MDP LCFG

Kenny MacDonald
Desktop Services, IS

Introduction

- Managed Desktop Project History
- MS Windows XP Pro, Apple MacOS X & Linux
- Devolved Management Model
- LCFG: Simplify and Devolve Access
- Scalability and Availability
- Future Plans

XP Managed Desktop Project History

- September 2002 - Desktop Services Team formed
- October 2003 - MS Windows XP available in UoER
- May 2004 - Devolution of OUs in UoEM
- July 2004 - All DST managed computers in UoEM
- September 2004 - Some Schools move to UoEM
- July 2005 - MIS deploy 1700 desktops in UoEM
- Spring 2006 - DST “deprecate” UoER

Linux MDP History

- Autumn 2004 – DST asked to deploy Linux in Open Access Labs
- Summer 2005 - Several hundred dual boot XP/LCFG FC3 computers in Open Access Labs
    - Greatly reduced packagelist
    - New “gconf” and “ffox”LCFG components
    - Custom “logon script” PAM module
    - PIE enhanced to build LCFG clients and configure GRUB for dual boot
- No devolution – not supportable beyond DST

Apple MacOS X MDP

- Add lcfg/os/macosx.h header file
- Include required defaults/<component>.h files from the OS headers, rather than from defaults.h
- Start “macboot” component asynchronously at startup
- Add “ipfw”, “launchd”, “macpkg”, “ffox” and “mdpauth” components

Standard Environments

- Create a tree of standard environments
- Devolve access below the standard MDP environment - ~40 devolved “units”
- Each inherits from above and becomes more specialised towards leaves
- Allow computers to be located in one of these environments
- Provide mechanisms to further customise each computer

Devolved Management in Active Directory

- MD Managers
  - Create OU structure and link GPOs to maintain standard environments. Create new options.
- MD Support Staff
  - Full control over individual client configuration via location in OU structure, or fine tuning options by security group membership. Local administrative rights on clients.
- MD Application Selectors
  - Only able to toggle application of individual options by security group membership.

Devolved Management in MDP LCFG

- MD Managers
  - Create directory structure and include options to maintain standard environments. Create new options.
- MD Support Staff
  - Full control over individual client configuration via location in directory structure, or fine tuning options by editing source profile. Local administrative rights on clients.
- MD Application Selectors
  - Not applicable.

Nomenclature

- Directories, Folders, OUs
- Computers, Sources, Profiles
- Group Policies, Options, Policies, Settings, Headers
- Linking, Including, Header, Footer, Forced
- SOM Filtering, Macro Defining
- Compiling, Generating

Policies Repository

- Write access for MD Managers
- Stores definitions of local standard environments
- Stores locally created and maintained options
- Only reserved top level directories allowed
  - “computers”
  - “defaults”
  - “hw”
  - “options”
  - “os”

Policies - “computers”

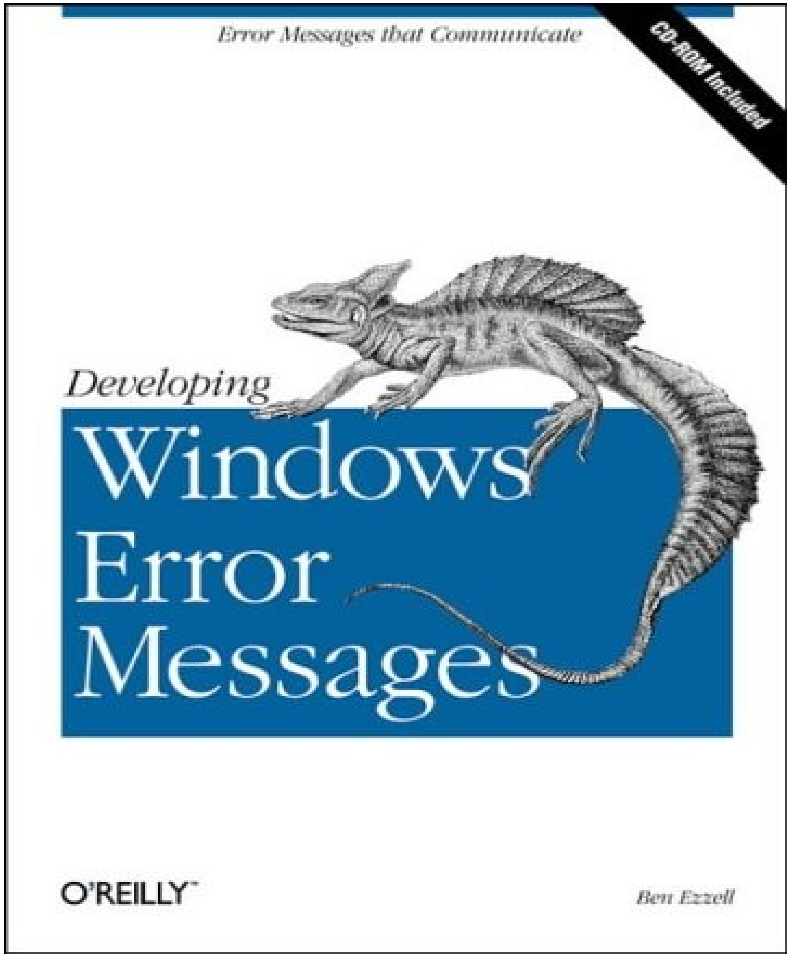
- Subdirectories mirrored automatically in the computers repository
- Only “define.h”, “header.h” and “footer.h” files allowed in each directory
- Only cpp directives allowed
  - includes (links), comments and macro definitions
- No including files outside an “options” directory

Computers Repository

- Write access to MD Support Staff
- Stores source profiles
- Directory structure maintained in the policies repository
- Only valid node names allowed
  - Unique, NetBIOS rules, lower case, no spaces

Subversion Hook “svn-agent”

- Pre-commit
  - Enforce policies
    - File Names
    - Permitted operations
    - Commit message
- Post-commit
  - Maintain rsync config data for LCFG server
    - Create generated source profiles
    - Create empty “define.h”, “header.h”, “footer.h” files
  - Maintain directories in computers repository
  - Send email notifications and update RSS feeds



Error Messages that Communicate
CD-ROM Included
Developing
Windows
Error
Messages
O'REILLY
Ben Ezzell

Example Computer

```
#define MDP_OS macosx
```

Example Generated Profile

```
/*** MDP Define BEGIN ****/
#include <mdp/define.h>
#include <foo/define.h>
/*** MDP Define END ****/
#define MDP_OS macosx
#include <mdp/os/macosx.h>
/*** MDP Headers BEGIN ****/
#include <mdp/header.h>
#include <foo/header.h>
/*** MDP Headers END ****/
#define MDP_OS macosx
/*** MDP Footers BEGIN ***/
#include <foo/footer.h>
#include <mdp/footer.h>
/*** Mandatory profile.group setting for the server status page ***/
!profile.group mSET (foo)
/*** MDP Footers END ****/
```

Configuration Server

- Stores all subversion repositories
- Shell and subversion access via ssh public keys
- Serves generated profiles and policies via rsync to slave servers
- EASE protected WebSVN
- Public RSS feeds

Slave Server

- Shell access via ssh public keys
- Fetches generated profiles and policies via rsync from configuration server
- Compiles and serves profiles to clients
- EASE protected LCFG CGI pages

Scalability

- Approx 1 – 2 seconds to compile a profile
- 3000 profiles could take up to two hours to compile
- Devolved units can have their own slave server
- Add slave server to clients' profile.url resource
- Customise slave server's server.srcpath resource
- Everyone's policies and computers available, but only required ones compiled
- Central slave server required for analysis

Availability

- All servers fully managed and easily rebuilt
- All clients continue to operate as normal while their slave server is unavailable
- Any client can be promoted to a server at any time
- Any slave server can be reconfigured to serve all clients
- Temporary DNS alias to allow clients to fetch profiles from temporary slave server

Future Work

- Get changes to OS header files upstream
- Add web interface to the server's dependency cache
- Don't allow used header files to be deleted
- Turn “svn-agent” into an LCFG component
- Bind Linux clients to AD
- Use XEN to build a slave server for each devolved unit
- EdLAN DB integration with computers repositories