FAQ's: MSI Packaging & Repackaging

Author: Mayank Johri (funmayank@yahoo.co.in)
Dated: 28 March 2011

Disclaimer

The information in this document is based on publicly available documents 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	0.0.2
Document Date	28 March 2011

!!! Special Thanks !!!

As you all know I am way 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 for the generous help, guidance and support I got from: **Roshan, Aalok, Amrita, Karthik, Uttam, Tony, Shssael, Durga, Shirisha, Sreejith, Sushma, Sunil, Balsaraj, Satish, Wei Wu** and many more.

My Babbling

I know that still many questions are unanswered, I am working on them and will get them finished in due course of time. Thanks for understanding. :)

I have shamelessly ripped quite a few Q & A's from **Balsaraj blog**. Thanks **Balsaraj** for creating and maintaining such an informative & wonderful blog.

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 developers are packaged to form a package.

The **major difference** between **packaging** and **re-packaging** is that in packaging the source files does not come in the form of package whereas 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
- Self Repair
- Unattended Install
- Reduce Support Costs
- Source Resiliency

The problems with Legacy Installations.

- High Support Costs
- Difficult & labor Intensive deployment & management
- Fragile Installs & un-installs

5. Name few MSI Packaging tools?

- Wise for Windows Installer
- Install Shield
- Marimba
- SharpDevelop
- Wise Package Studio
- SMS Installer
- WIX
- Visual Studio

6. Name few MSI Re-Packaging tools

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

- Orca

7. Name few Deployment tools?

- Radia
- CA DSM
- Altiris Client Management Suite
- Altiris Notification Server Console
- System_Center_Configuration_Manager (SCCM)

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 Storage 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 uninstall 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 options} {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. i.e. all its features are reinstalled.

18. What are Shortcuts & Types?

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

- ***Advertised***
  - File should be Installed by the Application

  - It should be a key path of a component
  - No other file can be designated as the key member in that component
- ***Non Advertised***
  - File do not need to be part of Installation
  - They are also called Command Line shortcuts

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

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

20. 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 …

21. 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.

22. Name few Merge Module tables?

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

23. Explain 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.

24. 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.

25. 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"

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

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

27. 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

28. 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*.

29. 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.

30. 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.

31. 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.

32. 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.

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

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

In InstallShield Admin Studio, by using the New Project → Transform

34. 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

35. 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

36. 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

37. 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.
- Installation of a package that resides in the first application's source tree.
- Installation of an application that is advertised or already installed.

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

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

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

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

40. 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.

41. 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

42. 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"***

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

- **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.
- **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.
- **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.
- **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.

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

- **Always Execute**: This action execute in all sequences
- **Run first time**: This action execute only the first time Windows Installer encounters it.
- **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.

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

45. What is Launch Condition?

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

46. What is App Search?

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

47. 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

48. What is the use of MSI Assembly tables?

It is used for the registration of .Net Assembly files

49. What is the latest version of Windows Installer?

The latest version of Windows Installer is 4.5

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

Will change with time

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

Need to find the info myself.

52. 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

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

- Group policy (Active Directory)
- Software Update Services (SUS)
- Windows Update Web site
- Systems Management service (SCCM), CA DSM, BigFix

54. 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.

55. 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.

56. 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
```

57. What is Wrapper MSI?

When an executable Installer is wrapped within an MSI then that msi is called wrapper msi.

58. 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

59. 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.

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

- CsDiff
- FileMon
- Icon Extractor & Icon Builder
- InstallRite
- Picture Taker
- Process Explorer
- procmon
- RegExtractor
- RegMon
- WiLogUtl
- Windows Install Master
- WiseComReg

61. Why multiple instances of MSIExec.exe are 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).

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

It will install on “D” drive 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:\

63. 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.

64. What is ICE?

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

generic issues.

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

There are in total 96 ICE Errors and 14 ICEM Errors. Some examples are as follows

- ***ICE03 -*** Basic data and foreign key validation
- ***ICE18 -*** Validates the KeyPath column of the Component table when it is NULL.
- ***ICE21 -*** Validates that all components in the Component table map to a feature in the FeatureComponents table.
- ***ICE33 -*** Checks for entries in the registry table that belong in other tables.
- ***ICE38 -*** Validates that components installed under the user's profile use a registry key under HKCU as their key path.
- ***ICE64 -*** Checks that new directories in the user profile are removed in roaming scenarios.
- ***ICE57 -*** Validates that individual components do not mix per-machine and per-user data.
- ***ICE59 -*** Checks that advertised shortcuts belong to components that are installed by the target feature of the shortcut.

66. 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

67. 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.

68. What is COM Component?

Component which is having the COM information

69. 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\Template
[WindowsFolder]	C:\Windows
[WindowsVolume]	C:\

70. How to sign an MSI file.

In wise package studio, use the “Digital Signature” Option to sign the MSI. Or use signtool.exe from visual studio to sign it.

71. What is Package Code?

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

72. What is Product Code?

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

73. 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

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

Reference: Windows Installer Team blog
There are few limits that you may hit when authoring a large, complex MSI package.
Total number of files
If your Windows Installer package contains more than 32767 files, you must change the schema of the database to increase the limit of the following columns: the Sequence column of the File table, the LastSequence column of the Media table, and the Sequence column of the Patch table. Note that transforms and patches cannot be created between two packages with different column types.
Total number of components
The maximum number of rows for the Component table is 65536.
This limit was discovered by Danish Waheed and posted on the WiX users mailing list. It has been confirmed by a Microsoft employee but hasn't been officially documented yet.
Number of components per feature
There is a maximum limit of 1600 components per feature using Windows NT/Windows 2000 and a maximum limit of 800 components per feature using Windows 95 and Windows 98. There will be a ICE47 validation warning if your package is above this limit.
Depth of Feature tree
There is a maximum limit of 16 for the depth of the feature tree. If you exceed this limit you get a runtime error message "2701. The Component table exceeds the acceptable tree depth of 16 levels." which is a bit misleading because the limit actually applies to the Feature tree, not the Component tree (in my understanding there is no component tree in MSI). In a test with a feature tree 20 levels deep msiexec.exe even crashed instead of displaying an error message.
This limit has been reported by Danish Waheed on the WiX users mailing list. It doesn't seem to be officially documented.
Number of disks or CAB files
For each disk or CAB file one row must be added to the Media table. Windows Installer packages are limited to a maximum of 80 Media table entries when installed using Windows Installer prior to version 2.0. The restriction of 80 Media table entries was removed with Windows Installer version 2.0.
Limits of the CAB file format
The only compressed file type supported natively by Windows Installer is the Cabinet (CAB) format. The following limits apply to this file format.

- No one file in a CAB can exceed 2GB
- Maximum size of all files in one folder (compressed) 2GB
- Maximum size of a CAB file (compressed) 2GB
- Maximum number of files in a single CAB 64K

You can avoid these limits by splitting your setup into multiple CABs, possibly up to 64K * 2GB, or by placing your application files on the distribution disk uncompressed.

75. Explain the difference between Property and PROPERTY

Public Property: Public properties can be changed anytime by a user, system or administrators on the command line while installing, by applying a transform or by interacting with the authored user interface (Installation Interface). They are always in upper case

Private Property: The installer uses then internally and their values are initialized in the installation database (msi) or set by the values determined by the OS. They are always in lower case.

76. What are the disadvantage/drawbacks of MSI?

(Special Thanks to Sushma & Shirisha)

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.

77. 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.
```

78. 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
```

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

```
Sub addToLog (strLog)
        Dim logText
        Const msiMessageTypeInfo = &H04000000
        Set logText = Session.Installer.CreateRecord(2)          // No of lines to add to msi log
        logText.StringData(1) = strLog
        Session.Message msiMessageTypeInfo, logText
End Sub
```

80. On what conditions will admin install will fail to extract all the files?

The admin install will not wield proper results if MSI package installs files based on selection, be it from command line, System Search or selected options while installing. Or it installs few files through custom actions.

Also Admin installation will fail if no files are present or FILE table is empty.

It will also fail for packages which contain nested msi's.

It will also not extract any file if “**InstallFiles**” is missing from “Administrative Installation” Sequence

81. 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.

82. How to apply multiple transforms to MSI?

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

83. Can to create an MSI package without using any commercial application?

An MSI package can be created using VBScript, Python or any other language which supports MSISDK. Also WIX can be used to create MSI packages. ORCA can also be used.

84. How do you troubleshoot an MSI installation?

Two most common items to look for while troubleshooting an MSI installation are “MSI Installation Log file” & “Event Viewer”.

85. What is the best way to troubleshoot using installation log file

Following items are the best practices when troubleshooting using MSI log file

- Always read the file from the bottom up, as the error will have occurred nearer the end of the file.
- An MSI log is split into two categories; 'Properties', which are displayed at the end of the article and 'Actions'.

  The Action looks like:

  ```
  MSI (s) (18:B8) [12:40:51:187]: Doing action: CA_SETAPPFOLDER Action ended 12:41:23: RollbackGet-
  AppFolder. Return value 1.
  ```

  and Property looks like:

  ```
  Property(S): ALLUSERSPROFILE = C:\Documents and Settings\All Users\
  ```

- As Actions represent the flow of installation thus they should be first investigated and later Properties if needed.
- To determine which action has failed during the installation, search for the error generated during installation using keyword such as “ERROR” & return codes :)
- Check for the action have caused the error, also check the Action that was performed just before the error. A return value code is written in the log to show if the action completed successfully or not. One of the following will be displayed:
  - Return Value 1 – The Action completed successfully
  - Return Value 2 – The user terminated the action
  - Return Value 3 – The Action failed (will cause the installation to terminate)
- Make a note of which of the actions gives a return value of 3, and record any additional error information in the log file that may not have been displayed in the on-screen error.
- View the MSI and try to resolve the issue.

wilogutl.exe from MSISDK can be used to view the log file. It can analyze and reports issues also and can also save the logs in HTML format

86. How to check the any component state during install or uninstall

Log file is the best way to go about.

87. 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

88. 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

89. How MSI installation be tested or How do you test the created MSI or What all do you test in Vendor MSI or in-house MSI?

Report the method which you follow.

90. How to slipstream a patch to the original msi file

Using the following command patch file can be slip streamed on the original msi file.

```
msiexec /a <original .msi file> /p <.msp file with path>
```

91. How can In-use file be deleted?

Best option is to add a runonce entry in registry to delete the file. Its simple and does not need any third party utility. Or can use wise script to delete the file.

92. Where is the Windows Installer GUID stored in the Machine.

They are stored under “HKEY_LOCAL_MACHINE\SOFTWARE\Classes\Installer\Products” registry key hive.

93. 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.

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

ServiceControl table can be of your help, Set event to 8 which will delete the service. Or you can use sc as custom action.

95. I am trying to remove a service as a process of uninstalling an applicaiton. After uninstallation the service is 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.

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

There are many ways to generate the MSI installation log files, but the most common is throught the command line,

```
msiexec /i test.msi /qb+ /l*v logfile.txt
```

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

MSISDK comes with the best tool to view the installation log files.

98. Can I change the product code of MSI using MST file?

No, you **should not** change the product code or the package code of MSI using MST file

99. What are the rules for creating Components?

According to http://www.symantec.com/connect/blogs/microsoft-best-practices-component-rules-while-creating-packages website the best practices of creating the components are as follows

- Match components in previous versions of the .MSI
- Add all executable files to their own components
- Add all .TLB files to their own components
- Group Matching .HLP & .CNT and .CHM & .CHI files together
- Put registry keys associated with files or components in matching component
- Put Current User registry keys in their own component
- Put non-Current User registry keys in their own component
- Group all non-executable files to their own component
- Name new non-advertised shortcuts by destination directory
- Group non-keypath resources by resource type
- Create new components for resources not matching other criteria

100. How do you select key item in the components

Every component in an MSI requires one Keypath. This is one file or registry entry which is marked as the key element of the component. The Windows Installer Service seems to verify the availability of the keypath element on occasion. If it has changed it will be reinstalled from the msi. If you have components which install files in the Users Area (eg Application Data) make sure you set the keypath to a Registry Entry in the same component and not to the file. The user or an application could modify or remove the file which would trigger unwanted activity with the Installer.

101. What is the significance of REINSTALL property

The value of **REINSTALL** property should be the list of features delimited by commas that are to be reinstalled. The features name should be present in Feature table.

REINSTALL=ALL means that all the features which are installed should be reinstalled. It will not install the none installed features.

102. Where can I find the schematic of the msi file

In "**_Validation**" table of the MSI package

103. What is the best method of adding Network shortcuts in an msi

- Create a property eg. INSTDIR in Property table
- Populate it with the network directory
- Create a property eg SHORTCUT1 in Property table. (If not using Wise)
- Polulate it with the full UNC path along with the executable name (If not using Wise.)
- Create a shortcut using the shortcut under the shortcut table
- Update the WkDir coloum in Shorcut table with INSTDIR (and **no do NOT add brakets [] before and**

after INSTDIR).

104. How to reregister the MSI engine.

```
MSIExec /unregister
MSIExec /regserver
```

105. Name few advantages of using admin installation while deployment.

- Installs faster
- Patching is easy

106. What conditions be set to allow MSI installation engine to run a custom action during installation & repair but not at uninstallation

Self to answer

107. How to create a nested MSI installation.

Check out http://support.microsoft.com/kb/306439

108. How to register Fonts File

Use Fonts table.

109. In which table files from "Execute Program from installation" are stored

In BINARY table.

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

!!!!!! Need to find out

111. What will you do if you can't solve a issue in a package.

Tell something about contacting your seniors, forums, contacting application owner, vendor and Microsoft support.

112. Which type of applications should not be repackaged in MSI

Windows Patches, Service Pack, Internet Explorer, Anti Viruses, Applications which connect to their licensing servers to authenticate installation, etc,

113. Which & why files are stored in C:\Windows\Installer

When any msi, msp and its associated files (such as icons) is installed on the system, then a copy of its msi file is cached in this folder.

114. Name 5 Mandatory properties for an MSI Package

- **ProductCode** - Unique identifier for particular product release (GUID).
- **ProductVersion** - Version of the product. The format of this string is: major.minor.build. Eg: 10.07.110
- **ProductLanguage** - Numeric language identifier (LANGID) installer will use for any strings in the user interface that are not authored in the database. This value must be one of the languages listed in the Template Summary property in the Summary Information stream.
- **ProductName** - Name of the application.

- **Manufacturer** - Name of the manufacturer of the product.

115. How advertised Installation is different from a normal Installation

In advertised installation, the installation is carried out without actually installing any files and only when one of the entry point is executed then the files are actually installed. Also shortcuts and file type association is configured so that user can initiated the actual copying of files from the installation location.

116. What is "Source Resilience" and how does it work.

The Windows Installer provides **source resiliency** for features that are installed on-demand by using a source list. It contain the locations searched by the installer for installation. The entries can be network locations, Uniform Resource Locators (URLs), etc. If one of these sources fails then installer seamlessly try the next one.

Source Resiliency can be set during or after the time of initial deployment:

- **MST Files**: Add and populate the SOURCELIST property
- **Command line**: Populate the SOURCELIST property at the command line when an application is initially installed. The parameter must be passed in at the end of the command line, after the package file location.

```
msiexec.exe /I \\myserver\share\test_package.msi
SOURCELIST="\\DP1\share\package_source;\\DP2\share\package_source;"
```

- **Programmatically**: An application that is already installed can be modified through:
  - **Windows Installer APIs (MsiSourceListAddSource)** can add to the existing list of source paths. This also allows for management of more than the 26 source paths that can be specified at install time.
  - **Windows Installer Object Installer.AddSource.** The AddSource method of the Installer object adds a source to the list of valid network sources in the sourcelist.

117. How to avoid repetitive self healing

Check for the following in the package

1. files in %temp% folder
2. temp files such as .bak, .tmp etc
3. registry keys in "Software\Microsoft\Windows\Shell", "Software\Microsoft\Windows\ShellNoRoam" etc

if all the above fails then, check the event viewer to find which component is responsible for self healing and try to fix it.

118. what is REINSTALLMODE=vomus

Option	Details
p	Reinstall only if the file is missing.
o	Reinstall if the file is missing or is an older version.
e	Reinstall if the file is missing, or is an equal or older version.
d	Reinstall if the file is missing or a different version is present.
c	Verify the checksum values, and reinstall the file if they are missing or corrupt. This flag only repairs files that have msidbFileAttributesChecksum in the Attributes column of the File Table.
a	Force all files to be reinstalled.
u	Rewrite all required registry entries from the Registry Table that go to the HKEY_CURRENT_USER or

	HKEY_USERS registry hive.
m	Rewrite all required registry entries from the Registry Table that go to the HKEY_LOCAL_MACHINE or HKEY_CLASSES_ROOT registry hive
s	Reinstall all shortcuts and re-cache all icons overwriting any existing shortcuts and icons.
v	Use to run from the source package and re-cache the local package. Do not use the v reinstall option code for the first installation of an application or feature.

119. What is the difference between TARGETDIR and INSTALLDIR

TARGETDIR property specifies the root destination directory for the installation and during the administrative installation it specifies the location to copy the files. If TARGETDIR is not specified then ROOTDRIVE property is used as root folder of installation.

INSTALLDIR property is the default root location for all components.

120. What is the difference between ALLUSERS = 0,1,2

The ALLUSERS property configures the installation context of the package. The Windows Installer performs a per-user installation or per-machine installation depending on the access privileges of the user, whether elevated privileges are required to install the application, the value of the ALLUSERS property, the value of the MSIINSTALLPERUSER property and the version of the operating system.

ALLUSERS =0: the per-user installation context

ALLUSERS = 1: the per-machine installation context.

ALLUSERS = 2: enables the system to reset the value of **ALLUSERS**, and the installation context, dependent upon the user's privileges and the version of Windows.

121. What needs to be done to make sure that one particular file remains on the system even after uninstallation.

Add the file/registry to a newly created empty component and then in Details, select “Leave Installed on Uninstall” check box.

122. What needs to be done to make sure that if a file is already present on the system then it is not updated on installation.

Add the file/registry to a newly created empty component and then in Details, select “Never overwrite if key path exists” check box.

123. How to test the application in the locked down environment

On the test machine login as administrator and perform a advertised installation using the following command

```
msiexec /jm package_file.msi ALLUSERS=1 /qb+ /l*v C:\temp\install_packagename.log
```

The above command will install the advertised shortcuts, icons, file types & extensions, COM class registratio and Package installation details.

Logout and login as non-admin user and test the application.

124. Where should a Custom action be placed which used a property populated by “System Search”

The Custom Action should be placed after “CostFinalize” as during CostFinalize only the “System Search” is carried out.

Upgrade

125. What does “Another version of this product is already installed” means

It means that some version of the product is already installed on the system and msiexec is unable to perform an upgrade properly. This can be due to many reasons, such as they are same version, or upgrade code is missing or not properly set.

126. 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.

127. 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”.

128. How to determine if an upgrade was performed or a clean installation by looking in installation log file.

Check for “FindRelatedProducts” in log files and if it returns 1 then its an upgrade else fresh install. :)

129. How to make a custom action get executed only on Upgrade

Still trying to figure it out myself., :(even

130. What happens when the upgrade failes and RemoveExistingProducts is placed in between InstallExecute and InstallFinalize

If the removal of the old application fails then the installer rolls back both the removal of the old application and the install of the new application. As a result entire application is removed from the machine and not even the older version remains on the machine.

Launch Conditions

131. How to stop the installation if a specific file is not present on the machine

Following steps should be used.

- Add a custom property and set the value to “NULL”
- Add a system search for File, add the above property in property section, Operation is “Search all fixed drive for the file”, search depth 255, and last the filename
- Add a “Terminate Installation” Custom Action after CostFinalize with Condition Property=”NULL”

132. How to stop the installation if any older version of the product is already installed

NEED TO DO myself, :(

133. How to make sure that the package only gets installed on WinXP SP3

Launch condition with conditions VersionNT = "501" And ServicePackLevel = "3"

Wise Package Studio

134. 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.

135. How to find the entry for corrupt components in a MSI Package using wise package studio?

In the Tables Section, by pressing “F4”

136. You are asked to setup a new infrastructure for the packaging team with 10 packagers. They have 2 Virtual Machines & one physical machines each. They use VM's to capture the application and then use the physical machine to do the remaining task. You wants to make sure the maintenance of this setup will be easy and centralized. How should you setup the WPS to achieve this goal.

137. What is the name of default msi template in WPS

“Windows Application.wsi”

138. You have an old machine, which can just run one virtual machine but if you try to create any big package on it then the entire machines becomes unresponsive. You have WPS installed on the base machine as well as virtual machine. You have been asked to create a very big package, What are your options.

Use **VirtualOS** feature to create the package. :)

139. 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.

Wise Script / SMS Installer

140. How to display a message informing the users to wait for the duration while your wise script is processing a long running task

141. How to execute a process using Wise script in such a way that display of the process is hidden from view.

142. Can we add text to msi log using WiseScript

143. How to search for all instances of a file on the installed machine.

144. What is the difference between InstallFiles & CopyFiles

MSISDK

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

146. Can we create MST using MSISDK

yes,

147. How can table details be obtained using MSISDK but not using orca

msidb.exe will provide the details

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

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

149. How to validate an MSI package using ORCA

150. How to create an MST files from two versions of MSI files of the same product.

http://mayankjohri.wordpress.com/2010/02/19/tips-msi-save-differences-between-two-msi-as-mst-file-using-ruby-python/

Ruby Language:-

```
require 'win32ole'
msiClass = WIN32OLE.new("WindowsInstaller.Installer")
msidb = msiClass.OpenDatabase("C:\\temp\\MountPointGenerator.msi",0)
newmsidb = msiClass.OpenDatabase("C:\\temp\\updatedMountPointGenerator.msi",0)
transform = newmsidb.GenerateTransform(msidb,"c:\\temp\\newtest.mst")
```

151. How will you add the upgrade details in an newly authored MSI file using ORCA.

Add the details in Upgrade Table.

MISC Questions

152. What is Dll Cache folder?

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

153. What is MSOCache folder?

Local Install Source is a Setup feature that copies the install source files from the Microsoft Office installation media (for example, the Office 2003 CD-ROM) to the Msocache folder. This is a hidden folder on your local hard disk.

154. How to recover corrupted MSI engine?

155. When will msocache folder will be created

It can be created during the installation, if one of the available hard drives has more than 1.5 gigabytes (GB) of free disk space available and The hard disk with sufficient space is not a removable drive or a network drive.

156. Should we delete the msocache folder using Windows Explorer

No, msocache folder should never be deleted using Windows Explorer, instead **Disk Cleanup** program should always be used to remove the folder.

157. Where is Service information stored?

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

"HKLM\System\CurrentControlSet\Name of the Service"

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

sc.exe command can be used for these tasks.

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

160. 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)

161. What are Services & its types?

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

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

162. 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)

163. 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.

164. 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)

165. What is VB Script?

Do it yourself

166. What is Wise Script?

Do it yourself

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

Do it yourself

168. 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.

169. 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

170. What are INI File & its format?

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

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

171. 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

172. The repackaged application is working for admin users but for normal users it is failing. What are the things you will look for during the course of resolving the issue.

173. What is the difference between task, process and a service.

174. What is “C:\Windows\CSC” folder ?

The CSC (Client Side Cache) folder is where Windows keeps the information about offline folders.

This folder location can be changed to some other location by editing the registry key “DatabaseLocation” under the following node.

```
HKLM\Software\Microsoft\Windows\CurrentVersion\NetCache
```

Tips to clear this cache:

Set the following registry key and restart the computer.

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\CSC\Parameters]
“FormatDatabase”=dword:00000001
```

175. How to register Fonts using batch file

copy the fonts files in %windir%\fonts folder and then update the registry using **reg.exe** command, the keypath to update is “**HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Fonts**”.

MSI Version >= 4.5

An application capable of being installed, updated, run and removed by a standard (non admin) users without using elevated privileges (elevation) is called a PUA.

184. What is Single Package Authoring

The development of a dual-purpose Windows Installer 5.0 package for installation on Windows 7 and Windows Server 2008 R2 is referred to as single package authoring.

More details at http://msdn.microsoft.com/en-us/library/dd408068%28v=vs.85%29.aspx

Really Really Tricky Questions[1]

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

185. 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 don't trust my friend as he is a practical joker and I fear that he might want to try some at my expence. 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

The only option is to query the MSI file for various tables & its contents using any programming language such as VB Scrpt, Python etc. Always start from _Validation table which will tell you a general idea of the package.

DO NOT in any condition try Admin Install.

186. Situation is same as above questions, list all reasons why I will not even try "Administrative Install"

1. Admin Install Sequence can contain custom actions which you do not want.
2. Installation can contain conditioned features installing different versions of the same file under different conditions.

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

The easiest way is to create an advertised shortcut pointing to a file which can be deleted by non-admin users and then create a Custom Action to launch command prompt which should only run on certain conditions such as repair.

As the MSI was installed using a Central Deployment method, the application will get repaired using local admin account and the resulting command prompt will have admin access. :)

188. What file types can not be extracted using Admin Installation from a properly created MSI.

INI files can not be extracted as they are stored in INI table ;) .

189. In which table files are stored

None, they can either be outside of the msi file in the form of CAB file or as separate files. Internal CAB files are (or can be) stored inside the MSI database, which is a COM structured storage file; the files don't appear in any table.

Source: http://community.flexerasoftware.com/showthread.php?t=118689&page=2

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