Dell | Professional Services

Service Delivery Guide

Application Packaging- Testing Dell Factory

Document Revision History		
Version	Date	Author
1.0	09/07/2004	Bill Walton

Table of Contents

1.0 Introduction

This document is the Service Delivery Guide for Application Packaging Testing. It is intended to provide instruction and direction to the engineers and managers involved in the delivery of Application Packaging Testing services.

1.1 Target Audience

The target audience for this guide is the service delivery resource that will be delivering the packaging services. This includes:

1. Packaging Project Executives
2. Packaging Project Managers
3. Packaging Project Administrators
4. Packaging Team Leads
5. Senior Packaging Engineers
6. Packaging Engineers

1.2 Purpose

The purpose of this guide is to provide the step by step instructions for consistent delivery of the application packaging services. This document is focused on the Application Packaging Testing service itself. Other guides will provide detailed instructions for Application Packaging Assessment, Application Packaging Startup, Application Packaging Requirements Gathering, and Application Packaging.

1.3 Related Documents

There are several sets of related documents. The first set consists of marketing guides, program feature guides, and sales guides. The second set is the service delivery guides for the related services, including:

1. Dell Packaging – Service Delivery Guide – Application Packaging Assessment
2. Dell Packaging – Service Delivery Guide – Application Packaging Startup
3. Dell Packaging – Service Delivery Guide – Application Packaging Requirements Gathering
4. Dell Packaging – Service Delivery Guide – Application Packaging

1.4 Glossary

Application Integration - The process of packaging applications for deployment through ESD (Electronic Software Deployment).

Best Practices - Industry standard or recommended methods for managing and executing processes. DPS uses both industry-recommended and Microsoft-recommended best practices for application integration.

Billing Trigger - A step or task in the AI process that is marked as a billable event. Used to mark percent of completion in the AI process.

Build - Also known as an IMAGE. An engineered OS (Operating System) with the core applications installed for a business. A base build is the packaging pre-capture load. Also refers to version or revision of this software.

Change Request Form - A document describing a proposed change, the reason for the change, and the potential effects of the change on the project used to facilitate the Change Control Process.

Change Control Process - The process which governs all changes to the scope of the project throughout the project life cycle.

COTS - **C**ommercial **O**ff **T**he **S**helf - Software that is available in a set and standard version available to purchase without special configuration by the end user.

Dell Object Generator Tool - Software developed to automate the packaging process required for packaging objects. The tool is written to site standards and must be configured for different customers (standards and environments).

Dell QA Generator Tool - Software developed to automate the packaging process required for checking MSI packages. The MSI scripts and tables are read to check for standards and best practices compliance. The tool is written to site standard and must be configured for different customers (standards and environments).

Dell Template Generator - Software developed to automate the packaging process required for standardizing MSI package starting templates. The tools will populate the application information from the documentation into the MSI tables and script. The tool is written to site standard and must be configured for different customers (standards and environments).

Dell Wrapper Generator Tool - Software developed to automate the packaging process required for inserting MSI packages into the ESD process.

Deployment - The process for sending an application to the end user desktop.

DPS - **D**ell **P**rofessional **S**ervices.

Engagement Manager (EM) - DPS manager responsible for resolution of issues and conflicts.

Escalation - The process of pushing a problem or issue to the next higher position for resolution.

ESD - **E**lectronic **S**oftware **D**eployment - A method of distributing integrated applications to the end user through a network. The process is often automated to handle a large amount of applications to a group of users.

Factory Aging Report - A report generated to indicate the length of time work has been in the queue.

Factory Process - The process of converting applications for integration using an automated and structured process that is tracked for status and billing.

In-House Developed Software - Software that has been developed or modified by a developer specifically for use by the customer.

Localization - The process of optimizing an MSI package for regional, language, or other environmental standards.

Migration - The process of moving from one operating or hardware platform to another. Typically involves deployment of both and includes the user software.

MSI File - Micro**s**oft **I**nstall (file extension) - A Microsoft standard of an SDK compliant application, converted to a series of tables to comply with the Windows Installer executable process. A correctly developed MSI application can be installed (remote and local) and uninstalled without conflicting with other applications in the build.

MSM File - Micro**s**oft **M**erge (file extension) - A Windows Installer merge module common to multiple applications. Merge modules are pre-compiled libraries of component files (files, registry changes, and other system changes) that install a discrete portion of multiple applications. They cannot be run alone, but must be merged with a Windows Installer database (.MSI file)

MSP File - Micro**s**oft **P**atch (file extension) - A patch is a software configuration (in a series of tables) that will change or update the existing installed application configuration.

MST File - Micros**o**ft **T**ransform (file extension) - A transform is based on an MSI package and converts the package to a desired configuration. Transforms can silence install screens, include user or configuration information, and other environmental settings.

Operating System (OS) - The core software that controls the hardware to application interface.

Packaging - The process of creating or converting MSI, MSP, or MST files from applications.

Packaging Engineer (PE) - Engineer trained in multiple computer engineering disciplines, responsible for creating or converting MSI, MSP, or MST files from applications.

Peer Review - A packaging process where a separate engineer reviews the work of the originating packaging engineer for compliance (operability and standards compliance).

Peer Review Engineer - A Packaging Engineer performing the Peer Review process is referred to by this title.

Platform - The hardware with the core operating system (and base application build).

PMO - **P**roject **M**anagement **O**ffice - The group responsible for the overall management and governance of the project with specific tasks and reporting structures.

Prerequisites - Hardware, software, facilities, documentation, information, and dependencies that must be provided prior to the start of a process.

Productivity Software - Software such as Microsoft Office Suite, Lotus Notes, databases, or other applications used to publish, track, update, or document project activities.

Project Administrator (PA) - The person responsible for tracking the status of all requests made for changes to each application. The PA also generates reports including billing, aging, work queue, and others.

Project Executive (PE) - The person responsible for the overall governance of the project from beginning to end.

Project Manager (PM) - The person responsible for specific process management and delivery of the project. PMs typically report to the PE and have several Team Leads reporting to the PM.

Promotion - The term used to describe the process of inserting a completed and tested application into the ESD for operational use.

QA Testing - **Q**uality **A**ssurance Testing - Testing performed to ensure proper functionality and compliance with customer requirements.

Remediation - The process of reworking, re-developing, consolidating, or retiring applications that will not operate (or cannot be deployed) on the target platform.

Requirements - The information and dependencies needed prior to the start of the packaging process. Requirements include the name and version of the application, the source code, other dependent applications and hardware, configuration information, and other information critical to successful packaging of the application.

Requirements Gathering Engineer - A trained Engineer responsible for gathering and evaluating requirements for the packaging process.

Rework - Any application process steps that are required after the initial package completion. Rework can be for warranty or requirements change reasons. The work is typically performed by the original Packaging Engineer.

Sales Support - A role assumed by the PMO or Engineers during the initial site visits to determine the work scope and customer issues.

Scope - The detailing of the work that DPS will and will not perform for the customer. The scope is documented in the Scope of Work / Service Agreement document and agreed upon by both the customer and DPS during the Startup Phase of the project.

SDK - **S**oftware **D**evelopment **K**it - The Microsoft document that details the process and standards for integration into the Microsoft O/S and applications.

Senior Packaging Engineer (SPE) - The trained engineer position that reports to the Team Lead. The Senior Packager has additional responsibilities and rights above Packaging Engineers. This typically includes the ability to change the task of an application and network administrative rights. The Senior Packager mentors the Packaging Engineers and can act as a temporary Team Lead.

SMS - Systems **M**anagement **S**erver - Microsoft application used for application management, deployment, and reporting.

Standards - The methodologies and parameters used by DPS to package, test, and deploy applications.

Statement of Work (**SOW**) - A work scope document developed by DPS that describes the technical and contractual details of a project.

Status Code - The reportable task step used for metrics in the management system. The status code is used for loading, billing, reporting, and other metrics.

Team Lead (TL) - The engineer responsible for leading the Engineers. Reports to the Project Manager and is considered the escalation point for technical issues.

Test Script - A document or software used fro verifying the correct operation of software.

UAT - User **A**cceptance **T**esting - Testing performed primarily by the customer technical content expert to ensure the installed application (package) works as expected.

User Acceptance Testing Engineer - The engineer responsible for performing the test with the test script and SME.

Visual Aids - Whiteboards, wall charts, graphs, bulletin boards, and other display media used to display status, production or other information.

Warranty Rework - The process of performing a rework that is not billable to the customer. This does not include any process or steps requiring additional information not available when the original package was developed. Post installation and configuration changes are not warranty items unless included in the original documentation.

Work Item - A packaging request that has been submitted and entered into the DPS database. Application packages are referred to as Work Items in the DPS Work Queue.

Work Queue - The Work Items in the process.

WSI - Wise **S**ervice **I**nstaller (file extension) - The WISE (software company) version of the MSI file. It includes special features and additional table entries for use with WISE deployment (and Alteris). WSI files can be converted from and to MSI files.

2.0 Overview of the Dell Application Packaging Service Offer

This program covers the development and implementation of a suite of services for Application Migration Management. Application packaging is a process by which multiple software applications are packaged for efficient delivery via a network to individual desktops with little or no touch required by the end user. The Dell suite of services will include: Discovery and Project Startup, Application Inventory and Requirements Gathering, Application Packaging, Application Remediation, and Application Testing.

2.1 Objective

As much as 80% of the lifetime PC TCO is incurred after the hardware is installed. A large portion of these costs is spent on managing software. Application Packaging services helps customers load and manage multiple applications, on multiple desktops, across multiple locations.
Application Packaging gives customers the ability to automate software installs and customize software consistently and accurately across a complex environment of business systems within an enterprise. Consequently, it drives:

1. Reduced desktop TCO.
2. Increased stability and reliability of applications.
3. Reduced time to test and deploy software.
4. Reductions in help desk support incidents.

2.2 List of Inter-Related Services

2.2.1 Application Packaging Assessment.

This is the service that assesses the customer's current packaging environment, processes, workflows, and tools. This service makes recommendations on how to improve the packaging capability of the customer by changing the environment, processes, workflows, and tools to be consistent with industry best practices.

2.2.2 Application Packaging Startup.

This is the service that creates a new Application Packaging environment for the customer. This service defines and documents standards, documents and implements processes and workflow, implements the package tracking mechanism, develops project plans including staffing, budget, and productivity plans, and hires and trains the project staff. Once completed the customer will be able to run their own packaging factory, or Dell can provide the Application Packaging service for the customer.

2.2.3 Application Packaging Requirements Gathering.

This is the service that gathers the detailed requirements for all applications to be packaged. These requirements include the source media, the installation instructions, the configuration files, the verification scripts, and all other detailed technical dependencies needed to complete the packaging. A document is created that captures all of these requirements. This Requirements Document is forwarded to the Application Packaging Service.

2.2.4 Application Packaging.

This is the service that creates the application package, given the Requirements Document. The service receives requests to package, validates the software functions properly on the targeted OS, captures the package, conducts package conflict resolution, creates the deployment object, creates the package documentation, and transmits the package to the testing team.

2.2.5 Application Packaging Testing.

This is the service that tests the packages produced by the Application Packaging service. Two primary types of tests are conducted: Quality Assurance (QA) and User Acceptance (UAT). During QA testing the package and documentation are compared to company standards. All standards problems are documented and the package is returned to packaging. During UAT, the user tests the package to assure it meets user functional requirements. All functional problems are documented and the package is returned to packaging.

2.2.6 Application Packaging Turn-key.

This is a service where Dell commits to deliver a set number of applications at a set price. It includes Application Packaging Assessment, Application Packaging Startup, and Application Packaging. This service is sold at a fixed price.

2.3 High Level Process Steps

The Application Integration process consists of the following high level process steps:

1. The Project Administrator enters the work requests from the customer generating work items in the work queue.
2. Team Leads assign the work items in the work queue to the Packaging Engineers.
3. Packaging Engineers review the requirements to verify that the application information provided by the customer is sufficient to start the process.
4. Packaging Engineers validate the operating system on targeted test hardware to ensure that the application will install and run on the targeted customer supplied image.
5. Packaging Engineers capture, edit, and compile the MSI, MST, or MSP file.
6. Packaging Engineers generate application integration documentation.
7. Packaging Engineers perform standards and compliance checks.
8. Peer review is conducted by a second objective Packaging Engineer, on second targeted test hardware, including validation.
9. Packaging Engineers generate peer review documentation.
10. Packaging Engineers and Team leads generate comments and notes on the packaging process and any issues with the process.
11. Work items are updated and tracked by the Project Administrator.

3.0 Pre-Engagement Stage

3.1 Internal Preparation

Dell Professional Services performs in-depth analysis and studies customer environments in great detail to determine customer requirements prior to the execution of a contract. The essential process steps include the following:

1. Analysis of target environment to determine the exact statement of work.
2. Definition of a validated application list and remediation process.
3. Definition of standards for packaging, Quality Assurance, IT, and other standards.
4. Identification of processes for application deployment.
5. Identification and analysis of network and other environmental factors.
6. Publication of a scoping outline and pricing in a white paper for the customer.
7. Publication of a service agreement and statement of work after white paper feedback from customer.

3.2 Customer Preparation

Certain infrastructure and equipment concerns must be reconciled prior to the commencement of any DPS Application Packaging project. Responsibility for these prerequisites belongs to the customer. DPS will consult with the customer or provide equipment and software when agreed upon during the Assessment stage of the project.

1. Identification and provision of a stable and complete core build on which to package.
2. Definition and implementation of packaging and QA standards.
3. Definition of a consistent and stable set of packaging tools.
4. Definition and implementation of a packaging tracking system.
5. Definition and provision of licensed packaging software.
6. Provision of necessary server space.
7. Provision of packaging-specific hardware, PCs, and servers.
8. Provision of security access, office space, access to e-mail, phone, and network, standard supplies and office equipment for any on-site work including visits.
9. Identification of customer contact(s) that have authority to act for the customer in all aspects of the project.

4.0 Testing Service Delivery

The Testing Service Delivery process is broken down to Inputs, Processes, Outputs, and Tools Used to complete each step of the process in the following tables.

4.1 Review of Application Integration document

INPUTS	PROCESSES	OUTPUTS	TOOLS USED
Packaging standards	UAT Testing Engineer reviews the Application Integration documentation for any deviations from packaging standards.	**IF** Application Integration document **PASSES** review package passes to quality assurance standards review.	Productivity software
Packaging requirements: ● Application information ● Source files ● Installation instructions ● Configuration information ● Technical dependencies ● Test script	UAT Testing Engineer reviews the Application Integration documentation for content.	**IF** Application Integration document **FAILS** review package passes back to troubleshooting for analysis and corrective action.	Packaging requirements
Packaged application			Packaged application
UAT test script			UAT test script
Test hardware			Test hardware
Customer base build			Customer base build
Best practices			Best practices

4.2 Review of Quality Assurance for standards.

INPUTS	PROCESSES	OUTPUTS	TOOLS USED
Packaging standards	UAT Testing Engineer configures a test machine with clean base build.	**IF** the package **PASSES** QA review it passes to the customer UAT testing stage.	Productivity software
Packaging requirements: ● Application information ● Source files ● Installation instructions ● Configuration information ● Technical dependencies ● Test script	UAT Testing Engineer installs the package on the test system.	**IF** the package **FAILS** QA review it passes back to troubleshooting for analysis and corrective action.	Packaging standards
Packaged application	UAT Testing Engineer documents any installation issues.		Packaged application
UAT test script	UAT Testing Engineer documents any execution related issues.		UAT test script
Best practices	UAT Testing Engineer inspects the MSI file for any deviations from the packaging standards.		Best practices
	UAT Testing Engineer opens the Application Integration document and checks for any unexplained standards deviations.		

4.3 Set up of the test computers with operating system and base build developed by the customer.

INPUTS	PROCESSES	OUTPUTS	TOOLS USED
Packaging standards	UAT Testing Engineer sets up test computers for customer UAT Testing.	Test systems are validated and prepared for customer UAT Testing.	UAT test script
Packaging requirements: ● Application information ● Source files ● Installation instructions ● Configuration information ● Technical dependencies ● Test script			Test hardware
Packaged application			Customer base build
UAT test script			Best practices
Test hardware			
Customer base build			
Best practices			

4.4 Scheduling of customer users for User Acceptance Testing.

INPUTS	PROCESSES	OUTPUTS	TOOLS USED
Packaging standards	UAT Testing Engineer schedules customer users through customer technical contact for UAT testing.	Customer users are scheduled for UAT testing to validate the packaging of applications.	UAT test script
Packaging requirements: ● Application information ● Source files ● Installation instructions ● Configuration information ● Technical dependencies ● Test script			Test hardware
Packaged application			Customer base build
UAT test script			Customer user headcount and availability
Test hardware			Best practices
Customer base build			
Customer user headcount and availability			
Best practices			

4.5 UAT Testing Execution

INPUTS	PROCESSES	OUTPUTS	TOOLS USED
Packaging standards	UAT Testing Engineer sets up test computers with new clean builds for customer UAT Testing.	**IF** package **PASSES** UAT Testing is passes to promotion stage.	UAT test script
Packaging requirements: ● Application information ● Source files ● Installation instructions ● Configuration information ● Technical dependencies ● Test script	Customer technical contact expert and / or customer user runs UAT test script.	**IF** package **FAILS** UAT Testing it passes back to troubleshooting for rework.	Test hardware
Packaged application	Customer technical contact and / or customer user provides detailed reporting of all issues discovered during UAT Testing.		Customer base build
UAT test script	UAT Testing Engineer runs UAT test script and performs analysis of issues.		Customer user headcount and availability
Test hardware			Best practices
Customer base build			
Customer user headcount and availability			
Best practices			

4.6 Documentation of any issues during User Acceptance Testing including failures or requirements changes.

INPUTS	PROCESSES	OUTPUTS	TOOLS USED
Results of customer user UAT Testing	UAT Testing Engineers document any issues found during UAT including failures or requirements changes.	Issues found during UAT are documented and appropriate changes are implemented.	Productivity software
Packaging standards	Customer user UAT testers document any issues found during UAT including failures or requirements changes.	**IF** package **PASSES** UAT Testing is passes to promotion stage.	Packaging standards
Packaging requirements: ● Application information ● Source files ● Installation instructions ● Configuration information ● Technical dependencies ● Test script		**IF** package **FAILS** UAT Testing it passes back to troubleshooting for rework.	Packaging requirements
Packaged application			Packaged application
UAT test script			UAT test script
Best practices			Best practices

4.7 Rewrite of the test / installation scripts to be sent back to packaging for packaging or documentation updates.

INPUTS	PROCESSES	OUTPUTS	TOOLS USED
Results of customer user UAT Testing	Customer technical contact rewrites test scripts or installation scripts as required to include packaging or documentation updates.	DPS reviews updated documentation.	Productivity software
Packaging standards		Packaging Engineers make required changes during next pass through the packaging process.	Results of customer user UAT Testing
Packaging requirements: ● Application information ● Source files ● Installation instructions ● Configuration information ● Technical dependencies ● Test script			UAT test script
Packaged application			Best practices
UAT test script			
Best practices			

4.8 Promotion of the package to customer EDS

INPUTS	PROCESSES	OUTPUTS	TOOLS USED
Results of customer user UAT Testing	UAT Testing Engineer reviews UAT Testing documentation.	**IF** package **PASSES** UAT Testing and review it passes to the promotion stage.	Productivity software
Packaging standards	Customer technical contact reviews UAT Testing documentation.	**IF** package **FAILS** UAT Testing and review it passes back to the troubleshooting stage for corrective action.	Results of customer user UAT Testing
Packaging requirements: ● Application information ● Source files ● Installation instructions ● Configuration information ● Technical dependencies ● Test script			UAT test script
Packaged application			Best practices
UAT test script			
Best practices			

5.0 Appendix

5.1 Examples of Requirements

The table below contains examples of requirements information needed and corresponding examples of client responses. This information is required for each individual application to be packaged.

Contact Information			
PROJECT ROLE	**NAME**	**PHONE**	**E-MAIL**
Customer Business Requestor	Rockwell Banker	(713) 216-1000	rbanker@bank.com
Customer Business Owner	Ben Swamped	(713) 216-1001	bswamped@bank.com
Customer Technical Contact	Geek Codewell	(713) 216-1002	gcodewell@bank.com
Customer Application Tester (UAT)	Gern Blandston	(713) 216-1003	gblandston@bank.com
Vendor Product Contact	Roger Ramjet	(909) 555-5554	rramjet@microsoft.com
Vendor Technical Contact	Steve Dreamjob	(909) 555-5555	sdreamjob@microsoft.com
Customer Location	Houston TX		
Customer Address	601 Travis Suite 1640 Houston TX 77002		
Customer Line of Business	Banking / Financial Services		
Customer Division	Data Processing		
Application Information			
Application Name	Microsoft Word 2003		
Application Version	2003		
Application Developed In-House or COTS	COTS		
Application 16 bit, 32 bit or 64 bit	32 bit		
Application License	FJOI7F-FH89F-EU89F-BUE4C-MOI3S		
Application License Owner	Big Bank		
Application License Limitations	[Per User / Per Machine]		
Validated Operating System (Current)	MS Windows 95		

Target Operating System (Intended)	MS Windows XP Professional SP-2
Stand Alone Application?	[Category- LOB, Core Build App, Developer App]
Current Operating System Environment	NT, Novell, Linux
Special Run Requirements	[Environment Variables, Login Script, Drive Mappings, Languages Local Machine Services, Launch Conditions, Removal of Previous Versions]
Client Server Based Application?	[Special Domain Access, Read / Write Access, IP or UNC Path]
Terminal Application?	Constantly
Web Based Application?	[ActiveX Control, Self-Updating, Website, Added to Trusted Websites]
Application Data Source	Offsite- AC Nielsen Marketing Database
Application Data Owner	AC Nielsen
Application Data Location	ACNielsen.Data.Com
Application User ID	BigBank1
Application Password	-Classified-
Application Source Location	[Read / Write Access to share data files]
Application Software Dependencies	None
Dependent Software Location	N/A
Application Hardware Dependencies	Card Scanner and Coffee Cup Holder
Dependent Hardware Location	Locker that we can't find the keys for
Updates / Patches Intended for inclusion	Yes- via Microsoft Office Update website
Upgrades Timeline	Not until Word Longhorn in Q3 2008
Delivery Date	Must be completed by next Wednesday (!)
Additional Information	Any information that will aid the packaging process

Application Installation Instructions	1. Start/Run/Setup.exe 2. [Welcome] Click *Next* 3. [License Agreement] Click *Yes* 4. [License Information] Enter Information- Name: **UberTester1**. Company: **BBank**. Serial Number: **FJOI7F-FH89F-EU89F-BUE4C-MOI3S**. Click *Next* 5. [Destination Folder] Enter **C:\Program Files\Microsoft Office\Microsoft Word**. Click *Next*. 6. [Client/Server] Click *Client*. Click *Next*. 7. [Start Installation] Click *Next*. 8. [Finish] Click *Finish*.

5.2 Examples of Standards

5.2.1 Packaging Standards

In order to ensure a smooth migration of applications to a new operating system, many standards must be set in place to keep packages consistent across many different fronts. The more consistent packages are the lower the maintenance costs are in deploying and troubleshooting them in the production environment. The following are the three main classifications of applications for packaging, and their associated integration method.

10. Setup.exe installation format. Installation source is provided by running a setup routine in a non-MSI format. Wise SetupCapture and related tools are used to capture changes to the workstation.
11. Vendor Supplied MSIs. It is preferential to use Vendor supplied MSIs. This creates problems in some cases due to custom actions & poorly scripted MSIs. Using a snapshot of the MSI is a suitable option. The packaging engineer decides which option is the most suitable.
12. No Setup Routine Provided. When the application is a set of files to be placed in a single directory or an application is provided in a previous packaging technology format a new installation is created from within Wise or Wise SetupCapture and related tools are used to build the package.

5.2.2 Documentation

All Dell produced packages will include an application integration (AI) document with them. This document will serve as an install / uninstall / troubleshooting guide for testing and deployment purposes. The document will include the following main sections:

1. Special Requirements.

2. Contact Information.
3. General Application Information.
4. Install Instructions.
5. Verification Script.
6. Conflict Management Report / Results.
7. Exceptions.
8. Troubleshooting.

5.2.3 General Application Standards

If any of the standards in this lost are not followed, an exception should be well documented in the AI document of what standard was not followed and a detailed explanation of why it was not followed.

1. All distribution functionality should be provided by MSI and not the deployment mechanism.
2. Windows Installer or System variables should be used whenever possible.
3. No hard coded settings will be included in any package.
4. DNS must be used in all integrations.
5. No IP addresses will be hard coded.
6. The total character length of the Package Name must not exceed 45 characters.
7. Directory paths must not exceed 220 characters.
8. Suppress reboot on installation. If reboot is required it must be listed in the exceptions section of Application Document.
9. All packages will be packaged in thin format meaning install files reside outside of the MSI file.
10. All developer packages should be packaged to same standards as end-user packages excluding rights issues.
11. Include the Vendor and Rx version information in the MSI Product Name and Summary fields including the first R1 release.
12. VB Script is the preferred programming language to be used for most custom actions.
13. Compiled scripts or executables are not recommended due to them requiring original source and a separate utility installed to make changes to them.
14. All files for a given application must be installed under the C:\Program Files directory.
15. The directory name under C:\Program Files must match the Application Name syntax of the package itself less any version numbers.
16. The package must not install any icons on the desktop.
17. The package must not install any uninstall icons in the Start Menu.
18. The Add/Remove entry for an installed package must be in the following format: <Application Name> <Version> <Revision #> Example- Avenue Remote 5.1.8 R1.

19. All shortcuts added to the Start Menu must reside in a unique customer defined folder. Example- Start Menu→<Program name>→<Shortcuts>.
20. Applications with a single shortcut should indicate the version number with the shortcut.
21. Multiple shortcuts for applications with them must reside in a new Start Menu folder named <Application Name> <Version>. The shortcuts themselves must not have version numbers on them.

5.2.4 Application Location

All applications will be installed onto the local machine, unless a technical or business need exists to have an application run from a directory on a server. Justification must be included in the AI document as to why the application cannot be installed locally. For server-based applications, details of server names drive letters and application directories must be included in the AI document.

5.2.5 Shared component upgrades (DLL, OCX)

Upgrades to DLL, OCX and EXE files containing registry/advertising information in the Windows System or Common Files directories are allowed, unless the file is protected by Windows File Protection (WFP). WFP does not allow MSI to update the files it protects; therefore there is no point in making the attempt. A list of most of the protected files can be obtained by listing the directory %SYSTEMROOT%\System32\Dllcache.

5.2.6 Environment

No changes to the PATH variable are allowed by default. Packaging Engineers should use isolation or the App Paths section of the registry, found under HKLM\Software\Microsoft\Windows\CurrentVersion\App Paths.

If it is not possible to use the App Paths sections of the registry and it is absolutely necessary to update the system path, then the Environment table should be used within the MSI. The Registry table must NOT be used. Changes to other environment variables should be made through the Environment table.

5.2.7 Application Dependencies and Requirements

Applications with particular system requirements such as disk space, memory size, and install software should have these requirements checked and built in to the MSI package.

5.2.8 Merge Modules

While the Merge Module technology is quite robust in the latest Wise products, it is believed by Dell that it is best practice to include all of the originally captured files supplied directly by the vendor or developer instead of replacing specific files by choosing to use Merge Modules. If Packaging Engineers are ever prompted to use a Merge Module during the capture process of a MSI package, they should choose No.

5.2.9 Dialogs

The following dialogs will be presented in all packages, including vendor-supplied packages that have been transformed. A native language-specific .MSI package will display the WISE dialogs in the same native language.

Dialog type	Dialogs shown	Description
Install Dialogs	Welcome, Progress	Seen upon first installation
Maintenance Dialogs	Progress only	Seen when reinstalling

NOTE: All dialogs can be reduced and/or suppressed using msiexec switches during the installation for deployment purposes.

5.2.10 Naming Standards

One of the most critical and often overlooked standards in application packaging is the naming standard. It is imperative that an easy to understand naming standard be put in place and used from the very first package created for the project. This ensures that the entire project team understands the package they are prescreening, engineering, quality testing and deploying to production is the correct one.

5.2.10.1 Package Naming Standard

Dell recommends the following naming standard when creating new Windows Installer packages: <Application Name>-<Application Version>-<Language>-<Internal Build Version>.

When creating a new package name all spaces and periods in the original name must be replaced with hyphens. Examples are shown below:
The 1st rev of Acrobat Reader 5.05 (French) → Acrobat-Reader-5-05-FR-R1.
The 2nd rev of Fee Manager 7.5 (English) → Fee-Manager-7-5-EN-R2.

The 3rd rev of Client Tools 4.3 (Multi Language) → Client-Tools-4-3-ML-R3.

Please use this legend when specifying what language the .MSI is engineered for:

Language	Abbreviation
English	EN
German	DE
Multi-Language (must include list of languages in AI document)	ML

This package name must be used in the following places:

1. As the folder name the package resides in on the network.
2. As the prefix to the .MSI name unless it is an existing Vendor .MSI.
3. As the prefix to the .MST name (multiple .MSTs for a single .MSI may have some additional descriptive text added between the <Application Version> and <Internal Build Version> to better depict its usage.
4. As the prefix to the .DOC name.
5. As the prefix for any other standardized import file that resides under the root of the package directory.

NOTE: The total character length of the Package Name must not exceed 45 characters.

5.2.10.2 Directory Structure Naming Standard

Dell recommends that a standardized network file system be put into place for both the engineering of a package as well as the production deployment. The following naming standard is recommended for all phases:

1. **Requirements Gathering**- Drive Letter>\Packaging\Packages\Prescreening\Application Name/Version
2. **Engineering**- Drive Letter>\Packaging\Packages\Engineering\<PackageName>
3. **QA**- Drive Letter>\Packaging\Packages\QA\<PackageName>
4. **Production**- Drive Letter>\Packaging\Packages\Production\<PackageName>

NOTE: In all phases, shares should be established at the root directory of where the packages reside so that variables can be used in command lines for testing purposes.

5.2.11 Allowed Packaging Exceptions

The Packaging Engineers will make an attempt to package each application to the defined Dell standards. If they are unable to package the application to standards and have it function properly, they have been permitted within limits, to have acceptable deviations from the Dell standard provided these deviations are well documented in the AI Document. The following are examples of exceptions that are pre-approved.

5.2.11.1 Files and Directories

1. Target Directory Naming Conventions.
   Example- COTS Applications.
2. Files located in C:\Windows\System32\Drivers
3. Files located in C:\Windows\System32\Fonts
4. Apps requiring Files in more than one directory under C:\Program Files.
5. Additional Files located in C:\Program Files\Common Files

5.2.11.2 File and Directory Permissions

1. Allowed to grant write permissions on files.
   Example- C:\Program Files\MyApp\myfile.ext.
2. Allowed to grant write permissions on folders.
   Example- C:\Program Files\MyApp\Logs.

5.2.11.3 Registry Permissions

1. Allowed to grant write permissions to the User Group to registry keys of the packaged application.
   Example- HKCR\MyApp (MyApp specific keys only).
2. Allowed to grant write permissions to the User Group to registry keys of the packaged application.
   Example- HKLM\Software\MyApp.
3. Allowed to grant write permissions to the User Group to registry values and data of the packaged application.
   Example- HKLM\Software\MyApp\Setting1 | mydata.

5.2.12 Disallowed Packaging Exceptions

The Packaging Engineers have been given limits on how far they can deviate from standards. For some standards there are hard and fast rules that may not be deviated from. The following is a list of standard items Packaging Engineers and Team Leads can not violate during the packaging process. These deviations need prior approval from the customer before being processed.

5.2.12.1 Files and Directories

1. Not allowed to write files located in other directories to enable the application to function.
   Example- C:\
2. Not allowed to write files located in C:\Windows apart from ini files.
3. Not allowed to write files located in C:\Windows\System – 16bit apps.
4. Not allowed to write files located in C:\Windows\System32 – binary hard coded to reference files in this location.

5.2.12.2 File System Permissions

1. Not allowed to grant write permissions to the User Group to the root of C:\ directory.
2. Not allowed to grant write permissions to the User Group to the root of C:\Windows and its children's directories.
3. Not allowed to grant write permissions to the User Group to the C:\Program Files directory.
4. Not allowed to grant write permissions to the User Group to the C:\Program Files\Common Files directory.
5. Not allowed to grant write permissions on files under C:\Windows directory – allowed if the app works when this is done and attempt is made to relocate file that fails.

5.2.12.3 Registry Permissions

1. Not allowed to grant write permissions to the User Group to any root hive.
2. Not allowed to grant write permissions to the User Group to HKLM\Software.

5.3 Examples of Status Codes

Status codes used for tracking work item progress and status through the packaging process consist of the general examples listed in the table below.

TASK	Definition
Reqts Def	Customer – AI Coordinator is obtaining initial requirements. Dell - Returns anything that is not completed as a package (including O/S Validation failure and missing information rejection).
Packaging	Customer – Engineer packages applications based on OneDesk standard.
Re-Packaging	Customer – Feedback from QA on packaging issues or standards.
UAT	Customer – Dell return work code for Work Item that has been successfully packaged but has not completed UAT (i.e. – new packages and Work Item returned from UAT as failed).
UAT (Failed)	Customer – Application does not meet business requirements.
PR	Customer – Used by resources new to the process to valid another engineer's work.
QA	Customer – Used to verify application standards are met and does not break the OneDesk build and Core applications.
QA (Failed)	Customer – Application does not meet OneDesk Standard or does not function according to requirements.
SCM	Customer – Ready for Source Code Management storage system.
Promotion	Customer – Ready for RepliStor.
AD Remediation	Customer – Application does not function under OneDesk and require developer to re-code.
Preliminary Test – WEB	Customer – Application is determined to be a website, initial testing is conducted to verify if any packaging is required.
(DELL) Project Administrator	Dell – Required input task from Customer for all requests.
(DELL) Team Lead Pre-O/S	Dell –Intermediate task. Work Item is pending assignment to engineer for O/S Validation.(Optional).
(DELL) Pre – O/S	Dell – Assignment to engineer for installation and O/S Validation.
(DELL) Team Lead – Packaging	Dell – Intermediate task used to reassign package either immediately after O/S Validation, Failed Packaging, or Peer Review Phase. Rarely used. (Optional).
(DELL) Packaging	Dell – Assigned to engineer for Packaging phase.
(DELL) Re-Packaging	Dell – Application encountered error during the Requirements Definition phase.
(DELL) Team Lead – Peer Review	Dell –Intermediate task used to reassign Work Item from Packaging to Peer Review phase. (Optional).
(DELL) Team Lead Post Peer Review	Dell – Peer Review phase completed successfully and Work Item is ready for return to Customer as UAT.
(DELL) Pre-Peer Review	Dell – Intermediate task. Work Item is assigned to engineer and awaiting Peer Review. (Optional).

(DELL) Peer Review	Dell – Assigned to engineer for Peer Review phase.
(DELL) RE-Peer Review	Dell – Equivalent to (Dell) Peer Review, used for additional Peer Review.
(DELL) Rejected: Reqts Def	Dell – Application encountered error during the Requirements Definition phase.
(DELL) Failed on install	Dell – Application encountered error during the Install phase.
(DELL) Failed on Validation (primary OS)	Dell – Application encountered error during the O/S Validation phase. (Billing trigger).
(DELL) Failed on Packaging (primary OS)	Dell - Application encountered error during the Packaging phase. Work Item is returned to Customer as "O/S Validation failure (Billing trigger).
(DELL) UAT Re-work	Dell - Work in progress (Project Administrator not assigned) after UAT (Failed), or return to Customer (Project Administrator assigned) as warranty work.
(DELL) UAT Re-work Peer Review	Dell - Equivalent to (Dell) Peer Review, used for Work Item returned to AI factory with 'Rework' status (and requires additional Peer Review).
(DELL) Waiting Information	Dell – Work Item is on hold pending resolution of external issue.
(DELL) – QA Rework	Dell - Work in progress (Project Administrator not assigned) after QA (Failed), or return to Customer (Project Administrator assigned) as warranty work.
(DELL) – UAT Reqts. Change	Dell – Used for Work Item reassigned to Dell after UAT (Failed) or QA (Failed) items that qualify as a change in the original requirements and require MSI edits and a Peer Review. Equivalent to (Dell) Packaging or Team Lead Post Peer Review (as appropriate).
(DELL) – UAT MSI Edit	Dell – Used for Work Item reassigned to Dell after UAT (Failed) or QA (Failed) items that qualify as a change in the original requirements and require MSI editing but no additional Peer Review. Equivalent to (Dell) Packaging or Team Lead Post Peer Review (as appropriate).
(DELL) – UAT Doc Edit	Dell – Used for Work Item reassigned to Dell after UAT (Failed) or QA (Failed) items that qualify as a change in the original requirements and requires only document changes (simple edits) but no MSI code changes or Peer Review. Equivalent to (Dell) Packaging or Team Lead Post Peer Review (as appropriate).
(DELL) – Localization	Dell – Work Item requires an MST and is based on an approved and existing MSI. Equivalent to (Dell) Project Administrator, (Dell) Packaging, or (Dell) Team Lead Post Peer Review.