

CA Client Automation

Software Delivery CLI リファレンス ガイド

12.9



このドキュメント（組み込みヘルプシステムおよび電子的に配布される資料を含む、以下「本ドキュメント」）は、お客様への情報提供のみを目的としたもので、日本 CA 株式会社（以下「CA」）により随時、変更または撤回されることがあります。本ドキュメントは、CA が知的財産権を有する機密情報であり、CA の事前の書面による承諾を受けずに本書の全部または一部を複写、譲渡、変更、開示、修正、複製することはできません。

本ドキュメントで言及されている CA ソフトウェア製品のライセンスを受けたユーザは、社内でユーザおよび従業員が使用する場合に限り、当該ソフトウェアに関連する本ドキュメントのコピーを妥当な部数だけ作成できます。ただし、CA のすべての著作権表示およびその説明を当該複製に添付することを条件とします。

本ドキュメントを印刷するまたはコピーを作成する上記の権利は、当該ソフトウェアのライセンスが完全に有効となっている期間内に限定されます。いかなる理由であれ、上記のライセンスが終了した場合には、お客様は本ドキュメントの全部または一部と、それらを複製したコピーのすべてを破棄したことを、CA に文書で証明する責任を負います。

準拠法により認められる限り、CA は本ドキュメントを現状有姿のまま提供し、商品性、特定の使用目的に対する適合性、他者の権利に対して侵害のないことについて、黙示の保証も含めいかなる保証もしません。また、本ドキュメントの使用に起因して、逸失利益、投資損失、業務の中断、営業権の喪失、情報の喪失等、いかなる損害（直接損害か間接損害かを問いません）が発生しても、CA はお客様または第三者に対し責任を負いません。CA がかかる損害の発生の可能性について事前に明示に通告されていた場合も同様とします。

本ドキュメントで参照されているすべてのソフトウェア製品の使用には、該当するライセンス契約が適用され、当該ライセンス契約はこの通知の条件によっていかなる変更も行われません。

本書の制作者は CA および CA Inc. です。

「制限された権利」のもとでの提供：アメリカ合衆国政府が使用、複製、開示する場合は、FAR Sections 12.212、52.227-14 及び 52.227-19(c)(1)及び(2)、ならびに DFARS Section 252.227-7014(b)(3) または、これらの後継の条項に規定される該当する制限に従うものとします。

Copyright © 2014 CA. All rights reserved. 本書に記載されたすべての商標、商号、サービス・マークおよびロゴは、それぞれの各社に帰属します。

CA Technologies 製品リファレンス

このマニュアルセットで参照されている CA 製品は、以下のとおりです。

- CA Advantage® Data Transport® (CA Data Transport)
- CA Asset Intelligence
- CA APM (CA Asset Portfolio Management)
- CA Business Intelligence
- CA Common Services™
- CA DMM (CA Desktop Migration Manager)
- CA Embedded Entitlements Manager (CA EEM)
- CA MDM (CA Mobile Device Management)
- CA NSM (CA Network and Systems Management)
- CA Patch Manager
- CA Process Automation
- CA Service Desk Manager
- CA WorldView™

CA への連絡先

テクニカルサポートの詳細については、弊社テクニカルサポートの Web サイト (<http://www.ca.com/jp/support/>) をご覧ください。

目次

第 1 章：概要	7
----------	---

第 2 章：sd_acmd -- エージェント管理コマンド	9
-------------------------------	---

AddActivateRecord -- ソフトウェア パッケージの起動.....	11
AddConfigureRecord - ソフトウェア パッケージの設定.....	13
AddDetectedRecord -- ソフトウェア パッケージの検出の Software Delivery への通知.....	15
AddInstallRecord -- ソフトウェア パッケージのインストールの Software Delivery への通知.....	17
AddLibDeliveryRecord -- スケーラビリティ サーバ ステージング ライブラリへのソフトウェア パッケージの追加の Software Delivery への通知.....	19
AddLibRemovalRecord -- スケーラビリティ サーバ ステージング ライブラリからのソフトウェア パッケージの削除の Software Delivery への通知.....	21
AddReinstallRecord -- ソフトウェア パッケージの再インストール.....	23
AddUndetectedRecord -- ソフトウェア パッケージの削除の Software Delivery への通知.....	25
AddUninstallRecord -- ソフトウェア パッケージのアンインストールの Software Delivery への通知.....	26
Execute Container -- Software Delivery ジョブ コンテナ オーダ ファイルの実行.....	28
Encrypt--任意の文字列の暗号化.....	30
SecureContainer-- Software Delivery ジョブ コンテナ オーダ ファイルを暗号化.....	31
WaitContainers-- ジョブの終了を待機.....	31
ExitCodeMsg - 失敗したジョブのエラー メッセージを設定.....	32
JobCheck--ジョブ確認の実行.....	33
JobProgress-- 現行ジョブのプログレス メッセージ/パーセントを設定.....	35
ターゲットの削除 -- コンピュータからのターゲットの削除.....	36
Remove Win32 Program.....	37
SetDownloadMethod.....	38
Signal-- Software Delivery エージェントが実行するアクションの Software Delivery にシグナルを送信.....	39
ReinstallTarget--ターゲットに再インストールされるソフトウェア.....	40
UserInfo - コンピュータのターゲットにユーザ情報を設定.....	41
コンテナ オーダ ファイル (COF) フォーマット.....	43
COF Section Locale.....	51
COF Section Job.....	52
COF Section Container.....	57
COF Section Library.....	58

COF ファイルの例	59
COF Library セクションの例	60
sd_acmd リターン コード	60

第 3 章 : sd_msiexe.exe 63

sd_msiexe-- .msi パッケージのインストール/設定/アンインストールを実行.....	63
sd_msiexe -scan -- 現在のユーザ プロファイルに対して .msi パッケージを実行.....	64
sd_msiexe -admdel-- 指定した管理者がインストールしたファイルの管理インストールを削除	64
sd_msiexe -sourceupdate-- 現在のプロファイルの MSI ソース リストを更新.....	65
sd_msiexe -updatedictionary-- SDMSILIB デictionaryを更新.....	65
sd_msiexe -removemsilibrecord-- SDMSILIB デictionaryのレコードを削除	66
sd_msiexe -addmsilibrecord-- SDMSILIB デictionaryにレコードを追加	66
sd_msiexe ? -? /? help-- sd_msiexe コマンドの使用方法的表示	66

第 4 章 : sd_sscmd -- ステージング ライブラリ コマンド ライン管理 67

コマンド通知.....	68
長さ制限.....	69
stagecheck.....	70
bulkupdate	71
sd_sscmd libraryaccess	72
sd_sscmd importmsi	73
sd_sscmd addshare	73
sd_sscmd removeshare.....	74
sd_sscmd import	75
aregsw -- ステージング ライブラリでのソフトウェアの自動登録.....	75
dereg -- ステージング ライブラリでのソフトウェア項目の登録解除.....	77
verbose.....	77
batch	78
バッチ処理中のエラーを無視.....	79
リターン コード	80

第 1 章：概要

「*Software Delivery CLI* リファレンス」では、3つの異なるソフトウェア配信コマンドラインインターフェースについて詳細な参照情報を提供しています。

- `sd_acmd`
- `sd_msiexe`
- `sd_sscmd`

第 2 章 : sd_acmd -- エージェント管理コマンド

sd_acmd コマンドは、手動でインストールされたソフトウェアに関するレポートを作成します。

sd_acmd を使用するには、**Software Delivery** エージェント ソフトウェアがインストールされ、設定されている必要があります。

sd_acmd は、以下の用途のためにジョブ スクリプトから実行することができます。

- 再起動またはログオフの通知
- インストール レコードまたはアンインストール レコードの書き込み
- スケーラビリティ サーバ ステージング ライブラリの配信レコードの書き込みまたは削除
- ソフトウェア検出レコードの書き込みまたは削除
- Software Delivery ジョブ コンテナ オーダ ファイルの保護および実行

構文

```
sd_acmd command command_arguments [arguments]
```

注 :

- オフラインでの再インストールでは、ReinstallTarget、Encrypt、および JobCheck コマンドを使用できます。
- 引数を省略する場合は、構文に「"」と入力して空の引数を指定します。

このセクションには、以下のトピックが含まれています。

[AddActivateRecord -- ソフトウェア パッケージの起動 \(P. 11\)](#)

[AddConfigureRecord - ソフトウェア パッケージの設定 \(P. 13\)](#)

[AddDetectedRecord -- ソフトウェア パッケージの検出の Software Delivery への通知 \(P. 15\)](#)

[AddInstallRecord -- ソフトウェア パッケージのインストールの Software Delivery への通知 \(P. 17\)](#)

[AddLibDeliveryRecord -- スケーラビリティ サーバステージング ライブラリへのソフトウェア パッケージの追加の Software Delivery への通知 \(P. 19\)](#)

[AddLibRemovalRecord -- スケーラビリティ サーバステージング ライブラリからのソフトウェア パッケージの削除の Software Delivery への通知 \(P. 21\)](#)

[AddReinstallRecord -- ソフトウェア パッケージの再インストール \(P. 23\)](#)

[AddUndetectedRecord -- ソフトウェア パッケージの削除の Software Delivery への通知 \(P. 25\)](#)

[AddUninstallRecord -- ソフトウェア パッケージのアンインストールの Software Delivery への通知 \(P. 26\)](#)

[Execute Container -- Software Delivery ジョブ コンテナ オーダ ファイルの実行 \(P. 28\)](#)

[Encrypt--任意の文字列の暗号化 \(P. 30\)](#)

[SecureContainer-- Software Delivery ジョブ コンテナ オーダ ファイルを暗号化 \(P. 31\)](#)

[WaitContainers-- ジョブの終了を待機 \(P. 31\)](#)

[ExitCodeMsg - 失敗したジョブのエラー メッセージを設定 \(P. 32\)](#)

[JobCheck--ジョブ確認の実行 \(P. 33\)](#)

[JobProgress-- 現行ジョブのプログレス メッセージ/パーセントを設定 \(P. 35\)](#)

[ターゲットの削除 -- コンピュータからのターゲットの削除 \(P. 36\)](#)

[Remove Win32 Program \(P. 37\)](#)

[SetDownloadMethod \(P. 38\)](#)

[Signal-- Software Delivery エージェントが実行するアクションの Software Delivery にシグナルを送信 \(P. 39\)](#)

[ReinstallTarget--ターゲットに再インストールされるソフトウェア \(P. 40\)](#)

[UserInfo - コンピュータのターゲットにユーザ情報を設定 \(P. 41\)](#)

[コンテナ オーダ ファイル \(COF\) フォーマット \(P. 43\)](#)

[COF Library セクションの例 \(P. 60\)](#)

[sd acmd リターン コード \(P. 60\)](#)

AddActivateRecord -- ソフトウェア パッケージの起動

このコマンドは、ソフトウェア パッケージが起動されたことを **Software Delivery** に対して通知するレコードを追加します。

構文

```
AddActivateRecord item
                        version
                        procedure
                        installprocedure
                        date
                        time
                        orderedby
                        comment
                        [target=<target>]
```

item

ソフトウェア パッケージの名前です。

version

ソフトウェア パッケージのバージョンです。

procedure

空の文字列（将来使用するために予約されています）。

installprocedure

使用されるインストール プロシージャの名前です。空の文字列を使用すると、任意のインストール プロシージャを指定することができます。アスタリスク（*）を使用すると、任意のインストール プロシージャを使用して行われたすべてのインストールを指定することができます。

date

インストールの日付を指定します。キーワード **current** を使用することも使用できます。
日付の形式は、YYYY-MM-DD です。

time

インストールの時間を指定します。キーワード **current** を使用することも使用できます。
時間の形式は、HH:MM です。

orderedby

インストールを注文したユーザ ID を指定します。

comment

コメントです。

[target=<target>]

コマンドがコンピュータ ユニットに適用可能か（デフォルト）、または同じコンピュータ上の任意の有効なまたは既存のユーザ プロファイルユニットに適用可能かどうかを指定します。以下のターゲットを除いて、任意のターゲットを指定できます。

- コマンドを実行しているコンピュータ上。
- コマンドを上に行っているのと同じコンピュータ上のユーザ プロファイルから。

例

以下の例では、プロシージャ **Activate1** とインストール プロシージャ **Install** を使用したソフトウェア項目 **Product X** バージョン **1.1** に関する起動レコードを追加するように、**Software Delivery** に通知します。インストールには、現在の日時が使用されます。**orderedby** は「**Automatic by Job**」、**comment** は「**Using Activate1 for Product X 1.1**」です。

```
sd_acmd addactivaterecord "Product X" "1.1" "Activate1" "Install" current current  
"Automatic by Job" "Using Activate1 for Product X 1.1"
```

起動レコードをインストールに関連付けることができない場合、コマンドの結果は保存されません。

AddConfigureRecord - ソフトウェア パッケージの設定

このコマンドは、ソフトウェア パッケージが設定されたことを **Software Delivery** に対して通知するレコードを追加します。設定レコードを関連付けることができるインストールがない場合、コマンドの結果は保存されません。

構文

```
AddConfigureRecord item
    version
    procedure
    installprocedure
    date | current
    time | current
    orderedby
    comment
    [target="targetname"]
```

item

ソフトウェア パッケージの名前です。

version

ソフトウェア パッケージのバージョンです。

procedure

空の文字列（将来使用するために予約されています）。

installprocedure

使用されるインストール プロシージャの名前です。空の文字列を使用すると、任意のインストール プロシージャを指定することができます。アスタリスク（*）を使用すると、任意のインストール プロシージャを使用して行われたすべてのインストールを指定することができます。

date

インストールの日付を指定します。キーワード **current** を使用することも使用できます。

日付の形式は、YYYY-MM-DD です。

time

インストールの時間を指定します。キーワード **current** を使用することも使用できます。

時間の形式は、HH:MM です。

orderedby

このオプションパラメータでは、アカウント名（例：DomainX¥UserY）を指定します。

デフォルト：SecureContainer コマンドを実行している現在のユーザ。

comment

コメントです。

[target="targetname"]

コマンドがコンピュータ ユニットに適用可能か（デフォルト）、または同じコンピュータ上の任意の有効なまたは既存のユーザ プロファイル ユニットに適用可能かどうかを指定します。以下のターゲットを除いて、任意のターゲットを指定できます。

- コマンドを実行しているコンピュータ上。
- コマンドを上を実行しているのと同じコンピュータ上のユーザ プロファイルから。

例：

プロシージャ **Configure1** とインストール プロシージャ **Install** を使用したソフトウェア項目 **Product X** バージョン **1.1** に関する設定レコードを追加するように、**Software Delivery** に通知します。インストールには、現在の日時が使用されます。**orderedby** は「Automatic by Job」、**comment** は「Using Configure1 for Product X 1.1」です。

```
sd_acmd addconfigurerecord "Product X" "1.1" "Configure1" "Install" current current  
"Automatic by Job" "Using Configure1 for Product X 1.1"
```

AddDetectedRecord -- ソフトウェア パッケージの検出の Software Delivery への通知

このコマンドは、ソフトウェア パッケージが検出されたことを **Software Delivery** に対して通知するレコードを追加します。インストール済みパッケージにはライブラリ内にソース ファイルがありますが、検出されたパッケージにはそのようなファイルはありません。

構文

```
AddDetectedRecord item
                        version
                        supplier
                        comment
                        [uninstall template]
                        [uninstall file]
                        [uninstall params]
                        [target="targetname"]
```

item

ソフトウェア パッケージの名前です。

version

ソフトウェア パッケージのバージョンです。

supplier

供給元の名前です。

comment

コメントです。

uninstall_template

テンプレート タイプです。たとえば、MSI パッケージの場合は *msidetup* となります。

uninstall_file

アンインストール ファイルの名前です。たとえば、MSI パッケージの場合は *x.msi* となります。

uninstall_params

アンインストール パラメータです。たとえば、MSI パッケージの場合は `/u $joid /x productGUID /lemo $rf /qn` となります。

マクロ `$joid` は、ジョブ オブジェクト識別子に展開されます。

マクロ `$rf` は、出力の表示を可能にする結果ファイルを作成します。

`x` パラメータは、アンインストールが必要であることを示します。

`qn` パラメータは、実行がサイレントに（ユーザ インターフェースを使用せずに）行われることを示します。

[target="targetname"]

コマンドがコンピュータ ユニットに適用可能か（デフォルト）、または同じコンピュータ上の任意の有効なまたは既存のユーザ プロファイル ユニットに適用可能かどうかを指定します。以下のターゲットを除いて、任意のターゲットを指定できます。

- コマンドを実行しているコンピュータ上。
- コマンドを上を実行しているのと同じコンピュータ上のユーザ プロファイルから。

例

たとえば、以下のようにします。

```
sd_acmd adddetectedrecord "Product V" "3.1" "Bestseller Inc." "Detection of Product V v3.1" "" "" ""
```

検出されたソフトウェア項目 **Product V** バージョン **3.1** のレコードを追加するように、**Software Delivery** に通知します。 **Supplier** は **Bestseller Inc.**、**comment** は「**Detection of Product V v3.1**」となります。

SXP、PIF、PKG、および RPM パッケージでは、**AddDetectedRecord** が発行され、ソフトウェア ライブラリにその項目が登録されている場合、検出レコードがプロシージャ **Install Package** を持つインストール レコードに自動的に変換されます。

AddInstallRecord -- ソフトウェア パッケージのインストールの Software Delivery への通知

このコマンドは、ソフトウェア パッケージがインストールされたことを Software Delivery に対して通知するレコードを追加します。AddInstallRecord が発行され、ソフトウェア ライブラリに項目または項目プロシージャが見つからない場合は、インストール レコードがプロシージャ **Detected** を持つ検出レコードに自動的に変換されます。

構文

```
AddInstallRecord item
    version
    procedure
    date | current
    time | current
    orderedby
    comment
    [target="targetname"]
```

item

ソフトウェア パッケージの名前です。

version

ソフトウェア パッケージのバージョンです。

procedure

空の文字列（将来使用するために予約されています）。

date

インストールの日付を指定します。キーワード **current** を使用することも使用できます。

日付の形式は、YYYY-MM-DD です。

time

インストールの時間を指定します。キーワード **current** を使用することも使用できます。

時間の形式は、HH:MM です。

orderedby

このオプション パラメータでは、アカウント名（例：DomainX¥UserY）を指定します。

デフォルト：SecureContainer コマンドを実行している現在のユーザ。

comment

コメントです。

[target="targetname"]

コマンドがコンピュータ ユニットに適用可能か（デフォルト）、または同じコンピュータ上の任意の有効なまたは既存のユーザ プロファイル ユニットに適用可能かどうかを指定します。以下のターゲットを除いて、任意のターゲットを指定できます。

- コマンドを実行しているコンピュータ上。
- コマンドを上に行っているのと同じコンピュータ上のユーザ プロファイルから。

例：

以下の例では、プロシージャ **Install** を使用したソフトウェア項目 **Product X** バージョン **1.1** に関するインストール レコードを追加するように、**Software Delivery** に通知します。インストールには、現在の日時が使用されます。**orderedby** は「Automatic by Job」、**comment** は「Required by Product Y v2.0 install」です。

```
sd_acmd addinstallrecord "Product X" "1.1" "Install" current current "Automatic by Job" "Required by Product Y v2.0 install"
```

AddLibDeliveryRecord -- スケーラビリティ サーバ ステージング ライブラリへのソフトウェア パッケージの追加の Software Delivery への通知

このコマンドは、ソフトウェア パッケージがスケーラビリティ サーバ ステージング ライブラリに追加されたことを **Software Delivery** に対して通知するレコードを追加します。

構文

```
AddLibDeliveryRecord item
                        version
                        procedure
                        date | current
                        time | current
                        orderedby
                        comment
                        [target="targetname"]
```

item

ソフトウェア パッケージの名前です。

version

ソフトウェア パッケージのバージョンです。

procedure

空の文字列（将来使用するために予約されています）。

date

インストールの日付を指定します。キーワード **current** を使用することも使用できます。

日付の形式は、YYYY-MM-DD です。

time

この必須パラメータは、現在のジョブ コンテナの実行プログラム ジョブのプログレス メッセージを設定します。

orderedby

インストールを注文したユーザ ID を指定します。

comment

コメントです。

[target="targetname"]

コマンドがコンピュータ ユニットに適用可能か（デフォルト）、または同じコンピュータ上の任意の有効なまたは既存のユーザ プロファイルユニットに適用可能かどうかを指定します。以下のターゲットを除いて、任意のターゲットを指定できます。

- コマンドを実行しているコンピュータ上。
- コマンドを上に行っているのと同じコンピュータ上のユーザ プロファイルから。

例：

以下の例では、ソフトウェア項目 **Product Z** バージョン **1.3** に関する配信レコードを追加するように、**Software Delivery** に通知します。配信には、現在の日時が使用されます。**orderby** は「**Gunnar**」、**comment** は「**Upgrade of Product Z v1.2**」です。

```
sd_acmd addlibdeliveryrecord "Product Z" "1.3" "" current current "Gunnar" "Upgrade of Product Z v1.2"
```

このコマンドがスケーラビリティ サーバで発行され、アップストリームのドメイン マネージャのライブラリにソフトウェア項目 **Product Z 1.3** が存在する場合、スケーラビリティ サーバのステージング ライブラリ フォルダにレコードが追加されます。ジョブ レコードもスケーラビリティ サーバの [ジョブ] フォルダに追加されます。

AddLibRemovalRecord -- スケーラビリティ サーバ ステージング ライブラリからのソフトウェア パッケージの削除の Software Delivery への通知

このコマンドは、ソフトウェア パッケージがスケーラビリティ サーバのステージング ライブラリから削除されたことを **Software Delivery** に通知するレコードを追加します。

構文

```
AddLibRemovalRecord item
                        version
                        procedure
                        date
                        time
                        orderedby
                        comment
                        [target="targetname"]
```

item

ソフトウェア パッケージの名前です。

version

ソフトウェア パッケージのバージョンです。

procedure

空の文字列（将来使用するために予約されています）。

date

インストールの日付を指定します。キーワード **current** を使用することも使用できます。

日付の形式は、YYYY-MM-DD です。

time

インストールの時間を指定します。キーワード **current** を使用することも使用できます。

時間の形式は、HH:MM です。

orderedby

このオプションパラメータでは、アカウント名（例：DomainX¥UserY）を指定します。

デフォルト：SecureContainer コマンドを実行している現在のユーザ。

comment

コメントです。

[target="targetname"]

コマンドがコンピュータ ユニットに適用可能か（デフォルト）、または同じコンピュータ上の任意の有効なまたは既存のユーザ プロファイル ユニットに適用可能かどうかを指定します。以下のターゲットを除いて、任意のターゲットを指定できます。

- コマンドを実行しているコンピュータ上。
- コマンドを上に行っているのと同じコンピュータ上のユーザ プロファイルから。

例：

たとえば、以下のようにします。

```
sd_acmd addlibremovalrecord "Product W" "1.1" "" current current "Admin" "Removal of Product W v1.1"
```

ソフトウェア項目 **Product W** バージョン **1.1** に関する配信レコードを削除するように、Software Delivery に通知します。現在の日時が使用されます。**orderedby** は「Admin」、**comment** は「Removal of Product W v1.1」です。

このコマンドがスケーラビリティ サーバで発行され、アップストリームのドメイン マネージャのライブラリにソフトウェア項目 **Product W 1.1** が存在し、スケーラビリティ サーバのステージング ライブラリ フォルダに **Product W 1.1** の配信レコードが存在する場合、その配信レコードが削除されます。また、ジョブ レコードがスケーラビリティ サーバの [ジョブ] フォルダに追加されます。

AddReinstallRecord -- ソフトウェア パッケージの再インストール

このコマンドは、コンピュータの [ジョブ] フォルダに 2 つのレコードを追加します。1 つは、検出されたアンインストール プロシーダを使用してソフトウェア パッケージが削除されたことを **Software Delivery** に通知する検出レコードで、もう 1 つは、ソフトウェア パッケージがインストールされたことを示すレコードです。インストール レコードは、**AddReinstallRecord** で提供される情報を使用して更新されます。

AddReinstallRecord が発行され、ソフトウェア ライブラリに項目または項目プロシーダが見つからない場合は、インストール レコードがプロシーダ **Detected** を持つ検出レコードに自動的に変換されます。

構文

```
AddInstallRecord item
    version
    procedure
    date | current
    time | current
    orderedby
    comment
    [target="targetname"]
```

item

ソフトウェア パッケージの名前です。

version

ソフトウェア パッケージのバージョンです。

procedure

空の文字列（将来使用するために予約されています）。

date

インストールの日付を指定します。キーワード **current** を使用することも使用できます。

日付の形式は、YYYY-MM-DD です。

time

インストールの時間を指定します。キーワード **current** を使用することも使用できます。

時間の形式は、HH:MM です。

orderedby

このオプション パラメータでは、アカウント名（例：DomainX¥UserY）を指定します。

デフォルト：SecureContainer コマンドを実行している現在のユーザ。

comment

コメントです。

[target="targetname"]

コマンドがコンピュータ ユニットに適用可能か（デフォルト）、または同じコンピュータ上の任意の有効なまたは既存のユーザ プロファイル ユニットに適用可能かどうかを指定します。以下のターゲットを除いて、任意のターゲットを指定できます。

- コマンドを実行しているコンピュータ上。
- コマンドを上を実行しているのと同じコンピュータ上のユーザ プロファイルから。

例

以下の例では、Software Delivery に対して、プロシージャ Install を使用したソフトウェア Product W Version 1.1 の再インストール レコードを追加するように通知します。現在の日時が使用されます。Ordered by は Automatic by Job、comment は Required by Product X v2.0 install となります。

```
sd_acmd addreinstallrecord "Product W" "1.1" "Install" current current "Automatic by Job" "Required by Product X v2.0 install"
```

AddUndetectedRecord -- ソフトウェア パッケージの削除の Software Delivery への通知

このコマンドは、以前検出されたソフトウェア パッケージが削除されたことを **Software Delivery** に対して通知するレコードを追加します。

構文

```
AddUndetectedRecord item
                        version
                        [target="targetname"]
```

item

ソフトウェア パッケージの名前です。

version

ソフトウェア パッケージのバージョンです。

[target="targetname"]

コマンドがコンピュータ ユニットに適用可能か（デフォルト）、または同じコンピュータ上の任意の有効なまたは既存のユーザ プロファイル ユニットに適用可能かどうかを指定します。以下のターゲットを除いて、任意のターゲットを指定できます。

- コマンドを実行しているコンピュータ上。
- コマンドを上を実行しているのと同じコンピュータ上のユーザ プロファイルから。

例：

```
sd_acmd addundetectedrecord "Product V" "3.1"
```

検出されたソフトウェア項目 **Product V** バージョン **3.1** に関するレコードを削除するように、**Software Delivery** に通知します。

AddUninstallRecord -- ソフトウェア パッケージのアンインストールの Software Delivery への通知

このコマンドは、以前インストールされたソフトウェア パッケージがアンインストールされたことを **Software Delivery** に対して通知するレコードを追加します。 アンインストール レコードを関連付けることができるインストールがない場合、コマンドの結果は保存されません。

構文

```
AddUninstallRecord item
                        version
                        procedure
                        date | current
                        time | current
                        orderedby
                        comment
                        [target="targetname"]
```

item

ソフトウェア パッケージの名前です。

version

ソフトウェア パッケージのバージョンです。

procedure

空の文字列（将来使用するために予約されています）。

date

インストールの日付を指定します。 キーワード **current** を使用することも使用できます。

日付の形式は、YYYY-MM-DD です。

time

インストールの時間を指定します。 キーワード **current** を使用することも使用できます。

時間の形式は、HH:MM です。

orderedby

このオプション パラメータでは、アカウント名（例：DomainX¥UserY）を指定します。

デフォルト：SecureContainer コマンドを実行している現在のユーザ。

comment

コメントです。

[target="targetname"]

コマンドがコンピュータ ユニットに適用可能か（デフォルト）、または同じコンピュータ上の任意の有効なまたは既存のユーザ プロファイル ユニットに適用可能かどうかを指定します。以下のターゲットを除いて、任意のターゲットを指定できます。

- コマンドを実行しているコンピュータ上。
- コマンドを上を実行しているのと同じコンピュータ上のユーザ プロファイルから。

例：

以下の例では、プロシージャ **Uninstall** を使用したソフトウェア項目 **Product Y** バージョン **1.2** に関するアンインストール レコードを追加するように、**Software Delivery** に通知します。アンインストールには、現在の日付と **sda_time** が使用されます。**Ordered by** は **Automatic by Job**、**comment** は **Upgraded by Product Y v2.0 install** となります。

```
sd_acmd adduninstallrecord "Product Y" "1.2" "Uninstall" "" current current  
"Automatic by Job" "Upgraded by Product Y v2.0 install"
```

上記の例では、空の文字列がインストール プロシージャに対して使用されています。以前、インストール プロシージャ **Install1** と **Install2** を使用した 2 つのインストールが行われた場合は、これらのうちの 1 つのインストールのみが削除されます。ただし、* が使用されている場合は、両方とも削除されます。

Execute Container -- Software Delivery ジョブ コンテナ オーダ ファイルの実行

このコマンドは、Software Delivery ジョブ コンテナ オーダ ファイルを実行します。

構文

```
ExecuteContainer ContainerFile  
                    [-o:results directory]  
                    [-p:password]  
                    [-t]
```

ContainerFile

コンテナ オーダ ファイルへの完全パス。

[-o:results directory]

結果ディレクトリへの完全パス（オプション）。

[-p:password]

パスワード（オプション）。

[-t]

秒数。

コンテナ オーダ ファイルは、実行時に内部ディレクトリ `sdjexec`（「`¥CA¥DSM¥Agent¥units¥00000001¥usd¥sdjexec`」）内の `name.cwf` ファイルにコピーされます。

コンテナの実行が完了すると、ファイルの名前は `name.crf` に変更され、指定された結果ディレクトリ（使用可能な場合。それ以外の場合、ファイルは `sdjexec` ディレクトリに配置されます）に書き込まれます。

プロシージャ パラメータで `$rf` マクロが使用されている場合、出力ログファイルには拡張子が「`res`」の一意な名前が付けられ、指定された結果ディレクトリにコピーされます。

注：空の結果ファイルは、コンピュータ上に保存されません。

コンテナ ファイルを実行するには、データが `sd_acmd SecureContainer` を使用して保護されている必要があります。

Windows NT/2000 環境の場合、`sd_acmd ExecuteContainer` を使用して `Software Delivery` のオフライン ジョブを実行するには、十分なセキュリティ権限を持っている必要があります。デフォルトでは、ドメインまたはグローバルの管理者権限が付与されているユーザのみが、`Software Delivery` のオフライン ジョブを実行することができます。オプションで、Windows NT/2000 ローカル グループ `SDOFFLIN` またはグローバル グループ `Domain SDOFFLIN` のメンバであるユーザは、`Software Delivery` のオフライン ジョブを実行することができます。`Software Delivery` をインストールすると、すべての Windows NT/2000 コンピュータ上に、ローカル グループ `SDOFFLIN` が作成されます。管理者は、グローバル グループ `Domain SDOFFLIN` を作成して、ユーザをこれらのグループに追加し、ユーザが `Software Delivery` のオフライン ジョブを実行できるようにすることができます。

UNIX の場合、NIS (Network Information Service) が NT でのドメインユーザ検証と同様な機能を持っています。オフライン ジョブの `SD NIS` ユーザグループは、`SDOFFLIN` と呼ばれます。

例 :

以下の例は、ジョブ コンテナ オーダ ファイルの実行をオーダする方法を示しています。

```
sd_acmd executecontainer "C:¥myorders¥test1¥order1.cof" -o:"C:¥myresults"
-p:mypassword
```

`Output Directory` 引数は、コンテナ オーダ ファイルに追加することもできます (セクション `Container` のパラメータ `ResultDirectory`)。この場合、コマンドライン オプションによって、コンテナ オーダ ファイル内の値が上書きされます。

以下の例ではコンテナ オーダ ファイルがパスワードを使用して保護されているため (`SecureContainer`)、`ExecuteContainer` を実行するときにも同じパスワードパラメータを指定する必要があります。コンテナ オーダ ファイル実行プログラムでは、使用する前に、機密情報が内部的に復号化されます。

-t parameter:

ジョブの実行またはサーバへの接続でエージェントがビジー状態の場合は、オフライン ジョブ実行プログラムを実行できません。 -t: パラメータを使用すると、オフライン ジョブ実行プログラムに対して、指定された秒数の間待機してからジョブの実行を放棄し、エージェントがビジー状態であることを示すエラー コードを返すように指示することができます。値を指定しない場合は、デフォルトのタイムアウトが使用されます。

```
sd_acmd executecontainer cont.cof -t:30
```

オフライン ジョブ実行プログラムに対して、エージェントを 30 秒待ってから失敗するように指示します。

Encrypt--任意の文字列の暗号化

このコマンドは、パラメータとして渡された任意のプレーン文字列を暗号化します。

構文

```
encrypt <any-string>
```

<any-string>

プレーン文字列を指定します。

例 :

以下の例では、共有にアクセスするためのアカウント名であるプレーン文字列「machine1¥user1」の暗号化された値を返します。

```
sd_acmd encrypt machine1¥user1
```

同様に、パスワードをプレーン文字列として入力し、暗号化することもできます。

SecureContainer-- Software Delivery ジョブ コンテナ オーダ ファイルを暗号化

このコマンドを実行すると、**Software Delivery** のジョブ コンテナ オーダ ファイルが暗号化されることによって保護されます。

構文

```
SecureContainer ContainerFile  
[-p:password]
```

ContainerFile

コンテナ オーダ ファイルへの完全パス。

[-p:password]

パスワード（オプション）。

例：

以下の例は、パスワード **mypassword** を使用して、ディレクトリ **C:¥myorders¥test** にあるコンテナ オーダ ファイル **order1.cof** で、すべての機密情報（パスワード）がどのように暗号化されるかを示しています。

```
sd_acmd securecontainer "C:¥myorders¥test¥order1.cof" -p:mypassword
```

WaitContainers-- ジョブの終了を待機

このコマンドは、バッチ ファイル内の **ExecuteContainer** の呼出し後に使用して、保留中のジョブおよび実行中のジョブがすべて終了するまで待機することもできます。

構文

```
WaitContainers [-t:Timeout]  
[-f:ContainerFile]
```

[-t:Timeout]

単位は秒です。待機時間を指定します。デフォルト値は無限です。範囲は 0 - 2**31-1（オプション）です。

[-f:ContainerFile]

コンテナのファイル名（オプション）。ファイル名に拡張子の `cof` が省略されている場合は、自動的に付加されます。

注：このファイルは、「.¥CA¥DSM¥Agent¥units¥00000001¥usd¥sdjexec」に常駐されると想定されています。

例：

以下の例では、コンテナ オーダ ファイル `order3.cof` 内のジョブがすべて終了するまで待機するコマンドを示しています。タイムアウトは 30 秒です。

```
sd_acmd waitcontainers -t:30 -f:order3.cof
```

注：コンテナ ファイル（`cof` ファイル）が用意されていない場合は、すべてのオフライン ジョブに対する待機が開始されます。

ExitCodeMsg - 失敗したジョブのエラーメッセージを設定

このコマンドによって、現在のジョブ コンテナを実行するプログラムのジョブのエラーメッセージテキストを設定します。

構文

```
ExitCodeMsg [Error message]
```

where

エラー メッセージ

オプション パラメータで、これによって、現在のジョブ コンテナを実行するプログラムのジョブのエラーメッセージを決定します。

例：

ExitCodeMsg コマンドは、コマンドラインでの使用には適していません。バッチファイルとスクリプトファイルで使用してください。たとえば、実行可能プログラムを実行すると、別のエラーコードを表す終了コードまたはリターンコードが生成される場合があります。switch/case ロジックを使用しないと、「終了コード xx はエラーの可能性があります」というわかりにくいメッセージが表示されますが、このロジックを使用すると、このような場合にそれぞれ、内容を理解しやすいエラーメッセージが生成されます。メッセージ **SDM228483** は、コマンドの使用に関連付けられます。

注：エラーメッセージのパラメータが必須でない場合でも、このコマンドはパラメータを付けて使用することをお勧めします。

JobCheck--ジョブ確認の実行

このコマンドはジョブ確認タスクを実行し、ソフトウェア配信エージェントを起動してスケラビリティ サーバに接続します。

その機能を拡張するために、ReportTemplateSwStateDbRecords、ReportInstanceSwStateDbRecords、ReportAllSwStateDbRecords という 3 つの新しい VDI サポート スイッチが追加されました。これらのスイッチは、インスタンス/テンプレート データベースからジョブ レコードを送信するためにのみ使用され、オフラインの再インストールは実行しません。

構文

```
JobCheck [target="targetname"]
        [update]
        [installonly]
        [/BG]
        [/wait]
        [/ReportTemplateSwStateDbRecords]
        [/ReportInstanceSwStateDbRecords]
        [/ReportAllSwStateDbRecords]
```

[target="targetname"]

(オプション) 値が存在する場合は、指定されたターゲットに対して実行します。ターゲットを指定しない場合、ユーザ エージェントがターゲット コンピュータ上で有効になっていれば、現在のコンテキストがデフォルトになります。

[update]

(オプション) 属性のみを更新します。

注： update オプションと installonly オプションを同時に指定することはできません。

[installonly]

(オプション) 属性を更新しません。

[/BG]

(オプション) ジョブ確認タスクをバックグラウンドモードで実行します。

[/wait]

(オプション) ジョブ確認が完了するまで **sd_acmd** から戻らないようにします。

[/ReportTemplateSwStateDbRecords]

(オプション) オフラインの再インストールを実行せずに、テンプレート状態データベース レコードのみを送信します。

[/ReportInstanceSwStateDbRecords]

(オプション) オフラインの再インストールを実行せずに、インスタンス状態データベース レコードのみを送信します。

[/ReportAllSwStateDbRecords]

(オプション) オフラインの再インストールを実行せずに、すべてのソフトウェア状態データベース レコードを送信します。

JobProgress-- 現行ジョブのプログレス メッセージ/パーセントを設定

このコマンドによって、現在のジョブにプログレス メッセージとプログレス パーセントを設定します。

構文

```
JobProgress Text  
[Percentage]
```

Text

この必須パラメータは、現在のジョブ コンテナの実行プログラム ジョブのプログレス メッセージを設定します。

[Percentage]

オプション パラメータで、現在のジョブ コンテナを実行するプログラムのジョブのプログレス パーセントを設定します。

例 :

たとえば、以下のようにします。

```
sd_acmd jobprogress "Running Job 1" 50
```

Software Delivery によって、現在のジョブ コンテナを実行するプログラムのジョブのプログレス メッセージ「Running Job 1」がパーセント「50」と共に表示されます。

注 : JobProgress コマンドは、コマンドラインでの使用には適していません。バッチ ファイルとスクリプト ファイルで使った方が適しています。たとえば、実行可能プログラムを実行すると、バッチまたはスクリプトの実行シナリオ全体で別の状態を表すリターン コードが生成される場合があります。そのような関連する状態ごとに、特定のプログレス メッセージが生成され、通常のメッセージを置き換えてしまうことがあります。

ターゲットの削除 -- コンピュータからのターゲットの削除

このコマンドはコンピュータからターゲットを削除します。

構文

```
sd_acmd RemoveTarget Propertystring
```

Propertystring

プロパティ文字列には、「**TARGET=<ターゲット名>**」または「**TYPE=<タイプ>**」が含まれる場合があります。

有効なタイプは、**MACHINE**、**LOCAL_USER**、**DOMAIN_USER**、**PALM**、**WIN_CE**、**NOKIA** です。

削除対象のターゲット タイプが複数ある場合は、**TYPE=** プロパティがコマンドに複数回出現する場合があります。

例 :

たとえば、以下のようにします。

```
sd_acmd removetarget TYPE=WIN_CE
```

Software Delivery によって、Win32 のデスクトップ コンパニオンに添付されている Windows CE タイプのドッキング デバイスのレコードがすべて削除されます。

これによって、Windows CE デバイスがデスクトップ コンパニオンのドッキング デバイスとして機能した後に、ネイティブ WinCE エージェントがインストールされた場合は、デスクトップ コンパニオン上のレコードがクリーンアップされます。

2 人のローカル ユーザ **User1** および **User2** がターゲット コンピュータと関連付けられている場合

以下のように入力したとします。

```
sd_acmd removetarget TARGET=T01M0234/User1
```

この場合は、ローカル ユーザ **User 1** が削除されます。

以下のように入力したとします。

```
sd_acmd removetarget TYPE=LOCAL_USER
```

この場合は、Software Delivery エージェントに関連付けられているローカルユーザ全員（User 1 と User 2）が削除されます。

Remove Win32 Program

このコマンドを実行すると、
HKEY_LOCAL_MACHINE¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Uninstall¥"Product string" で登録されている製品に関連付けられているアンインストール コマンドが実行されます。この "Product string" は、<registrykey> パラメータで入力します。

構文

```
sd_acmd RemoveWin32Pgm RegistryKey [Propertystring]
```

RegistryKey

```
"HKEY_LOCAL_MACHINE¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion  
¥Uninstall¥""Product string"" の「Product String」"
```

[Propertystring]

<Propertystring> パラメータには、以下のプロパティが含まれる場合があります。

PARAMETERS=<ユーザが追加するパラメータ> - これは、自動処理用のように、コマンドラインのパラメータである場合があります。この値は、「QuietUninstallString」値にあるコマンドラインに付加されます。この値がない場合は「UninstallString」値に付加されます。

KEYEXISTS=1 - は、確認のみを実行することを意味しています。（この機能を使用すると、「インストール内容の検出」プロシージャおよび「アンインストール内容の確認」プロシージャを簡単に作成することができます。この戻り値は、キーが存在する場合は 0、存在しない場合は 6020 となります）。

TIMEOUT=<タイムアウト値> - 初期のアンインストール処理によって、実際のアンインストール処理が生成されるとき、RemoveWin32Pgm は、レジストリ キーが削除されるまで、秒単位で指定された時間だけ待機する場合があります。処理が完了し、さらにタイムアウトの時間が経過してもキーが残っている場合は、エラー コードが返されます。

OUTPUT=<ファイル名> - 処理中にファイルへのログを可能にします。

例：

以下の例で説明します。

```
sd_acmd removewin32pgm registrykey="Jukebox 1.0" timeput=60
```

この場合、**Software Delivery** によって、タイムアウトの 60 秒以内にコンピュータから **Jukebox** プログラムのバージョン **1.0** が削除されます。

SetDownloadMethod

このコマンドは、エージェント用のダウンロード方法を指定します。

構文

```
SetDownloadMethod {NOS|NONE|DTS}
```

NOS

内部 NOS

なし

内部 NOS なし

DTS

DTS NOS なし

注：指定したダウンロード方法が無効である場合は、操作が失敗します。

たとえば、マシンに **DTS** がインストールされていない場合、「sd_acmd SetDownloadMethod DTS」は、「SD_ACMD <6027>：無効なダウンロード方法」というメッセージが表示され失敗します。

Signal-- Software Delivery エージェントが実行するアクションの Software Delivery にシグナルを送信

このコマンドを実行すると、インストール中に **Software Delivery** エージェントが必要とする個別のアクションを実行したときに、**Software Delivery** にシグナルを送信できます。

構文

Signal

[reboot]
[rerun]
[logoff]
[rebootae]
[logoffae]

[reboot]

現在のジョブの実行後にコンピュータが再起動されるようにします。

[rerun]

次回も現在のジョブが再起動されます。このコマンドは、**logoff** パラメータおよび **reboot** パラメータと共に使用できます。

[logoff]

現在のジョブが実行された後でユーザをログオフします。

[rebootae]

バッチ内の最後のジョブが実行された後でコンピュータを再起動します。

[logoffae]

バッチ内の最後のジョブが実行された後でユーザをログオフします。

少なくとも 1 つの引数が必要です。その要件以外に、あらゆる組み合わせによる引数を選択できます。ただし、**rebootae** と **rerun** を同時に使用することはできません。

例：

以下の例は、現在のジョブを実行した後にコンピュータを再起動する必要があること、および再起動時にこのジョブを実行する必要があることを示すシグナルを、**Software Delivery** エージェントに送信する方法を示しています。これは、**SXP** パッケージの関係するジョブを実行したときなどに、**Software Delivery** で使用されます。この機能を効率的に利用するには、ユーザが記述したプロシージャに適切なロジックを組み込む必要があります。

```
sd_acmd signal reboot rerun
```

ReinstallTarget--ターゲットに再インストールされるソフトウェア

このコマンドは、エージェントに展開されたソフトウェアがすべて復元される、オフライン **RAC** プロセスの実行を開始します。

注： このコマンドは **VDICompose.dms** 統合スクリプトから呼び出されるため、通常は手動では呼び出されません。ただし、オフライン **RAC** プロセスが **VMware View** インテグレーション以外の目的に使用されている場合は、他のメカニズムを使用して呼び出すことが必要になる可能性があります。

構文

```
ReinstallTarget [abort]
```

[abort]

(オプション) 保留中のターゲットの再インストール リクエストをすべてキャンセルします。

例：

```
sd_acmd ReinstallTarget
```

この例では、オフラインの再インストールを実行し、エージェントを呼び出します。このエージェントは、インスタンス ソフトウェア状態データベースを開き、エージェントの現在のソフトウェア ライブラリを使用して、すべてのプロシージャを時間順に再実行します。ジョブ ステータスがドメイン マネージャに報告され、標準のすべてのソフトウェア配信ジョブと同様に [DSM Software Delivery ジョブ確認] ダイアログ ボックスに表示されます。また、エージェントはテンプレート ソフトウェア状態データベースを開き、エントリごとにインストール/起動/設定レコード (sd_acmd AddXXXRecord によって生成されるものと同じ) を作成します。再インストールプロセスの最後に、生成されたすべてのレコードがスケラビリティ サーバを使用してドメイン マネージャに送信されます。

注: このメカニズムは、再構成または更新されるたびに新しいホスト UUID を取得するエージェントに依存します。これにより、コンピュータの MDB 内の既存のソフトウェア レコードの削除がトリガされます (オフライン RAC が完了するとすぐに新しいレコードに置き換えられるだけです)。

UserInfo - コンピュータのターゲットにユーザ情報を設定

このコマンドを実行すると、現在のコンピュータのターゲットにユーザ情報が設定されます。

構文

```
UserInfo [location]
         [user]
         [phone]
         [target]
         [comment]
```

[location]

オプション パラメータで、これによって、現在のコンピュータのロケーション テキストを設定します。

[user]

オプション パラメータで、これによって、現在のコンピュータのユーザ情報テキストを設定します。

[phone]

オプション パラメータで、これによって、現在のコンピュータの電話情報テキストを設定します。

[target]

コマンドがコンピュータ ユニットに適用可能か（デフォルト）、または同じコンピュータ上の任意の有効なまたは既存のユーザ プロファイルユニットに適用可能かどうかを指定します。以下のターゲットを除いて、任意のターゲットを指定できます。

- コマンドを実行しているコンピュータ上。
- コマンドを上を実行しているのと同じコンピュータ上のユーザ プロファイルから。

[comment]

オプション パラメータで、これによって、現在のコンピュータのコメント テキストを設定します。

例：

以下の例で説明します。

```
sd_acmd userinfo location="Room 5" user="The user" phone=123456 target=MyComputer  
Comment="Building 22"
```

Client Automation は、[DSM プロパティ] の [ユーザ データ] タブ (Systray) で DSM エクスプローラのユーザに指定した情報を表示します。

注：Target など、どの引数でも自由に省略できます。また、大文字と小文字のどのような組み合わせでも選択できます。

このコマンドでユーザ情報を変更すると、ほかの変更と食い違いが生じる場合があります。 コマンドが実行されたコンピュータで行われた変更は、エージェントのマネージャに報告されます。

DSM エクスプローラでコンピュータ ターゲットを選択して情報の変更を行った場合、その変更が適用されるのはエージェント コンピュータではなくサーバのみです。

エージェント コンピュータ自体の情報が変更されるのは、影響を受けるコンピュータ ターゲットに対して、変更を開始する `sd_acmd userinfo` コマンドを実行したときです。 行った変更は、次にマネージャに報告され、それまでに DSM エクスプローラを使用して行った変更が上書きされます。

ユーザ情報を設定する 3 番目の方法としては、Systray ([DSM プロパティ] / [ユーザ データ] タブ) のヘルプを使用します。

コンテナ オーダ ファイル (COF) フォーマット

コンテナ オーダ ファイル、つまり COF (.cof) は、以下のセクションに分かれています。

```
Locale
    codepage=
Container
    Name=
    Password=
    Type=
    Transaction=
    OrderedBy=
    CreationTime=
    ManagerAddress=
```

```
ResultDirectory=  
NumberOfJobs=  
NumberOfLibraries=  
BackgroundProcess=  
ShowProgress=  
Status=  
StatusMessage=  
CompletionTime=  
Secure=  
Validate=  
Hash=  
UserName=  
DomainName=  
Job_1  
Library_1
```

Locale

codepage=

このパラメータはオプションです。

コード ページが指定されていない場合は、[SecureContainer](#) (P. 31) によって 3 に指定されます。

以下の値を指定できます。

1=ANSI

2=OEM

3=UTF8

また、これ以外にも、現在のシステムがサポートしているコード ページを使用できます (例 850) 。

Container

Name=

ログ用および出力用のコンテナの名前。

デフォルトでは、COF ファイル名が名前になります。

Password=

パスワードは、COF ファイルの保護に使用します。

パラメータはオプションです。 パラメータを使用した場合は、

SecureContainer コマンドを実行したときに値が暗号化されます。

パラメータは、**SecureContainer** を使用し、オプションの引数 -p: を指定して入力することもできます。 その場合、ここで入力した値よりも、コマンドで指定した値が優先されます。

Type=

このパラメータは、コンテナのタイプを表します。
有効な値は以下のとおりです。

- 0 リンクなし
- 1 予約済み
- 2 バッチ

デフォルト : 0

Transaction=

このパラメータは、コンテナ内のジョブにトランザクションが使用されているかどうかを表します。これは、ソフトウェア管理 (SM) のバッチ コンテナにのみ適用されます。
有効な値は以下のとおりです。

- 0 False (使用されていない)
- 1 True (使用されている)

デフォルト : 0

OrderedBy=

このオプション パラメータでは、アカウント名 (例 : DomainX¥UserY) を指定します。
デフォルト : SecureContainer コマンドを実行している現在のユーザ。

CreationTime=

(オプション) コンテナの作成日時を、YYYY-MM-DD hh:mm という形式で指定します。YYYY は年、MM は月、DD は日、hh は時間、mm は分を表します。

ManagerAddress=

(オプション) ドメイン マネージャにアドレスを提供します。値が指定されない場合は、共通設定の「currentmanageraddress」からデフォルト値が取り出されます。
このパラメータは、\$csa マクロを置き換えるために使用されます。

ResultDirectory=

このパラメータはオプションであり、出力ディレクトリへの完全パスを指定します。

また、コマンドラインから **ExecuteContainer** を実行したときに、**Output Directory** パラメータの値が優先される場合があります。

NumberOfJobs=

このパラメータは必須です。このパラメータは、COF ファイル内の後続 **Job** セクションの数を指定します。

デフォルト値は **1** です。

注：この値が **1** の場合は、**Job_1** という名前の **Job** セクションが **1** つしかありません。この値が **2** の場合は、**Job_1** および **Job_2** という **2** つのセクションがあります。このように、値の数が増えると、それに応じてセクションの数も増えていきます。

NumberOfLibraries=

このパラメータは必須です。このパラメータは、COF ファイル内の後続 **Library** セクションの数を指定します。

デフォルト値は **1** です。

注：この値が **1** の場合は、**Library_1** という名前の **Library** セクションが **1** つしかありません。この値が **2** の場合は、**Library_1** および **Library_2** という **2** つのセクションがあります。このように、値の数が増えると、それに応じてセクションの数も増えていきます。

BackgroundProcess=

このパラメータでは、ジョブをバックグラウンドで処理するかどうかを指定します。

有効な値は以下のとおりです。

- 1 関連するプロシージャにある値を使用します
- 0 いいえ
- 1 はい

デフォルト値は **-1** です。

注：関連するプロシージャにある値は、**\$#bg** マクロを使用する場合はバックグラウンドで処理する、マクロがない場合はバックグラウンドで処理しない、となります。

ShowProgress=

このパラメータでは、プログレス ダイアログ ボックスを表示するかどうかを指定します。

有効な値は以下のとおりです。

- 0 いいえ
- 1 はい

デフォルト値は 1 です。

Status=

このパラメータは、コンテナの実行状態を表します。

以下に有効な (ENU-) 値を示します。

- 0 OK
- 1 少なくとも 1 つのジョブが失敗しました。
- 2 ジョブは処理されません。
- 3 COF が進行中。
- 4 再起動が開始されました。
- 5 ログオフが開始されました。
- 6 COF ファイルがありません。
- 7 一般 COF ファイルの読み取り中にエラーが発生しました。ファイルが不正に変更された可能性があります。
- 8 指定されたパスワードが無効です。
- 9 COF ファイルが保護されていません。

OUT、つまり、パラメータの値が、指定された結果ディレクトリのファイル *.crf に出力されます。

StatusMessage=

このパラメータでは、ネイティブの OS メッセージのように、ステータスのテキストを指定します。この値は、**Status** パラメータの値に対応します。**OUT**、つまり、パラメータの値が、指定された結果ディレクトリのファイル *.crf に出力されます。

この値は、**Status** パラメータの値に対応します。

OUT、つまり、パラメータの値が、指定された結果ディレクトリのファイル *.crf に出力されます。

CompletionTime=

このパラメータは、ジョブ コンテナの完了時間を表します。

その構造は、YYYY-MM-DD hh:mm (年-月-日 時間:分) です (例、2001-04-09 15:34:52)。

OUT、つまり、パラメータの値が、指定された結果ディレクトリのファイル *.crf に出力されます。

Secure=

sd_acmd SecureContainer コマンドを実行することによって、このパラメータの値は 1 に設定されます。

OUT、つまり、パラメータの値が、指定された結果ディレクトリのファイル *.crf に出力されます。

Validate=

暗号化された値で、COF パスワードの検証に使用されます。

sd_acmd ExecuteContainer によって作成され、処理が終了すると削除されます。

Hash=

-v と共に使用した場合にのみ有効です。**SDPing** が失敗した場合に、設定を強制的に変更します。

UserName=

このパラメータは、**ExecuteContainer** を実行したときに挿入されます。現在ログオンしているユーザの名前が指定されます。

DomainName=

このパラメータは、`ExecuteContainer` を実行したときに挿入されます。現在ユーザがログオンしているドメインがあれば、その名前が指定されます。

Job_n

詳細については、[Job セクション](#) (P. 52)の説明を参照してください。

Library_n

詳細については、[Library セクション](#) (P. 58)の説明を参照してください。

COF ファイルの例

以下に示す **cof** ファイルの例は、デフォルトの設定値をすべて使用した、最小のものです。

ネットワーク ライブラリを使用するための COF ファイル

```
[Container]
NumberOfJobs=1
NumberOfLibraries=1

[Library_1]
Type=1
Path=¥¥THE_SERVER¥SDLIB

[Job_1]
SoftwareName=SuperOffice for Win2000
SoftwareVersion=7.1 ENU (I386)
ProcedureName=InstallSO
```

CD ライブラリを使用するための COF ファイル

```
[Container]
NumberOfJobs=1
NumberOfLibraries=1

[Library_1]
Type=2
Path=Lib
Label=001229_1045

[Job_1]
SoftwareName=SuperOffice for Win2000
SoftwareVersion=7.1 ENU (I386)
ProcedureName=InstallSO
```

注：テンプレートファイル **template.cof** が、**CONF** ディレクトリにあります（Windows 環境では %SDROOT%\¥ASM¥CONF、UNIX では \$SDROOT/asm/conf）。

注： Job と Library のタイプのセクションは複数存在する場合があります。
このような場合、これらのセクションには Job_2 や Library_2 などのラベルが付けられます。

重要： COF を作成する際にテキスト エディタを使用する場合は、COF を保存するときに適切なコーディングを選択する必要があります。つまり、コンピュータの現在のコード ページが、使用した文字をすべてサポートしているかどうかを確認する必要があります。サポートしていない場合は、codepage= parameter を使用して適切なコードを選択してください。

たとえば、ドイツ語のウムラウトを使用した場合は、UTF-8 を選択する必要があります (codepage 3 の場合)。

コード ページが指定されていない場合は、[SecureContainer](#) (P. 31) によって 3 に指定されます。

COF Section Locale

codepage=

このパラメータはオプションです。

コード ページが指定されていない場合は、[SecureContainer](#) (P. 31) によって 3 に指定されます。

以下の値を指定できます。

1=ANSI

2=OEM

3=UTF8

また、これ以外にも、現在のシステムがサポートしているコード ページを使用できます (例 850)。

COF Section Job

COF ファイルには、この名前のセクションが 1 つ (Job_1) または複数 (Job_2 など) があります。ジョブ コンテナには、個々のジョブの特性が含まれています。

Name=

このパラメータでは、ログ用および出力用のジョブの名前を指定します。

デフォルト値は、**SoftwareName** および **ProcedureName** です。

例：

SW:SuperOffice for Win2000 Proc:InstallSO

SoftwareName=

このパラメータでは、ソフトウェア パッケージの名前（ソフトウェアのプロパティに表示）を指定します。

例：

SuperOffice for Win2000

SoftwareVersion=

このパラメータでは、パッケージのバージョン（ソフトウェアのプロパティに表示）を指定します。

例：

7.1 ENU (I386)

ProcedureName=

このパラメータでは、ジョブに使用するプロシージャの名前（プロシージャのプロパティに表示）を指定します。

UserParameters=

このパラメータ値は、関連するプロシージャに \$sup マクロがある場合に使用されます（ここで **UserParameters** が設定可能になります）。

これはオプションです。

デフォルト値は空の文字列です。

BootBefore=

このパラメータでは、ジョブを起動する前にターゲット コンピュータを再起動するかどうか、再起動する場合は、どのレベルで行うかを指定します。

有効な値は以下のとおりです。

- 1 プロシージャのデフォルトを使用
- 0 再起動しない
- 1 再起動する
- 2 ログオフする

デフォルト値は -1 です。

このパラメータが適用されるのは、Windows NT / 2000 および Windows 9x / ME のみです。

BootAfter=

このパラメータでは、ジョブを起動した後でターゲット コンピュータを再起動するかどうか、また再起動をどのレベルで行うかを指定します。

有効な値は以下のとおりです。

- 1 プロシージャのデフォルトを使用
- 0 再起動しない
- 1 再起動する
- 2 ログオフする
- 4 最後のジョブの後で再起動する
- 8 最後のジョブの後でログオフする

最後の 2 つの値が該当するのは、複数のジョブがリンクされ、1 つのジョブ コンテナにまとめられた場合のみです。

デフォルト値は -1 です。

このパラメータが適用されるのは、Windows NT / 2000 および Windows 9x / ME のみです。

ExecutionTimeOut=

このパラメータでは、ジョブの処理に使用するタイムアウト値を指定します（分単位）。

有効な値は、デフォルトで **1 - 71582** です。

デフォルト値は **120** です。

ExecutionTimeoutAction=

このパラメータでは、上記のタイムアウト値が経過した場合に実行するアクションを指定します。

有効な値は以下のとおりです。

0 アクション不要

1 処理を停止

デフォルト値は **0** です。

PromptRemovable=

このパラメータでは、メディアが挿入されていない場合にメディアを要求するかどうかを指定します。

このパラメータが有効になるのは、関連する **Library** セクションにおける **Type** パラメータの値が **2** の場合のみです。

有効な値は以下のとおりです。

0 要求しない（デバイスが存在しない場合は次のライブラリで試行）

1 要求する

デフォルト値は **0** です。

LogonShield=

このパラメータでは、**Logon Shield** を起動するかどうかを指定します。

このパラメータが適用されるのは、エージェント ポリシーが「**per job**」に設定されている場合のみです（つまり、ターゲット コンピュータに対して、**Logon Shield** のプロシージャ **Enable per job** が前もって実行されます）。

有効な値は以下のとおりです。

0 起動しない

1 起動する

デフォルト値は **0** です。

このパラメータが適用されるのは、**Windows NT / 2000** のみです。

LibrarySearchOrder=

このパラメータでは、**Library** セクションの検索順序を特定する、インデックスのカンマ区切りのリストを指定します。

このパラメータはオプションです。

値が指定されていない場合、セクションが **Library_1** しかないときは、検索順序はデフォルトで **1** になります。セクションに **Library_1** と **Library_2** がある場合、検索順序は **1,2** となります。このように、セクションの数が増えると、それに応じて検索順序の値も増えていきます。

CompletionTime=

このパラメータは、ジョブ コンテナの完了時間を表します。

その構造は、YYYY-MM-DD hh:mm (年-月-日 時間:分) です (例、2001-04-09 15:34:52)。

OUT、つまり、パラメータの値が、指定された結果ディレクトリのファイル *.crf に出力されます。

ExitCode=

このパラメータでは、プロシージャによって参照された実行可能ファイルから提供された終了コードを格納します。

OUT、つまり、パラメータの値が、指定された結果ディレクトリのファイル *.crf に出力されます。

ReturnCode=

このパラメータでは、実行可能プラグイン **SD_JEXEC** から提供された終了コードを格納します。

有効な値は以下のとおりです。

- 0 OK
- 1 エラー
- 2 実行されていない

OUT、つまり、パラメータの値が、指定された結果ディレクトリのファイル *.crf に出力されます。

ErrorMessage=

このパラメータでは、用意されたエラー テキストを格納します。このテキストは、ネイティブの OS メッセージのようになる場合もあります。OUT、つまり、パラメータの値が、指定された結果ディレクトリのファイル *.crf に出力されます。

ResultFileIn=

このパラメータでは、ユーザが指定した結果ファイルの名前を格納します。結果ファイルは、関連するプロシージャで \$rf マクロを使用した場合に自動生成されます。

このファイルへのパスは、Container セクションの ResultDirectory パラメータで指定したパスとなります。

ResultFileOut=

このパラメータでは、出力 / ログ ファイルへの完全パスをファイル名付きで格納します。出力 / ログ ファイルは、関連するプロシージャで \$rf マクロを使用した場合に自動生成されます。

OUT、つまり、パラメータの値が、指定された結果ディレクトリのファイル *.crf に出力されます。

Reinstall=

このパラメータでは、ジョブが再インストール ジョブかどうかを指定します。

有効な値は以下のとおりです。

0 いいえ

1 はい

デフォルト : 0

COF Section Container

COF ファイルには、この名前のセクションが **1** つあります。このセクションには、ジョブ コンテナの特性が含まれています。

Name=

Password=

Type=

Transaction=

OrderedBy=

CreationTime=

ManagerAddress=

ResultDirectory=

NumberOfJobs=

NumberOfLibraries=

BackgroundProcess=

ShowProgress=

Status=

StatusMessage=

CompletionTime=

Secure=

Validate=

Hash=

UserName=

DomainName=

COF Section Library

COF ファイルには、この名前のセクションが 1 つ (Library_1) または複数 (Library_2 など) あります。このセクションには、COF ファイルが参照するライブラリの特徴が含まれています。

Type=

Path=

MapDrive=

Label=

NOSType=

Account=

Password=

COF ファイルの例

以下に示す **cof** ファイルの例は、デフォルトの設定値をすべて使用した、最小のものです。

ネットワーク ライブラリを使用するための COF ファイル

```
[Container]
NumberOfJobs=1
NumberOfLibraries=1

[Library_1]
Type=1
Path=¥¥THE_SERVER¥SDLIB
```

```
[Job_1]
SoftwareName=SuperOffice for Win2000
SoftwareVersion=7.1 ENU (I386)
ProcedureName=InstallSO
```

CD ライブラリを使用するための COF ファイル

```
[Container]
NumberOfJobs=1
NumberOfLibraries=1

[Library_1]
Type=2
Path=Lib
Label=001229_1045
```

```
[Job_1]
SoftwareName=SuperOffice for Win2000
SoftwareVersion=7.1 ENU (I386)
ProcedureName=InstallSO
```

注：テンプレートファイル **template.cof** が、**CONF** ディレクトリにあります（Windows 環境では **%SDROOT%¥ASM¥CONF**、UNIX では **\$SDROOT/asm/conf**）。

COF Library セクションの例

[Library_1] ;local media	[Library_2] ;network	[Library_3] :removable
Type=0	Type=1	Type=2
Path=c:¥lib	Path=¥¥server¥lib	Path=¥lib
	Account=eunt-a01¥xylzw01 Label="Master CD"	
	Password=xyWtUG	
	MapDrive=0	

sd_acmd リターン コード

sd_acmd によって問題が検出された場合は、問題の内容を説明したメッセージが標準的な形式で出力されます。このメッセージのヘッダは「sd_acmd<xxx>:」です。ここで xxx は、報告されたリターンコードです。以下の表に、さまざまなリターンコードを示します。

-100	引数が多すぎます。
-101	引数が見つかりません。
-102	構文エラーです。
-103	日付形式が無効です。
-104	時間形式が無効です。
-105	コマンドが不明です。
4	内部エラーが発生しました。
110	ソフトウェアの名前を指定してください。
111	ソフトウェアのバージョンを指定してください。
5000	プロシージャを指定してください。
5001	レコードが 255 バイトを超えています。
6000	Software Delivery が見つかりませんでした。
6001	ジョブがすでに実行中です。
6002	ファイルが見つかりません。
6003	ファイルタイプが間違っています。

6004	コンテナの作業ディレクトリを作成できません。
6005	COF ファイルを使用できません。
6006	ジョブを実行できません。 Software Delivery が動作していない可能性があります。
6007	ジョブはすでに実行中です。後で再試行してください。
6008	パスワードが無効です。
6009	このファイルを実行するにはパスワードが必要です。
6010	暗号化エラー
6011	暗号化のロードエラーです
6012	コンテナがすでに保護されています。
6013	この操作を実行する権限がありません。
6015	タイムアウトが発生しました。
6016	この COF ファイルは無効です。 セクションがない可能性があります。
6017	ターゲットが見つかりません。
6018	任意のターゲットのジョブ確認をトリガできませんでした。
6019	文字列が最大長を超えています。
6020	要求されたレジストリ キーが見つかりませんでした。
6021	実行に失敗しました。
6022	実行後もレジストリ キーは持続されます。
6023	指定されたアドレスの Ping に失敗しました。
6027	無効なダウンロード方法（ダウンロード方法が使用できません）。
6028	FIPS のみモードで動作しているときは、 Client Automation の旧バージョン上で安全な COF ファイルを実行できません。
6014	ジョブを実行できません。 COF ファイルが保護されていません。
801	指定したターゲットに、この操作に無効なタイプがあります。
1101	指定したターゲットが見つかりません。
1102	名前の変更が失敗しました。
1103	この時点では、指定したターゲットの名前が変更されていない可能性があります。 このターゲットに対して、次回 JobCheck を実行したときに再試行します。

第 3 章 : sd_msiexe.exe

sd_msiexe.exe は、Windows インストーラ タスクの実行に使用できます。sd_msiexe.exe は、Microsoft の msixec.exe のバージョン 1.1 および 2.0 と類似していますが、sd_msiexe.exe は Software Delivery 用に最適化されています。

注： Software Delivery マクロは、このドキュメントでも使用されています（\$msi、\$iv など）。これらはジョブの処理中にさまざまな文字列に展開され、その後 sd_msiexe.exe が処理されます。コマンドが直接（Software Delivery ジョブの外で）実行される場合、マクロは使用されない場合がありますが、実際の値/文字列に置換される必要があります。

sd_msiexe-- .msi パッケージのインストール/設定/アンインストールを実行

sd_msiexe を実行すると、msi パッケージのインストール/設定/アンインストールが実行されます。

コマンドの構文

```
sd_msiexe /i|x|a|j|m|j|u|f|t|p <package> /u <$joid> [/uv <"$iv">] [/un <"$in">]
[/q<uimode>] [/l<logmode> <LogFile>] [Property strings]
```

注：SD 固有のパラメータについてのみ説明しています。仕様の詳細については、msixec.exe ツール（Microsoft の資料）を参照してください。

/u--jobID

uv--SD パッケージのバージョン。

un--SD パッケージの名前。

注：/uv と /un が必要になるのは、SDMSILIB を使用した管理とネットワークのみです。

sd_msiexe -scan -- 現在のユーザ プロファイルに対して .msi パッケージを実行

sd_msiexe -scan を使用して、現在のユーザ プロファイルに対して .msi パッケージを実行します。

コマンドの形式

```
sd_msiexe -scan $cn -out:$rf (66以下のページで定義参照：)
```

\$cn -- 現在のプロファイルの名前に置き換えられます。

-out -- 出力ファイルを指定します。

sd_msiexe -admdel-- 指定した管理者がインストールしたファイルの管理インストールを削除

このコマンド sd_msiexe -admdel では、指定した管理者がインストールしたファイルの管理者インストールを削除します。

コマンドの形式

```
sd_msiexe -admdel <msipath> [-out:$rf]
```

-out -- 出力ファイルを指定します。

sd_msiexe -sourceupdate-- 現在のプロファイルの MSI ソース リストを更新

sd_msiexe -sourceupdate を使用して、現在のプロファイルの MSI ソース リストを更新します。

コマンドの形式

```
sd_msiexe -sourceupdate [-out:$rf]
```

-out -- 出力ファイルを指定します。

sd_msiexe -updatedictionary-- SDMSILIB デictionaryを更新

コマンド sd_msiexe -updatedictionary では、SDMSILIB Dictionaryを更新します。

管理者インストールの情報が、不完全なレコードに追加されます。

コマンドの形式

```
sd_msiexe -updatedictionary
```

sd_msiexe -removemsilibrecord-- SDMSILIB Dictionaryのレコードを削除

sd_msiexe -removemsilibrecord を使用して、SDMSILIB Dictionaryからレコードを削除します。

コマンドの形式

```
sd_msiexe -removemsilibrecord ADMINPATH="$msi" | SDNAME="$in" SDVER="$iv"
```

SD ソフトウェアの名前およびバージョンのプロパティ（SDNAME および SDVER）、またはパスのプロパティ（ADMINPATH）のいずれかを追加する必要があります。

sd_msiexe -addmsilibrecord-- SDMSILIB Dictionaryにレコードを追加

コマンド sd_msiexe -addmsilibrecord では、SDMSILIB Dictionaryにレコードを追加します。

コマンドの形式

```
sd_msiexe -addmsilibrecord SDNAME="$in" SDVER="$iv" ADMINPATH="$msi"
```

正しい SD SW の名前（SDNAME）、SD SW のバージョン（SDVER）、およびディレクトリのパス（ADMINPATH）を指定する必要があります。

sd_msiexe ?|-?|/?|help-- sd_msiexe コマンドの使用方法的表示

"?|-?|/?|help" を使用して、sd_msiexe コマンドの使用方法的を表示します。

コマンドの形式

```
sd_msiexe ?|-?|/?|help
```

出力ファイルを指定します。

第 4 章 : sd_sscmd -- ステージング ライブラリ コマンド ライン 管理

このセクションでは、コマンドラインを使用して新規ソフトウェア プログラムを登録したり、ステージング ライブラリでこれらのソフトウェア プログラムを登録解除する方法について説明します。

コマンドライン インターフェイスでロケール パラメータを使用するには、コンソールのフォントを **Lucida** に変更します。これにより、コマンド プロンプトから **sd_sscmd verbose** を使用したときに、ロケール文字が正しく表示されるようになります。

どの **sd_sscmd** コマンドを使用できるかは、システム上に **DSM** マネージャが存在するかどうかによって異なります。マネージャ マシン上でスケラビリティ サーバが実行されている場合は、**sd_sscmd** コマンドのサブセットのみが使用可能です。

コマンド通知

各コマンドの構文は、コマンド、および複数の関連キーワードとパラメータで構成されています。パラメータ値では、大文字と小文字が区別されます。コマンド名およびキーワードでは、大文字と小文字は区別されません。

コンポーネント間に使用する区切り文字は、スペースです。

スペースを含むパラメータ（およびそのキーワード）は、引用符で囲む必要があります（詳細モードの場合は不要です）。以下に例を示します。

```
item="Software Test1"
```

キーワードの後にパラメータが続く場合は、1つのパラメータ値のみを指定することができます。

バッチファイルでは、下位互換性のために両方の形式がサポートされています。パラメータ全体を引用符で囲むことをお勧めします。

ステージング ライブラリをホストしているスケーラビリティ サーバから、ステージング ライブラリの管理を開始します。

この場合、以下のコマンド構文が適用されます。

```
sd_sscmd
{
  stagecheck (P. 70)
  | bulkupdate (P. 71)
  | libraryaccess {user={domainname | local%}user password=password | REMOVE}
(P. 72)
  | importmsi {mapfile=Mapfile [path=path] [move=false|true]} (P. 73)
  | addshare {SDLIBRARY | MSILIB | [UseFQDN=N|Y]} (P. 73)
  | removeshare {SDLIBRARY|MSILIB} (P. 74)
  | import {[path=path] [move=false|true]} (P. 75)
  | areqsw {path=path [reginfo=reginfo] [logfile=logfile]} (P. 75)
  | dereg {item=item version=version [logfile=logfile]} (P. 77)
  | batch {path=path [logfile=logfile]} (P. 78)
  | verbose (P. 77)
}
```

マネージャ マシン上でスケーラビリティ サーバが実行されている場合は、コマンドのサブセットのみが使用可能です。

```
sd_sscmd
{
  stagecheck (P. 70)
  | bulkupdate (P. 71)
  | libraryaccess {user={domainname | local¥}user password=password | REMOVE}
(P. 72)
  | importmsi {mapfile=Mapfile \[path=path\] \[move=false|true\]} (P. 73)
  | addshare {SDLIBRARY | MSILIB | \[UseFQDN=N|Y\]} (P. 73)
  | removeshare {SDLIBRARY|MSILIB} (P. 74)
  | verbose (P. 77)
}
```

長さ制限

以下の表は、コマンドラインでパラメータ値を入力するときに適用される最大長制限をまとめたものです。

path 255

item 128

version 128

logfile 255

stagecheck

Windows および UNIX で有効です。

stagecheck コマンドを使用すると、SD サーバのステージ確認を開始することができます。このコマンドは、マネージャ上でも実行できます。このコマンドは、キャッシュされたメッセージをスケラビリティサーバからドメインマネージャに強制的に送信したい場合に実行できます。ただし、一括更新モードが有効になっており、ステージチェックモードが無効になっている場合、ステージチェック機能は未登録のユニットのみを確認し、キャッシュされたメッセージを送信しません。登録済みのユニットに関するキャッシュされたメッセージは、一括更新中に送信されます。

このコマンドは、指定されたステージチェック間隔に依存しません。

コマンドの形式

```
sd_sscmd stagecheck
```

bulkupdate

Windows および UNIX で有効です。

bulkupdate コマンドは、スケーラビリティ サーバに対して、ドメイン マネージャへのメッセージの一括送信リクエストを送信します。ドメイン マネージャに送信される各メッセージには、設定ポリシーの [一括更新: メッセージ中のジョブ結果の最大数] および [一括更新: 単一のメッセージに含めて送信するソフトウェア検出/ジョブ レコードの最大単位] の値に基づいて指定された数のサブメッセージが格納されます。設定ポリシーがスケーラビリティ サーバに適用されていない場合、設定ポリシーのデフォルト値が使用されます。

bulkupdate コマンドは、キャッシュされたメッセージを一括送信するたびに実行できます。このコマンドは一括更新間隔には依存せず、一括更新を即座に実行します。また、このコマンドは指定された一括更新間隔をリセットしません。たとえば、一括更新間隔を 10 分に設定し、その 5 分後に **sd_sscmd bulkupdate** コマンドを実行したとします。この場合、このコマンドは一括更新を即座に実行しますが、設定されている更新間隔をリセットしません。このため、実際の一括更新間隔が経過する 10 分後に再び一括更新が実行されます。

重要: スケーラビリティ サーバが一括更新モードで動作していない場合でも、このコマンドを実行してメッセージを一括送信できます。この場合、デフォルトのメッセージ サイズ値が使用されてメッセージが一括送信されます。

コマンドの形式

```
sd_sscmd bulkupdate
```

注: このコマンドは、スタンドアロン スケーラビリティ サーバと同様、ドメイン マネージャ上で動作するスケーラビリティ サーバに対しても有効です。

sd_sscmd libraryaccess

Windows および UNIX で有効です。

コマンド **libraryaccess** では、SD ソフトウェア ライブラリにアクセスする場合に接続側エージェントが使用するクレデンシャルを設定することができます。また、このコマンドは、マネージャ上でも実行することができます。

コマンドの形式

Windows の場合：

```
sd_sscmd libraryaccess {user=(domainname | local¥)user password=password |  
REMOVE}
```

UNIX/Linux の場合

```
sd_sscmd libraryaccess {user=(domainname | local¥¥)user password=password |  
REMOVE}
```

注：円マークは 2 重にする必要があります。

user

ドメインまたは「ローカル」（ローカル マシン上の現在のユーザ）の
名前¥ユーザの名前

パスワード

ユーザのパスワード

REMOVE

ユーザ名とパスワードを削除します。

例：

```
sd_sscmd libraryaccess user=MyComputer¥TheUser password=MyPassword
```

以下のコマンドは、ライブラリ アクセスのために **Comstore** に暗号化されているユーザ名とパスワードを削除します。

```
sd_sscmd libraryaccess REMOVE
```

sd_sscmd importmsi

Windows および UNIX で有効です。

importmsi コマンドを使用すると、SD 4.0 インストールから MSI ライブラリをインポートできます (マイグレーションフェーズで使用されます)。このコマンドは、マネージャ上でも実行できます。

コマンドの形式

```
importmsi {mapfile=Mapfile [path=path] [move=false | true]}
```

mapfile

マネージャのマイグレーション中に作成されるファイル。

path

MSILIB のルートへのソースパスを指定します。

path を指定しない場合は、SD 4 の MSILIB の検索がローカルマシン上で実行されます。

move

ファイルを元の場所から削除するか (True)、削除しないか (False) を指定します。

デフォルト: いいえ (False)

sd_sscmd addshare

Windows および UNIX で有効です。

コマンド **addshare** を使用すると、共有を有効にすることができます。このコマンドは、マネージャ上でも実行できます。

コマンドの形式

Windows の場合

```
sd_sscmd addshare {SDLIBRARY | MSILIB | [UseFQDN=N|Y]}
```

UNIX の場合

```
sd_sscmd addshare {SAMBA | NFS | SDLIBRARY | MSILIB | [UseFQDN=N|Y]}
```

[UseFQDN=N|Y]

(オプション) 共有名に、コンピュータ名のみ (例 :
¥¥mycomputer¥SDMSILIB) ではなく「完全修飾ドメイン名」 (例 :
¥¥mycomputer.ca.com¥SDMSILIB) が含まれることを指定します。

デフォルト : UseFQDN=Y

注 : デフォルトでは、インストール時に SDLIBRARY\$ および SDMSILIB 共有は作成されません (詳細設定で指定されている場合を除く)。

例 :

```
sd_sscmd addshare SDLIBRARY MSILIB
sd_sscmd addshare SDLIBRARY UseFQDN=Y
```

sd_sscmd removeshare

Windows および UNIX で有効です。

コマンド **removeshare** を使用すると、共有 **SDLIBRARY\$** および **SDMSILIB** を削除することができます。このコマンドは、マネージャ上でも実行できます。

コマンドの形式

Windows の場合

```
sd_sscmd removeshare {SDLIBRARY | MSILIB}
```

UNIX の場合

```
sd_sscmd removeshare {SAMBА | NFS | SDLIBRARY | MSILIB}
```

注 : デフォルトでは、インストール時に **SDLIBRARY\$** および **SDMSILIB** 共有は作成されません (詳細設定で指定されている場合を除く)。

例 :

```
sd_sscmd removeshare SDLIBRARY MSILIB
```

sd_sscmd import

Windows および UNIX で有効です。

コマンド **import** を使用すると、SD 4.0 インストールから SD ライブラリをインポートすることができます。このコマンドは、マイグレーションフェーズで使用されます。

コマンドの構文

```
import {[path=path] [move=false | true]}
```

path

SDLIBRARY のルートへのソースパスを指定します。

path を指定しない場合は、SD 4 の SDLIBRARY の検索がローカルマシン上で実行されます。

move

ファイルを元の場所から削除するか (True) 、削除しないか (False) を指定します。

デフォルト : いいえ (False)

aregsw -- ステージング ライブラリでのソフトウェアの自動登録

Windows および UNIX で有効です。

aregsw コマンドを使用すると、標準の形式ですでにパッケージされている新しいソフトウェアプログラム (たとえば、ソフトウェアライブラリから以前にエクスポートされたソフトウェア) を登録することができます。

コマンドの形式

```
aregsw {path=path [reginfo=reginfo logfile=logfile]}
```

path

スケーラビリティ サーバ上で登録対象の項目のイメージが配置されているディレクトリです。**path** エントリは、オペレーティングシステムの表記規則に従う必要があります。

登録情報のサブディレクトリは、このディレクトリ内にある必要があります。

reginfo

登録情報が格納されているディレクトリのパスおよびディレクトリの名前です。

reginfo を指定しない場合、**sd_sscmd** は、**path** 引数で指定されたディレクトリの下で **reginfo** を検索します。

logfile

コマンドの実行中にトレース情報が書き込まれるファイルのパスおよびファイルの名前です。

例：

以下の例は、**Windows** スケーラビリティ サーバから、サブディレクトリ **swtest** にあるソフトウェアを自動的に登録する方法を示しています。

¥swtest¥reginfo で見つかった登録情報を使用して、自動登録が実行されます。

```
sd_sscmd aregsw path=C:¥swtest reginfo=C:¥swtest¥reginfo logfile=C:¥test¥trace1.txt
```

注： **aregsw** を使用する場合、ドメインマネージャのソフトウェアライブラリに含まれないスケーラビリティ サーバのステージング ライブラリに、ソフトウェアを追加しないでください。

このソフトウェアは、次のジョブ確認の実行後、ドメインマネージャの **DSM** エクスプローラの「ステージング ライブラリ」に表示されます。コマンドが正常に完了すると、ジョブ確認が自動的に開始されます。

dereg -- ステージング ライブラリでのソフトウェア項目の登録解除

Windows および UNIX で有効です。

このコマンドは、スケーラビリティ サーバ **SD** ライブラリからソフトウェアを登録解除または除去する場合に使用します。

コマンドの形式

```
dereg {item=item version=version [logfile=logfile]}
```

item

登録解除する項目の名前です。

version

登録解除する項目のバージョンです。

logfile

コマンドの実行中にトレース情報が書き込まれるファイルのパスおよびファイルの名前です。

例：

```
dereg item="My Software Test", version=1.01, logfile=C:\test\trace2.txt
```

verbose

詳細モードとは、特定のコマンドのキーワードとパラメータに対して、それぞれ値を指定するよう要求するインタラクティブ モードのことです。以下の例では、詳細モードで返されるプロンプトを示します。

```
D:\TEST>sd_ssccmd verbose
```

```
SD Server Command Line 11.n.nnn Copyright (c) 2008 CA. All rights reserved.
```

有効なコマンドを次に示します。

```
aregsw
```

```
dereg
```

```
quit
```

command: aregsw

path : D:%test

reginfo (optional) : D:%test%reginfo

logfile (optional) : D:%test%loglb.txt

SD_SSCMD<0>: コマンドは正常に実行されました。

verbose コマンド サポートを有効にするには、最初に以下のコマンドを発行します。

sd_sscmd verbose

リストからコマンドを選択することができます。

コマンドを入力した後、キーワード、および指定したコマンドに関連付けられているパラメータそれぞれに対して、値を入力するように求められます。たとえば、**aregsw** で最初に入力する必要のあるパラメータ値は、**path** です。**path** に対して有効な値を入力した後、次のプロンプト（この例の場合は **logfile**）が表示されます。すべてのパラメータに対して値を入力し終わるまで、プロンプトが表示されます。

batch

Windows および UNIX で有効です。

バッチモードでは、バッチファイルが参照され、その内容が抽出されて実行されます。バッチファイルには、1つのコマンドを含めたり、一連のコマンドを含めることができます。

バッチ処理用のファイルを作成する場合は、ファイル内の各コマンドの前に **sd_sscmd** コマンドを入力しません。**sd_sscmd** コマンドは、バッチファイルが呼び出された後、コマンド実行の始めにのみ発行されます。

コマンド文が含まれる各行は、**1023** 文字を超えないようにする必要があります。

batch コマンドは再帰的に使用できます。たとえば、引数のバッチファイル内に **batch** を記述して別のバッチファイルを参照できます。

コマンドの形式

```
sd_sscmd batch {path = path [logfile=logfile]}
```

バッチ モードを有効にするには、**sd_sscmd** コマンドを入力し、その後にキーワード **batch** とコマンドが含まれるファイルの名前を入力します。たとえば、**Windows** サーバからの場合、以下のように入力します。

```
sd_sscmd batch path =D:¥test¥regswpr.txt logfile=D:¥logs¥log21.txt
```

regswpr.txt には、ソフトウェアを登録するコマンド文が含まれています。

このバッチ ファイルでは、1 つのソフトウェアを登録します。

```
aregsw path=D:¥test2 reginfo=D:¥test2¥reginfo logfile=D:¥logs¥log11.txt
```

行の先頭にある # 以降の文字は、すべてコメントと見なされます。

バッチ処理中のエラーを無視

デフォルトでは、エラーが発生すると、コマンドラインインターフェースはバッチ ファイルの実行を停止します。コマンドラインインターフェース オプションを設定して、エラーを無視し、バッチ ファイル内の次のコマンドの実行を継続することができます。バッチ処理中のエラーを無視するには、該当するコマンドを入力します。

Windows の場合は、以下のように入力します。

```
set sd_sscmd_continue = on
```

C シェルを使用している場合は、以下のように入力します。

```
setenv SD_SSCMD_CONTINUE ON
```

Bourne シェルを使用している場合は、以下のように入力します。

```
SD_SSCMD_CONTINUE=ON export SD_SSCMD_CONTINUE"
```

リターンコード

コマンドラインのすべての管理コマンドは、正常に開始されると 0 (ゼロ) リターンコードを生成し、エラーが発生するとゼロ以外のリターンコードを生成します。コマンドラインの管理コマンドによって生成されるメッセージには、プレフィクス `sd_sscmd` が付けられます。

例：

`sd_sscmd<4>`： コマンドの実行に失敗しました： 項目削除エラーです。

Windows の場合、リターンコードを使用してエラーの原因を特定することができます。UNIX の場合、リターンコードは 0 (成功)、1 (構文エラー)、2 (セマンティックエラー) のいずれかになります。

`sd_sscmd` が実行されると、以下のリターンコードが生成されます。

`sd_sscmd<-1>`： 構文エラー。

`sd_sscmd<0>`： コマンドは正常に実行されました。

`sd_sscmd<1>`： コマンドの実行に失敗しました： ディスク領域が不足しています。

`sd_sscmd<2>`： コマンドの実行に失敗しました： ライブラリパスが指定されていません。

`sd_sscmd<3>`： コマンドの実行に失敗しました： 項目コピーエラーです。

`sd_sscmd<4>`： コマンドの実行に失敗しました： 項目削除エラーです。

`sd_sscmd<5>`： コマンドの実行に失敗しました： ボリュームが見つかりません。

`sd_sscmd<6>`： コマンドの実行に失敗しました： Reg 情報を読み取れませんでした。

`sd_sscmd<7>`： コマンドの実行に失敗しました： Reg 情報が見つかりません。

`sd_sscmd<8>`： コマンドの実行に失敗しました： Reg 情報が一意ではありません。

`sd_sscmd<9>`： コマンドの実行に失敗しました： 内部エラーです。

`sd_sscmd<10>`： コマンドの実行に失敗しました： 文字列変換エラーです。

`sd_sscmd<11>`： ソフトウェアはすでに登録されています。

`sd_sscmd<12>`： ソフトウェアが見つかりません。すでに登録を解除していませんか？

sd_sscmd<13> : パスが長すぎます。

sd_sscmd<14> : 項目名が長すぎます。

sd_sscmd<15> : バージョン文字列が長すぎます。

sd_sscmd<16> : コマンドの実行に失敗しました : バッチ ファイルを開けませんでした。

sd_sscmd<17> : タイムアウトを超過しました : 操作はすでに使用されています

sd_sscmd<18> : コマンドの実行に失敗しました : 削除操作が失敗しました

sd_sscmd<19> : コマンドの実行に失敗しました : 権限を設定できません

sd_sscmd<20> : コマンドの実行に失敗しました : 共有を作成できません

sd_sscmd<21> : コマンドの実行に失敗しました : 共有を削除できません

sd_sscmd<22> : コマンドの実行に失敗しました : サーバが実行していません

sd_sscmd<23> : コマンドの実行に失敗しました : ファイルが見つかりません

sd_sscmd<24> : コマンドの実行に失敗しました : ライブラリが見つかりません

sd_sscmd<25> : コマンドの実行に失敗しました : 以前の **Software Delivery** のインストールが見つかりません

sd_sscmd<26> : コマンドの実行に失敗しました : アイテムが見つかりません

sd_sscmd<27> : 少なくとも 1 つの操作に失敗しました