

Rollout-Vorbereitung mit Sconfig

Best Practices

Schon bei einer geringen Zahl von Arbeitsplätzen, auf denen eine bestimmte Software, z.B. Hummingbird Exceed, identisch installiert werden soll, stellt sich das Problem des automatisierten Software-Rollouts. Bereits bei 10 PCs möchte niemand 10 Mal die Installations-CD einschieben und Punkt für Punkt die Installationsroutine abarbeiten.

Microsoft hat mit Windows 2000 zu diesem Zweck den sog. Windows-Installer eingeführt und auch als verbindliches Installationsverfahren vorgeschrieben. Software, die irgend eine Kompatibilitätsbescheinigung von Microsoft erhalten will, muss bei der Installation auf den Windows-Installer aufsetzen.

Das MSI-Paket

Software-Pakete, die durch den Windows-Installer installiert werden sollen, haben das Format eines MSI-Pakets. Zentral für solche Pakete ist eine Datei mit der Endung .msi. Sie enthält die komplette Beschreibung der Installation:

- welche Dateien werden in welche Verzeichnisse kopiert
- welche Einträge werden in die Registry gemacht
- welche Komponenten der Software insgesamt zur Verfügung stehen und welche per Default installiert werden
- diverse sog. Installer-Properties, die das Installationsverhalten steuern
- zu registrierende DLLs, neue oder zu verändernde Umgebungsvariablen und vieles mehr

Im einfachsten Fall enthält die *.msi-Datei bereits als Archiv sämtliche zu installierenden Dateien der Software. Bei größeren Programmpaketen besteht das MSI-Paket aus einer Verzeichnisbaumstruktur mit den zu installierenden Dateien und der *.msi-Datei mit Verweisen auf diese Dateien.

TRANSFORMS: Abweichungen vom Default

Führt man eine automatisierte Installation alleine mit dem MSI-Paket durch, so werden immer die Default-Komponenten mit den in der *.msi-Datei vorgegebenen Parametern installiert. In der Regel will man aber eine abweichende Komponenten-Auswahl, am besten bereits vorkonfiguriert für die Bedürfnisse des jeweiligen Standorts ausrollen. Der Windows-Installer sieht dafür sog. TRANSFORMS vor; dies sind Datei der Endung .mst, die genau die Differenz zwischen dem in der *.msi-Datei festgelegten Default und der Wunschinstallation beschreibt. Mit der Kombination msi/mst ist der Windows-Installer dann in der Lage, auch voll automatisiert die Wunschinstallation durchzuführen. Alle gängigen Software-Verteilssysteme sind heute in der Lage, solche differenziert angepassten Rollout-Pakete auf der Basis von *.msi-/*mst-Dateien auf eine große Anzahl von Arbeitsplätzen in kurzer Zeit zu verteilen.

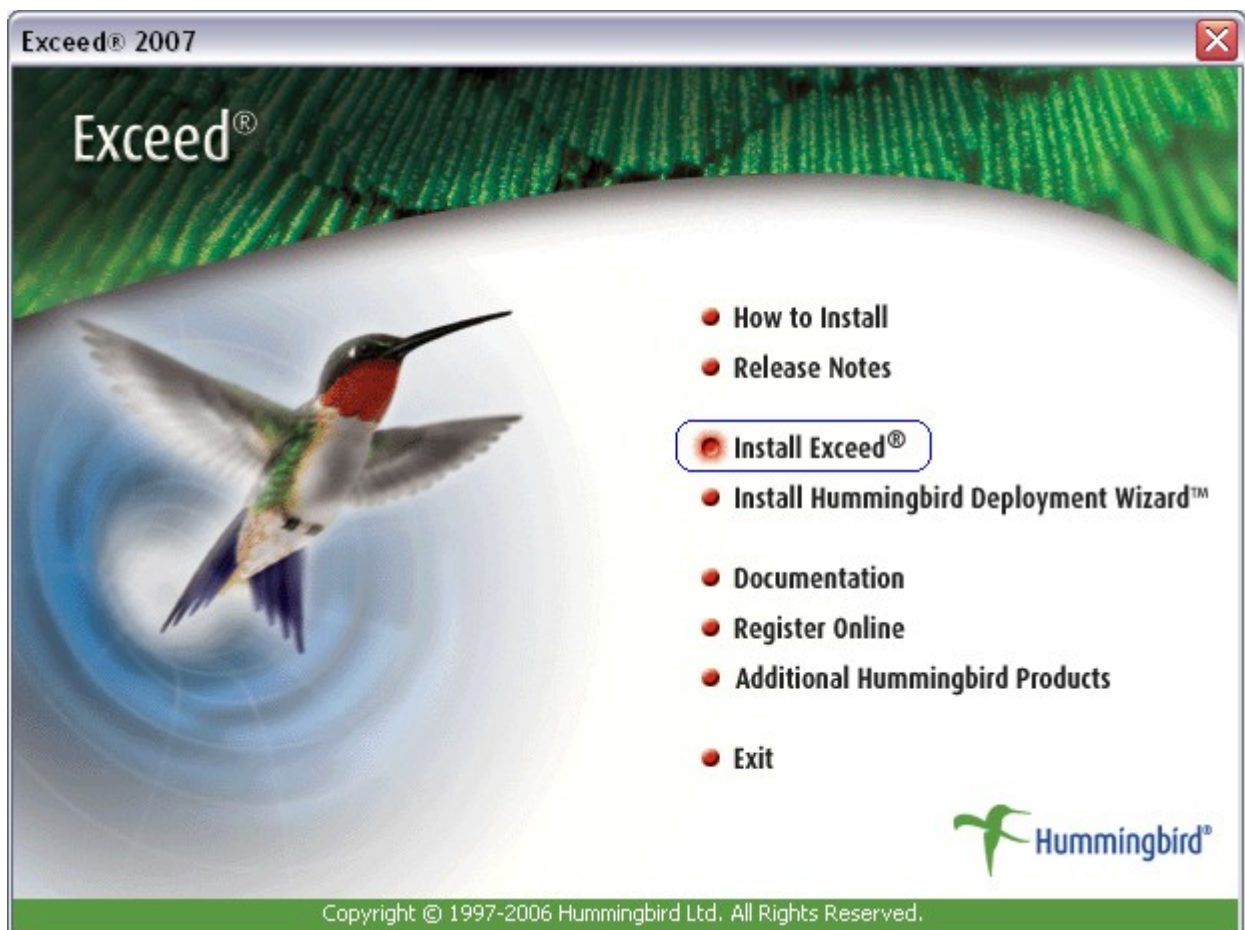
Zur Erstellung von TRANSFORMS (dh. *.mst-Dateien) stehen generische Dritthersteller-Tools zur Verfügung, doch die Aufgabe gestaltet sich oft als sehr schwierig, da man in vielen Fällen erraten muss, was einzelne Installer-Properties im MSI-Paket bedeuten.

Hummingbird Connectivity (heute Teil der Opentext Corporation) stellt für Exceed sowie für seine anderen traditionellen Connectivity-Produkte (HostExplorer, NFS Maestro-Client etc.) ein einfach zu bedienendes Tool zur Erstellung von Transforms (*.mst-Dateien) zur Verfügung: **Sonfig**. Im Folgenden soll als Beispiel Schritt für Schritt gezeigt werden, wie ein optimal angepasstes Rollout-Paket für *Exceed Version 2007* erstellt werden kann. Die Empfehlungen beruhen auf den Erfahrungen zahlreicher großer Exceed-Rollouts auf jeweils einige Hundert bis einige Tausend Arbeitsplätze.

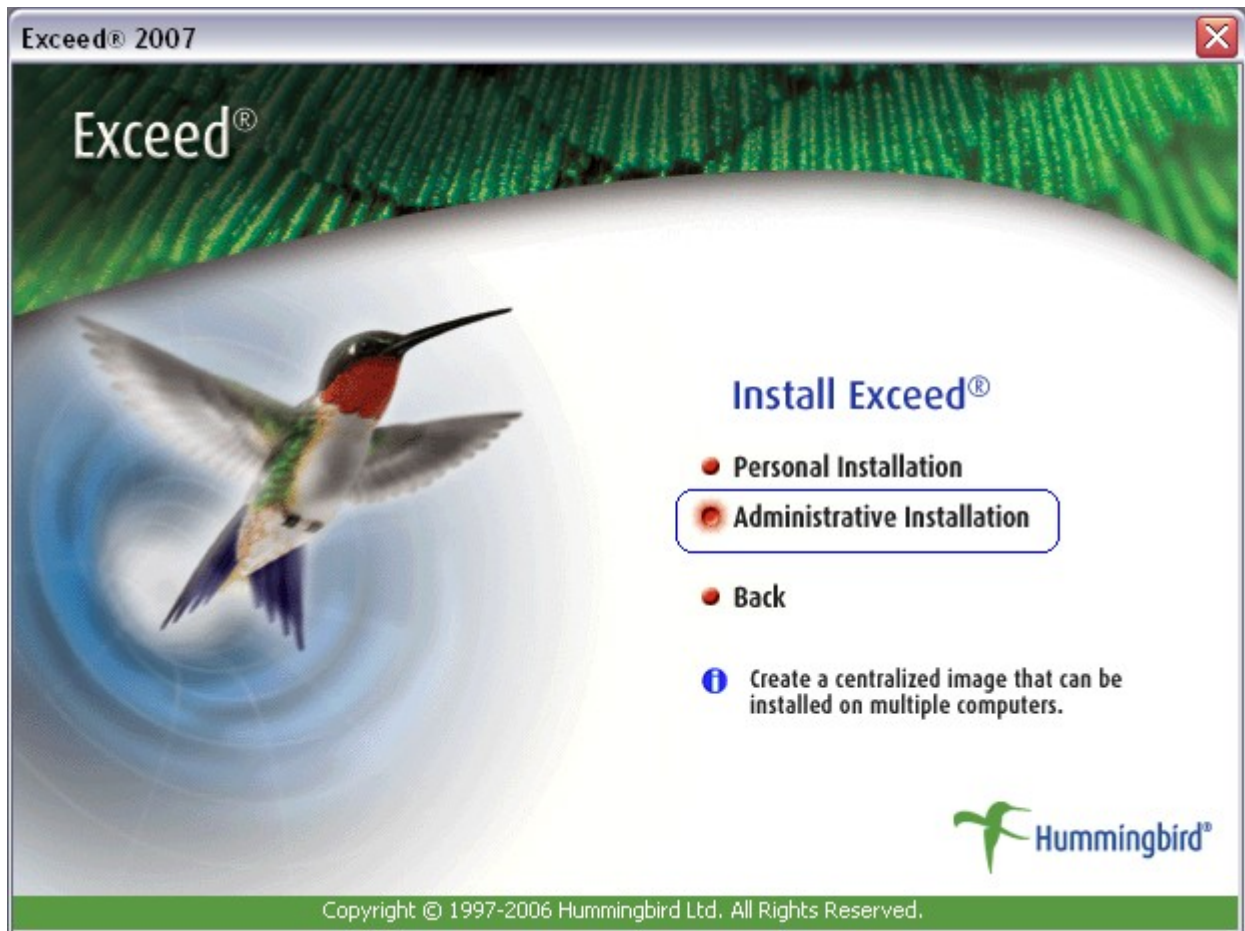
Das MSI-Paket von Exceed

Im Prinzip liegt Exceed bereits auf der Installstions-CD als MSI-Paket vor, und zwar als Verzeichnisbaum mit sämtlichen zu installierenden Dateien und der „Exceed 2007.msi“, die den gesamten Installationsvorgang steuert. Um aber für den Software-Rollout eine optimal angepasste Installtionsquelle vorzubereiten, muss zunächst eine sog. Administrative Installation durchgeführt werden.

Nach Einlegen der CD oder Aufruf von Msetup.exe erscheint:



Nach Klick auf „Install Exceed“ wählt man nun die „Administrative Installation“:



Eine der wenigen Fragen, die bei der administrativen Installation gestellt werden, ist die nach der Installationsprache:

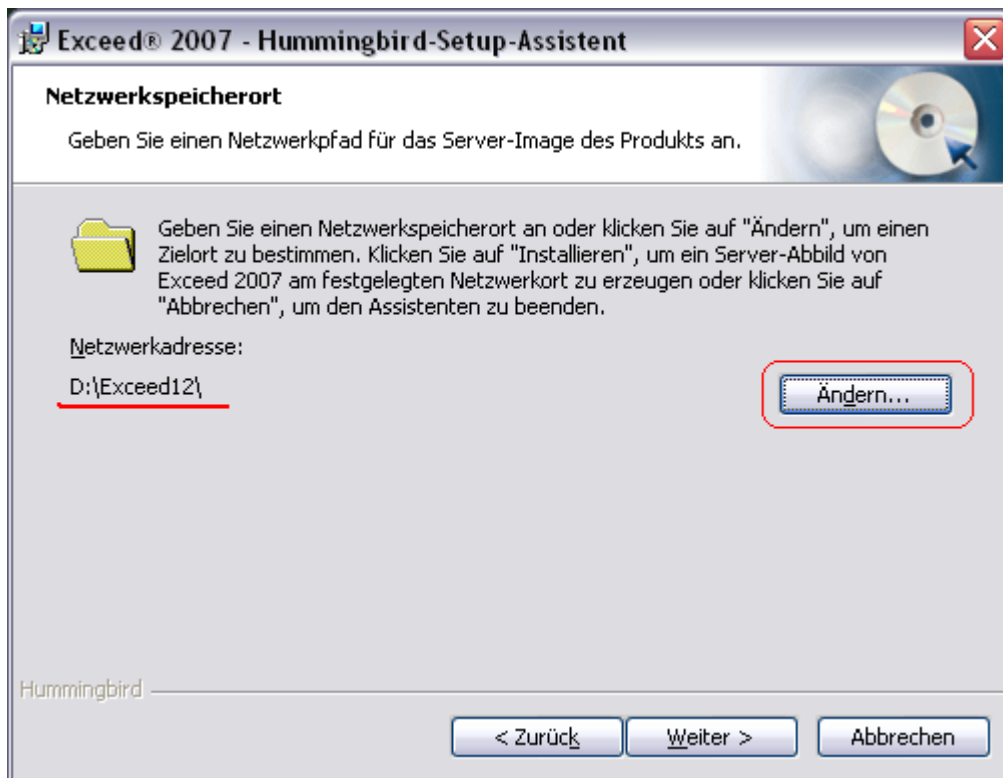


In dieser Sprache wird die Installation selbst ablaufen. Die Sprache, die Exceed dann für seine Benutzeroberfläche wählt, ist davon unabhängig. Sie richtet sich meist nach den Regions-Einstellungen des darunter liegenden Windows und lässt sich jederzeit mit dem Language Chooser von Exceed – auch im laufenden Betrieb – umstellen.

Eine Vorentscheidung wird damit aber doch getroffen, die später nicht mehr umstellbar ist (wegen der engen Integration in den Windows Explorer):

Je nach der hier gewählten Sprache, wird später im Explorer eine „Hummingbird Umgebung“ oder eine „Hummingbird Neighborhood“ erscheinen.

Msetup bietet nun einen Pfad auf eine Netzlaufwerk an. Zunächst ist es aber sinnvoller, das administrative Abbild (= die künftige Installationsquelle) auf der lokalen Festplatte anzulegen, in unserem Beispiel in das Verzeichnis D:\Exceed12\.



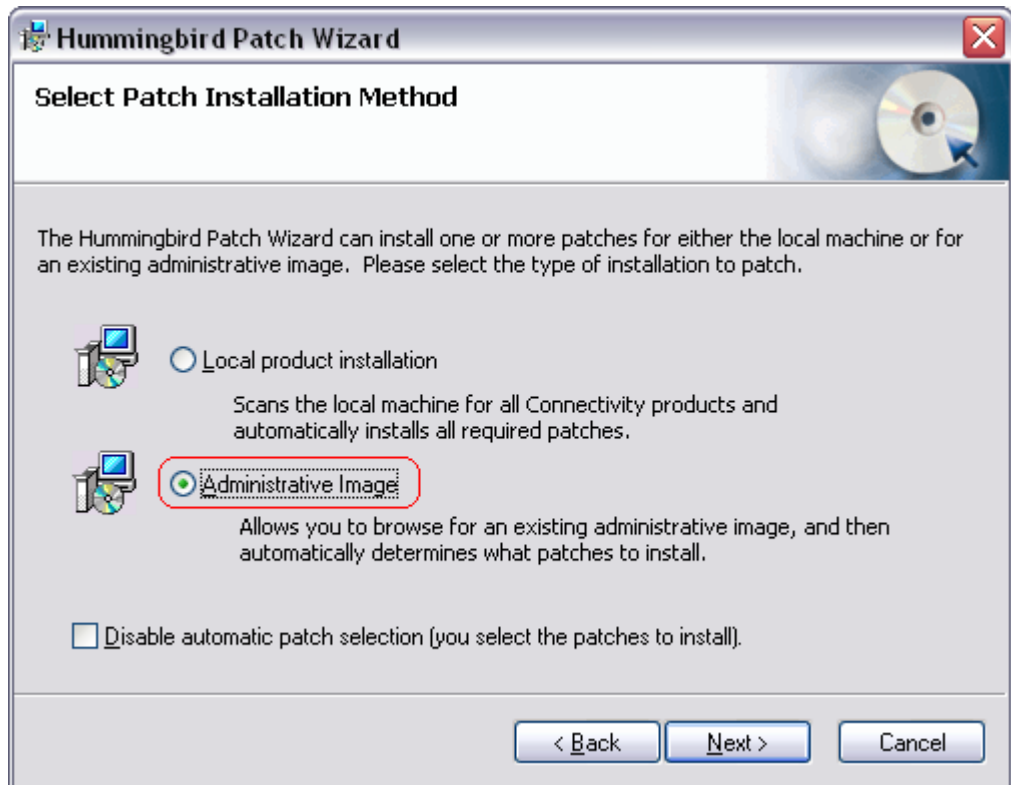
Dies beschleunigt die weiteren Arbeitsschritte, insbesondere das Patchen der Installationsquelle. Erst wenn alles fertig ist, wird man die angepasste Installationsquelle auf ein zentral erreichbares Netzwerklaufrwerk kopieren.

Aktualisierung der Installationsquelle: UpdatePack

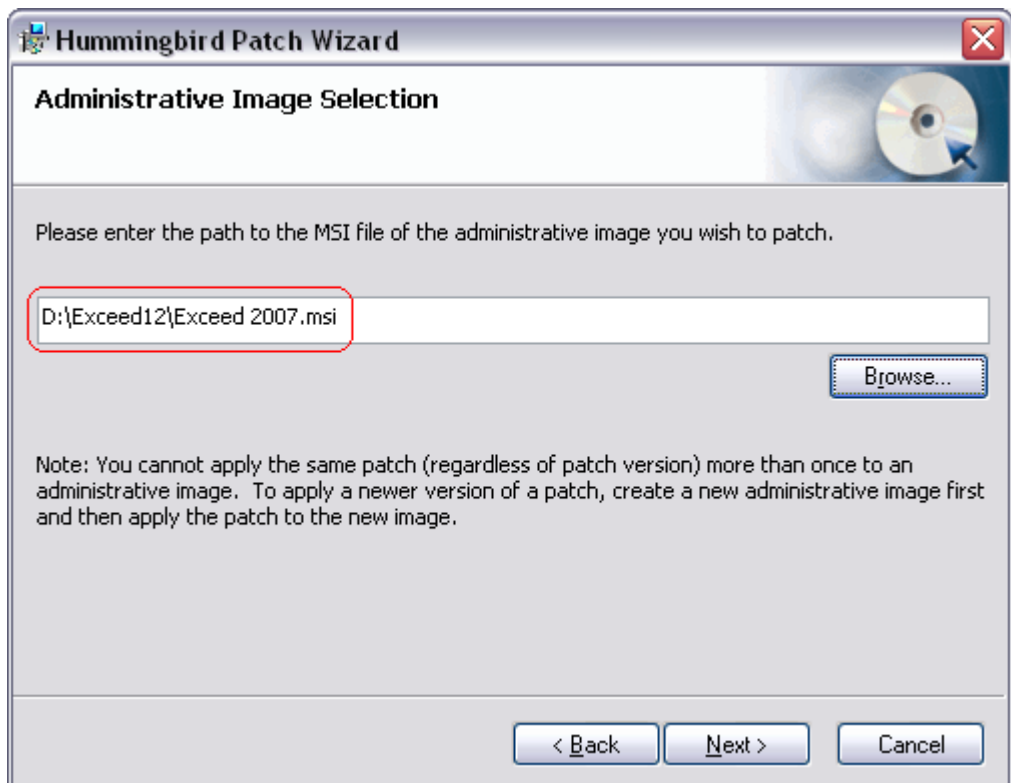
Hummingbird Connectivity stellt für supportete Software-Versionen kumulative UpdatePacks bereit, die alle bis dahin veröffentlichten relevanten Patches enthalten. Für Version 2007 steht – Stand Mai 2007 – das UpdatePack 3 in einer ausführbaren Datei (Connectivity2007_UpdatePack3.exe) zur Verfügung. Es wird unbedingt empfohlen, vor einem Rollout das Installationspaket mit Hilfe des UpdatePacks auf den aktuellen Stand zu bringen. Dies geschieht folgendermaßen.

Nach Aufruf des UpdatePacks ist zur Aktualisierung der Installationsquelle „Administrative Image“ zu wählen.

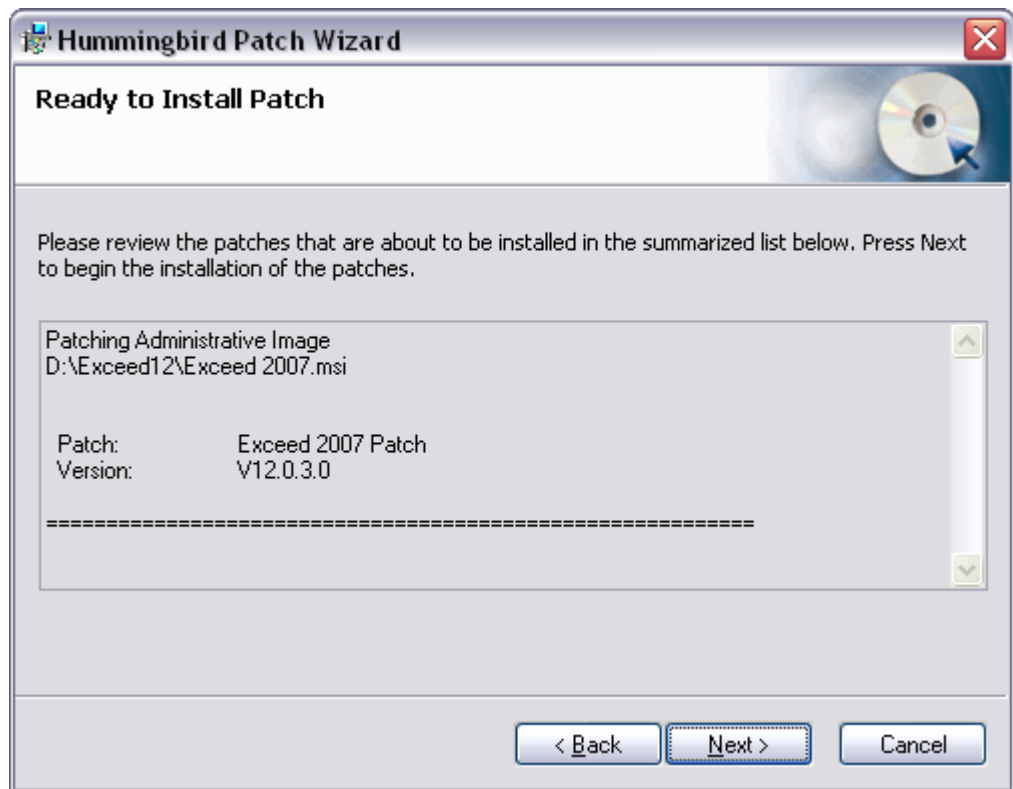
„Local Product Installation“ dagegen patcht bereits installierte Produkte.



Nun ist die *.msi-Datei des Installationspakets anzugeben.



Die zu aktualisierenden Komponenten werden automatisch identifiziert ...



... und schon ist das Update fertig.

Alle Installationen, die von dieser Quelle aus werden somit bereits das aktuelle Patch-Level enthalten.



Eine Musterinstallation aus Vorbild

Als nächsten Schritt wird man „händisch“ eine Exceed-Installation vornehmen, indem man das setup.exe in der aktualisierten Installationsquelle aufruft und im Rahmen einer benutzerdefinierten Installation die gewünschten Komponenten und eben auch Sconfig auswählt:



Mit dieser Musterinstallation wird man angepasste Startprofile (*.xs- und *.ses-Dateien für Exceed, *.hep-Dateien für HostExplorer), Tastaturbelegungen und sonstige Konfigurationsdateien erstellen (z.B. eine Exceed.xcfg mit den gewünschten X-Server-Einstellungen und geeignet vorkonfigurierte exceed.ini, xstart.ini etc.).

Nach geeigneten Testläufen kopiert man alle angepassten Dateien in ein noch anzulegendes Verzeichnis „CustomFiles“ in der Installationsquelle, also in unserem Beispiel in

D:\Exceed12\CustomFiles\.

Dies dient lediglich der Übersichtlichkeit und hat noch keinen Einfluss auf künftige Installationen.

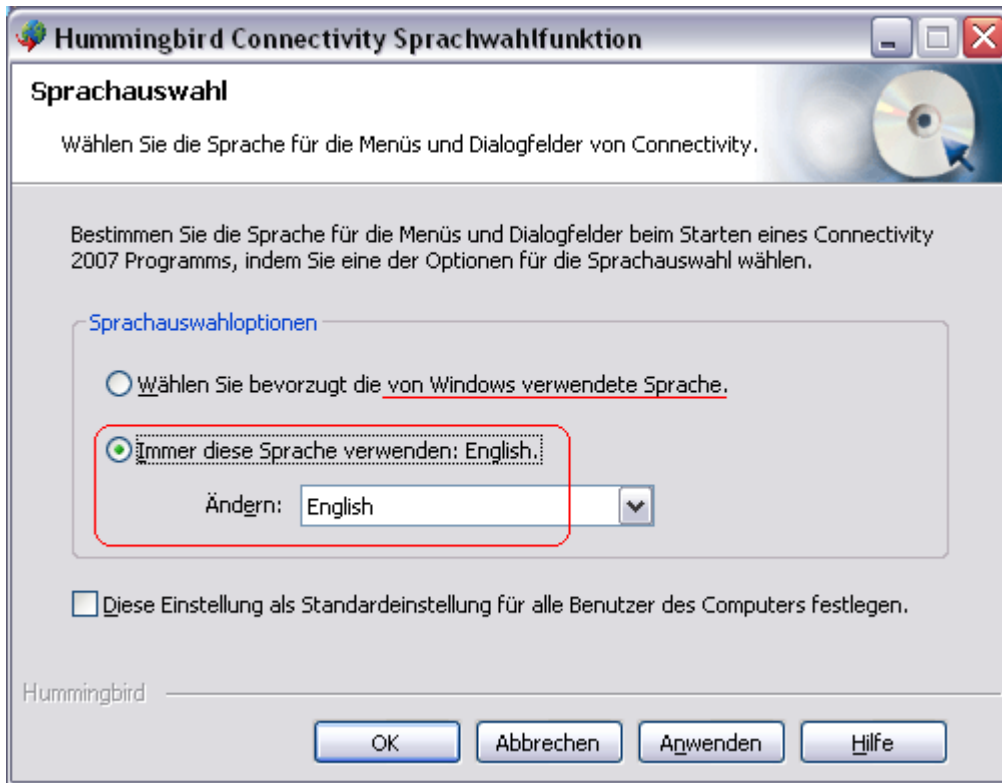
Tip: In der Exceed.xcfg sollten nur Einstellungen hinterlegt werden, die für alle Exceed-Operationen gelten: z.B. Cpy&Paste-Verhalten, X-Extensions.

Eigenschaften einzelner Session (z.B. XDMCP) sollten über Xsessions definiert werden. Detaillierte deutsche Erläuterungen dazu findet man auf <http://x11.gweb.info/win/>.

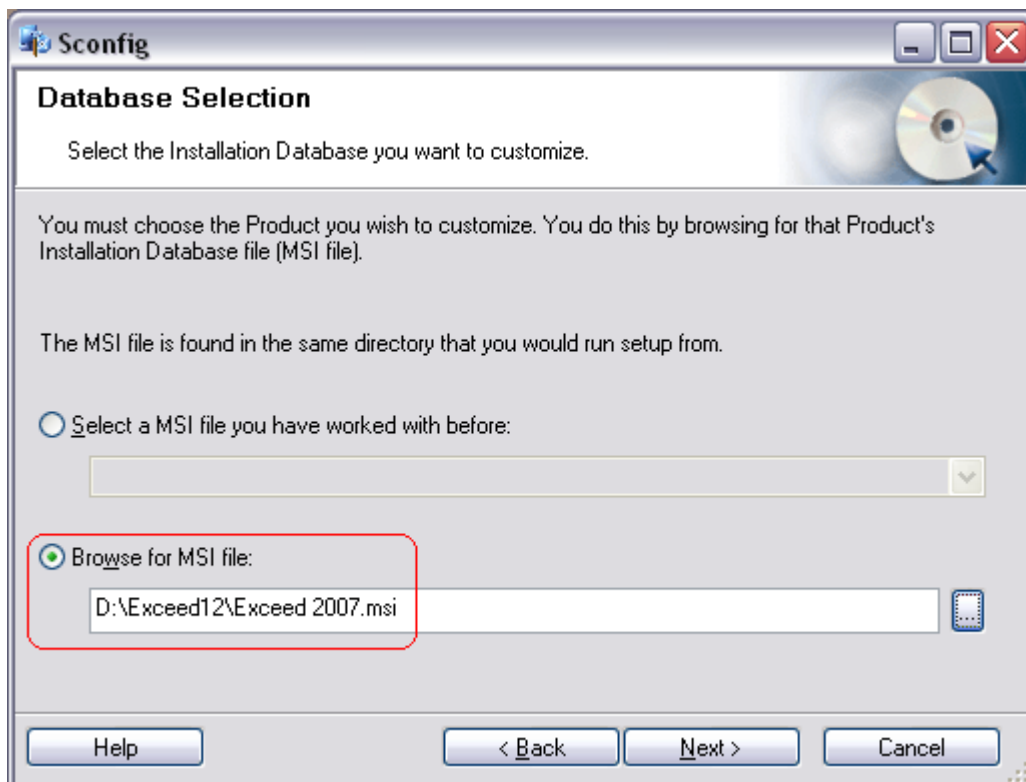
Außerdem sei hier erwähnt, dass eine komplette *.xcfg-Datei oder auch einzelne Teile davon mit einem Passwort geschützt werden können, so dass sie der Anwender nicht mehr verändern kann.

Der entscheidende Schritt: Sconfig

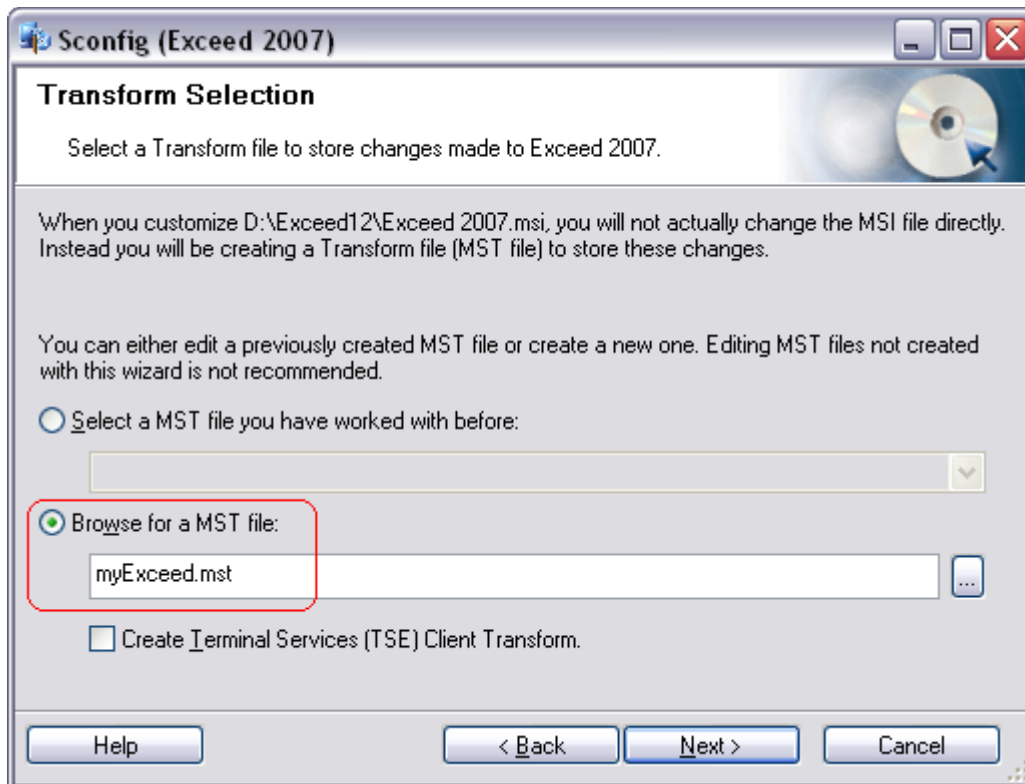
Nun geht es darum, mit Sconfig alle gewünschten Anpassungen in einer *.mst-Datei für die automatisierte Installation zu hinterlegen. Da Sconfig gelegentlich Schwierigkeiten mit der deutschen Bedienungsoberfläche hat, sollte man aber zunächst mit dem „Language Chooser“ (den man in der Hummingbird Connectivity-Programmgruppe unter den Administrative Tools findet) die GUI-Sprache auf Englisch eingestellt werden:



Nun kann es losgehen. Sconfig verlangt (logischerweise) zunächst die *.msi-Datei:



Dann ist der frei wählbare Name der *.mst-Datei anzugeben:

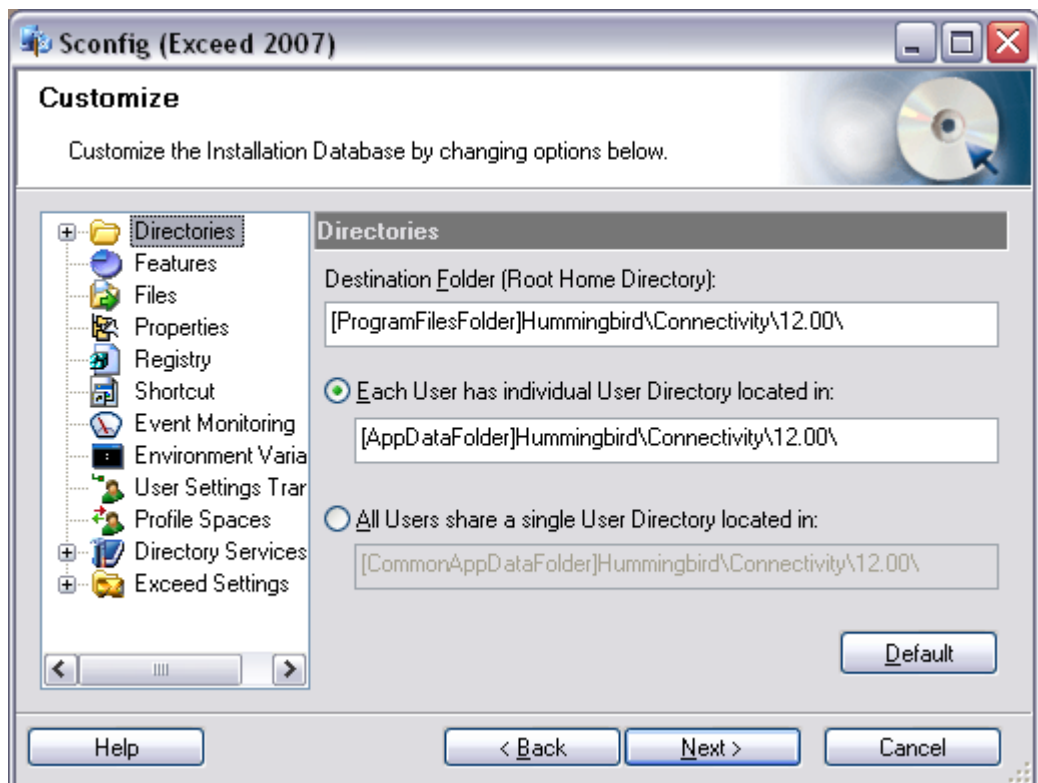


Auch wenn auf einem Windows Terminal-Server installiert werden soll, ist hier von der Option „Create Terminal Services (TSE) Client Transform“ abzuraten. Vielmehr ist in diesem Fall die entsprechende Transform-Datei ExceeTSE.mst bereits auf die administrative Installation anzuwenden. Detailliert wird dies in einem Whitepaper von Hummingbird Connectivity erläutert, ebenso in der auf der Installations-CD enthaltenen Dokumentation (ConnectivityInstallation.pdf).

Installationspfade

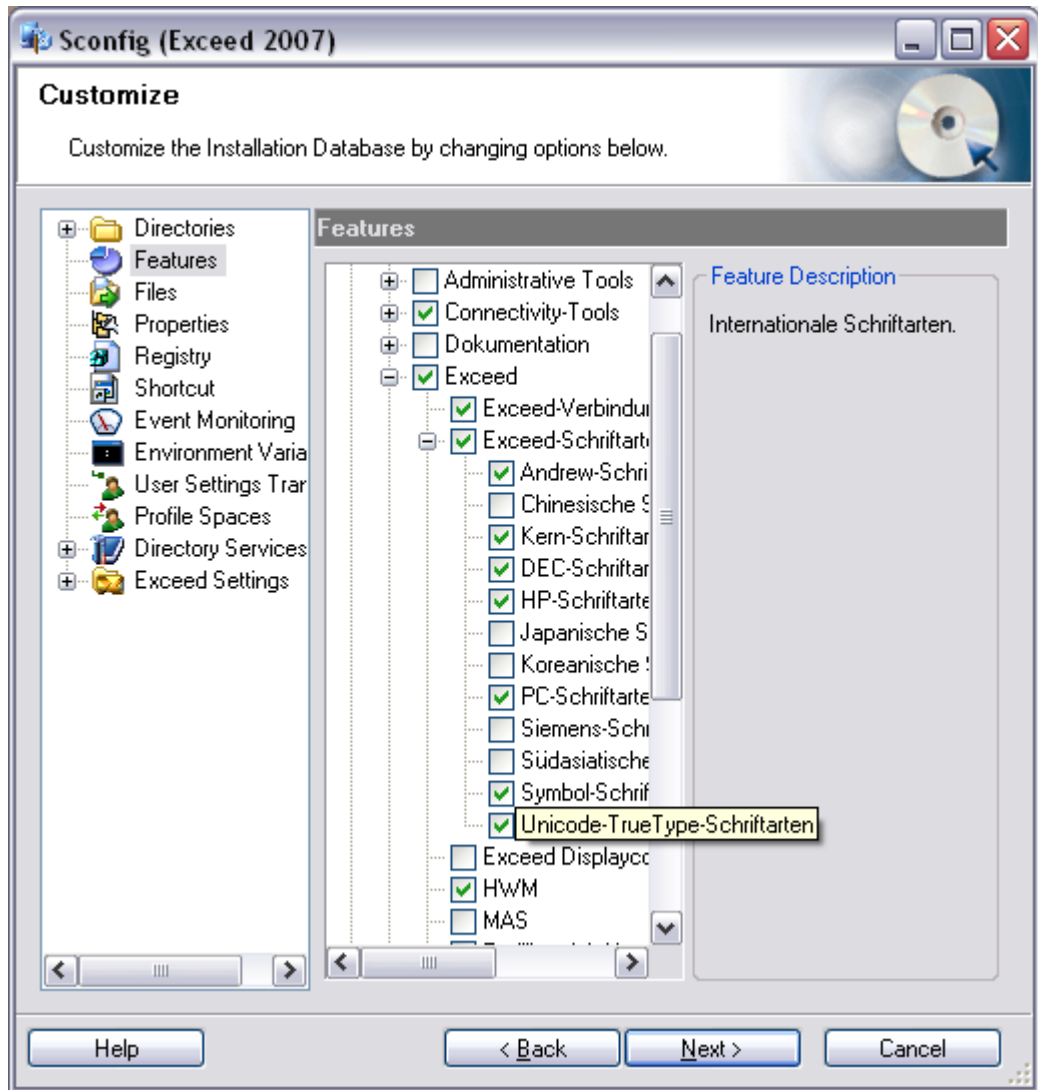
Das oberste Menü von Sconfig erlaubt, Installationspfade für die Programme und für die Konfigurationsdateien anzugeben.

Favorisiert wird hier die Variante, für jeden User („Each User“) ein eigenes Verzeichnis anzulegen (meist in seinem Profil). Dies ist heute Standard und am besten ausgetestet.



Zu installierende Komponenten

Unter „Features“ lässt sich jede Software-Komponente für die Installation aus- oder abwählen.



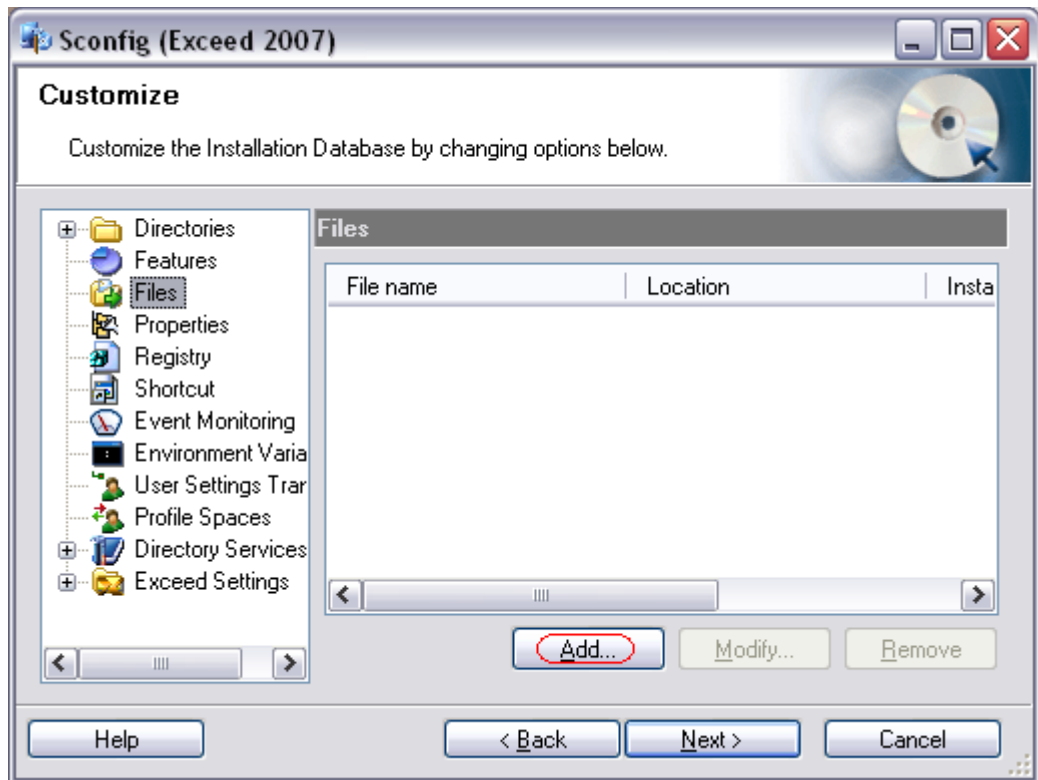
Im Hinblick auf den optimalen Betrieb des Exceed X-Servers selbst sollte man sorgfältig die benötigten Schriftarten auswählen.

Zu wenig Schriften führen zu Problemen bei der Darstellung, zu viele können zu Performance-Verlusten führen.

Allerdings kann diese Auswahl auch noch nachträglich in der Exceed-Konfiguration eingeschränkt werden.

Angepasste Konfigurationsdateien und Startprofile

Unter „Files“ erlaubt Sconfig, die zuvor angepassten Konfigurationsdateien in das Installationspaket zu integrieren.

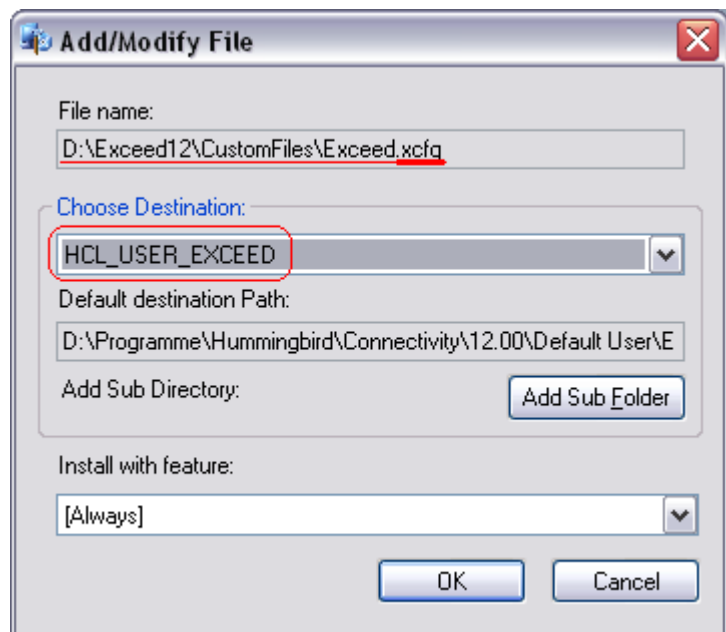


Mit „Add“ wählt man eine Datei nach der anderen aus D:\Exceed12\CustomFiles\ aus und erreicht dann folgendes Menü:

Etwas heikel ist hier lediglich die Auswahl des geeigneten Zielordners („Destination“) für die jeweilige Konfigurationsdatei. Hier ist aus einer beeindruckenden Liste von internen Properties zu wählen, die jeweils ein Verzeichnis repräsentieren. Jedes Programm- oder Konfigurationsverzeichnis von Exceed ist anwählbar, zudem generische Verzeichnisse wie „Desktop“. Es ist auch möglich, zusätzliche Unterverzeichnisse anzulegen.

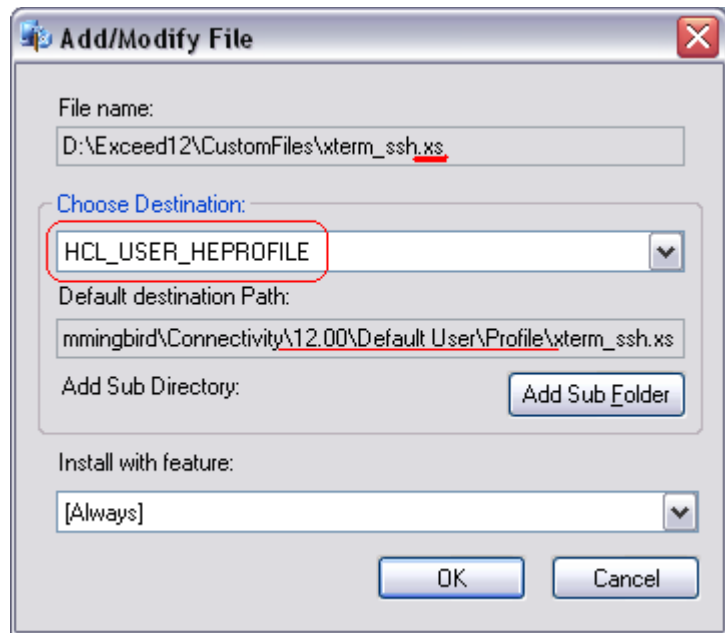
Kurz gesagt, kommen Exceed-Konfigurationsdateien, also .xcfg-Dateien sowie Tastaturbelegungen (*.kbf) und diverse ini-Dateien (exceed.ini, xstart.ini, ecm.ini) in „HCL_USER_EXCEED“.

Man sieht jeweils darunter den tatsächlichen Pfad, in den die Datei bei einer Installation in die Default-Pfade kopiert würde. Nicht berücksichtigt sind hier allerdings abweichende Verzeichnis-Einstellungen unter der Rubrik „Directories“ (siehe oben).

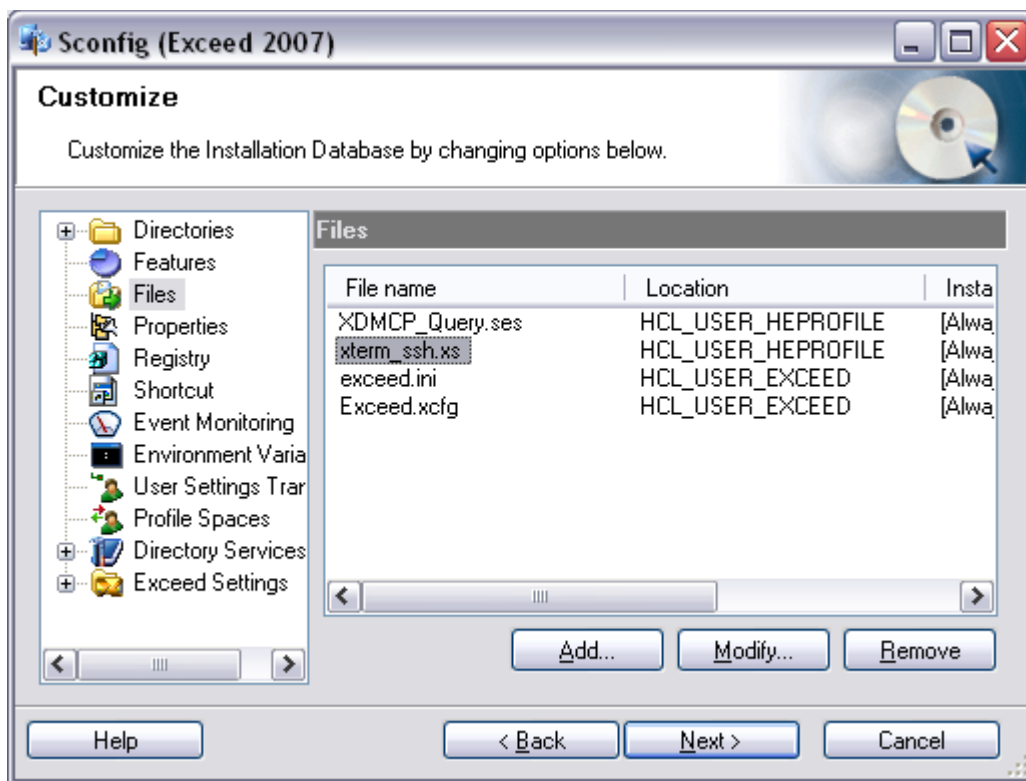


Startprofile hingegen (d.h. Dateien *.xs, *.ses, *.hep und *.hfs) kommen – seit Version 2007 – in das Verzeichnis „HCL_USER_HEPROFILE“. Dadurch werden sie auch in der Hummingbird Neighborhood zugänglich.

Eben dies ist für Xstart- und Xsession-Profile (dh. Dateien *.xs und *.ses) erst seit Version 2007 möglich, weswegen sich der Default-Pfad gegenüber Vorgängerversionen geändert hat.



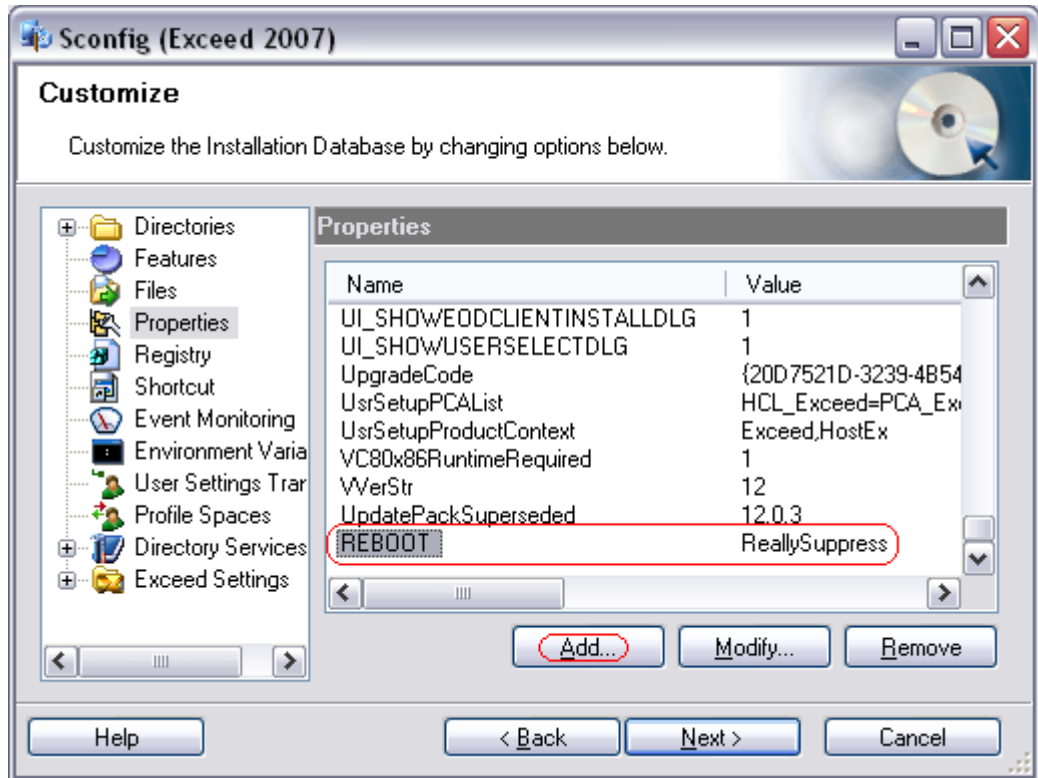
Am Ende listet Sconfig sämtliche ins Installationspaket integrierten Dateien:



So können auch nachträglich alle Veränderungen jederzeit bequem gesichtet – und bei Bedarf weiter bearbeitet werden. Um die Abweichungen vom ursprünglichen MSI-Paket vollständig zu erfassen, ist der Inhalt des Verzeichnisses CustomFiles und die MST-Datei ausreichend. Zwecks Übersichtlichkeit empfiehlt es sich, auch die verwendeten MSI- und MST-Dateien mit in CustomFiles aufzunehmen. Die MSI-Datei (in unserem Beispiel „Exceed 2007.msi“) gibt nämlich auch Auskunft über den eingespielten UpdatePack. Schließlich kann die mit Sconfig erstellte Transfrom-Datei (*.mst) nur mit genau der MSI-Datei funktionieren, mit der sie eben erstellt wurde. Ein nachträgliches Umbenennen der MSI-Datei hat fatale Folgen!

Interne Properties

Die meisten hier gelisteten „Properties“ haben eher informativen Charakter und können gar nicht geändert werden bzw. Änderungen sind selten sinnvoll.

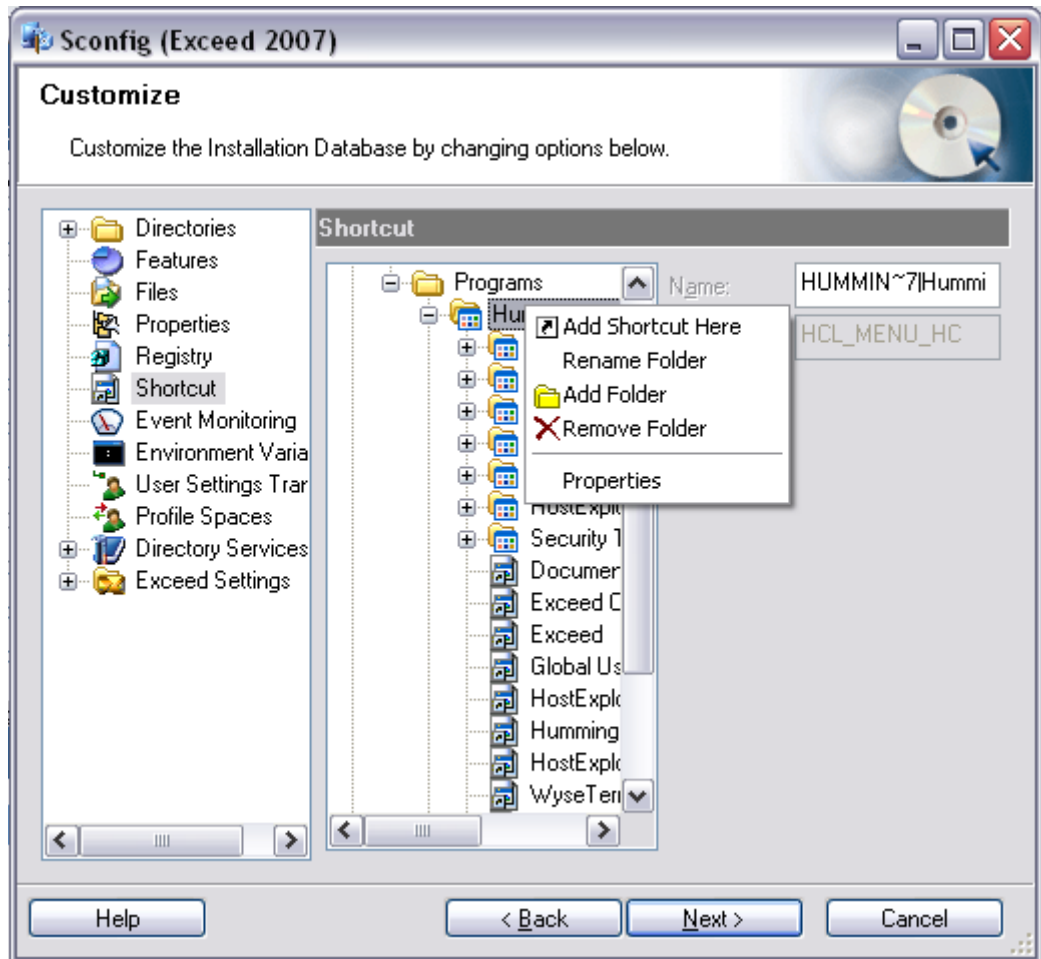


Über „Add“ lassen sich noch weitere Properties hinzu-

fügen. Da Reboots in der Regel vom Software-Verteilsystem selbst ausgeführt werden, ist es ratsam, hier die Property „REBOOT“ auszuwählen und auf „ReallySuppress“ zu setzen. Nur einfaches „Suppress“ hat bei vollautomatischer Installation nicht den gewünschten Effekt (es ist vielmehr für eine Abfrage bei interaktiver Installation vorgesehen).

Verknüpfungen („Shortcuts“) in Start->Programme

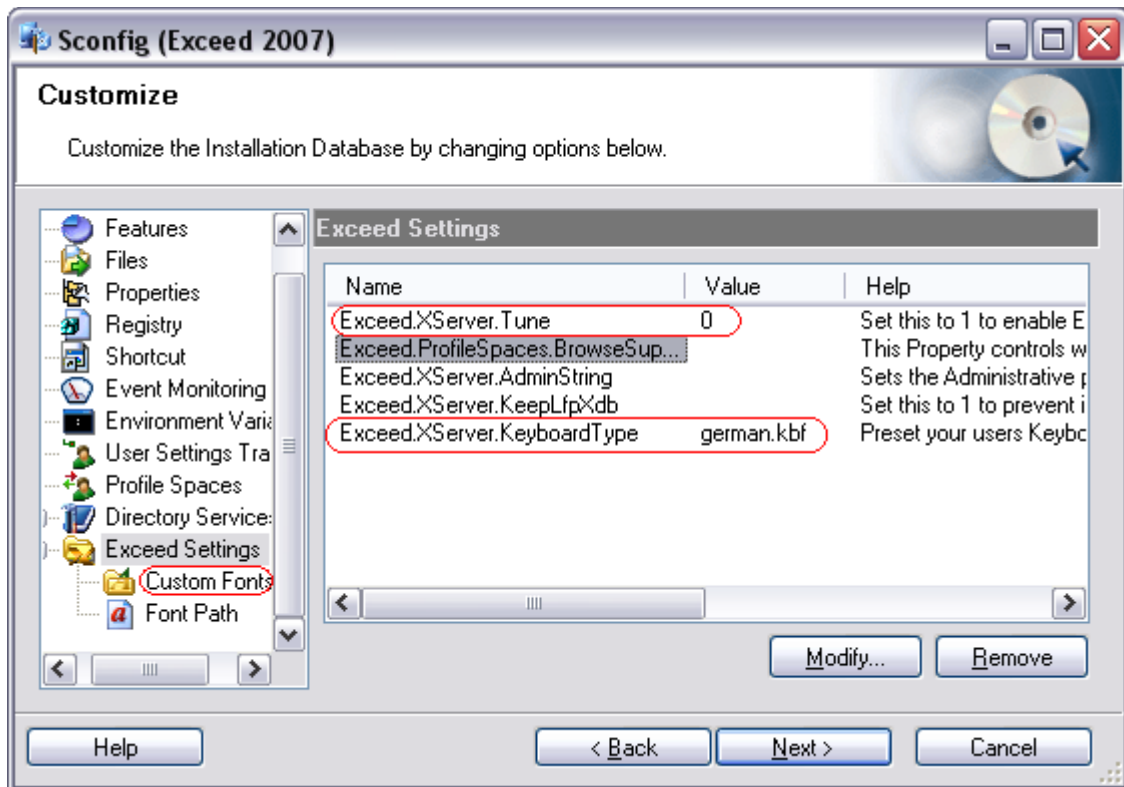
Hier lässt sich differenziert steuern, welche Verknüpfungen für Exceed-Komponenten wo angelegt werden.



Es ist durchaus möglich, Exceed-Komponenten zu installieren aber nicht als Verknüpfung erscheinen zu lassen. Ebenso kann man Ordner umbenennen, neue anlegen oder ganz verschwinden lassen.

Übrigens werden nur Verknüpfungen zu Komponenten angelegt, die auch wirklich installiert werden (d.h. unter der Sconfig-Rubrik „Features“ ausgewählt wurden). Man soll sich also nicht davon irritieren lassen, dass hier Verknüpfungen für sämtliche Komponenten erscheinen.

Exceed-Spezifisches



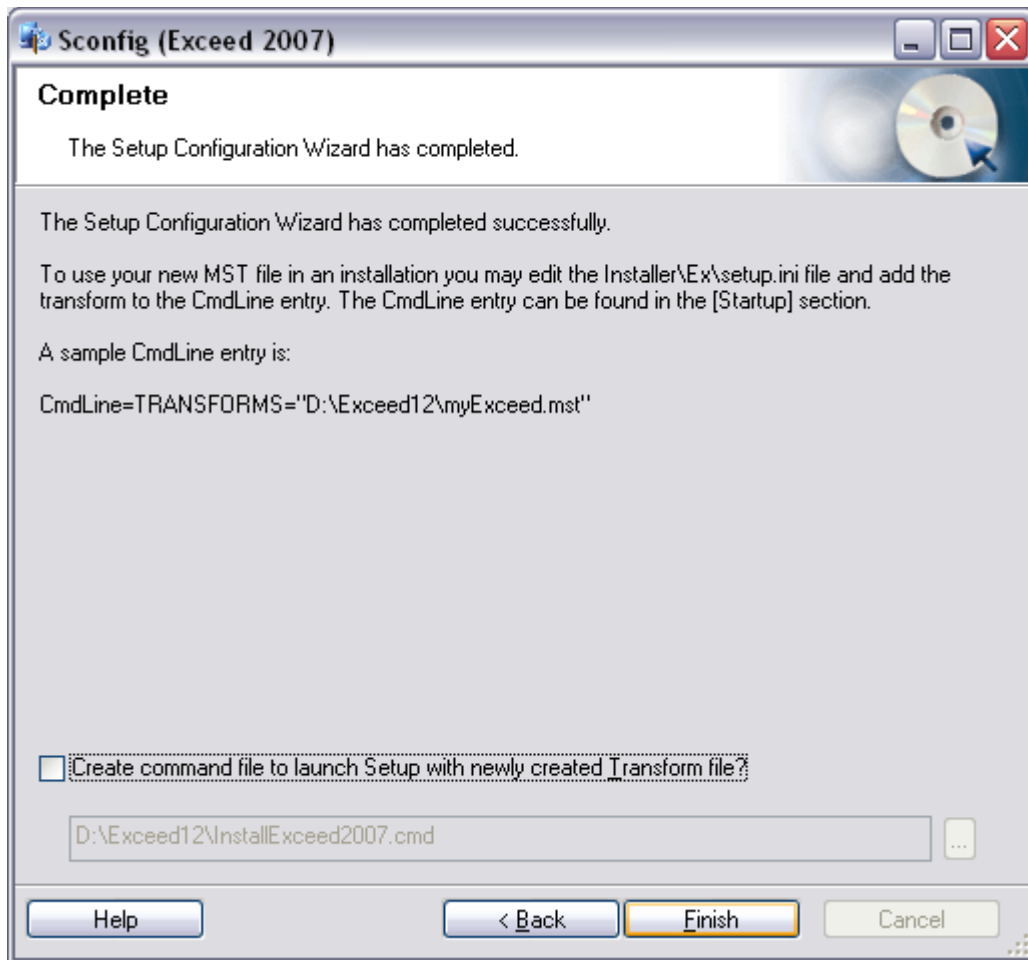
Die Property „Tune“ veranlasst ein Performance-Tuning des X-Servers während der Installation. Dabei öffnet sich ein kleines Fenster, in dem unterschiedliche Zeichenmethoden mit einer Reihe von Mustern und Bitmaps getestet werden. Dies läuft einer automatisierten Installation eher zuwider und sollte deshalb auf 0 gestellt werden.

Der Keyboard-Typ sollte hier schon richtig gewählt werden, sonst erhalten alle Anwender nach der Installation eine US-Tastaturbelegung in Ihrer Exceed-Standardkonfiguration.

Mit dem Punkt „Custom Fonts“ ist es möglich, eigene zusätzliche Schriftarten („Fonts“) in das Installationspaket zu integrieren. Dies ist dann erforderlich, wenn bestimmte X-Applikationen Fonts verwenden, die nicht im üblichen Bestand der X11-Fonts sind. Diese werden Linux- bzw. UNIX-seitig meist erst mit diesen Applikationen installiert.

Detaillierte Erläuterungen zum Thema Font-Import in Exceed sowie die Alternative Font-Server sind auf <http://x11.gweb.info/win/> zu finden.

Sconfig-Abschluss



Mit Klick auf „Finish“ wird die *.mst-Datei, in unserem Beispiel myExceed.mst, abgespeichert. Anstelle des hier angebotenen „command file“, das letztlich einen Wrapper für msixec.exe (das Kommandozeilen-Tool des Windows-Installers) verwendet, empfiehlt es sich eher, den Windows-Installer direkt aufzurufen, z.B. so:

```
@echo off
rem skript zur Installation von Exceed 2007
rem
msiexec.exe /I „Exced 2007.msi“ TRANSFORMS=myExceed.mst /L*v install.log /QB
```

Vorteile dieser Variante sind die Pfadunabhängigkeit des Aufrufs von msiexec.exe sowie der einfachere Zugriff auf die zahlreichen Parameter des Windows-Installers. In diesem Beispiel wird eine extrem detaillierte Log-Datei (install.log) über den gesamten Installationsvorgang geschrieben und mit /QB zwar sämtliche interaktiven Abfragen unterdrückt, jedoch noch ein Fortschrittsbalken angezeigt. Eine völlig lautlose Installation erreicht man mit purem /Q (Q steht für „quiet“).

Tipp: Software-Verteilssysteme, die mit sog. Snapshots arbeiten (z.B. NetInstall), sollten genau diesen Vorgang für Ihre Paketierung „mitschreiben“.

Anhang: Parameter des Windows-Installers

Das Verhalten des Windows-Installers lässt sich mit zahlreichen Parametern auf der Kommandozeile von `msiexec.exe` genau steuern, so dass man das obige Skript noch differenzierter gestalten kann. Zur Information ist hier die engl. Dokumentation von Microsoft (<http://msdn2.microsoft.com/en-us/library/aa367988.aspx>) wiedergegeben:

Command-Line Options

The executable program that interprets packages and installs products is `Msiexec.exe`. Note that `Msiexec` also sets an error level on return that corresponds to [system error codes](#). Command-line options are case-insensitive.

The command-line options in the following table are available with Windows Installer 3.0 and earlier versions. The [Standard Installer Command-Line Options](#) are also available beginning with Windows Installer 3.0.

Option	Parameters	Meaning
/I	<i>Package</i> <i>ProductCode</i>	Installs or configures a product.
		Repairs a product. This option ignores any property values entered on the command line. The default argument list for this option is 'omus.' This option shares the same argument list as the REINSTALLMODE property.
		p - Reinstalls only if file is missing.
		o - Reinstalls if file is missing or an older version is installed.
		e - Reinstalls if file is missing or an equal or older version is installed.
		d - Reinstalls if file is missing or a different version is installed.
/f	[p o e d c a u m s v] <i>Package</i> <i>ProductCode</i>	c - Reinstalls if file is missing or the stored checksum does not match the calculated value. Only repairs files that have msidbFileAttributesChecksum in the Attributes column of the File table.
		a - Forces all files to be reinstalled.
		u - Rewrites all required user-specific registry entries.
		m - Rewrites all required computer-specific registry entries.
		s - Overwrites all existing shortcuts.
		v - Runs from source and re-caches the local package. Do not use the v reinstall option for the first installation of an application or feature.
/a	<i>Package</i>	Administrative installation option. Installs a product on the network.
/x	<i>Package</i> <i>ProductCode</i>	Uninstalls a product.

[u|m]Package

or

Advertises a product. This option ignores any property values entered on the command line.

[u|m]Package/tTransform List

u - Advertises to the current user.

m - Advertises to all users of machine.

or

g - Language identifier.

[u|m]Package/gLanguageID

t - Applies transform to advertised package.

/L

[i|w|e|a|r|u|c|m|o|p|v|x|+|!|*] Logfile

Writes logging information into a logfile at the specified existing path. The path to the logfile location must already exist. The installer does not create the directory structure for the logfile. Flags indicate which information to log. If no flags are specified, the default is 'iwearmo.'

i - Status messages.

w - Nonfatal warnings.

e - All error messages.

a - Start up of actions.

r - Action-specific records.

u - User requests.

c - Initial UI parameters.

m - Out-of-memory or fatal exit information.

o - Out-of-disk-space messages.

p - Terminal properties.

v - Verbose output.

x - Extra debugging information.

Windows Installer 2.0: Not supported. The x option is available with Windows Installer version 3.0.3790.2180 and later.

+ - Append to existing file.

! - Flush each line to the log.

"*" - Wildcard, log all information except for the v and x options. To include the v and x options, specify "/!*vx".

Note For more information about all the methods that are available for setting the logging mode, see [Normal Logging](#) in the [Windows Installer Logging](#) section

Generates an SMS status .mif file. Must be used with either the install (-i), remove (-x), administrative installation (-a), or reinstall (-f) options. The ISMIF32.DLL is installed as part of SMS and must be on the path.

The fields of the status mif file are filled with the following information:

Manufacturer - [Author](#)

Product - [Revision Number](#)

Version - [Subject](#)

/m *filename*

Locale - [Template](#)

Serial Number - not set

Installation - set by ISMIF32.DLL to "DateTime"

InstallStatus - "Success" or "Failed"

Description - Error messages in the following order: 1) Error messages generated by installer. 2) Resource from Msi.dll if installation could not commence or user exit. 3) System error message file. 4) Formatted message: "Installer error %i", where %i is error returned from Msi.dll.

/p *PatchPackage[;patchPackage2...]*

Applies a patch. To apply a patch to an installed administrative image you must combine the following options:

/p <PatchPackage>[;patchPackage2...] /a <Package>

/q n|b|r|f

Sets [user interface level](#).

q , qn - No UI

qb - [Basic UI](#). Use qb! to hide the **Cancel** button.

qr - [Reduced UI](#) with no modal dialog box displayed at the end of the installation.

qf - *Full UI* and any authored [FatalError](#), [UserExit](#), or [Exit](#) modal dialog boxes at the end.

qn+ - No UI except for a modal dialog box displayed at the end.

qb+ - Basic UI with a modal dialog box displayed at the end. The modal box is not displayed if the user cancels the installation. Use qb+! or qb!+ to hide the **Cancel** button.

qb- - Basic UI with no modal dialog boxes. Please note that /qb+- is not a supported UI level. Use qb-! or qb!- to hide the **Cancel** button.

Note that the ! option is available with Windows Installer 2.0 and works only with basic UI. It is not valid with full UI.

/? or **/h**

Displays copyright information for Windows Installer.

Calls the system function **DllRegisterServer** to self-register modules passed in on the command line. Specify the full path to the DLL. For example, for MY_FILE.DLL in the current folder you can use:

/y *module*

msiexec /y .\MY_FILE.DLL

This option is only used for registry information that cannot be added using the registry tables of the .msi file.

Calls the system function **DllUnRegisterServer** to unregister modules passed in on the command line. Specify the full path to the DLL. For example, for MY_FILE.DLL in the current folder you can use:

/z *module*

msiexec /z .\MY_FILE.DLL

This option is only used for registry information that cannot be removed using the registry tables of the .msi file.

/c

Advertises a new instance of the product. Must be used in conjunction with /t. Available starting with the Windows Installer version that is shipped with Windows Server 2003 and Windows XP SP1.

/n *ProductCode*

Specifies a particular instance of the product. Used to identify an instance installed using the multiple instance support through a product code changing transforms. Available starting with the Windows Installer version shipped with Windows Server 2003 and Windows XP SP1.

The options /i, /x, /f[p|o|e|d|c|a|u|m|s|v], /j[u|m], /a, /p, /y and /z should not be used together. The one exception to this rule is that patching an [administrative installation](#) requires using both /p and /a. The options /t, /c and /g should only be used with /j. The options /l and /q can be used with /i, /x, /f[p|o|e|d|c|a|u|m|s|v], /j[u|m], /a, and /p. The option /n can be used with /i, /f, /x and /p.

To install a product from A:\Example.msi, install the product as follows:

```
msiexec /i A:\Example.msi
```

Only [public properties](#) can be modified using the command line. All property names on the command line are interpreted as uppercase but the value retains case sensitivity. If you enter **MyProperty** at a command line, the installer overrides the value of MYPROPERTY and not the value of **MyProperty** in the Property table. For more information, see [About Properties](#).

To install a product with PROPERTY set to VALUE, use the following syntax on the command line. You can put the property anywhere except between an option and its argument.

Correct syntax:

```
msiexec /i A:\Example.msi PROPERTY=VALUE
```

Incorrect syntax:

```
msiexec /i PROPERTY=VALUE A:\Example.msi
```

Property values that are literal strings must be enclosed in quotation marks. Include any white spaces in the string between the marks.

```
msiexec /i A:\Example.msi PROPERTY="Embedded White Space"
```

To clear a public property by using the command line, set its value to an empty string.

```
msiexec /i A:\Example.msi PROPERTY=""
```

For sections of text set apart by literal quotation marks, enclose the section with a second pair of quotation marks.

```
msiexec /i A:\Example.msi PROPERTY="Embedded ""Quotes"" White Space"
```

The following example shows a complicated command line.

```
msiexec /i testdb.msi INSTALLLEVEL=3 /!* msi.log COMPANYNAME="Acme  
""Widgets"" and ""Gizmos.""
```

The following example shows advertisement options. Note that switches are not case-sensitive.

```
msiexec /JM msisample.msi /T transform.mst /LIME logfile.txt
```

The following example shows you how to install a new instance of a product to be advertised. This product is authored to support multiple instance transforms.

```
msiexec /JM msisample.msi /T :instance1.mst;customization.mst /c /LIME logfile.txt
```

The following example shows how to patch an instance of a product that is installed using multiple instance transforms.

```
msiexec /p msipatch.msp;msipatch2.msp /n {00000001-0002-0000-0000-624474736554} /qb
```

When you apply patches to a specific product, the /i and /p options cannot be specified together in a command line. In this case, you can apply patches to a product as follows.

```
msiexec /i A:\Example.msi PATCH=msipatch.msp;msipatch2.msp /qb
```

The **PATCH** property cannot be set in a command line, when /p option is used. If the **PATCH** property is set when the /p option is used, the value of **PATCH** property is ignored and overwritten.