
LifeKeeper® for Linux

Generic ARK for JP1/AJS3 Agent 管理ガイド

第 1.4 版



目次

改定履歴	3
1. 概要	4
2. Generic ARK for JP1/AJS3 Agent リソース階層	4
3. 設定上の検討事項	5
3.1 ファイルシステムリソースの制限	5
3.2 IP バックアップインターフェース機能の制限	5
3.3 JP1/AJS3 Agent, JP1/Base の保護 : ベストプラクティス	5
3.4 論理ホスト名と仮想 IP リソース	5
3.5 複数インスタンスの実行(Active/Active 構成)	5
4. LifeKeeper と JP1/AJS3 Agent の設定	6
4.1 LifeKeeper のインストールとコミュニケーションパスの作成	7
4.2 ファイルシステムリソース, 仮想 IP リソースの作成	7
4.3 JP1/Base, JP1/AJS3 Agent のインストールと構築作業	7
4.4 Generic ARK for JP1/AJS3 Agent のインストールとセットアップ	7
4.4.1 インストール	7
4.4.2 パラメーターの設定	8
4.4.3 Generic ARK for JP1/Base のセットアップ	12
4.4.4 Generic ARK for JP1/AJS3 Agent のセットアップ	13
4.5 依存関係の作成	14
5. Generic ARK for JP1/AJS3 Agent リソースのその他の操作	15
5.1 Generic ARK for JP1/AJS3 Agent リソースの依存関係の削除	15
5.2 Generic ARK for JP1/AJS3 Agent リソース階層の削除	16
5.3 Generic ARK for JP1/AJS3 Agent リソース階層のテスト	16
6. ログメッセージの解説	17



改定履歴

版	改定日	改定内容
第 1 版	2010 年 8 月	新規作成
第 1.1 版	2016 年 7 月	「LifeKeeper のインストールとコミュニケーションパスの作成」改定
第 1.2 版	2018 年 11 月	ログメッセージの改定
第 1.3 版	2021 年 2 月	待機時間パラメータ導入
第 1.4 版	2022 年 3 月	コーポレートロゴの更新



1. 概要

Generic ARK for JP1/AJS3 Agent は、”JP1/Base” と”JP1/Automatic Job Management System 3”の”Agent”を、LifeKeeper for Linux の保護対象リソースとして登録し、保護する機能を提供します。各JP1 ソフトウェア、LifeKeeper コア、およびGeneric ARK for JP1/AJS3 Agent は、クラスターを構成する各サーバーにインストールされます。

JP1 ソフトウェアが使用する共有ディスクリソースと、JP1 の論理ホスト名に紐づく仮想IP リソースは、別途作成しておく必要があります。また、Generic ARK for JP1/AJS3 Agent リソース作成後に、共有ディスクリソースと仮想IP リソースに対する依存関係を手動で設定する必要があります。

保護対象のJP1 ソフトウェアに障害が発生した場合は、LifeKeeper はまずローカルサーバー上でリカバリーを試みます。そのリカバリーに失敗した場合は、バックアップサーバーに切り替えます。

2. Generic ARK for JP1/AJS3 Agent リソース階層

以下は典型的なGeneric ARK for JP1/AJS3 Agent リソース階層です。

[JP1_AJS3_Agent Resource]
+ [JP1_Base Resource]
+ [IP Resource]
+ [Filesystem Resource]

上記の例の従属関係は、以下の保護されるリソースに対応しています。なお、リソース名はいずれも任意につけることが出来ます。

リソースタグ名	機能
lkjp1_ajsagt	JP1/AJS3 Agent の制御を行います。
lkjp1_base	JP1/Base の制御を行います。
lkjp1_vip	JP1 の論理ホスト名に紐づいた仮想IP アドレスです。これはLifeKeeper のIP リカバリーキットにより作成されています。
lkjp1_shared	JP1 が使用する共有ファイルシステムです。これはLifeKeeper のFileSystem リカバリーキットにより作成されています。
deviceXXX	保護対象ディスクのパーティションの制御を行います。これはLifeKeeper のFileSystem リカバリーキットにより作成されています。
diskXXX	保護対象ディスクのディスクそのもの(LU)の制御を行います。これはLifeKeeper のFileSystem リカバリーキットにより作成されています。



フェイルオーバーが発生した場合、LifeKeeper はバックアップサーバーでは、ファイルシステム、IP アドレス、JP1/Base、JP1/AJS3 Agent の各リソースを、リソースの依存関係を含んだ形で起動します。

3. 設定上の検討事項

本セクションでは、Generic ARK for JP1/AJS3 Agent の設定と管理を開始する前に検討すべき事項について説明します。

3.1 ファイルシステムリソースの制限

Generic ARK for JP1/AJS3 Agent は、FileSystem リカバリーキットで作成することが出来る共有ファイルシステムリソースのみをサポートします。SteelEye Data Replication(SDR)、NAS RecoveryKit、DRBD RecoveryKit などによるファイルシステムリソースはサポートされません。

3.2 IP バックアップインターフェース機能の制限

Generic ARK for JP1/AJS3 Agent では、LifeKeeper Core のIP リカバリーキットが提供するIP バックアップインターフェース機能をサポートしません。ネットワークの冗長化を行う場合は、OS のチーミング・デバイスの機能を使用してください。

3.3 JP1/AJS3 Agent、JP1/Base の保護：ベストプラクティス

本キットにて、JP1/AJS3 Agent、JP1/Base を保護する場合のJP1 側の設定については、日立製作所のマニュアルを参照してください。

3.4 論理ホスト名と仮想 IP リソース

JP1 の論理ホスト名に紐づいた仮想IP アドレスを作成する必要があります。仮想IP アドレスはLifeKeeper のIP リカバリーキットにより作成されていなければなりません。また、論理ホスト名が仮想IP アドレスに解決されることを確実にするため、`/etc/hosts` に論理ホスト名のエントリを記載してください。

3.5 複数インスタンスの実行(Active/Active 構成)

Generic ARK for JP1/AJS3 Agent では、論理ホスト名に紐付く形で複数のインスタンスを制御することが出来るため、Active/Standby、Active/Active のいずれも構成可能です。Active/Active 構成とする場合は、単体ノード上で複数のインスタンスが起動できることを事前に確認してください。



Active/Standby 構成例

lkjp1A (Active)		lkjp1B (Standby)	
TAG	PRI	TAG	PRI
lkjp1_ajsagt	1	lkjp1_ajsagt	10
lkjp1_base	1	lkjp1_base	10
lkjp1_vip	1	lkjp1_vip	10
lkjp1_shared	1	lkjp1_shared	10
deviceXXX	1	deviceXXX	10
diskXXX	1	diskXXX	10

Active/Active 構成例

lkjp1A (Active)		lkjp1B (Active)		lkjp1C (Standby)	
TAG	PRI	TAG	PRI	TAG	PRI
lkjp1A_ajsagt	1			lkjp1A_ajsagt	10
lkjp1A_base	1			lkjp1A_base	10
lkjp1A_vip	1			lkjp1A_vip	10
lkjp1A_shared	1			lkjp1A_shared	10
deviceXXX	1			deviceXXX	10
diskXXX	1			diskXXX	10
		lkjp1B_ajsagt	1	lkjp1B_ajsagt	10
		lkjp1B_base	1	lkjp1B_base	10
		lkjp1B_vip	1	lkjp1B_vip	10
		lkjp1B_shared	1	lkjp1B_shared	10
		deviceYYY	1	deviceYYY	10
		diskYYY	1	diskYYY	10

4. LifeKeeper と JP1/AJS3 Agent の設定

JP1/Base, JP1/AJS3 Agent とLifeKeeper のインストールは、以下の手順をお勧めします。各ステップは詳細作業を後述しています。

- (1) LifeKeeper のインストールとコミュニケーションパスの作成
- (2) ファイルシステムリソース, 仮想IP リソースの作成
- (3) JP1/Base, JP1/AJS3 Agent のインストールと構築作業
- (4) Generic ARK for JP1/AJS3 Agent のインストールとセットアップ



(5) 依存関係の作成

上記が完了すると、LifeKeeper はJP1/Base, JP1/AJS3 Agent を保護できるようになります。

4.1 LifeKeeper のインストールとコミュニケーションパスの作成

本手順は、以下のドキュメントを参照してください。

- SIOS Protection Suite for Linux 9.2.2 インストレーションガイド
- SIOS Protection Suite for Linux 9.2.2 テクニカルドキュメンテーション

4.2 ファイルシステムリソース、仮想 IP リソースの作成

ファイルシステムリソースの作成は、テクニカルドキュメンテーションの「ファイルシステムリソース階層の作成」に詳細が記載されています。また、マルチパス環境では、使用するマルチパスドライバによって必要なリカバリーキットが変わります。詳細はリリースノートでご確認ください。

4.3 JP1/Base, JP1/AJS3 Agent のインストールと構築作業

JP1/Base, JP1/AJS3 Agent のインストール、設定に関しては、日立製作所のマニュアルを参照してください。

4.4 Generic ARK for JP1/AJS3 Agent のインストールとセットアップ

LifeKeeper に Generic ARK for JP1/AJS3 Agent をリソースとして設定する手順を次に示します。

4.4.1 インストール

- (1) Linux 版 Generic ARK for JP1/AJS3 Agent の CD に格納されている、以下のファイルを任意の path(ここでは/tmp/gen_jp1 とする)にコピーします。

- Generic_ARK_for_JP1_AJS3_Agent.tgz
- Generic_ARK_for_JP1_Base.tgz

実行例：

```
# cp -p Generic_ARK_for_JP1_AJS3_Agent.tgz /tmp/gen_jp1/.
# cp -p Generic_ARK_for_JP1_Base.tgz /tmp/gen_jp1/.
```

- (2) コピーしたファイルを展開します。

実行例：

```
# cd /tmp/gen_jp1
# tar xvzf Generic_ARK_for_JP1_AJS3_Agent.tgz
# tar xvzf Generic_ARK_for_JP1_Base.tgz
```



展開結果

/tmp/gen_jp1 ディレクトリ配下に次のディレクトリが作成されます。

Generic_ARK_for_JP1_AJS3_Agent

Generic_ARK_for_JP1_Base

上記のディレクトリ配下に各製品が提供するファイルが作成されます。

restore

remove

quickCheck

recover

4.4.2 パラメーターの設定

Generic ARK for JP1 が提供するスクリプトをリソースに設定する前に、次のパラメーターの設定変更が必要か確認し、設定が必要なパラメーターの変更を行ってください。

内容	パラメーター名	設定条件
リソースタグ名	APP	任意
論理ホスト名	VHOSTNAME	必須
タイムアウト値	DEFAULT_TIMEOUT	任意
	HANDLE_STOP_TIMEOUT	任意
待機時間(※)	APP_FORCE_STOP_WAIT_TIME	任意

※監視スクリプトは、設定の必要はありません。

(1) リソースタグ名の変更

リソースタグ名は、保護対象リソースを一意に認識するための名称です。Generic ARK for JP1では、パラメーター名「APP」に次のリソースタグ名をデフォルト値として設定しています。

Generic ARK for JP1	デフォルトのリソースタグ名
Generic ARK for JP1/Base	lkjp1_base
Generic ARK for JP1/AJS3 Agent	lkjp1_ajsagt

Active/Standby構成のようにLifeKeeperに1つのリソースしか設定しない場合は、デフォルト値の変更は必要ありません。しかし、Active/Activeのように複数のリソースを設定する場合は、1つのリソースに対して1つのリソースタグ名を設定する必要があり、それぞれのスクリプトで変更が必要となります。

指定するリソースタグ名は任意ですが、リソース設定を行うプログラムと、論理ホスト名を分ける



ように設定することを推奨します。例えば、論理ホスト名を「VirtualHostA」と「VirtualHostB」として作成した場合、リソースタグ名は、「lkjp1A_base」と「lkjp1B_base」にすると分かりやすく管理できます。なお、指定可能な文字は「A～Z, a～z, 0～9」「_」で指定文字数は最大 256 バイトです。

リソースタグ名を変更する場合は、展開された Generic ARK for JP1 の提供スクリプト (restore,remove,recover,quickCheck) 全てを変更します。リソースタグ名の変更方法を次に示します。なお、各スクリプトでの変更箇所はすべて同じであるため、Generic for ARK JP1/Base の起動スクリプト (restore) のリソースタグ名を「lkjp1A_base」に変更する場合を例に説明します。

- ① 展開された起動スクリプト (restore) をエディタで表示する
- ② 変更するパラメーターである「APP="lkjp1_base"」を検索する
- ③ 「APP="lkjp1_base"」の記載を「APP="lkjp1A_base"」に変更する
- ④ 起動スクリプトを保存して終了する

上記変更を、停止スクリプト (remove)、監視スクリプト (quickCheck)、再起動スクリプト (recover) についても実施します。

(2) 論理ホスト名の設定

論理ホスト名は、この JP1 インスタンスに対して定義した論理ホスト名である。論理ホスト名は、JP1 製品の構築時にユーザーが決定し使用するため、Generic ARK for JP1 では、論理ホスト名のパラメーター設定が必須となります。パラメーター名は VHOSTNAME を使用します。

論理ホスト名の設定は、展開された Generic ARK for JP1 の提供スクリプト (restore,remove,recover,quickCheck) 全てを変更します。論理ホスト名の設定方法を次に示します。なお、各スクリプトでの変更箇所はすべて同じであるため、起動スクリプト (restore) の論理ホスト名を「lkjp1」に変更する場合を例に説明します。

- ① 展開された起動スクリプト (restore) をエディタで表示する
- ② 変更するパラメーターである「VHOSTNAME="VirtualHostName"」を検索する
- ③ 「VHOSTNAME="VirtualHostName"」の記載を「VHOSTNAME="lkjp1"」に変更する
- ④ 起動スクリプトを保存して終了する

上記変更を、停止スクリプト (remove)、監視スクリプト (quickCheck)、再起動スクリプト (recover) についても実施します。

(3) タイムアウト値の変更

タイムアウト値は各スクリプトに設定するタイマー監視用の値です。提供スクリプトで実行した JP1 製品のコマンドが無応答となった場合は、設定したタイムアウト値によってタイマー監視を行い、提供スクリプトを終了させます。タイムアウト値を設定するパラメーター名には、



「DEFAULT_TIMEOUT」と「HANDLE_STOP_TIMEOUT」を使用します。

Generic ARK for JP1 で提供するスクリプトのタイムアウトのデフォルト値を次に示します。

スクリプト名	処理待ち時間 (DEFAULT_TIMEOUT)	後処理の待ち時間 (HANDLE_STOP_TIMEOUT)
起動スクリプト	480 秒	90 秒
停止スクリプト	120 秒	5 秒
再起動スクリプト	480 秒	90 秒
監視スクリプト	12 秒	5 秒

Generic ARK for JP1 で提供するスクリプトでは、タイムアウトは2フェーズで実施します。

- ・ 処理待ち時間
各スクリプトが目的とする処理を待つ時間です。
- ・ 後処理の待ち時間
処理待ち時間に設定した時間に、実際の処理が完了せずタイムアウトが発生した時に行われる後処理の完了を待つ時間です。なお、この値についての変更は推奨しません。

(a) 起動、停止、再起動スクリプト

JP1 製品によっては、データベースの再構成等を含むコマンドを実行する場合があります。デフォルトのタイムアウト値で管理してもタイムアウトが発生する場合があります。その場合は、各製品の動作を確認し、タイムアウト値を変更してください。

起動・停止・再起動スクリプトについては、Generic ARK for JP1 で提供するスクリプトのパラメーターで値を変更します。なお、各スクリプトでの変更箇所はすべて同じであるため、起動スクリプト (restore) の処理待ち時間を「1800 秒」に変更する場合を例に説明します。

- ① 展開された起動スクリプト (restore) をエディタで表示する
- ② 変更するパラメーターである「DEFAULT_TIMEOUT=480」を検索する
- ③ 「DEFAULT_TIMEOUT=480」の記載を「DEFAULT_TIMEOUT=1800」に変更する
- ④ 起動スクリプトを保存して終了する

上記のようにタイムアウト値の変更は、各スクリプトの対象パラメーターを検索し、編集して保存終了します。停止スクリプト (remove), 再起動スクリプト (recover) についても変更が必要な場合は同様の手順で変更してください。

(b) 監視スクリプト

監視スクリプトのタイムアウト値は、LifeKeeper によって管理されます。監視スクリプトの実行がタイムアウトに達した場合は、異常検出扱いとなり、再起動処理、また再起動処理の設定が無い



場合は、フェイルオーバーの対象となります。

監視スクリプトのデフォルトのタイムアウト値は 12 秒を設定しています。

監視スクリプトのタイムアウト値の変更方法は、起動、停止、再起動スクリプトと異なり、LifeKeeper ファイル (/etc/default/LifeKeeper) に、パラメーターを設定することでスクリプトのタイムアウト値を指定した秒数に調整することができます。LifeKeeper ファイルをエディタで表示し“リソースタグ名_TIMEOUT=タイムアウト値 (秒)”を追記します。

タイムアウト値として使用する値は、LifeKeeper ファイルに設定したタイムアウト値 (秒) から 10 秒引いた値がスクリプトの処理待ち時間となります。例として JP1/Base リソースのタグ名が lkjp1_base でタイムアウト値を 90 秒としたい場合は、次のように指定します。

```
lkjp1_base_TIMEOUT=100
```

監視スクリプトのタイムアウト値の変更を実施する場合は、実行系サーバー、待機系サーバーの両方で作業を行います。上記のパラメーターを修正すると、監視スクリプトの処理待ち時間 (DEFAULT_TIMEOUT) の値として適用します。監視スクリプトのタイムアウト値は、LifeKeeper ファイルの編集が行われた後に実施される監視処理で反映されます。

なお、タイムアウト値として設定できる最大値は、監視処理のサイクルの値となる。LifeKeeper の全リソースに対する監視処理のサイクルの初期値は 120 秒です。この設定は LifeKeeper ファイルの LKCHECKINTERVAL パラメーターで変更することができます。

監視スクリプトで使用するタイムアウト値を、リソースに対する監視処理のサイクルのデフォルト値である 120 秒より大きな値に変更したい場合は、リソースに対する監視処理のサイクルのデフォルト値 (LKCHECKINTERVAL パラメーターの値) もあわせて変更してください。

(4) 待機時間の変更

待機時間は各スクリプトに設定する待機時間の値です。提供スクリプトで実行した JP1 製品の強制停止コマンドがバックグラウンドで動作中の可能性を考慮し、待機時間待機します。待機時間を設定するパラメーター名には、「APP_FORCE_STOP_WAIT_TIME」を使用します。

JP1 製品によっては、データベースの再構成等を含むコマンドを実行する場合があります。デフォルトの待機時間で待機しても強制停止が完了しないケースがあります。その場合は、各製品の動作を確認し、待機時間を変更します。

起動・停止・再起動スクリプトについては、Generic ARK for JP1 で提供するスクリプトのパラメーターで値を変更します。なお、各スクリプトでの変更箇所はすべて同じであるため、起動スクリプト (restore) の待機時間 5 秒に変更する場合を例に説明します。

1. 展開された起動スクリプト (restore) をエディタで表示する
2. 変更するパラメーターである「APP_FORCE_STOP_WAIT_TIME=5」を検索する
3. 「APP_FORCE_STOP_WAIT_TIME=5」の記載を



「APP_FORCE_STOP_WAIT_TIME=変更後の待機時間」に変更する

4. 起動スクリプトを保存して終了する

上記のように待機時間の変更は、各スクリプトの対象パラメーターを検索し、編集して保存終了します。停止スクリプト、再起動スクリプトについても変更が必要な場合は同様の手順で実施してください。

4.4.3 Generic ARK for JP1/Base のセットアップ

Generic ARK for JP1/Base で提供する各スクリプトは、LifeKeeper の Generic ARK リソースとして設定する必要があります。リソースを設定する際には、ノード上で先に作成した仮想 IP リソース、ファイルシステムリソースが、共に ISP(稼動中)であることを確認してから実施してください。

各スクリプトを Generic ARK リソースとして設定する手順を次に示します。なお、スクリプトに必用なユーザーインターフェースとして、論理ホスト名の設定(「4.4.2 パラメーターの設定」(2) 論理ホスト名の設定)はリソース設定よりも先に実施しておいてください。手順において指定するファイルのディレクトリは「/tmp/gen_jp1/Generic_ARK_for_JP1_Base」ディレクトリに展開したものと説明します。

なお、Generic ARK リソースの設定に関する一般的な情報は、テクニカルドキュメンテーションの「Generic Application リソース階層の作成」を参照してください。

・設定手順

- ① LifeKeeper の GUI 管理画面から、「Create Resource Hierarchy」を選択し、Create Resource Wizard を起動します。リカバリーキットの選択画面で「Generic Application」を選択します。
- ② Restore Script 選択画面で、以下のファイルを指定します。
/tmp/gen_jp1/Generic_ARK_for_JP1_Base/restore
- ③ Remove Script 選択画面で、以下のファイルを指定します。
/tmp/gen_jp1/Generic_ARK_for_JP1_Base/remove
- ④ quickCheck Script 選択画面で、以下のファイルを指定します。
/tmp/gen_jp1/Generic_ARK_for_JP1_Base/quickCheck
- ⑤ Local Recovery Script 選択画面で、以下のファイルを指定します。
/tmp/gen_jp1/Generic_ARK_for_JP1_Base/recover
- ⑥ Application Information フィールドは任意ですが、作成する JP1 リソースの論理ホスト名を入力しておくことを推奨します。
- ⑦ Bring Resource In Service で Yes を選択します。この時点でリソース設定対象プログラムの restore スクリプトが実行され、対象プログラムが起動状態となります。既に対象プログラムが起動状態であっても問題はありません。ここで No を応答した場合は、他ノードへの拡張が行われません。その場合は後で他ノードへ拡張する作業が必要となります。
- ⑧ Root Tag には、リソースタグ名を指定する。ユーザーインターフェースで変更を行っていない場合は、デフォルトのリソースタグ名を指定します。リソースタグ名の変更を行っている場



合は、変更したリソースタグ名を指定してください。

- ⑨ **Create Instance** を押して、リソースの作成を行います。その後、必要なノードへの **extend** を行います。

注意事項

- ・設定手順④の **quickCheck Script** 選択、または設定手順⑤の **Local Recovery Script** 選択は、スクリプトを指定せずにスキップすることができます。スキップした場合は、次の動作となります。
 - **quickCheck Script** 選択をスキップすると、**LifeKeeper** は **JP1** 製品の障害を検知しません。
 - **Local Recovery Script** 選択をスキップすると、障害検知時に実行系サーバーでの再起動を試みることなく、待機系サーバーへフェイルオーバーを行います。
- ・**quickCheck** を実施する際の **JP1** システムのパフォーマンスの影響により、デフォルトのタイムアウト値である 12 秒で監視が完了しない場合があります。実業務中に監視の応答が返らないことによって再起動、または、フェイルオーバーが発生すると実業務に影響があります。このため、監視スクリプト(**quickCheck**)と再起動スクリプト(**recover**)を **LifeKeeper** に設定する場合は、**JP1** システムのパフォーマンス状況を確認して適切なタイムアウト値を設定してください。
- ・**Bring Resource In Service** に **NO** を指定した場合、他ノードへの拡張が行われません。リソースの拡張に関する具体的な手順は、テクニカルドキュメンテーションの「**Generic Application** リソース階層の拡張」を参照して他ノードへの拡張を実施してください。

4.4.4 Generic ARK for JP1/AJS3 Agent のセットアップ

Generic ARK for JP1/AJS3 Agent で提供する各スクリプトは、**LifeKeeper** の **GenericARK** リソースとして設定する必要があります。リソースを設定する際には、ノード上で先に作成した仮想 IP リソース、ファイルシステムリソース、**JP1/Base** のリソースが、共に **ISP(稼動中)**であることを確認してから実施してください。

各スクリプトを **GenericARK** リソースとして設定する手順を次に示します。なお、スクリプトに必要なユーザーインターフェースとして、論理ホスト名の設定（「4.4.2 パラメーターの設定」(2) 論理ホスト名の設定）はリソース設定よりも先に実施しておいてください。手順において指定するファイルのディレクトリは「/tmp/gen_jp1/Generic_ARK_for_JP1_AJS3_Agent」ディレクトリに展開したものとして説明します。

なお、**GenericARK** リソースの設定に関する一般的な情報は、テクニカルドキュメンテーションの「**Generic Application** リソース階層の作成」を参照してください。

・設定手順

- ① **LifeKeeper** の GUI 管理画面から、「**Create Resource Hierarchy**」を選択し、**Create Resource Wizard** を起動する。リカバリーキットの選択画面で「**Generic Application**」を選択します。
- ② **Restore Script** 選択画面で、以下のファイルを指定します。



- /tmp/gen_jp1/Generic_ARK_for_JP1_AJS3_Agent/restore
- ③ Remove Script 選択画面で、以下のファイルを指定します。
- /tmp/gen_jp1/Generic_ARK_for_JP1_AJS3_Agent/remove
- ④ quickCheck Script 選択画面で、以下のファイルを指定します。
- /tmp/gen_jp1/Generic_ARK_for_JP1_AJS3_Agent/quickCheck
- ⑤ Local Recovery Script 選択画面で、以下のファイルを指定します。
- /tmp/gen_jp1/Generic_ARK_for_JP1_AJS3_Agent/recover
- ⑥ Application Information フィールドは任意ですが、作成する JP1 リソースの論理ホスト名を入力しておくことを推奨します。
- ⑦ Bring Resource In Service に Yes を選択します。この時点でリソース設定対象プログラムの restore スクリプトが実行され、対象プログラムが起動状態となります。既に対象プログラムが起動状態であっても問題はありません。ここで No を応答した場合は、他ノードへの拡張が行われません。その場合は後で他ノードへ拡張する作業が必要となります。
- ⑧ Root Tag には、リソースタグ名を指定します。ユーザーインターフェースで変更を行っていない場合は、デフォルトのリソースタグ名を指定します。リソースタグ名の変更を行った場合は、変更したリソースタグ名を指定してください。
- ⑨ Create Instance を押して、リソースの作成を行います。その後、必要なノードへの extend を行います。

注意事項

- ・ 設定手順④の quickCheck Script 選択、または設定手順⑤の Local Recovery Script 選択は、スクリプトを指定せずにスキップすることができます。スキップした場合は、次の動作となります。
 - quickCheck Script 選択をスキップすると、LifeKeeper は JP1 製品の障害を検知しません。
 - Local Recovery Script 選択をスキップすると、障害検知時に実行系サーバーでの再起動を試みることなく、待機系サーバーへフェイルオーバーを行います。
- ・ quickCheck を実施する際の JP1 システムのパフォーマンスの影響により、デフォルトのタイムアウト値である 12 秒で監視が完了しない場合があります。実業務中に監視の応答が返らないことによって再起動、または、フェイルオーバーが発生すると実業務に影響があります。このため、監視スクリプト(quickCheck)と再起動スクリプト(recover)を LifeKeeper に設定する場合は、JP1 システムのパフォーマンス状況を確認して適切なタイムアウト値を設定してください。
- ・ Bring Resource In Service に NO を指定した場合、他ノードへの拡張が行われません。リソースの拡張に関する具体的な手順は、テクニカルドキュメンテーションの「Generic Application リソース階層の拡張」を参照して他ノードへの拡張を実施してください。

4.5 依存関係の作成

ここまでの作業で、以下のリソースが作成されていることとなります。



- JP1/AJS3 Agent Generic ARK リソース
- JP1/Base Generic ARK リソース
- 仮想IP リソース
- ファイルシステムリソース

これらのリソースは、この時点では依存関係が作成されていません。これらの各リソースが同時にフェイルオーバーを行うためには、各リソースの間に依存関係を設定する必要があります。

依存関係を作成して、1つのJP1 リソースツリーが以下のような構成となるよう設定してください。

リソース種別	順番
JP1/AJS3 Agent Generic ARK リソース	0 (root)
JP1/Base Generic ARK リソース	1
仮想IP リソース	2
ファイルシステムリソース	3
disk	4
device	5

依存関係を設定するには、GUI 管理画面より、**Create Dependency** メニューを選択してください。なお、依存関係作成の具体的な手順は、テクニカルドキュメンテーションの「リソース依存関係の作成」を参照してください。

- ① **LifeKeeper** の GUI 管理画面より、「**Create Dependency**」メニューを選択します。
必用に応じて、「**Server**」と、上位のリソースを「**Parent Resource Tag**」から選択して、「**Next**」をクリックします。
- ② 下位のリソースを「**Child Resource Tag**」から選択して、「**Next**」をクリックします。
- ③ リソースの依存関係が表示されます。「**Create Dependency**」をクリックして依存関係を作成してください。
- ④ 拡張サーバーの依存関係も作成されます。「**Done**」をクリックして終了します。

5. Generic ARK for JP1/AJS3 Agent リソースのその他の操作

5.1 Generic ARK for JP1/AJS3 Agent リソースの依存関係の削除

LifeKeeper で各リソースが同時にフェイルオーバーを行うために、各リソースの間に依存関係を設定している場合は、登録した依存関係の削除が必要となります。

依存関係を削除するには、**LifeKeeper** の GUI 管理画面より、「**Delete Dependency**」メニューを選択してください。



依存関係作成の具体的な手順は、テクニカルドキュメンテーションの「リソース依存関係の削除」を参照してください。

- ① LifeKeeper の GUI 管理画面より、「Delete Dependency」メニューを選択します。
必用に応じて、「Server」と、上位のリソースを「Parent Resource Tag」から選択して、「Next」をクリックします。
- ② 下位のリソースが「Child Resource Tag」に表示されます。「Next」をクリックします。
- ③ リソースの依存関係が表示されます。「Delete Dependency」をクリックして依存関係を削除してください。
- ④ 拡張サーバーの依存関係も削除されます。「Done」をクリックして終了します。

注意事項

依存関係の削除を実施せず、リソースの拡張解除またはリソースの削除を実施すると、root として登録されているリソースをターゲットリソースとしますのでご注意ください。

5.2 Generic ARK for JP1/AJS3 Agent リソース階層の削除

LifeKeeper 設定内のすべてのサーバーから登録したリソース階層を削除するには、LifeKeeper の GUI 管理画面より、「Delete Resource Hierarchy」メニューを選択してください。

依存関係作成の具体的な手順は、テクニカルドキュメンテーションの「すべてのサーバからの階層の削除」を参照してください。

- ① 「Edit」メニューの「Resource」から「Delete Resource Hierarchy」を選択します。
- ② JP1/AJS3 Agent リソース階層を削除する「Target Server」の名前を選択します。
注：左側ペインのグローバルリソースまたは右側ペインの個々のリソースインスタンスを右クリックして「Delete Resource Hierarchy」作業を選択した場合、このダイアログボックスは表示されません。
- ③ 「Hierarchy to Delete」を選択します（右側ペインでリソースインスタンスを右クリックして「Delete Resource」作業を選択した場合は、このダイアログボックスは表示されません）。「Next」をクリックします。
- ④ 選択したターゲットサーバーと、削除の対象として選択した階層を確認する情報ボックスが表示されます。「Next」をクリックします。
- ⑤ JP1/AJS3 Agent リソースが正常に削除されたことを確認する別の情報ボックスが表示されます。
- ⑥ 「Done」をクリックして終了します。続いてJP1/Base リソースに対して、1~6の手順を実施してください。

5.3 Generic ARK for JP1/AJS3 Agent リソース階層のテスト

JP1/AJS3 Agent リソース階層のテストは、手動でリソースの切り替えを実施してください。この



操作は、プライマリサーバーからバックアップサーバーへのリソースインスタンスのフェイルオーバーをシミュレートして実行できます。

LifeKeeper GUI からの手動切り替えの実行

LifeKeeper GUI で「Edit」 > 「Resource」 > 「In Service」を選択すると、手動切り替えを開始できます。たとえば、バックアップサーバーでサービス中リクエストが実行されると、JP1/AJS3 Agent リソース階層がバックアップサーバー側でサービス中になり、プライマリサーバー側ではサービス休止になります。この時点で、元のバックアップサーバーがプライマリサーバーになり、元のプライマリサーバーがバックアップサーバーになります。

「Out of Service」要求を実行した場合、リソース階層はサービス休止になりますが、別のサーバーでサービス中にはなりません。

6. ログメッセージの解説

Generic ARK for JP1 では、障害発生時に備えログファイルにメッセージを出力しログ情報を残します。Generic ARK for JP1 で出力するメッセージと出力ファイルを、次に示します。

(1) 出力ファイル

Generic ARK for JP1 では、メッセージを次に示すファイルに出力します。

(a) LifeKeeper ログ

```
/var/log/lifekeeper.log
```

(2) メッセージの形式

Generic ARK for JP1 で出力するメッセージの形式を次に示します。

```
出力時刻 ホスト名 スクリプト名「数値」:重要度:ソース名:リソース動作:リソースタグ名:メッセージ ID:論理ホスト名 :メッセージ
```

●説明

出力時刻 : メッセージ出力時刻

ホスト名 : 実行ホスト名

スクリプト名「数値」: 実行スクリプト名 (数値は Lifekeeper が設定する)

重要度 : INFO (情報) , WARN (警告) , または ERROR (エラー)

ソース名 : gen 固定

リソース動作 : restore, remove, recover, または quickCheck

リソースタグ名 : 対象となるリソースのタグ名



メッセージ ID : 127000 固定

論理ホスト名 : メッセージを出力するリソースに割り当てられている論理ホスト名

メッセージ : 出力メッセージ



(3) LifeKeeper ログに出力するメッセージ一覧

LifeKeeper ログに Generic ARK for JP1 が出力するメッセージの一覧を次に示します。

(a) 共通メッセージ

各製品で共通に出力するメッセージを次に示します。

●起動, 停止, 再起動スクリプトで出力する共通メッセージ

出力メッセージ	重要度	内容	対処
Starting.	情報	起動コマンドの実行ログ	—
Stopping.	情報	停止コマンドの実行ログ	—
Force stopping.	情報	強制停止コマンドの実行ログ	—

●起動, 停止, 監視, 再起動スクリプトで出力する共通メッセージ

出力メッセージ	重要度	内容	対処
Script was hung. Forcibly terminating. Timeout value=タイムアウト値	エラー	各スクリプトにおいて, タイムアウトが発生したログ	タイムアウト値を確認して再設定してください。
コマンド名 doesn't exist.	エラー	コマンドまたはスクリプトファイルが存在しなかった, または, 実行権限が付加されていないファイルであった	表示されたコマンドまたは, スクリプトファイルのファイルの存在と実行権限を確認してください。
実行コマンド	情報	実行したコマンド	—

●起動スクリプトで出力する共通メッセージ

出力メッセージ	重要度	内容	対処
Restore start.	情報	リソース起動処理の開始	—
Already started.	警告	起動を試みたが対象のインスタンスは既に起動状態であった	—
Start up completed.	情報	起動処理の成功	—
Start up failed. ret=リターンコード	エラー	起動処理の失敗	起動に失敗したリソースを確認し, 起動失敗の要因を取り除き「In Service」を実施してください。



●停止スクリプトで出力する共通メッセージ

出力メッセージ	重要度	内容	対処
Remove start.	情報	リソース停止処理の開始	—
Already stopped.	警告	停止を試みたが対象のインスタンスは既に停止状態であった	—
Stop completed.	情報	停止処理の成功	—
Stop failed. ret=リターンコード	エラー	停止処理の失敗	停止に失敗したリソースを確認してください。 停止失敗の要因を取り除き対象リソースを停止してください。 フェイルオーバーが必要な場合は、待機系サーバーで「In Service」を実施してください。

●監視スクリプトで出力する共通メッセージ

出力メッセージ	重要度	内容	対処
Resource is not ISP status.	警告	リソースが ISP(In-Service)状態ではない	—
QuickCheck failed. ret=リターンコード	エラー	監視で障害を検知	— (再起動, またはフェイルオーバーが実施されます)

●再起動スクリプトで出力する共通メッセージ

出力メッセージ	重要度	内容	対処
Recover start.	情報	リソース再起動処理の開始	—
Now functioning properly. No recovery needed.	警告	quickchk で障害を検知したが、再確認で正常起動が確認された	—
Recovery start up completed.	情報	再起動処理の成功	—
Recovery start up failed. ret=リターンコード	エラー	再起動処理の失敗	— (フェイルオーバーが実施されます)

以上

