



**SteelEyer® LifeKeeper® for Linux**

汎用アプリケーション スクリプト

for アプレッソ® DataSpider® Servista

構築手順書

初版

2011年4月15日

この文章内における掲載情報の二次利用においては、ご自身の判断と責任の下におこなってください。株式会社アプレッソは、それらの情報を利用した事によって生じたいかなる損害についても、その責を負わないものとします。

アプレッソ、Appresso、DataSpider、およびアプレッソのロゴ、DataSpider のロゴは株式会社アプレッソの登録商標です。

SteelEye Technology and LifeKeeper are registered trademarks and SteelEye is a trademark of SteelEye Technology,Inc.

その他、記載されている会社名および製品名は、各社の商標または登録商標です。

記載内容は予告無く変更される場合があります。予めご了承ください。

Copyright© 2011 Appresso KK.

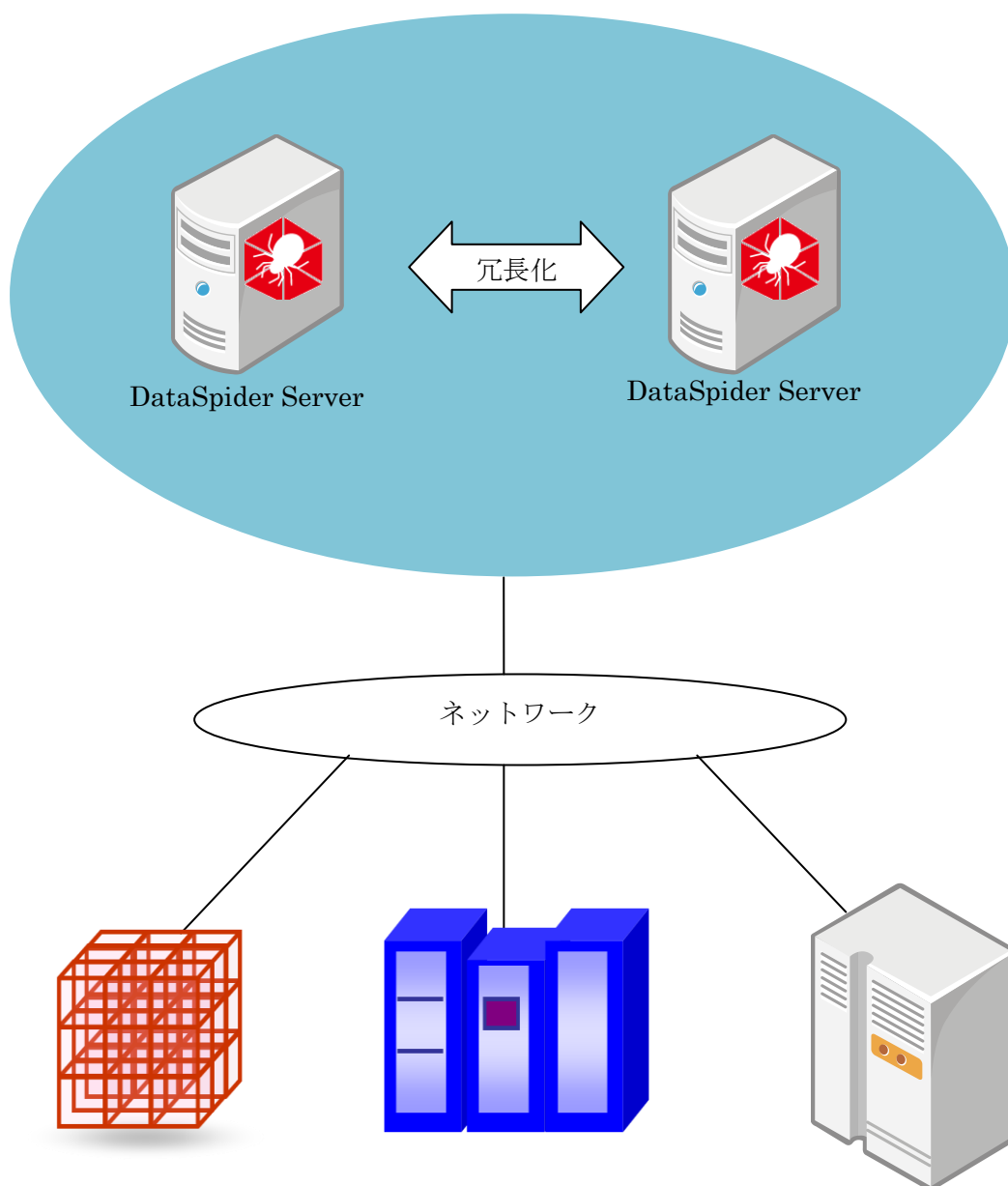
## 【内容】

1. DataSpider Servista の冗長化.....	3
1.1 単体構成での冗長化.....	3
1.2 ドキュメントについて .....	4
2. 構築手順 .....	4
2.1 DataSpider Server .....	4
2.2 LifeKeeper ソフトウェア .....	4
2.3 構成図.....	4
2.4 前提 .....	6
2.5 構築手順 .....	6
3. DataSpider Server インストール方法 .....	12
4. 動作確認コマンド作成方法 .....	13
5. 改版履歴 .....	17

## 1. DataSpider Servista の冗長化

DataSpider Servista は、DataSpider Server と DataSpider Studio で構成されます。本ドキュメントは、DataSpider Server の単体構成での冗長化について説明いたします。

### 1.1 単体構成での冗長化



## 1.2 ドキュメントについて

本ドキュメントにおける構築手順では、DataSpider Server を含んだ新規冗長化構成の構築を対象としています。そのため、既存の DataSpider Server の冗長化に関しては本ドキュメントに記載されておられません。既存の DataSpider Server の冗長化に関するお問い合わせにつきましては、「お問い合わせ」をご参照下さい。

また、本ドキュメントの内容上、DataSpider Server のセットアップ手順が簡易に示されますが、詳細な DataSpider Server のセットアップ手順や構成手順の関しましては、DataSpider Servista のマニュアルをご参照下さい。

## 2. 構築手順

### 2.1 DataSpider Server

DataSpider Server は、LifeKeeper で冗長化する際は、共有データ領域に格納します。

### 2.2 LifeKeeper ソフトウェア

単体構成の冗長化で使用する LifeKeeper ソフトウェアは以下となります。

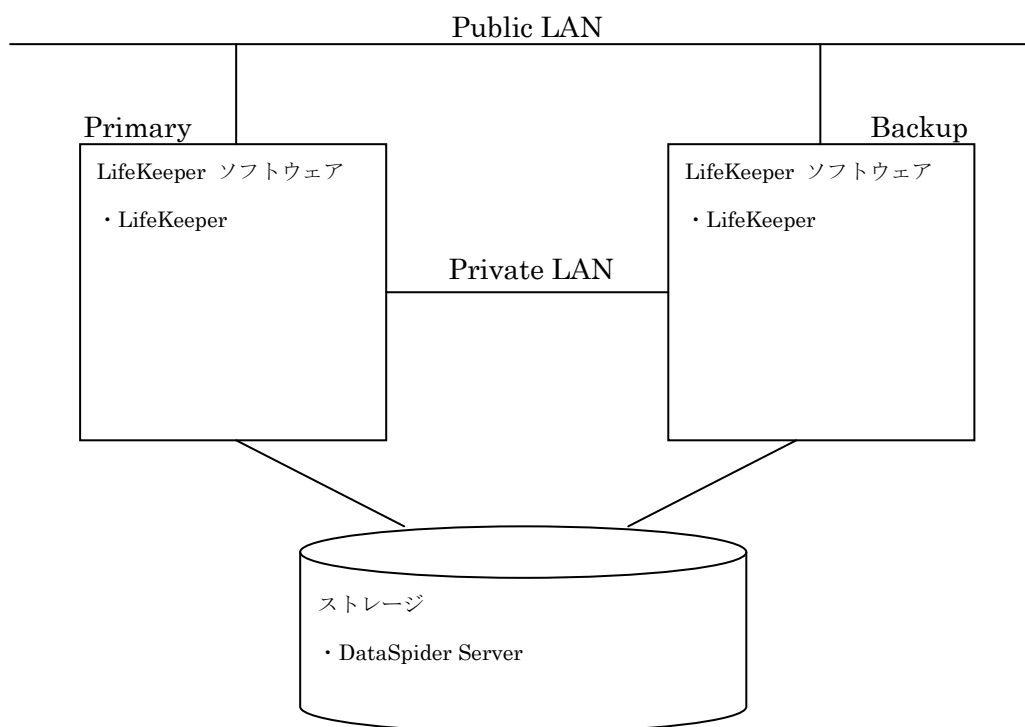
- LifeKeeper for Linux
- Protection Suite for Linux (データレプリケーション構成の場合。)

### 2.3 構成図

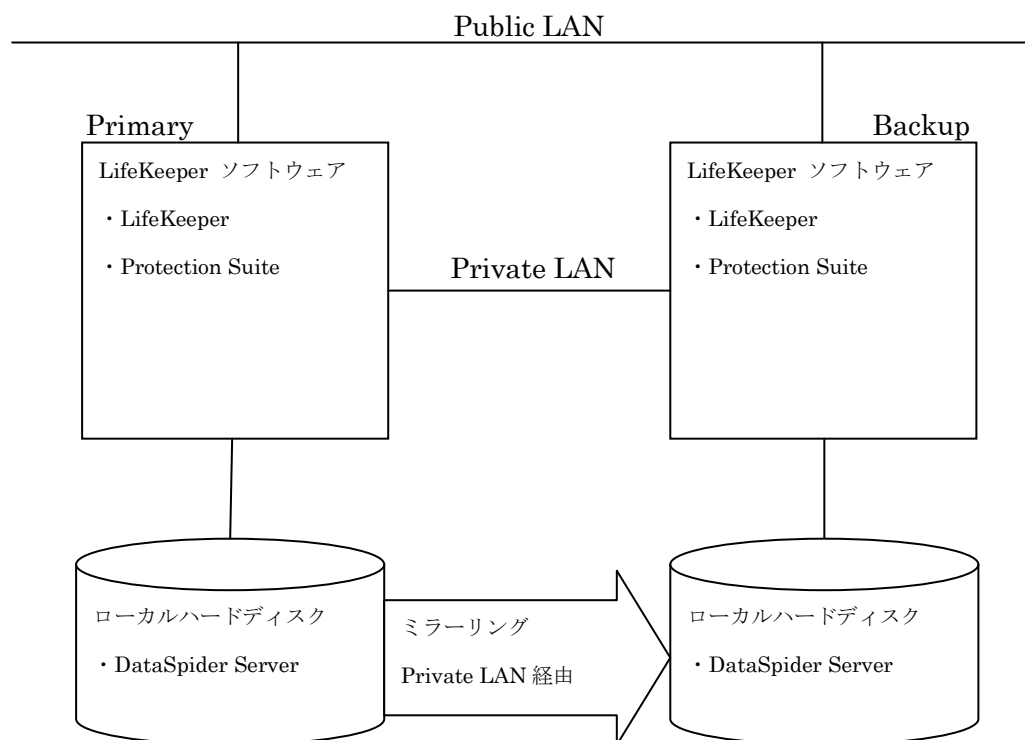
以下の 2 種類の方式で DataSpider Server の冗長化構成をおこなうことができます。

- 共有ディスク構成
- データレプリケーション構成

共有ディスク構成の場合、最終的に以下の構成になります



データレプリケーション構成の場合は、以下のような構成になります。



## 2.4 前提

この構築手順では、以下のことを前提として説明しています。

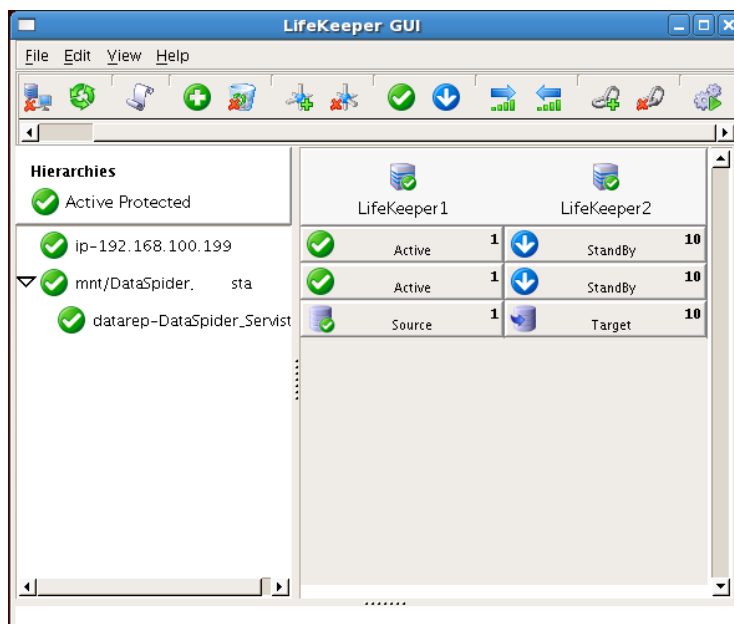
- スーパーユーザ（root）または同等の権限を持つユーザで作業をおこないます。
- LifeKeeper の基本的な使用方法（コミュニケーションパス、リソース作成、依存関係の作成）を理解しています。
- LifeKeeper が既にインストールされています。
- Primary サーバーを “LifeKeeper1”、Backup サーバーを “LifeKeeper2” とします。
- 共有データ領域（マウントボリューム）を /mnt/DataSpider\_Servista と定義しています。

## 2.5 構築手順

新規に冗長化する DataSpider Server を構築する手順を以下に示します。

今回の構築手順はデータレプリケーション構成で示しますが、共有ディスク構成でも手順は同一となります。

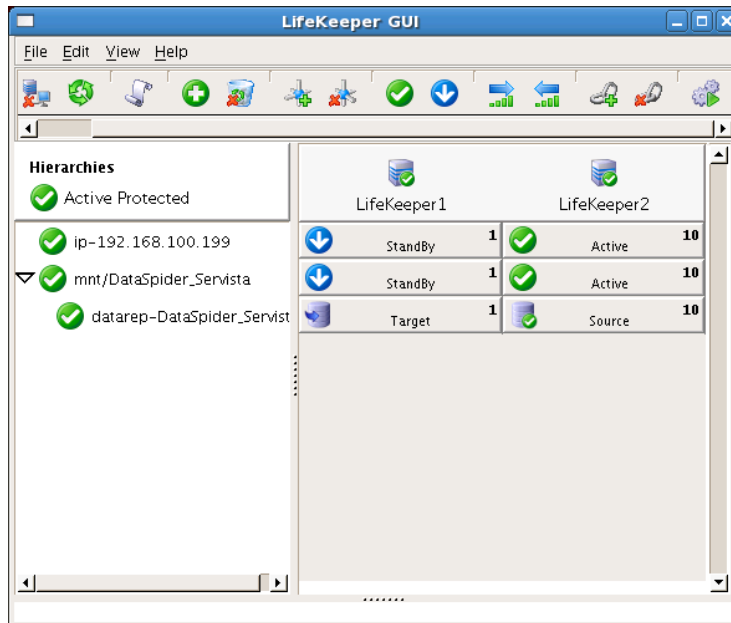
(1)仮想 IP 用の IP リソースと Data Replication を作成します。



(2)Primary サーバーで共有データ領域上に DataSpider Server をインストールします。

※DataSpider Server のインストールに関しては、「3 DataSpider Server インストール方法」を確認下さい。

(3)IP リソースとボリュームリソースを Backup サーバーに切り替えます。



(4)Primary サーバーと同様に、共有データ上に DataSpider Server をインストールします。

(5)IP リソースとボリュームリソースを Primary サーバーに切り替えます。

(6)DataSpider Server サービスの汎用リソースを作成します(スクリプト作成は任意のフォルダ)。

- Restore スクリプト : /opt/LifeKeeper/Dataspider\_Script/restore
- Remove スクリプト : /opt/LifeKeeper/Dataspider\_Script/remove
- QuickCheck スクリプト : /opt/LifeKeeper/Dataspider\_Script/quickcheck

3つのスクリプトは、LifeKeeper で用意されているサンプルスクリプトを、利用します。

サンプルスクリプトは、サイオステクノロジ株式会社の Web よりダウンロードして下さい。また、スクリプトの修正方法は、“SteelEye Lifekeeper for Linux GenericARK 開発ガイド” を参照して下さい。



(a)Restore スクリプト作成方法

Restore スクリプトでは、DataSpider の起動コマンドを次のように定義します。

```
APP_START="/mnt/DataSpider_Servista/server/bin/DataSpiderServer" # 起動コマンド  
APP_START_ARGS="" # 起動コマンドの引数
```

起動コマンドは、必ずバックグラウンドで実行するよう “&” を付加して下さい。

```
# 起動処理  
APP_start()  
{  
    echo "$(date +%Y/%m/%d %H:%M:%S) : $TAG starting on $HOSTNAME" >>  
$lklog  
    $APP_START $APP_START_ARGS &>> $lklog 2>&1 &  
    wait $!  
}
```

Restore スクリプトでは、起動コマンド発行後に動作確認コマンドを発行し、正常に起動できたかを確認します。動作確認コマンドは、「4.動作確認コマンド作成方法」を参照し、作成して下さい。

以下のように作成した動作確認コマンドを指定してください。

```
APP_CHECK[0]="/mnt/DataSpider_Servista/server/bin/ScriptRunner" # 動作確認コマンド1  
APP_CHECK_ARGS[0]="/mnt/DataSpider_Servista/server/bin/Dead-Alive.xml" # 上記コマンドの引数
```

サンプルスクリプトでは、動作確認コマンドの戻り値で、リトライするかを判断します。そのため、以下の様な修正が必要となります。

```
# 確認処理
# 引数: チェック成功と扱う値
# 戻り値: 成功時,引数と同じ値。失敗時,監視コマンドの戻り値。
APP_check()
{
    local S_CODE=$1
    local RET=1
    for (( CNT=0; $CNT < ${#APP_CHECK[@]}; CNT=$((CNT+1)) );
    do
        if [ -x "${APP_CHECK[$CNT]}" ]; then
            ${APP_CHECK[$CNT]} ${APP_CHECK_ARGS[$CNT]} >>
/dev/null 2>&1 &
            wait $!
            SRSTAT=$?
            if [ $SRSTAT -eq 0 ]
            then
                RET=$SRSTAT
            else
                RET=12
            fi
        else
            RET=255 # コマンドがなければエラー
        fi

        case "$RET" in
            $S_CODE) ;; # 何もせずに次のチェックへ進む
            *) return $RET ;; # エラー
        esac
    done

    return $S_CODE;
}
```

上記例では、動作確認コマンドが正常終了しない場合、リトライのステータスを返しています。

DataSpider の場合、起動コマンドを発行してから正常に DataSpider Server が起動するまでの間、リトライを何度か繰り返す必要がありますが、何らかの障害が発生して、正常に起動出来ない場合もありますので、リトライの回数に制限を持たせるなどの対応をして下さい。

### (b)quickChek スクリプト作成方法

「(a)Restore スクリプト作成方法」と同様に“動作確認コマンド”を追加して下さい。

quickCheck スクリプトは、動作確認コマンドが一定時間内に応答がない場合、タイムアウトするように作成してください。DataSpider Server が起動していても、動作確認コマンドがタイムアウトした場合、正常に起動しているとはいえません。必ずフェールオーバーするように設定して下さい。設定は以下のように /etc/default/lifekeeper に、設定を 1 行追加します。

```
“$TAG_TIMEOUT=秒”
```

“\$TAG”は、登録する汎用アプリケーション名になります。  
今回の例ですと、“APP1”となり、その設定例は以下のようになります。

**APP1\_TIMEOUT=60**

※サンプルスクリプトでは、デフォルトの 22 秒でタイムアウトするように設定されています。

### (c)remove スクリプト作成方法

「(a)Restore スクリプト作成方法」と同様に“動作確認コマンド”と、DataSpider Server の“停止コマンド”を追加します。  
以下のように、“動作確認コマンド”と“停止コマンド”を指定して下さい。

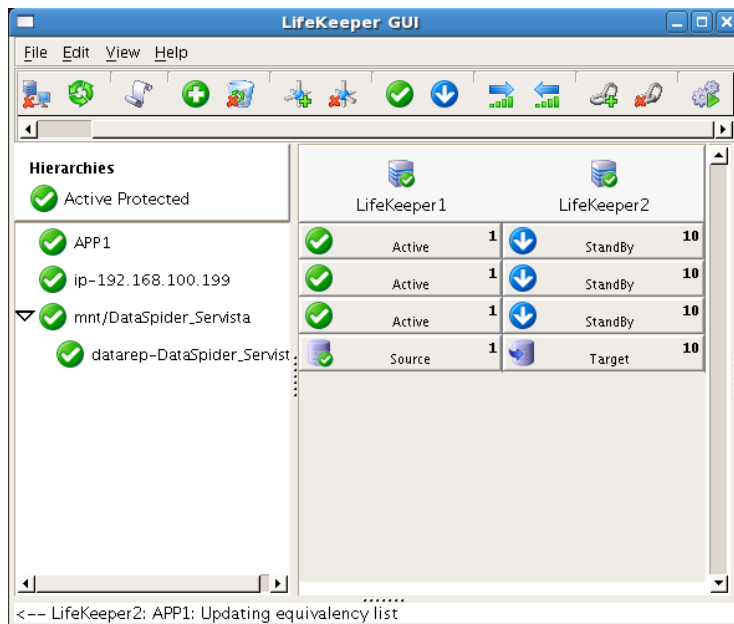
```
APP_STOP="/mnt/DataSpider_Servista/server/bin/Shutdown" # 停止コマンド
APP_STOP_ARGS="" # 停止コマンドの引数

APP_FORCE_STOP="" # 強制停止コマンド
APP_FORCE_STOP_ARGS="" # 強制停止コマンドの引数

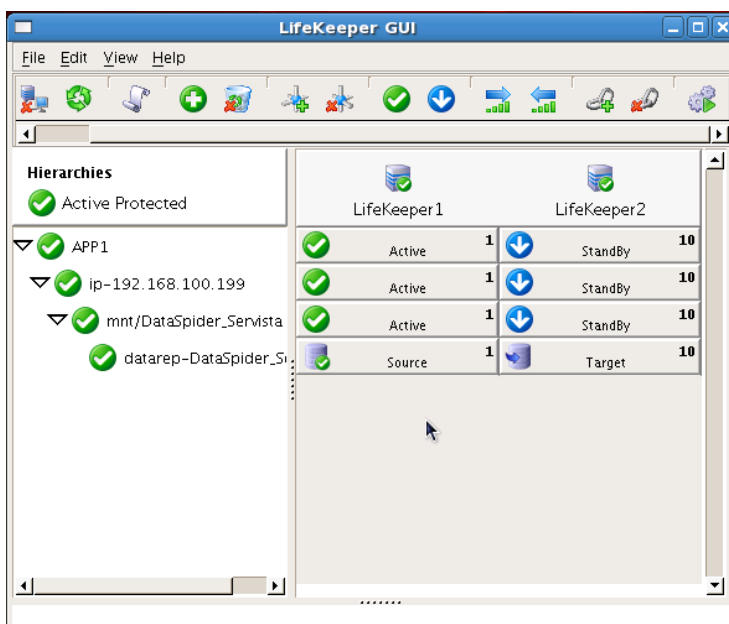
APP_CHECK[0]="/mnt/DataSpider_Servista/server/bin/ScriptRunner" # 動作確認コマンド
1
APP_CHECK_ARGS[0]="/mnt/DataSpider_Servista/server/bin/Dead-Alive.xml" # 上記コマンドの引数
```

動作確認コマンドが正常終了しない場合は、終了をあらわすリターンコード (1)を、返すように修正します。

正常に汎用アプリケーションが追加されますと、以下の様な表示になります。



最後に、以下のようにリソース階層を作成します。



### 3. DataSpider Server インストール方法

DataSpider Server のクラスタ構成には、リポジトリ DB のあり/なし、共有ディスクへのインストールの有無の組み合わせを選択することができます。

インストール方法の詳細については、製品に同梱の「クラスタ構成ガイド」を参照して下さい。

#### (1) 共有ディスク型 (DataSpider Server を共有ディスクに配置)

(a) LifeKeeper のリソース設定において、次のリソースを登録する。

- ・ IP リソース
- ・ ファイルシステムリソース

(b) リポジトリ DB 用データベースを用意する (リポジトリ DB を使用する場合)。

リポジトリ DB 使用時は、共有ディスク内または、データレプリケーションのパーティション内に作成して下さい。

(c) 共有ディスクに DataSpider Server をインストールする。

この際、Primary/Backup それぞれのサーバーから共有ディスクの同じフォルダに DataSpider Server をインストールして下さい。

※インストール方法については、DataSpider Servista 本体に付属のインストールガイドを参照して下さい。

#### (2) データレプリケーション型

(a) LifeKeeper のリソース設定において、次のリソースを登録する。

- ・ IP リソース
- ・ データレプリケーションリソース

(b) リポジトリ DB 用データベースを用意する (リポジトリ DB を使用する場合)。

リポジトリ DB 使用時は、データレプリケーションのパーティション内に作成して下さい。

(c) データレプリケーション内に、DataSpider Server をインストールする。

※インストール方法については、DataSpider Servista 本体に付属のインストールガイドを参照して下さい。

#### 4. 動作確認コマンド作成方法

DataSpider Servista では、外部のアプリケーションからスクリプトをコマンド実行するための ScriptRunner 機能を用意しています。

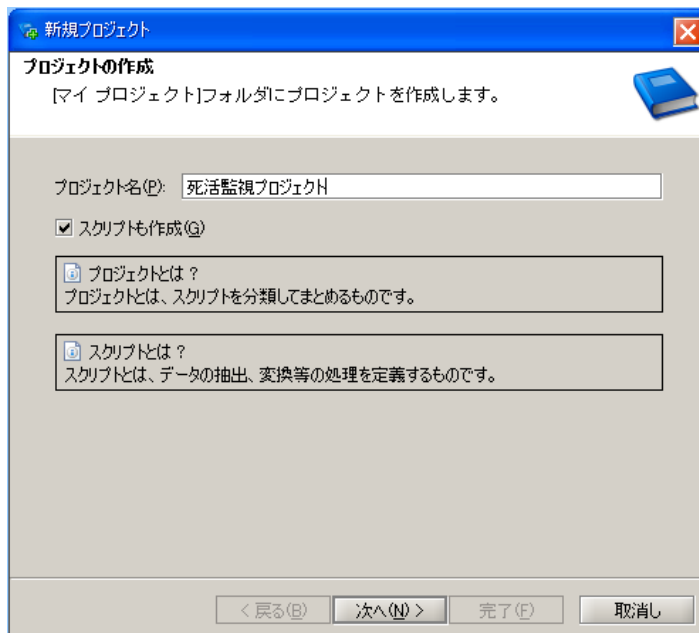
この機能を用いて動作確認用スクリプトをコマンド実行することで、ネットワーク疎通確認に加え、DataSpider Servista の実行状況の観点でも動作確認を実施することができます。動作確認用コマンド作成は、以下の手順でおこなってください。

(1)DataSpider Studio を起動する。

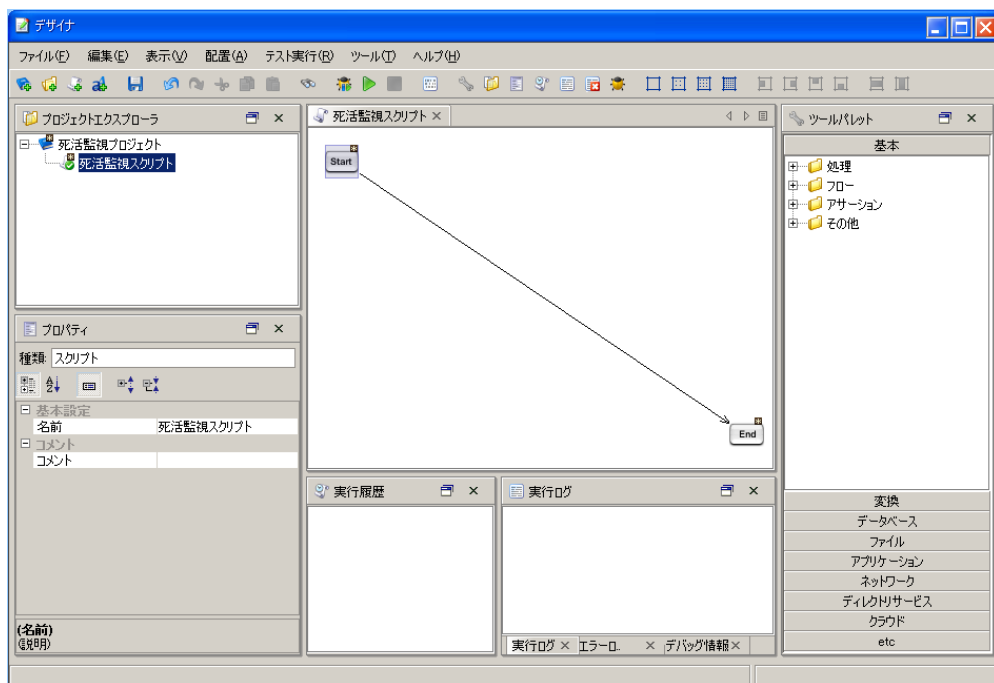
(2)[新規プロジェクト]アイコンより[プロジェクト作成ウィザード]を開く。



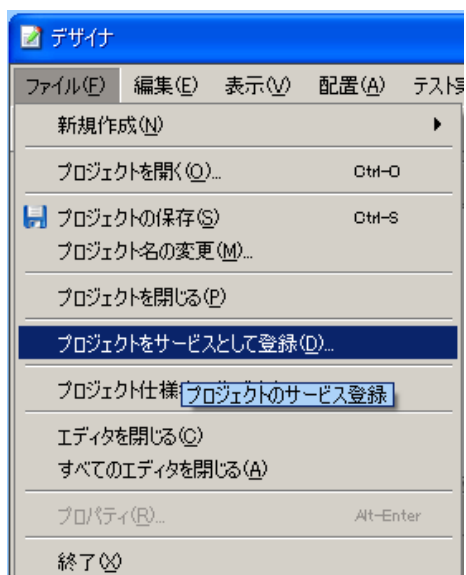
(3)プロジェクト名、スクリプト名 に任意の名称を入力し、プロジェクト、スクリプトを作成する。

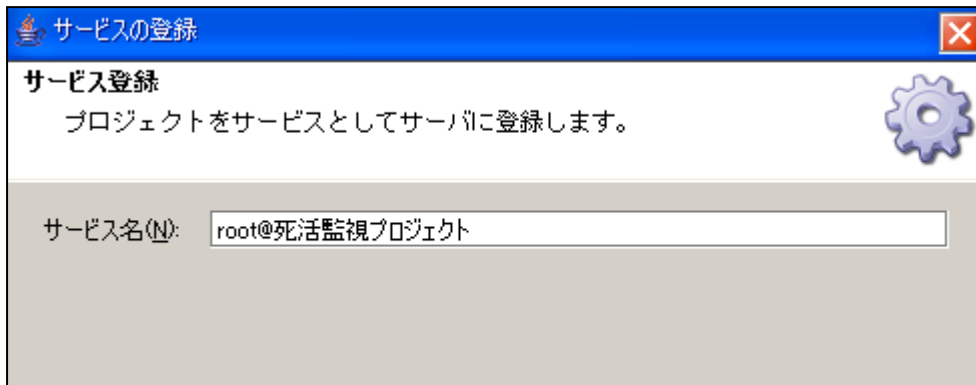


- (4) [デザイナ] ツールよりスクリプトの処理内容を作成する。  
[Start] アイコンから [End] アイコンまでフローを引きます。



- (5) メニュー [ファイル]-[プロジェクトをサービスとして登録] よりスクリプトをサービスに登録する。



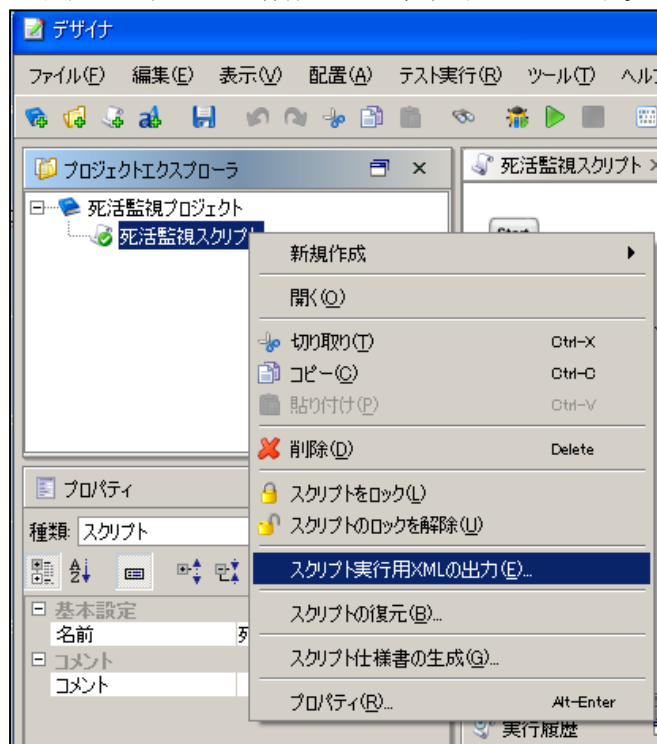


サービス名は、デフォルトで“作成ユーザ名+プロジェクト名”となります。  
サービス名を変更する場合は、ここで変更してください。

#### (6)ScriptRunner の設定

作成したスクリプトを LifeKeeper より実行するための起動設定ファイルを作成します。

起動設定ファイルの作成方法は、以下となります。



- 実行したいスクリプトをマウスの右ボタンでクリックします。
- メニューが表示されますので、[スクリプト実行用XMLの出力(E)]を選択します。
- 起動ファイルの保存先を聞いてきますので、保存先を指定します。

ScriptRunner での起動方法としましては、DataSpider Server のインストールディレクトリの下 の server¥bin フォルダにある ScriptRunner.exe を実行します。

```
>scriptrunner.exe <起動設定ファイル> <第一引数> <第二引数>
```



※ScriptRunner の設定方法については、DataSpider Servista のオンラインヘルプ [ScriptRunner]の項を参照してください。

5. 改版履歴

日付	版数	更新内容
2011/04/15	初版	・初版