MSI Packaging & Repackaging FAQ's

Author: Mayank Johri (funmayank@yahoo.co.in)

Dated: Tuesday, September 28, 2010

Disclaimer

The information in this document is based on publicly available documentations and author's personal & professional experience. In no event shall author be liable for any direct, indirect, consequential, punitive, special or incidental damages (including, without limitation, damages for loss of profits, business interruption or loss of information) arising out of the use or inability to use this document, even if author has been advised of the possibility of such damages. Author makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to this document at any time without notice. Author does not make any commitment to update the information contained in this document.

Title	FAQ on MSI packaging & re-packaging
Author	Mayank Johri
Email	funmayank@yahoo.co.in
Version	0.0.1 Alpha 4
Document Date	Tuesday, September 28, 2010

Special Thanks

As you guys all know I am below average when it comes to MSI technology, thus I have no doubt when saying that this document would not have even started if not the generous help, guidance and support I got from: Aalok, Amrita, Karthik, Uttam, Shssael, Shirisha (hope this time I have spelled it right), Shrijeet, Sushma, Sunil, Wei Wu, Tony, Roshan, and many more.

Basic MSI Questions

1. What is Application Packaging?

Process of creating an installer for an application is called application packaging. Usually in it binary files provided by developer's are packaged to form a package.

The **Main difference** between **packaging** and **re-packaging** is that the source files does not come in the form of package in packaging where as in re-packaging they come in a form of a package, which might be MSI or legacy package (such as executable, batch files, etc)

2. What is Application Repackaging?

Repackaging (Customized Installation) is the process of capturing the changes made by an Installation Program (Package) and it is customized to support company standards and distribution methods.

It is not necessary to do setup capture to call it repackaging, even creating mst files or iss files can be called application repackaging.

3. What are the steps of Repackaging?

- Review the packaging requirements (User Requirement Review)
- Analyze the vendor package (Tech Review)
- Repackage the application (Setup capture)
- Customize the package (Scripting)
- Test the package (Testing & UAT)
- Release the package to end users (Deployment)

4. Why repackaging is required & what are the problems in Legacy Installation?

Most common reasons for Re-packaging are as follows

- Customized Installation
- Reduce Support Costs
- Self Repair
- Source Resiliency
- Unattended Install

The problems with Legacy Installations.

- High Support Costs
- Fragile Installs & uninstalls
- Difficult & labor Intensive to deploy

5. Name few MSI Packaging tools?

- Wise for Windows Installer
- Wise Package Studio

- Install Shield
- SMS Installer
- Marimba
- WIX (It cannot be used for re-packaging)
- Sharpdevelop
- Visual Studio

6. Name few MSI Re-Packaging tools

- Wise Package Studio
- Install Shield
- SMS Installer
- Marimba

7. Name few Deployment tools?

- Radia
- CA DSM
- Altiris Client Management Suite
- Altiris Notification Server Console
- Microsoft Systems Management Server (SMS)

8. What is Windows Installer?

Windows Installer (previously known as Microsoft Installer) is a built-in Operating System service for Installing and Managing Applications. It provides a standard method for developing, customizing, installing and updating applications.

It is an engine for the installation, maintenance, and removal of software on modern Microsoft Windows systems. The installation information, and often the files themselves, are packaged in installation packages, loosely relational databases structured as OLE COM Structured Storages and commonly known as “MSI files”, from their default file extension. Windows Installer contains significant changes from its predecessor, Setup API. New features include a GUI framework and automatic generation of the uninstallation sequence.

It is positioned as an alternative to stand-alone executable installer frameworks such as older versions of InstallShield and Wise Package Studio (later versions of both supports Windows Installer) and NSIS.

9. Benefits of the Windows Installer?

- Advertising
- Installation on Demand
- Repair (Self-healing)
- Rollback (Transactional operations)
- Managed Shared Resources

- Installation in locked-down environments

10. MSI Installation Mechanism (Background Mechanism)?

Acquisition: The Installer first installs the feature and then progresses through the actions specified in the sequence tables of the installation database. These actions query the installation database and generate a script that gives a step-by-step procedure for performing the installation.

Execution: The installer passes the information to a process with elevated privileges and runs the script.

Rollback: If an installation is unsuccessful, the installer restores the original state of the computer. When the installer processes the installation script, it simultaneously generates a rollback script. In addition to the rollback script, the installer saves a copy of every file it deletes during the installation. These files are kept in a hidden, system directory. Once the installation is complete, the rollback script and the saved files are deleted.

11. What is a MSI?

MSI is “Microsoft Windows Installer”. It is an installation, in the form of a single file. It is actually a database that contains several tables (80+). Each of these tables contains instructions and set-up information.

12. Structure of MSI?

• Products	(Collection of Features)
• Features	(Collection of Components)
• Components	(Collection of files and Registries)

13. What is Product?

A single, installed, working program (or set of programs) is a product. A product is identified by a unique GUID (the ProductCode property). A product is not the same as a package: a single MSI package might install multiple different products. For example, an MSI might install French and English versions of a program, each of which is a different product.

14. What is Feature?

Features are buckets for Components. Windows Installer configuration commands operate only on Features (installing, advertising, Uninstalling). Self-healing, install-on-demand and user profile fix-up operate at the Feature level.

15. What is Component?

Components are collections of resources that are always installed or removed as a unit from a user's system. A resource can be a file, registry key, shortcut, or anything else that may be installed. Every component is assigned a unique component code GUID.

16. What is self-healing

When an MSI-based application is launched (by clicking on an advertised shortcut or file type association), Windows Installer checks the existence of key path items. If there is a mismatch between the current system state and the value specified in the MSI package (e.g., a key file or registry is missing), then the related feature is re-installed. This process is also known as self-healing or self-repair.

17. What is the difference between Self-Healing and Repair

Self Heal and Repair are two different concepts in Windows Installer which people many times consider to be the same thing however there is difference in these two.

Self Heal is triggered by advertised shortcuts, or other advertising information in the package which eventually Repairs the application.

When the application is launched by advertised shortcut, it checks for all the key paths of the Current Feature, if any of the key paths is missing it will launch Repair.

Note that if there are multiple features then it will not check the missing key paths of the other features, but only the feature of which the advertised shortcut is launched.

Repair of an MSI can be triggered by

- Repair button in Add/Remove programs
- Giving the command line msiexec /f{other option} {MSI name}
- Self Heal by advertised shortcut or other advertising information
- Active setup

Once the repair of the package is triggered, even with Self Heal, then the whole of the MSI is reinstalled. Then it does not see that only the feature which triggered the self heal should be repaired, but the whole MSI, by which I mean, all its features are reinstalled.

18. What is Registry, Tell the Structure & types of Registry?

The Registry is a single place for storing information about the Windows OS (Hardware & Software)

- Root Keys / Subtrees
- Subkeys
- Hives
- Entries

Types of Registry keys

- Machine-Specific (HKCR, HKLM, HKCC, HKU)
- User-Specific (HKCU, HKU)

Types of Registry Root keys

- HKEY_CLASS_ROOT (HKCR)
- HKEY_LOCAL_MACHINE (HKLM)
- HKEY_CURRENT_CONFIG (HKCC)
- HKEY_CURRENT_USER (HKCU)
- HKEY_USERS (HKU)

19. What are Shortcuts & Types?

Shortcuts are the entry points to the applications installed on the system which is normally points to a file

- ***Advertised*** (File should be Installed by the Application)
- ***Non Advertised*** (File that is not part of Installation. It is also called Command Line shortcuts)

20. What are INI File & its format?

INI files are plain-text files that contain configuration information. "INI" stands for initialization.

```
[Section]
Keyname=value
```

21. What are Services & its types?

A windows service is a background process which is loaded by the Service Control Manager of the OS.

- ***Win32 Service*** (Win32 services are the services which is running by the executable file installed by the Application).
- ***System or Kernel Services*** (Kernel services are the services which are used by the OS to communicate to the hardware devices).

22. Where is Service information stored?

Most of the Service information are stored under the windows registry hive

“HKLM\System\CurrentControlSet\Name of the Service”

23. In the MSI, which tables contain information about the service details?

- ServiceInstall (Service Details)
- ServiceControl (Controlling the service during Installation & UnInstallation)

24. What is ODBC & DSN and its types?

ODBC means Open Database Connectivity. The purpose of ODBC is to allow the user to access data from any application. The layer between the application and the DBMS called DSN.

- System DSN (DSN will be available for all users)
- User DSN (DSN will be available for that particular user)

25. What is File Association?

The Windows operating system recognizes file types and associates them with programs based on their file extension. A file that carries no extension or no associated program is called Orphaned.

26. What is Environment Variable & its types?

Environment Variables are the variables that are set by the Operating System & Application.

- System Variable (Available for all users)
- User Variable (Available for that particular user)

27. What is Property & types of Properties, give some Examples?

Properties are global variables that the Microsoft Windows Installer uses during an installation.

Private: The installer can be use only internally (values can’t be changed during the run time).

Manufacture, ProductCode, ProductID, ProductName, ProductVersion ……

Public: The installer can be uses both internally & externally (values can be changed during the run time also).

INSTALLLEVEL, ….

Restricted Public: The user can’t change the value both internally & externally due to security purposes.

ALLUSERS, REBOOT, REINSTALLMODE …

28. What is Merge Module?

Merge modules are a mechanism in Windows Installer that allows companies to prepackage and share standard component definitions. Merge modules are used to deliver shared code, files, resources, registry entries and setup logic to applications as a single compound file.

29. Name few Merge Module tables?

ModuleSignature, ModuleComponents, ModuleDependency, ModuleExclusion, ModuleIgnore, ModuleSubstitution, ModuleAdminUISequence, ModuleAdminExecuteSequence, ModuleConfiguration, ModuleAdvtUISequence, ModuleAdvtExecuteSequence, ModuleInstallUISequence, ModuleInstallExecuteSequence,

30. Detail the Background mechanism of Merge Module?

If there are a number of applications that require a specifically configured component, it would be possible to create a merge module that installs and configures that component. That merge module could then be added to the installation packages of each product that required that particular component. This saves the effort of having to individually add the necessary files, registry entries, and other components to every installation. It also saves time if updates are needed, as instead of updating the installations for all five applications, only the merge module is updated, and the installations only need to be rebuilt.

31. How to give Permission for files, folders & Registry keys in MSI?

In the MSI, we can give permissions through Lock Permission table. But using subinacl.exe custom action is the best way to set permissions.

32. How to give Permission for files, folders & Registry keys through VB Script & what is the syntax?

We can give permission for files & folders through VB Script by using the **CACLS** & **XCACLS** commands. CACLs should only run on NTFS partitions.

CACLS – **C**hanges **A**ccess **C**ontrol **L**ist**S**

“Cacls <file name> [/T] [/E] [/C] [/G user: perm] [/R user [...]] [/P user: perm [...]] [/D user [...]] “

/T	Changes ACLs of specified files in the current directory and subdirectories
/E	Edit ACL instead of replacing it
/C	Continue (ignore) access denied errors
/G	user: perm where access rights granted can be: R C F (read, change, full control)
/R	user Revoke specified user's access rights (only valid with /E)
/P	user: perm Replace specified user's access rights. Permission can be: N R C F (none, read, change, full control)
/D	user Deny specified user access

E.g. ***“cacls c:\myfile.txt /E /G <user name>: F”***

33. What is VB Script?

Do it yourself

34. What is Wise Script?

Do it yourself

35. Tell some objects in the VB Script & when & why it is used in the Application?

Do it yourself

36. How to install only one particular feature during the Installation through Command line?

```
msiexec /i <msi> ADDLOCAL=<Feature Name>
```

37. How to disable ARP (Add/Remove Programs) Details during the Installation through Command line, tell some ARP properties?

```
msiexec /i <msi> ARPSYSTEMCOMPONENT=1
```

Following entries are various ARP properties

a) ARPAUTHORIZEDCDFPREFIX
b) ARPCOMMENTS
c) ARPCONTACT
d) ARPINSTALLLOCATION
e) ARPNOMODIFY
f) ARPNOREMOVE
g) ARPNOREPAIR
h) ARPPRODUCTICON
i) ARPREADME
j) ARPSIZE
k) ARPSYSTEMCOMPONENT
l) ARPURLINFOABOUT
m) ARPURLUPDATEINFO

38. What is Advertisement?

It means that, the Availability of an application to users or others with out actually the full Installation. There are two types of Advertising

- **Assigning:** An Application appears (shortcuts, files & registries) to a user or others, when an Application is "*assigned*". When the user tries to open, it is installed upon demand.
- **Publishing:** No Entry points appear to a user or others, when an Application "*published*" to the group. It is activated only if the group Application activates the published Application i.e. *Installation on Demand*.

39. What is Advertised Feature & Component?

If a Feature or Component is advertised, only the interfaces required for loading and launching the application are installed to the user or others. If a user activates an advertised interface the installer then proceeds to install the necessary Components & Features.

40. What is Installation on Demand?

When a user or application activates an advertised feature or product, the installer proceeds with installation of the needed components.

41. What is Transform?

A transform is a windows installer file with the extension (.MST). It should be used along with a

MSI to customize or change the installation package without modifying the MSI. The installer can only apply transforms during an installation.

42. What are the types of Transform?

a) Embedded transform

Embedded transforms are stored inside the .msi file of the package.

b) Secured transform

Secured transforms are stored locally on the user's computer in a location where, on a secure file system, the user does not have write access. Such transforms are cached in this location during the installation or advertisement of the package. During subsequent installation-on-demand or maintenance installations of the package, the installer uses the cached transforms.

c) Unsecured transform

Transforms that have not been secured are called unsecured transforms. To apply an unsecured transform, pass the transform file names in the *TRANSFORMS property* or command line string during the installation.

43. How to Create Transform in Wise / Install Shield?

In Wise Package Studio, by using Install Tailor or New Project → other templates → Transform

In Install Shield Admin Studio, by using the New Project → Transform

44. How many Transform can be created for one Vendor MSI?

There is no restriction on the numbers of transforms which can be created for one Vendor MSI

45. How many Transforms can be supplied in the Command line?

There is no restriction on the numbers of transforms which can be supplied in the command line

46. What is Custom Action?

The Microsoft Windows Installer provides many built-in actions for performing the installation process. For some cases the developer writes an action to execute his own installation is called custom action

47. What are the types of Custom Actions?

- DLL file stored in a Binary table stream
- DLL file that is installed with a product
- EXE file stored in a Binary table stream
- EXE file that is installed with a product
- Displays a specified error message and returns failure, terminating the installation
- EXE file having a path specified by a property value
- EXE file having a path referencing a directory
- JScript file stored in a Binary table stream
- JScript file that is installed with a product

- JScript text specified by a property value
- JScript text stored in this sequence table
- VBScript file stored in a Binary table stream
- VBScript file that is installed with a product
- VBScript text specified by a property value
- VBScript text stored in this sequence table
- Property set with formatted text
- Directory set with formatted text
- Installation of a package nested inside of the first package. See Nested Installation Actions
- Installation of a package that resides in the first application's source tree. See Nested Installation Actions
- Installation of an application that is advertised or already installed. See Nested Installation Actions

48. What are the types of Sequences in the Custom Actions?

- Normal User Interface
- Normal Execute Immediate / Deferred
- Administrative User Interface
- Administrative Execute Immediate / Deferred

49. What are the types of Conditions in the Custom Actions and what is the use?

- Not Installed - During Installation only
- REMOVE - During UnInstallation only
- NOT REMOVE - During both Install & UnInstall

50. What are the types of In Script options in the Custom Actions?

a) Immediate Execution

Immediate custom actions, can be sequenced anywhere within any of the sequence tables. It has access to the installation database (read & set installation properties, modify feature & component states, add temporary columns, rows, and tables).

b) Deferred Execution – User Context

Deferred custom actions can only be sequenced between the InstallInitialize and InstallFinalize actions in execute sequence tables. It doesn't have access to the installation database. Deferred custom actions are not executed immediately. Instead they are scheduled to run later during the execution script. The execution script isn't processed until the InstallExecute, InstallExecuteAgain, or InstallFinalize action is run.

If the Current User doesn't have the elevated privileges (Custom actions make changes in the system directly), the custom actions should run in Deferred Execution in User Context only.

c) Rollback only

This Action should be executed during the Installation of the Rollback script or if the Installation is Unsuccessful

d) Commit only

This Action should be executed during the Installation of the Commit script.

e) Deferred Execution - System Context

Deferred custom actions can only be sequenced between the InstallInitialize and InstallFinalize actions in execute sequence tables. It doesn't have access to the installation database. Deferred custom actions are not executed immediately. Instead they are scheduled to run later during the execution script. The execution script isn't processed until the InstallExecute, InstallExecuteAgain, or InstallFinalize action is run.

If the Current User have the elevated privileges (Custom actions make changes in the system directly), then it should run in Deferred Execution in System Context only.

51. What is the difference between "Immediate Execute / Deferred Execute"?

- ***Immediate*** custom actions, can be sequenced anywhere within any of the sequence tables
- ***Deferred*** custom actions can only be sequenced between the InstallInitialize and InstallFinalize actions in execute sequence tables
- ***Immediate*** custom actions have access to the Installation database
- ***Deferred*** custom actions doesn't have access to the Installation database
- ***Immediate*** custom actions can only run in the User Context
- ***Deferred*** custom actions can run both in the context of the user and elevated using the system context.

52. What is the difference between "Deferred in System Context / Deferred in User Context"?

If the Custom action which installs or modify a file under the INSTALLDIR or Installation should be run in ***"Deferred in User Context"***

If the Custom action which installs or modify the system file directly should be run in ***"Deferred Execution in System Context"***

53. What are the types of Processing Options in the Custom Actions and what is the use?

a) **Synchronous**: Windows Installer runs the custom action synchronously to the main installation. It waits for the custom action to complete successfully before continuing the main installation.

b) **Synchronous, ignore exit code**: Windows Installer runs the custom action synchronously to the main installation. It waits for the custom action to complete before continuing the main installation; the action can be either success or fail.

c) **Asynch, wait at end of sequence:** Windows Installer runs the custom action simultaneously with the main installation. At the end it waits for the exit code from the custom action before continuing.

d) **Asynch, no wait**: Windows Installer runs the custom action simultaneously with the main installation. It doesn't wait for completion of the custom action and doesn't check the exit code also.

54. What are the types of Scheduling Options in the Custom Actions and what is the use?

a) **Always Execute**: This action execute in all sequences

b) **Run first time**: This action execute only the first time Windows Installer encounters it.

c) **Run once per process**:This action execute only one time either Execute sequence that should not run if the installation is running in silent mode.

d) **Run only if UI sequence was run**: This action execute only if either Execute sequence is run following User Interface sequence.

55. What is Launch Condition?

Launch Condition is used to check system requirements on the destination computer

56. What is App Search?

App Search action is used to search for existing versions of products (Files, Registry, INI, Directory & Component)

57. What is Isolated Component, why we are using and its types?

It means that to prevent overwriting of previous versions of shared components, and ensures that other applications do not overwrite your version of shared components.

a) Manifest file concept
b) Local file concept

58. What is the use of MSI Assembly tables?

It is used for the registration of .Net Assembly files

59. What is the latest version of Windows Installer?

The latest version of Windows Installer is 4.5

60. What is the latest version of Wise Package studio & Install Shield Admin Studio?

Will change with time

61. What is the difference between Wise Package Studio & Install Shield Admin Studio?

Need to find the info myself.

62. What is Conflict Management?

When two or more applications install the same system files (DLLs, .VBXs, and .OCXs), Windows registry, and other items. To detect, Conflict Management should be use and for resolve the software conflicts, Application Isolation concept should be use

63. What are the types of Deployment (Software Distribution)?

- Group policy (Active Directory)
- Software Update Services (SUS)
- Windows Update Web site
- Systems Management service (SMS)

64. What is Software Distribution?

One of the more critical aspects to managing a Windows environment is the ability to deploy new applications, updates, upgrades & patches. Distributing new or updated software is called as Software Distribution.

65. What is Group Policy (GPO) & how to set it?

Administrators use Group Policy to define options for managing, configuration of servers, desktops, and groups of users. It is used to set policies across a given site, domain, or range of organizational units. Use “gpedit.msc” in the run command to set the policy.

66. What is Elevated User & how to create it?

If the user having the privileges of MSI features (Windows Installer) is called as Elevated User

You can create through “gpedit.msc” in the run Command or registry keys

```
HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\Installer
Type : DWORD Key: AlwaysInstallElevated   Value : 1
```

67. What is Wrapper MSI?

When an Executable is wrapped within an MSI then that msi is called wrapper msi

68. What is Lock down environment?

Software restriction policies provide administrators with a Policy-driven mechanism to identify software running on computers in a domain, and control its ability to execute. This policy can be used to block malicious scripts, help lockdown a computer, or prevent unwanted applications from running

69. What is IntelliMirror?

IntelliMirror management technologies is a set of powerful features for change and configuration management. It ensures that users’ data, software, and personal settings are available when they move from one computer to another, and persist when their computers are connected to the network.

70. What is Active Directory?

Deploying applications through the Active Directory is done through the use of group policies, and therefore applications are deployed either on a per user basis or on a per computer basis.

71. What are the other tools which are used during the testing & finding solving the Issues in the Application Packaging?

- Picture Taker
- Windows Install Master
- InstallRite
- RegMon
- FileMon
- procmon
- Process Explorer

- CsDiff
- Icon Extractor & Icon Builder
- RegExtractor
- WiLogUtl
- WiseComReg

72. What is Orca tool & what is the purpose and Advantages?

Microsoft utility to view, update, validate MSI & MST files.

73. Why multiple MSIExec is running during the Installation?

Windows Installer uses a client-server model for performing installations. Additionally for security reasons, Windows Installer hosts DLL and script custom actions in a "sandbox" process. Depending on how the install was initiated, one of the MSIExec processes can be the client process (Current User). Another MSIExec process is Windows Installer service (System).

74. Which drive your Application will Install, C drive has less space and D drive has more space? Why & how to solve?

It will install “D” drive only due to Windows Installer features. We can solve by adding “WindowsVolume” entry in the directory table as parent of “TARGETDIR” or you can use INDSTALLDRIVE=C:\

75. What is a Patch?

Patching is a streamlined process for updating earlier versions of a Windows Installer setup package i.e. when you update only files that already exist in your installation package. Only the package code is changed.

76. What is Upgrade?

Upgrade is a process of updating the earlier versions of a Windows Installer setup package i.e. Adding, changing & deleting new Files & Registries. But here product code, product version & package code should be changed.

77. What are the types of Upgrades and what is the difference?

a) ***Small Update***: A small update is a product update that changes a few files or possibly adds some new content. But there is a limitation for the changes that can be made to the feature-component structure for the package. Only the package code is changed. It is also called as a "Hotfix" or “Quick Fix Engineering (QFE)”.

b) ***Minor Upgrade:*** A minor update is a product update that makes enough changes. But there is a limitation for the changes that can be made to the feature-component structure for the package. The package code & product version is changed for the product. It is also called as a “Service Pack”.

c) ***Major Upgrade***: A major update is a product update with a large number of changes. There is no limitation for the changes that can be made to the feature-component structure for the package. The package code, product code & product version is changed for the product. It is also called as a "Product Upgrade”.

78. How you validate the MSI using Wise Package Studio?

In Wise Package Studio, by using “Package Validation” option by selecting the MSI with the default Cub file or Browse your own Cub file.

In Install Shield, by Build → Validate → Default Cub file or Browse your own Cub file.

79. What is ICE?

It means “Internal Consistency Evaluation”. ICEs are used to validate installation packages against various generic issues.

80. Give some ICE Error number and tell how to solve the ICE Errors?

There are totally 96 ICE Errors and 14 ICEM Errors in the Windows Installer. Some examples are as follows

a) ***ICE03 -*** Basic data and foreign key validation

b) ***ICE18 -*** Validates the KeyPath column of the Component table when it is NULL.

c) ***ICE21 -*** Validates that all components in the Component table map to a feature in the FeatureComponents table.

d) ***ICE33 -*** Checks for entries in the registry table that belong in other tables.

e) ***ICE38 -*** Validates that components installed under the user's profile use a registry key under HKCU as their key path.

f) ***ICE64 -*** Checks that new directories in the user profile are removed in roaming scenarios.

g) ***ICE57 -*** Validates that individual components do not mix per-machine and per-user data.

h) ***ICE59 -*** Checks that advertised shortcuts belong to components that are installed by the target feature of the shortcut.

81. What are the File types of Application Packaging?

•	**WSI**	-	Microsoft Windows Installer Project file
•	**ISM**	-	Install Shield Project file
•	**MSI**	-	Microsoft Windows Installer
•	**MST**	-	Microsoft Transform
•	**PCP**	-	Windows Installer Patch Project file
•	**MSP**	-	Microsoft Windows Installer Patch
•	**WSM**	-	Microsoft Windows Installer Merge Module Project file
•	**MSM**	-	Microsoft Windows Installer Merge Module
•	**EXE**	-	Executable file

82. How to install the .Exe file through silent mode?

Its mostly /s, but will depend up on which tool has been used to create it.

83. How to register the DLL manually?

By using the command line option “regsvr32”. For example

regsvr32 Dll name	For Register the Dll
regsvr32 /u	For Unregistered the Dll
regsvr32 /s	For Silent register

84. What is COM Component?

Component which is having the COM information

85. What are the default Windows Installer properties?

- [AdminToolsFolder]

  C:\Document and Settings\Current User\Start Menu\Programs\Administrative Tools\
- [AppDataFolder]
- C:\Document and Settings\Current User\Application Data\
- [CommonAppDataFolder]

  C:\Document and Settings\All Users\Application Data\
- [CommonFilesFolder]

  C:\Program Files\Common Files\
- [DesktopFolder]
- C:\Documents and Settings\Current User\Desktop\
- [FavoritesFolder]

  C:\Documents and Settings\Current User\Favorites\\
- [FontsFolder]

  C:\Windows\Fonts\
- [LocalAppDataFolder]

  C:\Documents and Settings\Current User\Local Settings\Application Data\
- [MyPicturesFolder]

  C:\Documents and Settings\Current User\My Documents\My Pictures\
- [NetHoodFolder]

  C:\Documents and Settings\Current User\NetHood\
- [PersonalFolder]

  C:\Documents and Settings\Current User\My Documents\
- [PrintHoodFolder]

  C:\Documents and Settings\Current User\PrintHood\
- [ProfilesFolder]
- C:\Documents and Settings\Current User\

- [ProgramFilesFolder]

  C:\Program Files\
- [ProgramMenuFolder]

  C:\Documents and Settings\Current User\Start Menu\Programs\
- [RecentFolder]

  C:\Documents and Settings\Current User\Recent\
- [SendToFolder]

  C:\Documents and Settings\Current User\SendTo\
- [StartMenuFolder]

  C:\Documents and Settings\Current User\Start Menu\
- [StartupFolder]

  C:\Documents and Settings\Current User\Start Menu\Programs\Startup\
- [System16Folder]

  C:\Windows\System\
- [SystemFolder]
- C:\Windows\System32\
- [TempFolder]

  C:\Documents and Settings\Current User\Local Settings\Temp\
- [TemplateFolder]

  C:\Documents and Settings\Current User\Templates\
- [WindowsFolder]

  C:\Windows\
- [WindowsVolume]

  C:\

86. What are the default Environment settings in XP?

- ALLUSERSPROFILE

  C:\Documents and Settings\All Users
- ***APPDATA***

  C:\Documents and Settings\current User\Application Data
- CommonProgramFiles

  C:\Program Files\Common Files
- COMPUTERNAME

System Name

- ComSpec

  C:\Windows\system32\cmd.exe

- HOMEDRIVE

  C:

- HOMEPATH

  C:\Documents and Settings\Current User

- ProgramFiles

  C:\Program Files

- SystemDrive

  C:

- SystemRoot

  C:\Windows

- TEMP

  C:\Documents and Settings\Local Settings\Temp

- TMP

  C:\Documents and Settings\Local Settings\Temp

- USERNAME

  Current User

- USERPROFILE

  C:\Documents and Settings\Current User

- Windir

  C:\Windows

87. What is Package Code?

The Package Code is a GUID identifying a particular Microsoft Windows Installer package. It associates an .MSI file.

88. What is Product Code?

The Product Code is a GUID identifying a particular Application or product.

89. Tell some complex Applications you did & what is the Issue and how you solve the Issue?

Provide few of your complex applications such as Office, etc

90. Maximum how many files you can add in msi package

91. Explain the difference between Property and PROPERTY

92. What are the differences between 3.1 & 4.5 version of MSI technology?

93. What are the disadvantage/drawbacks of MSI?

Resiliency: Resiliency can be inconsistent with repackaged applications because the repackager utility may not fully understand the component dependencies or what the key paths of the application should be. Therefore, an application may be packaged into one large feature that gets entirely reinstalled if a component keypath is missing. If it were broken up into multiple smaller features it would enable a more manageable resiliency.
COM/ActiveX Registration: Component Object Model (COM) and ActiveX controls may not be properly registered. Prior to Windows Installer, COM and ActiveX registration was a black box. Except for the exported functions DLLRegisterServer and DLLUnregister server, COM and ActiveX controls offered very few hints of their registration process. RegSvr32.exe was responsible for calling the previously mentioned functions and then the DLL was responsible for registering itself. There is no utility that can view a DLL, an OCX, or an EXE and figure out what goes on inside DllRegisterServer and DllUnregisterServer for that file. There are standard registry entries that most COM and ActiveX controls register, such as HKCR\CLSID, HKCR\ProgID, and HKCR\TypeLib. Information on COM registration may or may not get entered into the appropriate MSI tables by the repackager
Shortcuts: Shortcuts may not be created as Windows Installer descriptor shortcuts, which enable resiliency. Legacy setup shortcuts were .lnk files that pointed to an executable in most cases. Sometimes when the repackager runs, all it knows is that an .lnk file was copied to a directory. For example, a legacy Setup.exe installed a shortcut to C:\Windows\Profiles\User1\Desktop. The repackager would copy the .lnk file directly to the directory listed previously. Therefore, the repackager is not actually copying a Windows Installer shortcut, but rather it is copying a file without any resiliency capabilities included
Isolated Components: The only way to take advantage of isolated components is to author a new MSI package. Repackagers currently do not support this feature.
Application Removal:When uninstalling a repackaged application, it is possible that the AllUsers profile may be removed. This is dependent on how the legacy setup was captured and definitely needs to be tested.

94. How to do admin install using non-admin account?

95. What is advertisement and Command for Advertisement?

It means that, the Availability of an application to users or others with out actually the full Installation. There are two types of Advertising

Assigning: An Application appears (shortcuts, files & registries) to a user or others, when an Application is “assigned”. When the user tries to open, it is installed upon demand.

Publishing: No Entry points appear to a user or others, when an Application “published” to the group. It is activated only if the group Application activates the published Application i.e. Installation on Demand.

Syntax

```
msiexec /j [{u|m}] package
msiexec {u|m} package /t TransformList
msiexec {u|m} package /g LanguageID
```

Parameters

/j : Advertises a product.
u : Advertises to the current user.
m : Advertises to all users of the computer.

package : Specifies the Windows Installer package file.

/g LanguageID : Identifies the language.

/t TransformList : Applies transform to advertised package.

96. What is admin install and Command for it?

The Windows Installer can perform an administrative installation of an application or product to a network for use by a workgroup. An administrative installation installs a source image of the application onto the network that is similar to a source image on a CD-ROM. Users in a workgroup who have access to this administrative image can then install the product from this source. A user must first install the product from the network to run the application. The user can choose to run-from-source when he installs and the installer uses most of the product's file directly from the network.

Administrators can run an administrative installation from the command line by using the /a command line option. The ADMIN action is the top-level action used to initiate an administrative installation. When this action is executed the installer calls the actions in the AdminExecuteSequence and AdminUISequence tables to perform the administrative installation.

Syntax: Msiexec /a package

97. How to find the entry for corrupt components in a MSI Package?

98. How to update the MSI install log through a VB Script custom action?

99. For which type of packages admin install fail?

Any MSI which installs files based on selection, be it from command line, System Search or selected options while installing.

100. What is Transaction processing?

One or more operations processed together as a single indivisible whole called a transaction. All the constituent operations must succeed for the transaction to succeed, otherwise all the operations are rolled back to the original state.

Windows Installer 4.5 includes support for installing multiple packages using transaction processing. The packages are chained together and processed as a single transaction. If one or more of the packages in the transaction cannot be installed successfully or if the end user cancels the installation, the Windows Installer initiates rollback for all of the packages to restore the system to its earlier state.

101. How to apply multiple transforms to MSI?

msiexec /i <msifile> transforms=”mst1.mst”, “mst2.mst”, “mst3.mst” /q

102. A patch was applied on a machine which had the old version of software installed. The patch was successfully installed; when system was restarted it was found few files are missing, what could be the reason?

103. Can you create an MSI without any commercial utility?

Using VBScript, Python or any other language which support MSISDK

104. How do you troubleshoot an MSI installation?

105. MSI installation is failing with 1603, what might be the cause?

The following is a non-exhaustive list of known causes for this error:

- Short file name creation is disabled on the target machine.
- An Install Script custom action is prototyped incorrectly.
- A file is locked and cannot be overwritten.
- The Microsoft Windows Installer Service is not installed correctly.
- The Windows Temp folders are full.
- The setup was corrupted after installation and, therefore, fails with this error during un-installation.
- An older version of Install Shield Developer is being used.
- A general error occurred during the installation.
- Print and File sharing is not installed or enabled when installing MSDE 2000.

For more details http://www.symantec.com/connect/articles/understanding-error-1603-fatal-error-during-installation

106. What is the recommended way of handling COM objects?

107. How can we find whether source files contain MSI or not?

- If Msiexec engine process runs more than 2 times in taskmanager
- If package keeps entry of MSI in temp folder

108. How MSI installation be tested or How do you test the created MSI or What all do you test in Vendor MSI or inhouse MSI?

109. How can In-use file be deleted?

110. What is DLL HELL and how to avoid it?

111. How to delete HKCU keys for all users at un-installation of an MSI application?

Create a custom action which will traverse entire HKEY_USERS and then remove required keys for all the users.

112. What is the proper method of uninstalling/removing a service?

113. I am trying to remove a service using MSI and that service is now reporting as “Marked for deletion” instead of getting removed. Can you guess the reason?

The service might be running or services.msc console might be open

114. How to create, delete, start, stop service using command prompt?

115. What are the various ways to generate MSI installation log?

116. How to view the installation log files using any third party tool?

117. Can I change the product code of MSI?

118. What are the rules of Components?

119. What is MSISDK? How you ever used it?

Wise Package Studio

120. What are the types of Setup Captures in Wise / Install shield?

In Wise Package Studio there are three types

- **Virtual Capture**: Creates a clean virtual OS on your computer, and the installation is redirected in the clean virtual OS.
- **SmartMonitor**: Watches the installation and records the changes the installation performs.
- **Snapshot**: Scan the computer before and after the installation and record the differences between the first scan and the second.

In Install Shield there are two types

- **Installation Monitor**: Repackager watches lower-level system activities and records related changes made to the system by the setup(s) programs
- **Snapshot**: Scan the computer before and after the installation and record the differences between the first scan and the second.

MISC Questions

121. What is Dll Cache folder?

It is the folder in which Microsoft Windows stores protected system files.

122. What is MSOCache folder?

Really Really Tricky Questions[1]

(*answers will be provided in release Ver: 0.0.2*)

123. I am at my home with a desktop and no internet connection. My hacker friend came and gave me an MSI file. I wants to find everything about this msi file without actually installing the msi as I do not trust my friend when it comes for some practical joke. My bad luck is that I do not have an MSI editors with me such as ORCA, wise package studio or InstallShield? What are my options

124. Situation is same as above questions, List all reasons why I will not even try “Administrative Install”

125. How can you provide temporary admin access to non-admins in your corporate using MSI technology? Your company is using CA-DSM & SCCM for package deployment

[1] These are very easy for guys with experience in MSI but not for freshers in msi technology.