



SteelEyer® LifeKeeper® for Windows

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2月18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。予めご了承下さい。

Copyright© 2011 Appresso KK.

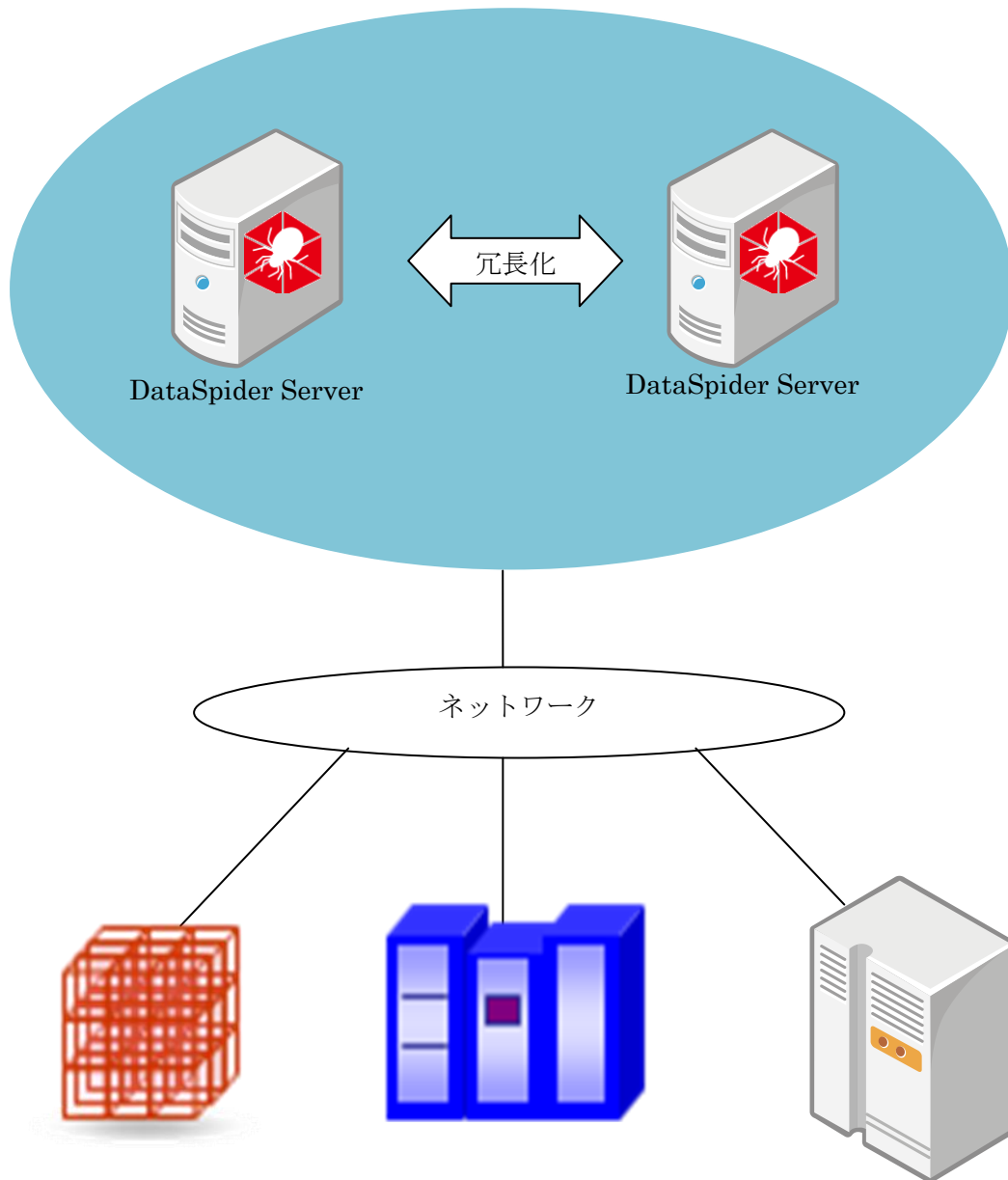
【内容】

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.....	11
4. 死活確認用スクリプトについて.....	12
5. TimeOut_Check について.....	15
6. 改版履歴.....	18

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りません。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。

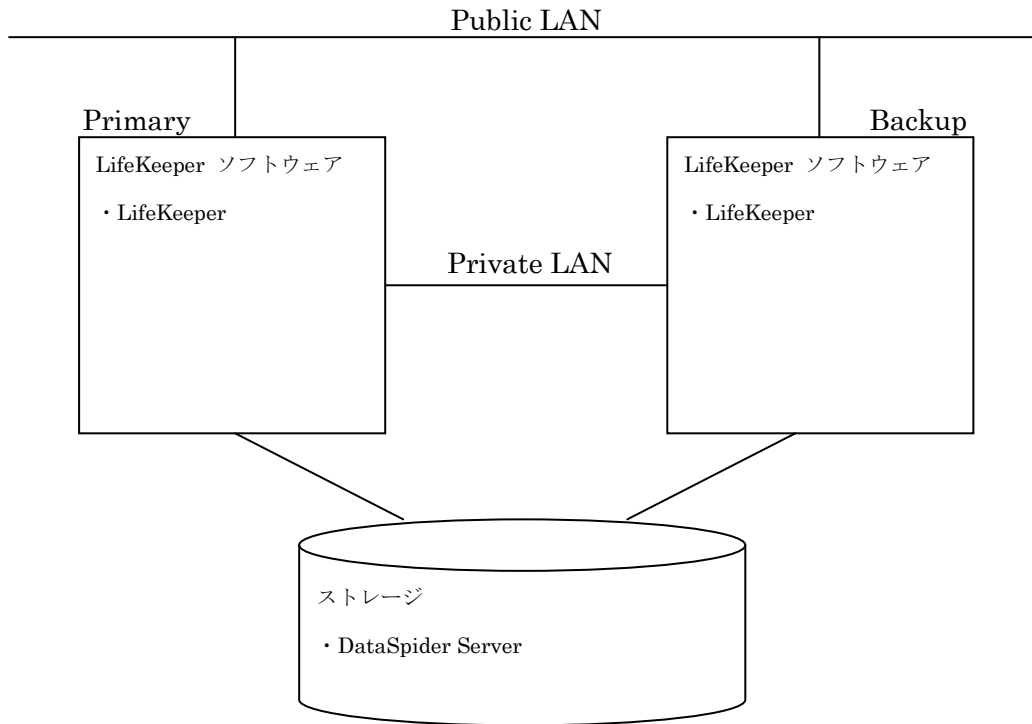
- LifeKeeper for Windows
- Protection Suite for Windows (データレプリケーション構成の場合。)

2.3 構成図

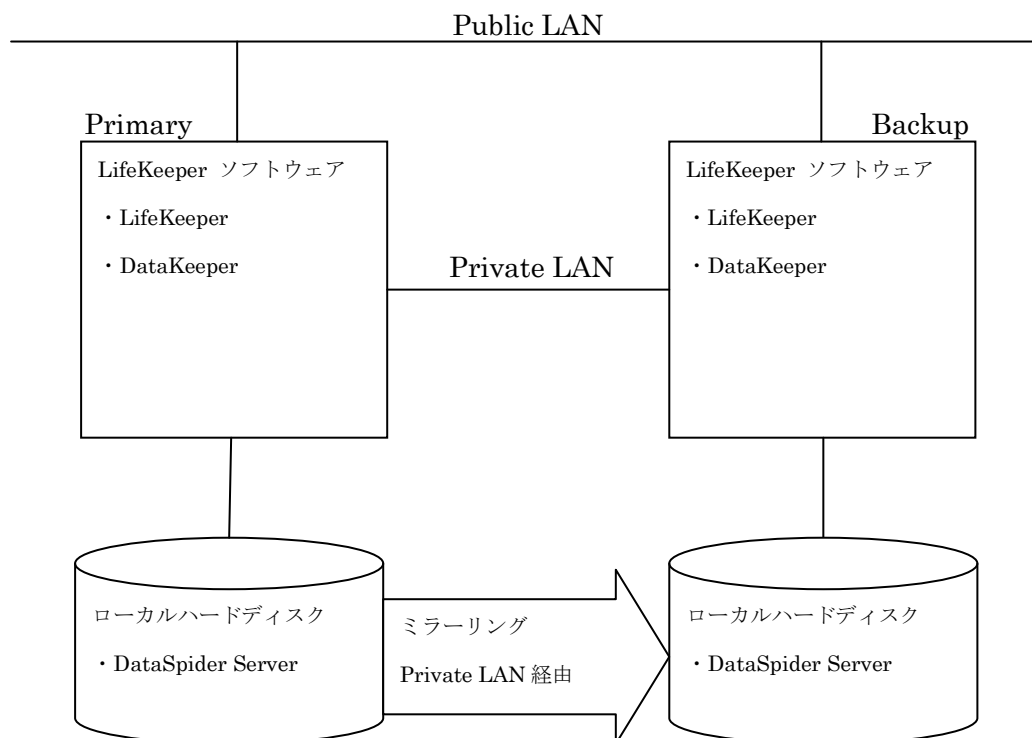
以下の 2 種類の方式で DataSpider Server の冗長化構成をおこなうことができます。

- 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

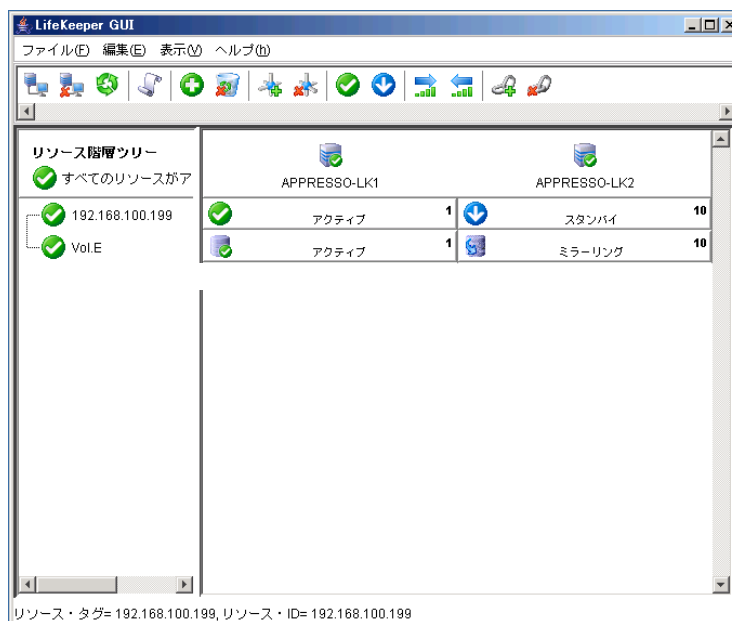
この構築手順では、以下のことを前提として説明しています。

- Administrator 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（通信パス、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“APPRESSO-LK1”、Backup サーバーを“APPRESSO-LK2”とします。
- 共有データ領域を D:とします。

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。

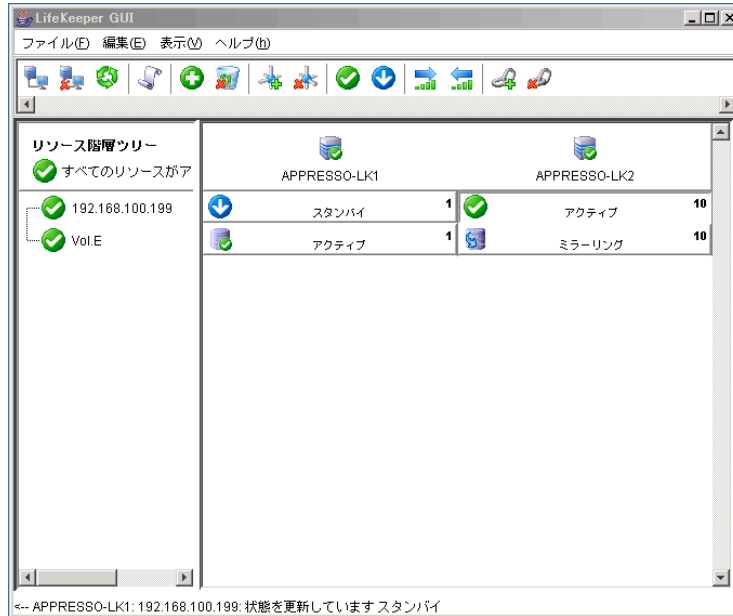
(1)仮想 IP 用の IP リソースとボリュームリソースを作成します。



(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。

- Restore スクリプト : C:\¥generic¥DataSpider_Servista¥restore
- Remove スクリプト : C:\¥generic¥DataSpider_Servista¥remove
- QuickCheck スクリプト : C:\¥generic¥DataSpider_servisa¥quickcheck

(a)Windows サービスに登録した場合

①Restore、Remove スクリプト作成方法

Restore、Remove スクリプトは、LifeKeeper で用意されているサンプルスクリプトが、そのまま使用できます。

Restore スクリプト

```
%LKROOT%\Admin¥kit¥app¥templates¥example_service¥restore.pl
```

Remove スクリプト

```
%LKROOT%\Admin¥kit¥app¥templates¥example_service¥remove.pl
```

%LKROOT%は、LifeKeeper インストールディレクトリを指します。

デフォルトでは、C:\LK となります。

②QuickChek スクリプト作成方法

Restore や Remove と同様に QuickCheck のサンプルスクリプトがあるので、これを修正し使用する方法を説明致します。

サンプルスクリプトは、Restore、Remove と同様のディレクトリにあります。

%LKROOT%\Admin\kit\app\templates\example_service\QuickCheck.pl

このサンプルスクリプトに、「4.死活確認スクリプトについて」を参照し、作成した DataSpider のスクリプトを実行して、そのリターンコードをチェックする処理を追加します。

QuickCheck テンプレートスクリプトの Windows サービスのステータスチェックの後に、以下のように追加します。

```
#-----  
#  
# Check the state of DataSpider ScriptRunner Check  
#  
#-----  
  
$DSROOT = "E:/DataSpider/DataSpiderServista245/server/bin";  
$SCRROOT = "E:/DataSpider/DataSpiderServista245/server/bin/Dead-Alive.xml";  
$RunCMD = "$DSROOT/ScriptRunner $SCRROOT";  
  
$DSStat = `$RunCMD`;  
$Pos1 = index $DSStat, "EXIT_STATUS=";  
$Pos2 = index $DSStat, "|", $Pos1;  
  
$Exit_Stat = substr($DSStat, $Pos1+12, $Pos2-($Pos1+12));  
  
if ( $Exit_Stat != 0 ) {  
    exit 1;  
}
```

下線の箇所は、DataSpider Server のインストール先、死活確認スクリプトの配置先に応じて適宜修正して下さい。

注意事項：

- ・ 汎用アプリケーション登録時、ウィザードの要求するアプリケーション情報には、「DataSpider のサービス名」を指定して下さい。
デフォルトの “DataSpider_Servista” を指定して下さい。
- ・ Windows サービスコントローラにおける 「DataSpider サービス」 の起動オプショ

ンは、手動に設定する必要があります。

(b)Windows サービスに登録しなかった場合

DataSpider Server インストールフォルダ $\%server\%bin$ にある実行ファイルを起動します。

Restore スクリプト : DataSpiderServer.exe

Remove スクリプト : Shutdown.exe

QuickCheck スクリプト : (a)Windows サービスに登録した場合と同様に作成します。



(7) スクリプトタイムアウトの TimeOut_Check 汎用リソースを作成します。
 ※タイムアウトの汎用リソース作成の意義とその方法に関しては、「5. TimeOut_Chek について」を確認下さい。

- Restore スクリプト : C:\¥generic¥TimeOut_Check¥restore
- Remove スクリプト : C:\¥generic¥TimeOut_Check¥remove
- QuickCheck スクリプト : C:\¥generic¥TimeOut_Check¥quickcheck



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
- ・ ボリュームリソース

(b)Primary/Backup 各サーバーのローカルディスクにリポジトリ 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)。

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

この際、DataSpider Server を Windows サービスに登録する場合には、両方のサーバーにサービス登録するために、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ーに 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
- ・ ボリュームリソース

(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、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. 死活確認用スクリプトについて

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

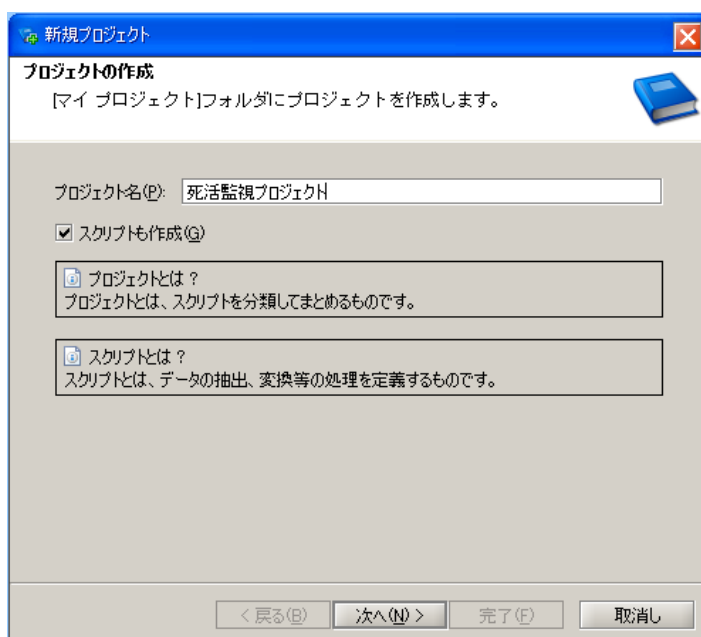
この機能を用いて確認用のスクリプトの実行を確認することで Windows サービスの実行状況やネットワーク疎通確認に加え、DataSpider Servista の実行状況の観点でも死活確認を実施することができます。死活確認用スクリプト作成は、以下の手順でおこなってください。

(1)DataSpider Studio を起動する。

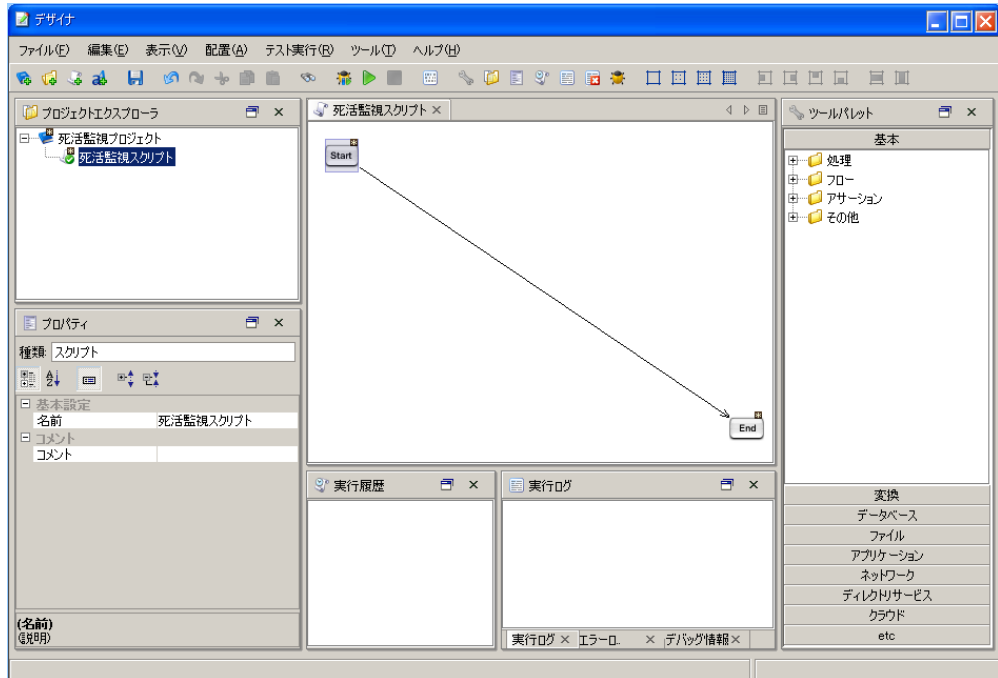
(2)[新規プロジェクト]アイコンより[プロジェクト作成ウィザード]を開く。



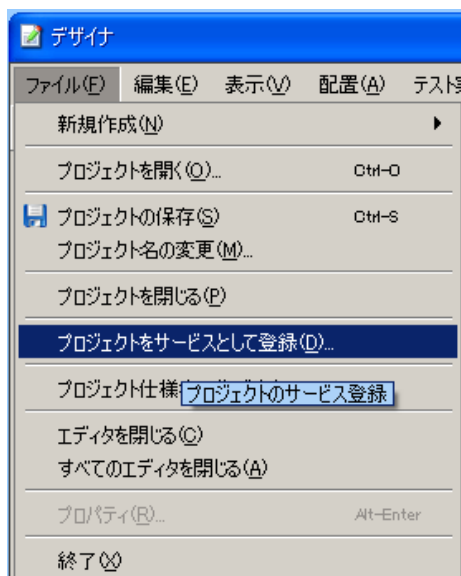
(3)プロジェクト名、スクリプト名 に任意の名称を入力し、プロジェクト、スクリプトを作成する。

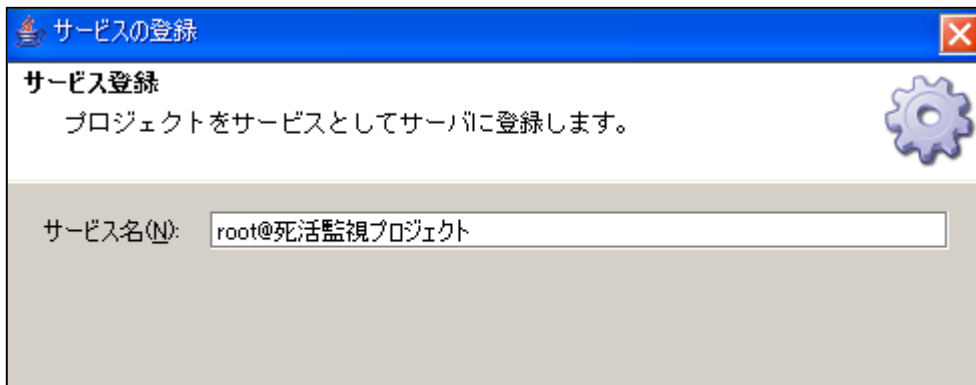


- (4) [デザイナ] ツールよりスクリプトの処理内容を作成する。
[Start] アイコンから [End] アイコンまでフローを引きます。



- (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。



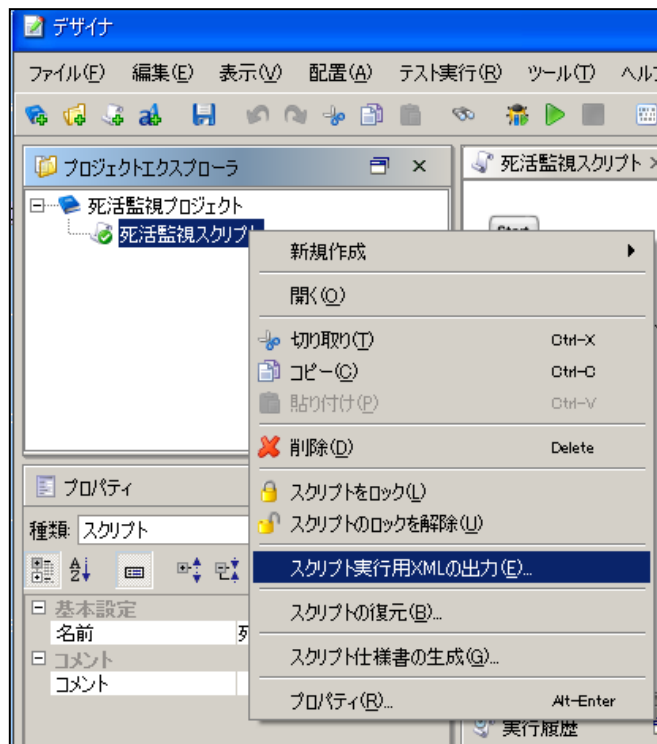


サービス名は、デフォルトで“作成ユーザ名+プロジェクト名”となります。
サービス名を変更する場合は、ここで変更してください。

(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.exe <起動設定ファイル> <第一引数> <第二引数>
```

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に `server¥bin` フォルダにある `ScriptRunner.exe` を実行します。

※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. TimeOut_Check について

2.で QuickCheck に設定した死活確認スクリプトが、通常のレスポンスが返せる状態であれば問題ありませんが、レスポンスを返せないような状態も想定されます。そのような場合、QuickCheck スクリプトが実行中のままとなり、フェールオーバーさせることが出来ません。

QuickCheck スクリプトをタイムアウトさせ、障害と判断し、フェールオーバーさせる必要があります。その設定方法を説明します。

(1)QuickCheck スクリプトのタイムアウト設定

Windows のレジストリに、LifeKeeper の QuickCheck スクリプトが一定時間を過ぎた場合、タイムアウトする設定が必要です。

`HKLM¥Software¥SteelEye¥LifeKeeper¥RK¥app¥MAX_QUICK_CHECK`

上記のレジストリに記載されている値は時間ではなくタイムアウトの時間を計算する上の数値です。実際のタイムアウト値はリソースの監視間隔を掛けた値になります。

<p>QuickCheck の監視間隔 = 3 分 (リソースを作成する際の設定値) MAX_QUICK_CHECK = 1 180 (3分を秒に) × 1 = 180 秒</p>
--

以上がデフォルトでのタイムアウト値となり、レジストリの修正で制御可能ですが構造上、監視間隔の倍数に限定されます。

また QuickCheck の処理がタイムアウトを超えた場合、チェックプロセスは abort され、以下のログに記録されます。

LifeKeeper インストールフォルダ¥out¥ResmonTimeout.log

(2)TimeOut_Check 汎用リソースのスクリプト作成

DataSpider の QuickCheck スクリプトがタイムアウトした場合、フェールオーバーする汎用リソースを登録するためのスクリプトを作成します。

DataSpider と同様に、

- Restore
- Remove
- QuickCheck

の 3 つが必要です。

このうち、Restore,Remove は、何も処理は必要ありません。

“exit 0;” のみで構いません。

QuickCheck においては、(1).でタイムアウト時に作成される log ファイルが存在するかチェックをおこない、存在する場合にはエラーとします。

この事により、LifeKeeper がエラーと感知し、フェールオーバーが発生します。

• サンプルスクリプト(perl)

```
#-----  
#  
# Check the state of DataSpider ScriptRunner Check  
#  
#-----  
  
if (-e "c:/lk/out/ResmonTimeout.log") {  
    unlink 'c:/lk/out/ResmonTimeout.log';  
    exit 1;  
}
```

下線の箇所は、LifeKeeper のインストール先、ログ出力先に応じて適宜修正して下さい。

※TimeOut_Check 用リソースは、DataSpider Servista リソース階層のみ存在する場合には問題ありませんが、他にもリソースが共存する場合には、そのリソース

が **TimeOut** を起こした場合にも機能致します。**TimeOut_Check** リソースを採用する場合には、ご注意ください。

6. 改版履歴

日付	版数	更新内容
2011/02/18	初版	・初版