Search the User Guide...

Msiexec.exe Command Line

The Windows Installer technology uses **Msiexec.exe** for installing MSI and MSP packages. This tool gives you full control over the installation process, allowing you to set:

- install options (install, uninstall, administrative install, advertise a product)
- display options (full, basic or no UI during the installation)
- restart options (if the machine will be restarted after the installation)
- logging options
- update options (apply or remove updates)
- repair options (only for an installed package)
- public properties which are used by the installation

The usual form of the msiexec command line is this:

```
msiexec.exe <install_option> <path_to_package> [package_parameters]
```

Install Options

When launching an installation package, you can set the install type through these options:

```
msiexec.exe [/i][/a][/j{u|m|/g|/t}][/x] <path_to_package>
```

/i - normal installation
/a - administrative install
/j - advertise the product
  u - advertise to the current user
  m - advertise to all users
  /g - the language identifier used by the advertised package
  /t - apply transform to advertise package
/x - uninstall the package

Sample command line:

```
msiexec.exe /i "C:\Example.msi"
```

Display Options

The user interface level of the installation can be configured according to the target environment. For example, a package distributed to clients should have a full UI, while a package deployed through Group Policy should have no user interface. Msiexec.exe sets the UI level of the installation through these options:

```
msiexec.exe /i <path_to_package> [/quiet][/passive][/q{n|b|r|f}]
```

/quiet - quiet mode (there is no user interaction)
/passive - unattended mode (the installation shows only a progress bar)
/q - set the UI level:
  n - no UI
  n+ - no UI except for a modal dialog box displayed at the end.
  b - basic UI
  b+ - 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.
  b- - 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.
  r - reduced UI

  f - full UI

Sample command line:

```
msiexec.exe /i "C:\Example.msi" /qn
```

Restart Options

Sometimes an installation overwrites files which are in use or needs to reboot the machine in order to finish it. The reboot policy used by the installation can be set through these options:

```
msiexec.exe /i <path_to_package> [/norestart][/promptrestart][/forcerestart]
```

/norestart - the machine will not be restarted after the installation is complete
/promptrestart - the user will be prompted if a reboot is required
/forcerestart - the machine will be restarted after the installation is complete

Sample command line:

```
msiexec.exe /i "C:\Example.msi" /norestart
```

Logging Options

When debugging an installation package you can use multiple logging parameters in order to create a log. This log will contain different information for each parameter you use:

```
msiexec.exe [/i][/x] <path_to_package> [/L{i|w|e|a|r|u|c|m|o|p|v|x+|!|*}][/log]
<path_to_log>
```

/L - enable logging
  i - include status messages
  w - include non-fatal warnings
  e - include all error messages
  a - mention when an action is started

  - **r** - include action-specific records
  - **u** - include user requests
  - **c** - include the initial UI parameters
  - **m** - include out-of-memory or fatal exit information
  - **o** - include out-of-disk-space messages
  - **p** - include terminal properties
  - **v** - verbose output
  - **x** - include extra debugging information
  - **+** - append to an existing log file
  - **!** - flush each line to the log
  - ***** - log all information, except for **v** and **x** options
- **/log** - the equivalent of **/l***

Sample command line:

```
msiexec.exe /i "C:\Example.msi" /L*V "C:\package.log"
```

Update Options

The Windows Installer command line can apply or remove updates (patches for example) through these options:

```
msiexec.exe [/p][/update][/uninstall[/package<product_code_of_package>]]
<path_to_package>
```

- **/p** - install an MSP patch. When installing a patch silently, you need to set REINSTALLMODE property to "ecmus" and REINSTALL to "ALL". Otherwise the patch will simply update the MSI cached on the target machine.
- **/update** - apply updates (if there are multiple updates, you can separate them through the "**;**" character).
- **/uninstall** - remove an update for a product (if there are multiple updates, you can separate them through the "**;**" character)
  - **/package** - specifies the package for which the update is removed.

Sample command lines:

```
msiexec.exe /p "C:\MyPatch.msp"
msiexec.exe /p "C:\MyPatch.msp" /qb REINSTALLMODE="ecmus" REINSTALL="ALL"
msiexec.exe /update "C:\MyPatch.msp"
msiexec.exe /uninstall {1BCBF52C-CD1B-454D-AEF7-852F73967318}
  /package {AAD3D77A-7476-469F-ADF4-04424124E91D}
```

In the above command line the first GUID is the Patch identifier GUID and the second one is the Product Code of the MSI for which the patch was applied.

Repair Options

If you have an installed package, you can use the Windows Installer command line for repairing it:

```
msiexec.exe [/f{p|o|e|d|c|a|u|m|s|v}] <product_code>
```

- **/f** - repair a package
  - **p** - repair only if a file is missing
  - **o** - repair if a file is missing or an older version is installed
  - **e** - repair if file is missing or an equal or older version is installed
  - **d** - repair if a file is missing or a different version is installed
  - **c** - repair if a file is missing or the checksum does not match the calculated value
  - **a** - forces all files to be reinstalled
  - **u** - repair all the required user-specific registry entries
  - **m** - repair all the required computer-specific registry entries
  - **s** - repair all existing shortcuts
  - **v** - run from source and recache the local package

Sample command line:

```
msiexec.exe /fa {AAD3D77A-7476-469F-ADF4-04424124E91D}
```

In the above command line the GUID is the Product Code of the MSI which will be repaired.

Set public properties

The name of a public property contains only uppercase letters (for example PROPERTY). This type of properties can be set through the command line like this: **PROPERTY="value"**.

Sample command line:

```
msiexec.exe /i "C:\Example.msi" MY_PROP="myValue"
```