

# Samba's New Registry Based Configuration

Michael Adam

[obnox@samba.org](mailto:obnox@samba.org)

Samba Team / SerNet

2008-10-10



SerNet

# Outline

- 1 Configuration in Samba 3.0 (and before)
  - What We Have
  - What We Need Beyond That
- 2 Configuration in Samba 3.2
  - The Registry
  - Storing Configuration in Registry
  - The `net conf` Utility
  - The `libsmbconf` Library
- 3 Configuring Clustered Samba
  - Clustered Samba
  - Configuring Clustered Samba via Registry
- 4 Current and Ongoing Work
  - Rewrite of `loadparm`
  - Plans / TODOs



SerNet

# Outline

- 1 Configuration in Samba 3.0 (and before)
  - What We Have
  - What We Need Beyond That
- 2 Configuration in Samba 3.2
  - The Registry
  - Storing Configuration in Registry
  - The `net conf` Utility
  - The `libsmbconf` Library
- 3 Configuring Clustered Samba
  - Clustered Samba
  - Configuring Clustered Samba via Registry
- 4 Current and Ongoing Work
  - Rewrite of `loadparm`
  - Plans / TODOs



SerNet

# Configuration in Samba 3.0

- text based configuration in win.ini style: smb.conf file
- modularized by include directives
- dynamic character by macro expansion at runtime
- default config file can be changed by "config file = ..."
- main module for processing configuration: param/loadparm.c,  
text file parser in param/params.c



SerNet

# Configuration in Samba 3.0

- text based configuration in win.ini style: smb.conf file
- modularized by include directives
- dynamic character by macro expansion at runtime
- default config file can be changed by "config file = ..."
- main module for processing configuration: param/loadparm.c,  
text file parser in param/params.c



SerNet

# Configuration in Samba 3.0

- text based configuration in win.ini style: smb.conf file
- modularized by include directives
- dynamic character by macro expansion at runtime
- default config file can be changed by "config file = ..."
- main module for processing configuration: param/loadparm.c,  
text file parser in param/params.c



SerNet

# Configuration in Samba 3.0

- text based configuration in win.ini style: smb.conf file
- modularized by include directives
- **dynamic character by macro expansion at runtime**
- default config file can be changed by "config file = ..."
- main module for processing configuration: param/loadparm.c,  
text file parser in param/params.c



SerNet

# Configuration in Samba 3.0

- text based configuration in win.ini style: smb.conf file
- modularized by include directives
- dynamic character by macro expansion at runtime
- default config file can be changed by "config file = ..."
- main module for processing configuration: param/loadparm.c,  
text file parser in param/params.c



SerNet



# Configuration in Samba 3.0

- text based configuration in win.ini style: smb.conf file
- modularized by include directives
- dynamic character by macro expansion at runtime
- default config file can be changed by "config file = ..."
- main module for processing configuration: param/loadparm.c,  
text file parser in param/params.c



SerNet

# Configuration in Samba 3.0

- text based configuration in win.ini style: smb.conf file
- modularized by include directives
- dynamic character by macro expansion at runtime
- default config file can be changed by "config file = ..."
- main module for processing configuration: param/loadparm.c, text file parser in param/params.c



SerNet

## example smb.conf

```
[global]
    netbios name = NIRVANA
    workgroup = sambaXP
    security = user
    include = /etc/samba/smb.conf.%I

[share1]
    path = /data/shares/share1
    guest ok = yes
    read only = no
```

## include file smb.conf.192.168.1.2

```
log file = /var/log/log.samba.%u
log level = 10
debug hires timestamp = yes
```

## example smb.conf

[global]

netbios name = NIRVANA

workgroup = sambaXP

security = user

include = /etc/samba/smb.conf.%I

[share1]

path = /data/shares/share1

guest ok = yes

read only = no

include file smb.conf.192.168.1.2

log file = /var/log/log.samba.%u

log level = 10

debug hires timestamp = yes

## example smb.conf

```
[global]
    netbios name = NIRVANA
    workgroup = sambaXP
    security = user
    include = /etc/samba/smb.conf.%I

[share1]
    path = /data/shares/share1
    guest ok = yes
    read only = no
```

## include file smb.conf.192.168.1.2

```
log file = /var/log/log.samba.%u
log level = 10
debug hires timestamp = yes
```

## example smb.conf

```
[global]
    netbios name = NIRVANA
    workgroup = sambaXP
    security = user
    include = /etc/samba/smb.conf.%I

[share1]
    path = /data/shares/share1
    guest ok = yes
    read only = no
```

## include file smb.conf.192.168.1.2

```
log file = /var/log/log.samba.%u
log level = 10
debug hires timestamp = yes
```

## example smb.conf

```
[global]
    netbios name = NIRVANA
    workgroup = sambaXP
    security = user
    include = /etc/samba/smb.conf.%I

[share1]
    path = /data/shares/share1
    guest ok = yes
    read only = no
```

## include file smb.conf.192.168.1.2

```
log file = /var/log/log.samba.%u
log level = 10
debug hires timestamp = yes
```

# Disadvantages/Needs

- memory consumption (in large environments)
  - `smbd` process builds list of all service structures
  - each service struct roughly 1KB (150 parameters)
  - $\Rightarrow$  for 1000 shares and 1000 connections 1GB of wasted RAM
- I/O-bottleneck
  - `smbd` always loads *the whole* `smb.conf`
  - `smb.conf` file with 1000 shares typically 250KB
  - imagine 1000 clients reloading `smb.conf` ...
- ease of use
  - programmatically change the configuration
  - change individual parameters
  - write configuration GUIs
  - distribute configuration in clusters
- Design: parsing parameters and activating them is not well separated but is performed in one run.



SerNet



# Disadvantages/Needs

- **memory consumption (in large environments)**
  - smbd process builds list of all service structures
  - each service struct roughly 1KB (150 parameters)
  - ⇒ for 1000 shares and 1000 connections 1GB of wasted RAM
- I/O-bottleneck
  - smbd always loads *the whole* smb.conf
  - smb.conf file with 1000 shares typically 250KB
  - imagine 1000 clients reloading smb.conf ...
- ease of use
  - programmatically change the configuration
  - change individual parameters
  - write configuration GUIs
  - distribute configuration in clusters
- Design: parsing parameters and activating them is not well separated but is performed in one run.



SerNet

# Disadvantages/Needs

- memory consumption (in large environments)
  - **smbd process builds list of all service structures**
  - each service struct roughly 1KB (150 parameters)
  - ⇒ for 1000 shares and 1000 connections 1GB of wasted RAM
- I/O-bottleneck
  - *smbd always loads the whole smb.conf*
  - *smb.conf file with 1000 shares typically 250KB*
  - *imagine 1000 clients reloading smb.conf ...*
- ease of use
  - *programmatically change the configuration*
  - *change individual parameters*
  - *write configuration GUIs*
  - *distribute configuration in clusters*
- Design: parsing parameters and activating them is not well separated but is performed in one run.



SerNet

# Disadvantages/Needs

- memory consumption (in large environments)
  - smbd process builds list of all service structures
  - each service struct roughly 1KB (150 parameters)
  - ⇒ for 1000 shares and 1000 connections 1GB of wasted RAM
- I/O-bottleneck
  - smbd always loads *the whole smb.conf*
  - smb.conf file with 1000 shares typically 250KB
  - imagine 1000 clients reloading smb.conf ...
- ease of use
  - programmatically change the configuration
  - change individual parameters
  - write configuration GUIs
  - distribute configuration in clusters
- Design: parsing parameters and activating them is not well separated but is performed in one run.



SerNet

# Disadvantages/Needs

- memory consumption (in large environments)
  - smbd process builds list of all service structures
  - each service struct roughly 1KB (150 parameters)
  - ⇒ for 1000 shares and 1000 connections 1GB of wasted RAM
- I/O-bottleneck
  - smbd always loads *the whole smb.conf*
  - smb.conf file with 1000 shares typically 250KB
  - imagine 1000 clients reloading smb.conf ...
- ease of use
  - programmatically change the configuration
  - change individual parameters
  - write configuration GUIs
  - distribute configuration in clusters
- Design: parsing parameters and activating them is not well separated but is performed in one run.



SerNet

# Disadvantages/Needs

- memory consumption (in large environments)
  - smbd process builds list of all service structures
  - each service struct roughly 1KB (150 parameters)
  - $\Rightarrow$  for 1000 shares and 1000 connections 1GB of wasted RAM
- I/O-bottleneck
  - smbd always loads *the whole* `smb.conf`
  - `smb.conf` file with 1000 shares typically 250KB
  - imagine 1000 clients reloading `smb.conf` ...
- ease of use
  - programmatically change the configuration
  - change individual parameters
  - write configuration GUIs
  - distribute configuration in clusters
- Design: parsing parameters and activating them is not well separated but is performed in one run.



SerNet

# Disadvantages/Needs

- memory consumption (in large environments)
  - smbd process builds list of all service structures
  - each service struct roughly 1KB (150 parameters)
  - $\Rightarrow$  for 1000 shares and 1000 connections 1GB of wasted RAM
- I/O-bottleneck
  - **smbd always loads *the whole smb.conf***
  - smb.conf file with 1000 shares typically 250KB
  - imagine 1000 clients reloading smb.conf ...
- ease of use
  - programmatically change the configuration
  - change individual parameters
  - write configuration GUIs
  - distribute configuration in clusters
- Design: parsing parameters and activating them is not well separated but is performed in one run.



SerNet

# Disadvantages/Needs

- memory consumption (in large environments)
  - smbd process builds list of all service structures
  - each service struct roughly 1KB (150 parameters)
  - $\Rightarrow$  for 1000 shares and 1000 connections 1GB of wasted RAM
- I/O-bottleneck
  - smbd always loads *the whole* `smb.conf`
  - **`smb.conf` file with 1000 shares typically 250KB**
  - imagine 1000 clients reloading `smb.conf` ...
- ease of use
  - programmatically change the configuration
  - change individual parameters
  - write configuration GUIs
  - distribute configuration in clusters
- Design: parsing parameters and activating them is not well separated but is performed in one run.



SerNet

# Disadvantages/Needs

- memory consumption (in large environments)
  - smbd process builds list of all service structures
  - each service struct roughly 1KB (150 parameters)
  - $\Rightarrow$  for 1000 shares and 1000 connections 1GB of wasted RAM
- I/O-bottleneck
  - smbd always loads *the whole* `smb.conf`
  - `smb.conf` file with 1000 shares typically 250KB
  - **imagine 1000 clients reloading `smb.conf` ...**
- ease of use
  - programmatically change the configuration
  - change individual parameters
  - write configuration GUIs
  - distribute configuration in clusters
- Design: parsing parameters and activating them is not well separated but is performed in one run.



SerNet



# Disadvantages/Needs

- memory consumption (in large environments)
  - smbd process builds list of all service structures
  - each service struct roughly 1KB (150 parameters)
  - $\Rightarrow$  for 1000 shares and 1000 connections 1GB of wasted RAM
- I/O-bottleneck
  - smbd always loads *the whole* smb.conf
  - smb.conf file with 1000 shares typically 250KB
  - imagine 1000 clients reloading smb.conf ...
- ease of use
  - programmatically change the configuration
  - change individual parameters
  - write configuration GUIs
  - distribute configuration in clusters
- Design: parsing parameters and activating them is not well separated but is performed in one run.



SerNet

# Disadvantages/Needs

- memory consumption (in large environments)
  - smbd process builds list of all service structures
  - each service struct roughly 1KB (150 parameters)
  - $\Rightarrow$  for 1000 shares and 1000 connections 1GB of wasted RAM
- I/O-bottleneck
  - smbd always loads *the whole* `smb.conf`
  - `smb.conf` file with 1000 shares typically 250KB
  - imagine 1000 clients reloading `smb.conf` ...
- ease of use
  - **programmatically change the configuration**
  - change individual parameters
  - write configuration GUIs
  - distribute configuration in clusters
- Design: parsing parameters and activating them is not well separated but is performed in one run.



SerNet

# Disadvantages/Needs

- memory consumption (in large environments)
  - smbd process builds list of all service structures
  - each service struct roughly 1KB (150 parameters)
  - $\Rightarrow$  for 1000 shares and 1000 connections 1GB of wasted RAM
- I/O-bottleneck
  - smbd always loads *the whole* `smb.conf`
  - `smb.conf` file with 1000 shares typically 250KB
  - imagine 1000 clients reloading `smb.conf` ...
- ease of use
  - programmatically change the configuration
  - **change individual parameters**
  - write configuration GUIs
  - distribute configuration in clusters
- Design: parsing parameters and activating them is not well separated but is performed in one run.



SerNet

# Disadvantages/Needs

- memory consumption (in large environments)
  - smbd process builds list of all service structures
  - each service struct roughly 1KB (150 parameters)
  - $\Rightarrow$  for 1000 shares and 1000 connections 1GB of wasted RAM
- I/O-bottleneck
  - smbd always loads *the whole* smb.conf
  - smb.conf file with 1000 shares typically 250KB
  - imagine 1000 clients reloading smb.conf ...
- ease of use
  - programmatically change the configuration
  - change individual parameters
  - **write configuration GUIs**
  - distribute configuration in clusters
- Design: parsing parameters and activating them is not well separated but is performed in one run.



SerNet

# Disadvantages/Needs

- memory consumption (in large environments)
  - smbd process builds list of all service structures
  - each service struct roughly 1KB (150 parameters)
  - $\Rightarrow$  for 1000 shares and 1000 connections 1GB of wasted RAM
- I/O-bottleneck
  - smbd always loads *the whole* smb.conf
  - smb.conf file with 1000 shares typically 250KB
  - imagine 1000 clients reloading smb.conf ...
- ease of use
  - programmatically change the configuration
  - change individual parameters
  - write configuration GUIs
  - **distribute configuration in clusters**
- Design: parsing parameters and activating them is not well separated but is performed in one run.



SerNet

# Disadvantages/Needs

- memory consumption (in large environments)
  - smbd process builds list of all service structures
  - each service struct roughly 1KB (150 parameters)
  - $\Rightarrow$  for 1000 shares and 1000 connections 1GB of wasted RAM
- I/O-bottleneck
  - smbd always loads *the whole* `smb.conf`
  - `smb.conf` file with 1000 shares typically 250KB
  - imagine 1000 clients reloading `smb.conf` ...
- ease of use
  - programmatically change the configuration
  - change individual parameters
  - write configuration GUIs
  - distribute configuration in clusters
- Design: parsing parameters and activating them is not well separated but is performed in one run.



SerNet

# Disadvantages/Needs

- memory consumption (in large environments)
  - smbd process builds list of all service structures
  - each service struct roughly 1KB (150 parameters)
  - $\Rightarrow$  for 1000 shares and 1000 connections 1GB of wasted RAM
- I/O-bottleneck
  - smbd always loads *the whole* `smb.conf`
  - `smb.conf` file with 1000 shares typically 250KB
  - imagine 1000 clients reloading `smb.conf` ...
- ease of use
  - programmatically change the configuration
  - change individual parameters
  - write configuration GUIs
  - distribute configuration in clusters
- Design: parsing parameters and activating them is not well separated but is performed in one run.



SerNet

# Outline

- 1 Configuration in Samba 3.0 (and before)
  - What We Have
  - What We Need Beyond That
- 2 Configuration in Samba 3.2
  - The Registry
  - Storing Configuration in Registry
  - The `net conf` Utility
  - The `libsmbconf` Library
- 3 Configuring Clustered Samba
  - Clustered Samba
  - Configuring Clustered Samba via Registry
- 4 Current and Ongoing Work
  - Rewrite of `loadparm`
  - Plans / TODOs



SerNet



# Samba's Registry

- Samba has an internal registry database
- needed for communication with Windows clients
- available on the network over the WINREG rpc pipe
- registry data model: tree structure of keys
  - a *key* consists of a name, a list of subkeys and a list of values
  - a *value* consists of a name and the value data
- idea: choose one key for storing samba configuration
  - share  $\leftrightarrow$  subkey
  - parameter in a share  $\leftrightarrow$  value in subkey
- internal storage: *tdb* database
  - access to individual records (parameters) possible
  - write access protected by locks and transactions



SerNet

# Samba's Registry

- Samba has an internal registry database
- needed for communication with Windows clients
- available on the network over the WINREG rpc pipe
- registry data model: tree structure of keys
  - a *key* consists of a name, a list of subkeys and a list of values
  - a *value* consists of a name and the value data
- idea: choose one key for storing samba configuration
  - share  $\leftrightarrow$  subkey
  - parameter in a share  $\leftrightarrow$  value in subkey
- internal storage: *tdb* database
  - access to individual records (parameters) possible
  - write access protected by locks and transactions



SerNet

# Samba's Registry

- Samba has an internal registry database
- **needed for communication with Windows clients**
- available on the network over the WINREG rpc pipe
- registry data model: tree structure of keys
  - a *key* consists of a name, a list of subkeys and a list of values
  - a *value* consists of a name and the value data
- idea: choose one key for storing samba configuration
  - share  $\leftrightarrow$  subkey
  - parameter in a share  $\leftrightarrow$  value in subkey
- internal storage: *tdb* database
  - access to individual records (parameters) possible
  - write access protected by locks and transactions



SerNet

# Samba's Registry

- Samba has an internal registry database
- needed for communication with Windows clients
- available on the network over the WINREG rpc pipe
- registry data model: tree structure of keys
  - a *key* consists of a name, a list of subkeys and a list of values
  - a *value* consists of a name and the value data
- idea: choose one key for storing samba configuration
  - share  $\leftrightarrow$  subkey
  - parameter in a share  $\leftrightarrow$  value in subkey
- internal storage: *tdb* database
  - access to individual records (parameters) possible
  - write access protected by locks and transactions



SerNet

# Samba's Registry

- Samba has an internal registry database
- needed for communication with Windows clients
- available on the network over the WINREG rpc pipe
- **registry data model: tree structure of keys**
  - a *key* consists of a name, a list of subkeys and a list of values
  - a *value* consists of a name and the value data
- idea: choose one key for storing samba configuration
  - share  $\leftrightarrow$  subkey
  - parameter in a share  $\leftrightarrow$  value in subkey
- internal storage: *tdb* database
  - access to individual records (parameters) possible
  - write access protected by locks and transactions



SerNet

# Samba's Registry

- Samba has an internal registry database
- needed for communication with Windows clients
- available on the network over the WINREG rpc pipe
- registry data model: tree structure of keys
  - a *key* consists of a name, a list of subkeys and a list of values
  - a *value* consists of a name and the value data
- idea: choose one key for storing samba configuration
  - share  $\leftrightarrow$  subkey
  - parameter in a share  $\leftrightarrow$  value in subkey
- internal storage: *tdb* database
  - access to individual records (parameters) possible
  - write access protected by locks and transactions



SerNet

# Samba's Registry

- Samba has an internal registry database
- needed for communication with Windows clients
- available on the network over the WINREG rpc pipe
- registry data model: tree structure of keys
  - a *key* consists of a name, a list of subkeys and a list of values
  - a *value* consists of a name and the value data
- idea: choose one key for storing samba configuration
  - share  $\leftrightarrow$  subkey
  - parameter in a share  $\leftrightarrow$  value in subkey
- internal storage: *tdb* database
  - access to individual records (parameters) possible
  - write access protected by locks and transactions



SerNet

# Samba's Registry

- Samba has an internal registry database
- needed for communication with Windows clients
- available on the network over the WINREG rpc pipe
- registry data model: tree structure of keys
  - a *key* consists of a name, a list of subkeys and a list of values
  - a *value* consists of a name and the value data
- **idea: choose one key for storing samba configuration**
  - share  $\leftrightarrow$  subkey
  - parameter in a share  $\leftrightarrow$  value in subkey
- internal storage: *tdb* database
  - access to individual records (parameters) possible
  - write access protected by locks and transactions



SerNet



# Samba's Registry

- Samba has an internal registry database
- needed for communication with Windows clients
- available on the network over the WINREG rpc pipe
- registry data model: tree structure of keys
  - a *key* consists of a name, a list of subkeys and a list of values
  - a *value* consists of a name and the value data
- idea: choose one key for storing samba configuration
  - share  $\leftrightarrow$  subkey
  - parameter in a share  $\leftrightarrow$  value in subkey
- internal storage: *tdb* database
  - access to individual records (parameters) possible
  - write access protected by locks and transactions



SerNet

# Samba's Registry

- Samba has an internal registry database
- needed for communication with Windows clients
- available on the network over the WINREG rpc pipe
- registry data model: tree structure of keys
  - a *key* consists of a name, a list of subkeys and a list of values
  - a *value* consists of a name and the value data
- idea: choose one key for storing samba configuration
  - share  $\Leftrightarrow$  subkey
  - **parameter in a share  $\Leftrightarrow$  value in subkey**
- internal storage: *tdb* database
  - access to individual records (parameters) possible
  - write access protected by locks and transactions



SerNet

# Samba's Registry

- Samba has an internal registry database
- needed for communication with Windows clients
- available on the network over the WINREG rpc pipe
- registry data model: tree structure of keys
  - a *key* consists of a name, a list of subkeys and a list of values
  - a *value* consists of a name and the value data
- idea: choose one key for storing samba configuration
  - share  $\Leftrightarrow$  subkey
  - parameter in a share  $\Leftrightarrow$  value in subkey
- **internal storage: *tdb* database**
  - access to individual records (parameters) possible
  - write access protected by locks and transactions



SerNet

# Samba's Registry

- Samba has an internal registry database
- needed for communication with Windows clients
- available on the network over the WINREG rpc pipe
- registry data model: tree structure of keys
  - a *key* consists of a name, a list of subkeys and a list of values
  - a *value* consists of a name and the value data
- idea: choose one key for storing samba configuration
  - share  $\Leftrightarrow$  subkey
  - parameter in a share  $\Leftrightarrow$  value in subkey
- internal storage: *tdb* database
  - access to individual records (parameters) possible
  - write access protected by locks and transactions



SerNet

# Samba's Registry

- Samba has an internal registry database
- needed for communication with Windows clients
- available on the network over the WINREG rpc pipe
- registry data model: tree structure of keys
  - a *key* consists of a name, a list of subkeys and a list of values
  - a *value* consists of a name and the value data
- idea: choose one key for storing samba configuration
  - share  $\Leftrightarrow$  subkey
  - parameter in a share  $\Leftrightarrow$  value in subkey
- internal storage: *tdb* database
  - access to individual records (parameters) possible
  - write access protected by locks and transactions



SerNet

# Samba's Registry

- Samba has an internal registry database
- needed for communication with Windows clients
- available on the network over the WINREG rpc pipe
- registry data model: tree structure of keys
  - a *key* consists of a name, a list of subkeys and a list of values
  - a *value* consists of a name and the value data
- idea: choose one key for storing samba configuration
  - share  $\Leftrightarrow$  subkey
  - parameter in a share  $\Leftrightarrow$  value in subkey
- internal storage: *tdb* database
  - access to individual records (parameters) possible
  - write access protected by locks and transactions



SerNet

# Caveats

The analogy between `smb.conf` and registry is not complete:

- There is no order in the registry database like there is the line order in the text file.
- Each value name (parameter) appears exactly once in a registry key, whereas in `smb.conf`, the last mention wins.
- Synonymous parameters need special treatment.
- Meta directives like `include` need special treatment



SerNet

# Caveats

The analogy between `smb.conf` and registry is not complete:

- There is no order in the registry database like there is the line order in the text file.
- Each value name (parameter) appears exactly once in a registry key, whereas in `smb.conf`, the last mention wins.
- Synonymous parameters need special treatment.
- Meta directives like `include` need special treatment



SerNet



# Caveats

The analogy between `smb.conf` and registry is not complete:

- There is no order in the registry database like there is the line order in the text file.
- Each value name (parameter) appears exactly once in a registry key, whereas in `smb.conf`, the last mention wins.
- Synonymous parameters need special treatment.
- Meta directives like `include` need special treatment



SerNet

# Caveats

The analogy between `smb.conf` and registry is not complete:

- There is no order in the registry database like there is the line order in the text file.
- Each value name (parameter) appears exactly once in a registry key, whereas in `smb.conf`, the last mention wins.
- **Synonymous parameters need special treatment.**
- Meta directives like `include` need special treatment



SerNet

# Caveats

The analogy between `smb.conf` and registry is not complete:

- There is no order in the registry database like there is the line order in the text file.
- Each value name (parameter) appears exactly once in a registry key, whereas in `smb.conf`, the last mention wins.
- Synonymous parameters need special treatment.
- Meta directives like `include` need special treatment



SerNet

# Caveats

The analogy between `smb.conf` and registry is not complete:

- There is no order in the registry database like there is the line order in the text file.
- Each value name (parameter) appears exactly once in a registry key, whereas in `smb.conf`, the last mention wins.
- Synonymous parameters need special treatment.
- Meta directives like `include` need special treatment



SerNet

# Step One: Registry Shares

- Volker Lendecke introduced *registry shares*
- located underneath registry key  
HKLM\Software\Samba\smbconf
- loaded *on demand* by the server (smbd/service.c)
- registry shares *can* be loaded all at once if required  
(e.g. for testparm)
- activated by "registry shares = yes" in smb.conf
- no other changes to loadparm.c
- access requires the SeDiskOperatorPrivilege



SerNet

# Step One: Registry Shares

- Volker Lendecke introduced *registry shares*
- located underneath registry key  
HKLM\Software\Samba\smbconf
- loaded *on demand* by the server (smbd/service.c)
- registry shares *can* be loaded all at once if required  
(e.g. for testparm)
- activated by "registry shares = yes" in smb.conf
- no other changes to loadparm.c
- access requires the SeDiskOperatorPrivilege



SerNet

# Step One: Registry Shares

- Volker Lendecke introduced *registry shares*
- located underneath registry key  
`HKLM\Software\Samba\smbconf`
- loaded *on demand* by the server (`smbd/service.c`)
- registry shares *can* be loaded all at once if required  
(e.g. for `testparm`)
- activated by "registry shares = yes" in `smb.conf`
- no other changes to `loadparm.c`
- access requires the `SeDiskOperatorPrivilege`



SerNet

# Step One: Registry Shares

- Volker Lendecke introduced *registry shares*
- located underneath registry key  
`HKLM\Software\Samba\smbconf`
- **loaded on demand by the server (`smbd/service.c`)**
- registry shares *can* be loaded all at once if required  
(e.g. for `testparm`)
- activated by "registry shares = yes" in `smb.conf`
- no other changes to `loadparm.c`
- access requires the `SeDiskOperatorPrivilege`



SerNet



# Step One: Registry Shares

- Volker Lendecke introduced *registry shares*
- located underneath registry key  
`HKLM\Software\Samba\smbconf`
- loaded *on demand* by the server (`smbd/service.c`)
- **registry shares *can* be loaded all at once if required**  
**(e.g. for `testparm`)**
- activated by "`registry shares = yes`" in `smb.conf`
- no other changes to `loadparm.c`
- access requires the `SeDiskOperatorPrivilege`



SerNet

# Step One: Registry Shares

- Volker Lendecke introduced *registry shares*
- located underneath registry key  
HKLM\Software\Samba\smbconf
- loaded *on demand* by the server (smbd/service.c)
- registry shares *can* be loaded all at once if required  
(e.g. for testparm)
- **activated by "registry shares = yes" in smb.conf**
- no other changes to loadparm.c
- access requires the SeDiskOperatorPrivilege



SerNet

# Step One: Registry Shares

- Volker Lendecke introduced *registry shares*
- located underneath registry key  
HKLM\Software\Samba\smbconf
- loaded *on demand* by the server (smbd/service.c)
- registry shares *can* be loaded all at once if required  
(e.g. for testparm)
- activated by "registry shares = yes" in smb.conf
- **no other changes to loadparm.c**
- access requires the SeDiskOperatorPrivilege



SerNet

# Step One: Registry Shares

- Volker Lendecke introduced *registry shares*
- located underneath registry key  
HKLM\Software\Samba\smbconf
- loaded *on demand* by the server (smbd/service.c)
- registry shares *can* be loaded all at once if required  
(e.g. for testparm)
- activated by "registry shares = yes" in smb.conf
- no other changes to loadparm.c
- **access requires the SeDiskOperatorPrivilege**



SerNet

# Step One: Registry Shares

- Volker Lendecke introduced *registry shares*
- located underneath registry key  
HKLM\Software\Samba\smbconf
- loaded *on demand* by the server (smbd/service.c)
- registry shares *can* be loaded all at once if required  
(e.g. for testparm)
- activated by "registry shares = yes" in smb.conf
- no other changes to loadparm.c
- access requires the SeDiskOperatorPrivilege



SerNet

## Step Two: Global Registry Configuration

- stored in subkey `global` parallel to registry shares
- function `process_registry_globals()` in `loadparm.c`
- accessed with the same functions as other shares
- `lp_load` loads only global section
- registry shares still loaded on demand



SerNet

## Step Two: Global Registry Configuration

- stored in subkey `global` parallel to registry shares
- function `process_registry_globals()` in `loadparm.c`
- accessed with the same functions as other shares
- `lp_load` loads only global section
- registry shares still loaded on demand



SerNet

## Step Two: Global Registry Configuration

- stored in subkey `global` parallel to registry shares
- `function process_registry_globals()` in `loadparm.c`
- accessed with the same functions as other shares
- `lp_load` loads only global section
- registry shares still loaded on demand



SerNet



## Step Two: Global Registry Configuration

- stored in subkey `global` parallel to registry shares
- function `process_registry_globals()` in `loadparm.c`
- **accessed with the same functions as other shares**
- `lp_load` loads only global section
- registry shares still loaded on demand



SerNet

## Step Two: Global Registry Configuration

- stored in subkey `global` parallel to registry shares
- function `process_registry_globals()` in `loadparm.c`
- accessed with the same functions as other shares
- **lp\_load loads only global section**
- registry shares still loaded on demand



SerNet

## Step Two: Global Registry Configuration

- stored in subkey `global` parallel to registry shares
- function `process_registry_globals()` in `loadparm.c`
- accessed with the same functions as other shares
- `lp_load` loads only global section
- **registry shares still loaded on demand**



SerNet

## Step Two: Global Registry Configuration

- stored in subkey `global` parallel to registry shares
- function `process_registry_globals()` in `loadparm.c`
- accessed with the same functions as other shares
- `lp_load` loads only global section
- registry shares still loaded on demand



SerNet

# How to use it?

## smb.conf - registry only

```
[global]
    config backend = registry
```

## smb.conf - mixed setup

```
[global]
    netbios name = nirvana
    workgroup = SambaXP
    include = registry
    log level = 10

[share1]
    path = /data/share1
```

# How to use it?

## smb.conf - registry only

```
[global]
```

```
config backend = registry
```

## smb.conf - mixed setup

```
[global]
```

```
netbios name = nirvana
```

```
workgroup = SambaXP
```

```
include = registry
```

```
log level = 10
```

```
[share1]
```

```
path = /data/share1
```

## How to use it?

### smb.conf - registry only

```
[global]
```

```
config backend = registry
```

### smb.conf - mixed setup

```
[global]
```

```
netbios name = nirvana
```

```
workgroup = SambaXP
```

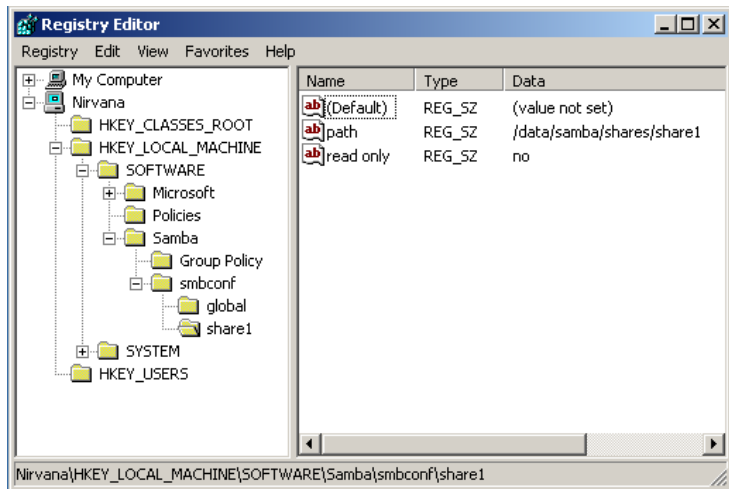
```
include = registry
```

```
log level = 10
```

```
[share1]
```

```
path = /data/share1
```

# Accessing registry with regedit.exe



SerNet



# Accessing registry config ... net conf

- Using `regedit.exe` from Windows to configure Samba is pretty inconvenient.
- Using `regedit.exe` from Windows to configure Samba *just feels horribly wrong* :-)
- There is a unix command line version of `regedit` in new Samba versions: `net [rpc] registry`. (This is better.)
- The new `net conf` subcommand of `net` provides a dedicated and specialized interface to locally read and write the registry based configuration. (Now this is convenient.)



SerNet

# Accessing registry config ... `net conf`

- Using `regedit.exe` from Windows to configure Samba is pretty inconvenient.
- Using `regedit.exe` from Windows to configure Samba *just feels horribly wrong* :-)
- There is a unix command line version of `regedit` in new Samba versions: `net [rpc] registry`. (This is better.)
- The new `net conf` subcommand of `net` provides a dedicated and specialized interface to locally read and write the registry based configuration. (Now this is convenient.)



SerNet

# Accessing registry config ... net conf

- Using `regedit.exe` from Windows to configure Samba is pretty inconvenient.
- Using `regedit.exe` from Windows to configure Samba *just feels horribly wrong :-)*
- There is a unix command line version of `regedit` in new Samba versions: `net [rpc] registry`. (This is better.)
- The new `net conf` subcommand of `net` provides a dedicated and specialized interface to locally read and write the registry based configuration. (Now this is convenient.)



SerNet

# Accessing registry config ... net conf

- Using `regedit.exe` from Windows to configure Samba is pretty inconvenient.
- Using `regedit.exe` from Windows to configure Samba *just feels horribly wrong* :-)
- There is a unix command line version of `regedit` in new Samba versions: `net [rpc] registry`. (This is better.)
- The new `net conf` subcommand of `net` provides a dedicated and specialized interface to locally read and write the registry based configuration. (Now this is convenient.)



SerNet

# Accessing registry config ... net conf

- Using `regedit.exe` from Windows to configure Samba is pretty inconvenient.
- Using `regedit.exe` from Windows to configure Samba *just feels horribly wrong* :-)
- There is a unix command line version of `regedit` in new Samba versions: `net [rpc] registry`. (This is better.)
- The new `net conf` subcommand of `net` provides a dedicated and specialized interface to locally read and write the registry based configuration. (Now this is convenient.)



SerNet

# Accessing registry config ... net conf

- Using `regedit.exe` from Windows to configure Samba is pretty inconvenient.
- Using `regedit.exe` from Windows to configure Samba *just feels horribly wrong* :-)
- There is a unix command line version of `regedit` in new Samba versions: `net [rpc] registry`. (This is better.)
- The new `net conf` subcommand of `net` provides a dedicated and specialized interface to locally read and write the registry based configuration. (Now this is convenient.)



SerNet

# the net conf commands

<code>net conf list</code>	Dump the complete configuration in smb.conf format.
<code>net conf listshares</code>	List the share names.
<code>net conf import</code>	Import configuration from file in smb.conf format.
<code>net conf drop</code>	Delete the complete configuration.
<code>net conf showshare</code>	Show the definition of a share.
<code>net conf addshare</code>	Create a new share.
<code>net conf delshare</code>	Delete a share.
<code>net conf setparm</code>	Store a parameter.
<code>net conf getparm</code>	Retrieve the value of a parameter.
<code>net conf delparm</code>	Delete a parameter.
<code>net conf getincludes</code>	Show the includes of a share definition.
<code>net conf setincludes</code>	Set includes for a share.
<code>net conf delincludes</code>	Delete includes from a share definition.



SerNet

# the net conf commands

<b>net conf list</b>	<b>Dump the complete configuration in smb.conf format.</b>
net conf listshares	List the share names.
net conf import	Import configuration from file in smb.conf format.
net conf drop	Delete the complete configuration.
net conf showshare	Show the definition of a share.
net conf addshare	Create a new share.
net conf delshare	Delete a share.
net conf setparm	Store a parameter.
net conf getparm	Retrieve the value of a parameter.
net conf delparm	Delete a parameter.
net conf getincludes	Show the includes of a share definition.
net conf setincludes	Set includes for a share.
net conf delincludes	Delete includes from a share definition.



SerNet



# the net conf commands

<code>net conf list</code>	Dump the complete configuration in smb.conf format.
<code>net conf listshares</code>	List the share names.
<code>net conf import</code>	Import configuration from file in smb.conf format.
<code>net conf drop</code>	Delete the complete configuration.
<code>net conf showshare</code>	Show the definition of a share.
<code>net conf addshare</code>	Create a new share.
<code>net conf delshare</code>	Delete a share.
<code>net conf setparm</code>	Store a parameter.
<code>net conf getparm</code>	Retrieve the value of a parameter.
<code>net conf delparm</code>	Delete a parameter.
<code>net conf getincludes</code>	Show the includes of a share definition.
<code>net conf setincludes</code>	Set includes for a share.
<code>net conf delincludes</code>	Delete includes from a share definition.



SerNet

# the net conf commands

net conf list	Dump the complete configuration in smb.conf format.
net conf listshares	List the share names.
net conf import	Import configuration from file in smb.conf format.
net conf drop	Delete the complete configuration.
net conf showshare	Show the definition of a share.
net conf addshare	Create a new share.
net conf delshare	Delete a share.
net conf setparm	Store a parameter.
net conf getparm	Retrieve the value of a parameter.
net conf delparm	Delete a parameter.
net conf getincludes	Show the includes of a share definition.
net conf setincludes	Set includes for a share.
net conf delincludes	Delete includes from a share definition.



SerNet

# the net conf commands

net conf list	Dump the complete configuration in smb.conf format.
net conf listshares	List the share names.
net conf import	Import configuration from file in smb.conf format.
<b>net conf drop</b>	<b>Delete the complete configuration.</b>
net conf showshare	Show the definition of a share.
net conf addshare	Create a new share.
net conf delshare	Delete a share.
net conf setparm	Store a parameter.
net conf getparm	Retrieve the value of a parameter.
net conf delparm	Delete a parameter.
net conf getincludes	Show the includes of a share definition.
net conf setincludes	Set includes for a share.
net conf delincludes	Delete includes from a share definition.



SerNet

# the net conf commands

<code>net conf list</code>	Dump the complete configuration in smb.conf format.
<code>net conf listshares</code>	List the share names.
<code>net conf import</code>	Import configuration from file in smb.conf format.
<code>net conf drop</code>	Delete the complete configuration.
<code>net conf showshare</code>	Show the definition of a share.
<code>net conf addshare</code>	Create a new share.
<code>net conf delshare</code>	Delete a share.
<code>net conf setparm</code>	Store a parameter.
<code>net conf getparm</code>	Retrieve the value of a parameter.
<code>net conf delparm</code>	Delete a parameter.
<code>net conf getincludes</code>	Show the includes of a share definition.
<code>net conf setincludes</code>	Set includes for a share.
<code>net conf delincludes</code>	Delete includes from a share definition.



SerNet

# the net conf commands

<code>net conf list</code>	Dump the complete configuration in smb.conf format.
<code>net conf listshares</code>	List the share names.
<code>net conf import</code>	Import configuration from file in smb.conf format.
<code>net conf drop</code>	Delete the complete configuration.
<code>net conf showshare</code>	Show the definition of a share.
<code>net conf addshare</code>	Create a new share.
<code>net conf delshare</code>	Delete a share.
<code>net conf setparm</code>	Store a parameter.
<code>net conf getparm</code>	Retrieve the value of a parameter.
<code>net conf delparm</code>	Delete a parameter.
<code>net conf getincludes</code>	Show the includes of a share definition.
<code>net conf setincludes</code>	Set includes for a share.
<code>net conf delincludes</code>	Delete includes from a share definition.



SerNet

# the net conf commands

<code>net conf list</code>	Dump the complete configuration in smb.conf format.
<code>net conf listshares</code>	List the share names.
<code>net conf import</code>	Import configuration from file in smb.conf format.
<code>net conf drop</code>	Delete the complete configuration.
<code>net conf showshare</code>	Show the definition of a share.
<code>net conf addshare</code>	Create a new share.
<code>net conf delshare</code>	Delete a share.
<code>net conf setparm</code>	Store a parameter.
<code>net conf getparm</code>	Retrieve the value of a parameter.
<code>net conf delparm</code>	Delete a parameter.
<code>net conf getincludes</code>	Show the includes of a share definition.
<code>net conf setincludes</code>	Set includes for a share.
<code>net conf delincludes</code>	Delete includes from a share definition.



SerNet

# the net conf commands

<code>net conf list</code>	Dump the complete configuration in smb.conf format.
<code>net conf listshares</code>	List the share names.
<code>net conf import</code>	Import configuration from file in smb.conf format.
<code>net conf drop</code>	Delete the complete configuration.
<code>net conf showshare</code>	Show the definition of a share.
<code>net conf addshare</code>	Create a new share.
<code>net conf delshare</code>	Delete a share.
<code>net conf setparm</code>	Store a parameter.
<code>net conf getparm</code>	Retrieve the value of a parameter.
<code>net conf delparm</code>	Delete a parameter.
<code>net conf getincludes</code>	Show the includes of a share definition.
<code>net conf setincludes</code>	Set includes for a share.
<code>net conf delincludes</code>	Delete includes from a share definition.



SerNet

Breakout: Demonstration of usage of net conf / registry config...



SerNet



# The libsmbconf Library

- abstraction of the operations of `net_conf` to a set of modules under `lib/smbconf`
- provide complete and stable API (hopefully achieved now)
- one `libsmbconf` "object" corresponds to one parsed config source
- delivers configuration as (lists of) strings
- allow different backends behind the API
- backends implemented:
  - registry backend
  - text backend (read-only, based on `params.c`)
- more backends can be implemented (tdb, LDAP, ...)



SerNet

# The libsmbconf Library

- abstraction of the operations of `net_conf` to a set of modules under `lib/smbconf`
- provide complete and stable API (hopefully achieved now)
- one `libsmbconf` "object" corresponds to one parsed config source
- delivers configuration as (lists of) strings
- allow different backends behind the API
- backends implemented:
  - registry backend
  - text backend (read-only, based on `params.c`)
- more backends can be implemented (tdb, LDAP, ...)



SerNet

# The libsmbconf Library

- abstraction of the operations of `net_conf` to a set of modules under `lib/smbconf`
- **provide complete and stable API (hopefully achieved now)**
- one `libsmbconf` "object" corresponds to one parsed config source
- delivers configuration as (lists of) strings
- allow different backends behind the API
- backends implemented:
  - registry backend
  - text backend (read-only, based on `params.c`)
- more backends can be implemented (tdb, LDAP, ...)



SerNet

# The libsmbconf Library

- abstraction of the operations of `net_conf` to a set of modules under `lib/smbconf`
- provide complete and stable API (hopefully achieved now)
- **one `libsmbconf` "object" corresponds to one parsed config source**
- delivers configuration as (lists of) strings
- allow different backends behind the API
- backends implemented:
  - registry backend
  - text backend (read-only, based on `params.c`)
- more backends can be implemented (tdb, LDAP, ...)



SerNet

# The libsmbconf Library

- abstraction of the operations of `net_conf` to a set of modules under `lib/smbconf`
- provide complete and stable API (hopefully achieved now)
- one `libsmbconf` "object" corresponds to one parsed config source
- **delivers configuration as (lists of) strings**
- allow different backends behind the API
- backends implemented:
  - registry backend
  - text backend (read-only, based on `params.c`)
- more backends can be implemented (tdb, LDAP, ...)



SerNet

# The libsmbconf Library

- abstraction of the operations of `net_conf` to a set of modules under `lib/smbconf`
- provide complete and stable API (hopefully achieved now)
- one `libsmbconf` "object" corresponds to one parsed config source
- delivers configuration as (lists of) strings
- **allow different backends behind the API**
- backends implemented:
  - registry backend
  - text backend (read-only, based on `params.c`)
- more backends can be implemented (tdb, LDAP, ...)



SerNet

# The libsmbconf Library

- abstraction of the operations of `net_conf` to a set of modules under `lib/smbconf`
- provide complete and stable API (hopefully achieved now)
- one `libsmbconf` "object" corresponds to one parsed config source
- delivers configuration as (lists of) strings
- allow different backends behind the API
- **backends implemented:**
  - registry backend
  - text backend (read-only, based on `params.c`)
- more backends can be implemented (tdb, LDAP, ...)



SerNet

# The libsmbconf Library

- abstraction of the operations of `net_conf` to a set of modules under `lib/smbconf`
- provide complete and stable API (hopefully achieved now)
- one `libsmbconf` "object" corresponds to one parsed config source
- delivers configuration as (lists of) strings
- allow different backends behind the API
- backends implemented:
  - **registry backend**
  - text backend (read-only, based on `params.c`)
- more backends can be implemented (tdb, LDAP, ...)



SerNet



# The libsmbconf Library

- abstraction of the operations of `net_conf` to a set of modules under `lib/smbconf`
- provide complete and stable API (hopefully achieved now)
- one `libsmbconf` "object" corresponds to one parsed config source
- delivers configuration as (lists of) strings
- allow different backends behind the API
- backends implemented:
  - registry backend
  - text backend (read-only, based on `params.c`)
- more backends can be implemented (tdb, LDAP, ...)



SerNet

# The libsmbconf Library

- abstraction of the operations of `net_conf` to a set of modules under `lib/smbconf`
- provide complete and stable API (hopefully achieved now)
- one `libsmbconf` "object" corresponds to one parsed config source
- delivers configuration as (lists of) strings
- allow different backends behind the API
- backends implemented:
  - registry backend
  - text backend (read-only, based on `params.c`)
- more backends can be implemented (tdb, LDAP, ...)



SerNet

# The libsmbconf Library

- abstraction of the operations of `net_conf` to a set of modules under `lib/smbconf`
- provide complete and stable API (hopefully achieved now)
- one `libsmbconf` "object" corresponds to one parsed config source
- delivers configuration as (lists of) strings
- allow different backends behind the API
- backends implemented:
  - registry backend
  - text backend (read-only, based on `params.c`)
- more backends can be implemented (tdb, LDAP, ...)



SerNet

# A small glimpse into the libsmbconf API

## smbconf service structure

```
struct smbconf_service {  
    char *name;  
    uint32_t num_params;  
    char **param_names;  
    char **param_values;  
};
```



SerNet

# A small glimpse into the libsmbconf API

## smbconf service structure

```
struct smbconf_service {  
    char *name;  
    uint32_t num_params;  
    char **param_names;  
    char **param_values;  
};
```



SerNet

# Application: libnetapi and the netdomjoin-gui

- Günther Deschner has developed a NetAPI library.
- We have extended the NetJoinDomain call to modify the registry configuration upon successful join.
- Similar for the unjoin (leave) operation.
- This allows joining a domain with an empty samba configuration.
- The example `netdomjoin-gui` GTK application lets you set the machine and workgroup name and join or leave an AD domain.



SerNet

# Application: libnetapi and the netdomjoin-gui

- **Günther Deschner has developed a NetAPI library.**
- We have extended the NetJoinDomain call to modify the registry configuration upon successful join.
- Similar for the unjoin (leave) operation.
- This allows joining a domain with an empty samba configuration.
- The example netdomjoin-gui GTK application lets you set the machine and workgroup name and join or leave an AD domain.



SerNet

# Application: libnetapi and the netdomjoin-gui

- Günther Deschner has developed a NetAPI library.
- We have extended the NetJoinDomain call to modify the registry configuration upon successful join.
- Similar for the unjoin (leave) operation.
- This allows joining a domain with an empty samba configuration.
- The example netdomjoin-gui GTK application lets you set the machine and workgroup name and join or leave an AD domain.



SerNet



# Application: libnetapi and the netdomjoin-gui

- Günther Deschner has developed a NetAPI library.
- We have extended the NetJoinDomain call to modify the registry configuration upon successful join.
- **Similar for the unjoin (leave) operation.**
- This allows joining a domain with an empty samba configuration.
- The example `netdomjoin-gui` GTK application lets you set the machine and workgroup name and join or leave an AD domain.



SerNet

# Application: libnetapi and the netdomjoin-gui

- Günther Deschner has developed a NetAPI library.
- We have extended the NetJoinDomain call to modify the registry configuration upon successful join.
- Similar for the unjoin (leave) operation.
- This allows joining a domain with an empty samba configuration.
- The example netdomjoin-gui GTK application lets you set the machine and workgroup name and join or leave an AD domain.



SerNet

# Application: libnetapi and the netdomjoin-gui

- Günther Deschner has developed a NetAPI library.
- We have extended the NetJoinDomain call to modify the registry configuration upon successful join.
- Similar for the unjoin (leave) operation.
- This allows joining a domain with an empty samba configuration.
- The example `netdomjoin-gui` GTK application lets you set the machine and workgroup name and join or leave an AD domain.



SerNet

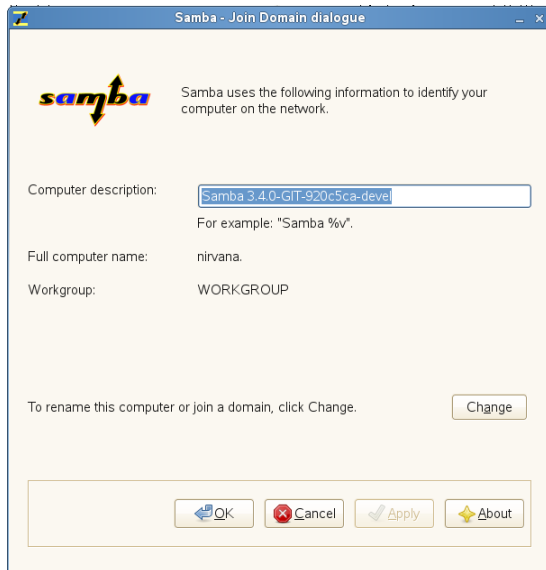
# Application: libnetapi and the netdomjoin-gui

- Günther Deschner has developed a NetAPI library.
- We have extended the NetJoinDomain call to modify the registry configuration upon successful join.
- Similar for the unjoin (leave) operation.
- This allows joining a domain with an empty samba configuration.
- The example `netdomjoin-gui` GTK application lets you set the machine and workgroup name and join or leave an AD domain.



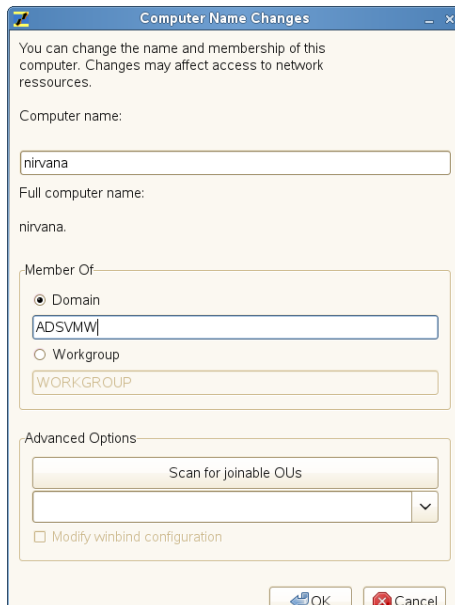
SerNet

# The netdomjoin-gui



SerNet

# Application: The netdomjoin-gui



The screenshot shows a window titled "Computer Name Changes" with a standard Windows XP-style title bar. The window contains the following elements:

- A message: "You can change the name and membership of this computer. Changes may affect access to network ressources." (Note the typo 'ressources').
- A label "Computer name:" followed by a text input field containing "nirvana".
- A label "Full computer name:" followed by the text "nirvana.".
- A section titled "Member Of" containing:
  - A radio button labeled "Domain" which is selected.
  - A text input field containing "ADSVMW".
  - A radio button labeled "Workgroup".
  - A text input field containing "WORKGROUP".
- A section titled "Advanced Options" containing:
  - A button labeled "Scan for joinable OUs".
  - A dropdown menu.
  - A checkbox labeled "Modify winbind configuration" which is currently unchecked.
- At the bottom right, there are "OK" and "Cancel" buttons.



SerNet

# Outline

- 1 Configuration in Samba 3.0 (and before)
  - What We Have
  - What We Need Beyond That
- 2 Configuration in Samba 3.2
  - The Registry
  - Storing Configuration in Registry
  - The `net conf` Utility
  - The `libsmbconf` Library
- 3 **Configuring Clustered Samba**
  - Clustered Samba
  - Configuring Clustered Samba via Registry
- 4 Current and Ongoing Work
  - Rewrite of `loadparm`
  - Plans / TODOs



SerNet

# Clustered Samba: Intentions

- Have Samba serve cluster storage from multiple cluster nodes simultaneously (active-active storage cluster).
- Make Samba daemons on multiple nodes appear as one CIFS server (basically turns your cluster into a NAS).
- Scale in performance with the number of nodes in the cluster.



SerNet



# Clustered Samba: Intentions

- Have Samba serve cluster storage from multiple cluster nodes simultaneously (active-active storage cluster).
- Make Samba daemons on multiple nodes appear as one CIFS server (basically turns your cluster into a NAS).
- Scale in performance with the number of nodes in the cluster.



SerNet

# Clustered Samba: Intentions

- Have Samba serve cluster storage from multiple cluster nodes simultaneously (active-active storage cluster).
- Make Samba daemons on multiple nodes appear as one CIFS server (basically turns your cluster into a NAS).
- Scale in performance with the number of nodes in the cluster.



SerNet

# Clustered Samba: Intentions

- Have Samba serve cluster storage from multiple cluster nodes simultaneously (active-active storage cluster).
- Make Samba daemons on multiple nodes appear as one CIFS server (basically turns your cluster into a NAS).
- Scale in performance with the number of nodes in the cluster.



SerNet

# Clustered Samba: Intentions

- Have Samba serve cluster storage from multiple cluster nodes simultaneously (active-active storage cluster).
- Make Samba daemons on multiple nodes appear as one CIFS server (basically turns your cluster into a NAS).
- Scale in performance with the number of nodes in the cluster.



SerNet

# Clustered Samba: Problems

- Coherence and scalability of Samba's many internal tdb databases.
- These are used for locking and other runtime data.
- Other, persistent data like passwords and registry configuration is also stored in tdb databases.
- For coherence, this could go into the cluster storage.
- But excessive use of fcntl locks and very many small write operations lead to very bad performance, even negative scaling, since the locks need to be propagated across the cluster nodes.



SerNet

# Clustered Samba: Problems

- **Coherence and scalability of Samba's many internal tdb databases.**
  - These are used for locking and other runtime data.
  - Other, persistent data like passwords and registry configuration is also stored in tdb databases.
  - For coherence, this could go into the cluster storage.
  - But excessive use of fcntl locks and very many small write operations lead to very bad performance, even negative scaling, since the locks need to be propagated across the cluster nodes.



SerNet

# Clustered Samba: Problems

- Coherence and scalability of Samba's many internal tdb databases.
- **These are used for locking and other runtime data.**
- Other, persistent data like passwords and registry configuration is also stored in tdb databases.
- For coherence, this could go into the cluster storage.
- But excessive use of fcntl locks and very many small write operations lead to very bad performance, even negative scaling, since the locks need to be propagated across the cluster nodes.



SerNet

# Clustered Samba: Problems

- Coherence and scalability of Samba's many internal tdb databases.
- These are used for locking and other runtime data.
- Other, persistent data like passwords and registry configuration is also stored in tdb databases.
- For coherence, this could go into the cluster storage.
- But excessive use of fcntl locks and very many small write operations lead to very bad performance, even negative scaling, since the locks need to be propagated across the cluster nodes.



SerNet



# Clustered Samba: Problems

- Coherence and scalability of Samba's many internal tdb databases.
- These are used for locking and other runtime data.
- Other, persistent data like passwords and registry configuration is also stored in tdb databases.
- For coherence, this could go into the cluster storage.
- But excessive use of fcntl locks and very many small write operations lead to very bad performance, even negative scaling, since the locks need to be propagated across the cluster nodes.



SerNet

# Clustered Samba: Problems

- Coherence and scalability of Samba's many internal tdb databases.
- These are used for locking and other runtime data.
- Other, persistent data like passwords and registry configuration is also stored in tdb databases.
- For coherence, this could go into the cluster storage.
- But excessive use of fcntl locks and very many small write operations lead to very bad performance, even negative scaling, since the locks need to be propagated across the cluster nodes.



SerNet

# Clustered Samba: Problems

- Coherence and scalability of Samba's many internal tdb databases.
- These are used for locking and other runtime data.
- Other, persistent data like passwords and registry configuration is also stored in tdb databases.
- For coherence, this could go into the cluster storage.
- But excessive use of fcntl locks and very many small write operations lead to very bad performance, even negative scaling, since the locks need to be propagated across the cluster nodes.



SerNet

# ctdb - Clustered tdb

- Idea: There is no need to store all data of a tdb in the cluster FS.
- All nodes have their local copy of the data.
- One node holds the authoritative copy of the tdb – *data master*.
- Changes by the data master are propagated to the other cluster nodes.
- Other nodes need to become data master before changing the tdb.
- Handling of the metadata is done by the userspace ctdb daemon (running on each node).
- The ctdb daemons communicate over an internal cluster network and a recovery lock file stored in the cluster file system.
- The ctdb commandline tool is used to manage ctddb.
- The onnode script can run commands on single or multiple nodes.



SerNet

# ctdb - Clustered tdb

- **Idea: There is no need to store all data of a tdb in the cluster FS.**
- All nodes have their local copy of the data.
- One node holds the authoritative copy of the tdb – *data master*.
- Changes by the data master are propagated to the other cluster nodes.
- Other nodes need to become data master before changing the tdb.
- Handling of the metadata is done by the userspace ctdb daemon (running on each node).
- The ctdb daemons communicate over an internal cluster network and a recovery lock file stored in the cluster file system.
- The ctdb commandline tool is used to manage ctddb.
- The onnode script can run commands on single or multiple nodes.



SerNet

# ctdb - Clustered tdb

- Idea: There is no need to store all data of a tdb in the cluster FS.
- **All nodes have their local copy of the data.**
- One node holds the authoritative copy of the tdb – *data master*.
- Changes by the data master are propagated to the other cluster nodes.
- Other nodes need to become data master before changing the tdb.
- Handling of the metadata is done by the userspace ctdb daemon (running on each node).
- The ctdb daemons communicate over an internal cluster network and a recovery lock file stored in the cluster file system.
- The ctdb commandline tool is used to manage ctddb.
- The onnode script can run commands on single or multiple nodes.



SerNet

# ctdb - Clustered tdb

- Idea: There is no need to store all data of a tdb in the cluster FS.
- All nodes have their local copy of the data.
- **One node holds the authoritative copy of the tdb – *data master*.**
- Changes by the data master are propagated to the other cluster nodes.
- Other nodes need to become data master before changing the tdb.
- Handling of the metadata is done by the userspace ctdb daemon (running on each node).
- The ctdb daemons communicate over an internal cluster network and a recovery lock file stored in the cluster file system.
- The ctdb commandline tool is used to manage ctddb.
- The onnode script can run commands on single or multiple nodes.



SerNet

## ctdb - Clustered tdb

- Idea: There is no need to store all data of a tdb in the cluster FS.
- All nodes have their local copy of the data.
- One node holds the authoritative copy of the tdb – *data master*.
- **Changes by the data master are propagated to the other cluster nodes.**
- Other nodes need to become data master before changing the tdb.
- Handling of the metadata is done by the userspace ctdb daemon (running on each node).
- The ctdb daemons communicate over an internal cluster network and a recovery lock file stored in the cluster file system.
- The ctdb commandline tool is used to manage ctddb.
- The onnode script can run commands on single or multiple nodes.



SerNet



## ctdb - Clustered tdb

- Idea: There is no need to store all data of a tdb in the cluster FS.
- All nodes have their local copy of the data.
- One node holds the authoritative copy of the tdb – *data master*.
- Changes by the data master are propagated to the other cluster nodes.
- **Other nodes need to become data master before changing the tdb.**
- Handling of the metadata is done by the userspace ctdb daemon (running on each node).
- The ctdb daemons communicate over an internal cluster network and a recovery lock file stored in the cluster file system.
- The ctdb commandline tool is used to manage ctddb.
- The onnode script can run commands on single or multiple nodes.



SerNet

## ctdb - Clustered tdb

- Idea: There is no need to store all data of a tdb in the cluster FS.
- All nodes have their local copy of the data.
- One node holds the authoritative copy of the tdb – *data master*.
- Changes by the data master are propagated to the other cluster nodes.
- Other nodes need to become data master before changing the tdb.
- Handling of the metadata is done by the userspace ctdb daemon (running on each node).
- The ctdb daemons communicate over an internal cluster network and a recovery lock file stored in the cluster file system.
- The ctdb commandline tool is used to manage ctddb.
- The onnode script can run commands on single or multiple nodes.



SerNet

## ctdb - Clustered tdb

- Idea: There is no need to store all data of a tdb in the cluster FS.
- All nodes have their local copy of the data.
- One node holds the authoritative copy of the tdb – *data master*.
- Changes by the data master are propagated to the other cluster nodes.
- Other nodes need to become data master before changing the tdb.
- Handling of the metadata is done by the userspace ctdb daemon (running on each node).
- The ctdb daemons communicate over an internal cluster network and a recovery lock file stored in the cluster file system.
- The ctdb commandline tool is used to manage ctddb.
- The onnode script can run commands on single or multiple nodes.



SerNet

## ctdb - Clustered tdb

- Idea: There is no need to store all data of a tdb in the cluster FS.
- All nodes have their local copy of the data.
- One node holds the authoritative copy of the tdb – *data master*.
- Changes by the data master are propagated to the other cluster nodes.
- Other nodes need to become data master before changing the tdb.
- Handling of the metadata is done by the userspace ctdb daemon (running on each node).
- The ctdb daemons communicate over an internal cluster network and a recovery lock file stored in the cluster file system.
- The ctdb commandline tool is used to manage ctddb.
- The onnode script can run commands on single or multiple nodes.



SerNet

## ctdb - Clustered tdb

- Idea: There is no need to store all data of a tdb in the cluster FS.
- All nodes have their local copy of the data.
- One node holds the authoritative copy of the tdb – *data master*.
- Changes by the data master are propagated to the other cluster nodes.
- Other nodes need to become data master before changing the tdb.
- Handling of the metadata is done by the userspace ctdb daemon (running on each node).
- The ctdb daemons communicate over an internal cluster network and a recovery lock file stored in the cluster file system.
- The ctdb commandline tool is used to manage ctddb.
- The **onnode** script can run commands on single or multiple nodes.



SerNet

## ctdb - Clustered tdb

- Idea: There is no need to store all data of a tdb in the cluster FS.
- All nodes have their local copy of the data.
- One node holds the authoritative copy of the tdb – *data master*.
- Changes by the data master are propagated to the other cluster nodes.
- Other nodes need to become data master before changing the tdb.
- Handling of the metadata is done by the userspace ctdb daemon (running on each node).
- The ctdb daemons communicate over an internal cluster network and a recovery lock file stored in the cluster file system.
- The ctdb commandline tool is used to manage ctddb.
- The onnode script can run commands on single or multiple nodes.



SerNet

# Setting up ctdb

- `/etc/sysconfig/ctdb` holds basic setup
- `/etc/ctdb/nodes` specifies the nodes of the ctdb cluster (internal network addresses)
- `/etc/ctdb/public_addresses` specifies the external IP addresses the cluster node can possibly take.
- with ctdb's IP takeover (of public addresses) and e.g. round robin DNS, a simple load balancing is easily implemented



SerNet

# Setting up ctdb

- **/etc/sysconfig/ctdb holds basic setup**
- /etc/ctdb/nodes specifies the nodes of the ctdb cluster (internal network addresses)
- /etc/ctdb/public\_addresses specifies the external IP addresses the cluster node can possibly take.
- with ctdb's IP takeover (of public addresses) and e.g. round robin DNS, a simple load balancing is easily implemented



SerNet



# Setting up ctdb

- `/etc/sysconfig/ctdb` holds basic setup
- `/etc/ctdb/nodes` specifies the nodes of the ctdb cluster (internal network addresses)
- `/etc/ctdb/public_addresses` specifies the external IP addresses the cluster node can possibly take.
- with ctdb's IP takeover (of public addresses) and e.g. round robin DNS, a simple load balancing is easily implemented



SerNet

# Setting up ctdb

- `/etc/sysconfig/ctdb` holds basic setup
- `/etc/ctdb/nodes` specifies the nodes of the ctdb cluster (internal network addresses)
- `/etc/ctdb/public_addresses` specifies the external IP addresses the cluster node can possibly take.
- with ctdb's IP takeover (of public addresses) and e.g. round robin DNS, a simple load balancing is easily implemented



SerNet

# Setting up ctdb

- `/etc/sysconfig/ctdb` holds basic setup
- `/etc/ctdb/nodes` specifies the nodes of the ctdb cluster (internal network addresses)
- `/etc/ctdb/public_addresses` specifies the external IP addresses the cluster node can possibly take.
- with ctdb's IP takeover (of public addresses) and e.g. round robin DNS, a simple load balancing is easily implemented



SerNet

# Setting up ctdb

- `/etc/sysconfig/ctdb` holds basic setup
- `/etc/ctdb/nodes` specifies the nodes of the ctdb cluster (internal network addresses)
- `/etc/ctdb/public_addresses` specifies the external IP addresses the cluster node can possibly take.
- with ctdb's IP takeover (of public addresses) and e.g. round robin DNS, a simple load balancing is easily implemented



SerNet

# Samba and ctdb

- Support for using ctdb is available since Samba 3.2.0.
- There is a clustered Samba branch `v3-2-ctdb` with special enhancements: `git://git.samba.org/obnox/samba-ctdb.git`
- Samba must be configured (at build time) with `--with-cluster-support`.
- Activated by global configuration option `clustering = yes`
- This makes samba talk to ctdbd instead of directly writing to local tdb databases (for most tdb's).
- This is used successfully in production on top of the GPFS clustered file system.



SerNet

# Samba and ctdb

- Support for using ctdb is available since Samba 3.2.0.
- There is a clustered Samba branch v3-2-ctdb with special enhancements: `git://git.samba.org/obnox/samba-ctdb.git`
- Samba must be configured (at build time) with `--with-cluster-support`.
- Activated by global configuration option `clustering = yes`
- This makes samba talk to ctdbd instead of directly writing to local tdb databases (for most tdb's).
- This is used successfully in production on top of the GPFS clustered file system.



SerNet

# Samba and ctdb

- Support for using ctdb is available since Samba 3.2.0.
- There is a clustered Samba branch `v3-2-ctdb` with special enhancements: `git://git.samba.org/obnox/samba-ctdb.git`
- Samba must be configured (at build time) with `--with-cluster-support`.
- Activated by global configuration option `clustering = yes`
- This makes samba talk to ctdbd instead of directly writing to local tdb databases (for most tdbs).
- This is used successfully in production on top of the GPFS clustered file system.



SerNet

# Samba and ctdb

- Support for using ctdb is available since Samba 3.2.0.
- There is a clustered Samba branch `v3-2-ctdb` with special enhancements: `git://git.samba.org/obnox/samba-ctdb.git`
- **Samba must be configured (at build time) with `--with-cluster-support`.**
- Activated by global configuration option `clustering = yes`
- This makes samba talk to ctdbd instead of directly writing to local tdb databases (for most tdb's).
- This is used successfully in production on top of the GPFS clustered file system.



SerNet



# Samba and ctdb

- Support for using ctdb is available since Samba 3.2.0.
- There is a clustered Samba branch `v3-2-ctdb` with special enhancements: `git://git.samba.org/obnox/samba-ctdb.git`
- Samba must be configured (at build time) with `--with-cluster-support`.
- **Activated by global configuration option `clustering = yes`**
- This makes samba talk to ctdbd instead of directly writing to local tdb databases (for most tdb's).
- This is used successfully in production on top of the GPFS clustered file system.



SerNet

# Samba and ctdb

- Support for using ctdb is available since Samba 3.2.0.
- There is a clustered Samba branch `v3-2-ctdb` with special enhancements: `git://git.samba.org/obnox/samba-ctdb.git`
- Samba must be configured (at build time) with `--with-cluster-support`.
- Activated by global configuration option `clustering = yes`
- This makes samba talk to ctdbd instead of directly writing to local tdb databases (for most tdb's).
- This is used successfully in production on top of the GPFS clustered file system.



SerNet

# Samba and ctdb

- Support for using ctdb is available since Samba 3.2.0.
- There is a clustered Samba branch `v3-2-ctdb` with special enhancements: `git://git.samba.org/obnox/samba-ctdb.git`
- Samba must be configured (at build time) with `--with-cluster-support`.
- Activated by global configuration option `clustering = yes`
- This makes samba talk to ctdbd instead of directly writing to local tdb databases (for most tdb's).
- This is used successfully in production on top of the GPFS clustered file system.



SerNet

# Samba and ctdb

- Support for using ctdb is available since Samba 3.2.0.
- There is a clustered Samba branch `v3-2-ctdb` with special enhancements: `git://git.samba.org/obnox/samba-ctdb.git`
- Samba must be configured (at build time) with `--with-cluster-support`.
- Activated by global configuration option `clustering = yes`
- This makes samba talk to ctdbd instead of directly writing to local tdb databases (for most tdb's).
- This is used successfully in production on top of the GPFS clustered file system.



SerNet

# Configuring Clustered Samba

This is the basic configuration file needs to be on all nodes:

smb.conf for clustering

```
[global]
    clustering = yes
    include = registry
```

The rest of the configuration goes into registry and is automatically propagated to all nodes.



SerNet

# Configuring Clustered Samba

This is the basic configuration file needs to be on all nodes:

smb.conf for clustering

```
[global]
    clustering = yes
    include = registry
```

The rest of the configuration goes into registry and is automatically propagated to all nodes.



SerNet

# Configuring Clustered Samba

This is the basic configuration file needs to be on all nodes:

**smb.conf for clustering**

```
[global]
    clustering = yes
    include = registry
```

The rest of the configuration goes into registry and is automatically propagated to all nodes.



SerNet

# Configuring Clustered Samba

This is the basic configuration file needs to be on all nodes:

**smb.conf for clustering**

```
[global]
    clustering = yes
    include = registry
```

The rest of the configuration goes into registry and is automatically propagated to all nodes.



SerNet



Breakout: demo ctdb and configuring clustered Samba...



SerNet

# Outline

- 1 Configuration in Samba 3.0 (and before)
  - What We Have
  - What We Need Beyond That
- 2 Configuration in Samba 3.2
  - The Registry
  - Storing Configuration in Registry
  - The `net conf` Utility
  - The `libsmbconf` Library
- 3 Configuring Clustered Samba
  - Clustered Samba
  - Configuring Clustered Samba via Registry
- 4 Current and Ongoing Work
  - Rewrite of `loadparm`
  - Plans / TODOs



SerNet

# changes to loadparm.c

- use libsmbconf throughout, i.e. remove all direct references to the parser (param.c)
- This cleans the design of loadparm and separates parsing of parameters from their activation
- replace concept of config file by that **config source**
- The smb.conf file is no longer necessary the initial config source.
- Registry configuration is no longer restricted to a single key. Configuration sources from different keys can be included.



SerNet

# changes to loadparm.c

- use libsmbconf throughout, i.e. remove all direct references to the parser (`param.c`)
- This cleans the design of loadparm and separates parsing of parameters from their activation
- replace concept of config file by that **config source**
- The `smb.conf` file is no longer necessary the initial config source.
- Registry configuration is no longer restricted to a single key. Configuration sources from different keys can be included.



SerNet

## changes to loadparm.c

- use libsmbconf throughout, i.e. remove all direct references to the parser (param.c)
- This cleans the design of loadparm and separates parsing of parameters from their activation
- replace concept of config file by that **config source**
- The smb.conf file is no longer necessary the initial config source.
- Registry configuration is no longer restricted to a single key. Configuration sources from different keys can be included.



SerNet

## changes to loadparm.c

- use libsmbconf throughout, i.e. remove all direct references to the parser (`param.c`)
- This cleans the design of loadparm and separates parsing of parameters from their activation
- **replace concept of config file by that config source**
- The `smb.conf` file is no longer necessary the initial config source.
- Registry configuration is no longer restricted to a single key. Configuration sources from different keys can be included.



SerNet

## changes to loadparm.c

- use libsmbconf throughout, i.e. remove all direct references to the parser (param.c)
- This cleans the design of loadparm and separates parsing of parameters from their activation
- replace concept of config file by that **config source**
- **The smb.conf file is no longer necessary the initial config source.**
- Registry configuration is no longer restricted to a single key. Configuration sources from different keys can be included.



SerNet

## changes to loadparm.c

- use libsmbconf throughout, i.e. remove all direct references to the parser (`param.c`)
- This cleans the design of loadparm and separates parsing of parameters from their activation
- replace concept of config file by that **config source**
- The `smb.conf` file is no longer necessary the initial config source.
- Registry configuration is no longer restricted to a single key. Configuration sources from different keys can be included.



SerNet



## changes to loadparm.c

- use libsmbconf throughout, i.e. remove all direct references to the parser (`param.c`)
- This cleans the design of loadparm and separates parsing of parameters from their activation
- replace concept of config file by that **config source**
- The `smb.conf` file is no longer necessary the initial config source.
- Registry configuration is no longer restricted to a single key. Configuration sources from different keys can be included.



SerNet

# Plans / TODOs

- finish and release rewrite of loadparm (3.4 - Summer 2009)
- make libsmbconf and registry shared libraries
- Add write support to text backend (comment handling?)
- write more backends:
  - GPO backend (with Günther Deschner)
  - LDAP or other DB backend
  - XML
  - ...
- reconcile registry library with Samba4 (with Jelmer Vernooij)
- make use of libsmbconf in Samba4/Franky



SerNet

# Plans / TODOs

- finish and release rewrite of loadparm (3.4 - Summer 2009)
- make libsmbconf and registry shared libraries
- Add write support to text backend (comment handling?)
- write more backends:
  - GPO backend (with Günther Deschner)
  - LDAP or other DB backend
  - XML
  - ...
- reconcile registry library with Samba4 (with Jelmer Vernooij)
- make use of libsmbconf in Samba4/Franky



SerNet

# Plans / TODOs

- finish and release rewrite of loadparm (3.4 - Summer 2009)
- **make libsmbconf and registry shared libraries**
- Add write support to text backend (comment handling?)
- write more backends:
  - GPO backend (with Günther Deschner)
  - LDAP or other DB backend
  - XML
  - ...
- reconcile registry library with Samba4 (with Jelmer Vernooij)
- make use of libsmbconf in Samba4/Franky



SerNet

# Plans / TODOs

- finish and release rewrite of loadparm (3.4 - Summer 2009)
- make libsmbconf and registry shared libraries
- **Add write support to text backend (comment handling?)**
- write more backends:
  - GPO backend (with Günther Deschner)
  - LDAP or other DB backend
  - XML
  - ...
- reconcile registry library with Samba4 (with Jelmer Vernooij)
- make use of libsmbconf in Samba4/Franky



SerNet

# Plans / TODOs

- finish and release rewrite of loadparm (3.4 - Summer 2009)
- make libsmbconf and registry shared libraries
- Add write support to text backend (comment handling?)
- **write more backends:**
  - GPO backend (with Günther Deschner)
  - LDAP or other DB backend
  - XML
  - ...
- reconcile registry library with Samba4 (with Jelmer Vernooij)
- make use of libsmbconf in Samba4/Franky



SerNet

# Plans / TODOs

- finish and release rewrite of loadparm (3.4 - Summer 2009)
- make libsmbconf and registry shared libraries
- Add write support to text backend (comment handling?)
- write more backends:
  - **GPO backend (with Günther Deschner)**
  - LDAP or other DB backend
  - XML
  - ...
- reconcile registry library with Samba4 (with Jelmer Vernooij)
- make use of libsmbconf in Samba4/Franky



SerNet

# Plans / TODOs

- finish and release rewrite of loadparm (3.4 - Summer 2009)
- make libsmbconf and registry shared libraries
- Add write support to text backend (comment handling?)
- write more backends:
  - GPO backend (with Günther Deschner)
  - **LDAP or other DB backend**
  - XML
  - ...
- reconcile registry library with Samba4 (with Jelmer Vernooij)
- make use of libsmbconf in Samba4/Franky



SerNet



# Plans / TODOs

- finish and release rewrite of loadparm (3.4 - Summer 2009)
- make libsmbconf and registry shared libraries
- Add write support to text backend (comment handling?)
- write more backends:
  - GPO backend (with Günther Deschner)
  - LDAP or other DB backend
  - XML
  - ...
- reconcile registry library with Samba4 (with Jelmer Vernooij)
- make use of libsmbconf in Samba4/Franky



SerNet

# Plans / TODOs

- finish and release rewrite of loadparm (3.4 - Summer 2009)
- make libsmbconf and registry shared libraries
- Add write support to text backend (comment handling?)
- write more backends:
  - GPO backend (with Günther Deschner)
  - LDAP or other DB backend
  - XML
  - ...
- reconcile registry library with Samba4 (with Jelmer Vernooij)
- make use of libsmbconf in Samba4/Franky



SerNet

# Plans / TODOs

- finish and release rewrite of loadparm (3.4 - Summer 2009)
- make libsmbconf and registry shared libraries
- Add write support to text backend (comment handling?)
- write more backends:
  - GPO backend (with Günther Deschner)
  - LDAP or other DB backend
  - XML
  - ...
- **reconcile registry library with Samba4 (with Jelmer Vernooij)**
- make use of libsmbconf in Samba4/Franky



SerNet

# Plans / TODOs

- finish and release rewrite of loadparm (3.4 - Summer 2009)
- make libsmbconf and registry shared libraries
- Add write support to text backend (comment handling?)
- write more backends:
  - GPO backend (with Günther Deschner)
  - LDAP or other DB backend
  - XML
  - ...
- reconcile registry library with Samba4 (with Jelmer Vernooij)
- **make use of libsmbconf in Samba4/Franky**



SerNet

# Plans / TODOs

- finish and release rewrite of loadparm (3.4 - Summer 2009)
- make libsmbconf and registry shared libraries
- Add write support to text backend (comment handling?)
- write more backends:
  - GPO backend (with Günther Deschner)
  - LDAP or other DB backend
  - XML
  - ...
- reconcile registry library with Samba4 (with Jelmer Vernooij)
- make use of libsmbconf in Samba4/Franky



SerNet

Thank you very much! — Questions?



SerNet