Samba's New Configuration System (libsmbconf)

Michael Adam

obnox@samba.org

SerNet / Samba Team

2008-04-18

Breakout: Demonstration of configuring Samba with `regedit.exe`

Outline

1. Configuration in Samba 3.0 (and before)
   - What We Have
   - What We Need Beyond That
2. Configuration in Samba 3.2.0
   - Ideas
   - First Steps
   - The `net conf` Utility
   - Global Registry Configuration
   - The `libsmbconf` Library
3. Current and Ongoing Work
   - Rewrite of Loadparm
   - Plans / TODOs

Outline

1. Configuration in Samba 3.0 (and before)
   - What We Have
   - What We Need Beyond That
2. Configuration in Samba 3.2.0
   - Ideas
   - First Steps
   - The `net conf` Utility
   - Global Registry Configuration
   - The `libsmbconf` Library
3. Current and Ongoing Work
   - Rewrite of Loadparm
   - Plans / TODOs

SerNet

Samba 3.0 configuration

- text based configuration in `win.ini` style: `smb.conf` file
- modularized by include directives
- 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

samba

Samba 3.0 configuration

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

Samba 3.0 configuration

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

Samba 3.0 configuration

- text based configuration in `win.ini` style: `smb.conf` file
- modularized by include directives
- 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

Samba 3.0 configuration

- text based configuration in `win.ini` style: `smb.conf` file
- modularized by include directives
- 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

Samba 3.0 configuration

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

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
```

Needs

- programmatically change the configuration
- change individual parameters
- distribute configuration in clusters

Needs

- programmatically change the configuration
- change individual parameters
- distribute configuration in clusters

SerNet

samba

Needs

- programmatically change the configuration
- change individual parameters
- distribute configuration in clusters

Outline

1 Configuration in Samba 3.0 (and before)
- What We Have
- What We Need Beyond That

2 Configuration in Samba 3.2.0
- Ideas
- First Steps
- The `net conf` Utility
- Global Registry Configuration
- The `libsmbconf` Library

3 Current and Ongoing Work
- Rewrite of Loadparm
- Plans / TODOs

SerNet

samba

Ideas

- use registry to store config - data model is well suited:
  - share ⇔ key
  - parameter ⇔ value
- write configuration tool as subcommand of `net`

SerNet

samba

Ideas

- use registry to store config - data model is well suited:
  - share ⇔ key
  - parameter ⇔ value
- write configuration tool as subcommand of `net`

SerNet

Ideas

- use registry to store config - data model is well suited:
  - share ⇔ key
  - parameter ⇔ value
- write configuration tool as subcommand of `net`

Ideas

- use registry to store config - data model is well suited:
  - share ⇔ key
  - parameter ⇔ value
- write configuration tool as subcommand of `net`

First Steps - by Volker Lendecke

- introduced **registry shares**
- located underneath registry key `HKLM\Software\Samba\smbconf`
- loaded on demand by server (`smbd/service.c`)
- activated by "`registry shares = yes`" in `smb.conf`
- no other changes to `loadparm.c`
- access requires the SeDiskOperatorPrivilege

First Steps - by Volker Lendecke

- introduced **registry shares**
- located underneath registry key `HKLM\Software\Samba\smbconf`
- loaded on demand by server (`smbd/service.c`)
- activated by "`registry shares = yes`" in `smb.conf`
- no other changes to `loadparm.c`
- access requires the SeDiskOperatorPrivilege

SerNet

First Steps - by Volker Lendecke

- introduced **registry shares**
- located underneath registry key `HKLM\Software\Samba\smbconf`
- loaded on demand by server (`smbd/service.c`)
- activated by "`registry shares = yes`" in `smb.conf`
- no other changes to `loadparm.c`
- access requires the SeDiskOperatorPrivilege

SerNet

First Steps - by Volker Lendecke

- introduced **registry shares**
- located underneath registry key `HKLM\Software\Samba\smbconf`
- loaded on demand by server (`smbd/service.c`)
- activated by "`registry shares = yes`" in `smb.conf`
- no other changes to `loadparm.c`
- access requires the SeDiskOperatorPrivilege

First Steps - by Volker Lendecke

- introduced **registry shares**
- located underneath registry key `HKLM\Software\Samba\smbconf`
- loaded on demand by server (`smbd/service.c`)
- activated by "`registry shares = yes`" in `smb.conf`
- no other changes to `loadparm.c`
- access requires the SeDiskOperatorPrivilege

SerNet

First Steps - by Volker Lendecke

- introduced **registry shares**
- located underneath registry key `HKLM\Software\Samba\smbconf`
- loaded on demand by server (`smbd/service.c`)
- activated by "`registry shares = yes`" in `smb.conf`
- no other changes to `loadparm.c`
- access requires the SeDiskOperatorPrivilege

SerNet

samba

First Steps - by Volker Lendecke

- introduced **registry shares**
- located underneath registry key `HKLM\Software\Samba\smbconf`
- loaded on demand by server (`smbd/service.c`)
- activated by "`registry shares = yes`" in `smb.conf`
- no other changes to `loadparm.c`
- access requires the SeDiskOperatorPrivilege

net conf

- `net conf`: subcommand `net` to locally read and write the registry based configuration.
  (using regedit is inconvenient)
- preliminary work: untangle samba3's registry from the server code (in order to be able link registry in at all)

net conf

- `net conf`: subcommand `net` to locally read and write the registry based configuration.
  (using regedit is inconvenient)
- preliminary work: untangle samba3's registry from the server code (in order to be able link registry in at all)

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.
```

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.`

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.

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.
```

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.

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.

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.
```

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.
```

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

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

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

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

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

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

How to use it?

smb.conf - registry only

```
[global]
          config bakend = 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 bakend = 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 bakend = registry
```

smb.conf - mixed setup

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

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

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

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

samba

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

samba

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, ...)

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, ...)

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, ...)

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, ...)

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, ...)

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, ...)

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, ...)

The libsmbconf API

init / shutdown

```
WERROR smbconf_init(TALLOC_CTX *mem_ctx,
                    struct smbconf_ctx **conf_ctx,
                    const char *source);


void smbconf_shutdown(struct smbconf_ctx *ctx);
```

source string

```
backend:path
```

The libsmbconf API

init / shutdown

```
WERROR smbconf_init(TALLOC_CTX *mem_ctx,
                    struct smbconf_ctx **conf_ctx,
                    const char *source);

void smbconf_shutdown(struct smbconf_ctx *ctx);
```

source string

```
backend:path
```

The libsmbconf API

init / shutdown

```
WERROR smbconf_init(TALLOC_CTX *mem_ctx,
                    struct smbconf_ctx **conf_ctx,
                    const char *source);

void smbconf_shutdown(struct smbconf_ctx *ctx);
```

source string

```
backend:path
```

The libsmbconf API

init / shutdown

```
WERROR smbconf_init(TALLOC_CTX *mem_ctx,
                    struct smbconf_ctx **conf_ctx,
                    const char *source);

void smbconf_shutdown(struct smbconf_ctx *ctx);
```

source string

```
backend:path
```

The libsmbconf API

init / shutdown

```
WERROR smbconf_init(TALLOC_CTX *mem_ctx,
                    struct smbconf_ctx **conf_ctx,
                    const char *source);

void smbconf_shutdown(struct smbconf_ctx *ctx);
```

source string

```
backend:path
```

The libsmbconf API

init / shutdown

```
WERROR smbconf_init(TALLOC_CTX *mem_ctx,
                    struct smbconf_ctx **conf_ctx,
                    const char *source);

void smbconf_shutdown(struct smbconf_ctx *ctx);
```

source string

```
backend:path
```

The libsmbconf API

parameter functions

```
WERROR smbconf_get_parameter(struct smbconf_ctx *conf_ctx,
                             TALLOC_CTX *mem_ctx,
                             const char *service, const char *param,
                             char **valstr);

WERROR smbconf_set_parameter(struct smbconf_ctx *ctx,
                             const char *service, const char *param,
                             const char *val);

WERROR smbconf_delete_parameter(struct smbconf_ctx *ctx,
                                const char *service, const char *param);
```

SerNet

The libsmbconf API

parameter functions

```
WERROR smbconf_get_parameter(struct smbconf_ctx *conf_ctx,
                             TALLOC_CTX *mem_ctx,
                             const char *service, const char *param,
                             char **valstr);

WERROR smbconf_set_parameter(struct smbconf_ctx *ctx,
                             const char *service, const char *param,
                             const char *val);

WERROR smbconf_delete_parameter(struct smbconf_ctx *ctx,
                                const char *service, const char *param);
```

SerNet

samba

The libsmbconf API

parameter functions

```
WERROR smbconf_get_parameter(struct smbconf_ctx *conf_ctx,
                             TALLOC_CTX *mem_ctx,
                             const char *service, const char *param,
                             char **valstr);

WERROR smbconf_set_parameter(struct smbconf_ctx *ctx,
                             const char *service, const char *param,
                             const char *val);


WERROR smbconf_delete_parameter(struct smbconf_ctx *ctx,
                                const char *service, const char *param);
```

SerNet

The libsmbconf API

parameter functions

```
WERROR smbconf_get_parameter(struct smbconf_ctx *conf_ctx,
                             TALLOC_CTX *mem_ctx,
                             const char *service, const char *param,
                             char **valstr);

WERROR smbconf_set_parameter(struct smbconf_ctx *ctx,
                             const char *service, const char *param,
                             const char *val);

WERROR smbconf_delete_parameter(struct smbconf_ctx *ctx,
                                const char *service, const char *param);
```

SerNet

The libsmbconf API

parameter functions

```
WERROR smbconf_get_parameter(struct smbconf_ctx *conf_ctx,
                             TALLOC_CTX *mem_ctx,
                             const char *service, const char *param,
                             char **valstr);

WERROR smbconf_set_parameter(struct smbconf_ctx *ctx,
                             const char *service, const char *param,
                             const char *val);

WERROR smbconf_delete_parameter(struct smbconf_ctx *ctx,
                                const char *service, const char *param);
```

The libsmbconf API

share functions

```
WERROR smbconf_get_share(struct smbconf_ctx *ctx,
                         TALLOC_CTX *mem_ctx,
                         const char *service, uint32_t *num_params,
                         char ***param_names, char ***param_values);

WERROR smbconf_create_share(struct smbconf_ctx *ctx,
                            const char *service);

WERROR smbconf_delete_share(struct smbconf_ctx *ctx,
                            const char *service);

bool smbconf_share_exists(struct smbconf_ctx *ctx, const char *service);
```

The libsmbconf API

share functions

```
WERROR smbconf_get_share(struct smbconf_ctx *ctx,
                         TALLOC_CTX *mem_ctx,
                         const char *service, uint32_t *num_params,
                         char ***param_names, char ***param_values);

WERROR smbconf_create_share(struct smbconf_ctx *ctx,
                            const char *service);

WERROR smbconf_delete_share(struct smbconf_ctx *ctx,
                            const char *service);

bool smbconf_share_exists(struct smbconf_ctx *ctx, const char *service);
```

The libsmbconf API

share functions

```
WERROR smbconf_get_share(struct smbconf_ctx *ctx,
                         TALLOC_CTX *mem_ctx,
                         const char *service, uint32_t *num_params,
                         char ***param_names, char ***param_values);

WERROR smbconf_create_share(struct smbconf_ctx *ctx,
                            const char *service);

WERROR smbconf_delete_share(struct smbconf_ctx *ctx,
                            const char *service);

bool smbconf_share_exists(struct smbconf_ctx *ctx, const char *service);
```

SerNet

The libsmbconf API

share functions

```
WERROR smbconf_get_share(struct smbconf_ctx *ctx,
                         TALLOC_CTX *mem_ctx,
                         const char *service, uint32_t *num_params,
                         char ***param_names, char ***param_values);

WERROR smbconf_create_share(struct smbconf_ctx *ctx,
                            const char *service);

WERROR smbconf_delete_share(struct smbconf_ctx *ctx,
                            const char *service);

bool smbconf_share_exists(struct smbconf_ctx *ctx, const char *service);
```

SerNet

The libsmbconf API

share functions

```
WERROR smbconf_get_share(struct smbconf_ctx *ctx,
                         TALLOC_CTX *mem_ctx,
                         const char *service, uint32_t *num_params,
                         char ***param_names, char ***param_values);

WERROR smbconf_create_share(struct smbconf_ctx *ctx,
                            const char *service);

WERROR smbconf_delete_share(struct smbconf_ctx *ctx,
                            const char *service);

bool smbconf_share_exists(struct smbconf_ctx *ctx, const char *service);
```

The libsmbconf API

share functions

```
WERROR smbconf_get_share(struct smbconf_ctx *ctx,
                         TALLOC_CTX *mem_ctx,
                         const char *service, uint32_t *num_params,
                         char ***param_names, char ***param_values);

WERROR smbconf_create_share(struct smbconf_ctx *ctx,
                            const char *service);

WERROR smbconf_delete_share(struct smbconf_ctx *ctx,
                            const char *service);

bool smbconf_share_exists(struct smbconf_ctx *ctx, const char *service);
```

SerNet

The libsmbconf API

global functions

```
WERROR smbconf_drop(struct smbconf_ctx *ctx);

WERROR smbconf_get_share_names(struct smbconf_ctx *ctx,
                               TALLOC_CTX *mem_ctx,
                               uint32_t *num_shares,
                               char ***share_names);

WERROR smbconf_get_config(struct smbconf_ctx *ctx,
                          TALLOC_CTX *mem_ctx,
                          uint32_t *num_shares,
                          char ***share_names, uint32_t **num_params,
                          char ****param_names, char ****param_values);
```

The libsmbconf API

global functions

```
WERROR smbconf_drop(struct smbconf_ctx *ctx);

WERROR smbconf_get_share_names(struct smbconf_ctx *ctx,
                               TALLOC_CTX *mem_ctx,
                               uint32_t *num_shares,
                               char ***share_names);

WERROR smbconf_get_config(struct smbconf_ctx *ctx,
                          TALLOC_CTX *mem_ctx,
                          uint32_t *num_shares,
                          char ***share_names, uint32_t **num_params,
                          char ****param_names, char ****param_values);
```

The libsmbconf API

global functions

```
WERROR smbconf_drop(struct smbconf_ctx *ctx);

WERROR smbconf_get_share_names(struct smbconf_ctx *ctx,
                               TALLOC_CTX *mem_ctx,
                               uint32_t *num_shares,
                               char ***share_names);

WERROR smbconf_get_config(struct smbconf_ctx *ctx,
                          TALLOC_CTX *mem_ctx,
                          uint32_t *num_shares,
                          char ***share_names, uint32_t **num_params,
                          char ****param_names, char ****param_values);
```

SerNet

The libsmbconf API

global functions

```
WERROR smbconf_drop(struct smbconf_ctx *ctx);

WERROR smbconf_get_share_names(struct smbconf_ctx *ctx,
                               TALLOC_CTX *mem_ctx,
                               uint32_t *num_shares,
                               char ***share_names);

WERROR smbconf_get_config(struct smbconf_ctx *ctx,
                          TALLOC_CTX *mem_ctx,
                          uint32_t *num_shares,
                          char ***share_names, uint32_t **num_params,
                          char ****param_names, char ****param_values);
```

The libsmbconf API

global functions

```
WERROR smbconf_drop(struct smbconf_ctx *ctx);

WERROR smbconf_get_share_names(struct smbconf_ctx *ctx,
                               TALLOC_CTX *mem_ctx,
                               uint32_t *num_shares,
                               char ***share_names);

WERROR smbconf_get_config(struct smbconf_ctx *ctx,
                          TALLOC_CTX *mem_ctx,
                          uint32_t *num_shares,
                          char ***share_names, uint32_t **num_params,
                          char ****param_names, char ****param_values);
```

The libsmbconf API

includes functions

```
WERROR smbconf_get_includes(struct smbconf_ctx *ctx,
                            TALLOC_CTX *mem_ctx,
                            const char *service,
                            uint32_t *num_includes,
                            char ***includes);

WERROR smbconf_set_includes(struct smbconf_ctx *ctx,
                            const char *service,
                            uint32_t num_includes,
                            const char **includes);

WERROR smbconf_delete_includes(struct smbconf_ctx *ctx,
                               const char *service);
```

The libsmbconf API

includes functions

```
WERROR smbconf_get_includes(struct smbconf_ctx *ctx,
                            TALLOC_CTX *mem_ctx,
                            const char *service,
                            uint32_t *num_includes,
                            char ***includes);

WERROR smbconf_set_includes(struct smbconf_ctx *ctx,
                            const char *service,
                            uint32_t num_includes,
                            const char **includes);

WERROR smbconf_delete_includes(struct smbconf_ctx *ctx,
                               const char *service);
```

The libsmbconf API

includes functions

```
WERROR smbconf_get_includes(struct smbconf_ctx *ctx,
                            TALLOC_CTX *mem_ctx,
                            const char *service,
                            uint32_t *num_includes,
                            char ***includes);

WERROR smbconf_set_includes(struct smbconf_ctx *ctx,
                            const char *service,
                            uint32_t num_includes,
                            const char **includes);

WERROR smbconf_delete_includes(struct smbconf_ctx *ctx,
                               const char *service);
```

The libsmbconf API

includes functions

```
WERROR smbconf_get_includes(struct smbconf_ctx *ctx,
                            TALLOC_CTX *mem_ctx,
                            const char *service,
                            uint32_t *num_includes,
                            char ***includes);

WERROR smbconf_set_includes(struct smbconf_ctx *ctx,
                            const char *service,
                            uint32_t num_includes,
                            const char **includes);

WERROR smbconf_delete_includes(struct smbconf_ctx *ctx,
                               const char *service);
```

The libsmbconf API

includes functions

```
WERROR smbconf_get_includes(struct smbconf_ctx *ctx,
                            TALLOC_CTX *mem_ctx,
                            const char *service,
                            uint32_t *num_includes,
                            char ***includes);

WERROR smbconf_set_includes(struct smbconf_ctx *ctx,
                            const char *service,
                            uint32_t num_includes,
                            const char **includes);

WERROR smbconf_delete_includes(struct smbconf_ctx *ctx,
                               const char *service);
```

Outline

1. Configuration in Samba 3.0 (and before)
   - What We Have
   - What We Need Beyond That
2. Configuration in Samba 3.2.0
   - Ideas
   - First Steps
   - The `net conf` Utility
   - Global Registry Configuration
   - The `libsmbconf` Library
3. Current and Ongoing Work
   - Rewrite of Loadparm
   - Plans / TODOs

SerNet

samba

changes to `loadparm.c`

- use libsmbconf throughout
- i.e. remove all references to param.c (parser)
- `config file` ⇒ **config source**
- change `lp_load()`
- change `handle_include()`
- drop `process_registry_globals()`

- CONFIGFILE ⇒ CONFIGSOURCE (dynconfig)
- `--configfile` ⇒ `--configsource` (popt)

changes to `loadparm.c`

- use libsmbconf throughout
- i.e. remove all references to param.c (parser)
- `config file` ⇒ **config source**
- change `lp_load`()
- change `handle_include`()
- drop `process_registry_globals`()

- CONFIGFILE ⇒ CONFIGSOURCE (dynconfig)
- `--configfile` ⇒ `--configsource` (popt)

changes to `loadparm.c`

- use libsmbconf throughout
- i.e. remove all references to param.c (parser)
- `config file` ⇒ **config source**
- change `lp_load()`
- change `handle_include()`
- drop `process_registry_globals()`

- CONFIGFILE ⇒ CONFIGSOURCE (dynconfig)
- `--configfile` ⇒ `--configsource` (popt)

SerNet

samba

changes to `loadparm.c`

- use libsmbconf throughout
- i.e. remove all references to param.c (parser)
- `config file` ⇒ **config source**
- change `lp_load()`
- change `handle_include()`
- drop `process_registry_globals()`

- CONFIGFILE ⇒ CONFIGSOURCE (dynconfig)
- `--configfile` ⇒ `--configsource` (popt)

SerNet

samba

changes to loadparm.c

- use libsmbconf throughout
- i.e. remove all references to param.c (parser)
- config file ⇒ **config source**
- change lp_load()
- change handle_include()
- drop process_registry_globals()

- CONFIGFILE ⇒ CONFIGSOURCE (dynconfig)
- --configfile ⇒ --configsource (popt)

changes to `loadparm.c`

- use libsmbconf throughout
- i.e. remove all references to param.c (parser)
- `config file` ⇒ **config source**
- change `lp_load()`
- change `handle_include()`
- drop `process_registry_globals()`

- CONFIGFILE ⇒ CONFIGSOURCE (dynconfig)
- `--configfile` ⇒ `--configsource` (popt)

changes to `loadparm.c`

- use libsmbconf throughout
- i.e. remove all references to param.c (parser)
- `config file` ⇒ **config source**
- change `lp_load()`
- change `handle_include()`
- drop `process_registry_globals()`
- CONFIGFILE ⇒ CONFIGSOURCE (dynconfig)
- `--configfile` ⇒ `--configsource` (popt)

changes to loadparm.c

- use libsmbconf throughout
- i.e. remove all references to param.c (parser)
- config file ⇒ **config source**
- change lp_load()
- change handle_include()
- drop process_registry_globals()
- CONFIGFILE ⇒ CONFIGSOURCE (dynconfig)
- --configfile ⇒ --configsource (popt)

SerNet

samba

changes to `loadparm.c`

- use libsmbconf throughout
- i.e. remove all references to param.c (parser)
- `config file` ⇒ **config source**
- change `lp_load`()
- change `handle_include`()
- drop `process_registry_globals`()

- CONFIGFILE ⇒ CONFIGSOURCE (dynconfig)
- `--configfile` ⇒ `--configsource` (popt)

changes to `loadparm.c`

- use libsmbconf throughout
- i.e. remove all references to param.c (parser)
- `config file` ⇒ **config source**
- change `lp_load`()
- change `handle_include`()
- drop `process_registry_globals`()

- CONFIGFILE ⇒ CONFIGSOURCE (dynconfig)
- `--configfile` ⇒ `--configsource` (popt)

Separation of Layers

`loadparm:lp_load()`

- read config data via libsmbconf
- process raw parameters in a simple for loop

`libsmbconf dispatcher`

- delegate queries to backend
- returns resulting config in backend independent format

libsmbconf backends

- fetch data from text / registry / ...

Separation of Layers

loadparm:lp_load()

- read config data via libsmbconf
- process raw parameters in a simple for loop

libsmbconf dispatcher

- delegate queries to backend
- returns resulting config in backend independent format

libsmbconf backends

- fetch data from text / registry / ...

Separation of Layers

loadparm:lp_load()

- read config data via libsmbconf
- process raw parameters in a simple for loop

libsmbconf dispatcher

- delegate queries to backend
- returns resulting config in backend independent format

libsmbconf backends

- fetch data from text / registry / ...

Separation of Layers

loadparm:lp_load()

- read config data via libsmbconf
- process raw parameters in a simple for loop

libsmbconf dispatcher

- delegate queries to backend
- returns resulting config in backend independent format

libsmbconf backends

- fetch data from text / registry / ...

SerNet

samba

Separation of Layers

`loadparm:lp_load()`

- read config data via libsmbconf
- process raw parameters in a simple for loop

`libsmbconf dispatcher`

- delegate queries to backend
- returns resulting config in backend independent format

libsmbconf backends

- fetch data from text / registry / . . .

Separation of Layers

loadparm:lp_load()

- read config data via libsmbconf
- process raw parameters in a simple for loop

libsmbconf dispatcher

- delegate queries to backend
- returns resulting config in backend independent format

libsmbconf backends

- fetch data from text / registry / ...

Separation of Layers

`loadparm:lp_load()`

- read config data via libsmbconf
- process raw parameters in a simple for loop

`libsmbconf dispatcher`

- delegate queries to backend
- returns resulting config in backend independent format

libsmbconf backends

- fetch data from text / registry / . . .

how `lp_load()` works *now* (simplified)

```
source = add_to_source_list(sources_list, source_name, sub_name);

smbconf_get_config(source->ctx, mem_ctx, &num_shares,
                   &share_names, &num_params,
                   &param_names, &param_values);

for (scount = 0; scount < num_shares; scount++) {
       do_section(share_names[scount], NULL);
       for (pcount = 0; pcount < num_params[scount]; pcount++) {
                do_parameter(param_names[scount][pcount],
                             param_values[scount][pcount]
                             NULL);
        }
}
```

how lp_load() works *now* (simplified)

```
source = add_to_source_list(sources_list, source_name, sub_name);

smbconf_get_config(source->ctx, mem_ctx, &num_shares,
                   &share_names, &num_params,
                   &param_names, &param_values);

for (scount = 0; scount < num_shares; scount++) {
        do_section(share_names[scount], NULL);
        for (pcount = 0; pcount < num_params[scount]; pcount++) {
                do_parameter(param_names[scount][pcount],
                             param_values[scount][pcount]
                             NULL);
        }
}
```

SerNet

samba

how lp_load() works *now* (simplified)

```
source = add_to_source_list(sources_list, source_name, sub_name);

smbconf_get_config(source->ctx, mem_ctx, &num_shares,
                   &share_names, &num_params,
                   &param_names, &param_values);

for (scount = 0; scount < num_shares; scount++) {
       do_section(share_names[scount], NULL);
       for (pcount = 0; pcount < num_params[scount]; pcount++) {
                do_parameter(param_names[scount][pcount],
                             param_values[scount][pcount]
                             NULL);
        }
}
```

how lp_load() works *now* (simplified)

```
source = add_to_source_list(sources_list, source_name, sub_name);

smbconf_get_config(source->ctx, mem_ctx, &num_shares,
                   &share_names, &num_params,
                   &param_names, &param_values);

for (scount = 0; scount < num_shares; scount++) {
       do_section(share_names[scount], NULL);
       for (pcount = 0; pcount < num_params[scount]; pcount++) {
                do_parameter(param_names[scount][pcount],
                             param_values[scount][pcount]
                             NULL);
        }
}
```

how lp_load() works *now* (simplified)

```
source = add_to_source_list(sources_list, source_name, sub_name);

smbconf_get_config(source->ctx, mem_ctx, &num_shares,
                   &share_names, &num_params,
                   &param_names, &param_values);

for (scount = 0; scount < num_shares; scount++) {
       do_section(share_names[scount], NULL);
       for (pcount = 0; pcount < num_params[scount]; pcount++) {
                do_parameter(param_names[scount][pcount],
                             param_values[scount][pcount]
                             NULL);
        }
}
```

how lp_load() works *now* (simplified)

```
source = add_to_source_list(sources_list, source_name, sub_name);

smbconf_get_config(source->ctx, mem_ctx, &num_shares,
                   &share_names, &num_params,
                   &param_names, &param_values);

for (scount = 0; scount < num_shares; scount++) {
       do_section(share_names[scount], NULL);
       for (pcount = 0; pcount < num_params[scount]; pcount++) {
                do_parameter(param_names[scount][pcount],
                             param_values[scount][pcount]
                             NULL);

        }
}
```

how lp_load() works *now* (simplified)

```
source = add_to_source_list(sources_list, source_name, sub_name);

smbconf_get_config(source->ctx, mem_ctx, &num_shares,
                   &share_names, &num_params,
                   &param_names, &param_values);

for (scount = 0; scount < num_shares; scount++) {
       do_section(share_names[scount], NULL);
       for (pcount = 0; pcount < num_params[scount]; pcount++) {
                do_parameter(param_names[scount][pcount],
                             param_values[scount][pcount]
                             NULL);
        }
}
```

SerNet

how `lp_load()` works *now* (simplified)

```
source = add_to_source_list(sources_list, source_name, sub_name);

smbconf_get_config(source->ctx, mem_ctx, &num_shares,
                   &share_names, &num_params,
                   &param_names, &param_values);

for (scount = 0; scount < num_shares; scount++) {
       do_section(share_names[scount], NULL);
       for (pcount = 0; pcount < num_params[scount]; pcount++) {
                do_parameter(param_names[scount][pcount],
                             param_values[scount][pcount]
                             NULL);
        }
}
```

Breakout: Demonstration of new features

Plans / TODOs

- finish and release rewrite of loadparm
- 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
  - ...
- reconcile registry library with Samba4 (with Jelmer Vernooij)
- make use of libsmbconf in Samba4

Plans / TODOs

- finish and release rewrite of loadparm
- 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
    - ...
- reconcile registry library with Samba4 (with Jelmer Vernooij)
- make use of libsmbconf in Samba4

Plans / TODOs

- finish and release rewrite of loadparm
- 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
  - ...
- reconcile registry library with Samba4 (with Jelmer Vernooij)
- make use of libsmbconf in Samba4

Plans / TODOs

- finish and release rewrite of loadparm
- 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
  - ...
- reconcile registry library with Samba4 (with Jelmer Vernooij)
- make use of libsmbconf in Samba4

Plans / TODOs

- finish and release rewrite of loadparm
- 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
  - . . .
- reconcile registry library with Samba4 (with Jelmer Vernooij)
- make use of libsmbconf in Samba4

Plans / TODOs

- finish and release rewrite of loadparm
- 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
  - ...
- reconcile registry library with Samba4 (with Jelmer Vernooij)
- make use of libsmbconf in Samba4

Plans / TODOs

- finish and release rewrite of loadparm
- 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
  - ...
- reconcile registry library with Samba4 (with Jelmer Vernooij)
- make use of libsmbconf in Samba4

SerNet

Plans / TODOs

- finish and release rewrite of loadparm
- 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
  - ...
- reconcile registry library with Samba4 (with Jelmer Vernooij)
- make use of libsmbconf in Samba4

Plans / TODOs

- finish and release rewrite of loadparm
- 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
  - ...
- reconcile registry library with Samba4 (with Jelmer Vernooij)
- make use of libsmbconf in Samba4

Plans / TODOs

- finish and release rewrite of loadparm
- 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
  - . . .
- reconcile registry library with Samba4 (with Jelmer Vernooij)
- make use of libsmbconf in Samba4

SerNet

Plans / TODOs

- finish and release rewrite of loadparm
- 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
  - . . .
- reconcile registry library with Samba4 (with Jelmer Vernooij)
- make use of libsmbconf in Samba4

Thank you very much!

SerNet