



Group Policy Client Side Extensions

Custom Extensions for Samba Clients

David Mulder
Software Engineer
SUSE/dmulder@suse.com

Progress Since Last SambaXP

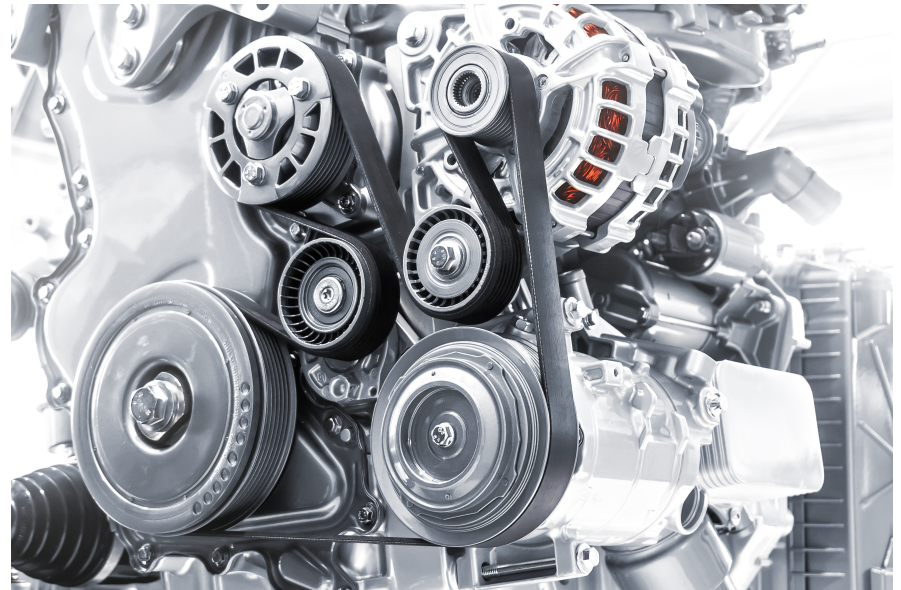
- KDC policy in 4.8
- GPO Creation via YaST Module
- Client Machine and User Policies
- No policy tattooing
- Offline apply of policy
- Dynamic CSE's via registry key



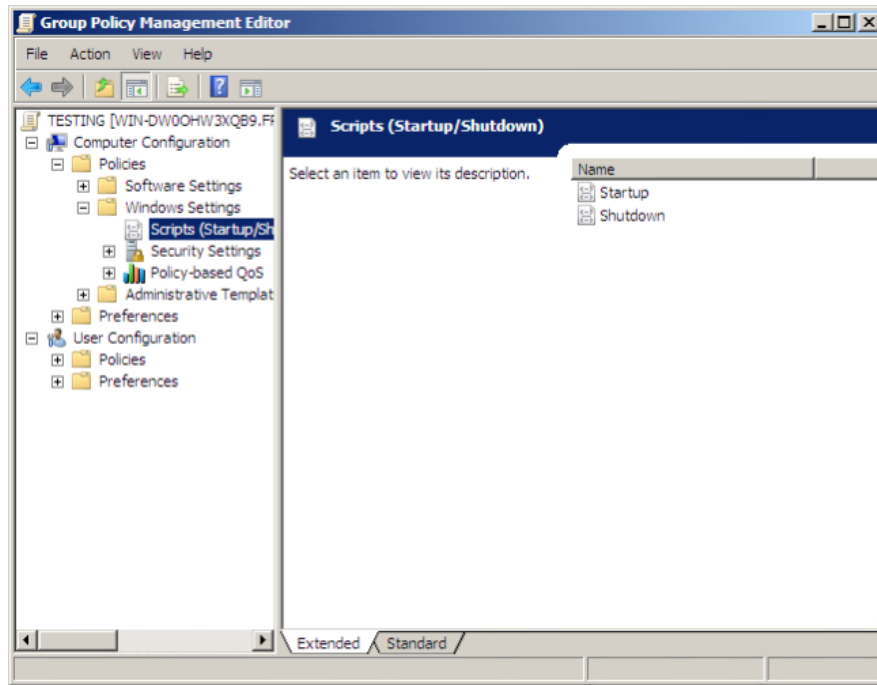
Client Side Extensions

What is a Client Side Extension?

Group Policy Client Side Extensions control the application of user and machine policies on a Windows client.



Where do the policies come from?



**Policies originate from the
Group Policy Management
Console**

Example of proxy User Policy



A screenshot of a terminal window titled "dmulder@linux-hufw:~". The window has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The terminal content shows a red prompt "WCL107:~" followed by the command "# ssh FROGGY\\administrator@localhost". A vertical cursor is positioned at the end of the command line.

```
dmulder@linux-hufw:~  
File Edit View Search Terminal Help  
WCL107:~ # ssh FROGGY\\administrator@localhost  
|
```

How a GP CSE works

```
class gp_ext(object):
    __metaclass__ = ABCMeta

    def __init__(self, logger, lp, creds, store):
        self.logger = logger
        self.lp = lp
        self.creds = creds
        self.gp_db = store.get_gplog(creds.get_username())

    @abstractmethod
    def process_group_policy(self, deleted_gpo_list, changed_gpo_list):
        pass

    @abstractmethod
    def read(self, policy):
        pass

    def parse(self, afile):
        # Generic function that correctly reads a file from the sysvol

    @abstractmethod
    def __str__(self):
        pass
```

```
class gp_inf_ext(gp_ext):
    @abstractmethod
    def process_group_policy(self, deleted_gpo_list,
        changed_gpo_list):
        pass

    def read(self, policy):
        inf_conf = ConfigParser()
        inf_conf.optionxform=str
        try:
            inf_conf.readfp(StringIO(policy))
        except:
            inf_conf.readfp(StringIO(policy.decode('utf-16')))
        return inf_conf

    @abstractmethod
    def __str__(self):
        pass
```


Installing via Registry

See:

[https://msdn.microsoft.com/en-us/library/aa373494\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/aa373494(v=vs.85).aspx)

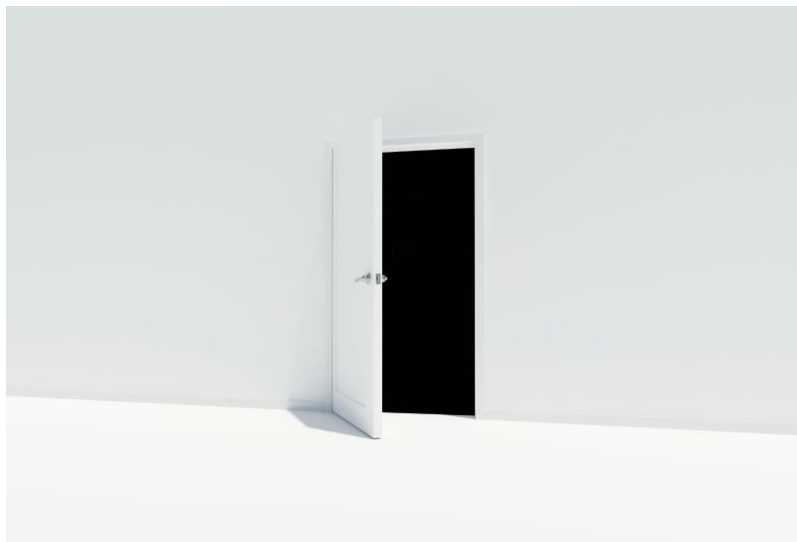
Stored under a unique guid in:

**HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Winlogon\GPExtensions**

Value	Meaning
DllName	A REG_EXPAND_SZ value that specifies the path to the DLL that contains the callback function.
NoMachinePolicy	If this value is 1, do not call the callback function when computer policy is processed.
NoUserPolicy	If this value is 1, do not call the callback function when user policy is processed.
ProcessGroupPolicy	A REG_SZ value that specifies the name of the application-defined callback function. Be aware that this value has been superseded by the value.

Let's try it out!

```
git clone https://github.com/dmulder/example-gp-ext.git  
cd example-gp-ext
```



In order to run it, you'll need:

```
git clone https://github.com/dmulder/samba.git  
cd samba  
git checkout gpo_extensible_030518
```


Questions?

