Arcserve® Replication/High Availability PowerShell

スクリプト実行ガイド

2014/10

arcserve®

目次

改訂履歴

2014/09 **r16.5 版 初版リリース**

2014/10 **製品名変更に伴う修正**

2015/10 **サンプルスクリプトに RHA コントロールサービスからの切断コマンドを追加**

はじめに

このドキュメントは Arcserve Replication / High Availability r16.5（以降 Arcserve RHA と表記）の Arcserve RHA PowerShell (以降 RHA PowerShell と表記) について、そのインストール方法やスクリプトの実効方法を解説します。このドキュメントではバージョン r16.5 の利用を想定して記載しますが、r16、r15 および r12.5 でも同様の方法で利用できます。

ドキュメントおよびサンプルスクリプト利用規定

このドキュメントに記載される内容やスクリプト利用にあたっての規定事項です。スクリプト利用の前に必ず一度お読みください。なお、この利用規定は巻末に記載している「付録：RHA PowerShell サンプルスクリプト」についても同様です。

1. このドキュメントは Microsoft Windows PowerShell (以降、Windows PowerShell と表記)についてある程度知識があることを前提に記述しています。Windows PowerShell の詳細についてはマイクロソフト社のウェブサイトなどで事前にご確認ください。
2. このドキュメントで紹介するスクリプトは全てサンプルです。利用の際には必ず事前検証を行い、希望する動作が正常に行われる事をご確認の上でご利用ください。
3. このドキュメントに記載されるスクリプトが、全ての環境で動作することは保障していません。
4. 改変・編集等は自由ですが、自己責任の上で実施してください。
5. このドキュメントに記載されるスクリプトを利用した事による直接あるいは間接的な損害も、著作者および Arcserve 社では一切の責任を負わないものとします。
6. RHA PowerShell の一部のコマンドは r16 で拡張されているため、r15 および r12.5 では利用することができないことがあります。
7. この利用規定は予告なく改編・加筆を行うことがあります。

第1章　RHA PowerShell について

1.1.　RHA PowerShell とは

RHA PowerShell は Windows PowerShell 1.0 もしくは 2.0 上でスナップインとして動作する、Arcserve RHA の管理用コマンドラインインターフェースです。

RHA PowerShell はシナリオをコントロールするためのコマンドを提供します。例えばシナリオの開始や停止、一時停止/再開、スイッチオーバーの実行、同期の実行などを行う際に利用します。ただし、必ずしも CA ARCserve RHA マネージャと同様の機能を提供するものではありませんのでご注意ください。RHA PowerShell で利用できる機能の詳細については製品マニュアル「CA ARCserve® Replication/High Availability PowerShell コマンド 操作ガイド」をご参照ください。

1.2.　Windows PowerShell とは

Windows PowerShell とはマイクロソフト社が開発した拡張可能なコマンドラインインターフェース（CLI）シェルおよびスクリプト言語です。オブジェクト指向に基づいて設計されており、.NET Framework 2.0 を基盤としています。Windows Server 2008 からは標準で搭載されるようになりました。バッチ・コマンドや WSH に比べ高度で利用しやすい機能を多く備えています。

1.3.　事前準備

RHA PowerShell を利用するには、以下の事前準備が必要です。

1)　Windows PowerShell のインストール

Windows Server 2008 SP2 以前の OS では Windows PowerShell を事前にインストールする必要があります。

Windows Server 2008 R2 以降の OS にはデフォルトで Windows PowerShell がインストールされているため、事前にインストール作業を行う必要はありません。

■ Windows Server 2003 の場合

Windows Server 2003 の場合、マイクロソフト社のウェブサイトより Windows PowerShell のインストーラをダウンロードしてください。尚、Windows PowerShell を利用するには別途 Microsoft .NET Framework 2.0 のインストールが必要です。 (下記のダウンロード URL は 2014 年 9 月 24 日現在のものです。)

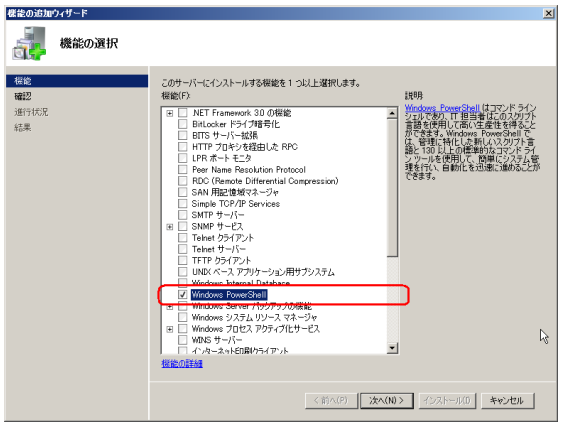
ダウンロードの詳細 ：Windows Server 2003 用 Windows PowerShell 1.0 インストール パッケージ (ローカライズ版) (KB926140)
http://www.microsoft.com/downloads/details.aspx?familyid=C61FB27B-E71C-4ECF-9D2C-9B299B149490&displaylang=ja#Requirements

ダウンロードの詳細 ：Windows Server 2003 x64 Edition 用 Windows PowerShell 1.0 インストール パッケージ (ローカライズ版) (KB926140)
http://www.microsoft.com/downloads/details.aspx?familyid=22E607F4-F854-497F-9548-770477E4B71D&displaylang=ja

■ Windows Server 2008 S1/SP2 の場合

Windows Server 2008 SP1/SP2 には Windows PowerShell 1.0 が組み込まれていますが、利用するにはインストール作業が必要です。[サーバマネージャ]の[機能の追加]よりインストールします。

例）Windows Server 2008 SP2 に Windows PowerShell をインストールする場合



2)　Arcserve RHA のインストール

RHA PowerShell を利用して Arcserve RHA の管理をするには、事前に CA ARCserve RHA コントロールサービス (以降、コントロールサービスと表記) および CA ARCserve RHA エンジン (以降、エンジンと表記) のインストールを済ませておく必要があります。コントロールサービスおよびエンジンのインストールについては製品マニュアル「CA ARCserve Replication/High Availability インストールガイド」をご参照ください。

3)　スクリプト実行権限の設定 (任意)

スクリプトを自動実行させるには権限の設定が必要な場合があります。以下の方法により環境で設定されている実行権限を確認し、必要に応じて変更してください。スクリプトを独自で開発して実行する場合には RemoteSigned がお勧めです。権限の詳しい説明はマイクロソフト社の Web サイトなどを確認してください。

実行権限の確認方法

PS > Get-ExecutionPolicy

実行権限の設定方法（例）

PS > Set-ExecutionPolicy RemoteSigned

なお、このドキュメントでは上記の権限設定を行わない場合でも実行できる方法を記載します。

1.4. 構成イメージ

RHA PowerShell はコントロールサービスに接続して命令を実行します。そのため、RHA PowerShell はコントロールサービスと通信できるコンピュータにインストールしてご利用ください。

以下は RHA PowerShell のコマンドがエンジンに渡り実行されるまでのイメージです。

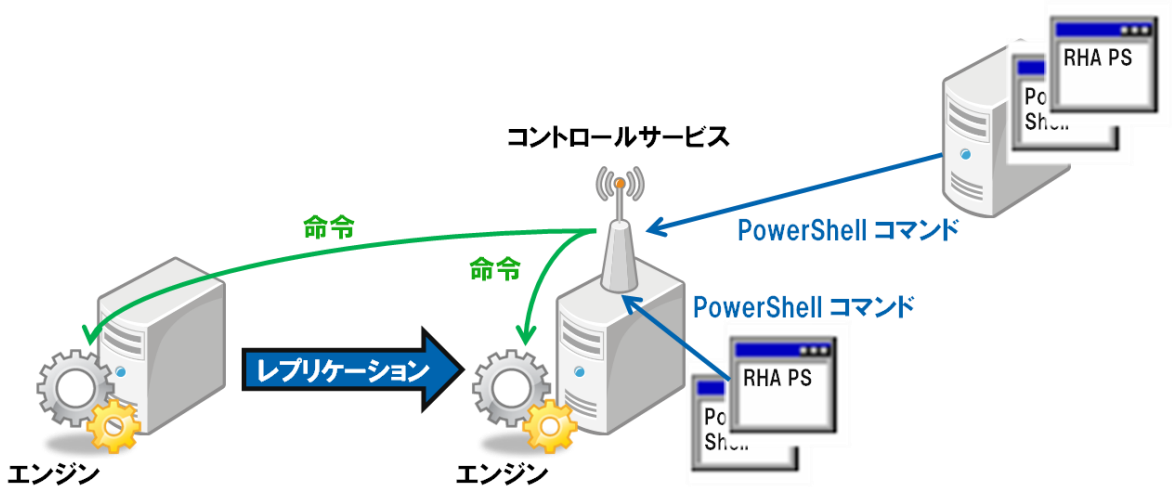


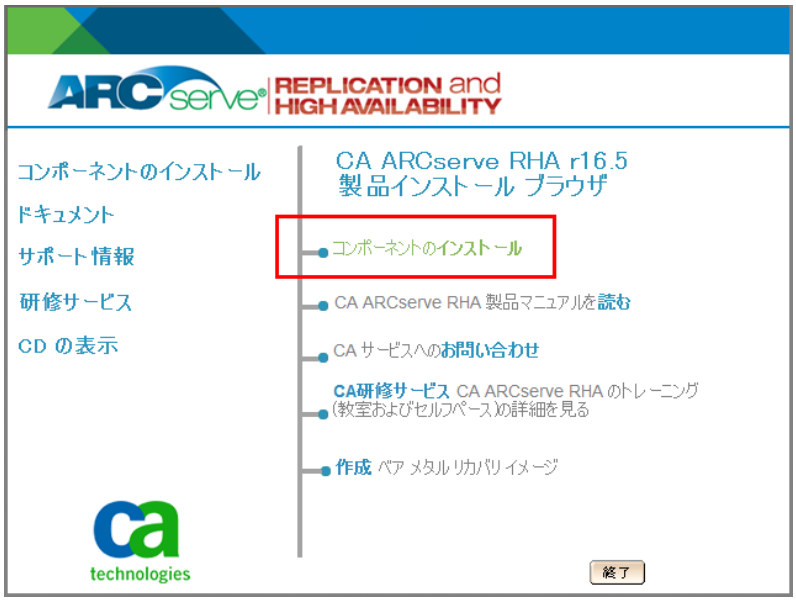
図 1： RHA PowerShell コマンド実行イメージ

このドキュメントではコントロールサービスがレプリカサーバに導入されている場合を想定して記載しますが、コントロールサービスがエンジンとは別のサーバに導入されている場合も RHA PowerShell コマンドの実行方法は同じです。

第2章 RHA PowerShell のインストール

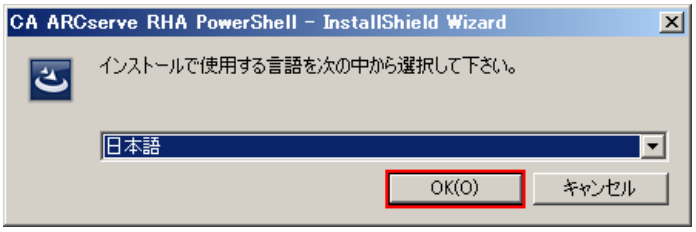
RHA PowerShell を以下の手順に従いインストールします。

Step1: RHA PowerShell をインストールするコンピュータに、Administrator または Administrators グループのユーザでログオンし、「CA ARCserve Replication and High Availability r16.5 メディア」をドライブにセットすると、インストーラ画面が自動的に起動します。起動しない場合は、エクスプローラより、メディアドライブのルート ディレクトリにある [setup.exe] を実行してください。「コンポーネントのインストール」をクリックします。



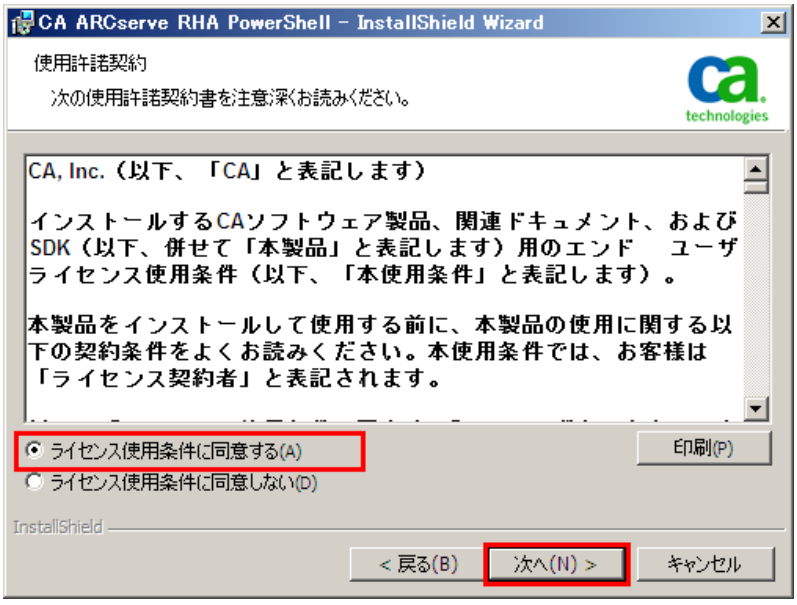
Step2: 「CA ARCserve RHA PowerShell のインストール」をクリックします。

Step3: [日本語]を選択し、[OK]をクリックします。

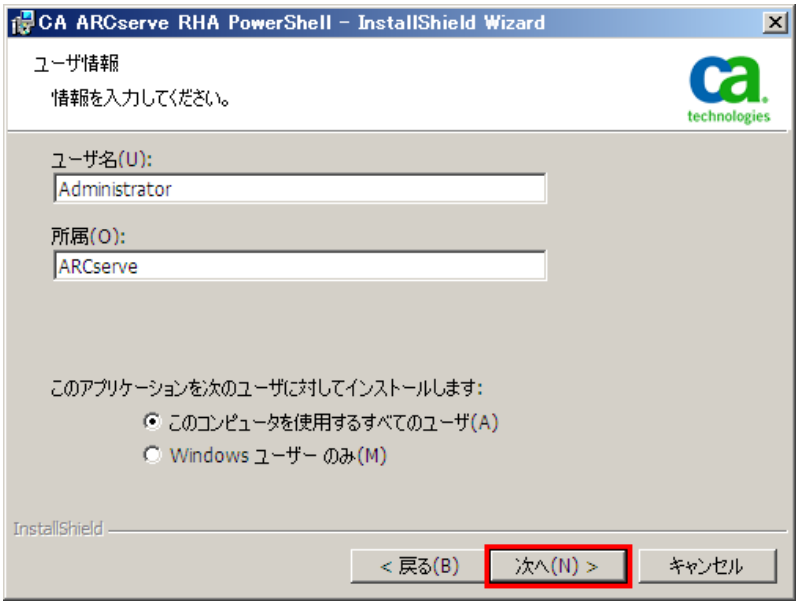


Step4: [次へ]をクリックします

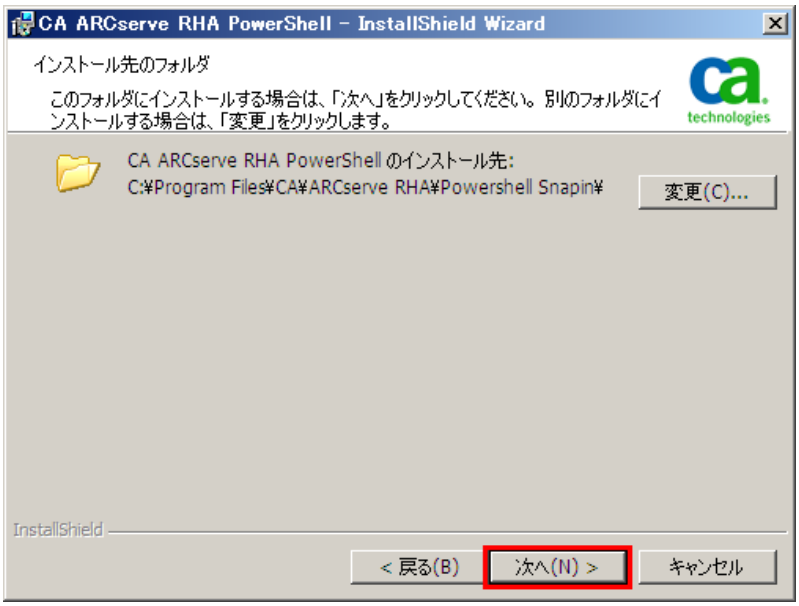
Step5: 使用許諾契約を最後まで読み、同意する場合は[ライセンス使用条件に同意する]を選択し、[次へ]をクリックします。



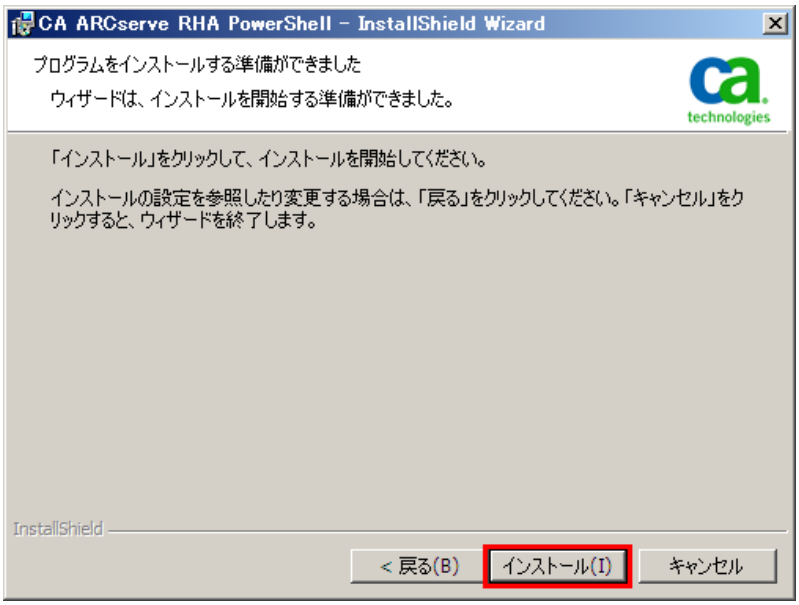
Step6: ユーザ名と所属を入力し、[次へ]をクリックします。



Step7: インストール先のフォルダを確認し、問題がなければ[次へ]をクリックします。
※64 ビット環境にインストールした場合にはデフォルトインストールパスは以下になります。
C:¥Program Files(x86)¥CA¥ARCserve RHA¥PowerShell Snapin¥

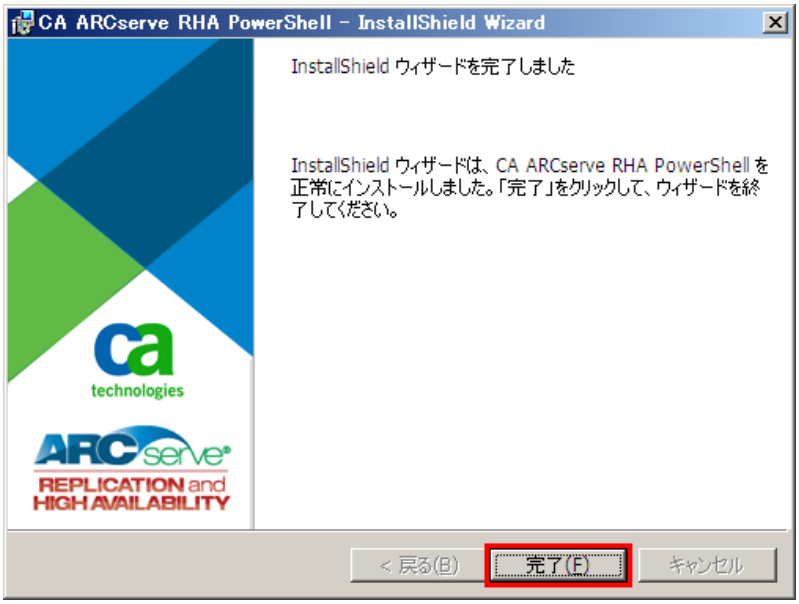


Step8: インストールをクリックします。

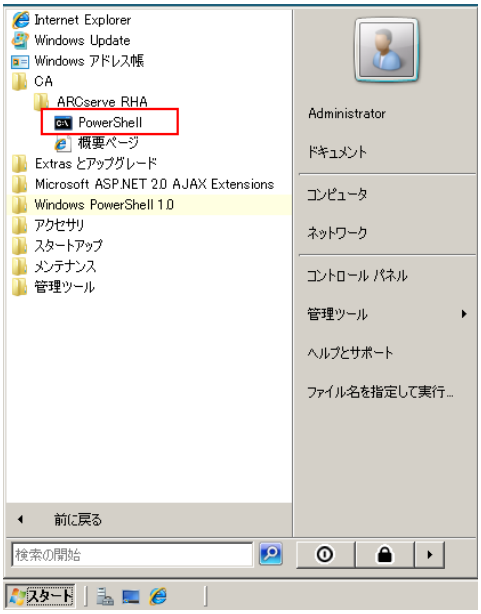


インストール完了までしばらくお待ちください。

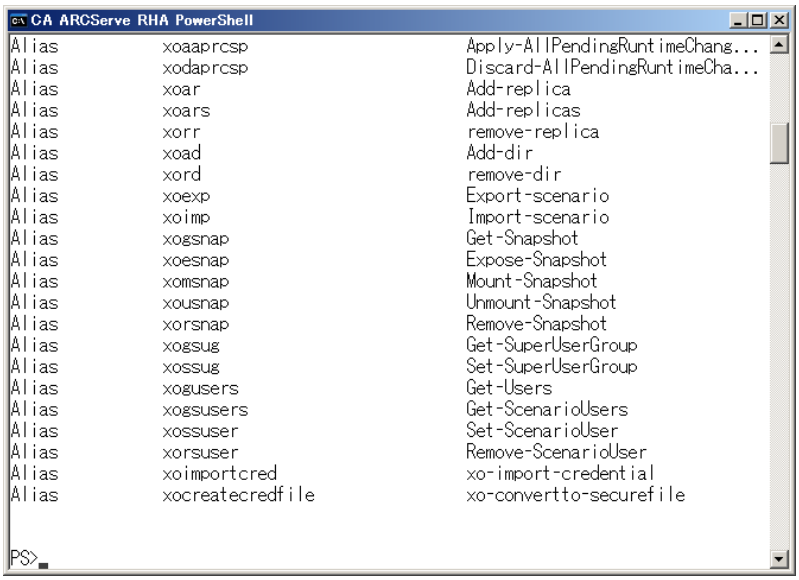
Step9: [完了]をクリックし、InstallShield Wizard 画面を閉じます。



Step10: Windows スタート メニューの[すべてのプログラム] - [CA] - [ARCserve RHA] - [PowerShell]を開きます。



Step11: RHA PowerShell のコンソール画面が表示され、特にエラーなく最後に「 PS> 」と表示されることを確認ください。



以上で RHA PowerShell のインストールは完了です。

第3章 RHA PowerShell スクリプト実行手順

RHA PowerShell スクリプトを実行してシナリオを操作･制御するには以下 3 つの手順を実行する必要があります。

1. パスワードファイルの作成
   Windows PowerShell の考え方にセキュリティ面への配慮からスクリプト中に直接パスワードを書き込まない、という暗黙のルールがあります。RHA PowerShell からコントロールサービスに接続する際にユーザ認証の操作やパスワード入力をせずに実行するため、接続に利用するユーザアカウント用に暗号化されたパスワードファイルを作成します。

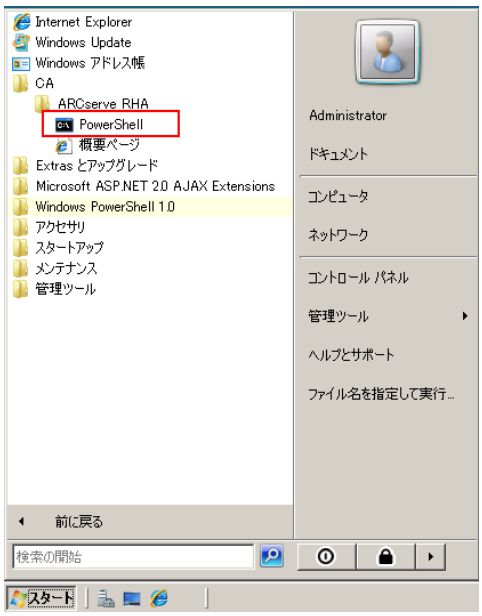
2. RHA PowerShell スクリプトファイルの作成
   RHA PowerShell のコマンドを利用したスクリプトファイルを作成します。RHA PowerShell コマンドの詳しい利用方法については、「CA ARCserve® Replication/High Availability PowerShell コマンド 操作ガイド r16.5」をご覧ください。

3. RHA PowerShell スクリプトを呼び出すためのバッチスクリプトの作成
   PowerShell スクリプトはダブルクリックするだけでは実行できません。作成した RHA PowerShell スクリプトを Windows 環境で呼び出すためのバッチスクリプトを作成します。Windows タスクスケジューラや他製品などから実行する場合などはこのバッチスクリプトを指定します。

3.1. パスワードファイルの作成

RHA PowerShell をインストールした環境で以下の手順を実行します。

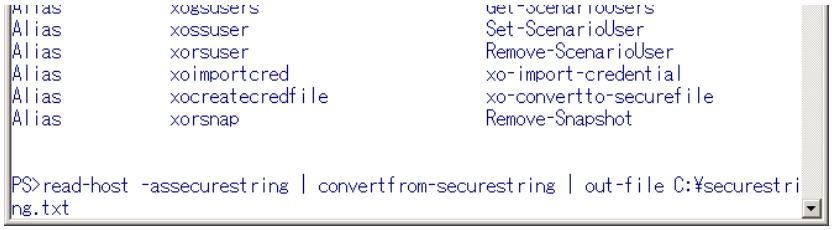
Step1: スタートメニューより[すべてのプログラム] – [CA] – [ARCserve RHA] – [PowerShell] を開きます



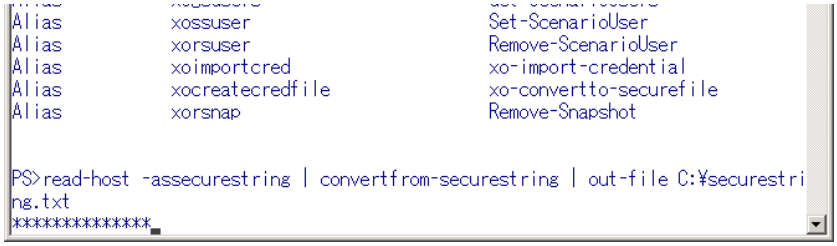
Step2: 以下のコマンドを実行します。

PS> read-host -assecurestring | convertfrom-securestring | out-file C:¥securestring.txt

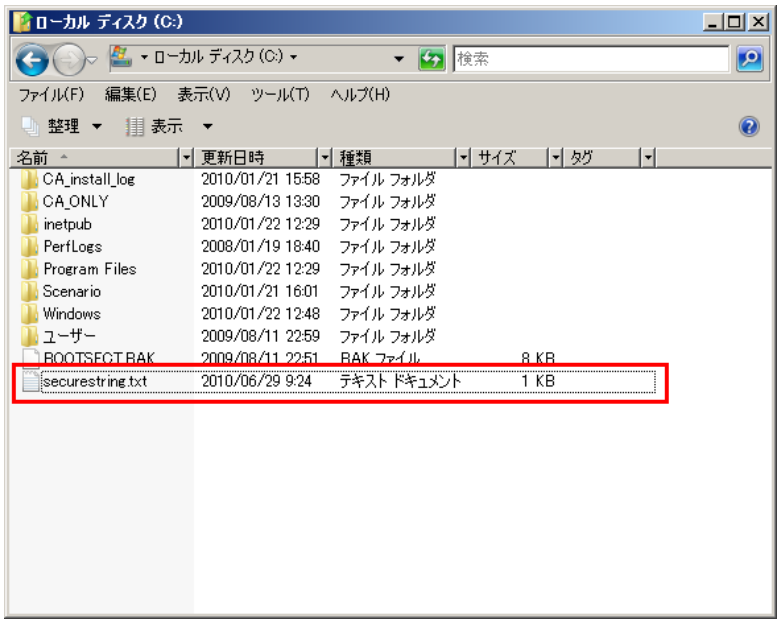
このコマンド入力後エンターを入力すると、文字入力の待ち状態となりカーソルが 1 行下に移動します。



Step3: コントロールサービスに接続するユーザアカウントのパスワードを入力します。パスワードは全てアスタリスク(＊)で表示されます。入力が終わったらエンターを入力します。



Step4: Cドライブ直下に securestring.txt というファイルが作成されていることを確認してください。
※ このファイルは移動しても名前を変更しても構いませんが、中身は編集しないでください。



【重要】パスワードファイルはスクリプト実行に必要なファイルです。RHA PowerShell スクリプトを実行するマシンを移動する場合には、パスワードファイルを移動した先のサーバにて再作成してください。

3.2. RHA PowerShell スクリプトファイルの作成

RHA PowerShell スクリプトファイルを作成します。

Step1: テキストファイルを任意の場所に作成し、以下のコマンドを書き込みます。

```
1: add-pssnapin XOPowerShell
2: $pass = cat C:¥securestring.txt | convertto-securestring
3: $mycred = new-object -typename System.Management.Automation.PSCredential
   -Argumentlist "<ドメイン名>¥<ユーザ名>",$pass
4: connect-xo -host < コントロールサービス導入サーバのホスト名または IP アドレス> $mycred -protocol
   <HTTP もしくは HTTPS>
```

注 1:上記サンプルの左側の数字は段落番号です。スクリプトを記述する際は無視してください。

注 2: Workgroup 環境で実行する場合にはドメイン名の代わりにコントロールサービス導入サーバのホスト名を入力します。

注 3:斜体文字は環境に合わせて適宜変更してください。

Step2: このスクリプトに続けてシナリオ操作を行う RHA PowerShell コマンドを記述します。

例)「FileServer」という名前のシナリオを実行する

Run-scenario –Name “FileServer” **–Mode** F **–Ignore** 1

Step3: コントロールサービスから切断する Disconnect-XO コマンドを追加して内容を保存し、このファイルの拡張子を「.ps1」に変更します。

スクリプトファイルの例： run_scenario.ps1

```
1: add-pssnapin XOPowerShell
2: $pass = cat C:¥securestring.txt | convertto-securestring
3: $mycred = new-object -typename System.Management.Automation.PSCredential
   -Argumentlist "replica¥Administrator",$pass
4: connect-xo -host replica $mycred -protocol http
5: Run-scenario –Name “FileServer” –Mode F –Ignore 1
6: Disconnect-XO
```

3.3. RHA PowerShell スクリプトを呼び出すためのバッチスクリプトの作成

RHA PowerShell スクリプトファイルを呼び出すバッチスクリプトを作成します。

Step1:　以下のコマンドを環境で 1 度実行し、スクリプトが正常に実行できるかを確認してください。引数の最後に、作成した PowerShell スクリプトファイルを直接パスで指定します。

■ 32 ビット システム上での実行

C:¥Windows¥system32¥WindowsPowerShell¥v1.0¥powershell.exe –Noninteractive –command set-executionpolicy RemoteSigned; <.ps1 ファイルへの直接パス>

■ 64 ビット システム上での実行

C:¥Windows¥SysWOW64¥WindowsPowerShell¥v1.0¥powershell.exe –Noninteractive –command set-executionpolicy RemoteSigned; <.ps1 ファイルへの直接パス>

例) run_scenario.ps1 ファイルを実行する場合

C:¥Windows¥system32¥WindowsPowerShell¥v1.0¥powershell.exe –Noninteractive –command set-executionpolicy RemoteSigned; C:¥script¥powershell¥run_scenario.ps1

Step2:　Step1 のコマンドをテキストファイルに保存します。その際、スクリプトのパス以外の引数はすべて削除します。（「- Noninteractive –command set-executionpolicy RemoteSigned;」を削除）

例：32 ビット システム上で実行する場合のコマンド

C:¥Windows¥system32¥WindowsPowerShell¥v1.0¥powershell.exe <.ps1 ファイルへの直接パス>

Step3:　ファイルの拡張子を.bat などに変更します。

以上で RHA PowerShell スクリプトを実行するために必要なファイルが作成できました。必要に応じて作成したバッチスクリプトを実行するアプリケーションや Windows タスクスケジューラに登録します。

付録： RHA PowerShell サンプルスクリプト

RHA PowerShell サンプルスクリプトです。ここでご紹介するサンプルは全て以下の設定を元に記載しています。環境に応じて適宜変更してください。

注： スクリプトの左側の番号は段落番号です。スクリプト利用時には必要ありませんので削除してください。

【想定環境】

1. 環境： Workgroup
2. マスタサーバ名： Master
3. レプリカサーバ名： Replica
4. コントロールサービス導入サーバ名： Replica
5. コントロールサービス接続プロトコル： http
6. コントロールサービス接続ユーザ： Administrator
7. パスワードファイルパス： C:¥securestring.txt
8. シナリオ名： ファイルサーバ

指定したシナリオを開始する

run_scenario.ps1

```
1: add-pssnapin XOPowerShell
2: $pass = cat C:¥securestring.txt | convertto-securestring
3: $mycred = new-object -typename System.Management.Automation.PSCredential
   -Argumentlist "Replica¥Administrator",$pass
4: Connect-XO -Host Replica $mycred -Protocol http
5: Run-scenario –Name “ファイルサーバ” –Mode F –Ignore 1
6: Disconnect-XO
```

指定したシナリオを一時停止する

suspend.ps1

```
1: add-pssnapin XOPowerShell
2: $pass = cat C:¥securestring.txt | convertto-securestring
3: $mycred = new-object -typename System.Management.Automation.PSCredential
   -Argumentlist "Replica¥Administrator",$pass
4: Connect-XO -Host Replica $mycred -protocol http
5: Suspend-Scenario –Name “ファイルサーバ” –Host Replica
6: Disconnect-XO
```

一時停止されたシナリオを再開する

resume.ps1

```
1: add-pssnapin XOPowerShell
2: $pass = cat C:¥securestring.txt | convertto-securestring
3: $mycred = new-object -typename System.Management.Automation.PSCredential
   -Argumentlist "Replica¥Administrator",$pass
4: Connect-XO -Host Replica $mycred -protocol http
5: Resume-Scenario –Name “ファイルサーバ” –Host Replica
6: Disconnect-XO
```

指定したシナリオの同期処理を実行する（ファイルレベル同期、同一サイズ/タイムスタンプのファイルを無視）

sync_scenario.ps1

```
1: add-pssnapin XOPowerShell
2: $pass = cat C:¥securestring.txt | convertto-securestring
3: $mycred = new-object -typename System.Management.Automation.PSCredential
   -Argumentlist "Replica¥Administrator",$pass
4: Connect-XO -Host Replica $mycred -protocol http
5: Sync-Scenario –Name “ファイルサーバ” –Mode F –Ignore 1
6: Disconnect-XO
```

指定したシナリオを停止する

stop_scenario.ps1

```
1: add-pssnapin XOPowerShell
2: $pass = cat C:¥securestring.txt | convertto-securestring
3: $mycred = new-object -typename System.Management.Automation.PSCredential
   -Argumentlist "Replica¥Administrator",$pass
4: Connect-XO -Host Replica $mycred -protocol http
5: Stop-Scenario –Name “ファイルサーバ”
6: Disconnect-XO
```

ホストメンテナンスを利用してマスタサーバの再起動の準備をする

reboot_master.ps1

```
1:add-pssnapin XOPowerShell
2:$pass = cat C:¥securestring.txt | convertto-securestring
3:$mycred = new-object -typename System.Management.Automation.PSCredential -Argumentlist
"Replica¥Administrator",$pass
4:Connect-XO -host Replica $mycred –protocol http
5:Prepare-Reboot Master
6:$i = 0
7:Start-Sleep –s 60
8:Do {$i = $i + 1;if(get-events "ファイルサーバ" |Select -last $i |Select ID |Select-String "SR00393"
-quiet){$i = 0;start-sleep -s 60} } Until (get-events "ファイルサーバ" |Select -last $i |Select ID
|Select-String "IR00556" -quiet)
9:Write-Host “再起動の準備ができました”
10:Disconnect-XO
```

注：６ ～ ８行目は再起動の準備が整ったことをイベントログから確認しています。