

VMware ThinApp Package.ini パラメータ リフ アレンス ガイド

2021年7月13日
VMware ThinApp 5.2

最新の技術ドキュメントは、VMware の Web サイト (<https://docs.vmware.com/jp/>)

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

VMware株式会社
105-0013 東京都港区浜松町 1-30-5
浜松町スクエア 13F
www.vmware.com/jp

Copyright © 2018, 2021 VMware, Inc. All rights reserved. [著作権および商標情報。](#)

目次

- 1** 本ガイドについて 7
- 2** パッケージ パラメータの構成 8
- 3** Package.ini ファイルの構造 9
- 4** Package.ini 設定をオーバーライドする Package.ini ファイルまたは ##Attributes.ini ファイル 10
- 5** ThinApp ランタイムの構成 11
 - RuntimeEULA パラメータ 11
 - VirtualComputerName パラメータ 12
 - QualityReportingEnabled パラメータ 13
- 6** 隔離の構成 14
 - DirectoryIsolationMode パラメータ 14
 - RegistryIsolationMode パラメータ 15
- 7** ファイルとプロトコルの関連付けの構成 17
 - FileTypes パラメータ 17
 - Protocols パラメータ 18
- 8** ビルド出力の構成 19
 - ExcludePattern パラメータ 19
 - Icon パラメータ 20
 - OutDir パラメータ 21
 - RetainAllIcons パラメータ 21
- 9** 権限の構成 22
 - AccessDeniedMsg パラメータ 22
 - PermittedGroups パラメータ 22
 - UACRequestedPrivilegesLevel パラメータ 24
 - UACRequestedPrivilegesUIAccess パラメータ 24
- 10** オブジェクトおよび DLL ファイルの構成 25
 - ExternalCOMObjects パラメータ 25
 - ExternalDLLs パラメータ 26
 - ForcedVirtualLoadPaths パラメータ 26

- IsolatedMemoryObjects パラメータ 27
- IsolatedSynchronizationObjects パラメータ 28
- NotificationDLLs パラメータ 28
- NotificationDLLSignature パラメータ 29
- ObjectTypes パラメータ 29
- SandboxCOMObjects パラメータ 30
- VirtualizeExternalOutOfProcessCOM パラメータ 30

- 11** ファイル ストレージの構成 32
 - CachePath パラメータ 32
 - UpgradePath パラメータ 33
 - VirtualDrives パラメータ 34

- 12** プロセスとサービスの構成 36
 - AllowExternalKernelModeServices パラメータ 36
 - AllowExternalProcessModifications パラメータ 37
 - AutoShutdownServices パラメータ 37
 - AutoStartServices パラメータ 37
 - ChildProcessEnvironmentDefault パラメータ 38
 - ChildProcessEnvironmentExceptions パラメータ 38

- 13** サイズの構成 39
 - BlockSize パラメータ 39
 - CompressionType パラメータ 40
 - MSICompressionType パラメータ 41

- 14** ログの構成 42
 - DisableTracing パラメータ 42
 - LogPath パラメータ 42
 - FileTimeLogging 43

- 15** バージョンの構成 44
 - CapturedUsingVersion パラメータ 44
 - StripVersionInfo パラメータ 44
 - Version.XXXX パラメータ 45

- 16** ロケールの構成 46
 - AnsiCodePage パラメータ 46
 - LocaleIdentifier パラメータ 46
 - LocaleName パラメータ 47

17 個々のアプリケーションの構成 48

- CommandLine パラメータ 48
- 無効なパラメータ 49
- ReadOnlyData パラメータ 49
- Shortcut パラメータ 50
- Shortcuts パラメータ 50
- Source パラメータ 51
- WorkingDirectory パラメータ 51

18 アプリケーション ユーティリティを使用した依存アプリケーションの構成 53

- Application Link のパス名の形式 53
- RequiredAppLinks パラメータ 54
- OptionalAppLinks パラメータ 55

19 Application Sync ユーティリティを使用したアプリケーション更新の構成 57

- AppSyncClearSandboxOnUpdate パラメータ 58
- AppSyncExpireMessage パラメータ 58
- AppSyncExpirePeriod パラメータ 58
- AppSyncURL パラメータ 59
- AppSyncUpdateFrequency パラメータ 59
- AppSyncUpdatedMessage パラメータ 59
- AppSyncWarningFrequency パラメータ 60
- AppSyncWarningMessage パラメータ 60
- AppSyncWarningPeriod パラメータ 60

20 MSI ファイルの構成 61

- MSIArpProductIcon パラメータ 61
- MSIDefaultInstallAllUsers パラメータ 62
- MSIFilename パラメータ 63
- MSIInstallDirectory パラメータ 63
- MSIManufacturer パラメータ 64
- MSIProductCode パラメータ 64
- MSIProductVersion パラメータ 64
- MSIRequireElevatedPrivileges パラメータ 65
- MSIUpgradeCode パラメータ 66
- MSIStreaming パラメータ 66
- MSIIs64Bit パラメータ 67

21 サンドボックス ストレージとインベントリ名の構成 68

- InventoryName パラメータ 68
- RemoveSandboxOnExit パラメータ 69

SandboxName	パラメータ	70
SandboxNetworkDrives	パラメータ	70
SandboxPath	パラメータ	71
SandboxRemovableDisk	パラメータ	71
SandboxWindowClassName	パラメータ	72

22 その他の構成パラメータ 73

DisableCutPaste	パラメータ	74
LoadDotNetFromSystem	パラメータ	74
PermittedComputers	パラメータ	74
Services	パラメータ	74
StatusbarDisplayName	パラメータ	74
DisableTransactionRegistry	パラメータ	74
PreventDLLInjection	74	
ProcessExternalNameBehavior	パラメータ	75
PreventDllInjectionExceptions	パラメータ	75
LargeAddressAware	パラメータ	75
PermittedComputers	パラメータ	75
PermittedComputersAccessDeniedMsg	パラメータ	76
PermittedComputersOfflineAccess	パラメータ	76
IgnoreDDEMessages	パラメータ	76
Comment	パラメータ	76
CopyManifestData	パラメータ	77
DllNamesMap	77	
EnableExternalHookInterception	77	
IgnoreFolderSecurity	78	
DisableWriteDAC	78	
AddIMESupport	78	

本ガイドについて

1

『ThinApp Package.ini パラメータ リファレンス ガイド』には、ThinApp Package.ini パラメータを構成およびカスタマイズする方法に関する情報が記載されています。このガイドを参照して、キャプチャ プロセスの外部にある仮想アプリケーションのパラメータをカスタマイズすることができます。

対象者

本書は、キャプチャしたアプリケーションの Package.ini パラメータをカスタマイズする必要があるユーザーを対象としています。企業ソフトウェア パッケージの配布とメンテナンスを担当するシステム管理者が、通常のユーザーとなります。

パッケージ パラメータの構成

2

上級ユーザーは、キャプチャ プロセスの外部で、仮想アプリケーションのパラメータをカスタマイズできます。パラメータは、MSI、更新、およびエントリ ポイントの設定を含む構築オプションの設定に影響する可能性があります。

Package.ini ファイルは、仮想アプリケーションのプロジェクト フォルダに配置されています。このファイルには、構築中にキャプチャしたアプリケーションを構成するパラメータが含まれています。セットアップ キャプチャ ウィザードでは、一部の Package.ini パラメータの初期値が設定されます。変更を Package.ini ファイルに保存し、プロジェクトを構築してパラメータの変更を有効にすることができます。

Package.ini ファイルの構造

3

Package.ini ファイルの構造には、すべてのアプリケーション、または個別のアプリケーションにそれぞれ適用されるセクションが含まれています。

大部分のパラメータは、特定のセクション見出しの下に記述する必要があります。Package.ini ファイルには、次の見出しが含まれています。

[BuildOptions]

Package.ini ファイルの [BuildOptions] セクションは、すべてのアプリケーションに適用されます。個々のアプリケーションは、アプリケーション固有のエントリによって既存の設定がオーバーライドされない限り、各パラメータを継承します。たとえば、Adobe Reader アプリケーションに適用される Package.ini ファイルの [Adobe Reader 8.exe] セクションには、これより大きい [BuildOptions] パラメータをオーバーライドする設定が含まれている場合があります。アプリケーション固有のパラメータには、ビルド プロセスで作成されたアプリケーションのエントリ ポイントが表示されます。

[<application>.exe]

[FileList]

[FileList] パラメータは [BuildOptions] パラメータとして機能します。これらのパラメータは、後方互換性を保持するために、その他の [BuildOptions] パラメータとは別に分類されています。見出し [FileList] は、パラメータ ExcludePattern を追加するときにファイルに手動で追加できます。

[圧縮]

[Compression] パラメータは [BuildOptions] パラメータとして機能します。これらのパラメータは、後方互換性を保持するために、その他の [BuildOptions] パラメータとは別に分類されています。

[隔離]

[Isolation] パラメータは [BuildOptions] パラメータとして機能します。これらのパラメータは、後方互換性を保持するために、その他の [BuildOptions] パラメータとは別に分類されています。

Package.ini 設定をオーバーライドする Package.ini ファイルまたは ##Attributes.ini ファイル

4

要件によって Package.ini または ##Attributes.ini ファイルには、ディレクトリ レベルで Package.ini の設定をオーバーライドする特定のパラメータを適用することができます。

##Attributes.ini ファイルで DirectoryIsolationMode、CompressionType、および ExcludePattern の各パラメータを使用することにより、Package.ini の設定をディレクトリ レベルでオーバーライドできます。

##Attributes.ini ファイルは、プロジェクト フォルダのフォルダ マクロにあります。

##Attributes.ini ファイル設定の変更の詳細については、『ThinApp ユーザー ガイド』を参照してください。

ThinApp ランタイムの構成

5

アプリケーションの起動パフォーマンスおよび仮想コンピュータ名に影響を与えるランタイム構成タスクの ThinApp パラメータを変更できます。

手順

1 RuntimeEULA パラメータ

RuntimeEULA パラメータは、パッケージのエンドユーザー使用許諾契約書の表示を制御します。このパラメータは、レガシー EULA の要件に対応しています。VMware では、ThinApp パッケージのランタイム EULA は必要ありません。

2 VirtualComputerName パラメータ

VirtualComputerName パラメータは、コンピュータ名を変更するかどうかを決定します。コンピュータ名の変更により、キャプチャ プロセスと展開プロセスの名前の競合を避けられます。

3 QualityReportingEnabled パラメータ

QualityReportingEnabled パラメータは、ThinApp アプリケーション サポートの向上を図るために、VMware がパッケージで匿名データを収集するかどうかを指定します。VMware は、バージョンやアプリケーション エラーの数などのアプリケーション情報を収集します。

RuntimeEULA パラメータ

RuntimeEULA パラメータは、パッケージのエンドユーザー使用許諾契約書の表示を制御します。このパラメータは、レガシー EULA の要件に対応しています。VMware では、ThinApp パッケージのランタイム EULA は必要ありません。

注： このパラメータの値は変更しないでください。

例：RuntimeEULA のデフォルト値

この例は、RuntimeEULA パラメータによってエンドユーザー使用許諾契約書の表示がどのように抑制されるかを示します。

```
[BuildOptions]
;Default: do not show an Eula
RuntimeEULA=0
```

VirtualComputerName パラメータ

VirtualComputerName パラメータは、コンピュータ名を変更するかどうかを決定します。コンピュータ名の変更により、キャプチャ プロセスと展開プロセスの名前の競合を避けられます。

アプリケーションは、自身がインストールされているコンピュータの名前を使用したり、データベースに接続し、接続文字列にそのコンピュータ名を使用したりすることができます。キャプチャ プロセスは展開プロセスと異なるため、コンピュータ名を必要とするキャプチャされたアプリケーションでは、コンピュータ名を仮想パッケージに追加して、どのマシンでもアプリケーションを実行できるようにする必要があります。

ThinApp では、VirtualComputerName パラメータの初期設定がコメントアウトされています。このパラメータは、GetComputerName と GetComputerNameEx API 関数が仮想アプリケーションに返す文字列を使用します。

例： 仮想コンピュータ名を含める

この例では、VirtualComputerName パラメータで LOCALHOST という名前のコンピュータに 2 番目の名前を作成する方法を示しています。これは、仮想アプリケーションでキャプチャされます。アプリケーションは、2 番目の名前を使用して仮想マシンに接続します。キャプチャ システムに LOCALHOST 名がない場合、ThinApp は VirtualComputerName パラメータをコメントアウトします。

```
;VirtualComputerName=<original_machine_name>
```

キャプチャ プロセスを実行する前にクリーン マシンの名前を LOCALHOST に変更すると、Package.ini ファイルは、VirtualComputerName パラメータで作成された名前をアクティブ化します。仮想アプリケーションは、名前が変更された LOCALHOST 名で動作します。これは、アプリケーションが実行されるコンピュータがこの値をコンピュータ名として取得するためです。

GetComputerName または GetComputerNameEx コマンドを実行すると、コンピュータは LOCALHOST と返します。Windows システムの標準の操作で、アプリケーションが実行されているコンピュータの実際の名前を返すために GetComputerName および GetComputerNameEx コマンドが必要な場合、マシンの名前を LOCALHOST に変更しないでください。

```
VirtualComputerName=LOCALHOST
```

例： 環境変数を含める

LOCALHOST などのリテラル文字列だけでなく、環境変数を指定することもできます。

環境変数を指定すると、返される値は環境変数の値になります。VirtualComputerName パラメータの値が %VCOMPNAME% で、%VCOMPNAME% 環境変数が EnvCompName に設定されている場合、GetComputerName API から EnvCompName が返されます。

```
VirtualComputerName=%VCOMPNAME%
```

QualityReportingEnabled パラメータ

QualityReportingEnabled パラメータは、ThinApp アプリケーション サポートの向上を図るために、VMware がパッケージで匿名データを収集するかどうかを指定します。VMware は、バージョンやアプリケーション エラーの数などのアプリケーション情報を収集します。

キャプチャ プロセスで匿名データの収集に同意すると、QualityReportingEnabled パラメータにより 10 日に 1 回の頻度でデータがアップロードされます。

例： データ収集の無効化

QualityReportingEnabled パラメータを変更して、ThinApp のデータ収集を無効にすることができます。

```
[BuildOptions]
QualityReportingEnabled=0
```

隔離の構成

6

ThinApp の隔離パラメータによって、ファイル システムおよびレジストリ キーへの読み取りおよび書き込みアクセスが決定されます。

この章には、次のトピックが含まれています。

- [DirectoryIsolationMode パラメータ](#)
- [RegistryIsolationMode パラメータ](#)

DirectoryIsolationMode パラメータ

DirectoryIsolationMode パラメータは、物理ファイル システムへのディレクトリの読み取りおよび書き込みアクセスのレベルを指定します。

キャプチャ プロセスでは、Package.ini ファイルの DirectoryIsolationMode パラメータの初期値を設定します。このパラメータは、個々のディレクトリの ##Attributes.ini ファイルで別の隔離モードを指定した場合を除き、仮想アプリケーションによって作成されるファイルのデフォルトの隔離モードを制御します。C:\myfolder などの未指定のディレクトリはすべて、Package.ini ファイルから隔離モードを継承します。

ThinApp は、キャプチャ プロセスで *Merged* と *WriteCopy* 隔離モードのオプションのみを提供します。セットアップ キャプチャ ウィザードの外部で完全隔離モードを使用して、仮想環境を保護することができます。

Merged 隔離モードでは、アプリケーションは仮想パッケージの外部で物理ファイル システムの要素の読み取りと変更を行うことができます。一部のアプリケーションは、ローカル システム イメージ内の DLL およびレジストリ情報の読み取りに依存しています。Merged モードを使用するメリットは、ユーザーが保存したドキュメントがサンドボックスではなく、ユーザーが想定している場所の物理システムに表示されることです。このモードの欠点は、システム イメージが乱雑になる可能性があることです。乱雑化の例としては、ライセンス プロセスの一部としてコンピュータのランダムな場所に書き込まれたシェアウェア アプリケーションの初回実行マーカーがあります。

WriteCopy 隔離モードでは、ThinApp は書き込み操作を傍受し、サンドボックスにリダイレクトすることができます。レガシー アプリケーションまたは信頼されていないアプリケーションに対して、WriteCopy 隔離モードを使用できます。このモードは、物理システムではなくサンドボックスに存在するユーザー データ ファイルを見つけることを困難にする可能性がありますが、ユーザーがローカル ファイル システムに影響を与えないようにするためにロックダウンされたデスクトップでは役立ちます。

Full 隔離モードでは、ThinApp は、仮想アプリケーション パッケージの外部のシステム要素に対する可視性をブロックします。このモードでは、ファイルまたはレジストリ キーに対する変更がサンドボックスに制限され、仮想アプリケーション パッケージの外部の環境との相互作用がないようになります。Full 隔離により、仮想アプリケーションと、物理システムにインストールされているアプリケーションとの間でのアプリケーションの競合を回避できます。このモードでは、システム DLL を検出およびロードする機能がブロックされるため、Package.ini ファイルで Full 隔離モードは使用しないでください。Full 隔離モードは、##Attributes.ini ファイルのオーバーライド メカニズムとして使用できます。

ThinApp は、ランタイムでのレジストリおよびファイル システムの隔離モードをサンドボックスにキャッシュします。プロジェクトの隔離モードを変更し、実行可能ファイルを再ビルドする場合、それらの変更内容を反映させるためにサンドボックスを削除することがあります。

パラメータは [Isolation] の見出しの下に配置する必要があります。

##Attributes.ini ファイル設定の変更の詳細については、『ThinApp ユーザー ガイド』を参照してください。

例： WriteCopy 隔離の使用

この例では、WriteCopy 隔離を使用して DirectoryIsolationMode パラメータを変更し、仮想アプリケーションによるローカル マシンのリソースの読み取りを可能にしますが、ホスト コンピュータへの書き込みはできないようにします。これは、snapshot.exe ユーティリティのデフォルト設定です。

```
[Isolation]
DirectoryIsolationMode=WriteCopy
```

例： マージ隔離の使用

この例では Merged 隔離モードを割り当てて、パッケージで指定されている場合を除き、仮想アプリケーションがリソースを読み取ることと、コンピュータ上の任意の場所に書き込むことができるようにします。Merged は、セットアップ キャプチャ ウィザードのデフォルト設定です。

```
[Isolation]
DirectoryIsolationMode=Merged
```

RegistryIsolationMode パラメータ

RegistryIsolationMode パラメータは、パッケージ内のレジストリ キーの隔離モードを制御します。この設定は、明示的な設定のないレジストリ キーに適用されます。

キャプチャ プロセスでは、このパラメータの値は設定されません。レジストリの隔離モードは、Package.ini ファイルでのみ構成できます。

ThinApp では、初期のレジストリ隔離モードは WriteCopy に設定されます。

このモードでは、システム DLL を検出およびロードする機能がブロックされるため、Package.ini ファイルで Full 隔離モードは使用しないでください。Full 隔離モードは、オーバーライド メカニズムとして使用できます。プロジェクト ディレクトリにあるレジストリ キーのテキスト ファイルで、構成された RegistryIsolationMode パラメータに例外を記述することができます。例外は、HKEY_CURRENT_USER.txt などのファイルに、isolation_full HKEY_CURRENT_USER\Software\Macromedia として表示されることがあります。

キャプチャされたアプリケーションの仮想ファイルに対する実行時の変更は、隔離モードの設定とは関係なくすべてサンドボックスに保存されます。実行時には、仮想レジストリ ファイルと物理レジストリ ファイルの区別はアプリケーションには認識されません。ただし、両方のレジストリ ファイルが同じ場所に存在する場合は、必ず物理レジストリ ファイルより仮想レジストリ ファイルが優先されます。仮想エントリと物理エントリが同じ場所に存在する場合、アプリケーションでは必ず仮想コンポーネントとやり取りを行うため、隔離モードはこれらのエントリへのアクセスに影響することはありません。

パッケージとは別個に物理レジストリを介して外部のグループ ポリシーの更新が行われる場合は、パッケージから仮想レジストリ ファイルを削除し、各仮想レジストリ ファイルの親ファイルで Full の隔離が使用されていないことを確認することができます。子ファイルで親コンポーネントから隔離モードが継承されるため、親ファイルに Full の隔離が設定されていると、アプリケーションが子の物理ファイルを参照できないようにブロックされる可能性があります。

例：レジストリでの WriteCopy 隔離の使用

この例では、RegistryIsolationMode パラメータを変更して、アプリケーションでホスト コンピュータからのキー読み取りは許可し、ホスト コンピュータへの書き込みは禁止する方法を示します。

```
[Isolation]
RegistryIsolationMode=WriteCopy
```

例：レジストリでの Merge 隔離の使用

この例では、パッケージで特に指定がない限り、アプリケーションがコンピュータ上に任意のキーで書き込めるようにする方法を示します。

```
[Isolation]
RegistryIsolationMode=Merged
```


ファイルとプロトコルの関連付けの構成

7

ThinApp パラメータを変更して、ファイル拡張子をアプリケーションに関連付けたり、物理環境に対して表示可能なプロトコルを指定したりすることができます。

この章には、次のトピックが含まれています。

- [FileTypes パラメータ](#)
- [Protocols パラメータ](#)

FileTypes パラメータ

FileTypes パラメータには、thinreg.exe ユーティリティが実行可能ファイルに関連付けるファイル拡張子がリストされます。

キャプチャ プロセスで生成される初期値に、ユーザーが追加することはできません。仮想パッケージに関連付けない拡張子は削除できます。リスト内のファイル拡張子の間に、区切り文字を使用しないでください。

例：物理環境内のアプリケーションのバージョンでパッケージを開けるようにするための、パッケージからのファイル拡張子の削除

Microsoft Word 2007 パッケージは、FileTypes パラメータの値として .doc.docx を指定します。Microsoft Office 2007 をキャプチャして、物理環境に Microsoft Office 2003 がインストールされている場合、.doc 拡張子を FileTypes パラメータから削除し、.docx 拡張子をそのままにして、確実に Microsoft Word 2003 で .doc ファイルが開き、Microsoft Word 2007 で .docx ファイルが開けるようにします。

```
[Microsoft Office Word 2007.exe]
FileTypes=.docx
```

例：ファイル タイプの拡張子の作成とアプリケーションへのリンク

キャプチャ プロセスでは、.doc および .dot 拡張子のファイル タイプの関連付けを作成し、それらを Microsoft Word にリンクさせることができます。

```
[Microsoft Office Word 2003.exe]
ReadOnlyData=bin\Package.ro.tvr
Source=%ProgramFilesDir%\Microsoft Office\OFFICE11\WINWORD.EXE
FileTypes=.doc.dot
```

Protocols パラメータ

Protocols パラメータは、物理環境でアプリケーションが認識できるプロトコル（HTTP など）を指定します。このパラメータは FileTypes パラメータと似ていますが、ファイル タイプではなくプロトコルを扱うアプリケーションに作用します。

キャプチャ プロセスで生成される初期値に、ユーザーが追加することはできません。ブラウザやその他のアプリケーションのエントリを削除できます。

例： Microsoft Outlook パッケージの mailto プロトコルの指定

キャプチャ プロセスでは、Protocols パラメータで、Microsoft Outlook パッケージに対する mailto プロトコルなどのプロトコルを指定できます。

```
[Microsoft Office Outlook 2007.exe]
Protocols=feed;feeds;mailto;Outlook.URL.mailto;stssync;webcal;webcals
```

ビルド出力の構成

8

ThinApp パラメータを変更して、ビルド出力の場所とパッケージ内のファイルの場所を指定することができます。

この章には、次のトピックが含まれています。

- [ExcludePattern パラメータ](#)
- [Icon パラメータ](#)
- [OutDir パラメータ](#)
- [RetainAllIcons パラメータ](#)

ExcludePattern パラメータ

ExcludePattern パラメータは、アプリケーションのビルド プロセスでファイルまたはディレクトリを除外します。

このパラメータのエントリの前に [FileList] の見出しを追加する必要があります。

リスト内のパターンは、カンマで区切ることができます。ワイルドカード (*) は、どの文字とも一致しないか、文字と疑問符 (?) のうちの少なくとも 1 つの文字と一致します。構文は、DOS dir コマンドと似ていますが、ディレクトリ名とファイル名にワイルドカード文字を適用することができます。

Package.ini ファイルで ExcludePattern パラメータを指定できます。このファイルでは、パターン除外がディレクトリ構造全体に適用されます。##Attributes.ini ファイルでは、ThinApp によってパターン除外が現在の除外リストに追加されますが、特定のディレクトリとサブディレクトリにのみ設定が適用されます。プロジェクト内の異なるディレクトリに対して別の除外リストを作成できます。

例：仮想ファイル システムからのバージョン管理情報の除外

パッケージをバージョン管理システムに格納し、仮想ファイル システムからバージョン管理情報を除外する場合、.svn または .cvs と呼ばれるディレクトリと、すべてのサブディレクトリを除外することができます。

```
[FileList]
ExcludePattern=\.svn,\.cvs
```

パターンは、文字列の中間部分に .svn や .cvs を含むファイル名またはディレクトリと一致しません。

例：.bak または .msi で終わるパスの除外

.bak または .msi で終わる任意のパスを除外することができます。

```
[FileList]
ExcludePattern=*.bak,*.msi
```

Icon パラメータ

Icon パラメータは、生成された実行可能ファイルに関連付けるアイコン ファイルを指定します。このアイコンは、Microsoft Word などのアプリケーション、およびアプリケーションに関連付けられているファイル（.doc ファイルなど）に表示されます。

各アプリケーションには、独自のアイコンがあり、アプリケーションの .exe ファイルまたは .dll ファイル内に .ico ファイルとして格納されています。

キャプチャ プロセスにより、実行可能ファイルにアイコンが関連付けられます。

アプリケーションは、Source パラメータ内の実行可能ファイルからのメイン グループ アイコンと、グループ アイコンがポイントしている個々のアイコン リソースを使用します。

例：代替アイコンの使用

Source パラメータ内の実行可能ファイルとは異なる実行可能ファイルを指定することにより、Icon パラメータを変更して代替アイコンを使用することができます。

代替アイコンは、サードパーティ企業にとって便利なものとなる可能性があります。

```
[<my_app>.exe]
Source=%ProgramFilesDir%\<my_app>\app.exe
Icon=%ProgramFilesDir%\<my_app>\app2.exe
```

例：アイコンの指定

アイコン セットを指定するには、**1**、**2** をアイコン パスの末尾に追加します。

```
[<my_app>.exe]
Source=%ProgramFilesDir%\<my_app>\<app>.exe
Icon=%ProgramFilesDir%\<my_app>\<app2>.exe,1
```

例：ICO ファイルの使用

.ico ファイルを使用して、アプリケーションのアイコンを指定できます。

```
[<my_app>.exe]
Source=%ProgramFilesDir%\<my_app>\<app>.exe
Icon=%ProgramFilesDir%\<my_app>\<my_icon>.ico
```

OutDir パラメータ

OutDir パラメータは、build.bat の出力を格納するディレクトリを指定します。このパラメータの値は変更しないでください。

例：プロジェクトの bin ディレクトリの指定

OutDir パラメータの静的な値は、プロジェクトの bin ディレクトリを指定します。

```
[BuildOptions]
OutDir=bin
```

RetainAllIcons パラメータ

RetainAllIcons パラメータは、アプリケーションの Source パラメータにリストされている実行可能ファイルの元のアイコンをすべて保持します。

アプリケーションの実行可能ファイルに割り当てられていないアイコンは、パッケージの仮想ファイル システムに配置されます。RetainAllIcons パラメータは、未使用のアイコンを仮想ファイル システムから実行可能ファイルにコピーするかどうかを決定します。

ディスク容量の節約のために、ThinApp は、物理環境で認識できる実行可能ファイルの一部から未使用のアイコンを削除する初期値を設定します。

例：アプリケーションの元のアイコンをすべて保持

RetainAllIcons パラメータを変更して、アプリケーションの元のアイコンをすべて保持することができます。

```
[app.exe]
Source=%ProgramFilesDir%\myapp\app.exe
RetainAllIcons=1
```

権限の構成

9

パッケージへのユーザー アクセスを定義し、データ実行防止 (DEP) の保護を変更するセキュリティ タスクのパラメータを変更できます。

権限のパラメータは、通常 [build options] セクションで指定されます。

この章には、次のトピックが含まれています。

- [AccessDeniedMsg パラメータ](#)
- [PermittedGroups パラメータ](#)
- [UACRequestedPrivilegesLevel パラメータ](#)
- [UACRequestedPrivilegesUIAccess パラメータ](#)

AccessDeniedMsg パラメータ

AccessDeniedMsg パラメータには、パッケージを実行する権限がないユーザーに対して表示されるエラー メッセージが含まれています。

ThinApp は、管理者に連絡するようユーザーに通知するデフォルト メッセージを設定します。

例：連絡先電話番号の追加

AccessDeniedMsg パラメータを変更して、テクニカル サポートの電話番号を追加することができます。

```
[BuildOptions]
PermittedGroups=Administrator;OfficeUsers
AccessDeniedMsg=You do not have permission to execute this application, please call support
@1-800-822-2992
```

PermittedGroups パラメータ

PermittedGroups パラメータは、パッケージを特定の Active Directory ユーザー セットに制限します。

PermittedGroups パラメータの同じ行にグループ名、SID 文字列、またはグループ名と SID 文字列の組み合わせを指定できます。ドメイン ベースのグループ名を使用する場合は、アプリケーション パッケージのビルド時に、そのドメインに接続する必要があります。パラメータ値に SID を追加する場合は、SID が定義されているドメインに接続する必要はありません。

Active Directory ドメイン サービスでは、セキュリティ グループと配布グループが定義されます。このパラメータは、ネストされたセキュリティ グループのみをサポートします。たとえば、あるユーザーがセキュリティ グループ A のメンバーで、セキュリティ グループ A がセキュリティ グループ B のメンバーになっている場合、ThinApp はこのユーザーを、セキュリティ グループ A とセキュリティ グループ B のメンバーとして検出できます。

ThinApp でアプリケーションをビルドする場合、指定されたグループ名がいずれも有効であることを前提とし、それぞれの名前を SID 値に変換します。ThinApp は、キャッシュされた認証情報を使用して、実行時にグループ所有権を解決できます。ラップトップ ユーザーがオフラインになっている場合でも、引き続き認証を行うことができます。ユーザーにパッケージを実行するためのアクセス権がない場合、AccessDeniedMsg パラメータをカスタマイズしてユーザーに指示することができます。

PermittedGroups パラメータを見出し [BuildOptions] の下に配置してパッケージに適用することも、見出し [<application>.exe] の下に配置して特定のアプリケーションに適用することもできます。該当するアプリケーションのデフォルト値 [BuildOptions] は、 [<application>.exe] の値でオーバーライドされます。

例： Active Directory メンバー リストの指定

PermittedGroups パラメータを変更して、Active Directory ユーザー グループ名のリストを指定することができます。リストは、セミコロンで区切って記述します。[BuildOptions] セクションに示す各パラメータにより、プロジェクト全体のグローバル設定を指定します。

```
[BuildOptions]
PermittedGroups=Administrator;OfficeUsers
AccessDeniedMsg=You do not have permission to execute this application, please call support @
1-800-822-2992
```

例： PermittedGroups グローバル設定の上書き

グローバル設定 PermittedGroups を上書きする特定のアプリケーションに適用するユーザー グループの設定を指定できます。

```
[App1.exe]
PermittedGroups=Guest
AccessDeniedMsg=You do not have permission to execute this application, please call support @
1-800-822-2992
```

例： グローバル設定 PermittedGroups の継承

アプリケーションに設定 PermittedGroups を指定しない場合、アプリケーションでは [BuildOptions] セクションのグローバル値 PermittedGroups が継承されます。

```
[App2.exe]
...
```

例： グループ名と SID 文字列の混在

PermittedGroups パラメータの同じエントリでグループ名と SID 文字列を混在させることができます。

```
PermittedGroups=S-1-5-32-544;Office Users
```

UACRequestedPrivilegesLevel パラメータ

UACRequestedPrivilegesLevel パラメータは、ユーザー アカウント制御 (UAC : User Account Control) 情報を必要とするプログラムに対する権限を指定します。このパラメータは、Windows Vista 以降のバージョンのオペレーティング システムで作業するユーザーに影響します。

以下の値を使用して権限を指定できます。

- asInvoker

この値を指定すると、Vista のプロファイルが使用されます。

- requireAdministrator

- highestAvailable

この値を指定すると、使用可能な権限のうち最も高い権限が使用され、UAC プロンプトの表示を避けることができます。

権限を指定しない場合、ThinApp はデフォルト値を割り当てませんが、asInvoker 設定に従って機能します。

例：アプリケーションの管理者権限の指定

UACRequestedPrivilegesLevel パラメータを変更して、アプリケーションの管理者権限を指定することができます。

```
[BuildOptions]
UACRequestedPrivilegesLevel=requireAdministrator
```

UACRequestedPrivilegesUIAccess パラメータ

UACRequestedPrivilegesUIAccess パラメータは、Windows Vista 以降のバージョンのオペレーティング システムでのユーザー インターフェイスへのアクセス権を指定します。これらのオペレーティング システムでは、ユーザー インターフェイスの一部の要素が保護されます。

ThinApp では、アプリケーションが保護された要素にアクセスできないように

UACRequestedPrivilegesUIAccess パラメータの初期値が割り当てられています。

UACRequestedPrivilegesUIAccess パラメータには true または false の値を割り当ててユーザー インターフェイスへのアクセス権を指定できますが、このパラメータは Microsoft の設定をサポートするために存在しています。

例：アプリケーションが保護された要素にアクセスできないようにする

UACRequestedPrivilegesUIAccess パラメータの初期値を保持して、仮想アプリケーションが保護された要素にアクセスできないようにすることができます。

```
[BuildOptions]
UACRequestedPrivilegesUiAccess=false
```


オブジェクトおよび DLL ファイルの構成

10

ThinApp パラメータを変更して、COM オブジェクトのアクセスおよび DLL 読み込みの要件を指定できます。

この章には、次のトピックが含まれています。

- [ExternalCOMObjects](#) パラメータ
- [ExternalDLLs](#) パラメータ
- [ForcedVirtualLoadPaths](#) パラメータ
- [IsolatedMemoryObjects](#) パラメータ
- [IsolatedSynchronizationObjects](#) パラメータ
- [NotificationDLLs](#) パラメータ
- [NotificationDLLSignature](#) パラメータ
- [ObjectTypes](#) パラメータ
- [SandboxCOMObjects](#) パラメータ
- [VirtualizeExternalOutOfProcessCOM](#) パラメータ

ExternalCOMObjects パラメータ

ExternalCOMObjects パラメータは、ThinApp とのアプリケーションの互換性を高めるために、Windows が仮想環境ではなく物理環境で COM オブジェクトを作成して実行するかどうかを指定します。

仮想環境の外部にある COM オブジェクトは、常に物理環境で実行されます。

ThinApp は、仮想環境で COM オブジェクトを作成および実行する ExternalCOMObjects パラメータの初期値を設定します。

COM は、プロセス外の実行可能なサーバおよびサービスベースの COM オブジェクトをサポートします。アプリケーションがホスト コンピュータで変更を生成する COM オブジェクトを作成できる場合、ホスト コンピュータの整合性が損なわれるおそれがあります。ThinApp は、仮想環境でプロセス外とサービスベースの COM オブジェクトを実行する場合、COM オブジェクトが行ったすべての変更をサンドボックスに保存します。

キャプチャ プロセスでは、このパラメータは生成されません。このパラメータは、[BuildOptions] セクションの Package.ini ファイルに追加できます。

例：仮想環境外での COM オブジェクトの実行

VMware サポートを利用して問題のトラブルシューティングを行い、アプリケーションが ThinApp と互換性のない COM オブジェクトを実装していることが判明した場合は、仮想環境外で COM オブジェクトを実行するように ExternalCOMObjects パラメータを変更することができます。

CLSID キーをリストすることができます。

```
[BuildOptions]
ExternalCOMObjects={8BC3F05E-D86B-11D0-A075-00C04FB68820};{7D096C5F-AC08-4F1F-BEB7-5C22C517CE39}
```

ExternalDLLs パラメータ

ExternalDLLs パラメータは、Windows に仮想ファイル システムから特定の DLL ファイルを強制的にロードさせることができます。

ThinApp が設定する初期値では、仮想ファイル システムから DLL ファイルをロードし、物理ファイル システム上の DLL ファイルのロード プロセスを Windows に渡すことになります。状況によっては、Windows が仮想ファイル システムに DLL ファイルをロードする必要があります。Windows フックを使用して他のプロセスに自身を挿入する DLL ファイルがある場合があります。フックを実装する DLL ファイルは、ホスト ファイル システムで使用可能であり、Windows によってロードされる必要があります。ExternalDLLs パラメータで DLL ファイルを指定すると、ThinApp は仮想ファイル システムからサンドボックスにファイルを抽出し、そのファイルをロードするように Windows に指示します。

仮想ディクテーション ソフトウェアは、DLL 間で情報を渡すネイティブ アプリケーションと連動する可能性のあるソフトウェアの一種です。ThinApp は、ローカル アプリケーションが DLL と確実に通信できるように、仮想環境内の DLL のロードを Windows に渡すことができます。

ExternalDLLs パラメータは、仮想ファイル システム内の他の DLL ファイルに依存する DLL ファイルをサポートしていません。この場合、Windows はその DLL ファイルをロードできません。

例：Windows に対する仮想ファイル システムからの DLL ファイルのロードの強制

ExternalDLLs パラメータを変更して、Windows に inject.dll および injectme2.dll を仮想ファイル システムから強制的にロードさせることができます。

```
[BuildOptions]
ExternalDLLs=inject.dll;injectme2.dll
```

ForcedVirtualLoadPaths パラメータ

ファイルがパッケージの外部に存在する場合でも、ForcedVirtualLoadPaths パラメータにより、ThinApp が DLL ファイルを仮想 DLL ファイルとしてロードするように指示できます。このパラメータは、パッケージ内にある DLL ファイルに依存する外部システム DLL ファイルをアプリケーションがロードする必要がある場合に便利です。

DLL パスには、マクロを含めることができます。複数のパスを入力する場合は、セミコロンで区切ります。

このパラメータは、AddForcedVirtualLoadPath API 関数と同じ結果を実現します。を参照してください。

例：ThinApp に対する仮想環境での DLL のロードの強制

外部 DLL ファイルに依存するアプリケーションがある場合に、ForcedVirtualLoadPaths パラメータを変更できます。

Microsoft Outlook を含めない Microsoft Office をキャプチャし、Microsoft Outlook のネイティブバージョンがローカル システムに存在する場合、Microsoft Outlook とともにインストールされるネイティブ envelope.dll ファイルは仮想環境で ThinApp によってロードされる mso.dll ファイルに依存するため、Microsoft Excel の仮想バージョンから E メールを送信することはできません。ThinApp が envelope.dll ファイルをネイティブ環境ではなく仮想環境にロードするように強制できます。

```
[BuildOptions]
ForcedVirtualLoadPaths=%ProgramFilesDir%\Microsoft Office\Office10\envelope.dll
```

IsolatedMemoryObjects パラメータ

IsolatedMemoryObjects パラメータには、他のアプリケーションやシステム オブジェクトから隔離する共有メモリ オブジェクトがリストされます。

CreateFileMapping および OpenFileMapping Windows 機能を使用するアプリケーションでは、共有メモリ オブジェクトが作成されます。メモリ オブジェクトを隔離しないと、これらのオブジェクトを共有している仮想アプリケーションとネイティブ アプリケーション間で競合が発生する可能性があります。

たとえば、アプリケーションの 2 つのバージョンがあり、ネイティブ環境に 1 つのバージョン、仮想環境に 1 つのバージョンが含まれているとします。これらのアプリケーション バージョンが同じメモリ オブジェクト内の情報を使用すると、アプリケーションが互いに干渉して失敗することがあります。共有メモリ オブジェクトを隔離して、仮想アプリケーションとシステム オブジェクトが相互に検出できないようにすることができます。

このパラメータは Package.ini ファイルには表示されませんが、パラメータを追加することができます。

ThinApp が設定する初期値では、ネイティブ バージョンの Internet Explorer が仮想環境で使用するメモリ オブジェクトが隔離されます。この値により、explorer.exe および iexplore.exe ユーティリティがサンドボックス ファイルをマッピングするときの競合が回避されます。

IsolatedMemoryObjects パラメータを使用して、追加の名前付き共有メモリ オブジェクトを隔離し、同じサンドボックスを使用して他の仮想アプリケーションにのみオブジェクトが表示されるようにすることができます。

IsolatedMemoryObjects パラメータは、セミコロン (;) 区切りのエントリのリストを受け入れます。

このセクションでは、以下のパラメータについて説明します。

例：メモリ オブジェクトの隔離

IsolatedMemoryObjects パラメータを変更して、My Shared Object 名と、名前に outlook を含んでいるあらゆるメモリ オブジェクトを隔離することができます。

```
[BuildOptions]
IsolatedMemoryObjects=*outlook*;My Shared Object
```

IsolatedSynchronizationObjects パラメータ

IsolatedSynchronizationObjects パラメータには、他のアプリケーションから隔離する同期オブジェクトがリストされます。

同期オブジェクトは、アプリケーション間のアクションを調整します。次の Windows 同期オブジェクトが、アプリケーション エラーのログに表示されることがあります。

- OpenMutex
- CreateMutex
- OpenSemaphore
- CreateSemaphore
- OpenEvent
- CreateEvent

これらのオブジェクトがログ ファイルに表示されている場合は、ネイティブ アプリケーションが作成する同期オブジェクトとの競合を回避するために、仮想環境内のオブジェクトを隔離します。

同期オブジェクトは、同一の仮想名前空間で実行されていないアプリケーションから分離できます。2 つのアプリケーションが同じサンドボックス パスを共有している場合、アプリケーションは隔離された同期オブジェクトに対して同じ名前空間を使用します。2 つのアプリケーションのサンドボックス名が同じで、サンドボックス パスが異なる場合、アプリケーションは別個の名前空間を使用します。

IsolatedSynchronizationObjects パラメータは Package.ini ファイルには表示されませんが、パラメータを追加することができます。ThinApp が設定する初期値では、同期オブジェクトが他のアプリケーションからアクセスできます。異なるサンドボックスを持つ仮想アプリケーションは、同期オブジェクトを検出できます。

IsolatedSynchronizationObjects パラメータは、セミコロン (;) 区切りのエントリのリストを受け入れます。

例：同期オブジェクトの隔離

IsolatedSynchronizationObjects パラメータを変更して、My Shared Object という名前の同期オブジェクトと、名前に outlook を含んでいる同期オブジェクトを隔離することができます。

```
[BuildOptions]
IsolatedSynchronizationObjects=*outlook*;My Shared Object
```

NotificationDLLs パラメータ

NotificationDLLs パラメータでは、サードパーティの DLL ファイルを呼び出して、アプリケーションの起動やシャットダウンといったイベントの通知を行います。

DLL は、物理ファイル システムまたは仮想パッケージに配置できます。ThinApp が DLL ファイルをロードできない場合は、パッケージでエラーが生成されます。このパラメータは、Package.ini ファイルには表示されません。ただし、ThinApp SDK ユーザーはこのパラメータをファイルに追加できます。

例： DLL ファイルの呼び出し

NotificationDLLs パラメータを変更して、First.dll および Second.dll ファイルを呼び出すことができます。

```
[BuildOptions]
NotificationDLLs=First.dll;Second.dll
```

NotificationDLLSignature パラメータ

NotificationDLLSignature パラメータは、NotificationDLLs パラメータと連携して、指定された DLL ファイルに署名があることを確認します。

DLL に署名がないと、ThinApp でファイルをロードされません。

例： DLL ファイルに Authenticode 署名があることを確認

NotificationDLLSignature パラメータをアスタリスク (*) で変更して、DLL ファイルに Authenticode 署名があることを確認できます。

```
[BuildOptions]
NotificationDLLSignature=*
```

例： DLL ファイルにエンティティ署名があることを確認

エンティティを設定して、DLL にそのエンティティの署名があることを確認できます。

```
[BuildOptions]
NotificationDLLSignature=VMware, Inc.
```

ObjectTypes パラメータ

ObjectTypes パラメータは、物理環境にある他のアプリケーションで認識できる仮想 COM オブジェクト タイプの一覧を指定します。VBScript などのスクリプトを使用して、キャプチャされたアプリケーションを起動するオブジェクトを呼び出すことができます。

オブジェクト タイプは、一度に1つのネイティブ アプリケーションまたは仮想アプリケーションにのみ登録されません。ネイティブ マシンに Office 2003 をインストールし、仮想 Office 2007 パッケージを使用する場合は、仮想アプリケーションまたはネイティブ アプリケーションのどちらでオブジェクト タイプに処理するかを決定する必要があります。

仮想 Office 2007 でオブジェクト タイプを処理する場合は、ObjectTypes の設定を Package.ini ファイルにそのまま残し、パッケージをビルドしてから、thinreg.exe ユーティリティを使用して登録できます。ネイティブ Office 2003 でオブジェクト タイプを処理する場合は、まず ObjectTypes の設定を Package.ini ファイルから削除してから、パッケージのビルドと登録を行う必要があります。

ObjectTypes パラメータにランダムなエントリを追加することはできません。

キャプチャ プロセスで生成したエントリのみを削除できます。

例：COM オブジェクトの作成時に仮想アプリケーションを起動

スクリプトまたはネイティブ アプリケーションで、ObjectTypes パラメータに記載されている Excel.Application COM オブジェクトまたはその他の COM オブジェクトを作成すると、ThinApp は仮想パッケージを開始します。

```
[Microsoft Office Excel 2007.exe]
ObjectTypes=Excel.Application;Excel.Application.12;Excel.Chart;
Excel.Macrosheet;Excel.Sheet; Excel.Workspace
```

SandboxCOMObjects パラメータ

SandboxCOMObjects パラメータは、物理環境のアプリケーションから仮想アプリケーションがランタイム時に登録した COM オブジェクトにアクセスできるかどうかを示します。

ThinApp では、物理環境のネイティブ アプリケーションから仮想アプリケーションが登録する COM オブジェクトにアクセスできないように初期値が設定されています。ThinApp は、仮想アプリケーションが登録する COM オブジェクトをサンドボックスに置きます。

例：サンドボックス外で登録された COM オブジェクトが認識されるようにする

SandboxCOMObjects パラメータを変更して、仮想アプリケーションが登録する COM オブジェクトをサンドボックス外でも認識できるように設定することができます。

たとえば、ネイティブ Microsoft Office 2003 と仮想 Microsoft Office 2007 をインストールし、Microsoft Word を起動して、文書を開き、変更し、保存するようなカスタムのメール マージ プログラムをネイティブ環境で実行する場合は、仮想 Microsoft Word が実行されていると Microsoft Word 2007 文書を生成できます。ネイティブ アプリケーションは、仮想アプリケーションから COM オブジェクトにアクセスできます。

```
SandboxCOMObjects=0
```

VirtualizeExternalOutOfProcessCOM パラメータ

VirtualizeExternalOutOfProcessCOM パラメータは、プロセス外 COM オブジェクトを仮想環境で実行できるかどうかを制御します。

仮想環境の外部にある COM オブジェクトは、常に物理環境で実行されます。

このパラメータは、ThinApp パッケージに含まれておらず、仮想レジストリにも登録されていないプロセス外 COM オブジェクトを処理します。

ThinApp では、外部のプロセス外 COM オブジェクトがホスト コンピュータを変更できないようにその COM オブジェクトを仮想環境で実行するように VirtualizeExternalOutOfProcessCOM パラメータの初期値が設定されています。

外部 COM オブジェクトを仮想環境で実行すると互換性の問題がある場合、ホスト システムに COM オブジェクトを作成して実行できます。

特定の COM オブジェクトのみを仮想環境外で実行するには、ExternalCOMObjects パラメータを使用して、各 COM オブジェクトの CLSID をリストします。[ExternalCOMObjects パラメータ](#)を参照してください。

例：物理環境でのプロセス外 COM オブジェクトの実行

VirtualizeExternalOutOfProcessCOM パラメータを変更して、外部のプロセス外 COM オブジェクトすべてを、仮想環境ではなく物理環境で実行するように設定できます。たとえば、仮想 Microsoft Access 2003 を使用してネイティブの IBM Lotus Notes セッション経由でメールを送信できます。

```
[BuildOptions]
VirtualizeExternalOutOfProcessCOM=0
```

ファイル ストレージの構成

11

ThinApp パラメータを変更して、ファイル ストレージを構成し、仮想ドライブを設定することができます。

この章には、次のトピックが含まれています。

- [CachePath パラメータ](#)
- [UpgradePath パラメータ](#)
- [VirtualDrives パラメータ](#)

CachePath パラメータ

CachePath パラメータは、フォント ファイルおよびスタブ実行可能ファイルのための、展開システムのパスをキャッシュ ディレクトリに設定します。

フォント ファイルおよびスタブ実行可能ファイルは頻繁に使用されるため、ThinApp はキャッシュ内のファイルを迅速に抽出し、物理ディスクに配置する必要があります。

ユーザーがキャッシュを削除した場合に、ThinApp はキャッシュを再構築できます。

`THINSTALL_CACHE_DIR` 環境変数を使用して、ランタイム時に CachePath パラメータをオーバーライドすることができます。 `THINSTALL_CACHE_DIR` 環境変数または CachePath パラメータを設定しない場合、ThinApp は SandboxPath パラメータに基づいて CachePath パラメータの初期値を設定します。値は次のルールに従って設定されます。

- SandboxPath パラメータが `Package.ini` ファイル内に存在し、相対パスを使用している場合、CachePath パラメータはサンドボックス パスを使用し、キャッシュをサンドボックスと同じディレクトリ レベルに格納します。
- SandboxPath パラメータが `Package.ini` ファイル内に存在し、絶対パスを使用している場合、または SandboxPath パラメータが `Package.ini` ファイルに存在しない場合、CachePath パラメータでは `%LocalAppData%\Thinstall\Cache` の場所が使用されます。これにより、ユーザーがサンドボックスを移動させる場所に関係なく、キャッシュ ディレクトリがローカル マシン上に配置されます。ThinApp は、キャッシュ内に `Stubs` ディレクトリを作成します。

例：絶対パスの設定

CachePath パラメータを変更して絶対パスを使用できます。

```
CachePath=C:\VirtCache
```

例：相対パスの設定

アプリケーションの実行可能ファイルがあるディレクトリへの相対的なパスとして ThinApp が検出する相対パスを設定できます。パッケージが C:\VirtApps にあり、CachePath パラメータに Cache の値が指定されている場合、キャッシュ ディレクトリは C:\VirtApps\Cache になります。

```
CachePath=Cache
```

例：キャッシュとサンドボックスを同じディレクトリ レベルで設定

USB デバイスを使用している場合にサンドボックスを USB デバイスに移動する際、ローカル マシンへの干渉を避けるためにキャッシュを USB デバイスに移動することがあります。この場合、キャッシュとサンドボックスは同じディレクトリ レベルに存在します。

```
CachePath=<sandbox_path>
```

UpgradePath パラメータ

UpgradePath パラメータは、Application Sync およびサイドバイサイドの整数更新に使用する情報およびファイルの場所を指定します。

UpgradePath のデフォルトの場所は、ローカル マシンでアプリケーションの実行可能ファイルが置かれているのと同じディレクトリです。

Application Sync ユーティリティは、ログ ファイルおよびキャッシュ ファイルを蓄積し、UpgradePath の場所に更新された実行可能ファイルを構築します。

サイドバイサイドの整数更新では、更新されたバージョンのアプリケーションを UpgradePath の場所で検索しません。

例：サイドバイサイドの更新を保存する別の場所の指定

USB デバイスなど、デフォルトの場所に限られた容量しかない場合、またはアップグレードをアプリケーションの実行可能ファイルと分離する必要がある場合は、UpgradePath パラメータを変更してサイドバイサイドの更新を保存する別の場所を指定できます。

このパラメータのパスには環境変数を使用できますが、フォルダ マクロはサポートされません。

```
[BuildOptions]
UpgradePath=C:\Program Files\<my_app_upgrades>
```

VirtualDrives パラメータ

VirtualDrives パラメータは、アプリケーションがランタイム時に使用できる追加のドライブ文字を指定します。

ThinApp は仮想環境を物理キャプチャ環境に似せて構築し、キャプチャ システムで利用可能な物理ドライブを模倣します。ThinApp は VirtualDrives パラメータと、ドライブ上の仮想ファイルを含む %drive_<drive_letter>% などのプロジェクト フォルダにより仮想ドライブを表します。このプロジェクト フォルダは、物理ドライブ上で書き込み操作が行われない場合、パッケージの読み取り専用ファイル システムおよびサードボックスに置くことができます。

VirtualDrives パラメータは、ランタイム時にドライブをアプリケーションに示します。VirtualDrives パラメータは、ドライブのシリアル番号とタイプなど、ドライブに関するメタデータを表示します。たとえば、ThinApp はキャプチャ システム上の物理 C: ドライブを検出して、そのドライブを *FIXED* タイプのドライブとしてシリアル番号とともにパラメータに入力します。

VirtualDrives パラメータには、以下の情報が含まれます。

- Drive - A ~ Z までの 1 文字。
- Serial - 8 桁の 16 進数。
- Type - FIXED、REMOVABLE、CD-ROM、または RAMDISK。
 - FIXED - 固定メディアを示します。
たとえば、ハード ドライブや内蔵フラッシュ ドライブなどです。
 - REMOVABLE - リムーバブル メディアを示します。
たとえば、ディスク ドライブ、サム ドライブ、またはフラッシュ カード リーダーなどです。
 - CD-ROM - CD-ROM ドライブを示します。
 - RAMDISK - RAM ディスクを示します。

仮想ドライブは、アプリケーションがドライブ文字へのハードコーディングされたパスを使用し、そのドライブ文字が展開システムで使用できない可能性がある場合に便利です。たとえば、レガシー アプリケーションでは、D: ドライブは CD-ROM で、データ ファイルは D:\media にあることを前提としている場合があります。

仮想ドライブの設定は、物理展開システムのドライブの物理プロパティをオーバーライドします。VirtualDrives パラメータでドライブのタイプを CD-ROM として定義していて、物理ドライブがハード ディスクである場合、展開システムのアプリケーションはそのドライブを CD-ROM ドライブとして検出します。

仮想ドライブの分離モード

仮想ドライブは、仮想環境で実行されているアプリケーションにのみ表示されます。仮想ドライブは、物理 Windows 環境には影響を与えません。仮想ドライブの分離モードは、プロジェクト ディレクトリ内のドライブ フォルダにある ##Attributes.ini ファイルでオーバーライドしない限り、プロジェクトのデフォルトの分離モードから継承されます。

アプリケーションの構築前に %drive_D% フォルダにファイルをコピーすると、このドライブで Full 分離モードを使用できます。アプリケーションは常に仮想ドライブから読み取り、展開システム上の対応する物理 CD-ROM ドライブからの読み取りを試行しません。

アプリケーションの構築前に %drive_D% フォルダにファイルをコピーしない場合は、展開システム上の物理ドライブからの読み取りを行うか、またはそのドライブへの書き込みを行うかに応じて、仮想ドライブ フォルダに Merged または WriteCopy 分離モードを使用できます。

仮想ドライブに Merged 分離モードを割り当てると、そのドライブが物理展開システムに存在しない場合は、書き込み操作がすべて失敗します。Merged 分離モードは ThinApp に物理ドライブへの書き込みを指示するため、ThinApp はサンドボックスに変更を転送しません。アプリケーションが指示どおりに物理ドライブに書き込みできない場合は、書き込み操作が失敗します。

VirtualDrives パラメータは、分離モードの設定をオーバーライドしません。仮想アプリケーションは、分離モードの設定が理由で物理ドライブ上のファイルを検出しない場合があります。

仮想ドライブの分離モードの変更

プロジェクトのデフォルトの分離モードをオーバーライドするには、仮想ドライブの分離モードを変更します。

- 1 ThinApp プロジェクトに %Drive_<letter>% ディレクトリを追加します。
- 2 そのドライブ文字用の分離モードのエントリを含む ##Attributes.ini ファイルを作成します。

```
[Isolation]
DirectoryIsolationMode=<isolation_mode>
```

- 3 ##Attributes.ini ファイルを %Drive_<letter>% ディレクトリに置きます。

例：仮想ドライブへのシリアル番号および FIXED タイプの割り当て

VirtualDrives パラメータでは、1つの文字列で複数のドライブ文字の情報と、それらのドライブ文字のオプション パラメータを保持できます。このパラメータでは、異なるドライブ文字に割り当てられた情報間はセミコロンで区切り、個々のドライブ文字のパラメータ間はカンマで区切ります。ThinApp はドライブにシリアル番号と FIXED タイプを割り当てます。

```
[BuildOptions]
VirtualDrives= Drive=A, Serial=12345678, Type=REMOVABLE; Drive=B, Serial=9ABCDEF0, Type=FIXED
```

例：仮想ドライブへのドライブ文字の割り当て

X、D、Z の仮想ドライブ文字を指定できます。

- ドライブ X は、シリアル番号が ff797828 のリムーバブル ディスクです。
- ドライブ D は、割り当てられたシリアル番号を持つ CD-ROM ドライブです。
- ドライブ Z は、割り当てられたシリアル番号を持つ FIXED ディスクです。

```
[BuildOptions]
VirtualDrives=Drive=X, Serial=ff897828, Type=REMOVABLE; Drive=D, Type=CDROM; Drive=Z
```

プロセスとサービスの構成

12

ThinApp パラメータを変更して、ネイティブ プロセスへの書き込みアクセスや、仮想サービスの起動およびシャットダウンを指定する可能性のあるプロセスおよびサービスを構成できます。

この章には、次のトピックが含まれています。

- [AllowExternalKernelModeServices](#) パラメータ
- [AllowExternalProcessModifications](#) パラメータ
- [AutoShutdownServices](#) パラメータ
- [AutoStartServices](#) パラメータ
- [ChildProcessEnvironmentDefault](#) パラメータ
- [ChildProcessEnvironmentExceptions](#) パラメータ

AllowExternalKernelModeServices パラメータ

AllowExternalKernelModeServices パラメータは、アプリケーションがネイティブのカーネル ドライバ サービスを作成および実行できるかどうかを制御します。サービスの実行可能ファイルは、物理ファイル システム上に存在する必要があります。

ThinApp では、Package.ini ファイルにデフォルトのパラメータは表示されませんが、アプリケーションがネイティブの Windows カーネル ドライバ サービスを起動できないようにする初期値が割り当てられます。

例：アプリケーションがネイティブの Windows カーネル ドライバ サービスを作成または開くことを許可

AllowExternalKernelModeServices パラメータを Package.ini ファイルに追加し、デフォルト値の 0 を 1 に変更して、アプリケーションがネイティブの Windows カーネル ドライバ サービスを作成または開くことができます。

```
[BuildOptions]
AllowExternalKernelModeServices=1
```

AllowExternalProcessModifications パラメータ

AllowExternalProcessModifications パラメータは、キャプチャされたアプリケーションがネイティブ プロセスに書き込むことができるかどうかを決定します。一部の仮想化アプリケーションでは、ネイティブ アプリケーションを操作する方法が必要です。

ThinApp では、キャプチャされたアプリケーションが自身をネイティブ アプリケーションに挿入する試みがすべてブロックされます。同じサンドボックス内で実行中の仮想アプリケーションに対しては、キャプチャされたアプリケーションは自身を挿入できます。ThinApp では、Package.ini ファイルにデフォルトのパラメータが表示されません。

ThinApp によって、ネイティブ アプリケーションへのキャプチャされたアプリケーションの挿入がブロックされると、Log Monitor によって、AllowExternalProcessModifications パラメータを参照するトレース ログが生成されます。

例： 仮想プロセスからネイティブ プロセスへの書き込み操作のサポート

Package.ini ファイルに AllowExternalProcessModifications パラメータを追加して、仮想プロセスからネイティブ プロセスへの書き込み操作をサポートすることができます。たとえば、テキストを音声化するには、音声認識アプリケーションが自身をネイティブ アプリケーションに挿入する必要があります。

```
[BuildOptions]
AllowExternalProcessModifications=1
```

AutoShutdownServices パラメータ

AutoShutdownServices パラメータは、最後のサービス以外のプロセスが終了したときに仮想サービスをシャットダウンするかどうかを制御します。

ThinApp が設定する初期値では、最後のサービス以外のプロセスが終了したときに仮想サービスが停止します。このパラメータは、仮想コンテキストの外部のサービスには影響しません。

例： アプリケーション終了後の仮想サービスの実行継続

Apache Web サーバを稼動し、サービスを開始するアプリケーションが終了した後も、仮想サービスを実行したままにする場合は、AutoShutdownServices パラメータを変更できます。

```
[BuildOptions]
AutoShutdownServices=0
```

AutoStartServices パラメータ

AutoStartServices パラメータは、最初の仮想アプリケーションの起動時に仮想サービスを開始するかどうかを制御します。

ThinApp が設定する初期値では、自動起動タイプ オプションを指定してインストールされた仮想サービスが開始されます。仮想サービスは、ユーザーが最初の親プロセスを実行したときに開始します。

例： 仮想サービスの起動の防止

アプリケーションがサービスをインストールしたが使用しない場合、AutoStartServices パラメータを変更して、仮想サービスが起動しないようにして、時間を節約することができます。

```
[BuildOptions]
AutoStartServices=0
```

ChildProcessEnvironmentDefault パラメータ

ChildProcessEnvironmentDefault パラメータは、ThinApp がすべての子プロセスを仮想環境内で実行するかどうかを決定します。

ThinApp は、すべての子プロセスを仮想環境内で作成します。プロセスの速度が遅い場合は、子プロセスを物理環境に移動することができます。子プロセスとして、Microsoft Outlook はメールボックス全体を仮想環境にコピーするときのパフォーマンスに影響を及ぼす可能性があります。

ChildProcessEnvironmentExceptions パラメータを使用して特定の例外を作成できます。

[ChildProcessEnvironmentExceptions パラメータ](#)を参照してください。

例： 物理環境での子プロセスの作成

子プロセスが仮想環境で動作したり、仮想環境の速度を低下させたりしたくない場合は、ChildProcessEnvironmentDefault パラメータを変更して、物理環境に子プロセスを作成することができます。

```
[BuildOptions]
ChildProcessEnvironmentDefault=External
```

ChildProcessEnvironmentExceptions パラメータ

ChildProcessEnvironmentExceptions パラメータは、子プロセスを指定するときの ChildProcessEnvironmentDefault パラメータの例外を注記します。

ChildProcessEnvironmentDefault パラメータを *Virtual* に設定すると、ChildProcessEnvironmentExceptions パラメータに仮想環境の外部で実行されるアプリケーションが一覧表示されます。ChildProcessEnvironmentDefault パラメータを *External* に設定すると、ChildProcessEnvironmentExceptions パラメータに仮想環境内で実行されるアプリケーションが一覧表示されます。

例： 仮想環境での子プロセスの実行に関する例外の指定

仮想環境で子プロセスを実行するための例外を指定できます。仮想アプリケーションが notepad.exe の子プロセスを開始すると、子プロセスは仮想環境の外部で実行されます。

```
[BuildOptions]
ChildProcessEnvironmentExceptions=AcroRd.exe;notepad.exe
ChildProcessEnvironmentDefault=Virtual
```

ThinApp パラメータを変更して、アプリケーションのファイル サイズとブロック サイズを圧縮することができます。

この章には、次のトピックが含まれています。

- [BlockSize パラメータ](#)
- [CompressionType パラメータ](#)
- [MSICompressionType パラメータ](#)

BlockSize パラメータ

BlockSize パラメータは、ThinApp がビルドのファイルを圧縮するときのみ、圧縮ブロックのサイズを制御します。

ブロック サイズが大きいほど、圧縮率が高くなる可能性があります。ブロック サイズが大きくなると、次の理由によりパフォーマンスが低下する場合があります。

- ブロック サイズが大きいと、構築プロセスが遅くなります。
- ブロック サイズが大きいと、アプリケーションの起動時間と読み取り操作が遅くなります。
- より大きいブロック サイズを使用する場合、ランタイム時により多くのメモリが必要になります。

BlockSize パラメータは、Package.ini ファイルと ##Attributes.ini で指定できます。1つのプロジェクト内の異なるディレクトリに異なるブロック サイズを使用できます。

例：ブロックのサイズの増加

BlockSize パラメータのデフォルト サイズの 64 KB を、128 KB、256 KB、512 KB、または 1 MB のサポートされているブロック サイズに増やすことができます。数値の後に *k* を追加してキロバイトを示すことや、*m* を追加してメガバイトを示すことができます。

```
[Compression]
BlockSize=128k
```

CompressionType パラメータ

CompressionType パラメータは、ポータブル実行可能ファイルを除く、パッケージ内のすべてのファイルを圧縮できます。

パッケージが大きい場合にディスク容量が最優先となる場合は、ファイルを圧縮できます。圧縮は迅速に行うことができ、ランタイム時のアプリケーションの起動時間やメモリの使用量への影響はほとんどありません。圧縮では、ZIP アルゴリズムと同様の圧縮率を実現します。

表 13-1. ローカル ハード ドライブで実行されている Microsoft Office 2003 パッケージのサンプル圧縮率と起動時間

圧縮タイプ	なし	高速
サイズ	448,616 KB	257,373 KB
圧縮率	100%	57%
起動時間 (最初の実行)	6 秒	6 秒
起動時間 (2 回目の実行)	0.1 秒	1 秒
構築時間 (最初の構築)	3 分	19 分
構築時間 (2 回目の構築)	2 分	1.2 分

圧縮によりパフォーマンス上の影響が生じるため、古いコンピュータの起動時間や、アプリケーションを複数回起動し、その各起動時にデータを提供するために Windows ディスク キャッシュに依存したりする環境における起動時間に影響する可能性があります。

CompressionType パラメータは MSI ファイルには影響しません。MSI ファイルの圧縮の詳細については、[MSICompressionType パラメータ](#)を参照してください。

ThinApp では、連携して最大のメモリ パフォーマンスと最適な起動時間を達成する OptimizeFor パラメータと CompressionType パラメータのデフォルト値を設定します。ThinApp は、すべてのデータを未圧縮の形式で保存します。

```
[Compression]
CompressionType=None
```

```
[BuildOptions]
OptimizeFor=Memory
```

ディスク容量がある程度重要となる場合に、この設定を使用できます。ThinApp は、実行可能ファイルを未圧縮の形式で保存しますが、他のすべてのデータは圧縮します。

例： 中程度のディスク容量要件の最適化

ディスク容量がある程度重要となる場合に、この設定を使用できます。ThinApp は、実行可能ファイルを未圧縮の形式で保存しますが、他のすべてのデータは圧縮します。

```
[Compression]
CompressionType=Fast
```

```
[BuildOptions]
OptimizeFor=Memory
```

例： 最大のディスク容量要件の最適化

ディスク容量が最優先の場合に、この設定を使用できます。ThinApp はすべてのファイルを圧縮します。

```
[Compression]
CompressionType=Fast
```

```
[BuildOptions]
OptimizeFor=Disk
```

MSICompressionType パラメータ

MSICompressionType パラメータにより、パッケージ配布用の MSI ファイルを圧縮するかどうかを指定します。

圧縮することにより、MSI ファイルのオープン時、および ThinApp SDK の使用時のパフォーマンスが向上します。

キャプチャ プロセスで MSI ファイルを作成すると、ThinApp により Package.ini ファイルに MSICompressionType パラメータが追加され、ファイルの圧縮時に使用する *Fast* の初期値が設定されます。解凍はインストール時に行われます。

CompressionType パラメータを *Fast* に設定する場合は、MSICompressionType パラメータを *Fast* に設定する必要はありません。両方のパラメータを設定しても、圧縮量が増えることはありません。

例： MSI ファイルの圧縮の防止

大規模なビルドで作業しているときに、パフォーマンスが優先事項でない場合は、MSICompressionType パラメータを変更して、MSI ファイルの圧縮を抑制できます。

```
[Compression]
MSICompressionType=none
```

ThinApp パラメータを変更して、ログのアクティビティを防止したり、ログ ファイルの場所をカスタマイズしたりすることができます。

この章には、次のトピックが含まれています。

- [DisableTracing パラメータ](#)
- [LogPath パラメータ](#)
- [FileTimeLogging](#)

DisableTracing パラメータ

DisableTracing パラメータは、セキュリティおよびリソース上の目的で Log Monitor の実行時に、.trace ファイルが生成されないようにします。

標準の .trace ファイルが生成されないようにして、ユーザーからアプリケーションの履歴を隠すことができます。テスト環境では、正常に機能していることがわかっているアプリケーションのトレースをオフにすることができます。不要な .trace ファイルを生成することは、ディスク容量と CPU 時間の浪費となります。

例： Log Monitor でのトレース ファイルの生成の防止

DisableTracing パラメータを設定して、Log Monitor での .trace ファイルの生成を防ぐことができます。

```
[BuildOptions]
DisableTracing=1
```

LogPath パラメータ

LogPath パラメータは、ロギング アクティビティ中に .trace ファイルを保存する場所を設定します。

デフォルトの場所は、アプリケーションの実行可能ファイルを保存するディレクトリと同じです。デフォルトの場所を、容量に余裕のあるディレクトリに変更することも、ログを USB デバイスからクライアント コンピュータにリダイレクトすることもできます。

ThinApp の通常のパスとは異なり、ログ パスに %AppData% や %Temp% などのマクロを指定することはできません。

例： ログ ストレージ ディレクトリの指定

ログ ファイルが C:\ThinappLogs に保存されるように、LogPath パラメータを設定できます。

```
[BuildOptions]
LogPath=C:\ThinappLogs
```

FileTimeLogging

FileTimeLogging パラメータを使用して、人間が判読可能な形式でタイムスタンプが付けられたログ エントリを含むログ ファイルを取得します。

このパラメータには、タイムスタンプが時間、分、秒、およびマイクロ秒単位で表示されます。

人間が判読可能な形式でログを取得するには、FileTimeLogging=1 を package.ini ファイルに追加し、パッケージを ThinApp 5.2.8 以降のバージョンを使用して再構築する必要があります。

バージョンの構成

15

ThinApp パラメータは、アプリケーションの実行可能ファイルおよび ThinApp パッケージのバージョンに関する情報を提供します。

この章には、次のトピックが含まれています。

- [CapturedUsingVersion パラメータ](#)
- [StripVersionInfo パラメータ](#)
- [Version.XXXX パラメータ](#)

CapturedUsingVersion パラメータ

CapturedUsingVersion パラメータには、キャプチャ プロセスの ThinApp のバージョンが表示され、ThinApp が拡張する必要があるファイル システム マクロを決定します。

このパラメータを Package.ini ファイルから変更または削除しないでください。ThinApp は、後方互換性およびテクニカル サポートのためにこのパラメータを使用します。

例：アプリケーションをキャプチャした ThinApp のバージョン番号の表示

CapturedUsingVersion パラメータに ThinApp バージョン 4.0.0-2200 が表示されることがあります。

```
[BuildOptions]
CapturedUsingVersion=4.0.0-2200
```

StripVersionInfo パラメータ

StripVersionInfo パラメータは、ThinApp がアプリケーションを構築するときにソース実行可能ファイルからすべてのバージョン情報を削除するかどうかを決定します。

ソース実行可能ファイルとは、Source パラメータでリストされているファイルです。実行可能ファイルのバージョン情報は、Windows プロパティに表示されます。プロパティ情報には、著作権、商標、バージョン番号が含まれます。StripVersionInfo パラメータによって、Windows のプロパティから [バージョン (Version)] タブが削除されます。

ThinApp では、ソース実行可能ファイルからすべてのバージョン情報をコピーするように StripVersionInfo パラメータの初期値が設定されています。

例：バージョン情報のないアプリケーションの生成

まれなケースですが、StripVersionInfo パラメータを変更して、バージョン情報のないアプリケーションを生成することがあります。たとえば、古いソフトウェアのデータベースに対してバージョンを比較するバージョン検出スキャンを回避する場合などに使用します。

```
[app.exe]
Source=%ProgramFilesDir%\myapp\app.exe
StripVersionInfo=1
```

Version.XXXX パラメータ

Version.XXXX パラメータは、Windows プロパティの [バージョン (Version)] タブのアプリケーション バージョン文字列をオーバーライドするか、新しいバージョン文字列を追加します。

キャプチャ プロセスでは、このパラメータは生成されません。このパラメータは、Package.ini ファイルに追加できます。

例：新しい製品名の設定

Version.XXXX パラメータで、新しい製品名を設定できます。製品名として Office ではなく ThinApp Office を使用したいとします。Version.<string_name>=<string_value> の形式を使用します。

```
[<app>.exe]
Version.ProductName=ThinApp Office
Version.Description=This Product is great!
```

ThinApp パラメータを使用して、ロケール情報の確認を行うことができます。

この章には、次のトピックが含まれています。

- [AnsiCodePage パラメータ](#)
- [LocaleIdentifier パラメータ](#)
- [LocaleName パラメータ](#)

AnsiCodePage パラメータ

AnsiCodePage パラメータには、アプリケーションをキャプチャするオペレーティング システムの言語を表す数値が表示されます。ThinApp は、マルチバイト文字列を管理するためにこの値を使用します。

このパラメータでは言語変換は実行されません。

例：AnsiCodePage パラメータの検証

展開されてキャプチャされたコンピュータのオペレーティング システムの言語が異なる場合は、AnsiCodePage パラメータを確認して、アプリケーションをキャプチャするコンピュータのオペレーティング システムの言語を知ることができます。

```
[BuildOptions]
AnsiCodePage=1252
```

LocaleIdentifier パラメータ

LocaleIdentifier パラメータは、レイアウトと書式に影響するロケールの数値 ID を表示します。この値により、アプリケーションから該当する言語リソースが特定されます。

ThinApp は、パッケージを実行するシステムの設定ではなく、キャプチャ システムの地域および言語の設定に従ってパッケージを実行します。必要な書式を持たないシステムで、日付書式などのロケール書式を必要とするアプリケーションをキャプチャする場合は、このパラメータをコメントアウトすれば、サポートされている書式を持つシステムでアプリケーションを実行できるようになります。

例：LocaleIdentifier パラメータの設定

オペレーティング システムで使用する地域の言語が米国英語の場合、キャプチャ プロセスでは、LocaleIdentifier パラメータが 1033 に設定されます。

```
[BuildOptions]  
LocaleIdentifier=1033
```

LocaleName パラメータ

LocaleName パラメータは、Microsoft Vista にアプリケーションをキャプチャするときのロケールの名前を表示します。

キャプチャしたアプリケーションの LocaleName の表示

LocaleName パラメータでは、日本語のロケール名を表示できます。

```
[BuildOptions]  
LocaleName=ja-JP
```

個々のアプリケーションの構成

17

ThinApp パラメータを変更して、特定のアプリケーションを構成することができます。

エントリ ポイントに固有のパラメータは、Package.ini ファイルの [<application>.exe] セクションで指定されます。たとえば、Adobe Reader アプリケーションに対する Adobe Reader 8.exe でのエントリは、コマンドラインの引数とアプリケーションのショートカットに影響する可能性があります。

この章には、次のトピックが含まれています。

- [CommandLine パラメータ](#)
- [無効なパラメータ](#)
- [ReadOnlyData パラメータ](#)
- [Shortcut パラメータ](#)
- [Shortcuts パラメータ](#)
- [Source パラメータ](#)
- [WorkingDirectory パラメータ](#)

CommandLine パラメータ

CommandLine パラメータは、ショートカット実行可能ファイルを起動するコマンドライン引数を指定します。ソース パラメータはショートカット実行可能ファイルへのパスを指定しますが、CommandLine パラメータは、そのファイルを起動するために必要なオプションまたはパラメータをファイルと一緒に指定します。

アプリケーションの [スタート] メニュー ショートカットにコマンドラインのオプションがある場合は、キャプチャプロセスがこれらのオプションに基づいて CommandLine パラメータの初期値を設定します。テクニカル サポートでのまれなトラブルシューティングのケースにおいて、このパラメータを変更する場合があります。オプションとパラメータは、基本のアプリケーション名に従います。アプリケーションに応じて、オプションまたはパラメータの前に「/」または「-」を使用します。パス名の規則には、フォルダ マクロを使用します。

例： CommandLine パラメータの変更

AnsiCodePage パラメータは、C:\Program Files\Mozilla Firefox\firefox.exe -safe-mode スタートメニューのショートカットに基づくエントリを使用して変更できます。

CommandLine="C:\Program Files\Mozilla Firefox\firefox.exe" -safe-mode コマンドラインの引数には、 /<option> <parameter> 形式を使用できます。

```
[<app>.exe]
Source=%ProgramFilesDir%\<base_app>\<app>.exe
Shortcut=<primary_data_container>.exe
CommandLine="%ProgramFilesDir%\<base_app>\<app>.exe" /<option> <parameter>
```

無効なパラメータ

Disabled パラメータは、アプリケーションのビルド対象がプレースホルダのみであるかどうかを判断します。この場合、 /bin ディレクトリで実行可能ファイルが ThinApp によって生成されないようにします。

ThinApp は、アプリケーション インストーラがデスクトップおよび [スタート (Start)] メニューにショートカットを持っている場合に、エントリ ポイントを有効にします。セットアップ キャプチャ ウィザードに表示されるエントリ ポイントを選択しない場合、ThinApp はビルド プロセス中にそのアプリケーションの実行可能ファイルを生成できないようにする Disabled パラメータの初期値を設定します。

例： エントリポイントを生成するための無効なパラメータの変更

キャプチャ プロセスで cmd.exe, regedit.exe, または iexplore.exe のトラブルシューティングのエントリ ポイントを選択せず、その後に環境をデバッグする必要がある場合は、これらのエントリ ポイントを生成するように Disabled パラメータを変更できます。

```
[app.exe]
Source=%ProgramFilesDir%\<my_app>\<app>.exe
Disabled=0
```

ReadOnlyData パラメータ

ReadOnlyData パラメータは、アプリケーションのビルド時に作成された読み取り専用の仮想レジストリ ファイルの名前を指定し、アプリケーションのプライマリ データ コンテナを指定します。

注： このパラメータは変更しないでください。プライマリ データ コンテナを検索できるように、Package.ini ファイルにはこのパラメータが表示されます。

プライマリ データ コンテナの容量が 200 MB 未満の場合、ThinApp はエントリ ポイントの実行可能ファイルにコンテナを保存します。プライマリ データ コンテナの容量が 200 MB 以上ある場合、ThinApp は、アプリケーションのエントリ ポイントとしては使用できない .dat ファイルとして、このコンテナを保存します。

例： ReadOnlyData パラメータ値の設定

ThinApp は、仮想レジストリ ファイルの名前として Package.ro.tvr を指定する ReadOnlyData に必要な値を設定します。

```
ReadOnlyData=bin\Package.ro.tvr
```

Shortcut パラメータ

Shortcut パラメータは、ショートカット実行可能ファイルが仮想ファイル システムと仮想レジストリを含むプライマリ データ コンテナを指すように指定します。プライマリ データ コンテナには `ReadOnlyData` エントリが含まれており、他のエントリ ポイントには `Shortcut` エントリが含まれているため、プライマリ データ コンテナを `Package.ini` ファイルの他のエントリ ポイントと区別できます。

アプリケーションを起動できるように、ショートカット実行可能ファイルは、プライマリ データ コンテナ ファイルが保存されているディレクトリに存在する必要があります。プライマリ データ コンテナの詳細については、[ReadOnlyData パラメータ](#)を参照してください。

注： `Shortcut` パラメータの値は変更しないでください。ThinApp は、キャプチャ中にプライマリ データ コンテナを検出します。

例：アプリケーションの実行可能ファイルを指す

ThinApp では、ショートカット実行可能ファイル `AcroRd32.exe` がプライマリ データ コンテナ ファイルである `Adobe Reader 8.exe` パラメータを指すように指定できます。

```
[AcroRd32.exe]
Shortcut=Adobe Reader 8.exe
Source=%ProgramFilesDir%\Adobe\Reader 8.0\Reader\AcroRd32.exe
```

ThinApp では、ショートカット実行可能ファイル `Microsoft Office Word 2007.exe` がプライマリ データ コンテナ ファイルである `Microsoft Office Enterprise 2007.dat` を指すように指定できます。

```
[Microsoft Office Word 2007.exe]
Source=%ProgramFilesDir%\Microsoft Office\Office12\WINWORD.EXE
Shortcut=Microsoft Office Enterprise 2007.dat
```

Shortcuts パラメータ

Shortcuts パラメータは、`thinreg.exe` ユーティリティが仮想アプリケーションへのショートカットを作成する場所のリストを示します。

キャプチャ プロセスは、アプリケーション インストーラが実装するショートカットに基づいて、Shortcuts エントリを判断します。MSI ファイルは、Shortcuts パラメータを使用して、作成するショートカットを判断します。

例：Shortcuts パラメータの変更

Shortcuts パラメータを変更して、[スタート (Start)] メニューの Microsoft Office フォルダに、Microsoft Word 2003 アプリケーションのショートカットを作成することができます。複数のショートカットの場所を追加する場合は、セミコロンを使用して各エントリを区切ります。各エントリにはフォルダ マクロを含めることができます。

```
[Microsoft Office Word 2003.exe]
ReadOnlyData=bin\Package.ro.tvr
Source=%ProgramFilesDir%\Microsoft Office\OFFICE11\WINWORD.EXE
Shortcuts=%Programs%\Microsoft Office
```

Source パラメータ

Source パラメータは、ショートカット実行可能ファイルを使用する場合に ThinApp がロードする実行可能ファイルを指定します。このパラメータは、仮想ファイル システムまたは物理ファイル システムの実行可能ファイルへのパスを提供します。

ThinApp では、実行可能ファイルごとにソースを指定します。アプリケーション スイートに 3 つのユーザー エントリ ポイントがある場合 (Winword.exe、Powerpnt.exe、Excel.exe など)、Package.ini ファイルにはその 3 つのアプリケーション エントリがリストされます。各エントリには一意のソース エントリがあります。

ThinApp が仮想ファイル システムでソース実行可能ファイルを見つけられない場合、ThinApp は物理ファイル システムを検索します。たとえば、仮想環境からネイティブの Internet Explorer を使用する場合、ThinApp は、物理ファイル システムからソース実行可能ファイルをロードします。

Source パラメータとプロジェクトの /bin ディレクトリには関連がありません。/bin ディレクトリには、生成された実行可能ファイルが格納されます。Source パスは、読み取り専用仮想ファイル システムに格納されたインストール済み実行可能ファイルを指します。

注： Source パスを変更しないでください。キャプチャ プロセスは、キャプチャ マシンの物理ファイル システム内でアプリケーション インストーラが実行可能ファイルを配置する場所に基づいてパスを判断します。ThinApp は、物理ファイル システムのパスに基づいて仮想ファイル システム パスを作成します。

例： Source パラメータを使用してエントリ ポイントを指す

Source パラメータは、C:\Program Files\<<base_app>\<app>.exe のエントリ ポイントを指すことができます。

```
[<app>.exe]
Source=%ProgramFilesDir%\<base_app>\<app>.exe
```

WorkingDirectory パラメータ

WorkingDirectory パラメータは、アプリケーションがファイルを検索したり、ファイルを配置したりする最初の場所を決定します。

ThinApp は、このパラメータを Package.ini ファイルにデフォルトでは含めません。これは、作業ディレクトリは実行可能ファイルが存在するディレクトリと想定されているためです。ThinApp 環境での通常の場合は、展開マシンのデスクトップです。

個々のアプリケーション用に作業ディレクトリを設定できます。作業ディレクトリは、分離モード設定に応じて、仮想ファイル システム、サンドボックス、または物理システム内に指定できます。パス名の規則には、フォルダ マクロを使用できます。

WorkingDirectory パラメータが作業ディレクトリの初期値を設定しますが、他の場所に移動するとディレクトリは動的に変化します。

例： WorkingDirectory パラメータ値の変更

USB ドライブにアプリケーションがある場合、WorkingDirectory の値をデフォルトの USB の場所からデスクトップの My Documents ディレクトリに変更できます。

```
[<app>.exe]  
WorkingDirectory=%Personal%
```

My Documents ディレクトリの場所は分離モード設定によって異なります。作業ディレクトリを物理システム上の My Documents ディレクトリにマッピングする場合は、Merged 分離モード設定を使用します。作業ディレクトリをローカル マシンのサンドボックスにマッピングする場合は、WriteCopy または Full 分離モード設定を使用します。

アプリケーションユーティリティを使用した依存アプリケーションの構成

18

Application Link ユーティリティは、共有コンポーネントまたは依存アプリケーションを、別個のパッケージで保持します。Package.ini ファイルでは、OptionalAppLinks および RequiredAppLinks エントリを使用して、エンドユーザー コンピュータのランタイムで ThinApp パッケージを動的に組み合わせることができます。このプロセスにより、コンポーネントを別個にパッケージ化、展開、および更新することが可能になり、アプリケーション仮想化のメリットを保持することができます。

ThinApp は、一度に最大 250 個のパッケージをリンクできます。各パッケージのサイズは任意です。リンクは、パッケージのプライマリ データ コンテナを指す必要があります。

リンクされたパッケージからのサンドボックスの変更は、基本パッケージには表示されません。たとえば、Acrobat Reader をスタンドアロンの仮想パッケージとして、および基本の Firefox アプリケーションにリンクされたパッケージとしてインストールすることができます。仮想パッケージを実行して Acrobat Reader をスタンドアロン アプリケーションとして起動し、環境設定を変更すると、ThinApp によって Acrobat Reader のサンドボックスに変更内容が保存されます。Firefox を起動しても、Firefox は独自のサンドボックスを持っているため、これらの変更を検出できません。Firefox で .pdf ファイルを開くと、スタンドアロンの Acrobat Reader アプリケーション内にある環境設定の変更は反映されません。

この章には、次のトピックが含まれています。

- [Application Link のパス名の形式](#)
- [RequiredAppLinks パラメータ](#)
- [OptionalAppLinks パラメータ](#)

Application Link のパス名の形式

Application Link ユーティリティは、次のパス名の形式をサポートしています。

- パス名は、ベースの実行可能ファイルに対する相対パスにすることができます。たとえば、RequiredAppLinks=..SomeDirectory とすると、ベースの実行可能ファイルを c:\MyDir\SubDir\Dependency.exe に展開する場合は、C:\MyDir\SomeDirectory となります。
- パス名には絶対パス名を使用できます。例として、RequiredAppLinks=C:\SomeDirectory のようになります。
- パス名には、ネットワーク共有または UNC パスを使用できます。例として、RequiredAppLinks=\\share\somedir\Dependency.exe のようになります。

- パス名には、ユーザーまたはコンピュータごとに特定の場所に動的に拡張される、システムまたはユーザー環境変数を含めることができます。例として、`RequiredAppLinks=%MyEnvironmentVariable%\Package.dat` のようになります。

環境変数を使用することのリスクとして、アプリケーションを起動する前にユーザーが値を変更し、管理者が設定したものと別に Application Link の依存関係を作成することが考えられます。

- パス名には、ThinApp フォルダ マクロを含めることができます。例として、`RequiredAppLinks=%SystemSystem%\Package.dat` のようになります。
- パス名には、スペースを含めることができます。
- パス名では、個々のファイル名を区切るセミコロンを使用して、複数のリンクまたは依存関係を指定することができます。例として、`RequiredAppLinks=Dependency1.exe; Dependency2.exe;` のようになります。
- パス名では、ファイル名とディレクトリ パスに、アスタリスクとクエリのワイルドカード文字 (* および ?) を含めることができます。例として、`RequiredAppLinks=WildPath*\WildFilename*.dat` のようになります。

ワイルドカード文字を含むパスがファイル システム内の複数のディレクトリに一致している場合は、それぞれの一致するディレクトリ名が返されます。これにより、追加のパスまたはファイル名の照合が有効になります。

環境変数と組み合わせて使用されるワイルドカードを使用すると、依存関係のあるアプリケーションに対して、カスタマイズされた強力な再帰検索を実行できます。例として、`OptionalAppLinks=%HOMEPATH%\OfficePlugins**` のようになります。

RequiredAppLinks パラメータ

RequiredAppLinks パラメータは、ランタイム時に基本パッケージにインポートする必要のあるパッケージのリストを指定します。このパラメータは、基本パッケージの `Package.ini` ファイルで構成できます。

依存するパッケージのインポート操作が失敗すると、エラー メッセージが表示され、ベースの実行可能ファイルが終了します。ロード エラーが発生した場合でも、OptionalAppLinks パラメータを使用して続行することができます。ワイルドカード パターンを使用してパッケージを指定し、ファイルがワイルドカード パターンに一致しない場合、ThinApp はエラー メッセージを生成しません。

パッケージのインポートには、次の操作が含まれます。

- インポートしたパッケージからの VBScripts の実行
- インポートしたパッケージからの自動起動サービスの開始
- インポートしたパッケージからのフォントの登録
- Windows XP から Windows Vista への SxS DLL ファイルの再配置

パッケージのプライマリ データ コンテナへのリンクを作成する必要があります。他のショートカット パッケージにリンクすることはできません。

リンクはクライアント マシン上での実行時に有効になるため、パス名は展開マシンに設定されています。リンクされたパッケージを区切るにはセミコロンを使用します。パス名の形式の詳細については、[Application Link のパス名の形式](#)を参照してください。

例：アプリケーションを .NET にリンク

dotnet.exe パッケージの .NET Framework をパッケージ化して、.NET アプリケーションを使用する場合は、そのアプリケーションを起動する前に dotnet.exe ファイルへのリンクを設定する必要があることを指定できます。

```
RequiredAppLinks=C:\abs\path\dotnet.exe
```

相対パスを指定できます。

```
RequiredAppLinks=<relative_path>\dotnet.exe
```

UNC パスを指定できます。

```
RequiredAppLinks=\\server\share\dotnet.exe
```

パスの値では ThinApp フォルダ マクロを使用できます。

```
RequiredAppLinks=%SystemSystem%\Package.dat
```

パスの値には環境変数を使用できます。環境変数を使用することのリスクとして、アプリケーションを起動する前にユーザーが値を変更し、管理者が設定したものと別 Application Link の依存関係を作成することが考えられます。

```
RequiredAppLinks=%MyEnvironmentVariable%\Package.dat
```

ベースの実行可能ファイルと同じディレクトリにある単一のパッケージをインポートできます。

```
RequiredAppLinks=Plugin.exe
```

ベースの実行可能ファイルのサブディレクトリにある単一のパッケージをインポートできます。

```
RequiredAppLinks=plugins\Plugin.exe
```

プラグイン ファイルのディレクトリにあるすべての実行可能ファイルをインポートできます。ファイルが有効な ThinApp パッケージになっていないこと、あるいはセキュリティの問題が発生したことが原因で、実行可能ファイルがインポートできない場合、ベースの実行可能ファイルのロードに失敗します。

```
RequiredAppLinks=plugins\*.exe
```

n:\plugins 絶対パスにあるすべての実行可能ファイルをインポートできます。

```
RequiredAppLinks=n:\plugins\*.exe
```

PLUGINS 環境変数を展開して、この場所にあるすべての実行可能ファイルをインポートできます。

```
RequiredAppLinks=%PLUGINS%\*.exe
```

指定した 2 つのプラグイン ファイルとプラグインの場所にある実行可能ファイルのリストを読み込むことができます。

```
RequiredAppLinks=plugin1.exe;plugin2.exe;plugins\*.exe
```

OptionalAppLinks パラメータ

OptionalAppLinks パラメータは、RequireAppLinks パラメータと似ていますが、インポート操作が失敗した場合でもエラーを無視してメイン アプリケーションを起動します。

パッケージのプライマリ データ コンテナへのリンクを作成する必要があります。他のショートカット パッケージにリンクすることはできません。

リンクはクライアント マシン上での実行時に有効になるため、パス名は展開マシンに設定されています。パスには、絶対パス (C:\abs\path\dotnet.exe など)、相対パス (relpath\dotnet.exe など)、および UNC パス (\server\share\dotnet.exe など) を指定できます。

RequireAppLinks と OptionalAppLinks のパラメータはいずれも同じ構文を使用します。

RequireAppLinks パラメータの詳細と例については、[RequiredAppLinks パラメータ](#)を参照してください。

OptionalAppLinks パラメータを介して Application Link ユーティリティを使用する場合、基本パッケージとこれにリンクされる依存関係はいずれも、同じバージョンの ThinApp で構築する必要があります。古いパッケージをアップグレードして、最新バージョンの ThinApp ランタイムを組み込むには、ThinApp に含まれている relink.exe コマンドライン ユーティリティを使用します。

Application Sync ユーティリティを使用したアプリケーション更新の構成

19

Application Sync ユーティリティは、展開された仮想アプリケーションを最新の状態に維持します。アプリケーションが起動すると、Application Sync ユーティリティは Web サーバにクエリして、パッケージの更新バージョンが使用可能かどうかを判断します。使用可能な更新がある場合は、ThinApp により既存のパッケージと新しいパッケージの差分がダウンロードされ、パッケージの更新バージョンが構築されます。

Application Sync ユーティリティは、バックグラウンドで更新をダウンロードします。引き続き古いバージョンのアプリケーションを使用できます。ダウンロードが完了する前にユーザーがアプリケーションを終了すると、仮想アプリケーションが再起動したときにダウンロードが再開されます。ダウンロード完了後、アプリケーションが次回起動したときに、ThinApp によって新しいバージョンが有効化されます。

すべての Application Sync パラメータを有効にするには、AppSyncURL パラメータのコメントを解除する必要があります。以下のエントリは、Application Sync パラメータのデフォルト設定です。

```
AppSyncURL=https://example.com/some/path/PackageName.exe
AppSyncUpdateFrequency=1d
AppSyncExpirePeriod=30d
AppSyncWarningPeriod=5d
AppSyncWarningFrequency=1d
AppSyncWarningMessage=This application will become unavailable for use in AppSyncWarningPeriod
days if it cannot contact its update server. Check your network connection to ensure
uninterrupted service
AppSyncExpireMessage=This application has been unable to contact its update server for
AppSyncExpirePeriod days, so it is unavailable for use. Check your network connection and try
again
AppSyncUpdatedMessage=
AppSyncClearSandboxOnUpdate=0
```

この章には、次のトピックが含まれています。

- [AppSyncClearSandboxOnUpdate パラメータ](#)
- [AppSyncExpireMessage パラメータ](#)
- [AppSyncExpirePeriod パラメータ](#)
- [AppSyncURL パラメータ](#)
- [AppSyncUpdateFrequency パラメータ](#)
- [AppSyncUpdatedMessage パラメータ](#)

- [AppSyncWarningFrequency](#) パラメータ
- [AppSyncWarningMessage](#) パラメータ
- [AppSyncWarningPeriod](#) パラメータ

AppSyncClearSandboxOnUpdate パラメータ

AppSyncClearSandboxOnUpdate パラメータは、更新後にサンドボックスをクリアするかどうかを決定します。ThinApp は、AppSyncClearSandboxOnUpdate パラメータにサンドボックスの内容を保持する初期値を設定します。

例： AppSyncClearSandboxOnUpdate パラメータの変更

AppSyncClearSandboxOnUpdate パラメータを変更して、アプリケーションの更新後にサンドボックスをクリアすることができます。

```
AppSyncClearSandboxOnUpdate=1
```

AppSyncExpireMessage パラメータ

AppSyncExpireMessage パラメータは、有効期間の終了後に仮想アプリケーションを起動して Web サーバへの接続が失敗したときに表示されるメッセージを設定します。このメッセージが表示されるとアプリケーションは終了します。

例： AppSyncExpireMessage パラメータの表示

ThinApp は、AppSyncExpireMessage パラメータのデフォルト メッセージを提供します。

```
AppSyncExpireMessage=このアプリケーションは <AppSyncExpirePeriod_value> 日間更新サーバに接続できないため、使用できません。ネットワーク接続を確認して、もう一度やり直してください。
```

AppSyncExpirePeriod パラメータの値が時間単位または分単位の場合は、メッセージを変更して日数ではなく時間または分を示すようにします。

AppSyncExpirePeriod パラメータ

AppSyncExpirePeriod パラメータは、パッケージの有効期限を分 (m)、時間 (h)、または日 (d) 単位で設定します。ThinApp が Web サーバに到達して更新を確認できない場合、パッケージは有効期限が切れるまで、またはユーザーによって終了されるまで継続して機能します。有効期限が終了しても、ThinApp は以降毎回の起動の試行時に、Web サーバに到達することを試みます。

例： AppSyncExpirePeriod パラメータのデフォルト値の設定

デフォルトの never 値を使用して、パッケージを有効期限なしにすることができます。

```
AppSyncExpirePeriod=never
```

AppSyncURL パラメータ

AppSyncURL パラメータは、アプリケーションの更新バージョンを格納する Web サーバ URL またはファイル共有の場所を設定します。ThinApp はこの場所をチェックし、更新されたパッケージをダウンロードします。

Application Sync は、HTTP (セキュアでない)、HTTPS (セキュア)、および File プロトコルを介して機能します。HTTPS プロトコルには、Web サーバの ID を確認するオプションが含まれます。基本認証用に、AppSyncURL パラメータにユーザー名とパスワードを含めることができます。ThinApp は、標準の Internet Explorer プロキシ設定に準拠しています。

注： すべての Application Sync パラメータを有効にするには、AppSyncURL パラメータのコメントを解除する必要があります。

例： AppSyncURL パラメータへの値の割り当て

HTTP または HTTPS 値は、次の形式に従って AppSyncURL パラメータに割り当てることができます。

```
AppSyncURL=https://<site.com>/<path>/<primary_data_container_name>
```

ローカル ドライブとネットワーク ドライブのパスを指定できます。プライマリ データ コンテナは、.exe または .dat ファイルのいずれかにすることができます。

```
file:///C:/<path>/<primary_data_container_name>
```

ネットワーク リソースの UNC パスとアクセス場所を使用できます。

```
file://<server>/<share>/<path>/<primary_data_container_name>
```

AppSyncUpdateFrequency パラメータ

AppSyncUpdateFrequency パラメータは ThinApp がアプリケーションの更新について Web サーバをチェックする頻度を指定します。更新頻度は、分 (m)、時間 (h)、または日 (d) 単位で設定できます。

ThinApp で設定される初期値の 1d は、パッケージを Web サーバに 1 日に 1 回接続して更新をチェックすることになります。別の実行中のアプリケーションが同じサンドボックスを共有している場合、ThinApp は更新をチェックしません。

例： AppSyncUpdateFrequency パラメータの変更

AppSyncUpdateFrequency パラメータを 0 の値で変更して、アプリケーションを起動するたびに更新をチェックするように設定できます。

```
AppSyncUpdateFrequency=0
```

AppSyncUpdatedMessage パラメータ

AppSyncUpdatedMessage パラメータは、更新されたパッケージの初回起動時に表示されるメッセージを設定します。

更新された AppSyncUpdatedMessage パラメータの表示

AppSyncUpdatedMessage パラメータを使用して、アプリケーションが更新されていることを確認できます。

```
AppSyncUpdatedMessage=Your application has been updated.
```

AppSyncWarningFrequency パラメータ

AppSyncWarningFrequency パラメータは、パッケージの有効期限が切れる前に警告が表示される頻度を指定します。分 (m)、時間 (h)、または日 (d) の値を指定できます。

ThinApp では初期値として、警告メッセージが 1 日に 1 回表示される 1d が設定されます。

例： AppSyncWarningFrequency パラメータの変更

AppSyncWarningFrequency パラメータを変更して、アプリケーションが起動するたびに警告が表示されるように設定できます。

```
AppSyncWarningFrequency=0
```

AppSyncWarningMessage パラメータ

AppSyncWarningMessage パラメータは、警告期間の開始時に表示されるメッセージを設定します。警告期間中にはじめてアプリケーションを起動したときには、警告メッセージが表示され、ThinApp がサーバから更新にアクセスしようとします。パッケージを更新できない場合、ThinApp はアプリケーションが起動するたびに試し直します。警告メッセージは、各 AppSyncWarningFrequency 期間の期限が切れる则表示されます。

例： AppSyncWarningMessage パラメータのデフォルト メッセージの表示

ThinApp には、Application Sync ユーティリティの警告に関するデフォルト メッセージが含まれています。

```
AppSyncWarningMessage=このアプリケーションは、更新サーバに接続できない場合、あと %%remaining_days%% 日で使用できなくなります。ネットワーク接続を確認して、サービスが中断されないようにします。
```

%%remaining_days%% 変数は、パッケージの有効期限までの残り日数です。AppSyncWarningPeriod パラメータの値が時間単位または分単位の場合は、メッセージを変更して日数ではなく時間または分を示すようにします。

AppSyncWarningPeriod パラメータ

AppSyncWarningPeriod パラメータは、パッケージの有効期限が切れる前の、警告期間の開始時点を設定します。分 (m)、時間 (h)、または日 (d) の値を指定できます。警告期間が開始すると、ThinApp はアプリケーションが起動するたびに Web サーバをチェックし、AppSyncUpdateFrequency パラメータの値を 0 に設定します。

例： AppSyncWarningPeriod パラメータのデフォルト期間の指定

AppSyncWarningPeriod パラメータのデフォルト期間は 5 日間です。

```
AppSyncWarningPeriod=5d
```

MSI ファイルの構成

20

ThinApp パラメータを変更して、デスクトップ管理システムを通じた展開のための MSI ファイルを構成できます。

MSI ファイルの圧縮に関する情報は、ファイル サイズを制御するその他のパラメータとともに表示されます。

[MSICompressionType パラメータ](#)を参照してください。

この章には、次のトピックが含まれています。

- [MSIArpProductIcon パラメータ](#)
- [MSIDefaultInstallAllUsers パラメータ](#)
- [MSIFilename パラメータ](#)
- [MSIInstallDirectory パラメータ](#)
- [MSIManufacturer パラメータ](#)
- [MSIProductCode パラメータ](#)
- [MSIProductVersion パラメータ](#)
- [MSIRequireElevatedPrivileges パラメータ](#)
- [MSIUpgradeCode パラメータ](#)
- [MSIStreaming パラメータ](#)
- [MSIIs64Bit パラメータ](#)

MSIArpProductIcon パラメータ

MSIArpProductIcon パラメータは、Windows の [プログラムの追加と削除] ダイアログでアプリケーションを表すのに使用するアイコンを指定します。アイコンは、ICO、DLL、または実行可能ファイルに配置できます。

注： このパラメータは変更しないでください。MSI パッケージにアプリケーション アイコンがない場合、アプリケーションは汎用アイコンで表されます。

例：MSIArpProductIcon パラメータによるアイコンの指定

MSIArpProductIcon パラメータでは、Microsoft Office 2007 のアイコンを指定できます。これは、インデックス番号を使用して DLL ファイルにある最初のアイコンを指す例を示します。

```
MSIArpProductIcon=%Program Files Common%\Microsoft Shared\OFFICE12\
Office Setup Controller\OSETUP.DLL,1
```

この MSIArpProductIcon=<path_to_icon_file>[,<icon_index_number>] フォーマットにある <icon_index_number> エントリは、DLL ファイルまたは実行可能ファイルに複数のアイコンが存在する場合にのみ適用されます。

MSIDefaultInstallAllUsers パラメータ

MSIDefaultInstallAllUsers パラメータにより、MSI データベースのインストール モードが設定されます。コンピュータ上の全ユーザーおよび個々のユーザーに対して、.msi ファイルをインストールできます。

MSIFilename パラメータで Windows Installer データベースの生成が要求された場合にのみ、パラメータが適用されます。

例：MSIDefaultInstallAllUsers パラメータの値の設定

ThinApp により、コンピュータにログインするすべてのユーザーに対して、ショートカットとファイル タイプの関連付けを使用して MSI データベースをインストールする MSIDefaultInstallAllUsers パラメータの初期値を設定します。データベースをインストールするユーザーには、管理者権限が必要です。この方法によって、すべてのユーザーを対象にアプリケーションをデスクトップにプッシュすることができます。

```
[BuildOptions]
MSIFilename=<my_msi>.msi
MSIDefaultInstallAllUsers=1
```

個々のユーザーは、該当するユーザーに限り、ショートカットおよびファイル タイプの関連付けによって MSI データベースをインストールできます。個別のユーザーのインストールについては、管理者権限は必要ありません。各ユーザーが個別にアプリケーションを展開できるようにする場合は、この方法を使用します。

```
[BuildOptions]
MSIFilename=<my_msi>.msi
MSIDefaultInstallAllUsers=0
```

管理者は、マシン上のすべてのユーザーについて MSI データベースをインストールできます。また、管理者権限のない個々のユーザーは、該当するユーザーについてのみデータベースをインストールできます。

```
[BuildOptions]
MSIFilename=<my_msi>.msi
MSIDefaultInstallAllUsers=2
```

MSIFilename パラメータ

MSIFilename パラメータは、MSI データベースの生成をトリガーし、そのファイル名を指定します。これ以外の MSI パラメータは、MSIFilename パラメータのコメントを解除する場合にのみ効果があります。

このパラメータにより、出力ディレクトリに指定されたファイル名を付けた Windows Installer が生成されます。MSI ファイルは、デスクトップ管理システムを介してリモートの場所にパッケージを配布する場合に作成できます。thinreg.exe ユーティリティを手動で使用する必要のある実行可能ファイルとは異なり、MSI ファイルでは、各ユーザーに応じたショートカットとファイル タイプの関連付けが自動的に作成されます。

キャプチャ プロセスで MSI の生成を指定しない限り、ThinApp では MSIFilename パラメータがコメントアウトされます。

例：MSIFilename パラメータの置き換え

ビルド プロセスで MSI ファイルを生成し、ファイル名をユーザー独自のファイル名に置き換えることができます。

```
[BuildOptions]
MSIFilename=<my_msi>.msi
```

インベントリ名は、MSIFilename パラメータのデフォルト名です。

```
[BuildOptions]
;MSIFilename=<inventory_name>.msi
```

MSIInstallDirectory パラメータ

MSIInstallDirectory パラメータは、MSI インストール ディレクトリの相対パスを指定します。パスは、各マシンのインストールでは %ProgramFilesDir% への相対パスになり、各ユーザーのインストールでは %AppData% への相対パスになります。

すべてのユーザーに対して MSI データベースをインストールすると、ThinApp では、各マシンへのインストール中にアプリケーションが C:\%ProgramFilesDir%\<InventoryName> (VMware ThinApp) ディレクトリに配置されます。

個々のユーザーに対して MSI データベースをインストールすると、アプリケーションは C:\%AppData%\<InventoryName> (VMware ThinApp) ディレクトリに配置されます。MSIFilename パラメータで Windows Installer データベースの生成が要求された場合にのみ、パラメータが適用されます。

例：MSIInstallDirectory パラメータに対するディレクトリの場所の指定

MSIInstallDirectory パラメータでインベントリ名に基づく場所の使用を避けたい場合は、.msi ファイルを C:\Program Files\<my_application> ディレクトリにインストールできます。

```
[BuildOptions]
MSIFilename=<my_msi>.msi
MSIInstallDirectory=<my_application>
```

MSIManufacturer パラメータ

MSIManufacturer パラメータは、MSI データベースのメーカーまたはパッケージング会社を指定し、Windows の [プログラムの追加と削除] ダイアログ ボックスに値を表示します。

ThinApp は、Windows のコピーが登録されている企業の名前に、MSIManufacturer パラメータの初期値を設定します。

MSIFilename パラメータで Windows Installer データベースの生成が要求された場合にのみ、パラメータが適用されます。

例：MSIManufacturer パラメータの変更

特定の部署名を表示するように、MSIManufacturer パラメータを変更できます。たとえば、Windows の [プログラムの追加と削除] ダイアログ ボックスで部署名を確認し、その部署のヘルプ デスクに連絡することができます。

```
[BuildOptions]
MSIFilename=<my_msi>.msi
MSIManufacturer=<department_or_company_name>
```

MSIProductCode パラメータ

MSIProductCode パラメータは、MSI データベースの製品コードを指定します。Windows Installer は、このコードで MSI パッケージを識別します。

キャプチャ プロセスでは、アプリケーションから取得したものではない、ランダムで一意的な製品コードが生成されます。値は、有効な GUID (Globally Unique Identifier) である必要があります。

MSIFilename parameter で Windows Installer データベースの生成が要求された場合にのみ、パラメータが適用されます。

注： MSIProductCode パラメータは変更しないでください。

例：MSIProductCode パラメータを使用した MSI ファイルの作成

キャプチャ プロセスでは、製品コードとして 590810CE-65E6-3E0B-08EF-9CCF8AE20D0E を使用して MSI ファイルを作成できます。

```
[BuildOptions]
MSIFilename=<my_msi>.msi
MSIProductCode={590810CE-65E6-3E0B-08EF-9CCF8AE20D0E}
```

MSIProductVersion パラメータ

MSIProductVersion パラメータは、MSI データベースの製品バージョン番号を指定します。これによりバージョンの管理がしやすくなります。このバージョン番号は、アプリケーションのバージョンや ThinApp のバージョンとは関係がありません。

ThinApp により、初期バージョン 1.0 が割り当てられます。このバージョン番号は、データベースのプロパティに表示されます。

すでにパッケージがインストールされているマシンにパッケージを展開すると、Windows Installer はバージョン番号をチェックし、更新されたバージョンの上に旧バージョンがインストールされないようにします。この場合は必ず、新しいバージョンをアンインストールしてください。

MSIProductVersion パラメータは、MSIFilename パラメータで Windows Installer データベースの生成が要求された場合にのみ適用されます。

例：MSIProductVersion パラメータの変更

MSI パッケージを変更するときに、MSIProductVersion パラメータの値を変更できます。2.0 の値を指定すると、1.0 バージョンのパッケージがアンインストールされ、2.0 バージョンのパッケージがインストールされます。

```
[BuildOptions]
MSIFilename=<my_msi>.msi
MSIProductVersion=2.0
```

MSIProductVersion 値の形式は X.Y.Z です。X と Y は 0~255 の範囲、Z は 0~65536 の範囲にあるそれぞれの値になります。

MSIRequireElevatedPrivileges パラメータ

MSIRequireElevatedPrivileges パラメータは、Windows Vista に適用され、MSI データベースに対する昇格権限の要件を指定します。

Windows Vista にログインするほとんどのユーザーは、権限に制限があります。ショートカットとファイルタイプの関連付けを必要とするすべてのユーザーに MSI パッケージをインストールする場合、ユーザーには昇格権限が必要になります。

ThinApp では、MSI データベースで昇格権限が必要であることを示す MSIRequireElevatedPrivileges パラメータの初期値が設定されます。システムが UAC プロンプト用としてセットアップされている場合は、アプリケーションをインストールすると、UAC プロンプトが表示されます。

MSIFilename parameter で Windows Installer データベースの生成が要求された場合にのみ、パラメータが適用されます。

例：MSIRequireElevatedPrivileges パラメータの変更

MSIRequireElevatedPrivileges パラメータを変更することにより、すべてのコンピュータを対象として UAC プロンプトとインストールをブロックすることができます。

```
[BuildOptions]
MSIFilename=<my_msi>.msi
MSIRequireElevatedPrivileges=0
```

MSIUpgradeCode パラメータ

MSIUpgradeCode パラメータは、更新を容易にする MSI データベースのコードを指定します。2 つのパッケージ（たとえばバージョン 1.0 パッケージとバージョン 2.0 パッケージなど）に同じアップグレード コードが含まれている場合、MSI インストーラはこのリンクを検出し、古い方のパッケージをアンインストールして、更新されたパッケージをインストールします。

キャプチャ プロセスでは、インベントリ名に基づいてランダムなアップグレード コードが生成されます。MSI データベースのバージョンで同じアップグレード コードが使用されるように、MSIwrapper の各バージョンでインベントリ名を統一するようにしてください。インベントリ名の詳細については、[InventoryName パラメータ](#)を参照してください。

MSIFilename パラメータで Windows Installer データベースの生成が要求された場合にのみ、パラメータが適用されます。

注： 新しい値が有効な GUID である場合を除き、UpgradeCode 値は変更しないでください。

例：MSIUpgradeCode パラメータを使用した MSIFile の作成

キャプチャ プロセスでは、アップグレード コードとして D89F1994-A24B-3E11-0C94-7FD1E13AB93F を使用して MSI ファイルを作成できます。

```
[BuildOptions]
MSIFilename=mymsi.msi
MSIUpgradeCode={D89F1994-A24B-3E11-0C94-7FD1E13AB93F}
```

MSIStreaming パラメータ

MSIStreaming パラメータにより、アプリケーションのパフォーマンスに影響を及ぼす可能性のある .cab ファイルの使用が決定されます。

デフォルト設定は MSIStreaming=0 です。この値を指定すると、パッケージされている .exe ファイルと .dat ファイルが .cab ファイルに圧縮されます。Package.ini のパラメータを無効にすると、値を 0 に設定した場合と同じ効果になります。MSI を実行すると、パッケージされたファイルが、Windows オペレーティングシステムの Program Files ディレクトリに抽出され、その他の Package.ini の設定に応じてユーザーまたはシステムのいずれかに登録されます。

パラメータ値 MSIStreaming=1 を設定すると、まず .exe と .dat のパッケージ ファイルが .cab ファイルから除外され、以降、.msi ファイルから除外されます。この値を指定すると、展開時にネットワーク共有からアプリケーションを登録するか、または Windows オペレーティングシステムの Program Files ディレクトリにインストールするかのいずれかを選択できるようになります。

この値を設定する場合は、ThinApp パッケージの MSI と .exe および .dat の各ファイルがいずれもネットワーク共有で利用可能になっていることを確認してください。

例：MSIStreaming パラメータの変更

MSIStreaming パラメータを変更することにより、アプリケーションのインストール プロセスが遅くなったときに、.cab ファイルを無視することができます。MSI ファイルと個々の実行可能ファイルを /bin ディレクトリに分散させて、アプリケーションをインストールできます。

```
[BuildOptions]
MSIStreaming=1
```

MSIIs64Bit パラメータ

MSIIs64Bit パラメータを使用して、ProgramFiles ディレクトリに 64 ビット アプリケーションをインストールできます。

64 ビット OS では ThinApp により、32 ビット MSI パッケージの MSI パッケージ アプリケーションが ProgramFiles(x86) にインストールされます。64 ビット MSI パッケージの場合、package.ini ファイルに MSIIs64Bit パラメータを設定してアプリケーションを ProgramFiles にインストールする必要があります。5.0 ビルドを使用してアプリケーションをキャプチャすると、このパラメータは Package.ini にデフォルトで表示されます。

```
[BuildOptions]
MSIIs64Bit=1
```

サンドボックス ストレージとインベントリ名の構成

21

ThinApp パラメータを変更して、キャプチャしたアプリケーションによって行われるすべての変更が保存されるサンドボックスを構成できます。ThinApp のインベントリ名が、サンドボックス名を変更する必要性に影響する可能性があります。

この章には、次のトピックが含まれています。

- [InventoryName パラメータ](#)
- [RemoveSandboxOnExit パラメータ](#)
- [SandboxName パラメータ](#)
- [SandboxNetworkDrives パラメータ](#)
- [SandboxPath パラメータ](#)
- [SandboxRemovableDisk パラメータ](#)
- [SandboxWindowClassName パラメータ](#)

InventoryName パラメータ

InventoryName パラメータは、インベントリ追跡ユーティリティがパッケージの識別に使用する文字列です。このパラメータにより、アプリケーションのキャプチャ プロセスでプロジェクト フォルダとサンドボックスのデフォルト名が決まります。

アプリケーションのキャプチャ プロセスでは、次のいずれかの場所で作成された新しい文字列に基づいて、InventoryName パラメータのデフォルト値が設定されます。

- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall
- HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall

thinreg.exe ユーティリティおよび ThinApp の MSI ファイルは、インベントリ名を参照して、コントロール パネルの [プログラムの追加と削除] ウィンドウに表示する製品名を決定します。たとえば、インベントリ名が SuperApp で、MSI ファイルをインストールするか、thinreg.exe ユーティリティを使用してパッケージを登録すると、[プログラムの追加と削除] ウィンドウに、インストールされているアプリケーションが SuperApp(VMware ThinApp) 文字列とともに表示されます。ThinApp は、インベントリのスキャン中に仮想化されたアプリケーションを識別するために VMware ThinApp をインベントリ名に追加します。

同じアプリケーションの異なる複数のバージョンで同じインベントリ名を使用して、[プログラムの追加と削除] ウィンドウには必ず最新のバージョンのみが表示されるようにすることができます。これらのアプリケーションは [プログラムの追加と削除] ウィンドウで相互に上書きされ、ユーザーが登録済みのすべてのパッケージをアンインストールすることが防止されます。複数のバージョンをアンインストールするには、バージョンごとに異なるインベントリ名を使用します。たとえば、インベントリ名として、単に Microsoft Office とするのではなく、Microsoft Office 2003 および Microsoft Office 2007 を使用します。同じ環境で異なるバージョンの仮想アプリケーションを維持している場合、SandboxName パラメータを変更して、新しいバージョンが別のサンドボックス内のユーザー設定を隔離するようにすることができます。

他のアプリケーションを含んでいるパッケージがある場合は、パッケージの実際のコンテンツを反映するように、インベントリ名を手動で更新することができます。たとえば、SuperApp アプリケーションをキャプチャし、パッケージに Java Runtime が含まれている場合、InventoryName 値は SuperApp ではなく Java Runtime Environment 1.5 として表示される可能性があります。[プログラムの追加と削除] ウィンドウには、パッケージ内にインストールされている最初のアプリケーションが表示されます。

例： InventoryName パラメータの変更

InventoryName パラメータは、Microsoft Office 2003 に変更することができます。

```
[BuildOptions]
InventoryName=Microsoft Office 2003
```

RemoveSandboxOnExit パラメータ

RemoveSandboxOnExit パラメータにより、最後の子プロセスが終了した時点でサンドボックスが削除され、アプリケーションがリセットされます。

ThinApp は、アプリケーションによるレジストリとファイル システムの各場所への変更すべてを、隔離モード WriteCopy または Full を使用してサンドボックスに保存します。ThinApp では、アプリケーションが何度実行されても、一貫する設定を維持した RemoveSandboxOnExit パラメータの初期値がサンドボックス ディレクトリに設定されます。

アプリケーションで子プロセスが生成された場合、ThinApp ではすべての子プロセスが終了するまで、サンドボックスを削除することはありません。アプリケーションは、子プロセスをそのまま残すように設計されている場合があります。これが原因でクリーンアップ操作がブロックされることがあります。たとえば、Microsoft Office 2003 では ctfmon.exe プロセスがそのまま保持されます。この場合は、スクリプトを使用して ctfmon.exe プロセスとその子プロセスを終了し、強制的にクリーンアップ操作を実行することができます。

RemoveSandboxOnExit スクリプトの API 関数を使用して、子プロセスの終了時にサンドボックスを削除するかどうかは、実行時に決定することができます。

例： RemoveSandboxOnExit パラメータの変更

複数のユーザーが同じユーザー名で1つのアプリケーションを共有している場合は、RemoveSandboxOnExit を変更してサンドボックスの削除を指定しておけば、サンドボックスが消去されるため、前のユーザーのレジストリやファイル システムの変更との混乱を避けることができます。

```
[BuildOptions]
RemoveSandboxOnExit=1
```

SandboxName パラメータ

SandboxName パラメータは、サンドボックスを保存するディレクトリの名前を設定します。ThinApp では、サンドボックス名としてインベントリ名を使用するように初期値が設定されています。

アプリケーションをアップグレードするとき、ユーザーが前の個人設定を保持するのか、新しい設定が必要なのかを判別するのにサンドボックス名が役立ちます。新しい展開でサンドボックス名が変更されると、別の設定でサンドボックスを作成する必要が生じる場合と、同じサンドボックスが保持される場合があります。

例： SandboxName パラメータの変更

アプリケーションを更新して、そのアプリケーションで新しいユーザー設定を使用するには、最新バージョンを反映させるように SandboxName パラメータを変更できます。

```
[BuildOptions]
SandboxName=My Application 2.0
```

SandboxNetworkDrives パラメータ

SandboxNetworkDrives パラメータは、分離モードの設定に関係なく、ThinApp が書き込み操作をネットワークドライブまたはサンドボックスのどちらに転送するのかを決定します。

このパラメータを使用して書き込み操作をネットワークドライブに転送すると、そのドライブの分離モードを Merged モードに設定した場合と同じ処理が行われます。

例： SandboxNetworkDrives パラメータの設定の変更

容量を節約したい場合や、共同作業などでファイルを共有したい場合は、SandboxNetworkDrives パラメータをデフォルト設定のままにして、変更をサンドボックスに保存せずに書き込み操作をネットワークドライブに転送させます。

```
[BuildOptions]
SandboxNetworkDrives=0
```

変更をサンドボックスに保存し、ユーザーがネットワークドライブに変更を書き込めないようにすることができます。

```
[BuildOptions]
SandboxNetworkDrives=1
```

SandboxPath パラメータ

SandboxPath パラメータは、サンドボックスへのパスを決定します。

サンドボックスへのパスは、相対パスと絶対パスのどちらも指定できます。また、フォルダ マクロまたは環境変数を含むことができ、ネットワーク ドライブ上にあっても構いません。

SandboxPath パラメータを設定して、ローカル、USB ドライブ、またはネットワークのニーズに対応したり、初期のサンドボックスの場所での容量制限を解決したり、トラブルシューティングの目的でサンドボックスをデスクトップに移動させたりすることができます。サンドボックスの名前は SandboxName パラメータによって決定されるため、サンドボックスのパスにはサンドボックス名を含めません。

例：SandboxPath パラメータの変更

SandboxPath を変更して、実行可能ファイルと同じディレクトリにサンドボックスを作成するように設定できます。SandboxName パラメータの値が Mozilla Firefox 3.0 である場合、Firefox が実行されるディレクトリに Mozilla Firefox 3.0 サンドボックスを作成できます。

```
[BuildOptions]
SandboxPath=.
```

実行可能ファイルの場所から従属するサブディレクトリにサンドボックスを作成できます。

```
[BuildOptions]
SandboxPath=LocalSandbox\Subdir1
```

ユーザーの %AppData% フォルダの Thinstall ディレクトリの下にサンドボックスを作成できます。

```
[BuildOptions]
SandboxPath=%AppData%\Thinstall
```

マッピングしたドライブにサンドボックスを格納して、サンドボックスをバックアップするか、または任意のマシンにログインするユーザー用にアプリケーションの設定を保持しておくことができます。SandboxName パラメータの値が Mozilla Firefox 3.0 である場合、Z:\Sandbox\Mozilla Firefox 3.0 にサンドボックスを作成できます。

```
[BuildOptions]
SandboxPath=Z:\Sandbox
```

SandboxRemovableDisk パラメータ

SandboxRemovableDisk パラメータは、アプリケーションがリムーバブル ディスクの変更をディスクまたはサンドボックスに書き込めるかどうかを決定します。リムーバブル ディスクには、USB フラッシュ デバイスやリムーバブル ハード ドライブなどがあります。

ThinApp では、アプリケーションがリムーバブル ディスク ファイルの変更をディスクに書き込むように初期値が設定されています。

例： SandboxRemovableDisk パラメータの変更

容量を節約するために、SandboxRemovableDisk パラメータを変更して、リムーバブル ディスクの変更をサンドボックスに転送するように設定できます。リムーバブル ディスクの分離モードに応じて、ディスクに格納されたファイルの変更内容をサンドボックスまたはリムーバブル ディスクに置くことができます。

```
[BuildOptions]
SandboxRemovableDisk=1
```

SandboxWindowClassName パラメータ

SandboxWindowClassName = 1 を設定すると、ThinApp パッケージ内で作成され、使用されるアプリケーション定義のウィンドウ クラス名を分離できます。

その他の構成パラメータ

22

いくつかの ThinApp パラメータを使用して、プリンタを構成したり、カット アンド ペースト オプションを無効にしたり、コンピュータやグループ SID の設定を指定することができます。また、ステータス バーに表示するユーザー名を変更することもできます。

この章には、次のトピックが含まれています。

- [DisableCutPaste](#) パラメータ
- [LoadDotNetFromSystem](#) パラメータ
- [PermittedComputers](#) パラメータ
- [Services](#) パラメータ
- [StatusbarDisplayName](#) パラメータ
- [DisableTransactionRegistry](#) パラメータ
- [PreventDLLInjection](#)
- [ProcessExternalNameBehavior](#) パラメータ
- [PreventDllInjectionExceptions](#) パラメータ
- [LargeAddressAware](#) パラメータ
- [PermittedComputers](#) パラメータ
- [PermittedComputersAccessDeniedMsg](#) パラメータ
- [PermittedComputersOfflineAccess](#) パラメータ
- [IgnoreDDEMessages](#) パラメータ
- [Comment](#) パラメータ
- [CopyManifestData](#) パラメータ
- [DllNamesMap](#)
- [EnableExternalHookInterception](#)
- [IgnoreFolderSecurity](#)
- [DisableWriteDAC](#)

■ AddIMESupport

DisableCutPaste パラメータ

ThinApp アプリケーションでカット アンド ペーストオプションを無効にするには、DisableCutPaste パラメータを使用します。

LoadDotNetFromSystem パラメータ

LoadDotNetFromSystem パラメータは、ThinApp アプリケーションが、ThinApp アプリケーションにある .NET インストールではなく、ユーザーのシステムに存在する .NET インストールを使用することを指定するものです。

PermittedComputers パラメータ

PermittedComputers パラメータを使用すると、仮想アプリケーションの使用を許可されるコンピュータの Active Directory グループのセットを少数に制限することで、セキュリティの強化が図れます。

Services パラメータ

Services パラメータは、ThinApp パッケージをシステム サービスとして登録する方法を指定します。

StatusbarDisplayName パラメータ

StatusbarDisplayName パラメータにより、ThinApp アプリケーションの開始ダイアログ ボックスのステータス バーにアプリケーションのタイトルが表示されます。この値を変更して、ステータス バーに別のアプリケーション名を表示することができます。

DisableTransactionRegistry パラメータ

DisableTransactionRegistry パラメータを使用して、書き込み操作中にアプリケーションに障害が発生した場合、電源の障害が発生した場合、または ThinApp パッケージに存在するリムーバブル ディスク上のディスク フラッシュが不完全な場合に、データの破損を防止します。

サンドボックスがネットワーク共有に配置されている場合、特殊な状況において、DisableTransactionRegistry パラメータによってパフォーマンスが向上することがあります。

PreventDLLInjection

PreventDllInjection パラメータを使用することにより、別の外部アプリケーションが SetWindowHook を呼び出してグローバル フックを設定するときに、ThinApp ランタイムが DLL を読み込まないようにすることができます。

このパラメータは、フックと仮想アプリケーションが競合する場合に限り使用します。たとえば、仮想アプリケーションで、別のアプリケーション NxPowerLite からグローバル フック oehook.dll を設定する呼び出し SetWindowsHook があったときに起動がうまくいかない場合、このエントリを用いて package.ini を変更することにより、仮想アプリケーションの起動が可能になります。

ProcessExternalNameBehavior パラメータ

ProcessExternalNameBehavior パラメータを使用して、子プロセスと親プロセスと区別できるようになります。デフォルトでは、タスク マネージャの子プロセスと親プロセスの名前が両方、エントリ ポイント名になります。ただし、ProcessExternalNameBehavior がデフォルトに設定されている場合、親プロセスには引き続きエントリ ポイント名が保持される一方、子プロセスにはネイティブ アプリケーションの名前が与えられます。このパラメータを使用すると、一部のシナリオで親プロセスと子プロセスを区別できます。

PreventDllInjectionExceptions パラメータ

PreventDllInjectionExceptions パラメータは、PreventDllInjection パラメータが 1 に設定されている場合にのみ使用できます。PreventDllInjectionExceptions パラメータを設定すると、PreventDllInjection が true に設定されている場合でも、指定した dll ファイルを読み込めるようになります。

例

```
PreventDllInjectionExceptions=<inject.dll>;<injectme2.dll>
```

注： このパラメータは、PreventDllInjection=1. の場合にのみ使用してください。

LargeAddressAware パラメータ

LargeAddressAware パラメータを使用して、32 ビットと 64 ビットの両方のオペレーティング システムにインストールするアプリケーションをキャプチャできます。

ThinApp では、32 ビットと 64 ビットの両方に対応したアプリケーションとコンポーネントを同時にインストールするアプリケーションをキャプチャすることができます。32 ビット版のアプリケーションで 64 ビット版を起動することができます。ThinApp を使用する場合、32 ビット アプリケーションからの 64 ビット アプリケーションの起動がまれに失敗することがあります。この問題を解決するには、package.ini ファイルで package.ini パラメータ LargeAddressAware を 1 に設定して、パッケージを再構築します。

```
[BuildOptions]
LargeAddressAware=1
```

PermittedComputers パラメータ

キャプチャしたアプリケーションが起動されると、PermittedComputers パラメータにより、そのコンピュータが指定された Active Directory グループのメンバーになっているかどうかを検証されます。コンピュータが Active Directory グループのメンバーでない場合、ThinApp はアプリケーションを起動しません。

このパラメータは、次のように使用できます。

```
PermittedComputers=xpsystemgroup;win7systems
```

アプリケーション PDC に対する Package.ini エントリにより、グローバル設定がオーバーライドされ、すべてのエントリ ポイントで PDC から各設定が継承されます。例：

```
[BuildOptions]
PermittedComputers=OfficeComputers

[Microsoft Office 2010.dat]
PermittedComputers=xpsystemgroup;

[Microsoft Word 2010.exe]
...
...
[Microsoft excel 2010.exe]
...
...
```

上の例では、[Microsoft Office 2010.dat] の設定 PermittedComputers（このファイルが PDC であるため）により、[BuildOptions] で指定されたグローバル オプションがオーバーライドされます。Word および Excel のアプリケーションでは、[Microsoft Office 2010. dat] から設定が継承されます。

PermittedComputersAccessDeniedMsg パラメータ

PermittedComputersAccessDeniedMsg パラメータを使用して、ユーザーにエラー メッセージを表示することができます。エラー メッセージは、システムが PermittedComputers のメンバーでない場合にのみ表示されません。ただし、このエラー メッセージは数秒後に自動的に消えます。

PermittedComputersOfflineAccess パラメータ

PermittedComputersOfflineAccess パラメータを有効にすると、ドメイン コントローラで許可されたコンピュータ グループに含まれるコンピュータからオフライン モードで仮想アプリケーションにはアクセスできなくなります。パラメータが有効でない場合は、許可されたコンピュータから引き続き仮想アプリケーションを使用できます。

IgnoreDDEMessages パラメータ

Internet Explorer 6 の仮想化パッケージに対して、IgnoreDDEMessages パラメータが自動的に設定されます。このパラメータが設定されている場合、アプリケーションはアプリケーションが受信する DDE メッセージを無視します。IgnoreDDEMessages を使用して、ドキュメントのオープン操作が、仮想アプリケーションではなくネイティブ アプリケーションで常開始するようにすることができます。

Comment パラメータ

Comment はエントリ ポイントごとに指定され、ユーザーがマウスをエントリ ポイントのショートカットの上に置くとときに表示される内容を決定します。何も指定しない場合は、エントリ ポイントへのパスが表示されます。

例：コメント

```
[Entry Point Section]
Comment=This is your default browser
```

CopyManifestData パラメータ

CopyManifestData パラメータを使用して、ソースの実行可能ファイルから、Package.ini ファイルに記載されている ThinApp ショートカットの実行可能ファイルに、アプリケーションのマニフェスト データをコピーします。

このパラメータは、Package.ini の General Purpose Parameters セクションに追加できます。アプリケーションのキャプチャ プロセスでは、このパラメータは生成されません。CopyManifestData パラメータの値は **1** にする必要があります。

```
CopyManifestData=1
```

DllNamesMap

DllNamesMap パラメータは、短いファイル名と対応する長いファイル名が一致しないために、仮想アプリケーションが正常に動作しない場合に使用されます。このパラメータを使用して、ファイルの短縮名をファイルの長い名前前にマッピングすることができます。

ネイティブ アプリケーションが存在しない場合、オペレーティング システムはファイルの短縮名を長い（または完全な）ファイル名に変換することができません。長い名前は、パッケージに保存されたファイルの名前です。パラメータ内の各マッピングはコロンで区切られており、別のマッピングはセミコロンで区切られています。

このパラメータは、Package.ini の General Purpose Parameters セクションに追加できます。

例：

たとえば、仮想アプリケーションでは、shortn~1.dll および shortn~2.dll のような短縮ファイル名が格納され使用されます。これらのファイルの長いファイル名は、それぞれ fullname1.dll、fullname2.dll になります。短縮ファイル名を対応する長いファイル名にマッピングするには、DllNamesMap パラメータを次のように使用する必要があります。DllNamesMap=shortn~1.dll:fullname1.dll; shortn~2.dll:fullname2.dll

EnableExternalHookInterception

一部のアプリケーションは、システムに追加のフック DLL をインストールします。フック DLL は ThinApp 内の子プロセスの作成に干渉するため、仮想アプリケーションに障害が発生する原因となります。フック DLL が存在する場合にアプリケーションが失敗しないようにするために、このパラメータを使用できます。

このパラメータは、Package.ini の General Purpose Parameters セクションに追加できます。

例：

たとえば、システムにフック DLL が存在する場合、次の手順を実行してアプリケーションが失敗しないようにします。

- 1 Package.ini ファイルで、EnableExternalHookInterception パラメータを **1** に設定します。

IgnoreFolderSecurity

IgnoreFolderSecurity パラメータは、特定のフォルダのセキュリティ例外を許可するために使用されます。

ThinApp パッケージからファイルに初めてアクセスするときには、ファイルパスに対応する完全なフォルダ構造がサンドボックスに作成されます。パス内のいずれかのフォルダに、子フォルダの作成を許可しないセキュリティ属性セットがある場合は、IgnoreFolderSecurity パラメータを使用できます。

パラメータの形式は、IgnoreFolderSecurity=foldername1;foldername2 のような、セミコロン区切りのフォルダ名のリストになります。この形式では、foldername はフォルダの名前またはフォルダへの完全パスのいずれかになります。

このパラメータは、Package.ini の General Purpose Parameters セクションに追加できます。

ここでは、XYZ がフォルダの名前であり、C:\abcd が abcd フォルダへのパス全体を示している例を考えてみます。パラメータの形式は次のとおりです。IgnoreFolderSecurity=XYZ;C:\abcd

名前の競合を回避するために、完全パスを使用することができます。これは、ThinApp パッケージ内にフォルダ名が同じで、パスが異なる複数のフォルダが存在していて、そのうちの1つにのみセキュリティ例外を適用する必要がある場合に適用されます。

DisableWriteDAC

サンドボックスがネットワーク上の場所にあり、ユーザーがその場所に対する WRITE_DAC 権限を持っていない場合、アクセスが拒否されました というエラーメッセージが表示され、ThinApp を使用して仮想化されたアプリケーションでは一部のファイルを開くことができません。この問題を解決するには、DisableWriteDAC パラメータが使用されます。

DisableWriteDAC = 1 の場合、サンドボックスでファイルを開いたり作成するときに、ThinApp のランタイムに WRITE_DAC 権限は確認されません。

このパラメータは、Package.ini の General Purpose Parameters セクションに追加できます。

AddIMESupport

仮想 Internet Explorer では、それぞれの言語パックを介してインストールされる追加の言語のサポートが必要になる場合があります。これらの追加の言語パックを仮想 Internet Explorer で使用するには、AddIMESupport パラメータを使用できます。

このパラメータは、Package.ini の General Purpose Parameters セクションに追加できます。

たとえば、次の手順を実行すると、仮想 Internet Explorer で日本語の言語パックを使用できます。

- 1 Package.ini で、AddIMESupport パラメータを **1** に設定します。