

# Thoughts on Server Provisioning, Management and Patching

Jan 24, 2008

---

## Table of Contents

<b><i>1 Introduction.....</i></b>	<b><i>3</i></b>
<b><i>2 Definitions.....</i></b>	<b><i>4</i></b>
<b><i>3 Analogy.....</i></b>	<b><i>5</i></b>
<b><i>4 Prerequisites.....</i></b>	<b><i>6</i></b>
<b><i>5 Opportunities.....</i></b>	<b><i>8</i></b>
<b><i>6 Dangers.....</i></b>	<b><i>10</i></b>
<b><i>7 Conclusion.....</i></b>	<b><i>12</i></b>

---

# 1 Introduction

## *Overview*

The ability to quickly and efficiently provision, manage and patch computer systems is important for any business that exists in a dynamic business environment. There are many solutions on the market that promise healthy returns for any business that agrees to deploy their particular flavor. In the author's experience, a new Provisioning, Management, and Patching (PMP) system is not an easy thing to implement, and is certainly not a drop-in and go solution. The background work that needs to be done for a successful rollout is extensive and should not be overlooked or minimized.

This paper attempts to identify major road blocks in the successful deployment of a PMP solution. It is written in the context of an organization already deeply involved in the roll out of such a solution. The project is currently experiencing considerable churn, missed deadlines and frequent fire drills. In the opinion of the author, many of these are a result of not considering the impacts of the issues in this document. If these issues are not successfully addressed, then even if the roll out 'succeeds' it will be a failure for the firm because the firm will not realize the cost savings and efficiencies that would be possible by addressing the issues.

---

## 2 Definitions

The following definitions will be used in this document

### ***Provisioning***

The process of taking a server which is currently in a data center and

- configuring any out of band management capabilities
- configuring the disk drives
- installing and configuring an operating system on it
- installing and configuring corporate standard software (monitoring, backup, security, etc)
- installing server specific applications and configurations
- registering all appropriate agents, etc
- installing an appropriate management agent

The provisioning may be in the context of a new server, of an additional server in an application cluster, of a replacement server for either a standalone application or for a cluster or a redeployment of an existing server. It should support both production and development varieties of servers.

### ***Management***

The ability to monitor and manage all appropriate components of the server from a central location including

- Hardware and performance monitoring
- OS and application configuration

### ***Patching***

The process to apply approved software patches or updates to a server

- In an approved time frame (service window)
- With appropriate logging
- With minimal application down time
- With the ability to undo an update
- The ability to apply appropriate firmware updates to the hardware

---

## 3 Analogy

To help in framing the discussions contained in this document, the following analogy will be used.

### ***Server factory***

The servers in a business without a good PMP solution will probably look very much like the goods available in a bazaar that caters to a specific type of goods (such as carpets). All of the carpets will be made from the same or similar material, and will perform the same basic functions. Each merchant's goods will be of different quality, different cost and will yield different returns to the merchants and to their customers. Customers will have to shop carefully to find the goods that meet their needs. It will be difficult to repair or replace an item that has been damaged or lost.

The servers in a business with a good PMP solution will look like the goods from a well run factory. The materials used, quality, functions and returns to the merchant and customer will be consistent. Shopping will be pointless, since all the goods are the same. Repair and replacement will be simple because the merchant knows how the item was constructed and can duplicate the process.

For purposes of a PMP solution, we will define the following groups of people for our analogy:

- Merchants - the central engineering team that develops the builds
- Suppliers: hardware and software manufacturers and developers
- Customers: deployment teams, support teams, application teams, end users

### ***Planning and standards***

A necessary prerequisite for a factory is to have some degree of planning and standards setting mechanism. The amount of centralization will vary with the organization, but there must be a mechanism to set, publicize, monitor and enforce the use of standards. For every standard that is violated, or for which there exists an 'exemption' some amount of the benefits of the factory are lost.

### ***Handling customization***

Customization is almost always necessary, but it needs to be driven as far towards the end of the process as possible and should be clearly delineated from the base product. It should be the last decision, not the first. A good analogy is that of the automotive industry. 'Customizations' that the factory is responsible for are actually just selections of standards that will be applied to thousands of vehicles. The customizations that the owners apply are almost all discrete, localized and clearly separate from the vehicle itself (the contents of the trunk, the passengers, the air freshener, etc). If the vehicle is damaged or replaced, the customizations are easily identified and moved to a new vehicle.

---

## 4 Prerequisites

Naturally there are prerequisites to gain the full advantage of a PMP system. **For each of the prerequisites that is not met or obtained, some portion of the investment in the system is squandered.** Without these pieces in place, a PMP system will never deliver any of the expected benefits. To get the full advantage from our server 'factory' or PMP system, the key is standards. By 'standards' we mean the entire process of collecting requirements, gaining consensus, communicating with stake holders, communicating with users, implementing the standards in the PMP system and providing a feedback loop to modify the standards. Standards processes need to be developed for the following areas

### ***Software Package Standards***

Software packages are the nuts and bolts of the server 'product'. Particular attention needs to be paid to:

- Naming conventions: Package names typically include the versions. The versions need to be specified in a way that is machine parsable and sortable so the PMP system can programmatically determine which is 'newer'.
- Development environment and process: packages need to be developed in an appropriate and documented environment. This will reduce the possibility of conflicts between packages and simplify maintenance and patching of the packages.
- Documentation: Each package needs to contain a standard document set including a contact name, lists of requirements or exclusions and basic purpose and usage information.
- Usage: Packages need to conform to standard usage guidelines as to where scripts are stored, how commands are named, etc.

### ***Hardware Standards***

Hardware (servers) is the foundation that the entire product is built upon.

- Platform: Better specification of the physical characteristics (CPU's, RAM, disks, add-in card locations, out of band management) allows greater integration with the PMP system.
- Usages: Platforms need to be selected and slotted for specific use cases (web server, small database, large database, general purpose). Customers need to be strongly encouraged to pick the appropriate server for their use. In an ideal situation, all applications would be load tested before the final servers are selected to ensure the correct platforms are used.
- Documentation: What is each platform capable of? How is the server physically configured and installed?

### ***Management standards***

- Who is responsible for each step?
- What are the appropriate actions for each phase of implementation?
- How often are standards updated, what are the SLA's involved?
- How do issues get escalated?
- How are cross team issues resolved?

### ***Buy in from each level of management involved***

All levels of management need to be engaged in the design and rollout of the PMP system. Realistic discussions need to occur about resource usage, resource reduction and resource gain. If a PMP system is correctly implemented, it would not be unusual to see the numbers of back end engineers grow (more standards work to be done) and the number of installation and systems administration personnel reduced (since a well implemented PMP system will allow each person to work more efficiently). It may be neces-

---

---

sary to shift responsibility for particular functions from one team to another in order to streamline the communications and accountability.

- Business management: How do the processes of managing applications change? Of installing new servers? What changes need to occur in the project process?
- Engineering management: How do standards get created, reviewed, updated, tracked? How is communication with the customers handled?
- Development management: How is the development process impacted? Development environments will need to be documented and normalized so the corresponding production standards can be put into place.

### ***Infrastructure standards***

- All the data centers the PMP system is deployed to should use the same network topology, or the differences must be clearly understood
- The interface to 'supporting services' such as ntp, mail, DNS, DHCP and network traffic rules should be standard across all the data centers. Ie, every datacenter has ntp.datacenterX.net, mail.datacenterX.net, dns.datacenterX.net, etc
- A server naming standard is in place and enforced

---

## 5 Opportunities

Implementation of a PMP system is usually done in order to take advantage of one or more cost savings opportunities that a central system can offer. To the extent that standards are not developed and followed, some portion of each of these opportunities will be squandered. Examples include

### ***Provide a standard interface across operating systems***

Potentially reducing training costs for support personnel

### ***Driving standards adoption***

Customers can be presented with the opportunity to adhere to standards. Those managing the PMP system should be able to provide metrics from the current environments to demonstrate the advantages.

### ***Increase the server per employer ratio***

Due to increased use of standards, each administrator should be able to manage more systems.

### ***Increase security***

The job of determining what security updates should be applied is much easier since each system is based on a standard, and all changes to the systems are controlled from and stored in a central location. The existence of clear lines of communications (a side effect of the standards work) should help streamline the decision making and approval process. The entire system should enable security issues to be remediated in a much more timely manner, with a much lower risk of unintended side effects.

### ***Increase reliability***

Reliability will rise because the amount of direct interaction with each system by system administrators 'working with out a net' will decline. In order for the PMP system to deliver the benefits, server configuration should move from the command line (where each admin is free to apply his own techniques) to a scripted solution which can be tested on representative machines in a lab environment before production deployment.

### ***Decrease the need for other tools***

As the PMP system is deployed, much of the information collected by various monitoring solutions will become resident in the PMP system. Package versions, configuration details, user access and similar data will already be in the database. More traditional items such as logs and performance data may also be collected, depending on the PMP solution. In most organizations, each of these functions will be performed by a separate agent (or by multiple agents) with each agent incurring licensing, support, personnel and performance costs.

### ***Reduce lab costs***

While a PMP system requires more testing, it should also enable an overall reduction in lab costs. As systems are standardized within the capabilities of the PMP system, it becomes easier to deploy a system

---

into a lab environment for one off testing. It should be possible to 'clone' any fully managed production system (sans data). A representative sample of data can be added for testing purposes if needed.

---

## 6 Dangers

As with any complex system, there are also dangers inherent within the use of a PMP system. The following are those which seem obvious to the author.

### ***Opacity***

The data needed for a PMP system to function correctly will most likely be stored in a proprietary format within a data base. If the decision is made to move away from a particular PMP system, it may prove difficult or impossible to extract the data in a useable format.

### ***Knowledge loss***

Collecting all the data to fully manage an enterprise's systems into a single system opens the door to the employees not knowing how to do tasks themselves. As more and more configurations and management is done via the PMP system there will be fewer and fewer people who actually understand what is being done. This may cause issues when decisions to replace systems are undertaken, or when the PMP system is retired or replaced.

### ***Unable to leverage for full benefit***

Successfully implementing a PMP system definitely falls into the 'non trivial' task bucket. It is highly likely that some of the prerequisites will not be met. To the extent that they are not met, the organization will lose benefits from the PMP system. Enough loss of functionality and the implementation becomes a net loss to the organization. This is particularly true if the attempt to implement the system has caused talent flight from the ranks of employees.

### ***Class break***

While all PMP systems tout their security, privilege management, and auditing capabilities, such a system presents a single point of attack. By nature of the tool, PMP systems must have complete control of the systems they are managing. If the system is compromised, then a 'class break' has occurred and every system in the organization must be considered compromised. This obviously applies to the case of a disgruntled employee as well.

### ***System Collapse/Performance***

Each organization provides unique challenges to how they intend to use a PMP system, the types of loads that will be generated, etc. It is entirely possible for the assumptions of the PMP designers to run aground on the realities of a particular organization's implementation. Depending on when this occurs, it may be difficult to implement a fall back plan, since the employees are no longer available or no longer have the access and knowledge which was transferred to the system.

### ***Infrastructure costs***

It is difficult to capture all the infrastructure costs for a PMP system upfront. Security policies or network topologies may require additional servers. Load distribution may vary greatly from the initial assumption. Services that are required may not be available. Almost by default PMP systems appeal to organizations that do not really understand the infrastructure across the entire environment.

---

***Change in assumptions/Mismatch in policy***

PMP systems are typically selected by fairly small groups of technologists who may not even be able to identify their assumptions. For instance, PMP systems are 'push' technology; perhaps the organization's networks and infrastructure are configured for a 'pull' technology. Assumptions about the willingness of the various 'customers' to accept the new order of things will almost certainly be wrong.

---

## 7 Conclusion

A properly implemented PMP system can be a great benefit to a company. An improperly implemented one is just another pile of bits waiting to rot and turn into a little fiefdom that will be defended til the end of time. In doing research the following quote was found on an email list. It seems an appropriate conclusion to this paper and a warning to the unwary.

We've got around 10 different homegrown, and incomplete, Configuration/Release/Inventory management tools. <product> was supposed to replace them all, but now it's just #11 on the list of management tools.