグループかを指定します。このプロパティの値が Failover の場合、RGM はグループの Maximum primaries プロパティを 1 に設定し、リソースグループが単一のノードでマスターされるように制限します。</p> <p>このプロパティの値が Scalable の場合、RGM は Maximum primaries プロパティが 1 より大きい値を持つことを許可し、複数のノードで同時にそのグループをマスターできるようにします。</p> <p>注: Failover プロパティの値が True のリソースを、RG_mode の値が Scalable のリソースグループに追加することはできません。</p> <p>カテゴリ: 任意</p> <p>デフォルト: Maximum primaries の値が 1 の場合は Failover</p> <p>Maximum primaries が 1 を超える場合は Scalable</p> <p>調整: 不可</p>

表 A-3 リソースグループプロパティ (続き)

プロパティ名	説明
RG_name (文字列)	<p>リソースグループの名前。これは必須プロパティです。この値は、クラスタ内で一意でなければなりません。</p> <p>カテゴリ: 必須</p> <p>デフォルト: なし</p> <p>調整: 不可</p>
RG_project_name	<p>リソースグループに関連付けられた Solaris プロジェクト名。このプロパティは、CPU の共有、クラスタデータサービスのリソースプールといった Solaris のリソース管理機能に適用できます。RGM は、リソースグループをオンラインにすると、</p> <p>Resource_project_name プロパティセットを持たないリソースに対して、このプロジェクトで関連付けられたプロセスを起動します。プロジェクトデータベース内に存在するプロジェクト名を指定する必要があります (SRM のマニュアルページを参照)。また、root ユーザーは、このプロジェクトのメンバーとして構成されている必要があります。このプロパティは、Solaris 9 以降のリリースでサポートされます。Solaris プロジェクト名の詳細は、『Solaris のシステム管理 (資源管理とネットワークサービス)』を参照してください。</p> <p>カテゴリ: 任意</p> <p>デフォルト: Default (上記を参照)</p> <p>調整: 任意の時点</p>
RG_state: 各クラスターノード (列挙)	<p>RGM によって Online、Offline、Pending_online、Pending_offline、Error_stop_failed に設定され、各クラスターノード上のグループの状態を示します。RGM の制御下でないときは、グループは管理されていない状態で存在することもできます。</p> <p>ユーザーは、このプロパティの設定を許可されていません。</p> <p>カテゴリ: 照会のみ</p> <p>デフォルト: Offline</p> <p>調整: 不可</p>

リソースプロパティの属性

表 A-4 に、システム定義プロパティの変更または拡張プロパティの作成に使用できるリソースプロパティの属性を示します。



注意 - boolean、enum、int タイプのデフォルト値に、NULL または空の文字列 ("") は指定できません。

表 A-4 リソースプロパティの属性

プロパティ	説明
Property	リソースプロパティの名前。
Extension	リソースタイプの実装によって定義された拡張プロパティが RTR ファイルのエントリで宣言されていることを示します。拡張プロパティが使用されていない場合、そのエントリはシステム定義プロパティです。
Description	プロパティを簡潔に記述した注記 (文字列)。RTR ファイル内でシステム定義プロパティに対する Description 属性を設定することはできません。
プロパティのタイプ	指定可能なタイプは、string、boolean、int、enum、stringarray です。RTR ファイル内で、システム定義プロパティの型の属性を設定することはできません。タイプは、RTR ファイルのエントリに登録できる、指定可能なプロパティ値とタイプ固有の属性を決定します。enum タイプは、文字列値のセットです。
Default	プロパティのデフォルト値を示します。
Tunable	クラスタ管理者がリソースのプロパティ値をいつ設定できるかを示します。管理者にプロパティの設定を許可しない場合は、None または False に設定します。管理者にプロパティの調整を許可する属性値は、次のとおりです。True または Anytime (任意の時点)、At_creation (リソースの作成時のみ)、When_disabled (リソースがオフラインのとき)。デフォルトは、True (Anytime) です。
Enumlist	enum タイプの場合、プロパティに設定できる文字列値のセット。
Min	int タイプの場合、プロパティに設定できる最小値。
Max	int タイプの場合、プロパティに設定できる最大値。
Minlength	string および stringarray タイプの場合、設定できる文字列の最小長。
Maxlength	string および stringarray タイプの場合、設定できる文字列の最大。

表 A-4 リソースプロパティの属性 (続き)

プロパティ	説明
Array_minsize	stringarray タイプの場合、設定できる配列要素の最小数。
Array_maxsize	stringarray タイプの場合、設定できる配列要素の最大数。

付録 B

有効な RGM 名と値

この付録では、Resource Group Manager (RGM) の名前と値に指定できる文字の条件について説明します。

有効な RGM 名

RGM 名は、次の 5 つのカテゴリに分類されます。

- リソースグループ名
- リソースタイプ名
- リソース名
- プロパティ名
- 列挙リテラル名

リソースタイプ名を除き、他の名前はすべて次の規則に従う必要があります。

- 必ず ASCII にする。
- 先頭は必ず文字にする。
- アルファベットの大文字と小文字、数字、ダッシュ (-)、下線 (_) を含むことができる。
- 255 文字以下にする。

リソースタイプ名は、簡単な名前 (RTR ファイルの `Resource_type` プロパティで指定) または完全な名前 (RTR ファイルの `Vendor_id` と `Resource_type` で指定) のどちらでもかまいません。 `Vendor_id` と `Resource_type` の両プロパティを指定した場合は、RGM は、これら 2 つのプロパティ間にピリオドを挿入して完全な名前を形成します。たとえば、`Vendor_id=SUNW` と `Resource_type=sample` の場合、完全な名前は `SUNW.sample` になります。RGM 名にピリオドを使用できるのはこのインスタンスの場合だけです。

RGM の値

RGM の値は、プロパティ値と説明値の 2 つのカテゴリに分類されます。これら 2 つのカテゴリは同じ規則を共有します。

- 値は ASCII であること。
- 値の最大長は 4M - 1 バイト (つまり、4,194,303 バイト) であること。
- 値に次の文字を含むことはできない。
 - null
 - 改行
 - カンマ
 - セミコロン

付録 C

データサービス構成のワークシートと記入例

この付録では、クラスタ構成のリソース関連構成要素を計画する場合に使用するワークシートを提供します。参考のために、ワークシートの記入例も掲載しています。クラスタ構成内のその他のコンポーネントのワークシートは、『Sun Cluster ソフトウェアのインストール (Solaris OS 版)』の「Sun Cluster のインストールと構成のためのワークシート」を参照してください。

構成のワークシート

リソースに関連するコンポーネントがクラスタ構成に多数ある場合は、ワークシートを適宜コピーしてください。これらのワークシートを完成させるには、『Sun Cluster ソフトウェアのインストール (Solaris OS 版)』および第 1 章の計画ガイドラインに従ってください。記入済みのワークシートを参照しながら、クラスタをインストールおよび構成します。

注 - ワークシートの記入例で使用されるデータはガイドとしてのみ提供されます。したがって、これらの例は、実際のクラスタの完全な構成を表しているわけではありません。

- 124 ページの「リソースタイプのワークシート」
- 126 ページの「ネットワークリソースのワークシート」
- 128 ページの「アプリケーションリソース - フェイルオーバーワークシート」
- 130 ページの「アプリケーションリソース - スケーラブルのワークシート」
- 132 ページの「リソースグループ - フェイルオーバーのワークシート」
- 134 ページの「リソースグループ - スケーラブルのワークシート」

リソースタイプのワークシート

論理ホストまたは共有アドレス以外のリソースタイプにはこのワークシートを使用してください。

表 C-1 リソースタイプのワークシート

リソースタイプ名	リソースタイプが動作するノード

例: リソースタイプのワークシート

表 C-2 例: リソースタイプのワークシート

リソースタイプ名	リソースタイプが動作するノード
<code>SUNW.nshttp</code>	<code>phys-schost-1, phys-schost-2</code>
<code>SUNW.oracle_listener</code>	<code>phys-schost-1, phys-schost-2</code>
<code>SUNW.oracle_server</code>	<code>phys-schost-1, phys-schost-2</code>

ネットワークリソースのワークシート

表 C-3 ネットワークリソースのワークシート

コンポーネント	名前	
リソース名		
リソースグループ名		
リソースタイプ (1 つに丸を付けてください)	論理ホスト名 共有アドレス	
リソースタイプ名		
依存性		
使用されているホスト名		
拡張プロパティ	名称	値

例: ネットワークリソース — 共有アドレスのワークシート

表 C-4 例: ネットワークリソース — 共有アドレスのワークシート

コンポーネント	名前	
リソース名	sh-galileo	
リソースグループ名	rg-shared	
リソースタイプ (1 つに丸を付けてください)	Shared address	
リソースタイプ名	SUNW.SharedAddress	
依存性	none	
使用されているホスト名	sh-galileo	
拡張プロパティ	名称	値
	netiflist	ipmp0@1, ipmp0@2

例: ネットワークリソース — 論理ホスト名のワークシート

表 C-5 例: ネットワークリソース — 論理ホスト名のワークシート

コンポーネント	名前	
リソース名	relo-galileo	
リソースグループ名	rg-oracle	
リソースタイプ (1 つに丸を付けてください)	Logical hostname	
リソースタイプ名	SUNW.LogicalHostname	
依存性	none	
使用されているホスト名	relo-galileo	
拡張プロパティ	名称	値
	netiflist	ipmp0@1, ipmp0@2

アプリケーションリソース — フェイルオーバー ワークシート

表 C-6 アプリケーションリソース — フェイルオーバーワークシート

コンポーネント	名前	
リソース名		
リソースグループ名		
リソースタイプ名		
依存性		
拡張プロパティ	名前	値

例: アプリケーションリソース — フェイルオーバーワークシート

表 C-7 例: アプリケーションリソース — フェイルオーバーワークシート

コンポーネント	名前	
リソース名	oracle-listener	
リソースグループ名	rg-oracle	
リソースタイプ名	SUNW.oracle_listener	
依存性	hasp_resource	
拡張プロパティ	名前	値
	ORACLE_HOME	/global/oracle/orahome/
	LISTENER_NAME	lsnr1

アプリケーションリソース — スケーラブルのワークシート

表 C-8 アプリケーションリソース — スケーラブルのワークシート

コンポーネント	名前	
リソース名		
論理ホストのリソースグループ名		
共有アドレスのリソースグループ名		
論理ホストのリソースタイプ名		
共有アドレスのリソースタイプ名		
依存性		
拡張プロパティ	名前	値

例: アプリケーションリソース — スケーラブルのワークシート

表 C-9 例: アプリケーションリソース — スケーラブルのワークシート

コンポーネント	名前	
リソース名	sh-galileo	
論理ホストのリソースグループ名		
共有アドレスのリソースグループ名	rg-shared	
論理ホストのリソースタイプ名		
共有アドレスのリソースタイプ名		
依存性		
拡張プロパティ	名前	値

リソースグループ — フェイルオーバーのワークシート

表 C-10 リソースグループ — フェイルオーバーのワークシート

コンポーネント	注	名前
リソースグループ名	この名前は、クラスタ内で一意のものでなければなりません。	
機能	このリソースグループの機能について記述してください。	
フェイルバック機能があるか (1 つに丸を付けてください)	稼動系が停止して復旧したあと、このリソースグループを稼動系に戻すかどうかを選択してください。	戻す 戻さない
ノードリスト	このリソースグループのホストになりえるクラスタノードを指定してください。このリストでは、最初のノードとして稼動系を指定し、続いて待機系を指定する必要があります。待機系の順序は、稼動系になる優先順位を示します。	
依存しているディスクデバイスグループ	このリソースグループが依存しているディスクデバイスグループを指定してください。	
構成ディレクトリ	管理作業のためにこのリソースグループ内のリソースがファイルを作成する必要がある場合、それらのリソースが使用するサブディレクトリを含めてください。	

例: リソースグループ — フェイルオーバーのワークシート

表 C-11 例: リソースグループ — フェイルオーバーのワークシート

コンポーネント	注	名前
リソースグループ名	この名前は、クラスタ内で一意のものでなければなりません。	rg-oracle
機能	このリソースグループの機能について記述してください。	Oracle リソースを含む
フェイルバック機能があるか (1 つに丸を付けてください)	稼動系が停止して復旧したあと、このリソースグループを稼動系に戻すかどうかを選択してください。	戻さない
ノードリスト	このリソースグループのホストになりえるクラスタノードを指定してください。このリストでは、最初のノードとして稼動系を指定し、続いて待機系を指定する必要があります。待機系の順序は、稼動系になる優先順位を示します。	1) phys-schost-1 2) phys-schost-2
依存しているディスクデバイスグループ	このリソースグループが依存しているディスクデバイスグループを指定してください。	schost1-dg
構成ディレクトリ	管理作業のためにこのリソースグループ内のリソースがファイルを作成する必要がある場合、それらのリソースが使用するサブディレクトリを含めてください。	

リソースグループ — スケーラブルのワークシート

表 C-12 リソースグループ — スケーラブルのワークシート

コンポーネント	注	名前
リソースグループ名	この名前は、クラスタ内で一意のものでなければなりません。	
機能		
稼動系の最大数		
稼動系の適切な数		
フェイルバック機能があるか (1 つに丸を付けてください)	稼動系が停止したあと、このリソースグループを稼動系に戻すかどうかを選択してください。	戻す 戻さない
ノードリスト	このリソースグループのホストになりえるクラスタノードを指定してください。このリストでは、最初のノードとして稼動系を指定し、続いて待機系を指定する必要があります。待機系の順序は、稼動系になる優先順位を示します。	
依存性	このリソースが依存するリソースグループをすべて挙げてください。	

例: リソースグループ — スケーラブルのワークシート

表 C-13 例: リソースグループ — スケーラブルのワークシート

コンポーネント	注	名前
リソースグループ名	この名前は、クラスタ内で一意のものでなければなりません。	rg-http
機能		Web サーバーリソースを含む
稼働系の最大数		2
稼働系の適切な数		2
フェイルバック機能があるか (1 つに丸を付けてください)	稼働系が停止したあと、このリソースグループを稼働系に戻すかどうかを選択してください。	No
ノードリスト	このリソースグループのホストになりえるクラスタノードを指定してください。このリストでは、最初のノードとして稼働系を指定し、続いて待機系を指定する必要があります。待機系の順序は、稼働系になる優先順位を示します。	1) phys-schost-1 2) phys-schost-2
依存性	このリソースが依存するリソースグループをすべて挙げてください。	rg-shared

索引

A

Affinity_timeout, リソースプロパティ, 99
API_version, リソースタイププロパティ, 92
Array_maxsize, リソースプロパティの属性, 118
Array_minsize, リソースプロパティの属性, 118
Auto_start_on_new_cluster, リソースグループプロパティ, 112
auxnodelist, ノードリストプロパティ, 19

B

BOOT, リソースタイププロパティ, 92

C

Cheap_probe_interval, リソースプロパティ, 99
Copy
リソースグループプロパティ, 112

D

Default, リソースプロパティの属性, 118
Description
リソースプロパティの属性, 118
Desired primaries, リソースグループプロパティ, 112

E

Enumlist, リソースプロパティの属性, 118
Extension, リソースプロパティの属性, 118

F

Failback, リソースグループプロパティ, 112
Failover, リソースタイププロパティ, 92
Failover_mode, リソースプロパティ, 99
FINI, リソースタイププロパティ, 92

G

Global_resources_uses, リソースグループプロパティ, 112

H

HASStorage
概要, 16
データサービスが必要とするかどうかの確認, 17
または HASStoragePlus, 18
HASStoragePlus
概要, 16
データサービスが必要とするかどうかの確認, 17
または HASStorage, 18
HA ローカルファイルシステム, HASStoragePlus
の設定, 81

I

`Implicit_network_dependencies`, リソースグループプロパティ, 112
`INIT`, リソースタイププロパティ, 92
`Init_nodes`, リソースタイププロパティ, 92
`installed_nodes`, ノードリストプロパティ, 19
`Installed_nodes`, リソースタイププロパティ, 92

L

`Load_balancing_policy`, リソースプロパティ, 99
`Load_balancing_weights`, リソースプロパティ, 99

M

`Max`, リソースプロパティの属性, 118
`Maximum primaries`, リソースグループプロパティ, 112
`Maxlength`, リソースプロパティの属性, 118
`method_timeout`, リソースプロパティ, 99
`Min`, リソースプロパティの属性, 118
`Minlength`, リソースプロパティの属性, 118
`Monitor_check`, リソースタイププロパティ, 92
`Monitor_start`, リソースタイププロパティ, 92
`Monitor_stop`, リソースタイププロパティ, 92
`Monitored_switch`, リソースプロパティ, 99

N

`Network_resources_used`, リソースプロパティ, 99
`node`
削除
リソースグループから, 70
追加
リソースグループに, 68
`nodelist`, ノードリストプロパティ, 19
`Nodelist`, リソースグループプロパティ, 112

`nsswitch.conf`, ファイルの内容の確認, 15
`Num_resources_restart`, リソースタイププロパティ, 92

O

`On_off_switch`, リソースプロパティ, 99

P

`Pathprefix`, リソースグループプロパティ, 112
`Pingpong_interval`, リソースグループプロパティ, 112
`Pkg_list`, リソースタイププロパティ, 92
`PMF`, 「プロセスモニター機能 (PMF)」を参照
`Port_list`, リソースプロパティ, 99
`Postnet_stop`, リソースタイププロパティ, 92
`Prenet_stop`, リソースタイププロパティ, 92
`Property`, リソースプロパティの属性, 118
`prtconf -v command`, 10
`prtdiag -v コマンド`, 10
`psrinfo -v コマンド`, 10

R

`R_description`, リソースプロパティ, 99
`Resource_dependencies`, リソースプロパティ, 99
`Resource_dependencies_weak`, リソースプロパティ, 99
`Resource_list`, リソースグループプロパティ, 112
`Resource_name`, リソースプロパティ, 99
`Resource_project_name`, リソースプロパティ, 99
`Resource_state`, リソースプロパティ, 99
`Resource_type`, リソースタイププロパティ, 92
`Retry_count`, リソースプロパティ, 99
`Retry_interval`, リソースプロパティ, 99
`RG_dependencies`, リソースグループプロパティ, 112

RG_description, リソースグループプロパティ, 112
RG_mode, リソースグループプロパティ, 112
RG_name, リソースグループプロパティ, 112
RG_project_name, リソースグループプロパティ, 112
RG_state, リソースグループプロパティ, 112
RGM, 「リソースグループマネージャ」を参照
RGOffload
 拡張プロパティ
 continue_to_offload, 87
 max_offload_retry, 87
 rg_to_offload, 87
 フォルトモニター, 88
RT_basedir, リソースタイププロパティ, 92
RT_description, リソースタイププロパティ, 92
RT_version, リソースタイププロパティ, 92

S

Scalable, リソースプロパティ, 99
scinstall -pv コマンド, 11
scrgadm コマンド, 22
scsetup ユーティリティー, 22
showrev -p コマンド, 10
Single_instance, リソースタイププロパティ, 92
START, リソースタイププロパティ, 92
Status, リソースプロパティ, 99
Status_message, リソースプロパティ, 99
STOP, リソースタイププロパティ, 92
Sun Management Center GUI, 22
SunPlex Manager GUI, 22

T

Thorough_probe_interval, リソースプロパティ, 99
Tunable, リソースプロパティの属性, 118
Type, リソースプロパティ, 99
Type_version, リソースプロパティ, 99

U

Udp_affinity, リソースプロパティ, 99
UPDATE, リソースタイププロパティ, 92

V

VALIDATE, リソースタイププロパティ, 92
Vendor_ID, リソースタイププロパティ, 92

W

Weak_affinity, リソースプロパティ, 99

あ

値, リソースグループマネージャ, 122
新しいリソースタイプバージョンへの移行, 33
アップグレード, リソースタイプ, 32
アプリケーションバイナリ, 格納先の決定, 14

い

インストール, 概要, 19
インストール, 作業の一覧, 20

お

オンラインにする, リソースグループ, 50

か

拡張プロパティ
 RGOffload
 continue_to_offload, 87
 max_offload_retry, 87
 rg_to_offload, 87
 リソースプロパティ, 99
確認, nsswitch.conf ファイルの内容, 15

き

起動の同期, リソースグループとディスクデバイスグループ間での, 75

け

計画

クラスタファイルシステム

構成, 15

データサービス, 13

現在の稼働系の切り替え, リソースグループ, 57

こ

構成

概要, 19

クラスタファイルシステムの計画, 15

構成, 作業の一覧, 20

構成と管理, Sun Cluster データサービス, 30

構成のガイドライン, 13

考慮事項, 18

コマンド, ノード情報, 10

さ

再登録, 登録済みのリソースタイプ, 66

作業マップ, データサービスリソース, 28

削除

共有アドレスを含むリソースグループから
ノードを

フェイルオーバー, 74

リソース, 56

リソースグループ, 54

リソースグループからノードを, 70

スケーラブル, 71

フェイルオーバー, 72

リソースタイプ, 53

作成

リソースグループ

スケーラブル, 40

フェイルオーバー, 39

し

消去, STOP_FAILEDリソースのエラーフラグ, 65

せ

設定

HAStorage

新しいリソース, 76

既存のリソース, 78

HAStoragePlus, 81

RGOffload, 84

つ

追加

リソースグループにノードを, 68

failover, 69

scalable, 68

リソースグループにリソースを, 42

logical hostname, 42

共有アドレス, 44

フェイルオーバーアプリケーション, 46

リソースをリソースグループに

スケーラブルアプリケーション, 48

ツール

scrgadm コマンド, 22

scsetup ユーティリティ, 22

Sun Management Center GUI, 22

SunPlex Manager GUI, 22

て

ディスクデバイスグループ, 15

リソースグループとの関係, 15

リソースグループとの起動の同期, 75

データサービス

計画, 13

考慮事項, 18

特殊な要件, 14

データサービスリソース, 作業マップ, 28

と

登録, リソースタイプ, 31
特殊な要件, 確認, 14

の

ノード

削除

リソースグループから, 71, 72, 74

追加

リソースグループに, 68, 69

ノードリストプロパティ

auxodelist, 19

installed_nodes, 19

nodelist, 19

ノードリソースの開放, RGOffload, 84

ひ

表示, リソースタイプ, リソースグループ, リ
ソース構成情報, 60

ふ

フォルトモニター

RGOffload, 88

概要, 24

プロセスモニター機能 (PMF), 24

プライオリティが低いリソースグループをオフ
ロードする, RGOffload, 84

プロセスモニター機能 (PMF), 24

プロパティのタイプ, リソースプロパティの属
性, 118

へ

変更

リソースグループプロパティ, 62

リソースタイププロパティ, 61

リソースプロパティ, 63

む

無効化

リソース

リソースグループを UNMANAGED に移
行する, 58

リソースグループモニター, 51

ゆ

有効化, リソースグループモニター, 51

有効な名前, リソースグループマネージャ, 121

よ

要件, データサービス, 14

り

リソース

構成情報の表示, 60

削除, 56

消去

STOP_FAILED エラーフラグ, 65

スケラブルアプリケーションリソースをリ
ソースグループに追加, 48

プロパティの変更, 63

リソースグループに共有アドレスリソースを
追加, 44

リソースグループにフェイルオーバーアプリ
ケーションリソースを追加, 46

リソースグループにリソースを追加, 42

リソースグループに論理ホスト名リソースを
追加, 42

リソースグループを無効にして

UNMANAGED 状態に移行する, 58

リソースタイプの再登録, 66

リソースタイプの削除, 53

リソースタイププロパティの変更, 61

リソースグループ, 15

オンラインにする, 50

現在の稼働系の切り替え, 57

構成情報の表示, 60

削除, 54

ノード, 70, 71, 72, 74

- リソースグループ (続き)
 - 作成
 - スケーラブル, 40
 - フェイルオーバー, 39
 - 追加
 - ノード, 68, 69
 - ディスクデバイスグループとの関係, 15
 - ディスクデバイスグループとの起動の同期, 75
 - プロパティの変更, 62
 - 無効化
 - リソースフォルトモニター, 51
 - モニターの無効化, 51
 - モニターの有効化, 51
 - 有効化
 - リソースフォルトモニター, 52
 - リソースの追加, 42
 - 共有アドレス, 44
 - スケーラブルアプリケーション, 48
 - フェイルオーバーアプリケーション, 46
 - 論理ホスト名, 42
 - リソースグループプロパティ
 - Auto_start_on_new_cluster, 112
 - Desired primaries, 112
 - Failback, 112
 - Global_resources_uses, 112
 - Implicit_network_dependencies, 112
 - Maximum primaries, 112
 - Nodelist, 112
 - Pathprefix, 112
 - Pingpong_interval, 112
 - Resource_list, 112
 - RG_dependencies, 112
 - RG_description, 112
 - RG_mode, 112
 - RG_name, 112
 - RG_project_name, 112
 - RG_state, 112
 - リソースグループマネージャ
 - 値, 122
 - 有効な名前, 121
 - リソースタイプ
 - HASStorage
 - 新しいリソース, 76
 - 既存のリソース, 78
 - RGOffload, 84
 - 新しいリソースタイプバージョンへの移行, 33
- リソースタイプ (続き)
 - アップグレード, 32
 - 構成情報の表示, 60
 - 再登録, 66
 - 削除, 53
 - 登録, 31
 - プロパティの変更, 61
 - リソースタイププロパティ
 - API_version, 92
 - BOOT, 92
 - Failover, 92
 - FINI, 92
 - INIT, 92
 - Init_nodes, 92
 - Installed_nodes, 92
 - Monitor_check, 92
 - Monitor_start, 92
 - Monitor_stop, 92
 - Num_resources_restart, 92
 - Pkg_list, 92
 - Postnet_stop, 92
 - Prenet_stop, 92
 - Resource_type, 92
 - RT_basedir, 92
 - RT_description, 92
 - RT_version, 92
 - Single_instance, 92
 - START, 92
 - STOP, 92
 - UPDATE, 92
 - VALIDATE, 92
 - Vendor_ID, 92
 - リソースフォルトモニター
 - 無効化, 51
 - 有効化, 52
 - リソースプロパティ
 - Affinity_timeout, 99
 - Cheap_probe_interval, 99
 - Extension properties, 99
 - Failover_mode, 99
 - Load_balancing_policy, 99
 - Load_balancing_weights, 99
 - method_timeout, 99
 - Monitored_switch, 99
 - Network_resources_used, 99
 - On_off_switch, 99
 - Port_list, 99
 - R_description, 99

リソースプロパティ (続き)

- Resource_dependencies, 99
- Resource_dependencies_weak, 99
- Resource_name, 99
- Resource_project_name, 99
- Resource_state, 99
- Retry_count, 99
- Retry_interval, 99
- Scalable, 99
- Status, 99
- Status_message, 99
- Thorough_probe_interval, 99
- Type, 99
- Type_version, 99
- Udp_affinity, 99
- Weak_affinity, 99

リソースプロパティの属性

- Array_maxsize, 118
- Array_minsize, 118
- Default, 118
- Description, 118
- Enumlist, 118
- Extension, 118
- Max, 118
- Maxlength, 118
- Min, 118
- Minlength, 118
- Tunable, 118
- プロパティ, 118
- プロパティのタイプ, 118

