

# LifeKeeper for Windows HULFT 動作検証レポート

---

第 1 版

## 目次

<b>1.環境</b> .....	3
<b>2.設定手順</b> .....	4
2-1. ノード 1 とノード 2 上に仮想サーバーを構築.....	6
2-2. ノード 1 とノード 2 に Windows Server 2016 をインストール.....	6
2-3.ネットワークの確立、ホスト名の名前解決.....	6
2-4. ドメインネットワークへの参加.....	7
2-5. LifeKeeper のインストール.....	8
2-6. License Key のインストール.....	8
2-7.LifeKeeper の起動.....	8
2-8.LifeKeeper GUI の起動.....	9
2-9.クラスタの設定.....	9
2-10. ボリュームリソース、IP リソースの作成.....	9
2-11. 各ノードへの HULFT のインストール.....	11
2-12. HULFT システム動作環境の設定(hulenv.cnf).....	22
2-13. HULFT リソースの作成.....	25
2-14. リソース依存関係の作成.....	31
2-15.HULFT 階層のバックアップサーバへの手動スイッチオーバー.....	36
2-16.詳細ホスト情報、転送グループ情報の設定.....	38
2-17.配信管理情報、集信管理情報の設定.....	43
2-18.ファイル配信.....	46
【参考情報】.....	51

本ドキュメントは、2台の仮想サーバーで共有ディスクを構成した環境に対して、LifeKeeper for Windows v8.6.1クラスタおよび HULFT for Windows v8.1.3を導入し、HULFTサービスを汎用アプリケーションリソースとしてリソース登録するためのガイドです。

注) 当資料が対象とするバージョン以外のOS、HULFT for Windows、LifeKeeper for Windows をご使用の場合は、インストールや設定の手順が異なる場合があります。その場合は、ご使用のバージョンのマニュアルやRelease Notes等に記載された手順に従ってください。

## 1.環境

---

実際に使用した環境は、以下になります。

### H/W環境

サーバー : VMware 仮想サーバー

メモリ : 4.0 GB

CPU: Intel(R) Xeon® E5-2637 v2 3.50GHz

### S/W環境

O/S : Microsoft Windows Server 2016 Standard Edition

クラスタソフトウェア : LifeKeeper for Windows v8.6.1

HULFTソフトウェア : HULFT for Windows v8.1.3

2台のサーバーでクラスタを構成します。各サーバーのOSや設定は同一にします。それぞれのサーバー上の OS,ホストを、本ドキュメントではクラスタに所属するホストを示すノード1、ノード2と表記します。

## 2. 設定手順

---

実際の手順の流れは以下になります。

- 1) ノード1とノード2上に仮想サーバーを構築
- 2) ノード1とノード2にWindows Server 2016 をインストール
- 3) ネットワークの確立、ホスト名の名前解決
- 4) ドメインネットワークへの参加
- 5) LifeKeeper のインストール
- 6) License Key のインストール
- 7) LifeKeeper の起動と停止
- 8) LifeKeeper GUI の起動
- 9) クラスタの設定
- 1 0) ボリュームリソース、IPリソースの作成
- 1 1) 各ノードへのHULFT のインストール
- 1 2) HULFTシステム動作環境の設定(hulenv.cnf)
- 1 3) HULFT リソースの作成
- 1 4) リソース依存関係の作成
- 1 5) HULFT 階層のバックアップサーバへの手動スイッチオーバー
- 1 6) 詳細ホスト情報、転送グループ情報の設定
- 1 7) 配信管理情報、集信管理情報の設定
- 1 8) ファイル配信

LifeKeeper for Windows の導入および設定について詳しくは、以下のドキュメントを参照してください。

### SIOS Protection Suite インストレーションガイド

<http://jpdocs.us.sios.com/WindowsSPS/8.6/SPS4W/SPSInstall/index.htm>

### SIOS Protection Suite テクニカルドキュメンテーション

#### 設定

<http://jpdocs.us.sios.com/WindowsSPS/8.6.1/SPS4W/TechDoc/index.htm#Configuration.htm>

## LifeKeeper for Windows HULFT クラスタ構成ガイド

LifeKeeper for Windows を仮想環境で使用する場合の考慮事項について詳しくは、以下のドキュメントを参照してください。

### SIOS Protection Suite for Windows

#### SIOS DataKeeper Cluster Edition 仮想環境構成ガイド(VMware vSphere6 編)

[https://sios.jp/products/lkdk/product/pdf/vmware\\_vsphere6.pdf](https://sios.jp/products/lkdk/product/pdf/vmware_vsphere6.pdf)

HULFT の導入および設定について詳しくは、HULFT のマニュアルを参照してください。

各項目の具体的な作業内容を以降に記述します。

以下の構成図（図 1）を参考に、構成します。

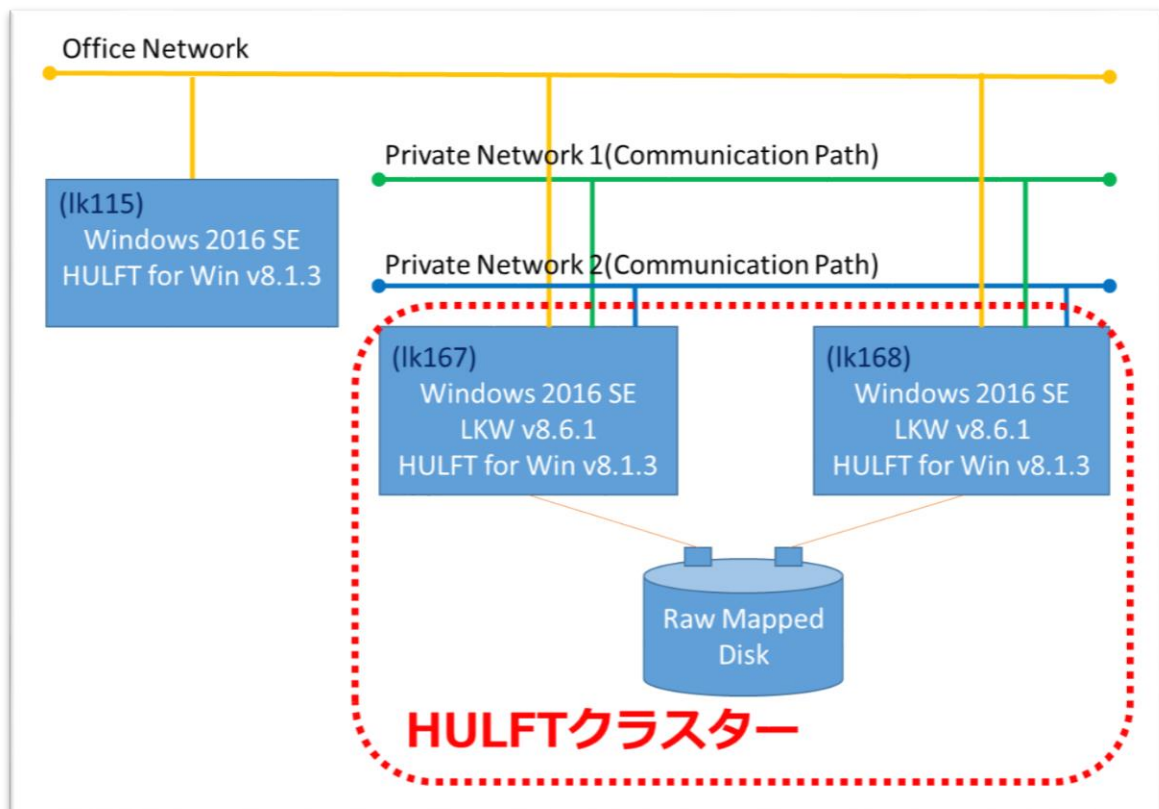


図 1

### 2-1. ノード 1 とノード 2 上に仮想サーバーを構築

2 台の仮想サーバーを作成します。ここでは、vSphere Client 上で「新規仮想マシンの作成」ウィザードを使用して、「標準」の構成を利用して作成します。

今回は、前述の構成図の通りとなるよう、以下の通り設定します。

[ストレージ]

ホストサーバーのローカルディスクを指定

[ネットワーク]

ネットワークの数 . . . . . 3

- NIC1, Office Network, E1000, “パワーオン時に接続”にチェック
- NIC2, Private Network 1, E1000, “パワーオン時に接続”にチェック
- NIC3, Private Network 2, E1000, “パワーオン時に接続”にチェック

### 2-2. ノード 1 とノード 2 に Windows Server 2016 をインストール

ノード1とノード2に Microsoft Windows Server 2016 をインストールします。グラフィカルインターフェースが必要となるため、「インストールするオペレーティングシステムを選んでください」の画面では “デスクトップエクスペリエンス” を選択します。

### 2-3. ネットワークの確立、ホスト名の名前解決

LifeKeeper for Windows では、クラスタで使用するサービスネットワークとは別セグメント上にコミュニケーションパスを2経路以上設定します。事前に、クラスタノード間のそれぞれのネットワークでpingによる疎通が可能であることを確認してください。

**注：Windows Server 2008 R2 以降では、初期インストール時は ping 応答が許可されていません。必要に応じて、Windows スタートメニュー > 管理ツール > セキュリティが強化された Windows ファイアウォール から ping 応答を許可するよう設定してください。**

## LifeKeeper for Windows HULFT クラスタ構成ガイド

また、LifeKeeper ではホスト名を使用して通信を行います。その為、ノード間では DNS や hosts ファイルを使用して名前解決できるようにしてください。

Windows Server 2016 の host ファイルは以下のパスにあります。

C:\Windows\System32\drivers\etc\hosts

今回は、Windows コントロール パネル¥ネットワークとインターネット¥ネットワーク接続 の画面を使用して以下のように設定します。

[ノード1]

Office Network	10.1.5.167/16
Private Network 1	192.168.1.167/24
Private Network 2	172.16.1.167/24
Default Gateway	10.1.0.1
DNS	10.1.5.90

[ノード2]

Office Network	10.1.5.168/16
Private Network 1	192.168.1.168/24
Private Network 2	172.16.1.168/24
Default Gateway	10.1.0.1
DNS	10.1.5.90

### 2-4. ドメインネットワークへの参加

LifeKeeper for Windows はドメイン環境での使用が推奨されます。Windows コントロール パネル¥システムとセキュリティ¥システム からシステムのプロパティを起動して、ドメインネットワークへ参加します。今回は、以下とおり設定します。

[ノード1]

コンピューター名	lk167
所属するグループ	ドメイン(lkg.local)

[ノード2]

コンピューター名	lk168
所属するグループ	ドメイン(lkg.local)

### 2-5. LifeKeeper のインストール

LifeKeeper for Windows v8.6.1 を、ノード 1、ノード 2 の両サーバー上にインストールします。今回は、全てデフォルトのインストールオプションを使用します。

インストールの具体的な手順については、以下のガイドを参照してください。

#### **SIOS Protection Suite インストレーションガイド(v8.6.1)**

<http://jpdocs.us.sios.com/WindowsSPS/8.6.1/SPS4W/SPSInstall/index.htm>

### 2-6. License Key のインストール

LifeKeeper for Windows のライセンスキーをノード 1、ノード 2 の両サーバーにインストールします。各ノードのサーバー上の任意のパスにライセンスキーを配置しておきます。インストールウィザード内で、またはインストール後に License Key Installer を起動することで、ライセンスの導入を行います。

ライセンス導入の具体的な手順については、以下のガイドを参照してください。

#### **ライセンスの取得とインストール**

[http://jpdocs.us.sios.com/WindowsSPS/8.6.1/SPS4W/SPSInstall/index.htm#Installation/Installing\\_SPS/Obtaining\\_and\\_Installing\\_the\\_License.htm](http://jpdocs.us.sios.com/WindowsSPS/8.6.1/SPS4W/SPSInstall/index.htm#Installation/Installing_SPS/Obtaining_and_Installing_the_License.htm)

### 2-7. LifeKeeper の起動

LifeKeeper を起動するために、両ノードで以下 1) または 2) いずれかの手順を実施します。

1) lkstart コマンドを実行します。

```
C:¥LK¥Bin¥lkstart
```



2) Windows サービスメニューから LifeKeeper サービス > 開始 を選択します。

### 2-8. LifeKeeper GUI の起動

A) LifeKeeper GUI を起動します。

Windows スタートメニューから、LifeKeeper(Admin Only)を選択します。

B) LifeKeeper にログインします。

Server Name には実行したノード名が入ります。ログインユーザ名とパスワードは LifeKeeper の管理ユーザーの情報を入力します。管理ユーザーの情報は、初期設定として OS のスーパーユーザー(administrator)とそのパスワードが設定されています。

### 2-9. クラスタの設定

クラスタード間コミュニケーションパスを構成します。クラスタの設定の具体的な手順については、以下のガイドを参照してください。

### SIOS Protection Suite の設定手順

[http://jpdocs.us.sios.com/WindowsSPS/8.6.1/SPS4W/TechDoc/index.htm#Configuration/SPS\\_Configuration\\_Steps.htm](http://jpdocs.us.sios.com/WindowsSPS/8.6.1/SPS4W/TechDoc/index.htm#Configuration/SPS_Configuration_Steps.htm)

### 2-10. ボリュームリソース、IP リソースの作成

クラスタ構成が完了したら、クラスタで保護するリソースを作成します。最終的に以下のようなリソース階層を構成します。

HULFT サービス (汎用アプリケーションリソース)

- └ 仮想 IP アドレス (IP リソース)
- └ 共有ディスク (ボリュームリソース)

A) ボリュームリソースを作成します。

事前に、2 台の仮想サーバーに対して、共有ディスクとしてひとつの LUN をマップし

ておきます。次に、Windows ディスクの管理画面で、ディスクのオンライン化、フォーマットを行い、ドライブレターをアサインしておきます。

今回は、共有ディスクにドライブレター H: をアサインします。仮想環境上の共有ディスク構成については、以下のガイドを参照してください。

### **SIOS Protection Suite for Windows**

#### **SIOS DataKeeper Cluster Edition 仮想環境構成ガイド(VMware vSphere6 編)**

[https://sios.jp/products/lkdk/product/pdf/vmware\\_vsphere6.pdf](https://sios.jp/products/lkdk/product/pdf/vmware_vsphere6.pdf)

ボリュームリソースの作成の具体的な手順については、以下のガイドを参照してください。

#### **ボリュームリソース階層の作成**

[http://jpdocs.us.sios.com/WindowsSPS/8.6.1/SPS4W/TechDoc/index.htm#Administration/Administrator\\_GUI\\_Tasks/Creating\\_Resource\\_Hierarchies/Creating\\_a\\_Volume\\_Resource\\_Hierarchy.htm](http://jpdocs.us.sios.com/WindowsSPS/8.6.1/SPS4W/TechDoc/index.htm#Administration/Administrator_GUI_Tasks/Creating_Resource_Hierarchies/Creating_a_Volume_Resource_Hierarchy.htm)

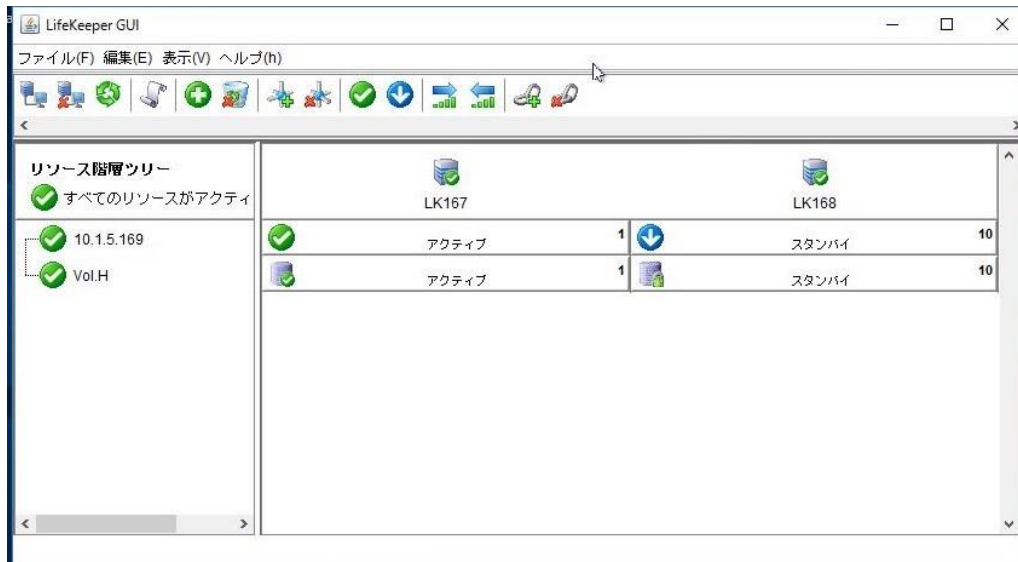
B) IP リソースを作成します。

IP リソースの作成の具体的な手順については、以下のガイドを参照してください。

#### **IP アドレスリソース階層の作成**

[http://jpdocs.us.sios.com/WindowsSPS/8.6.1/SPS4W/TechDoc/index.htm#Administration/Administrator\\_GUI\\_Tasks/Creating\\_Resource\\_Hierarchies/Creating\\_an\\_IP\\_Address\\_Resource\\_Hierarchy.htm](http://jpdocs.us.sios.com/WindowsSPS/8.6.1/SPS4W/TechDoc/index.htm#Administration/Administrator_GUI_Tasks/Creating_Resource_Hierarchies/Creating_an_IP_Address_Resource_Hierarchy.htm)

ここで設定した仮想 IP アドレスを使用して、この後 HULFT クラスタの設定を行います。  
ここまでの手順で、以下の図の通りリソースが作成されています。



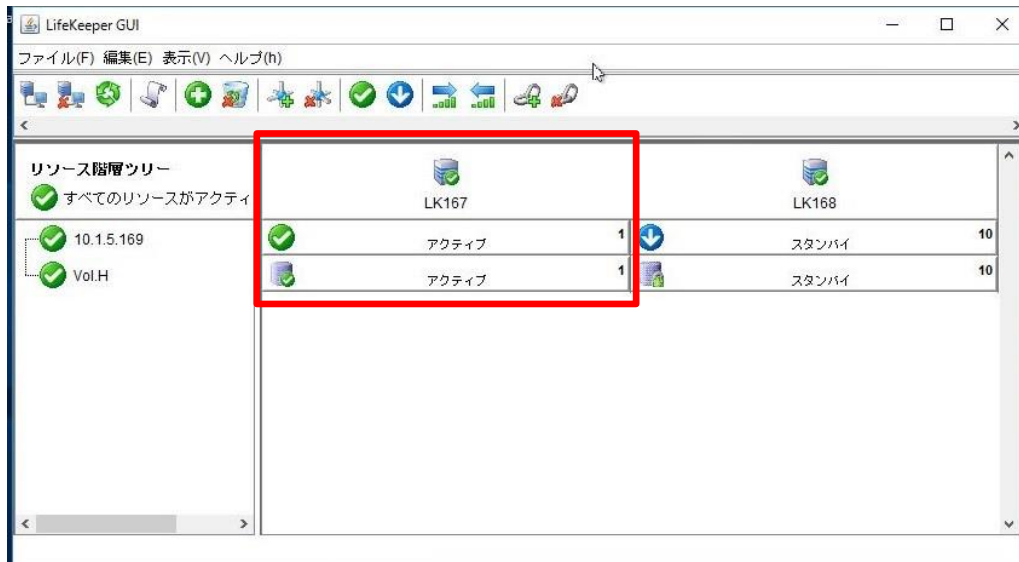
### 2-11. 各ノードへの HULFT のインストール

HULFT for Windows v8.1.3 を、ノード 1、ノード 2 の順にインストールします。  
インストールについては、HULFT のガイドを参照してください。

### HULFT 8 導入マニュアル(Windows)

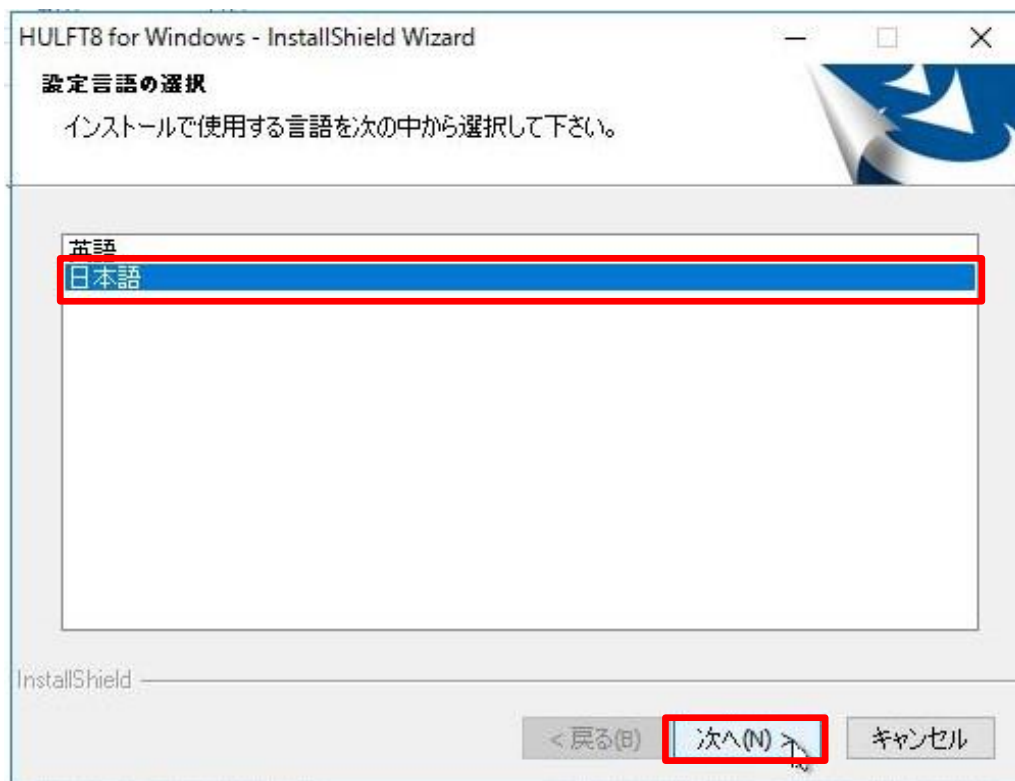
<HULFT メディア>¥Manual(PDF)¥jpn¥HULFT8\_JP\_INS\_WIN

- A) まずはノード 1 上に HULFT をインストールします。はじめは以下のように、ノード 1 側でリソースがアクティブになっていることを確認します。



- B) ノード 1 上で、HULFT のインストールメディアから setup を起動します。HULFT インストール時に「必須ソフトウェアがインストールされていません。」といったメッセージが出力しインストールが中断する場合は、インストールメディアの ISSetupPrerequisites フォルダに含まれる Visual C++再頒布可能パッケージのインストールを行ってから、setup を起動します

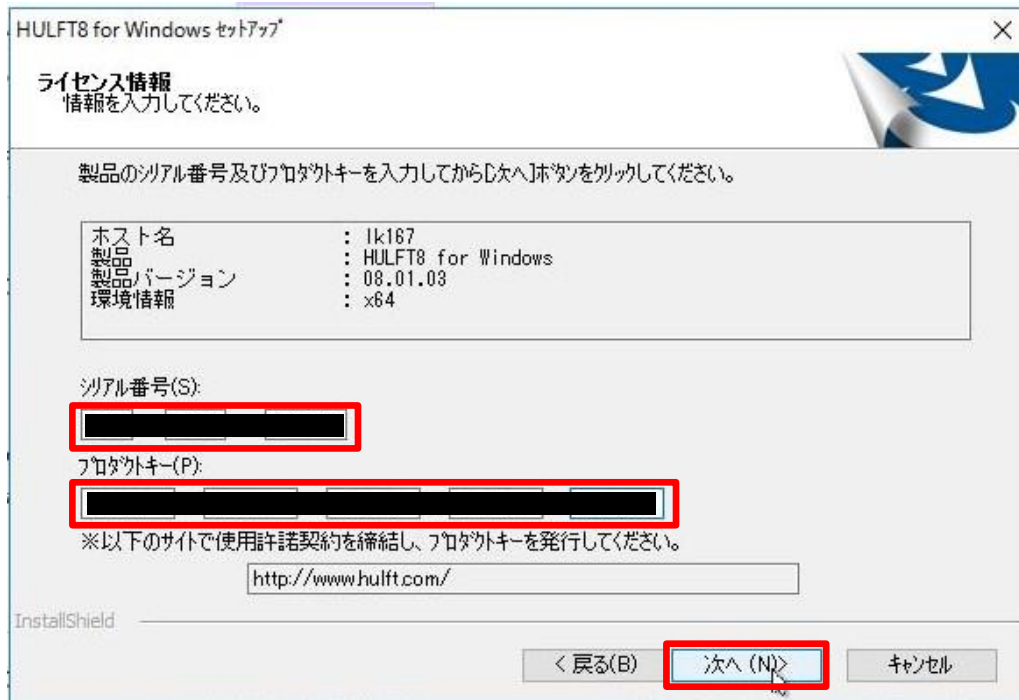
C) 設定言語の選択画面で、日本語を選択し[次へ]を押下します。



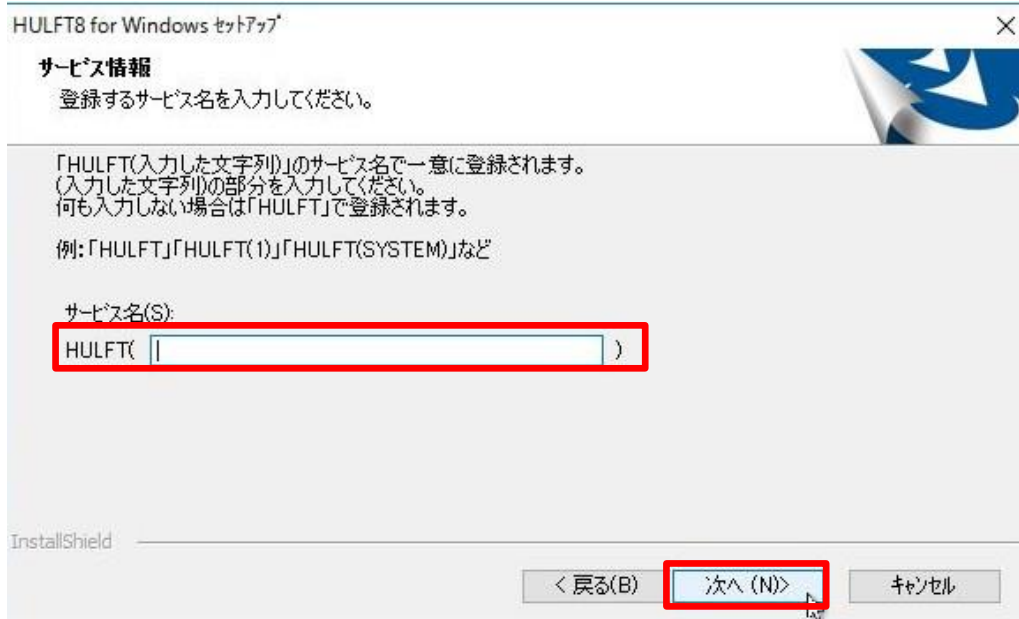
D) [次へ]を押下します。



- E) ライセンス情報の入力画面で、シリアル番号とプロダクトキーを入力し[次へ]を押下します。



- F) サービス名を入力して[次へ]を押下します。今回は、何も入力せずデフォルトのサービス名のままとします。

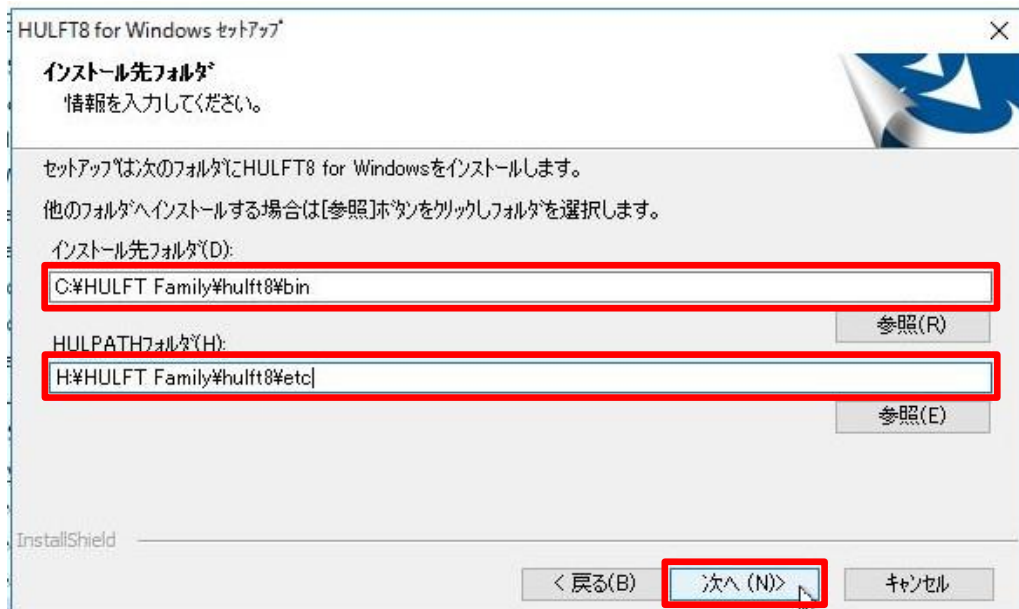


何も入力しない場合注意喚起のウィンドウが表示されるので、[はい]を押下します。

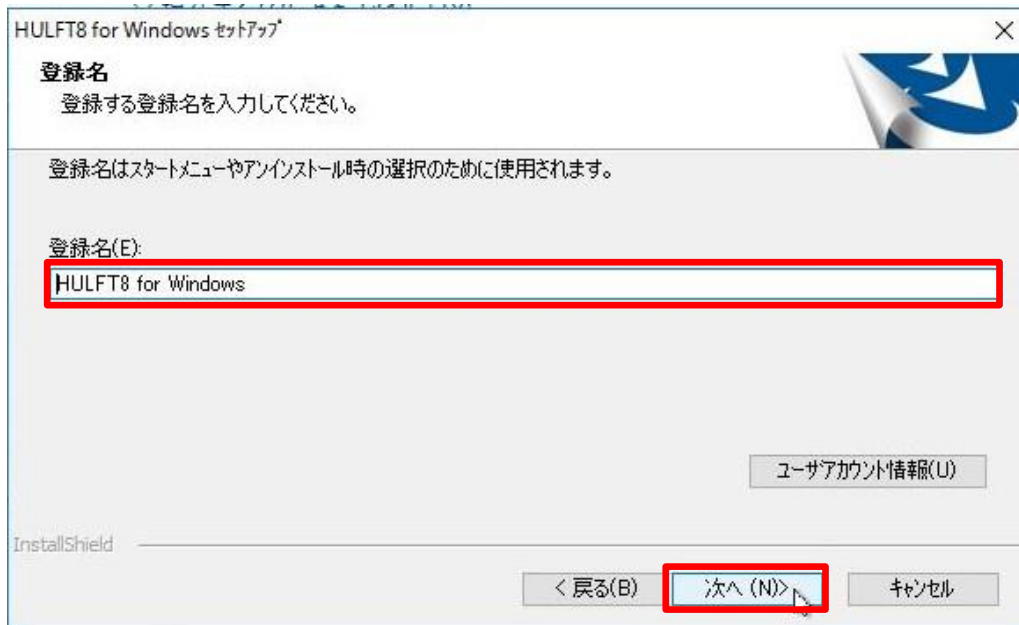
- G) インストール先フォルダを設定します。HULPATH には、HULFT のシステム動作環境設定、配信情報、集信情報等が保管されます。そのため HULFT クラスタを構成する場合、HULPATH は共有ディスク上に配置する必要があります。

今回は、H: が共有ディスクなので、以下のように設定します。

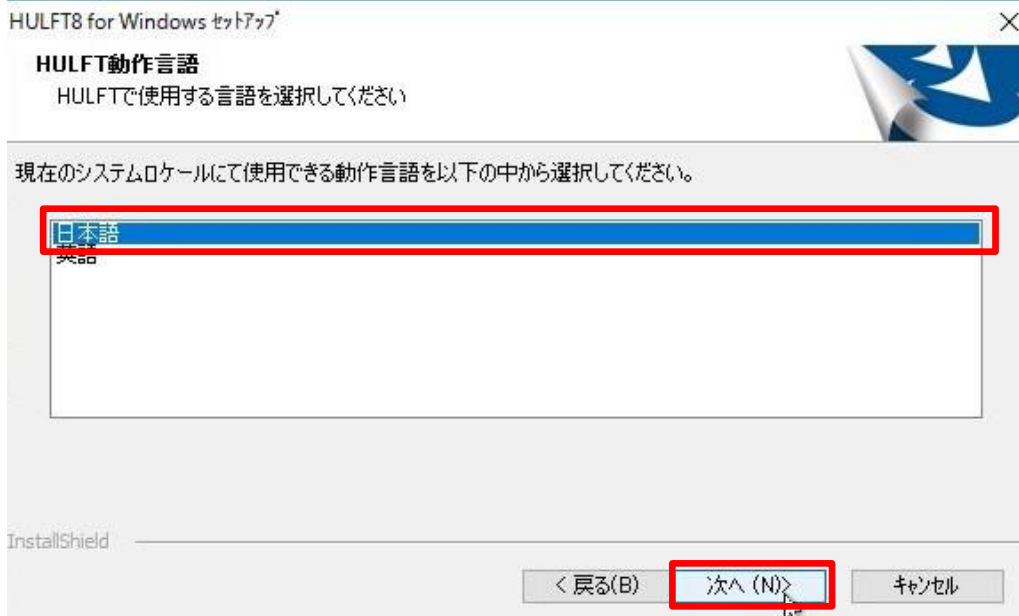
インストール先フォルダ	C:¥HULFT Family¥hulft8¥bin
HULPATH フォルダ	H:¥HULFT Family¥hulft8¥etc



H) 登録名を入力して[次へ]を押下します。今回はデフォルトの登録名のままとします。

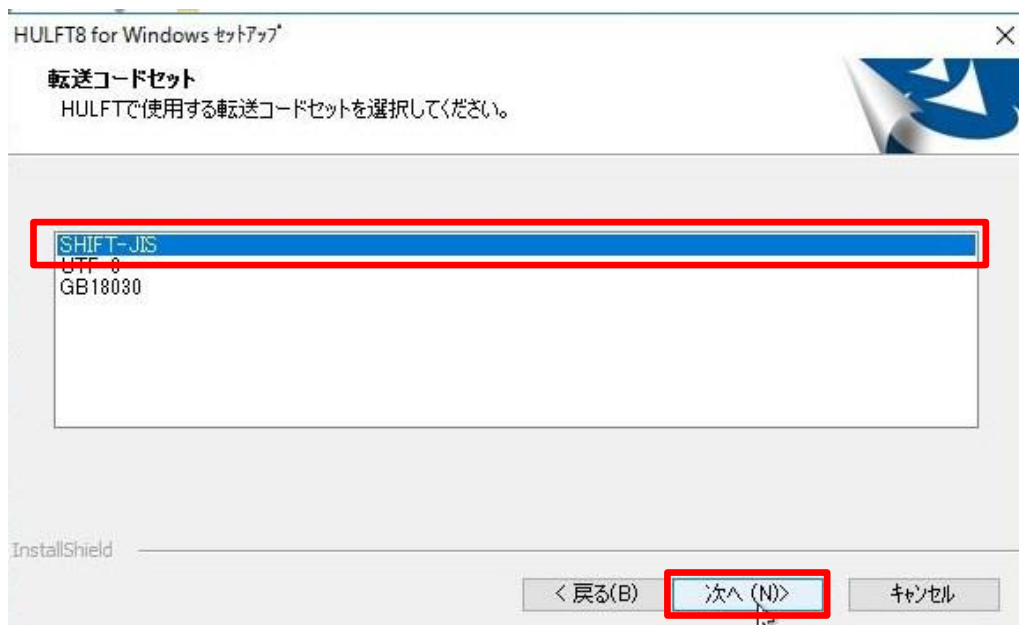


I) HULFT 動作言語を設定します。今回は日本語を選択し[次へ]を押下します。

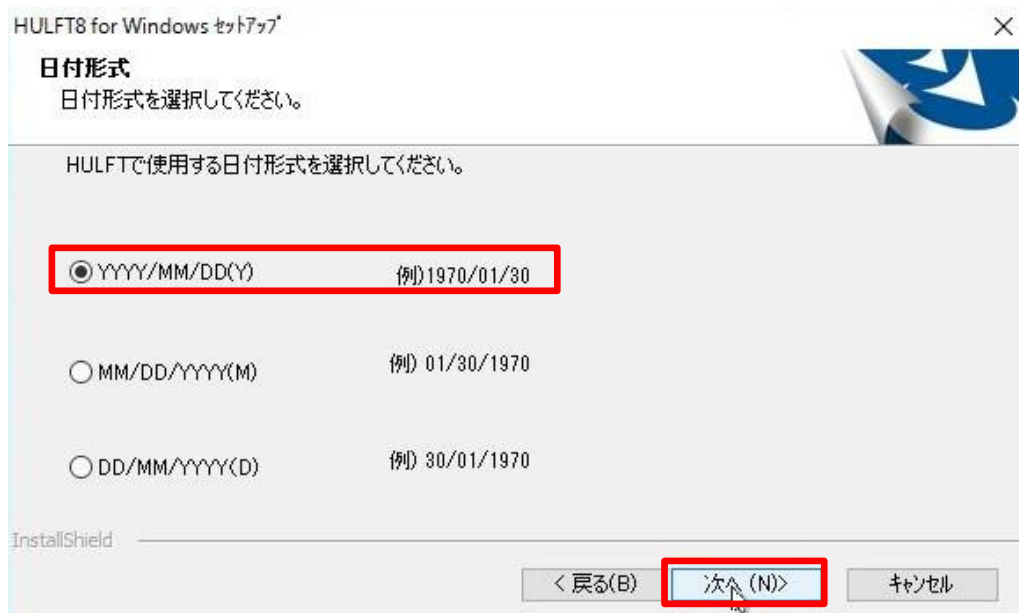




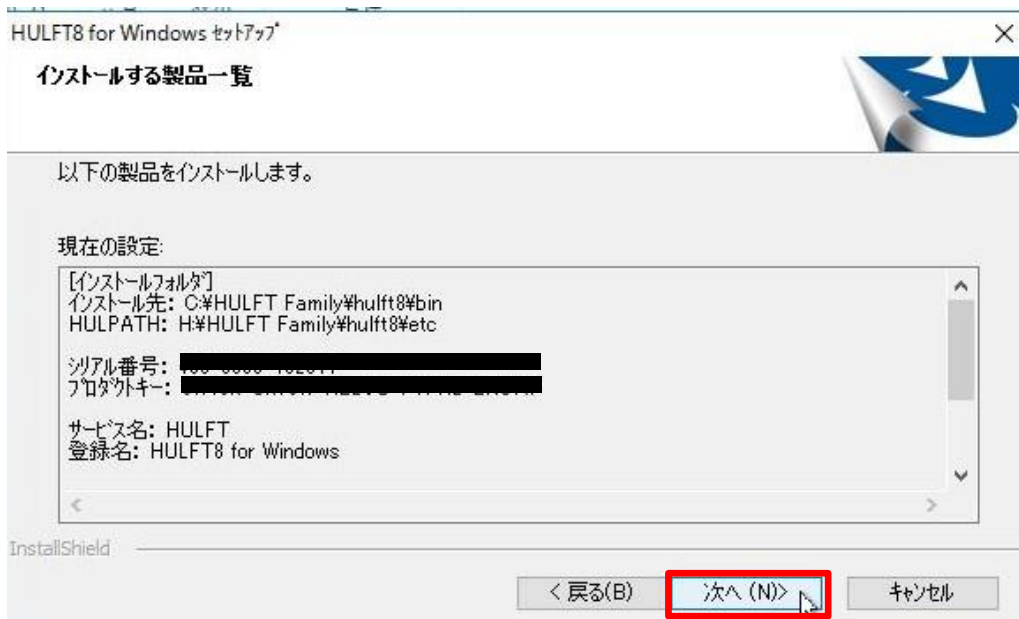
- J) 転送コードセットを選択し[次へ]を押下します。今回は SHIFT-JIS を選択し[次へ]を押下します。



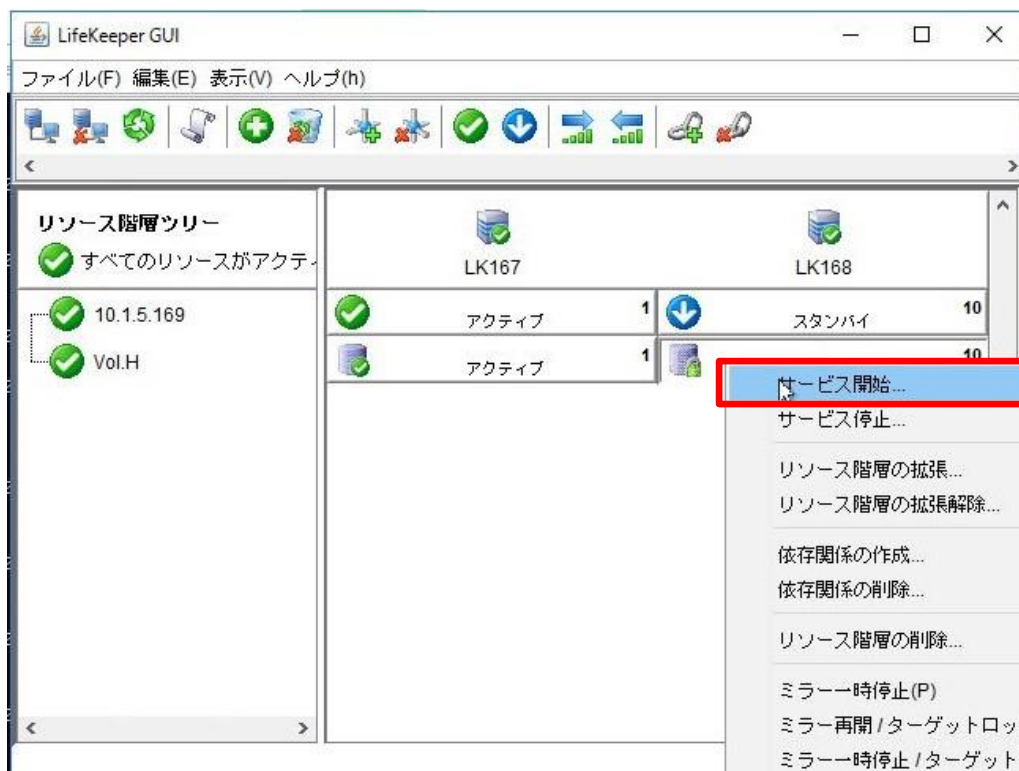
- K) 日付形式を選択し、[次へ]を押下します。



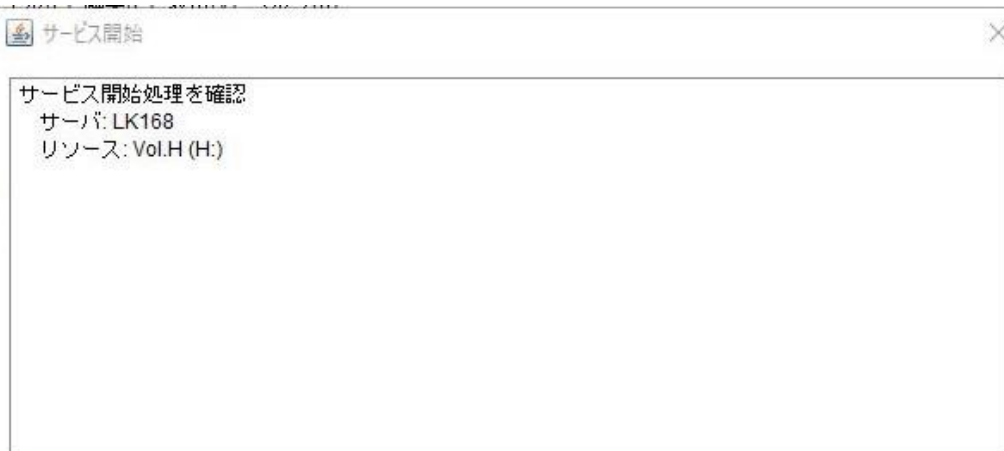
- L) インストールする製品一覧 の画面で内容を確認し、[次へ]を押下します。



- M) インストールが終了したら、ウィザードを閉じます。  
 N) LifeKeeper GUI を起動して、ボリュームリソースをノード 2 側にスイッチオーバーします。ノード 2 側のボリュームリソースを右クリックし、[サービス開始]を選択します。



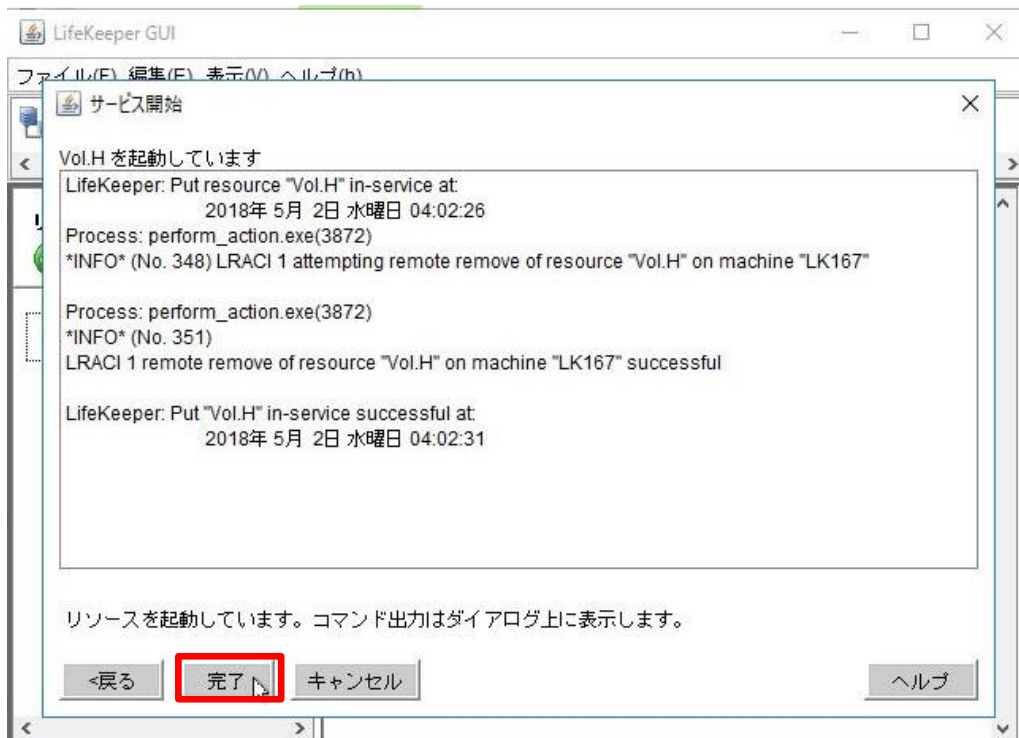
[サービス開始]を押下します。



LK168上で選択したリソースを起動する場合は**サービス開始**をクリックしてください。出力パネルが開いている場合は、出力パネルにコマンド出力を表示します。出力パネルが開いていない場合は、ダイアログ上に表示します。発生する全てのエラーはLK168上のLifeKeeperログおよびGUIログに表示されます。



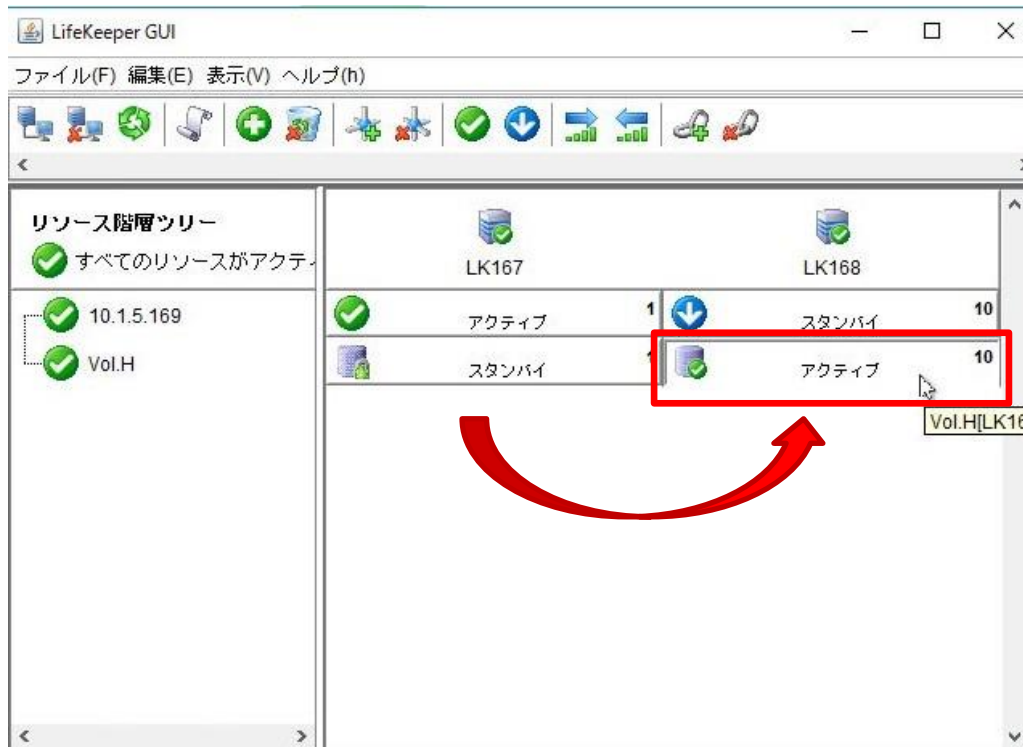
“LifeKeeper Put “[リソース名]” in-service successful at :”と表示されたことを確認して[完了]を押下します。



<-- LK167: Vol.H: 状態を更新しています スタンバイ

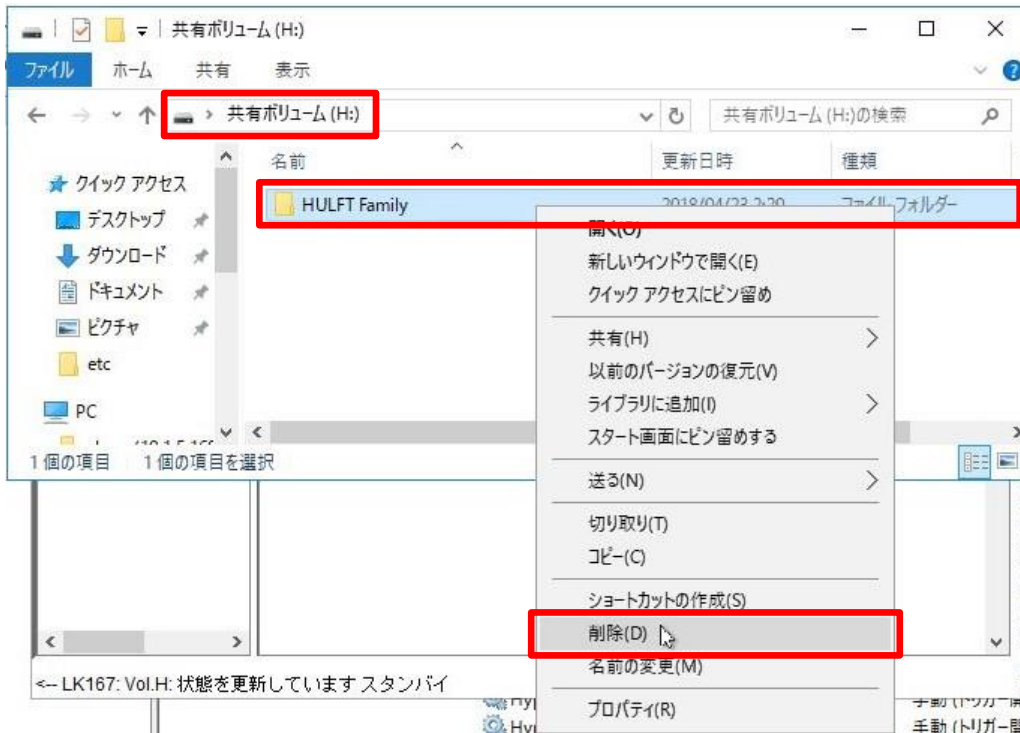
## LifeKeeper for Windows HULFT クラスター構成ガイド

LifeKeeper GUI 上で、ノード 2 側でボリュームリソースがアクティブであることを確認します。



リソース・タグ= Vol.H, リソース・ID= H:

- O) 次にノード 2 上に HULFT をインストールします。ノード 2 で Windows エクスプローラを起動し、共有ディスク (H:)に存在する HULFT Family ディレクトリー以下をすべて削除します。



HULFT Family ディレクトリーが削除されました。

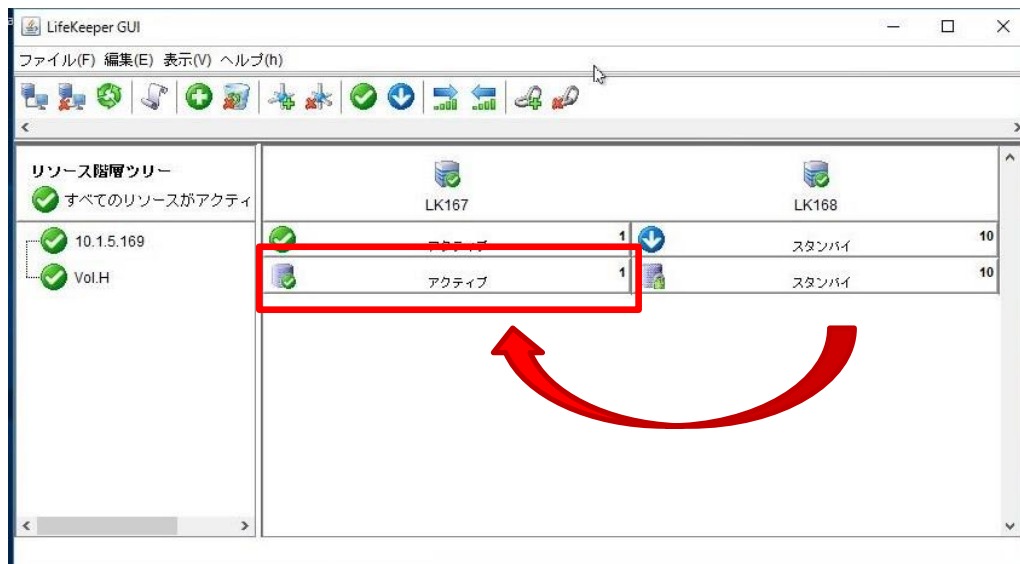


- P) ノード 2 上で、HULFT のインストールメディアから setup を起動します。HULFT インストール時に「必須ソフトウェアがインストールされていません。」といったメッセージが出たしインストールが中断する場合は、インストールメディアの

## LifeKeeper for Windows HULFT クラスタ構成ガイド

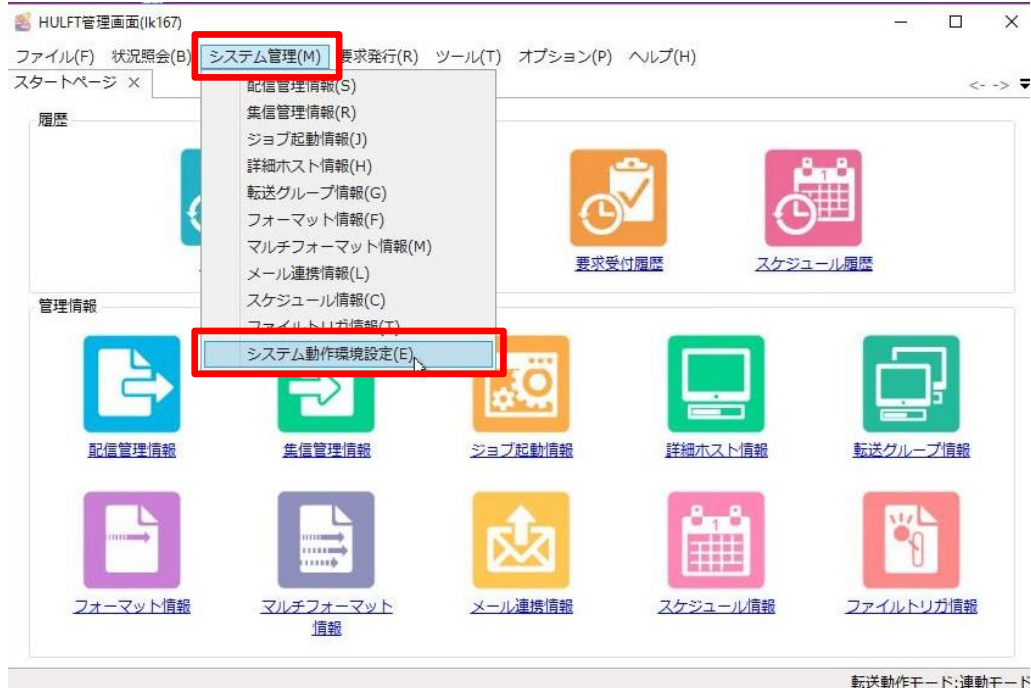
ISSetupPrerequisites フォルダに含まれる Visual C++再頒布可能パッケージのインストールを行ってから、setup を起動します。

- Q) C)~M) と同様の手順で、ノード 2 への HULFT インストールを完了します。インストールが完了したら、N)と同様の手順でボリュームリソースをノード 1 側に戻しておきます。



### 2-12. HULFT システム動作環境の設定(hulenv.cnf)

- A) スタートメニュー > HULFT を選択し、HULFT の管理 GUI を起動します。  
B) 上部メニューの システム管理 > [システム動作環境設定]を選択します。



- C) HULFT がクラスタとして動作するために必要な、システム動作環境設定を行います。  
なおこの設定は、実際には共有ディスク上に配置される HULPATH フォルダ (H:¥HULFT Family¥hulft8¥etc¥)以下にある hulenv.cnf ファイル内に記述されます。  
そのため、両ノードそれぞれで設定を行う必要はありません。

システム動作環境設定、hulenv.cnf、および各パラメータについて詳細は、HULFT のガイドを参照してください。

### HULFT 8 アドミニストレーションマニュアル(Windows)

<HULFT メディア>¥Manual(PDF)¥jpn¥HULFT8\_JP\_ADM\_WIN

### HULFT 8 クラスタ対応マニュアル(Windows)

<HULFT メディア>¥Manual(PDF)¥jpn¥HULFT8\_JP\_CLS\_WIN

今回は、以下のように設定します。

起動	自ホスト名(myhostname)	Lkhulftcluster
	サービスプロセスポートNo.(srvcprocport)	1600
	スケジューラポートNo.(schport)	1700
集配信	未配信ファイルのクリア(delreqcpfile)	しない(0)
クラスタ	管理情報の二重化(adjoinadminfile)	する(1)
	処理状況の出力およびイベントログ(alertmsgput)	出力する(1)
	処理状況格納ファイル件数(consolelogcnt)	10000
	サービス起動同期(startsync)	非同期(1)
通信	コネクションリトライ間隔(retrytime)	60
	自動再配信リトライ回数(sockerr_autoretry)	1

D) 設定を行ったら、設定を有効にするため HULFT サービスを起動します。

上部メニューの ツール > [プロセスコントローラ]を選択します。

HULFT サービス > [起動] を押下します。すでにサービスが起動していた場合は、一度[終了]を押下してから[起動]を押して再起動を行います。





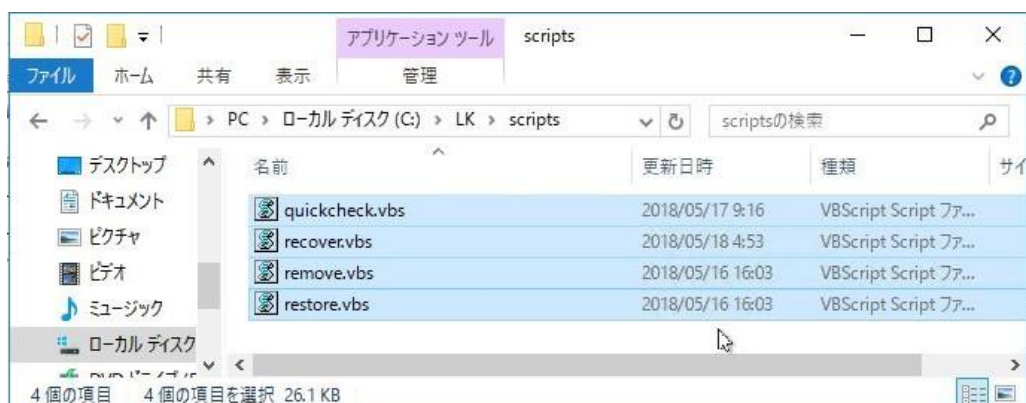
## LifeKeeper for Windows HULFT クラスタ構成ガイド



### 2-13. HULFT リソースの作成

LifeKeeper 上で、クラスターで保護する HULFT リソース（汎用アプリケーションリソース）を作成します。

- A) ノード 1 上の任意のパスに、汎用アプリケーションで使用する各種スクリプト(restore, remove, quickcheck, deepcheck, recover)を配置します。



## LifeKeeper for Windows HULFT クラスタ構成ガイド

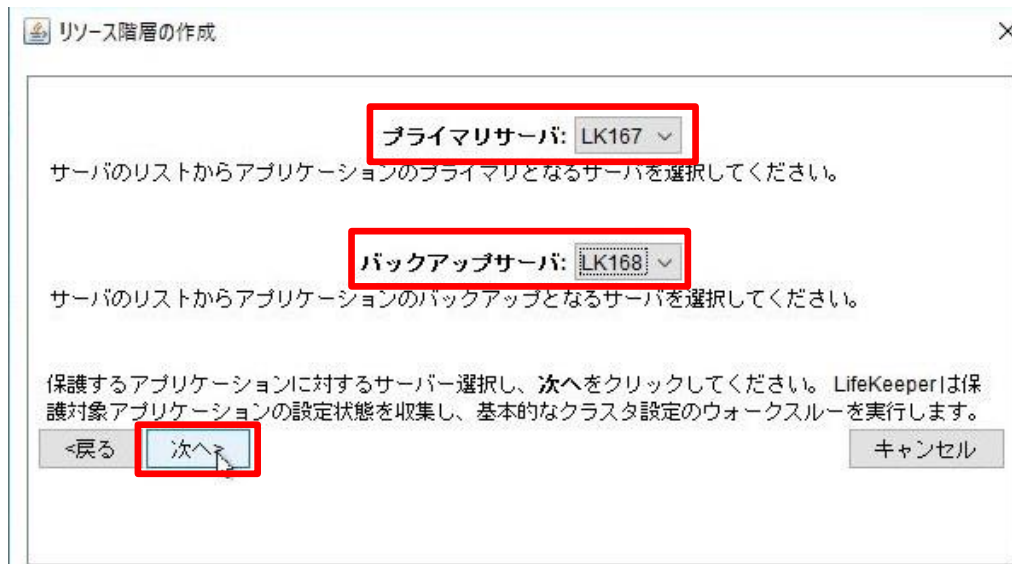
- B) LifeKeeper GUI 画面上部のアイコンをクリックして、リソース階層の作成ウィザードを起動します。



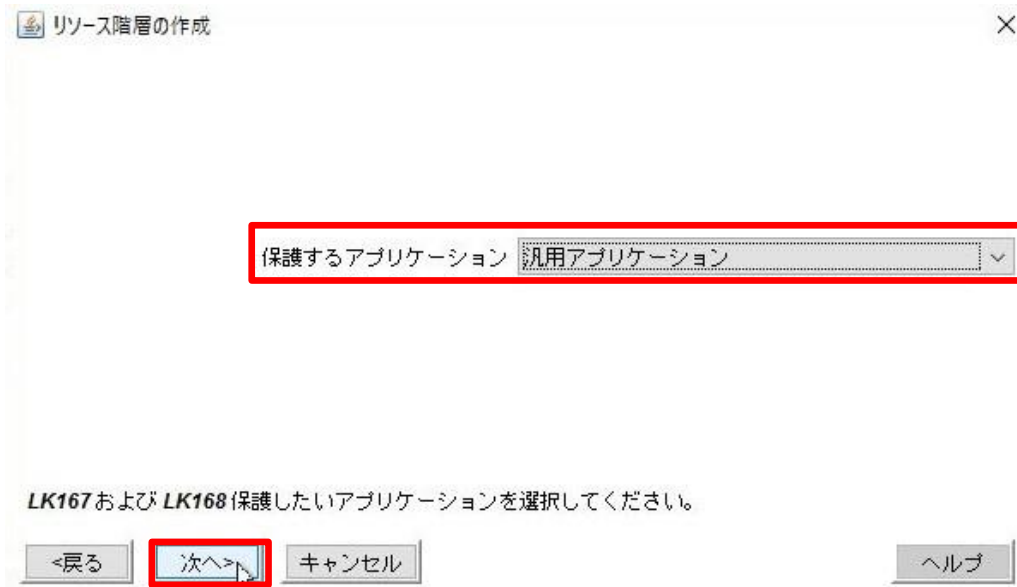
- C) プライマリサーバとバックアップサーバの情報を入力し、[次へ]を押下します。

今回は以下のように入力します。

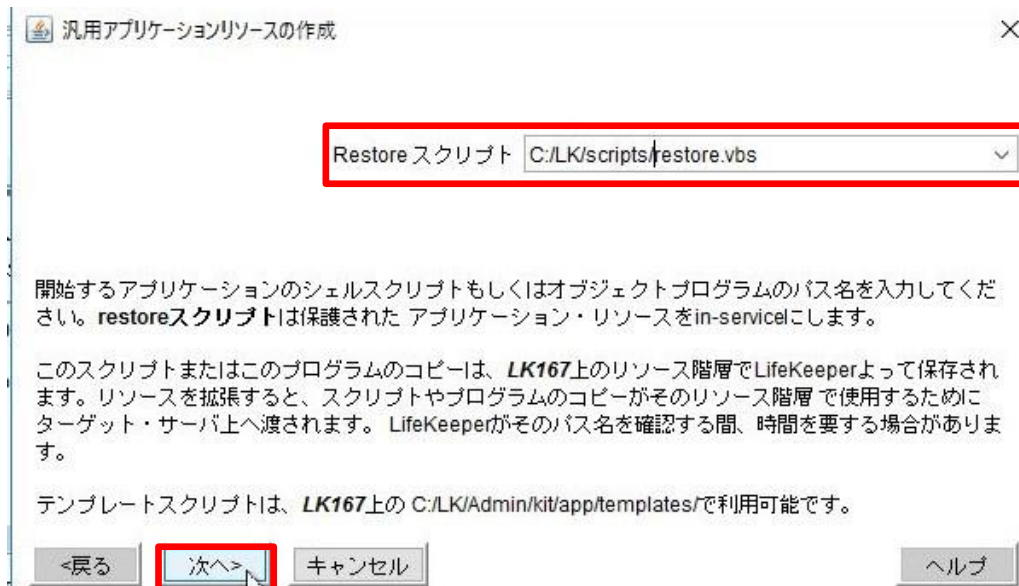
プライマリサーバ	lk167
バックアップサーバ	lk168



- D) 保護するアプリケーションとして[汎用アプリケーション]を選択し、[次へ]を押下します。



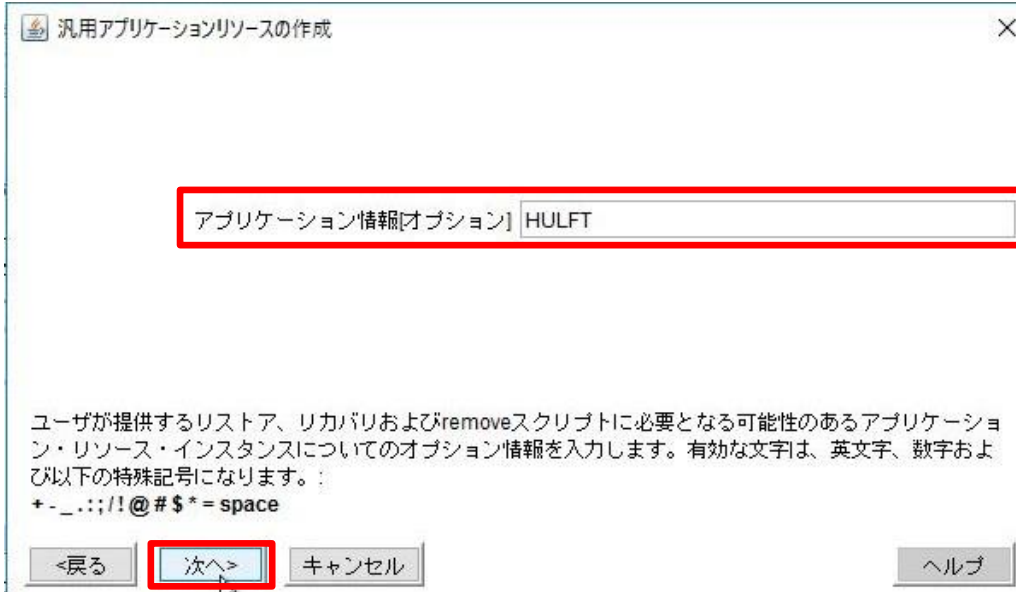
- E) A)で配置した restore スクリプトのパスを指定して、[次へ]を押下します。



- F) 同様に、A)で配置した remove スクリプトのパスを指定して、[次へ]を押下します。
- G) 必要な場合は同様に、A)で配置した quickcheck スクリプトのパスを指定して、[次へ]を押下します。
- H) 必要な場合は同様に、A)で配置した deepcheck スクリプトのパスを指定して、[次へ]

を押下します。

- I) 必要な場合は同様に、A)で配置した recover スクリプトのパスを指定して、[次へ]を押下します。
- J) 保護するサービス名を入力して [次へ]を押下します。ここで登録した名称をもつサービスが、LifeKeeper の保護対象になります。今回は HULFT サービスを保護するので、HULFT と入力します。



汎用アプリケーションリソースの作成

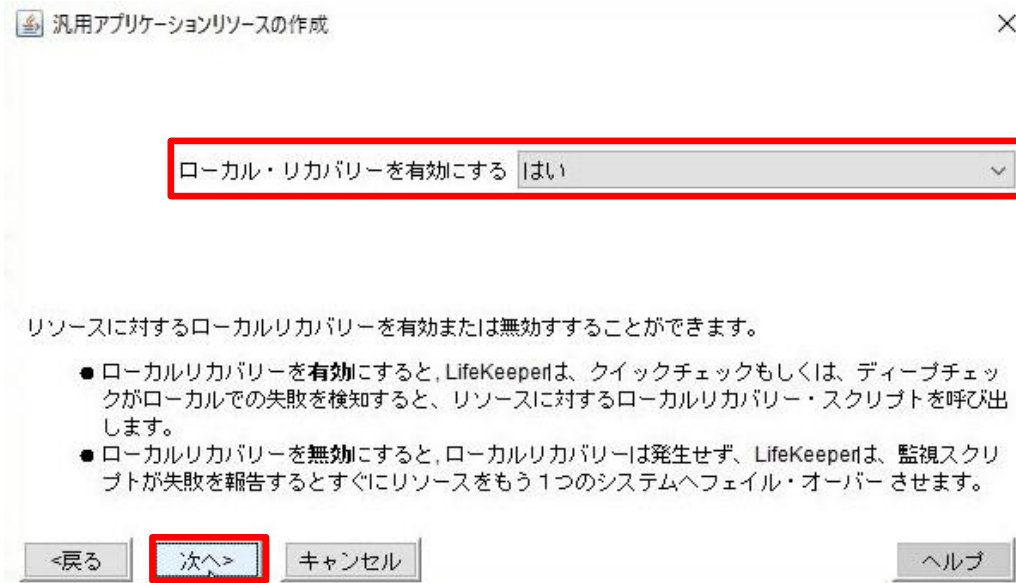
アプリケーション情報[オプション] HULFT

ユーザが提供するリストア、リカバリおよびremoveスクリプトに必要となる可能性のあるアプリケーション・リソース・インスタンスについてのオプション情報を入力します。有効な文字は、英文字、数字および以下の特殊記号になります。:  
+ - \_ . ; ! / @ # \$ \* = space

<戻る 次へ> キャンセル ヘルプ

- K) ローカル・リカバリーの設定を行い、[次へ]を押下します。

ここでは[はい]を選択します。この設定はリソース作成後に変更が可能です。



汎用アプリケーションリソースの作成

ローカル・リカバリーを有効にする はい

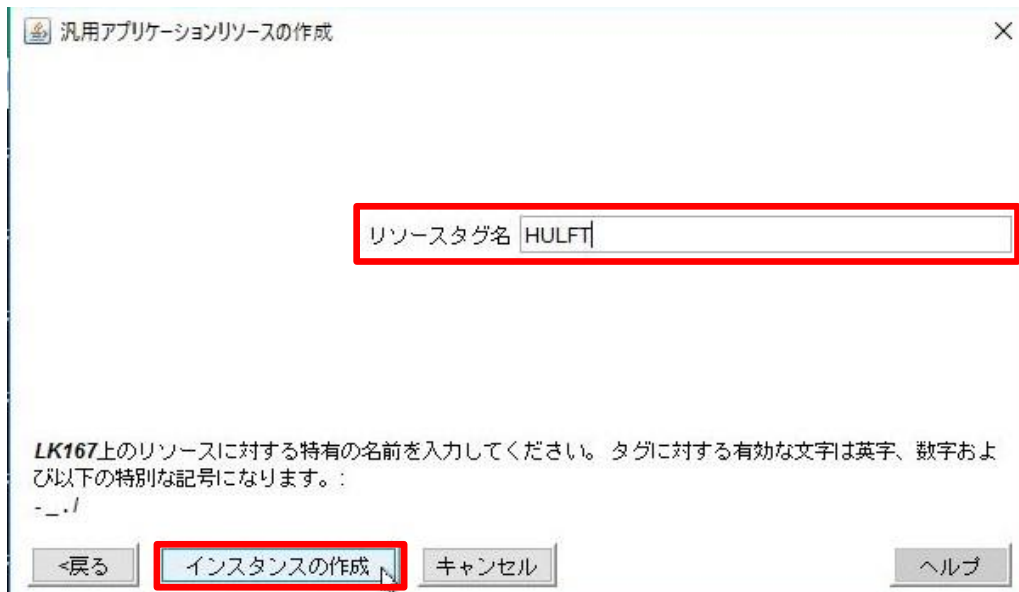
リソースに対するローカルリカバリーを有効または無効することができます。

- ローカルリカバリーを**有効**にすると、LifeKeeperは、クイックチェックもしくは、ディープチェックがローカルでの失敗を検知すると、リソースに対するローカルリカバリー・スクリプトを呼び出します。
- ローカルリカバリーを**無効**にすると、ローカルリカバリーは発生せず、LifeKeeperは、監視スクリプトが失敗を報告するとすぐにリソースをもう1つのシステムフェイル・オーバーさせます。

<戻る 次へ> キャンセル ヘルプ

- L) リソースタグ名を入力し[インスタンスの作成]を押下します。

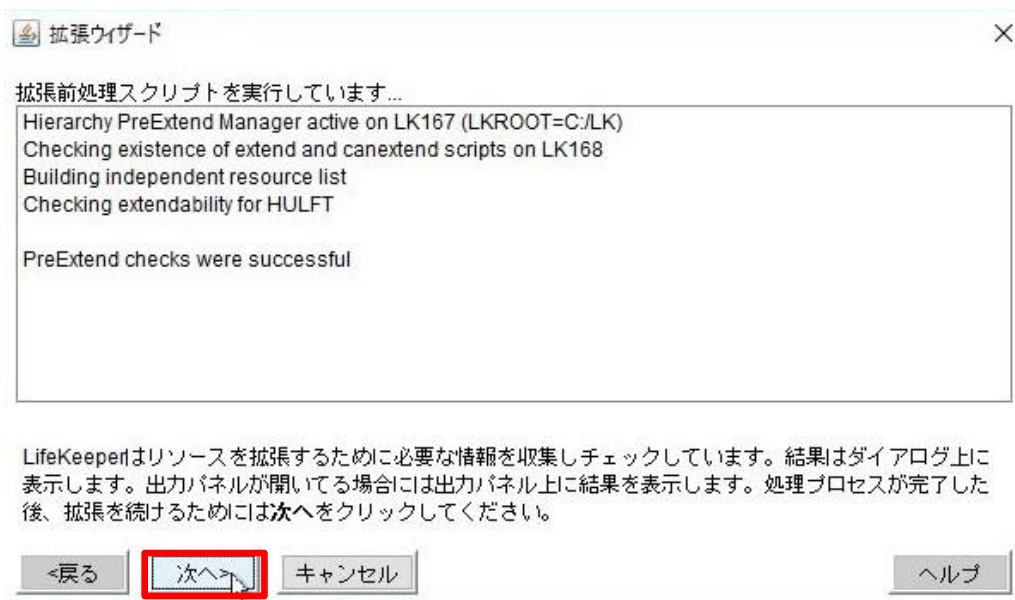
ここでは HULFT と入力します。



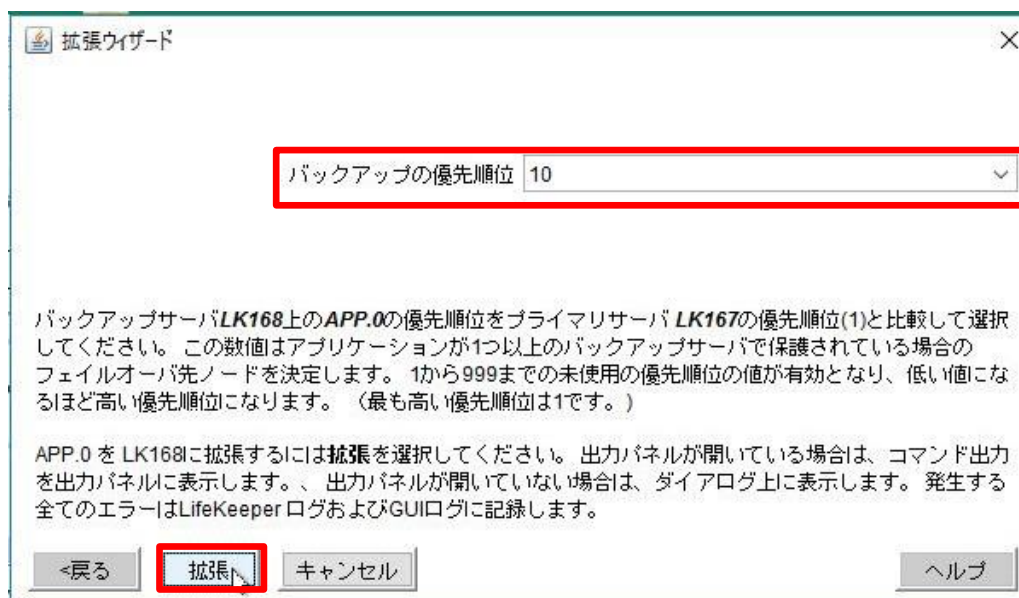
“Resource Instance “[リソース名]” on machine “[サーバー名]” restored”の表示を確認して[次へ]を押下します。



- M) 続けてリソースを拡張します。拡張前処理スクリプトが実行されるので、“PreExtend Checks were successful”の表示を確認して[次へ]を押下します。

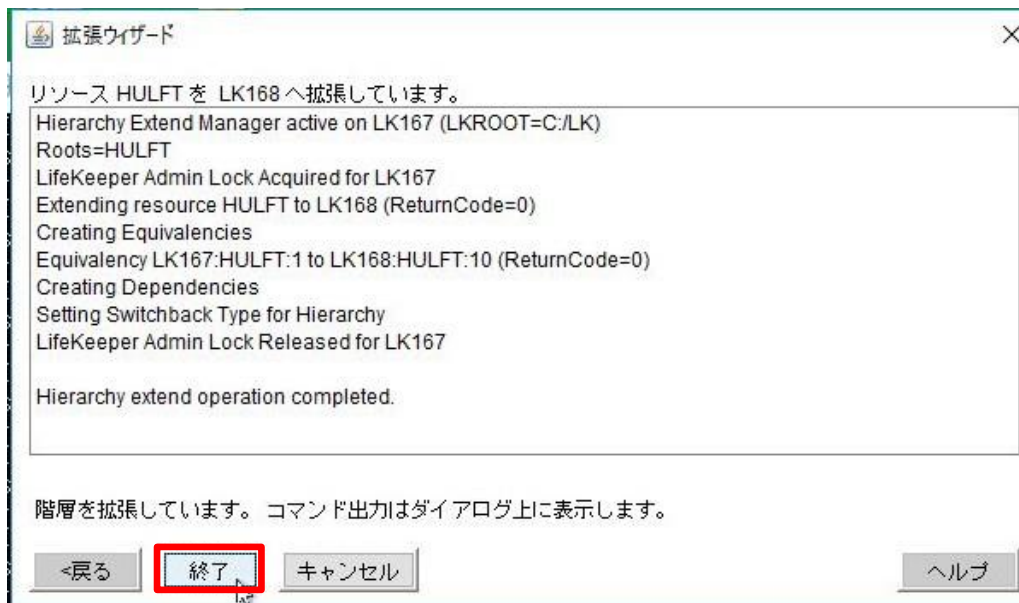


- N) バックアップの優先順位の設定を行い、[拡張]を押下します。ここではデフォルトの[10]に設定します。

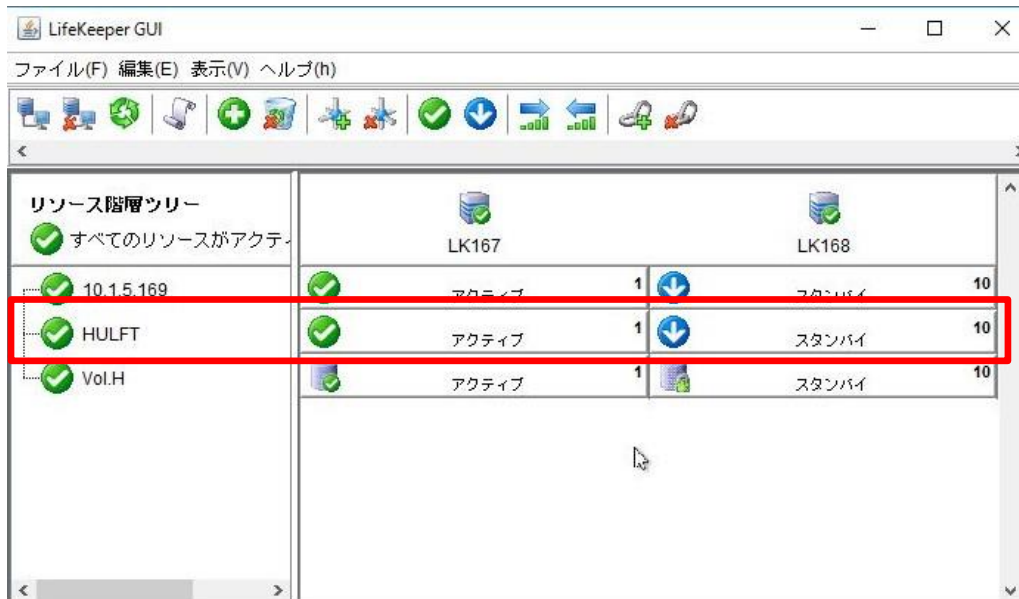


- O) “Hierarchy extend operation completed”と表示されたことを確認して、[終了]を押下します。





P) 汎用アプリケーションリソースが作成されました。



## 2-14. リソース依存関係の作成

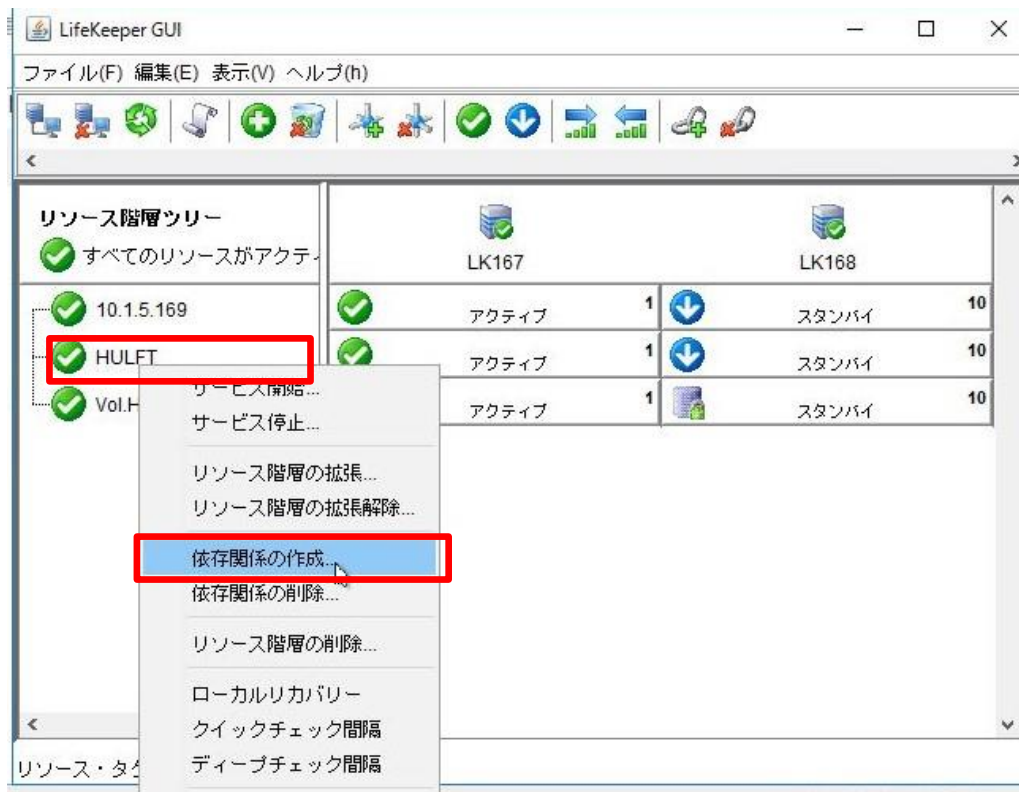
リソース間の依存関係を作成します。今回は以下のようなリソース階層を構成します。

HULFT サービス (汎用アプリケーションリソース)

- └ 仮想 IP アドレス (IP リソース)
  - └ 共有ディスク (ボリュームリソース)

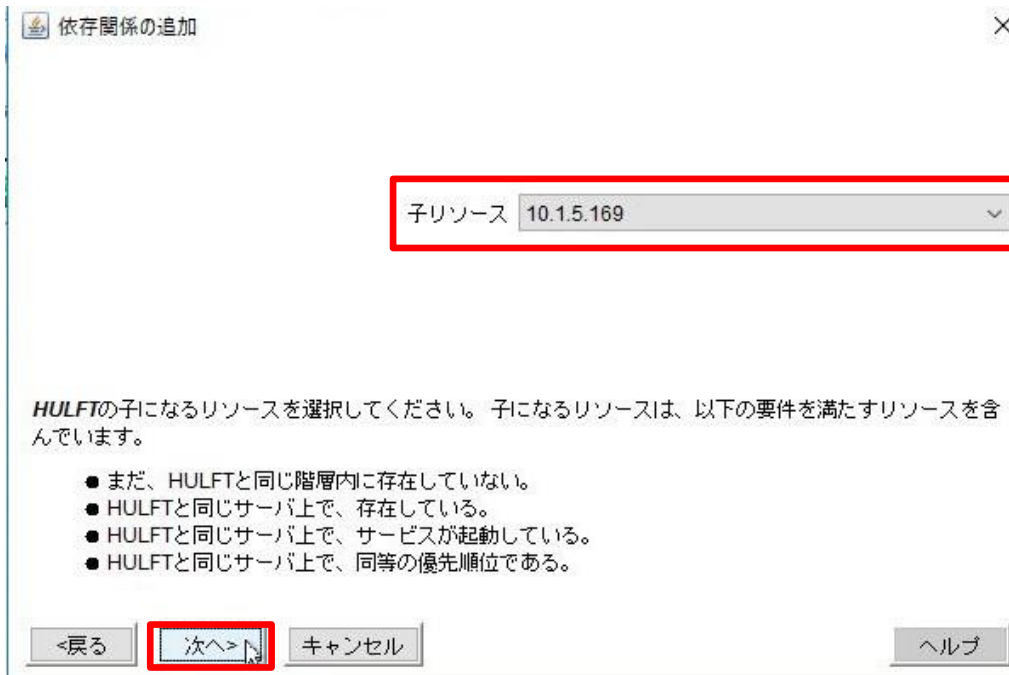
## LifeKeeper for Windows HULFT クラスタ構成ガイド

- A) LifeKeeper GUI のリソース階層ツリーに表示されるリソースを右クリック > [依存関係の作成]を選択して、依存関係の作成ウィザードを起動します。HULFT リソースのリソース依存関係を作成します。

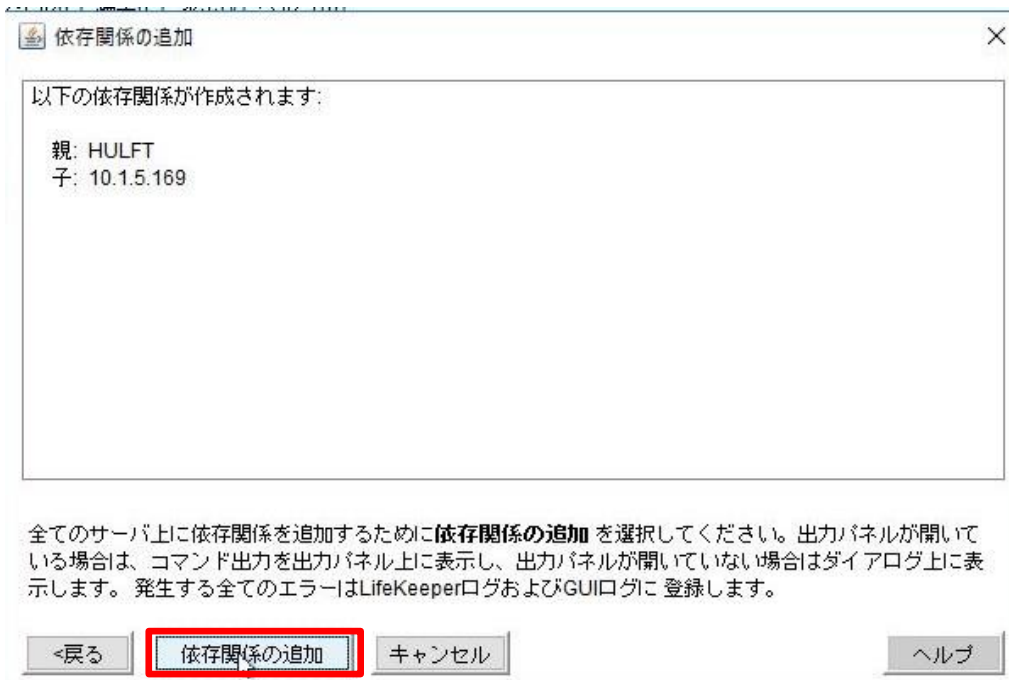


- B) 子リソースを選択して、[次へ]を押下します。ここでは IP リソースを選択します。





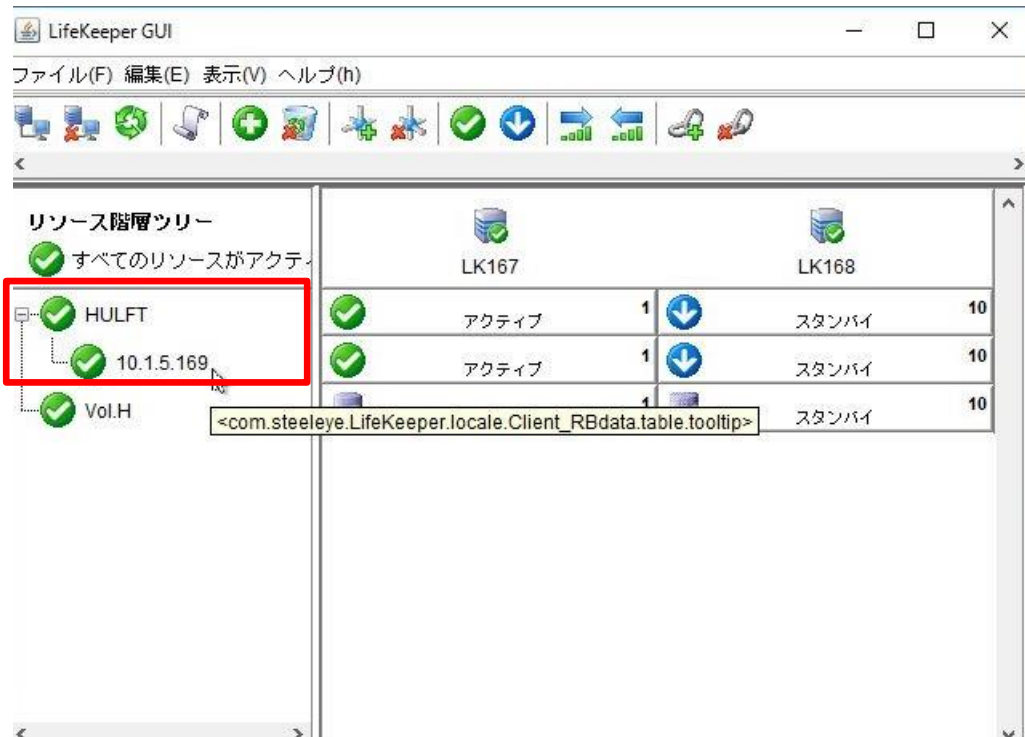
C) 内容を確認して、[依存関係の追加]を押下します。



D) “The dependency creation was successful”の表示を確認して、[完了]を押下します。

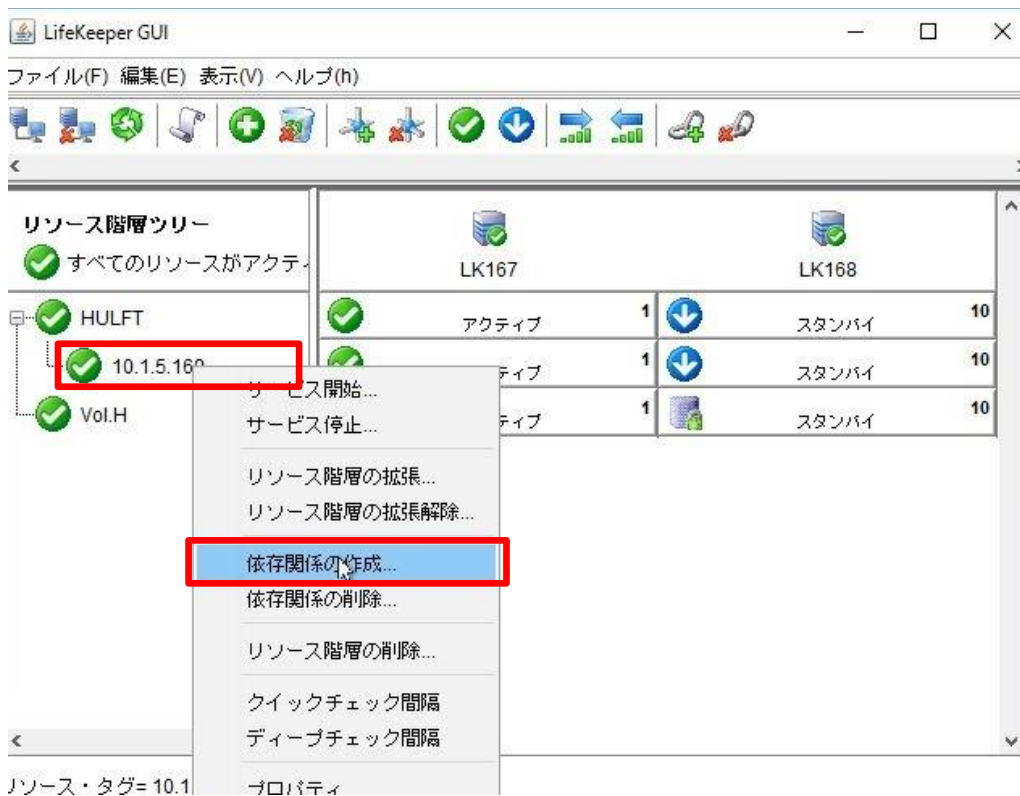


E) リソースの依存関係が作成されました。

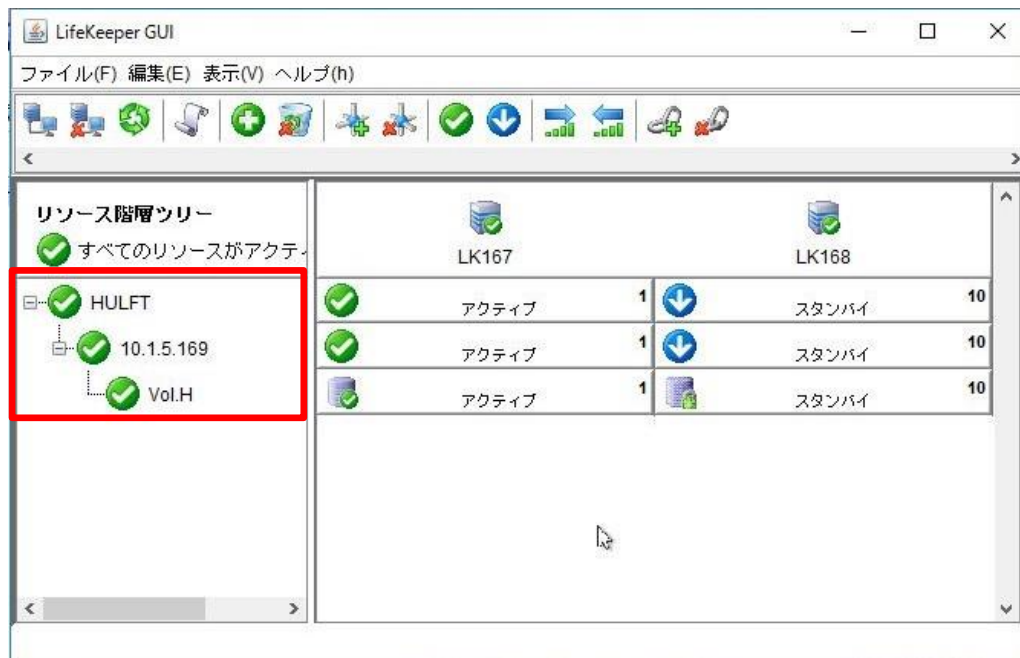


F) A)~E) と同様の手順で、IP リソースとボリュームリソースの依存関係を作成します。

## LifeKeeper for Windows HULFT クラスタ構成ガイド



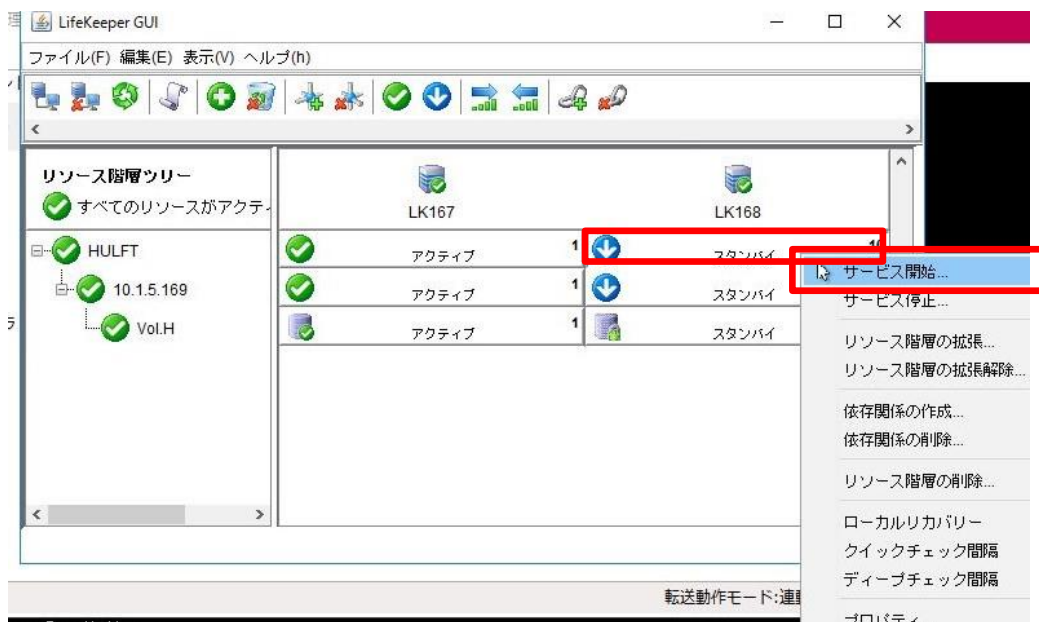
G) リソースの依存関係が作成されました。



## 2-15.HULFT 階層のバックアップサーバへの手動スイッチオーバー

リソースの切替えが実施可能か確認するため、手動でリソースのスイッチオーバーを実施します。

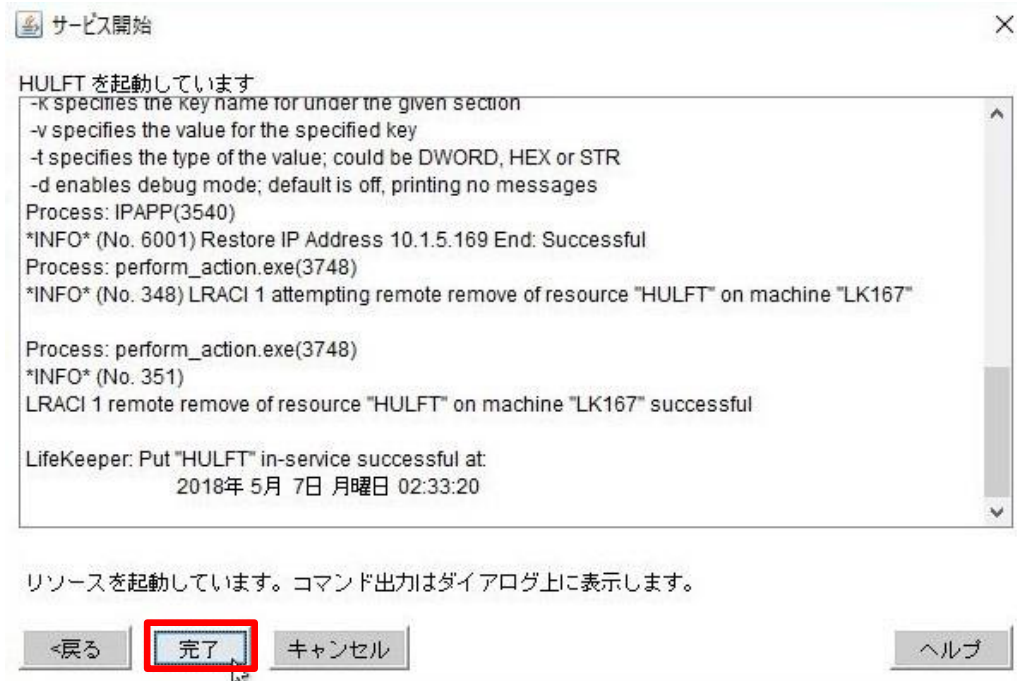
A) バックアップサーバ側の HULFT リソースを右クリックし、[サービス開始]を選択します。



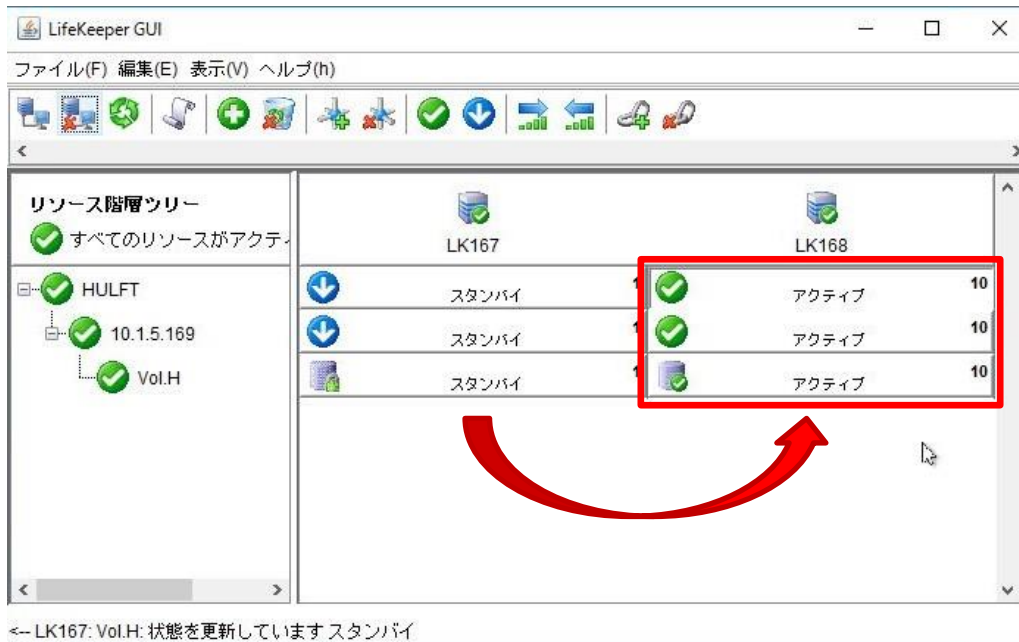
B) 内容を確認し、[サービス開始]を押下します。



C) "Put "[リソース名]" in-service successful at:"の表示を確認して[完了]を押下します。



D) 全てのリソースがバックアップサーバに移動し、“アクティブ”となっていることを確認します。



LifeKeeper 上での設定および確認は以上で終了です。

## 2-16. 詳細ホスト情報、転送グループ情報の設定

HULFT 上で、ファイル転送のためのホスト情報の登録を行います。

A) 稼働系(lk167)で HULFT を起動し、スタートページで[詳細ホスト情報]を押下します。



B) [新規作成]を押下します。



C) 集信ホスト（ファイルを受け取る端末）の情報を入力し、[保存]を押下します。  
ここで入力するホスト名は、DNS または host ファイルにて名前解決ができることが前提です（項番 2-3 参照）。

ここでは、以下のように入力します。

ホスト名	lk115
ホスト種	Windows



## LifeKeeper for Windows HULFT クラスタ構成ガイド

HULFT管理画面(lkhulftcluster)

ファイル(F) 状況照会(B) システム管理(M) 要求発行(R) ツール(T) オプション(P) ヘルプ(H)

スタートページ 詳細ホスト情報一覧 詳細ホスト情報更新(新規作成)\* x

保存 削除 ? ヘルプ

ホスト名 lk115

コメント

基本設定

PROXYサーバ	基本設定
セキュリティ	ホスト種 Windows
	転送コードセット SHIFT-JIS
	集信ポートNo. 30000 要求受付ポートNo. 31000
	日本語規格 83JIS 接続形態 LAN
	ホスト別配信多重度 0
	HULFT7通信モード 無効

D) ホスト情報が保存されました。

HULFT管理画面(lkhulftcluster)

ファイル(F) 状況照会(B) システム管理(M) 要求発行(R) ツール(T) オプション(P) ヘルプ(H)

スタートページ 詳細ホスト情報一覧 x

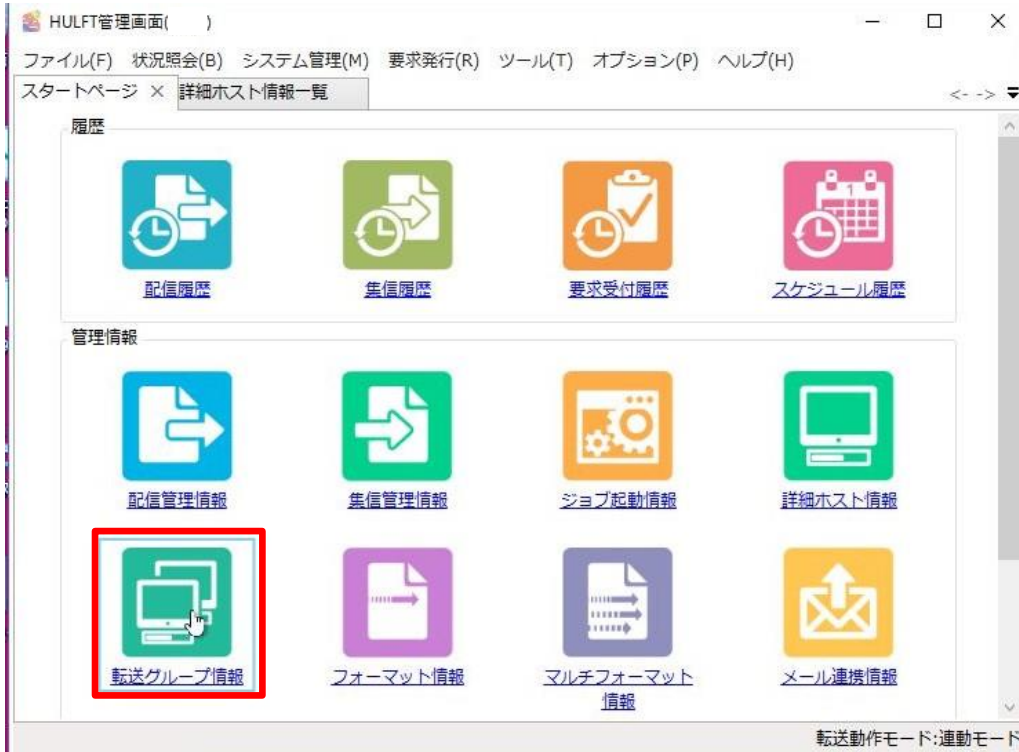
新規作成 編集 コピー 削除 検索 更新 ? ヘルプ

1件 ホスト名 lk115 OK(O)

ホスト名	コメント
lk115	



E) 続けて、転送グループを設定します。スタートページから[転送グループ情報]を押下します。



F) [新規作成]を押下します。

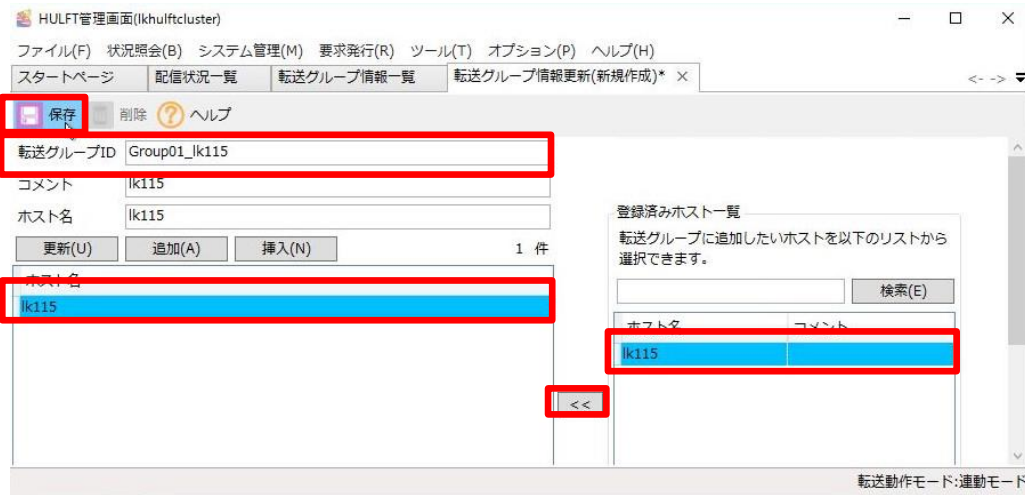


G) 任意の転送グループ ID を入力し、登録済みホスト一覧からグループに含むホストを選択して、[保存] > [OK]の順に押下します。

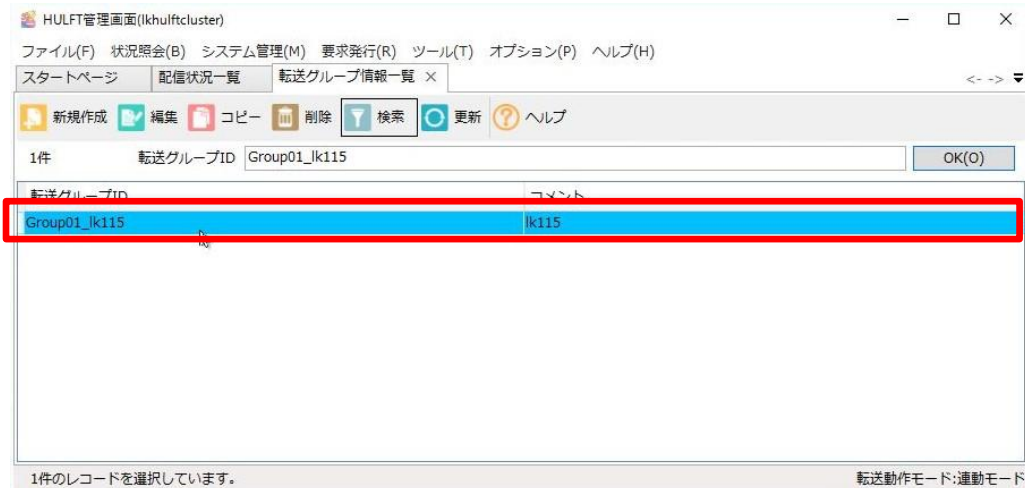
今回は、以下のように設定します。

転送グループ ID	Group01_ik115
ホスト名	ik115

## LifeKeeper for Windows HULFT クラスタ構成ガイド



H) 転送グループ ID が登録されました。



I) 集信ホスト上に配信ホストの情報を登録します。集信ホスト（今回は ik115）にログイン後 HULFT を起動します。A)～D)と同様の手順で、クラスタのホスト情報を登録します。ここでは、以下のように入力します。

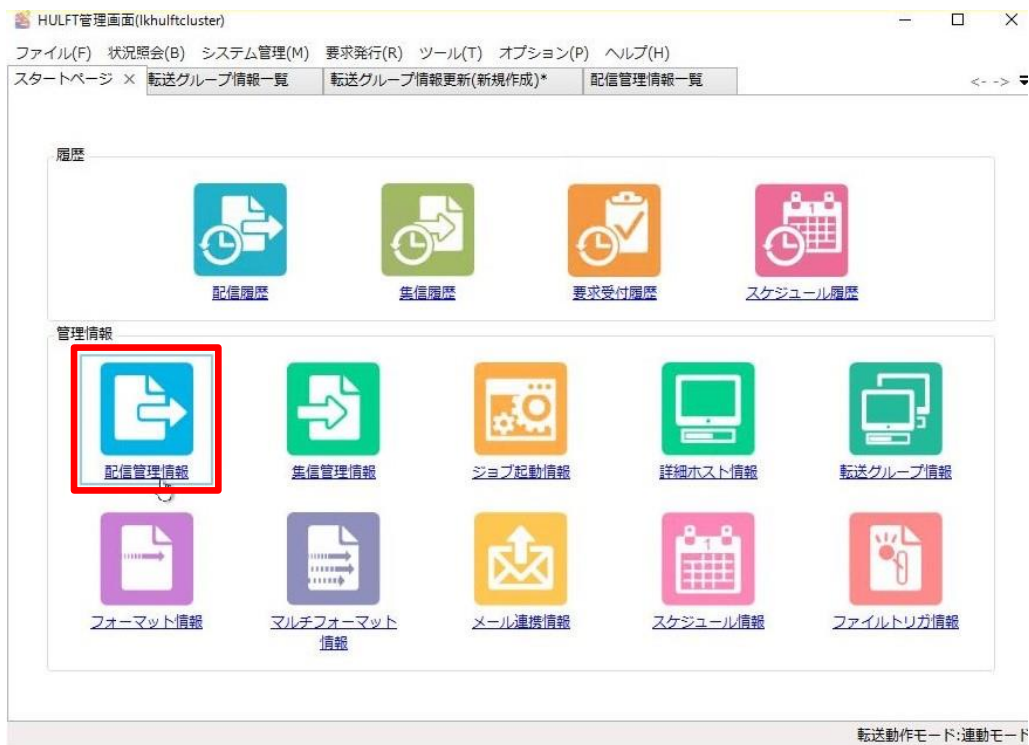
ホスト名	lkhulftcluster
ホスト種	Windows

## 2-17. 配信管理情報、集信管理情報の設定

配信対象のファイルの情報を、配信ホスト、集信ホストにそれぞれ登録します。

A) まず、配信ホストにファイル情報を登録します。

稼働系(lk167)の HULFT のスタートページから [配信管理情報]を押下します。



B) [新規作成]を押下します。



C) 配信するファイルを登録し、[保存] > [OK]の順に押下します。

今回は、以下のように登録します。

ファイル ID	FILE0001_TESTFILE
ファイル名	H:¥SendData¥testfile.txt
転送タイプ	TEXT
転送グループ ID	Group01_lk115

## LifeKeeper for Windows HULFT クラスタ構成ガイド



D) 配信ホストにファイル情報が登録されました。



E) 次に、集信ホストにファイル情報を登録します。

集信ホスト(Ik115)の HULFT のスタートページから [集信情報管理]を押下します。



F) [新規作成]を押下します。



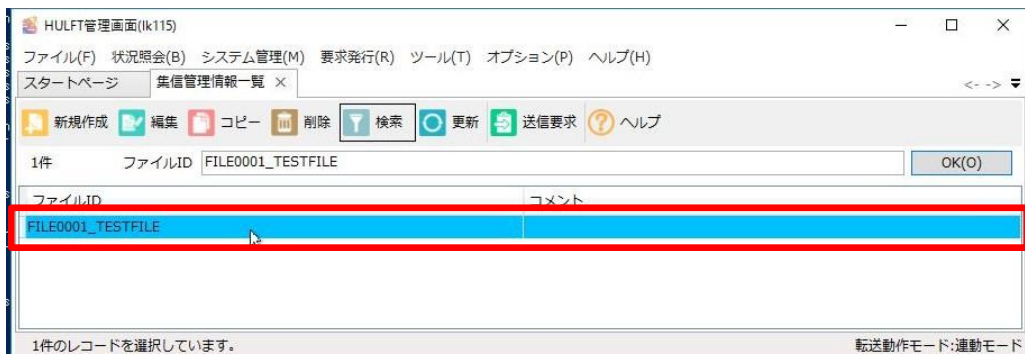
G) 配信ホストに登録したものと同一ファイル ID のファイル情報を登録します。今回は以下のように登録します。

ファイル ID	FILE0001_TESTFILE
ファイル名	C:\¥Users¥administrator.LKG¥Desktop¥ReceivedData¥testfile
登録モード	新規作成

## LifeKeeper for Windows HULFT クラスタ構成ガイド



H) 集信ホストにファイル情報が登録されました。



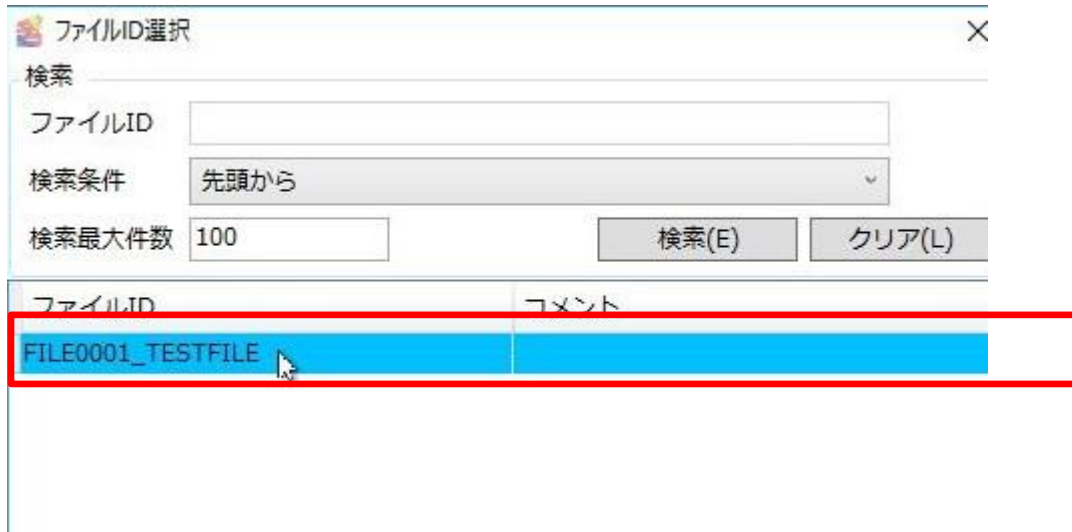
## 2-18. ファイル配信

A) 最後に、実際にファイルの配信を行います。稼働系(ik167)の HULFT スタートページを開き、上部メニューの [要求発行 > [配信要求] > [配信要求]を選択します。





B) 配信するファイルのファイルID を選択し、[OK]を押下します。



C) ファイル ID を確認して、[配信要求]を押下します。



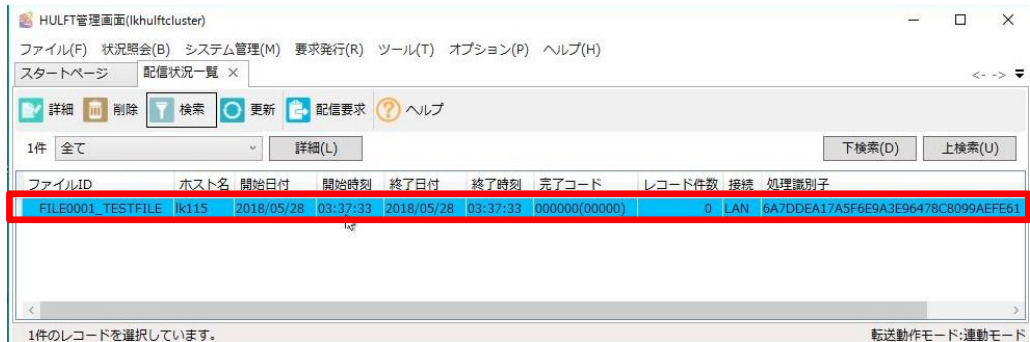
D) 確認画面が表示されたら[OK]を押下します。続けて[閉じる]を押下します。



E) HULFT のスタートページから [配信履歴]を押下します。



F) 完了コードが 000000(00000) となっていることから、配信が成功で完了したことを確認します。また、集信ホスト側での確認に使用するため、処理識別子の ID を控えておきます。





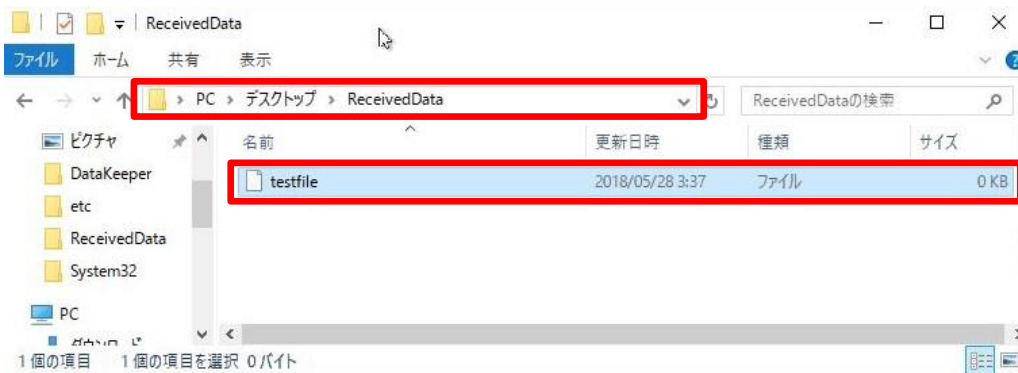
G) 集信ホスト(Ik115)の HULFT スタートページから [集信履歴]を押下します。



H) 表示される履歴の処理識別子の ID が、配信ホスト側で控えた処理識別子の ID と同じであることを確認します。完了コードが 000000(00000) となっていることから、集信が成功で完了したことを確認します。



- I) G) で[集信管理情報]に登録したパス上に、配信されたファイルが存在することを確認します。



設定および確認は以上ですべて完了です。

### 【参考情報】

本テストでは、以下の4つのスクリプト (restore.vbs, remove.vbs, quickcheck.vbs, recover.vbs) を使用して動作確認を行いました。

本項に掲載するスクリプト情報はあくまで参考情報であり、実際の動作を保証するものではありません。

### restore.vbs

```
###restore script for HULFT###
Option Explicit

' Declare Program Variables
Dim WshShell          ' Wsh Root Object
Dim ts                ' Text Stream
Dim oFS               ' File System Object
Dim oTempFile         ' Temporary File Object
Dim objArgs           ' Input Arguments Object
Dim numErrCode        ' Error Code
Dim numRetCode        ' Return Code
Dim strCmd            ' Script Name
Dim strErrMsg         ' Error Message Description
Dim strID             ' LK Resource ID Dim
Dim strLKAppName     ' LK Resource Application Name
Dim strLKCmd          ' LK Command
Dim strLKCmdStr       ' LK Command String
Dim strLKCmdLine     ' LK Command Line
Dim strLKROOT        ' LK Home Directory
Dim strLKResType     ' LK Resource Type
Dim strStdOut         ' Standard Output
Dim strTag            ' LK Resource Tag name
Dim strTempFilename  ' Temporary File Name
Dim strWindir        ' Windows Home Directory

' Initialize Constants & Variables
Const ForReading = 1
Const TemporaryFolder = 2
Const TristateUseDefault = -2

strLKAppName = "app"
strLKResType = "gen"
strCmd = Wscript.ScriptName ' Pick up Script Name
Set WshShell = CreateObject("WScript.Shell")
Set oFS = CreateObject("Scripting.FileSystemObject")

' Validate Input Parameters
set objArgs = WScript.Arguments
select Case objArgs.Count
    Case 4
        If objArgs(0) = "-t" Then
            strTag = objArgs(1)
        Else
            Usage
        End If

        If objArgs(2) = "-i" Then
            strID = objArgs(3)
```

## LifeKeeper for Windows HULFT クラスタ構成ガイド

```
Else
    Usage
End If
Case Else
    strErrMsg = "An incorrect number of input parameters were provided to the " &strCmd & " script."
    WshShell.LogEvent 1, "LifeKeeper Generic Application Recovery Kit scripting error." &vbCrLf
&strErrMsg
    Usage
End Select

Function Usage
    WScript.Echo "USAGE:"
    WScript.Echo "cscript //nologo " &strCmd & " -t tagname -i ID"
    WScript.Echo "tagname    - Server unique LifeKeeper tag name for protected resource"
    WScript.Echo "ID          - Cluster unique LifeKeeper ID for the protected resource"
    WScript.Echo "Windows Scripting Host (cscript.exe) must be installed on the server"
    WScript.Quit(1)
End Function

Function LKFailure(strErrMsg)
    ' Use this function to report a problem with the LifeKeeper system
    ' This function also references numErrCode, strCMD, strTag, strID
    WshShell.LogEvent 1, "LifeKeeper Generic Application Recovery Kit failure." &vbCrLf &"TagName=" &strTag &"`,
ID=" &strID &"`, Error No. " &numErrCode'&vbCrLf &strErrMsg
    WScript.Quit(1)
End Function

Function ResFailure(strErrMsg)
    ' Use this function to report a problem with the protected resource
    ' This function also references numErrCode, strCMD, strTag, strID, numRetCode
    RunLKCcmd("lk_err -c FRS_ERR -n " &numErrCode &" -d TO_STDERR -p " &strCMD &" LifeKeeper: TAG="
&strTag &"`, ID=" &strID &"`, Return Code=" &numRetCode &"`; " &strErrMsg)
    WScript.Quit(1)
End Function

Function RunLKCcmd(strLKCcmd)
    strStdOut=""
    strTempFilename = oFS.GetSpecialFolder(TemporaryFolder) & "¥" & oFS.GetTempName
    While oFS.FileExists(strTempFilename)
        strTempFilename = oFS.GetSpecialFolder(TemporaryFolder) & "¥" & oFS.GetTempName
    Wend
    ' Setup and run the command
    strLKCcmdStr = strLKROOT &"¥bin¥" &strLKCcmd
    strLKCcmdLine = "%comspec% /c " & strLKCcmdStr &" > " & strTempFilename
    numRetCode = WshShell.Run(strLKCcmdLine, 7, True)
    Set oTempFile = oFS.GetFile(strTempFilename)
    Set ts = oTempFile.OpenAsTextStream(ForReading, TristateUseDefault)
    If not ts.AtEndOfStream Then
        strStdOut = ts.ReadAll ' Get complete output
    End If
    ts.Close
    oTempFile.Delete
End Function

' Validate LKROOT Environment Variable
strLKROOT = WshShell.ExpandEnvironmentStrings("%LKROOT%")

' Standard setup is done, now perform the desired task
' Perform commands that restore the resource to the In-Service state
' -----
' Custom code goes here
' Set final value of numRetCode; for example
numRetCode=99
```

## LifeKeeper for Windows HULFT クラスター構成ガイド

```
'-----  
' numRetCode is used to identify successful or failure  
  
'=====
```

'(メイン処理)変数宣言

```
'=====
```

Dim objShell,ret,counter,scquery,scqueryretry,scqueryret,scqueryretrysleep,netstart,usererrcode,usererrmsg

```
'=====
```

'(メイン処理)ユーザーパラメーター

```
'=====
```

'net start コマンド

```
netstart = "net start " & Chr(34) & "HULFT" & Chr(34)
```

'sc query のリトライ回数

```
scqueryretry = 2
```

'sc query のリトライ待ち(ミリ秒)

```
scqueryretrysleep = 3000
```

'sc query コマンド

```
scquery = "sc query HULFT | findstr STATE | findstr RUNNING > nul"
```

'エラーコード

```
usererrcode = 27451
```

'エラーメッセージ

```
usererrmsg = "HULFT Service restore failed"
```

```
'=====
```

'(メイン処理)初期化

```
'=====
```

scqueryret = 99

```
'=====
```

' メイン処理

```
'=====
```

'シェルオブジェクト作成

```
Set objShell = CreateObject("WScript.Shell")
```

'net start & sc query 実行

```
counter = 0  
Do  
    If counter > scqueryretry Then  
        numErrCode = usererrcode  
        strErrMsg = usererrmsg  
        numRetCode = 1  
        WScript.Echo "*ERROR* (No." & numErrCode & ")" & strErrMsg  
        'ResFailure(strErrMsg)  
        Wscript.Quit(numRetCode)  
    ElseIf counter > 0 Then  
        WScript.Sleep scqueryretrysleep  
        WScript.Echo "リトライ実施"  
    End If  
    Set ret = objShell.Exec("cmd.exe /c " & netstart)  
    Do While ret.Status = 0  
        WScript.Sleep 100  
    Loop  
    Set ret = objShell.Exec("cmd.exe /c " & scquery)  
    Do While ret.Status = 0  
        WScript.Sleep 100  
    Loop
```

## LifeKeeper for Windows HULFT クラスター構成ガイド

```
        scqueryret = ret.exitcode
        counter = counter + 1
    Loop Until scqueryret = 0

'シェルオブジェクト破棄
Set objShell = Nothing
'Set WshShell = Nothing
'Set oFS = Nothing

'スクリプト終了
' End of Restore script
Wscript.Quit (0)
```

### remove.vbs

```
###remove script for HULFT###
Option Explicit

' Declare Program Variables
Dim WshShell          ' Wsh Root Object
Dim ts                ' Text Stream
Dim oFS               ' File System Object
Dim oTempFile         ' Temporary File Object
Dim objArgs           ' Input Arguments Object
Dim numErrCode        ' Error Code
Dim numRetCode        ' Return Code
Dim strCmd             ' Script Name
Dim strErrMsg         ' Error Message Description
Dim strID              ' LK Resource ID Dim
Dim strLKAppName      ' LK Resource Application Name
Dim strLKCmd           ' LK Command
Dim strLKCmdStr        ' LK Command String
Dim strLKCmdLine      ' LK Command Line
Dim strLKROOT         ' LK Home Directory
Dim strLKResType      ' LK Resource Type
Dim strStdOut         ' Standard Output
Dim strTag             ' LK Resource Tag name
Dim strTempFilename   ' Temporary File Name
Dim strWindir         ' Windows Home Directory

' Initialize Constants & Variables
Const ForReading = 1
Const TemporaryFolder = 2
Const TristateUseDefault = -2

strLKAppName = "app"
strLKResType = "gen"
strCmd = Wscript.ScriptName ' Pick up Script Name
Set WshShell = CreateObject("WScript.Shell")
Set oFS = CreateObject("Scripting.FileSystemObject")

' Validate Input Parameters
set objArgs = WScript.Arguments
select Case objArgs.Count
    Case 4
        If objArgs(0) = "-t" Then
            strTag = objArgs(1)
        Else
            Usage
        End If
```

## LifeKeeper for Windows HULFT クラスタ構成ガイド

```

                If objArgs(2) = "-i" Then
                    strID = objArgs(3)
                Else
                    Usage
                End If
            Case Else
                strErrMsg = "An incorrect number of input parameters were provided to the " &strCmd & " script."
                WshShell.LogEvent 1, "LifeKeeper Generic Application Recovery Kit scripting error." &vbCrLf
&strErrMsg
                Usage
            End Select

Function Usage
    WScript.Echo "USAGE:"
    WScript.Echo "cscript //nologo " &strCmd & " -t tagname -i ID"
    WScript.Echo "tagname - Server unique LifeKeeper tag name for protected resource"
    WScript.Echo "ID - Cluster unique LifeKeeper ID for the protected resource"
    WScript.Echo "Windows Scripting Host (cscript.exe) must be installed on the server"
    WScript.Quit(1)
End Function

Function LKFailure(strErrMsg)
    ' Use this function to report a problem with the LifeKeeper system
    ' This function also references numErrCode, strCMD, strTag, strID
    WshShell.LogEvent 1, "LifeKeeper Generic Application Recovery Kit failure." &vbCrLf &"TagName=" &strTag &"`,
ID=" &strID &"`, Error No. " &numErrCode &vbCrLf &strErrMsg
    WScript.Quit(1)
End Function

Function ResFailure(strErrMsg)
    ' Use this function to report a problem with the protected resource
    ' This function also references numErrCode, strCMD, strTag, strID, numRetCode
    RunLKCcmd("lk_err -c FRS_ERR -n " &numErrCode &" -d TO_STDERR -p " &strCMD &" LifeKeeper: TAG=" &
&strTag &"`, ID=" &strID &"`, Return Code=" &numRetCode &"`; " &strErrMsg)
    WScript.Quit(1)
End Function

Function RunLKCcmd(strLKCcmd)
    strStdOut=""
    strTempFilename = oFS.GetSpecialFolder(TemporaryFolder) &"¥" &oFS.GetTempName
    While oFS.FileExists(strTempFilename)
        strTempFilename = oFS.GetSpecialFolder(TemporaryFolder) &"¥" &oFS.GetTempName
    Wend
    ' Setup and run the command
    strLKCcmdStr = strLKROOT &"¥bin¥" &strLKCcmd
    strLKCcmdLine = "%comspec% /c " &strLKCcmdStr &" >" &strTempFilename
    numRetCode = WshShell.Run(strLKCcmdLine, 7, True)
    Set oTempFile = oFS.GetFile(strTempFilename)
    Set ts = oTempFile.OpenAsTextStream(ForReading, TristateUseDefault)
    If not ts.AtEndOfStream Then
        strStdOut = ts.ReadAll ' Get complete output
    End If
    ts.Close
    oTempFile.Delete
End Function

' Validate LKROOT Environment Variable
strLKROOT = WshShell.ExpandEnvironmentStrings("%LKROOT%")

' Standard setup is done, now perform the desired task
' Perform commands that remove the resource from the In-Service state
' to the Out-Of-Service state.
```

## LifeKeeper for Windows HULFT クラスター構成ガイド

```
'-----  
' Custom code goes here  
' Set final value of numRetCode; for example  
numRetCode=99  
'-----  
  
' numRetCode is used to identify successful or failure  
  
'=====
```

(メイン処理)変数宣言

```
'=====
```

Dim objShell,ret,counter,scquery,scqueryretry,scqueryret,scqueryretrysleep,netstop,usererrcode,usererrmsg

```
'=====
```

(メイン処理)ユーザーパラメーター

```
'=====
```

'net stop コマンド  
netstop = "C:\HULFTF~1\hulft8\bin\utlsvctl.exe -c stop -f"

'sc query のリトライ回数  
scqueryretry = 2

'sc query のリトライ待ち(ミリ秒)  
scqueryretrysleep = 3000

'sc query コマンド  
scquery = "sc query HULFT | findstr STATE | findstr STOPPED > nul"

'エラーコード  
usererrcode = 27852

'エラーメッセージ  
usererrmsg = "HULFT Service remove failed"

```
'=====
```

(メイン処理)初期化

```
'=====
```

scqueryret = 99

```
'=====
```

メイン処理

```
'=====
```

'シェルオブジェクト作成  
Set objShell = CreateObject("WScript.Shell")

'net stop & sc query 実行  
counter = 0  
Do  
    If counter > scqueryretry Then  
        numErrCode = usererrcode  
        strErrMsg = usererrmsg  
        numRetCode = 1  
        WScript.Echo "\*ERROR\* (No." & numErrCode & ") " & strErrMsg  
        'ResFailure(strErrMsg)  
        Wscript.Quit(numRetCode)  
    Elseif counter > 0 Then  
        WScript.Sleep scqueryretrysleep  
        WScript.Echo "リトライ実施"  
    End If  
    Set ret = objShell.Exec("cmd.exe /c " & netstop)  
    Do While ret.Status = 0  
        WScript.Sleep 100  
    Loop



## LifeKeeper for Windows HULFT クラスタ構成ガイド

```
Set ret = objShell.Exec("cmd.exe /c " & scquery)
Do While ret.Status = 0
    WScript.Sleep 100
Loop
scqueryret = ret.exitcode
counter = counter + 1
Loop Until scqueryret = 0
```

```
'シェルオブジェクト破棄
Set objShell = Nothing
'Set WshShell = Nothing
'Set oFS = Nothing
```

```
'スクリプト終了
' End of Remove script
Wscript.Quit (0)
```

### quickcheck.vbs

```
###quickcheck script for HULFT###
Option Explicit

' Declare Program Variables
Dim WshShell      ' Wsh Root Object
Dim ts            ' Text Stream
Dim oFS           ' File System Object
Dim oTempFile    ' Temporary File Object
Dim objArgs      ' Input Arguments Object
Dim numErrCode   ' Error Code
Dim numRetCode   ' Return Code
Dim strCmd       ' Script Name
Dim strErrMsg    ' Error Message Description
Dim strID        ' LK Resource ID Dim
Dim strLKAppName ' LK Resource Application Name
Dim strLKCmd     ' LK Command
Dim strLKCmdStr  ' LK Command String
Dim strLKCmdLine ' LK Command Line
Dim strLKROOT    ' LK Home Directory
Dim strLKResType ' LK Resource Type
Dim strStdOut    ' Standard Output
Dim strTag       ' LK Resource Tag name
Dim strTempFilename ' Temporary File Name
Dim strWindir   ' Windows Home Directory

' Initialize Constants & Variables
Const ForReading = 1
Const TemporaryFolder = 2
Const TristateUseDefault = -2

strLKAppName = "app"
strLKResType = "gen"
strCmd = Wscript.ScriptName ' Pick up Script Name
Set WshShell = CreateObject("WScript.Shell")
Set oFS = CreateObject("Scripting.FileSystemObject")

' Validate Input Parameters
set objArgs = WScript.Arguments
select Case objArgs.Count
    Case 4
```

## LifeKeeper for Windows HULFT クラスタ構成ガイド

```
        If objArgs(0) = "-t" Then
            strTag = objArgs(1)
        Else
            Usage
        End If

        If objArgs(2) = "-i" Then
            strID = objArgs(3)
        Else
            Usage
        End If
    Case Else
        strErrMsg = "An incorrect number of input parameters were provided to the " &strCmd &" script."
        WshShell.LogEvent 1, "LifeKeeper Generic Application Recovery Kit scripting error." &vbCrLf
&strErrMsg
        Usage
    End Select

Function Usage
    WScript.Echo "USAGE:"
    WScript.Echo "cscript //nologo " &strCmd &" -t tagname -i ID"
    WScript.Echo "tagname - Server unique LifeKeeper tag name for protected resource"
    WScript.Echo "ID - Cluster unique LifeKeeper ID for the protected resource"
    WScript.Echo "Windows Scripting Host (cscript.exe) must be installed on the server"
    WScript.Quit(1)
End Function

Function LKFailure(strErrMsg)
    ' Use this function to report a problem with the LifeKeeper system
    ' This function also references numErrCode, strCMD, strTag, strID
    WshShell.LogEvent 1, "LifeKeeper Generic Application Recovery Kit failure." &vbCrLf &"TagName=" &strTag &"`,
ID=" &strID &"`, Error No. " &numErrCode &vbCrLf &strErrMsg
    WScript.Quit(1)
End Function

Function ResFailure(strErrMsg)
    ' Use this function to report a problem with the protected resource
    ' This function also references numErrCode, strCMD, strTag, strID, numRetCode
    RunLKCcmd("lk_err -c FRS_ERR -n " &numErrCode &" -d TO_STDERR -p " &strCMD &" LifeKeeper: TAG="
&strTag &"`, ID=" &strID &"`, Return Code=" &numRetCode &"`; " &strErrMsg)
    WScript.Quit(1)
End Function

Function RunLKCcmd(strLKCcmd)
    strStdOut=""
    strTempFilename = oFS.GetSpecialFolder(TemporaryFolder) &"¥" &oFS.GetTempName
    While oFS.FileExists(strTempFilename)
        strTempFilename = oFS.GetSpecialFolder(TemporaryFolder) &"¥" &oFS.GetTempName
    Wend
    ' Setup and run the command
    strLKCcmdStr = strLKROOT &"¥bin¥" &strLKCcmd
    strLKCcmdLine = "%comspec% /c " &strLKCcmdStr &" > " &strTempFilename
    numRetCode = WshShell.Run(strLKCcmdLine, 7, True)
    Set oTempFile = oFS.GetFile(strTempFilename)
    Set ts = oTempFile.OpenAsTextStream(ForReading, TristateUseDefault)
    If not ts.AtEndOfStream Then
        strStdOut = ts.ReadAll ' Get complete output
    End If
    ts.Close
    oTempFile.Delete
End Function

' Validate LKROOT Environment Variable
```

## LifeKeeper for Windows HULFT クラスター構成ガイド

```
strLKROOT = WshShell.ExpandEnvironmentStrings("%LKROOT%")

' Standard setup is done, now perform the desired task
' Perform commands that accurately assess the state of the protected application
'
-----
' Custom code goes here
' Set final value of numRetCode; for example
numRetCode=99
'
-----

' numRetCode is used to identify successful or failure

'=====
' (メイン処理)変数宣言
'=====
Dim
objShell,colenv,tmp,ret,checkcmd,envname,checkpath,checkargumentarray,checkret,counter,checkretry,scquery,scqueryretry,sc
queryret,scqueryretrysleep,checkretrysleep,usererrcode,usererrmsg

'=====
' (メイン処理)ユーザーパラメーター
'=====
'utlalivecheck のリトライ回数
checkretry = 1

'utlalivecheck のリトライ待ち(ミリ秒)
checkretrysleep = 0

'utlalivecheck の実行パス
checkpath = "C:\HULFTF\1\hulft8\bin\utlalivecheck"

'utlalivecheck のオプション
Set checkargumentarray = CreateObject("System.Collections.ArrayList")
checkargumentarray.add " -p 32000 -w 10"
checkargumentarray.add " -p 30000 -w 10"
checkargumentarray.add " -p 31000 -w 10"
checkargumentarray.add " -p 1700 -w 10"

'sc query のリトライ回数
scqueryretry = 2

'sc query のリトライ待ち(ミリ秒)
scqueryretrysleep = 5000

'sc query コマンド
scquery = "sc query HULFT | findstr STATE | findstr RUNNING > nul"

'エラーコード
usererrcode = 27653

'エラーメッセージ
usererrmsg = "HULFT Service quickcheck error"

'=====
' (メイン処理)初期化
'=====
checkret = 99
scqueryret = 99

'=====
' メイン処理
'=====
'シェルオブジェクト作成
```

## LifeKeeper for Windows HULFT クラスタ構成ガイド

```
Set objShell = CreateObject("WScript.Shell")

'環境変数取得
checkcmd = Chr(34) & checkpath & Chr(34)

'sc query 実行
counter = 0
Do
    If counter > scqueryretry Then
        numErrCode = usererrcode
        strErrMsg = usererrmsg
        numRetCode = 1
        WScript.Echo "*ERROR* (No." & numErrCode & ") " & strErrMsg
        'ResFailure(strErrMsg)
        Wscript.Quit(numRetCode)
    ElseIf counter > 0 Then
        WScript.Sleep scqueryretrysleep
        WScript.Echo "リトライ実施"
    End If
    Set ret = objShell.Exec("cmd.exe /c " & scquery)
    Do While ret.Status = 0
        WScript.Sleep 100
    Loop
    scqueryret = ret.exitcode
    counter = counter + 1
Loop Until scqueryret = 0

'utlalivecheck 実行
Dim i
For i=0 To checkargumentarray.Count - 1
    counter = 0
    Do
        If counter > checkretry Then
            numErrCode = usererrcode
            strErrMsg = usererrmsg
            numRetCode = 1
            WScript.Echo "*ERROR* (No." & numErrCode & ") " & strErrMsg
            'ResFailure(strErrMsg)
            Wscript.Quit(numRetCode)
        ElseIf counter > 0 Then
            WScript.Sleep checkretrysleep
            WScript.Echo "リトライ実施"
        End If

        Set ret = objShell.Exec(checkcmd & " " & checkargumentarray(i))
        Do While ret.Status = 0
            WScript.Sleep 100
        Loop
        checkret = ret.exitcode
        counter = counter + 1
    Loop Until checkret = 0
Next

'シェルオブジェクト破棄
Set objShell = Nothing
Set checkargumentarray = Nothing
'Set WshShell = Nothing
'Set oFS = Nothing

'スクリプト終了
' End of Quickchk script
Wscript.Quit (0)
```

## recover.vbs

```

###recover script for HULFT###
Option Explicit

' Declare Program Variables
Dim WshShell      ' Wsh Root Object
Dim ts           ' Text Stream
Dim oFS         ' File System Object
Dim oTempFile   ' Temporary File Object
Dim objArgs     ' Input Arguments Object
Dim numErrCode  ' Error Code
Dim numRetCode  ' Return Code
Dim strCmd      ' Script Name
Dim strErrMsg   ' Error Message Description
Dim strID       ' LK Resource ID Dim
Dim strLKAppName ' LK Resource Application Name
Dim strLKCmd    ' LK Command
Dim strLKCmdStr ' LK Command String
Dim strLKCmdLine ' LK Command Line
Dim strLKROOT   ' LK Home Directory
Dim strLKResType ' LK Resource Type
Dim strStdOut   ' Standard Output
Dim strTag      ' LK Resource Tag name
Dim strTempFilename ' Temporary File Name
Dim strWindir   ' Windows Home Directory

' Initialize Constants & Variables
Const ForReading = 1
Const TemporaryFolder = 2
Const TristateUseDefault = -2

strLKAppName = "app"
strLKResType = "gen"
strCmd = Wscript.ScriptName ' Pick up Script Name
Set WshShell = CreateObject("WScript.Shell")
Set oFS = CreateObject("Scripting.FileSystemObject")

' Validate Input Parameters
set objArgs = WScript.Arguments
select Case objArgs.Count
    Case 4
        If objArgs(0) = "-t" Then
            strTag = objArgs(1)
        Else
            Usage
        End If

        If objArgs(2) = "-i" Then
            strID = objArgs(3)
        Else
            Usage
        End If
    Case Else
        strErrMsg = "An incorrect number of input parameters were provided to the " &strCmd & " script."
        WshShell.LogEvent 1, "LifeKeeper Generic Application Recovery Kit scripting error." &vbCrLf
&strErrMsg
        Usage
End Select

```

## LifeKeeper for Windows HULFT クラスター構成ガイド

### Function Usage

```
WScript.Echo "USAGE:"
WScript.Echo "cscript //nologo " &strCmd &" -t tagname -i ID"
WScript.Echo "tagname    - Server unique LifeKeeper tag name for protected resource"
WScript.Echo "ID            - Cluster unique LifeKeeper ID for the protected resource"
WScript.Echo "Windows Scripting Host (cscript.exe) must be installed on the server"
WScript.Quit(1)
```

### End Function

### Function LKFailure(strErrMsg)

```
' Use this function to report a problem with the LifeKeeper system
' This function also references numErrCode, strCMD, strTag, strID
WshShell.LogEvent 1, "LifeKeeper Generic Application Recovery Kit failure." &vbCrLf &"TagName=" &strTag &"`,
ID=" &strID &"`, Error No. " &numErrCode &vbCrLf &strErrMsg
WScript.Quit(1)
```

### End Function

### Function ResFailure(strErrMsg)

```
' Use this function to report a problem with the protected resource
' This function also references numErrCode, strCMD, strTag, strID, numRetCode
RunLKCcmd("lk_err -c FRS_ERR -n " &numErrCode &" -d TO_STDERR -p " &strCMD &" LifeKeeper: TAG=" &strTag &"`, ID=" &strID &"`, Return Code=" &numRetCode &"`; " &strErrMsg)
WScript.Quit(1)
```

### End Function

### Function RunLKCcmd(strLKCcmd)

```
strStdOut=""
strTempFilename = oFS.GetSpecialFolder(TemporaryFolder) & "¥" & oFS.GetTempName
While oFS.FileExists(strTempFilename)
    strTempFilename = oFS.GetSpecialFolder(TemporaryFolder) & "¥" & oFS.GetTempName
Wend
' Setup and run the command
strLKCcmdStr = strLKROOT &"¥bin¥" &strLKCcmd
strLKCcmdLine = "%comspec% /c " & strLKCcmdStr &" > " & strTempFilename
numRetCode = WshShell.Run(strLKCcmdLine, 7, True)
Set oTempFile = oFS.GetFile(strTempFilename)
Set ts = oTempFile.OpenAsTextStream(ForReading, TristateUseDefault)
If not ts.AtEndOfStream Then
    strStdOut = ts.ReadAll ' Get complete output
End If
ts.Close
oTempFile.Delete
```

### End Function

```
' Validate LKROOT Environment Variable
```

```
strLKROOT = WshShell.ExpandEnvironmentStrings("%LKROOT%")
```

```
' Standard setup is done, now perform the desired task
```

```
' Perform commands that recover the resource to the In-Service state
```

```
' -----
' Custom code goes here
```

```
' Set final value of numRetCode; for example
```

```
numRetCode=99
' -----
```

```
' numRetCode is used to identify successful or failure
```

```
' =====
```

```
' (メイン処理)変数宣言
```

```
' =====
```

```
Dim objShell,ret,counter,scquery,scqueryret,netstop,netstart,netstopret,netstartret,usererrcode,usererrmsg
```

```
' =====
```

## LifeKeeper for Windows HULFT クラスタ構成ガイド

```
'(メイン処理)ユーザーパラメーター
=====
'net stop コマンド
netstop = "C:\HULFTF~1\hulft8\bin\utlsvctl.exe -c stop -f"

'net start コマンド
netstart = "net start " & Chr(34) & "HULFT" & Chr(34)

'sc query コマンド
scquery = "sc query HULFT | findstr STATE | findstr RUNNING > nul"

'エラーコード
usererrcode = 27554

'エラーメッセージ
usererrmsg = "HULFT Service recover failed"

=====
'(メイン処理)初期化
=====
scqueryret = 99
netstartret = 99
netstopret = 99

=====
' メイン処理
=====
'シェルオブジェクト作成
Set objShell = CreateObject("WScript.Shell")

'sc query & net stop 実行
Set ret = objShell.Exec("cmd.exe /c " & scquery)
Do While ret.Status = 0
    WScript.Sleep 100
Loop
scqueryret = ret.exitcode
If scqueryret = 0 Then
    Set ret = objShell.Exec("cmd.exe /c " & netstop)
    Do While ret.Status = 0
        WScript.Sleep 100
    Loop
    netstopret = ret.exitcode
    If netstopret <> 0 Then
        numErrCode = usererrcode
        strErrMsg = usererrmsg
        numRetCode = 1
        WScript.Echo "*ERROR* (No." & numErrCode & ") " & strErrMsg
        'ResFailure(strErrMsg)
        Wscript.Quit(numRetCode)
    End If
End If

'sleep 処理
WScript.Sleep 10000

'net start 実行
Set ret = objShell.Exec("cmd.exe /c " & netstart)
Do While ret.Status = 0
    WScript.Sleep 100
Loop
netstartret = ret.exitcode
If netstartret <> 0 Then
    numErrCode = usererrcode
```

## LifeKeeper for Windows HULFT クラスタ構成ガイド

```
strErrMsg = usererrmsg
numRetCode = 1
WScript.Echo "*ERROR* (No." & numErrCode & ")" & strErrMsg
'ResFailure(strErrMsg)
Wscript.Quit(numRetCode)
End If
```

```
'sc query 実行
scqueryret = 99
Set ret = objShell.Exec("cmd.exe /c " & scquery)
Do While ret.Status = 0
    WScript.Sleep 100
Loop
scqueryret = ret.exitcode
If scqueryret <> 0 Then
    numErrCode = usererrcode
    strErrMsg = usererrmsg
    numRetCode = 1
    WScript.Echo "*ERROR* (No." & numErrCode & ")" & strErrMsg
'ResFailure(strErrMsg)
Wscript.Quit(numRetCode)
End If
```

```
'シェルオブジェクト破棄
Set objShell = Nothing
Set WshShell = Nothing
Set oFS = Nothing
```

```
'スクリプト終了
' End of Recover script
Wscript.Quit (0)
```



## 免責事項

- 本書に記載された情報は予告なしに変更、削除される場合があります。最新のものをご確認ください。
- 本書に記載された情報は、全て慎重に作成され、記載されていますが、本書をもって、その妥当性や正確性についていかなる種類の保証もするものではありません。
- 本書に含まれた誤りに起因して、本書の利用者に生じた損害については、サイオステクノロジー株式会社は一切の責任を負うものではありません。
- 第三者による本書の記載事項の変更、削除、ホームページ及び本書等に対する不正なアクセス、その他第三者の行ためにより本書の利用者に生じた一切の損害について、サイオステクノロジー株式会社は一切の責任を負うものではありません。
- システム障害などの原因によりメールフォームからのお問い合わせが届かず、または延着する場合がありますので、あらかじめご了承ください。お問い合わせの不着及び延着に関し、サイオステクノロジー株式会社は一切の責任を負うものではありません。

### 【著作権】

本書に記載されているコンテンツ（情報・資料・画像等種類を問わず）に関する知的財産権は、サイオステクノロジー株式会社に帰属します。その全部、一部を問わず、サイオステクノロジー株式会社の許可なく本書を複製、転用、転載、公衆への送信、販売、翻案その他の二次利用をすることはいずれも禁止されます。またコンテンツの改変、削除についても一切認められません。

本書では、製品名、ロゴなど、他社が保有する商標もしくは登録商標を使用しています。

---

サイオステクノロジー株式会社

住所：〒106-0047

東京都港区南麻布 2 丁目 12-3 サイオスビル

URL : <https://sios.jp>