

Samuli Laamanen

Application Deployment in an Enterprise

Bachelor's Thesis
Information Technology

December 2016



KYAMK
University of Applied Sciences

Tekijä Samuli Laamanen	Tutkinto Insinööri	Aika Joulukuu 2016
Opinnäytetyön nimi Application Deployment in an Enterprise		33 sivua 1 liitesivua
Toimeksiantaja Salainen		
Ohjaaja Yliopettaja Martti Kettunen		
Tiivistelmä <p>Yritysten kasvaessa myös erilaisten sovellusten tarve kasvaa. Tämä aiheuttaa tilanteita, joissa yhtiöillä on kymmeniä sovelluksia, eikä mitään tapaa hallita niitä. Sovellusten ja niiden levityksen hallinta ovat tämän opinnäytetyön tarkastelun pääkohteita.</p> <p>Opinnäytetyössä tutkitaan kolmea eri aihetta. Suuren kansainvälisen yrityksen sovelluksen käyttöönottoprosessi, virheiden selvitys sovellusten käyttöönotoissa ja raportointi sovellusten käyttöönototuloksista. Opinnäytetyö kuvaa sovelluksen käyttöönoton vaiheita ja mitä rooleja edellytetään turvalliselle ja hallitulle sovelluksen käyttöönotolle. Tavoitteena on sovellusten käyttöönotosta standardisoitu prosessi, jota, voitaisiin hyödyntää luomaan SOP eli Standard Operating Procedure.</p> <p>Opinnäytetyö kuvaa myös yleisimpiä sovelluksen asennuksen yhteydessä tulevia virheilmoituksia kohdeyrityksessä ja miten niitä voidaan ratkaista. Opinnäytetyötä voidaan käyttää ohjeen tekemiseen paikalliselle IT-tuelle. Näin voidaan järjestää aikaisempaa nopeampi ja parempi käyttäjien IT-tuki</p> <p>Yksi opinnäytetyön tavoitteista oli luoda sovelluksen käyttöönotosta raportteja, jotka näyttäisivät, kuinka hyvin sovelluksen käyttöönotto onnistui. Sovelluksen levitysraporttia testattiin muutamalla sovelluslevityksellä ja palaute oli hyvin positiivista. IT-tuelle suunniteltiin ja parametrisoitiin raportti, mutta, ajan puutteen takia raporttia ei saatu valmiiksi tätä opinnäytetyötä varten.</p> <p>Tulokset opinnäytetyöstä olivat hyvin positiivisia. Sovelluksen käyttöönottojen vaiheet luotiin selvillä rooleilla ja nimeämisstandardeilla, yleisimmät virhekoodit selvitettiin ja virhekoodien tutkimisen aikana löytyi hyödyllistä dataa, jota voidaan käyttää myöhemmin luomaan monimutkaisempia virheraportteja. Tätä opinnäytetyötä voidaan käyttää pohjana, kun suunnitellaan, miten luoda sovelluksen käyttöönottoprosessi muihin yrityksiin.</p>		
Asiasanat sovellus, levitys, ongelmanratkenta, raportointi, Microsoft SCCM, Report Builder		

Author Samuli Laamanen	Degree Bachelor of Engineering	Time December 2016
Thesis Title Application Deployment in an Enterprise		33 pages 1 pages of appendices
Commissioned by Classified		
Supervisor Martti Kettunen, Principal Lecturer		
Abstract <p>When companies grow their need for different kind of applications grows as well. This causes situations where companies have dozens of applications and no way to manage them. Managing how these applications and their updates are deployed is the main focus of this thesis.</p> <p>In this thesis three main topics are investigated. Application deployment process, troubleshooting errors with application deployments and reporting the results of application deployments in a large international company. This thesis describes the phases of application deployment and what roles are required for a safe and managed application deployment process. The aim is to create a standardized process of application deployments which could be used to create an SOP.</p> <p>This thesis also describes the most common errors in the company and how to solve them. This troubleshooting step can be used to create a guide for IT-support which could result in faster support for the users.</p> <p>One of the goals of this thesis was to create application deployment reports that show how well application deployment went. This report was tested with few deployments and the feedback was very positive. The report for IT-support was parametrized and planned, but due to time constrains not finished for this thesis.</p> <p>The results of this thesis were very positive. The phases of application deployments were created with clear roles and naming standards, most common error codes were explained and during the investigation of error codes, valuable data was found that can be used later to create more complex error reporting. This thesis can be used as a base when planning on how to create a proper application deployment process.</p>		
Keywords application, deployment, troubleshooting, reporting, Microsoft SCCM, Report Builder		

CONTENT

TERMS AND ABBREVIATIONS	5
1 INTRODUCTION	7
1.1 Idea for thesis	8
1.2 The Company	9
1.3 Application Request and Approval in Enterprise.....	9
2 APPLICATION MANAGEMENT	10
2.1 Application Deployment.....	11
2.2 Deployment Groups.....	12
3 APPLICATIONS USED.....	13
3.1 System Center Configuration Manager.....	13
3.2 Orchestrator.....	15
3.3 Active Directory.....	16
3.4 Group Policy Object.....	17
3.5 Report Builder 3.0.....	18
3.6 PowerShell App Deployment Toolkit.....	19
4 FROM AN IT PERSPECTIVE	20
5 DEPLOYMENTS.....	21
5.1 Planning Phase.....	22
5.2 Piloting Phase.....	22
5.3 Rollout Phase	22
5.4 Closing Phase.....	23
5.5 Troubleshooting	23
5.6 Special Deployments.....	26
6 REPORTING	27
6.1 For Application Owners	27
6.1.1 Building the Report.....	28
6.2 For IT Personnel.....	29
7 CONCLUSION.....	30
8 REFERENCES	32

APPENDICES

Appendix 1. Audit Diagram of application deployment phases.

TERMS AND ABBREVIATIONS

.MSI	MSI-file is an installer package file format used by Windows. It is used for installations, storage and the removal of applications.
.MST	MST-file contains modifications that are to be applied on a .MSI-file, like language selection.
AD	Active Directory is Microsoft's solution to the issue of keeping track of everything in a company's network, enforcing policies, authenticating users and computers.
Application Owner	Owner of the application. Is responsible for managing possible licenses and supports IT Administrators with issues with the application.
Asset	Application or a computer.
ConfigMgr	Part of the Microsoft System Center Configuration Manager suite of tools for IT Administrators.
Deployment	Way to distribute applications via either forcing an installation or publishing it in software center.
Deployment Owner	Creator of the deployment, is responsible for the deployment.
IT Administrator	IT Administrator is either a member of the Application Management team or SCCM team.
Local IT	Local IT is the first level IT support which manages certain regions IT issues and escalates all unsolvable issues to IT Administrators.
Runbook	Is a set of defined actions used in Microsoft Orchestrator.

SCCM	Microsoft System Center Configuration Manager, Microsoft's solution for managing large amount of computers and applications.
SQL	Structured Query Language which is used when modifying reports in Report Builder 3.0
Software Center	Part of the System Center suite of applications, used to distribute applications to users. Also has application catalog for users to browse and download applications.
SOP	Standard Operational Procedure, a clear procedure on how to make certain things for example how to make an application deployment.
System Center	Microsoft's Suite of application used to assets in an enterprise.
TTL	Time to Live, how long should deployment be active.
Report Builder	Microsoft's tool for building reports by fishing data from SQL database.
VBA	Visual Basic for Applications is a programming language used to automate processes and building user-defined functions. Used in many Microsoft products like Microsoft Report Builder.

1 INTRODUCTION

In modern companies there are many applications, each designed to do certain tasks or activities for the user. These applications need to be kept up-to-date by IT support personnel, but when the amount of applications grows over certain point, managing them manually is no longer a cost-effective solution.

Applications are one of the most important aspects of IT. Applications are required by both IT personnel and users to perform their daily jobs effectively and without proper management, distributing applications or application updates might cause large scale problems. Applications deployments must be managed by IT Administrators so applications can be reliably distributed to users, supporting the company and therefore gain major boosts in effectiveness for the users (Singh, 2016).

This thesis focuses on the end steps of the Application deployment process in a large, international company. The plan is to study the current status of application deployment inside the company and improve upon on it. One main area of focus will be the reporting aspect to the application owners within the company and how this reporting will be done and managed. The reports should only be visible to certain people, like application owner and IT administrators, and these reports should be self-updating. Application owner should be able to effectively see the reports, therefore they would have to be built on either on some internal website, like SharePoint or distributed by email.

Application deployments are a big part of this thesis. The company has hundreds of deployments and managing them is a difficult task and requires a lot of effort. This thesis proposes a standard operational procedure or commonly known as SOP for application deployments and possible documentations for such process. The goal is to use this thesis as a basis for creating a SOP in application deployments with SCCM Configuration manager, which would include reporting for application owners and IT Personnel and also TTL (time to live) for deployments.

1.1 Idea for thesis

In any large to mid-size company there comes a time when IT personnel can no longer manage its user's computers without some sort of managing software. Users will require applications to do their job and for managing these applications there are a few good options. This thesis is focusing on Microsoft Software Center Configuration Manager or ConfigMgr for short, because there was an already built ConfigMgr environment available.

While working at the company as a summer employee and tasked with creating multiple application deployments for various reasons it became clear that closing them and determining when the deployment was done, was not so simple. The company had a complete setup of system center configuration manager (SCCM) server with around 10 000 computers and over 15000 users. There were many applications for different departments and these applications had regional differences, like license servers. There was a need for a standard on how to manage application deployments with a clear definition when said deployment should be done. With application deployments, there were approximately 1 to 2% of computers where the deployment failed due to different reasons and as the deployments were getting older and new ones were being made, this resulted in these "problem computers" never getting fixed and them always being behind on application updates.

One large part of this thesis is the reporting of application deployments since owners of the applications often wanted to see how well their application installed on the company computers and if there were any errors in the process. When the application owner requested the report, on how their application performed, only a crude excel chart could be provided. ConfigMgr has some basic reporting capabilities which will be used to improve upon the reporting or used as a template for new and better application deployment reports with visual aspects, such as pie charts.

This thesis will propose possible solutions to most common problems in application deployment process and reporting and provide a list of the most common problem codes. This can be used to create some sort of guide for local IT or service desk so that they can solve these issues more efficiently.

1.2 The Company

The company referred to in this thesis is a global industrial engineering and manufacturing company. It specializes in rotating equipment maintenance, chemical industry mixing technologies and pumping systems. Total revenue in 2015 was approximately 3 billion euros and it has over 10 000 employees worldwide with over 15 000 IT devices. This thesis will contain information about IT infrastructure of the company and due to security reasons the name of the company will not be provided. Since this thesis is made for the company it is written in English.

The company has one centralized global IT management location from where SCCM is used to manage the company's IT assets.

1.3 Application Request and Approval in Enterprise

Another thesis about application approval process has been created by another student also employed at the company, while this thesis focuses on the application deployment process which is the other end of the application lifecycle process. Application Approval is needed when an application which is licensed is being requested from a software center, which is like an application store. The application request is then processed by the application owner, who then decides if that request gets approved or denied, occasionally the application owner would consult with the user's line manager on whether or not the application request should be approved. After the application request has been approved, the user can then install the application from the software center which results in the application being deployed to their computer.

The thesis focused on the research and development of application distribution and approval, while this thesis focuses on application deployment and reporting. Two proposals were made in their thesis on how to handle application deployments (via using kiosk like application catalog where users can install application by themselves) in the future. One way was to utilize SCCM provided tools like Software Center or to use Service Now, which is a separate application to present more informative and visually prettier view of the application catalog (Hartikainen, 2016). This thesis focuses on application push deployment, which is a way to force installation of applications to user's computer.

2 APPLICATION MANAGEMENT

Application management is a process that centers around application-related aspects of designing, testing, operating and improving on IT services. It is vital for providing value to a business as without proper tools the users can't provide value to the company. Application management revolves around managing application throughout its entire lifecycle (Zitek, 2014).

This thesis focuses on the end steps of Application Management, which is a large process with several steps and therefore the first steps are only mentioned briefly. Every step of Application management is a large subject on its own and multiple studies could be made covering the entire process. Application management begins with a user having a need for a specific function. User then requests for an application that could do that needed function and once the application has been deemed necessary an application packaging request is created and passed on to the application management team. Usually the user who made the request for the application will become the application owner and is then responsible for license management and any possible questions and concerns about that application.

The application request is then processed by the application management team who, in close collaboration with the packaging team will acquire the installation medias and instructions on how the application should be installed from the application owner. Once Application Management team has the required medias and instructions the application request goes to the application packaging team who then creates a "package" of the application. This package includes the installation medias with the selections and modifications required in the installation instructions. Then the package is tested by the application owner and if they approve it, it will be deployed.

In application management there are 3 main figures, application owner, application manager and application user. Application owner is the one who requests the new application, manages licenses and is responsible for the application during its lifecycle. Application manager manages new application requests and is responsible for application deployment and maintenance. Application owner and application manager can be the same person. Application user is the one using the application. New applications should always provide value to users (techopedia, 2016).

As more and more features are required due to new acquisitions or expansions, the applications become more and more complex. Of course some applications need to be interconnected to enable interaction sharing across the company which creates more complex environments. To keep these environments working and up to date they require maintenance (Siemens AG, 2010). In the company application management was managed via SharePoint site queue system and in close collaboration with packaging team.

2.1 Application Deployment

Application Deployment process is the main focus of this thesis. It is the process of deploying applications such as updates, application updates and security updates and it has many steps. Once an application has been packaged and tested by the application owner it can be deployed to a kiosk-style software center where the users can choose the application and freely download it or request it. Or the application can be “pushed” to the target computer remotely, so that the users do not have to install the application themselves.

his process has many factors that have to be taken into consideration. Rarely, the IT environment where application is being deployed is homogeneous which means there will most likely be some errors.

Application Deployment could be spilt into 4 phases, which will be explained later in this thesis. First there is the clarification of the targets. Who needs this software and why? Then the creation and starting of the deployment. Third is the maintenance phase which is followed by the last phase known as the closing phase, where the deployment is removed from being deployed and all the failure clients have been either fixed or identified.

Application Deployment can end in four different statuses on the user’s computer:

- § **Success** – The deployment was successful and the application was found to be installed.
- § **In Progress** – The application deployment is installing the deployed application or downloading the necessary files.
- § **Unknown** – The state of the application deployment could not be determined. This state is typically displayed when state messages from the client is not yet received.

- § **Requirements Not Met** – The application was not deployed due to it not being compliant with a dependency or a requirement rule, or because the operating system to which it was deployed was not compatible with the application or the free space on the computer hard drive ran out.
- § **Error** – The application failed to deploy because of an error.

2.2 Deployment Groups

Using ConfigMgr, two types of deployment methods can be used. This choice has to be made preferably when building the SCCM environment to the company as it is very integral to ConfigMgr. Application can be deployed using either Active Directory's user groups or ConfigMgr's Device collections. Both of these methods have their own pros and cons.

Deploying applications using AD groups:

Pros:

- If the user ever needs to change their computer the applications are automatically deployed to the new computer, removing the need to re-deploy applications to the new computer.
- Local IT can assign people into groups who the software is then installed. Modifying an AD group can be done by local IT and does not require IT administrators.
- These groups can be made without SCCM knowledge.
- Applications are automatically pushed to the user's computer so there is no need to request applications using Software Center.
- When a user is removed from the user group in AD it will automatically trigger the uninstallation of the installed application.

Cons:

- It will require a lot of work from the IT Administrators to make sure that the primary device for each user is correct.
- Having multiple machines as a user's primary device might result in multiple, unnecessary installations of a licensed application.
- There has to be a user group for each application.
- When removing the user from the AD's user group the uninstallation might not trigger, as it is not very reliable.
- As more people have rights to add people into AD user groups tracking users will become difficult. Determining how, why and when a user got added to the AD group for an application installation will be more difficult this way. For example, a certain user might change departments or a new version of that application gets released it is easy to forget to remove that user from the old group.

Deploying applications using ConfigMgr's device collections:

Pros:

- Device collections are controlled so much better, targeted deployments can be made.
- Smaller group of people working on the deployment, each with knowledge of SCCM lowering the chance of errors.
- Easy to deploy applications to servers and to computers with multiple primary owners.

Cons:

- SCCM knowledge required to make deployments.

In the company, ConfigMgr's device collections was decided to be used due to the fact that AD group deployments have more cons than pros.

3 APPLICATIONS USED

3.1 System Center Configuration Manager

Microsoft System Center Configuration Manager also known as SCCM is a part of Microsoft System Center line of products that are developed for IT administrators.

System Center Configuration Manager is Microsoft's solution for comprehensively assessing, deploying and updating servers, clients and different environments like mobile. It is optimized for Windows and it is built around other Microsoft systems such as Windows Server Update Services, Windows Server Active Directory and Windows Architecture (Microsoft, 2016).

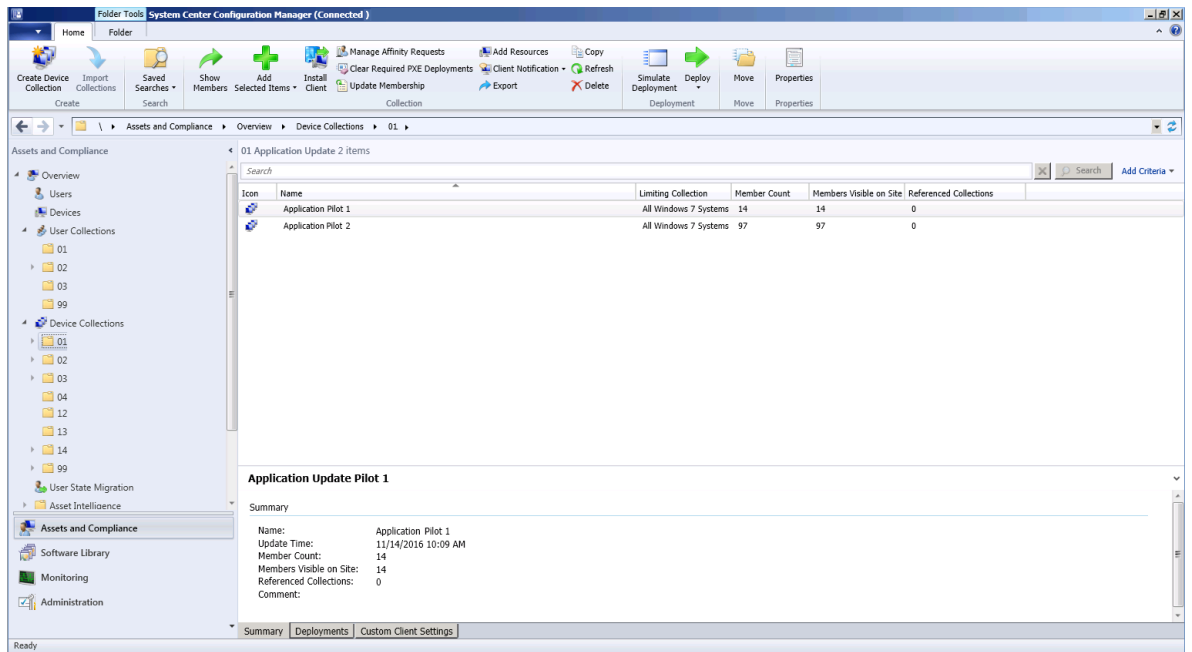


Figure 1: SCCM Default View of the Assets

Main features of the Configuration Manager are the abilities to distribute software, manage patching, hold software and hardware inventory and deploy operating systems. ConfigMgr is one of the most important applications used in the company (Hartikainen, 2016). Application deployment, reporting and staging new computers are a few of the main uses for ConfigMgr within the company.

In ConfigMgr the applications are managed in hierarchical arrangement sorting the applications in a few categories, making it easier to manage large amounts of applications. Deploying applications is done by using ConfigMgr and its features like device collections, which is a collection of user's devices, servers or mobile devices within the company.

In the company most users have one computer and this computer is called their "Primary Device". This primary device can be tagged to a user in a couple of ways:

- During Staging, which is a process where a computer has its operating system installed with the company's chosen modifications and added to Active Directory with information about the computer for example its user or the location.
- After installation from the Software Center
- Via ConfigMgr using the "Usage Agent Defined" attribute which marks certain a user as that computers primary user if the user uses that computer enough.

3.2 Orchestrator

In IT, scripting complex systems takes a long time and requires extreme diligence. Microsoft's solution for this is System Center Orchestrator which allows automate scripting using rather simple graphical user interface or GUI for short. It is capable of managing multiple operating systems and also supports virtual machines. Orchestrator was originally a third-party product called Opalis, which Microsoft acquired in 2009 (Rouse, 2013).

Orchestrator integrates with many different Microsoft products and non-Microsoft products to enable building very complex system across data center. With Orchestrator's GUI it's easy to create run books without any scripting. Orchestrator then translates these visual run books into PowerShell, .Net or SSH commands to automate workflows (Microsoft, 2016). These Runbook's are a set of activities, most common would be a check if a service is running. The Runbook monitors a service to see if it's running and if it notices that the service is down it will try to restart the service and it will send a notification to the IT Administrator that the service has been restarted.

Orchestrator is mostly used in the following tasks:

- Automating processes in data centers, regardless of hardware or platform.
- Automating IT operations and standardizing so called "best practices" to improve efficiency.
- Connecting many different systems from different manufacturers without the need of scripting (Microsoft, 2016).

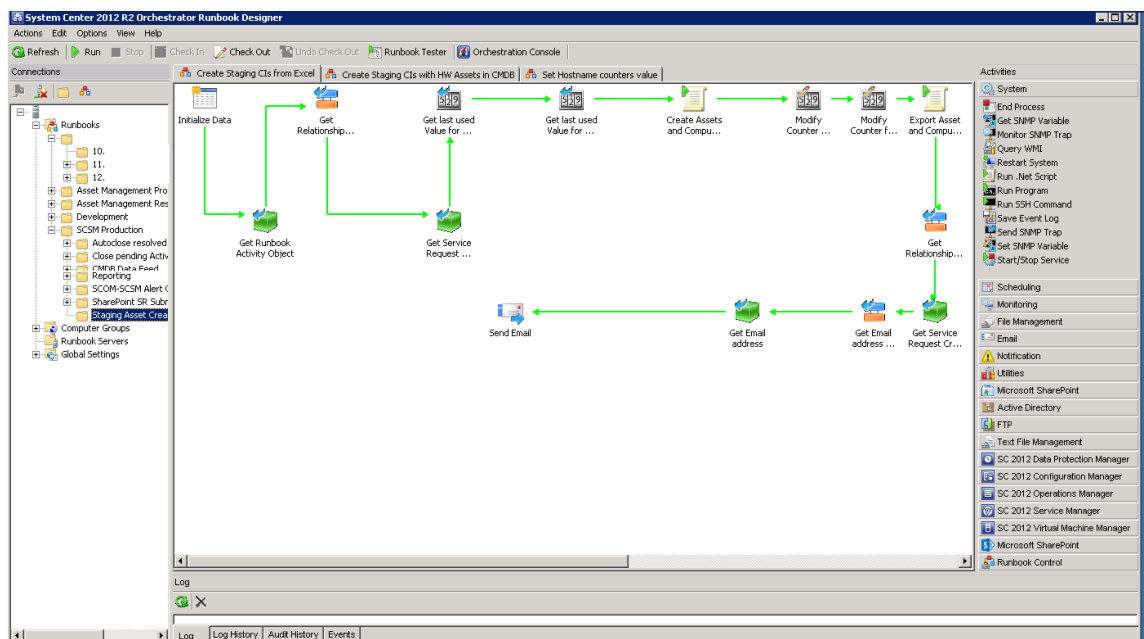


Figure 2: Example of Orchestrator Runbook.

3.3 Active Directory

Active Directory also known as AD is Microsoft's solution to the issue of keeping track of everything in a company's network. It is a combination of different services like Domain services, Certificate services, Federate services and Lightweight directory services.

Domain services are a comprehensive list of users, user's information such as phone numbers, addresses and computers. It also verifies their credentials and defines their access rights. Everything that Active Directory tracks are called *objects* and they can be sorted into groups making managing a large amount of them easy and efficient (Microsoft, 2016).

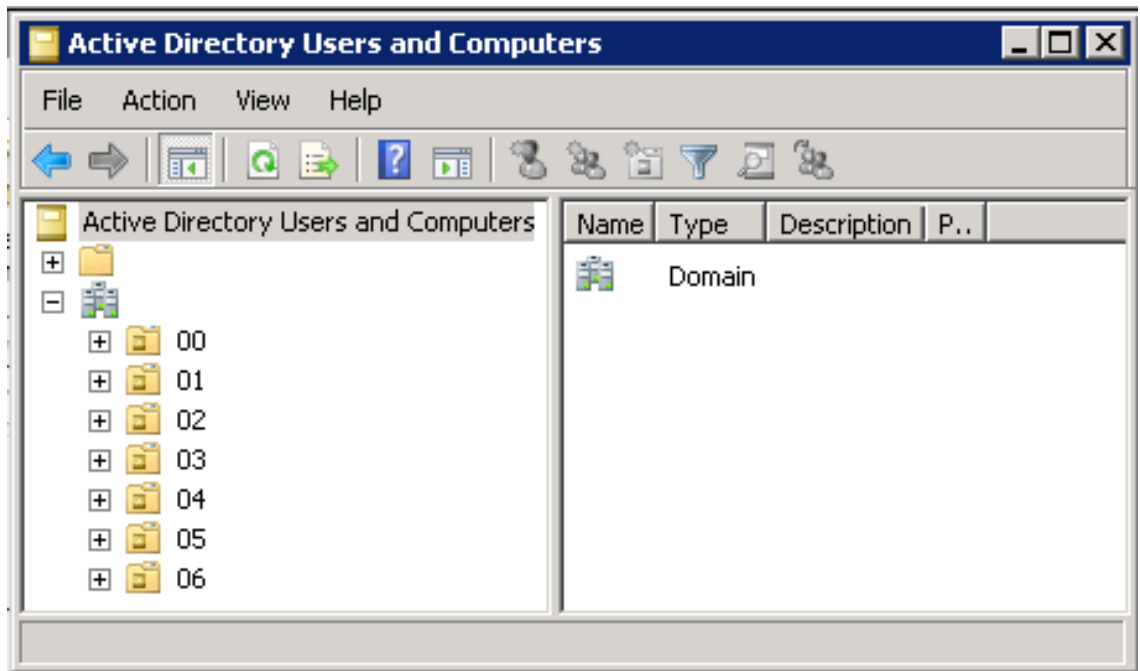


Figure 3: Example of Active Directory

Active Directory stores information about the components in the network. The server which is running Active Directory Domain Services is called a domain controller and its responsibility is to authenticate and authorize all users and computers, assign and enforce security policies in the domain.

Usually AD and SCCM's ConfigMgr are synched, which allows new computers created in ConfigMgr also to be created in AD. Changes made using AD also affect ConfigMgr's user data. ConfigMgr allows IT administrators to use AD's user groups to deploy updates and applications.

For example, if an application is licensed only to Swedish users it is then only available to users in Sweden.

3.4 Group Policy Object

In the company there is also a Group Policy Object (also known as GPO) management system in place. Group Policies allows the IT Administrators to manage, configure and modify, user- and computer settings from a centralized management view in an AD environment. GPO's can be used to define how the system will look like and what settings are used for certain AD users or user groups. IT Administrators can create GPO's that define registry-based policies, application settings, script options, internet security options and network folders. GPO's are very often used to set Java exceptions, XML sites and to maintain Microsoft Office products settings (Lee, 2010). IT Administrators can also use GPO's to install applications but that is not used commonly. GPO's are used in the company to set certain application settings, create desktop shortcuts and maintain Internet settings.

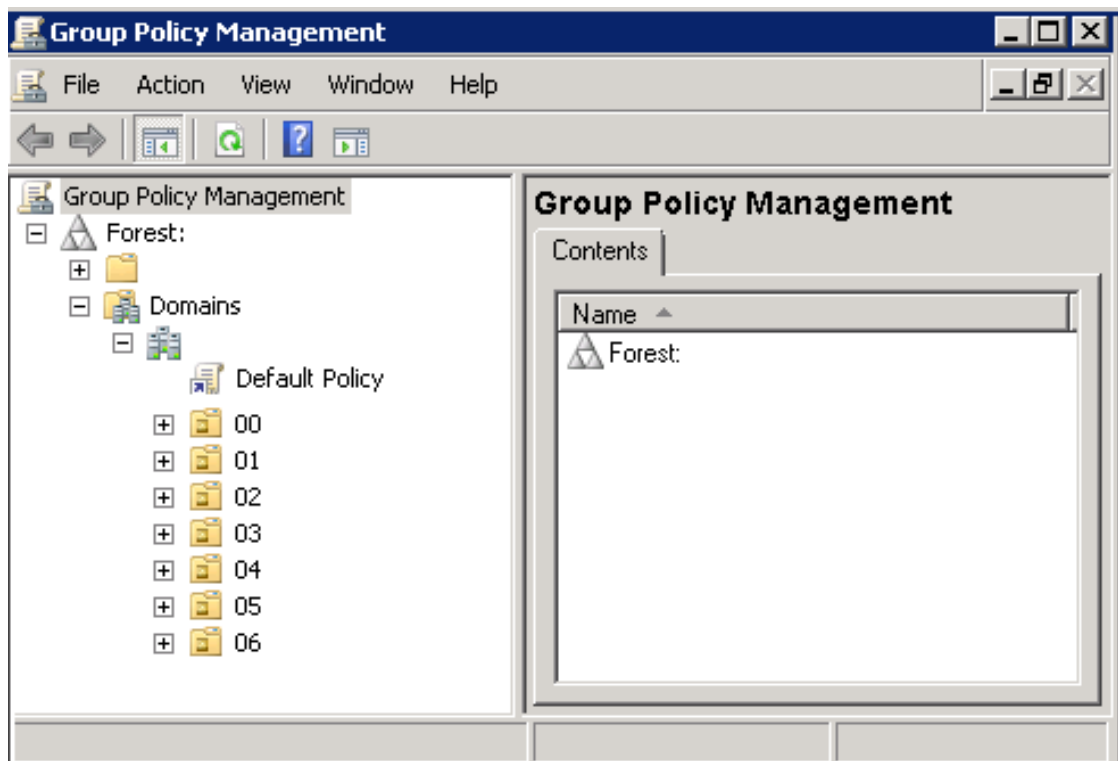


Figure 4: Example of Group Policy Management console

3.5 Report Builder 3.0

Microsoft Report Builder 3.0 is a part of the Microsoft SQL Server assortment of tools. With report builder IT administrators can build complex reports with relative ease. Report Builder allows this by utilizing simple GUI and easy to learn features. Report Builder's main features are:

- Easy to use Wizard – Creating a new Table or Chart start a wizard that makes making reports much easier.
- Freely modifiable layout – IT Administrators can create different type of reports for different types of data. Charts for graphical reports and column-based data for table reports. Reports can have many different types and combinations of charts and matrix layouts.
- Support for other reports – IT Administrates can modify existing reports that have been made with other report designers.
- Interactive reports – IT Administrators can add interactive features for users like show/hide buttons, drill through links to sub-reports and document maps.
- Supports many formats – IT Administrators can choose what format the report will be opened. Formats include HTML, MHTML, PDF, XML, CSV, TIFF, Word, and Excel.
- Pictures – IT Administrators can embed images and other resources in their reports.
- Subscriptions - After making a report on the reporting server, the report can be configured to run at specific times o to set up an email subscription.

To use Report Builder there has to be a SQL server already in place. In the company the target SQL server is the SCCM's SQL database where all the information such as computers, installed software's and users are stored. By using Report Builder and ConfigMgr's default reports it is much easier to create largescale and complex reports.

Report Builder 3.0 includes several useful features, such as The Report Part Gallery which allows IT Administrators to re-use existing report parts as building blocks when creating new reports (Microsoft, 2016).

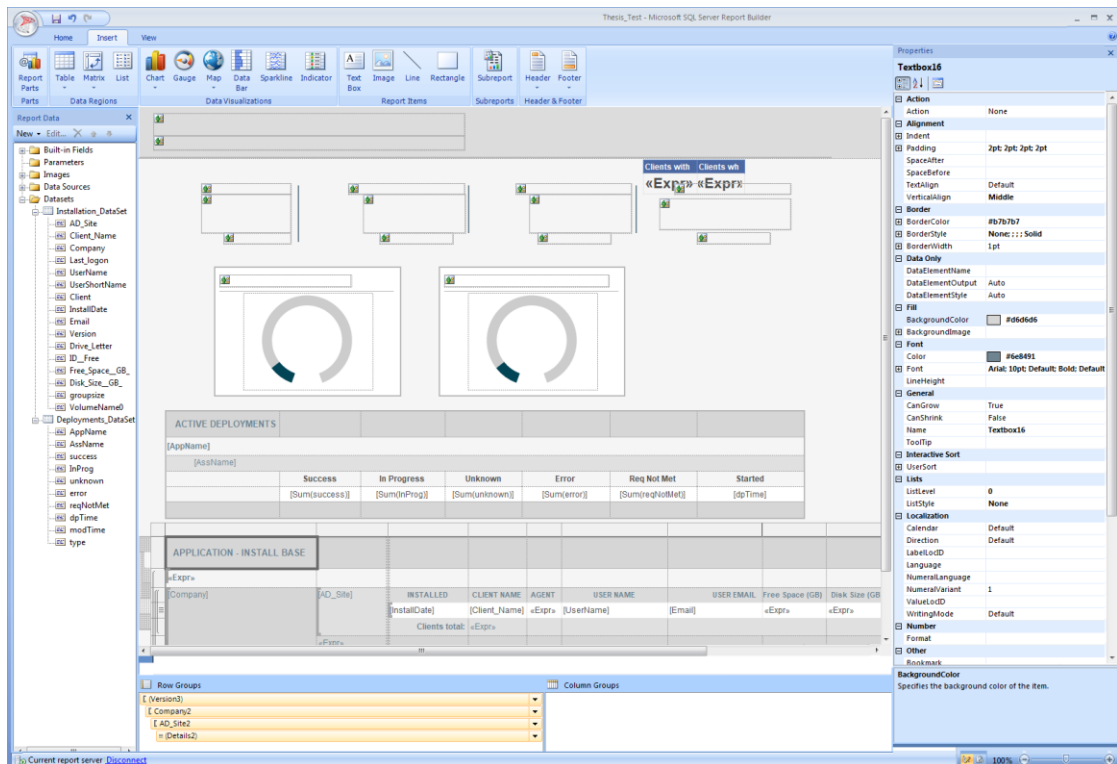


Figure 5: Example of self-created report

3.6 PowerShell App Deployment Toolkit

PowerShell App Deployment Toolkit, referred to in this thesis as the Toolkit is a set of functions used to perform common application deployment tasks and to provide notices and pop-ups for the user (Lillis, 2013). Main features of the Toolkit are:

- The Toolkit is very easy to use and any IT Administrator with basic PowerShell knowledge can use it.
- Using the Toolkit, you can create standardize application deployments regardless of the application.
- Has a set of functions to perform often used deployment tasks, such as installing and uninstalling multiple applications, setting registry keys, copying files and prompting users to close applications.
- The Toolkit has features to provide user dialog boxes and notifications.
- The UI supports different languages.
- Integrates with SCCM.
- The Toolkit gets regular updates which are always backwards compatible.
- The Toolkit also support custom scripts and functions.

With the Toolkit it is possible to create user interface screenshots, create pop-ups that prompt the user to close certain application that might interfere with the installation, give the user the option to delay the installation X number of times or X number of days, prevent user from launching applications that might cause issues with installation of the

deployed application and many other useful features. The Toolkit is used in the company to create installation logs, prompt the users to close currently used applications that the installer requires, prompt log off, create logging for installations/uninstallations and setting registry keys (Lillis, 2013).

```

1  <#
2  .SYNOPSIS
3  This script performs the installation or uninstallation of an application(s).
4  .DESCRIPTION
5  The script is provided as a template to perform an install or uninstall of an application(s).
6  The script either performs an "Install" deployment type or an "Uninstall" deployment type.
7  The install deployment type is broken down into 3 main sections/phases: Pre-Install, Install, and Post-Install.
8  The script dot-sources the AppDeployToolkitMain.ps1 script which contains the logic and functions required to install or uninstall an application.
9  .PARAMETER DeploymentType
10 The type of deployment to perform. Default is: Install.
11 .PARAMETER DeployMode
12 Specifies whether the installation should be run in Interactive, Silent, or NonInteractive mode. Default is: Interactive. Options: Interactive = Shows dialogs, Silent = No dialogs,
13 NonInteractive = Very silent, i.e. no blocking apps. NonInteractive mode is automatically set if it is detected that the process is not user interactive.
14 .PARAMETER AllowRebootPassThru
15 Allows the 3010 return code (requires restart) to be passed back to the parent process (e.g. SCCM) if detected from an installation. If 3010 is passed back to SCCM, a reboot prompt will be triggered.
16 .PARAMETER TerminalServerMode
17 Changes to "user install mode" and back to "user execute mode" for installing/uninstalling applications for Remote Desktop Session Hosts/Citrix servers.
18 .EXAMPLE
19 Deploy-Application.ps1
20 .EXAMPLE
21 Deploy-Application.ps1 -DeployMode 'Silent'
22 .EXAMPLE
23 Deploy-Application.ps1 -AllowRebootPassThru -AllowDefer
24 .EXAMPLE
25 Deploy-Application.ps1 -DeploymentType Uninstall
26 .NOTES
27 .LINK
28 http://psappdeploytoolkit.codeplex.com
29 #>
30 [CmdletBinding()]
31 Param (
32     [Parameter(Mandatory=$false)]
33     [ValidateSet('Install','Uninstall')]
34     [string]$DeploymentType = 'Install',
35     [Parameter(Mandatory=$false)]
36     [ValidateSet('Interactive','Silent','NonInteractive')]
37     [string]$DeployMode = 'Interactive',
38     [Parameter(Mandatory=$false)]
39     [switch]$AllowRebootPassThru = $false,
40     [Parameter(Mandatory=$false)]
41     [switch]$TerminalServerMode = $false
42 )
43
44
45 ## Set the script execution policy for this process
46 Try { Set-ExecutionPolicy -ExecutionPolicy 'ByPass' -Scope 'Process' -Force -ErrorAction 'Stop' } Catch {}
47
48 ##=====
49 ## VARIABLE DECLARATION
50 ##=====
51 ## Variables: Application

```

Figure 6: Example of the Toolkit being used in an installation script.

4 FROM AN IT PERSPECTIVE

One of the problems in the company is with Local IT not having enough information to identify and fix issues. If the IT Administrators deploy an application update and ConfigMgr's deployment status screen shows that the installation has failed on 90% of the targeted clients, the IT Administrators have to start informing local IT personnel about the failed deployment with guide lines on how to fix the issue. Local IT has very little visibility on the overall status of application deployments. This results in cases where problems unnecessarily escalate and where large scale problems can happen unnoticed by local IT.

In a perfect world when IT Administrators would notify local IT about a failed deployment, they could see the status of the deployment via a separate site like SharePoint or some other internal site and see the list of users in their location and on whom computer the deployment has failed. Local IT could then pick users who have experienced an issue from their location and attempt to fix it and if any further help is needed they could contact IT Administrators.

Largest issue with providing the error codes to Local IT is that the error codes generated by ConfigMgr are not very clear. Most common error is the error

code 1603 which signifies a general error meaning that “something went wrong”. From the installation logs of the application package some hints could be found why the installation failed, but most of the time resolving these kinds of errors requires a lot of time and effort. Most of the packages deployed have a common or already detected problem where general error codes mean certain things. For example, application X requires that application Y is in a certain version, but application Y has been updated, which would cause the installation of application X to fail and then the installation package would generate error code 1603 which the users’ computer would then report to ConfigMgr. That’s why a separate document explaining these error codes and their solutions should be created by the IT Administrators. This way Local IT can provide fast and reliable support to the users without unnecessary escalations.

5 DEPLOYMENTS

This chapter covers the steps that should be taken in order to make a safe deployment with clear roles and responsibilities. Package testing is not included since it is a part of the application packaging process.

For application deployments there should be four different deployment types: Pilot, baseline, standard and test. Pilot deployments are made when the application has been tested and approved to be working by the application owner. After Pilot Deployment has been done and the applications has been proven to work, the application can be rolled out.

Rollout deployments can be either Standard or Baseline deployments. Standard deployments are done to certain group and once the deployment has expired or all the installations have completed the deployment is closed or moved to software center. Baseline deployments are often made to all users and baseline deployments never expire, for example, the company requires that all of its users always have newest version of Adobe Flash on their computers due to security reasons so Adobe Flash will be continually pushed to all of the company’s users. To see an example of how this process works, see appendix 1.

5.1 Planning Phase

When creating a deployment, the first step is to determine the target group and assessing possible risks. Even a small application might cause a large scale issue, for example the application being deployed is a browser plug-in and for example some bank sites do not support that plug-in, which again might cause large scale issues within these sites like payments not going through. Proper testing referred to as piloting is a must.

In the planning phase the pilot deployment is planned, target group is defined and the application owner is set. The target group must be select with the knowledge that they actually need the software. Deploying an application to users who do not actually use that application will just waste space on their computer and the test results might not be accurate. Target groups should be clearly defined by the application owner.

5.2 Piloting Phase

Piloting phase is one of the most important parts of any application deployment. The application has to be properly tested in all situations with variety of computers so all possible problems are found and then fixed. This however rarely happens. Application Management is a process that takes time which might not be apparent to the users and when a user requires a new application the schedule is very tight so proper testing is often neglected. This will result in inaccurate testing results and that all problems with the application package will not be found during the rollout phase.

Piloting should include proper testing of the application and all its add-ins and possible compatibility features with, for example browsers. The amount of pilot users should be approximately 10% out of all users for this application.

5.3 Rollout Phase

In the Rollout phase deployment is scheduled with the application owner. IT Administrators should schedule the deployment during a time when it does not disrupt the users daily work activities or cause unnecessary traffic on the network. If the application installation requires a restart, the deployment should be done during the weekend or at pre-arranged time scheduled with the users. In most cases deployments, such as application updates and small modifications, can be done during business hours by making the installation or

update silent. Application owner has to be very active with this phase as they are the one responsible for the application.

IT Administrator who makes the application deployment in ConfigMgr becomes the deployment owner. After making the deployment, the deployment owner should keep an eye out for errors or other issues. No other actions are required, but if the deployment owner notices that the deployment is failing in high numbers he can cancel the deployment and start troubleshooting. This screening should not last for more than 24 hours when most of the installations for the deployed application will take place.

An application can also be deployed to Software Center. IT Administrators can list applications there which would have been previously pushed to certain user groups whom the application owner has selected if necessary. The application is then published in Software center for new users to freely download. This process is not always required, sometimes the application gets pushed to the user's computer first and then added to Software center.

5.4 Closing Phase

Closing the deployment means removing the active deployment and cleaning up any folders created in ConfigMgr to list a few examples. Defining when this is done is not so clear.

Closing the deployment should be dependent of two things, deployment status and the duration of the deployment. If the installation has finished on every computer, the deployment can be closed. Closing the deployment is the responsibility of the Deployment Owner. If the deployment has been on-going for two months it will be closed. Two months is long enough time so that every computer that is going to get the application, has already gotten the application and the rest that do not have the application, probably have some problems in their computers preventing the application from installing. After the deployment has been closed the computers where the installation failed or did not start have been detected so troubleshooting those computers may begin. Troubleshooting will be also done during the deployment.

5.5 Troubleshooting

Troubleshooting the issues with the application should be resolved together with the application owner by either local IT or the application management

team and problems with the deployment should be troubleshot by the deployment management team in collaboration with the application management team.

When troubleshooting deployment errors installation logs are extremely useful. They can tell why the installation failed and when. The Toolkit creates these installation logs to a file share which is set in the script. In the company it was *C:\Windows\Maintenance\Logs\Applications*. For each packaged application installation there will be an installation log that shows what the installer tried to do and at which point it failed, if it failed. For example, if the installation fails while the installer is uninstalling previous versions, that might mean that either the previous version, which is installed, cannot be uninstalled. Or that the previous version has been uninstalled, but some registry keys are still present which confuses the installer causing it to still think that previous version is installed.

ConfigMgr client also creates logs that help to troubleshoot issues with the deployment. ConfigMgr creates many different logs, but for application installations AppEnforce.log is the most important. AppEnforce.log contains detailed information about the installation and uninstallation of the package. Logs can be found from *C:\Windows\CCM\Logs*. Specific error codes and the location of the installer can be found from the AppEnforce log (Almeida, 2015).

getPointer()->Get(BString(pszName), 0, pvValue, &ctPropType, 0), HRESULT=80041002 (e:\nts_sccm_release\smis\framework\core\ccmcore\wmiobject.cpp,110)	AppEnforce	9.11.2016 16:56:40	4956 (0x135C)
Get(pszName, vValue, VT_BOOL), HRESULT=80041002 (e:\nts_sccm_release\smis\framework\core\ccmcore\wmiobject.cpp,514)	AppEnforce	9.11.2016 16:56:40	4956 (0x135C)
No reconnect data was found.	AppEnforce	9.11.2016 16:56:40	4956 (0x135C)
DeleteInstance(vValue.getBstr(), pContext), HRESULT=80041002 (e:\nts_sccm_release\smis\framework\core\ccmcore\wminamespace.cpp,855)	AppEnforce	9.11.2016 16:56:40	4956 (0x135C)
m_spNamespace.DeleteInstance(m_spEnforceStatus), HRESULT=80041002 (e:\nts_sccm_release\smis\framework\dcml\wmiprovider\appprovider\appenforcestatus.cpp,209)	AppEnforce	9.11.2016 16:56:40	4956 (0x135C)
Status is switching to Success	AppEnforce	9.11.2016 16:56:40	4956 (0x135C)
DeleteInstance(vValue.getBstr(), pContext), HRESULT=80041002 (e:\nts_sccm_release\smis\framework\core\ccmcore\wminamespace.cpp,855)	AppEnforce	9.11.2016 16:56:40	4956 (0x135C)
m_spNamespace.DeleteInstance(m_spEnforceStatus), HRESULT=80041002 (e:\nts_sccm_release\smis\framework\dcml\wmiprovider\appprovider\appenforcestatus.cpp,209)	AppEnforce	9.11.2016 16:56:40	4956 (0x135C)
Status is switching to Success	AppEnforce	9.11.2016 16:56:40	4956 (0x135C)
+++ Starting Install enforcement for App DT " " ApplicationDeliveryType - ScopeId, /DeploymentType, Revision - 4, ContentPath - C:\Windows\ccmcache\ck, Execution Context - S...	AppEnforce	9.11.2016 16:56:40	4956 (0x135C)
Cached CCM (user, session) is: ().	AppEnforce	9.11.2016 16:56:40	4956 (0x135C)
==GetSMSConsoleSessionId() - SMS Console Session ID is: 1	AppEnforce	9.11.2016 16:56:40	4956 (0x135C)
WTSQueryUserToken(2)(SessionId=1) retried 0 times!	AppEnforce	9.11.2016 16:56:40	4956 (0x135C)
A user is logged on to the system.	AppEnforce	9.11.2016 16:56:40	4956 (0x135C)
App is targeted to Machine. Execution context is System	AppEnforce	9.11.2016 16:56:40	4956 (0x135C)
Performing detection of app deployment type (ScopeId, /DeploymentType, Revision 4) for system.	AppEnforce	9.11.2016 16:56:40	4956 (0x135C)
Discovering app: AppDeliveryType: ScopeId, /DeploymentType: 624da2fc-7771-4762-af96-bb9b7ca810fa, Revision: 4	AppEnforce	9.11.2016 16:56:40	4956 (0x135C)
Expression Xml: <DiscoverySourceInfo SourceType="WMI"> <Resource LogicalNames="RegKey,"> <InstanceDescription Namespace="root\CCM\CM...	AppEnforce	9.11.2016 16:56:40	4956 (0x135C)
+++ Application not discovered. [AppDT Id: ScopeId, /DeploymentType, Revision: 4]	AppEnforce	9.11.2016 16:56:40	4956 (0x135C)
App enforcement environment: Context: MachineCommand line: "Deploy-Application.EXE" Allow user interaction: NoUI mode: 0 User token: null Session Id: 4294967295 Content path: C:\Windows\ccmcache\ck\Working directory:	AppEnforce	9.11.2016 16:56:40	4956 (0x135C)
Prepared working directory: C:\Windows\ccmcache\ck	AppEnforce	9.11.2016 16:56:40	4956 (0x135C)
Parsed CmdLine: "Deploy-Application.EXE"	AppEnforce	9.11.2016 16:56:40	4956 (0x135C)
Parsed CmdLine: "C:\Windows\ccmcache\ck\Deploy-Application.EXE"	AppEnforce	9.11.2016 16:56:40	4956 (0x135C)
Executing Command line: "C:\Windows\ccmcache\ck\Deploy-Application.EXE" with system context	AppEnforce	9.11.2016 16:56:40	4956 (0x135C)
Parsed CmdLine: "C:\Windows\ccmcache\ck\Deploy-Application.EXE"	AppEnforce	9.11.2016 16:56:40	4956 (0x135C)
Commandline length: 47 evaluate: 1	AppEnforce	9.11.2016 16:56:40	4956 (0x135C)
Working directory: C:\Windows\ccmcache\ck	AppEnforce	9.11.2016 16:56:40	4956 (0x135C)
Post-install behavior is BasedOnExitCode	AppEnforce	9.11.2016 16:56:40	4956 (0x135C)
Process creation time: 2016-11-09 15:56:40.342	AppEnforce	9.11.2016 16:56:40	4956 (0x135C)
Current time: 2016-11-09 15:56:40.358	AppEnforce	9.11.2016 16:56:40	4956 (0x135C)
Actual elapsed time (in seconds): 0	AppEnforce	9.11.2016 16:56:40	4956 (0x135C)
Reconnecting to - Process Id: 6696, MaxRunTime: 7200000	AppEnforce	9.11.2016 16:56:40	4956 (0x135C)
Waiting for process 6696 to finish. Timeout = 120 minutes.	AppEnforce	9.11.2016 16:56:40	4956 (0x135C)
Process 6696 terminated with exitcode: 1603	AppEnforce	9.11.2016 16:56:56	4956 (0x135C)
Loading for exit code 1603 in exit codes table.	AppEnforce	9.11.2016 16:56:56	4956 (0x135C)
Unmatched exit code 1603 is considered an execution failure.	AppEnforce	9.11.2016 16:56:56	4956 (0x135C)
Status is switching to Failure	AppEnforce	9.11.2016 16:56:56	4956 (0x135C)
Reading status 'Failure' from current status object.	AppEnforce	9.11.2016 16:56:56	4956 (0x135C)
+++++ App enforcement completed (16 seconds) for App DT " " [ScopeId, /DeploymentType, Revision: 4, User SID:] +++++	AppEnforce	9.11.2016 16:56:56	4956 (0x135C)

Figure 7: Example of the error 1603 found in AppEnforce.log.

Most common and hard to solve error code is 1603 which means a general error. There are multiple cases where error code 1603 will be given and there

is no clear way to determine which of those known cases caused the error. For example, the installation package has a requirement that Microsoft .Net version 4.6.0 is installed, but instead .Net 4.6.1 has been installed on the computer which results the installation failing on 1603. To determine that this was the cause of the error cannot be found from the logs, but by running the installer package manually.

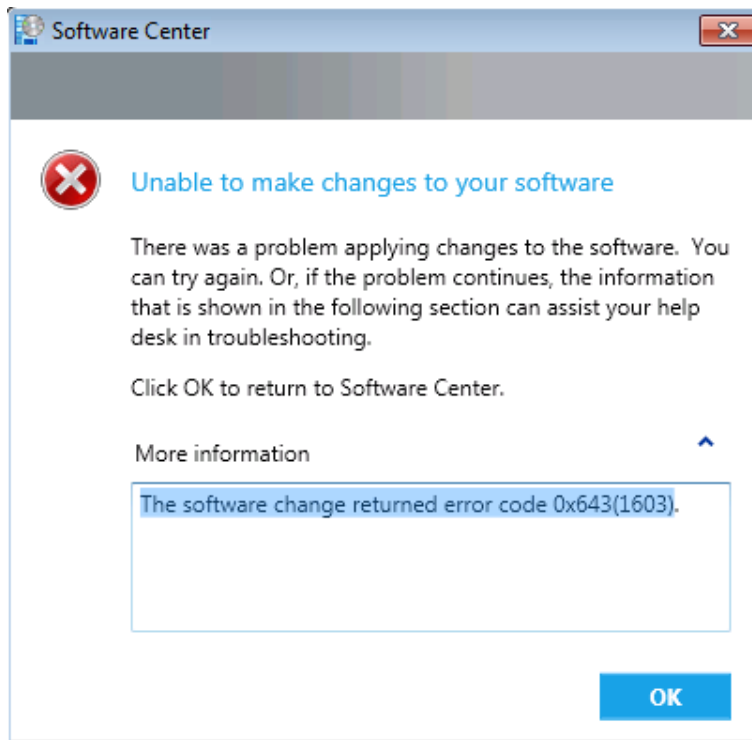


Figure 8: Example of failed installation due to error 1603.

To install the application manually first the location of the installation package must be determined. This can be found from the installation logs that the Toolkit creates during an installation process or from AppEnforce.log that the ConfigMgr client creates.

Log Text	Component
[Initialization] :: *****	PSAppDeployToolkit
[Initialization] :: *****	PSAppDeployToolkit
[Initialization] :: [] setup started.	PSAppDeployToolkit
[Initialization] :: Script [C:\WINDOWS\ccmcache\%0\AppDeployToolkit\AppDeployToolkitExtensions.ps1] dot-source invoked by [C:\WINDOWS\ccmcache\%0\AppDeployToolkit\AppDeployToo...	PSAppDeployToolkitEx
[Initialization] :: [Deploy Application] script version is [3.5.0]	PSAppDeployToolkit
[Initialization] :: The following non-default parameters were passed to [Deploy Application]: [-DeploymentType "Uninstall"]	PSAppDeployToolkit
[Initialization] :: [App Deploy Toolkit Main] script version is [3.5.0]	PSAppDeployToolkit
[Initialization] :: [App Deploy Toolkit Extensions] version is [1.5.0]	PSAppDeployToolkit
[Initialization] :: Computer Name is []	PSAppDeployToolkit
[Initialization] :: Current User is [NT AUTHORITY\SYSTEM]	PSAppDeployToolkit
[Initialization] :: OS Version is [Microsoft Windows 7 Enterprise Service Pack 1 64-bit 6.1.7601.65536]	PSAppDeployToolkit
[Initialization] :: OS Type is [Workstation]	PSAppDeployToolkit
[Initialization] :: Current Culture is [fr-FI] and UI language is [FI]	PSAppDeployToolkit
[Initialization] :: Hardware Platform is [Physical]	PSAppDeployToolkit
[Initialization] :: PowerShell Host is [ConsoleHost] with version [3.0]	PSAppDeployToolkit
[Initialization] :: PowerShell Version is [3.0 x64]	PSAppDeployToolkit
[Initialization] :: PowerShell CLR (.NET) version is [4.0.30319.42000]	PSAppDeployToolkit
[Initialization] :: *****	PSAppDeployToolkit

Figure 9: Example of installation log that the Toolbox creates.

When the installation package has been found, the installation should be launched. If the installer is a .msi-file, then there is usually a .mst-file with it. This .msi-file is the installer and the .mst-file is the configuration for the installer. For example, what language is selected during the installation. To launch the installer, the .msi-file has to be launched with the .mst-file with the command: `msiexec /i application.msi transforms=application.mst`. This command has to be run via the command line from the installation packages location. After the installer progresses the installer might show a clear message why the installation failed, like that it requires .Net version 4.6.0. This is just one example of many errors that error code 1603 might be a representation of, but listing them all would be impossible due to every application being different.

In the company the most common error codes are:

- 0 – Installation was successful.
- -2016410844 – Content was not found after installation.
- -2016411117– Timeout.
- 1603 – Fatal error during installation
- 0x87D01202 – Computer does not have enough disc space or the client cache is set below the requirements of the package.

Some of these errors will resolve themselves, like 2016410844 and 2016411117, but some of them need to be fixed by IT support, like 0x87D01202 and 1603.

5.6 Special Deployments

When planning a deployment for a high risk application, there should be at least two people who could dedicate their time for the troubleshooting of the possible problems with the deployment. Most of the installations will happen within 24 hours, but the application should be under surveillance for about a week after the initial installation of the application. One of the people monitoring the application installation should be the application owner so that there is always a person available with information about the application. The other person should be from either the application management team or deployment management team who can assist in problems relating to the deployment or the installation.

6 REPORTING

Reporting will be split in to two parts. One for Application owner and the other one for IT personnel. The report for application owner contains more visual feedback while the one for IT personnel contains information about errors, like error codes.

6.1 For Application Owners

The reports will be made using the default reports in ConfigMgr as a template and building the rest using Report Builder 3.0. Deployment report should include visual information of the deployment meaning the amount of installations, number of successes and errors. It should also be presented with visual diagrams instead of a rough excel chart. IT should always advertise itself to the users as it is there to support the users and help them with their daily work routine. When these reports would include easy to understand visual aspects it would then send a message to the users of a well-functioning IT department.

Preferable visual diagrams are pie chart and bars. Building them in Report Builder requires some knowledge of MySQL coding.

During planning it was decided that following information should be visible for the application owner:

- Application name and version.
- Divisions of the users.
- Users names.
- Computer names.
- Users location.
- Amount of installations for the deployed package.
- Amount of installations for previous version (if available).
- Amount of Errors in the deployment.
- Amount of computers where the installation is in progress.
- Amount of computers where the installation has not yet started.
- Name of the deployment.
- Time when the deployment has been started

By using these parameters, building of the report was possible.

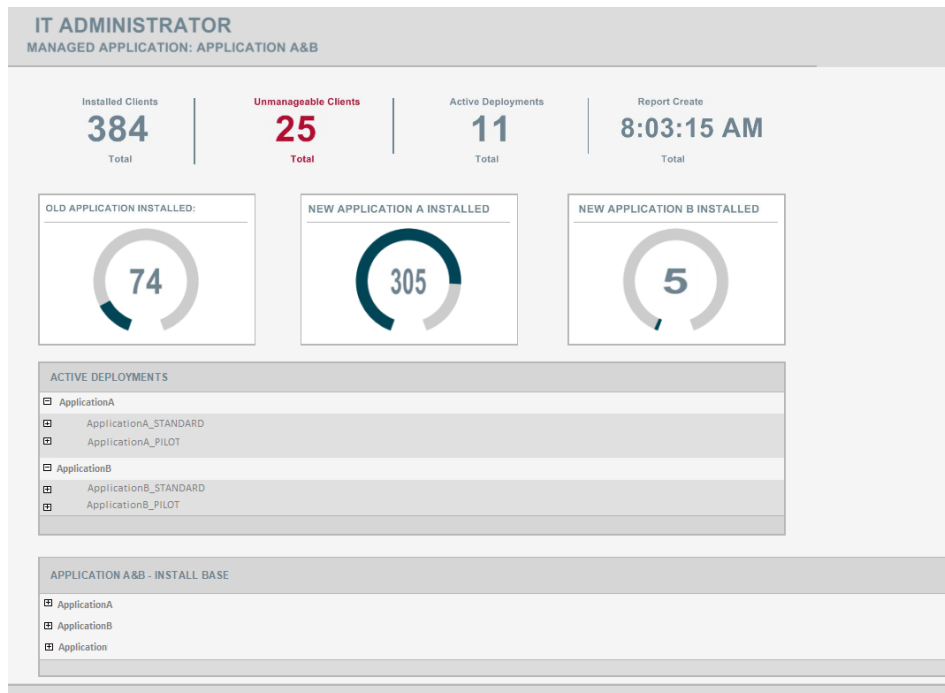


Figure 10: Example of the first Application Owner report

When asked for feedback from the application owners, the response was extremely positive:

“It was really useful to have the information about all current (old application) installations and then how many of them were installed with new application...”

“And it was really nice that I could see the user’s names and computers with the installations status.”

“Also disc space data was very handy when troubleshooting why the application was not installed.”

Only negative comment was about the quality of the data:

“The report showed that there were 350 machines that had the old version but I think there were some computers that are no longer in use so the amount is not correct.”

6.1.1 Building the Report

Building reports in the Report Builder 3.0 is a combination of SQL coding and understanding the basic functions of Report Builder 3.0. The first thing when beginning to use Report Builder is to connect to the correct SQL server which will be used when building the reports. After connecting building the reports can begin.

First setting that needs to be added is to set the Datasets. Datasets refer to the SQL database and by utilizing MySQL IT Administrator can search and combine tables in the SQL database to create reports that search the required information.

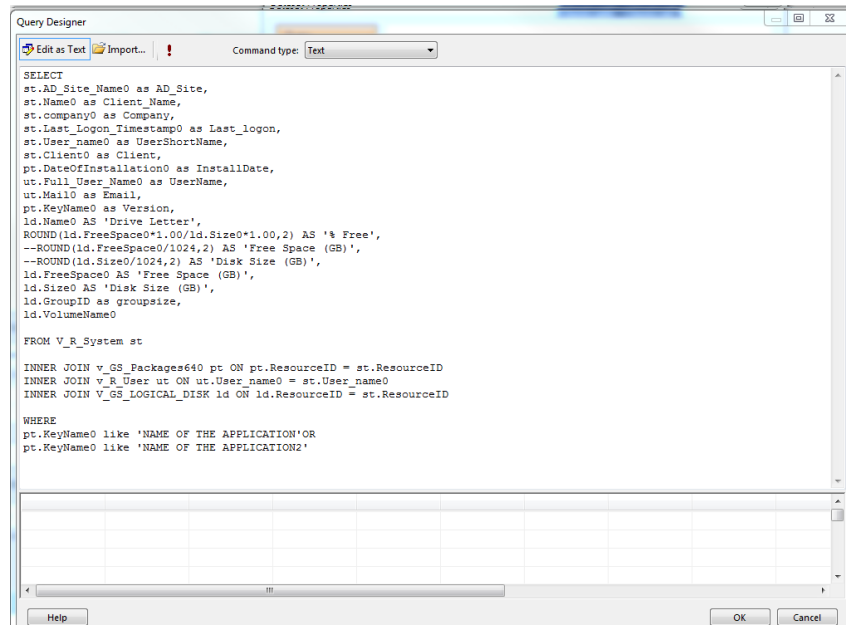


Figure 11: Example of the query used when creating manual reports.

Like shown in Figure 8 there are plenty of tables which are used. These Datasets often are a combination of multiple different tables in the SQL Database.

6.2 For IT Personnel

Application Deployment error reports for IT Personnel will be more technical and should contain the following information:

- Application name and version.
- Users names.
- Computer names.
- Users location.
- Amount of Errors in the deployment.
- Error codes.
- Amount of computers where the installation has not yet started.
- Name of the deployment.
- Time when the deployment has been started

Providing the knowledge on how to solve the different errors via referencing the error codes should be done by using a separate document where all of the most common errors and solutions are listed. This list originally created by IT

Administrative teams would be then maintained by the Local IT with Service Desk.

To answer the question, how the local IT knows what deployment errors are for their region some filtering must be done, otherwise it is very difficult to detect them. By utilizing the different deployment types, reports could be made with by first filtering all Standard deployments from then filtering certain region, for example Finland from Standard deployments. This way making a list of computers that have deployments errors is possible.

ConfigMgr collects large amounts of data from its client computers to its SQL database and from that database one table was very useful when planning the Local IT report. vAppDeploymentErrorAssetDetails table contains all error codes for all deployments in the ConfigMgr environment. Unfortunately, due to time constrains this thesis does not include Local IT report, but by utilizing these parameters and previous reports it is possible to do the reports on a later date.

7 CONCLUSION

Beginning to create this thesis was not easy. Application deployments are an integral part of any large enterprise, but finding information how this should be done was difficult. One of the reasons for this was the fact that each company is different and each of their application deployment environment in different. Microsoft provided good instructions about the basics, but more complex and defined methods were missing. There were no ready-made phases or steps of application deployment, therefore the only option was to gather the small amount of information that could be found on the internet and fill the rest based on the results of interviews and discussions with the IT Administrators. Before stating to deploy application centrally, they were installed manually by Local IT which still causes some issues. The problem is that users are in a rush to do their work and the easy and fast solution is to do what the users ask. Application deployment should be made easier and faster and more understandable for Local IT, because currently ConfigMgr is this entity that belongs to IT administrators and Local IT does not have any connectivity to it. First step would be explaining the problems manual installations can cause and provide them as much tools and help as possible so they can support users to the best of their abilities.

Reporting proved to be very complex. Microsoft has never released a proper list or a document about the SQL tables that ConfigMgr uses, so the only way to search for the correct data was manually checking each and every table. The benefit was that I learned much about many of the tables and during that search I found many other SQL tables that contain data that can be used later for different kind of reports.

World of IT is constantly changing and staying with the flow can be a difficult task for large companies. During my thesis a new way of reporting was released by Microsoft called PowerBI which is a much more visual and prettier way to build reports using cloud based storage. I believe that PowerBI will replace Report Builder in a couple of years due to its much simpler interface and better features. ConfigMgr is constantly getting updates, which improves its features in application deployment and asset management. In the company newest version of ConfigMgr is already in testing and it has some great life improvement features like the ability to check what computers are online and which are not. Work with updating and improving applications is never ending, and only by standardization and with proper management it is possible to stay on the edge of modern IT.

8 REFERENCES

- Almeida, G., 2015. *Geo'sblog*. [Online]
Available at: <http://www.georgealmeida.com/2015/03/the-software-change-returned-error-code-1603/>
[Accessed 10 November 2016].
- Hartikainen, A., 2016. APPLICATION REQUEST AND APPROVAL IN ENTERPRISE. In: *APPLICATION REQUEST AND APPROVAL IN ENTERPRISE*. Kotka: Theseus, pp. 53-54.
- Hartikainen, A., 2016. APPLICATION REQUEST AND APPROVAL IN ENTERPRISE. In: *APPLICATION REQUEST AND APPROVAL IN ENTERPRISE*. Kotka: Theseus, pp. 19-21.
- Lee, T., 2010. *Step-by-Step Guide to Managing Multiple Local Group Policy Objects*. [Online]
Available at: <https://technet.microsoft.com/fi-FI/library/9c7ecc7d-8784-4b8d-ba1f-ba1882ba8374>
[Accessed 14 November 2016].
- Lillis, S., 2013. *PowerShell App Deployment Toolkit*. [Online]
Available at: <http://psappdeploytoolkit.com/>
[Accessed 9 November 2016].
- Microsoft, 2015. *Technical Reference for Log Files in Configuration Manager*. [Online]
Available at: https://technet.microsoft.com/en-us/library/hh427342.aspx#BKMK_AppManageLog
[Accessed 10 November 2016].
- Microsoft, 2016. *Active Directory*. [Online]
Available at: <https://msdn.microsoft.com/en-us/library/bb742424.aspx>
[Accessed 20 November 2016].
- Microsoft, 2016. *Microsoft*. [Online]
Available at: <https://www.microsoft.com/en-us/download/details.aspx?id=6116>
[Accessed 15 November 2016].
- Microsoft, 2016. *Microsoft System Center*. [Online]
Available at: [https://technet.microsoft.com/en-us/library/hh420338\(v=sc.12\).aspx](https://technet.microsoft.com/en-us/library/hh420338(v=sc.12).aspx)
[Accessed 15 November 2016].
- Microsoft, 2016. *Orchestrator Capabilities*. [Online]
Available at: <https://technet.microsoft.com/en-us/library/hh420338%28v=sc.12%29.aspx>
[Accessed 18 November 2016].
- Microsoft, 2016. *System Center Configuration Manager and Microsoft Intune*. [Online]
Available at: http://download.microsoft.com/download/5/D/B/5DBEBA38-8D5D-4119-B2E8-B8369B74BF43/system_center_configuration_manager_and_microsoft_intune_datasheet.pdf
[Accessed 18 November 2016].

Rouse, M., 2013. *Microsoft System Center Orchestrator*. [Online]
Available at:
<http://searchservvirtualization.techtarget.com/definition/Microsoft-System-Center-Orchestrator>
[Accessed 18 November 2016].

Siemens AG, 2010. *What is Application Management*. [Online]
Available at: <http://www.slideshare.net/ApplicationManagement/what-is-application-management>
[Accessed 18 November 2016].

Singh, G., 2016. *How IT strategy supports business growth*. [Online]
Available at: <http://www.examiner.com/article/how-it-strategy-supports-business-growth>
[Accessed 16 October 2016].

techopedia, 2016. <https://www.techopedia.com/definition/28008/application-management-am>. [Online]
Available at: <https://www.techopedia.com/definition/28008/application-management-am>
[Accessed 19 November 2016].

Zitek, N., 2014. *ITIL Application Management Lifecycle - within IT Service Lifecycle*. [Online]
Available at: <http://advisera.com/20000academy/blog/2014/03/18/itil-application-management-lifecycle-within-service-lifecycle/>
[Accessed 18 November 2016].

