



Franz Vater	Thats me
vimacon GmbH	My company
videt.org	My tclhttpd playground and information portal server

1. The beginning
2. Requirements, wishes
3. Reference installation
4. Technical details
5. Status, examples
6. Outlook

- A webserver to serve all kinds of information
- A base to realize all kinds of services for
 - documentation
 - data management
 - live data
 - cooperation, colaboration

1. A small company developing HMI systems for machine manufacturers
2. A self-governed cohousing project

Manuals, Project Management, Issue Tracking
Todo-Lists, Blog, File Exchange, Presentations,
Inventory, Change Notes, File Distribution, Calendars,
Online Help, Date Finder, Image Gallery, Slide Show,
Live Data View, Database, Management of Resources,
Events

- NGINX|Apache reverse proxy
- Customized components, active content
- InfoServer + packages + configurations
- tclhttpd
- Tcl 8.6 + standard packages
- OS (Linux) + tools (wkhtmltopdf, uuid, zip..)

- Webserver as front end to the user (audience)
- Content creation and management also by means of standard tools (editors, database tools etc)
- High level management functions

- HTML as default document format
- File/folder based storage
 - Easy editing and managing
 - mountable
- Arbitrary content must be possible
- Tclhttpd and reverse proxy
- Database for administration and customer data
 - SQLite3 (embedded, portable) or PostgreSQL (server)

- Folder structure with roles in mind

`server:` basic components

`serveradmin:` configuration + control

`siteadmin:` active components (Tcl)

`works:` payload data (files, database)

Folder hierarchy in „works“ branch

works

„content“ (fix name)

language (en|de|...)

user content (=DocFolder)

DocFolder:

- atomic unit of information storage
- any content
- usually at least one HTML file
- optional subfolders
 - file storage
 - components (images, css, JS,...)
- access controlled

`site`

contains the standard pages of the web surface
(index, sitemap, contact, ...)

`skin.$name`

multiple folders holding all components for branding
(css, templates, images, JS, ...)

➔ skins and standard pages are „works“

person

any kind of person

user

a person may be made a user

usergroup

usergroups have users as members

peergroup

peergroups have persons and/or usergroups

as members

- Persons (users), usergroups and peergroups are represented by numeric integer IDs in non-overlapping number ranges
- When a user logs in a list is built with the IDs of all usergroups and peergroups he is member of, directly or via a usergroup.
- The list of membership-IDs is part of the session

- Access is controlled on the level of a DocFolder
- A DocFolder has an entry in the database with access relevant properties („needed privileges“):
 - publisher, author, audience
 - write, upload, contribute
- The needed privileges are integer IDs.
- They are matched against the user/group IDs.


```
/coop/works/en/wikipage/vista.manual/_wiki.html
```

coop: tclhttpd-domain and rev. proxy

works: the „branch“

en: the language

wikipage: kind of document

vista.manual: the DocFolder

_wiki.html: the file to be shown

`/coop/works/en/wikipage/vista.manual/_wiki.html`

- Branch, language and name of the DocFolder are checked for access control

`/coop/works/en/wikipage/vista.manual/_wiki.html`

- The complete URL is used to determine the type of a document.
- It is compared to a list of regexp patterns in `/siteadmin/config/docs.xml`
- The first match delivers 3 results:
 - 1.Processor
 - 2.Skin
 - 3.PageTemplate

- Processors build any kind of document on call.
- Processors are selected by URL patterns while domain handlers are selected by only the first component of the URL.
- Processors are „branch handlers“ of any complexity.
- You may have any number and flavours of it.
- They may load other active code to fulfill their specific task.
- They are called by the domain handler with HTTP array, name of the skin and name of the pagetemplate

- With every call a namespace is created where all active components, like the processor, are loaded into.
- Code is executed in this namespace.

- Code is executed in the connection namespace only.
- Code is loaded from the siteadmin area only, not from arbitrary places or customer files.
- Tcl commands cannot be executed by customer documents, only by the processor.
- There is no evaluation or substitution at no time
- Variables or procedure calls are ignored and interpreted as text.
- The only active element in documents is a procedure call, executed in the connection namespace.
- Database access through stored queries only

- The processor called „ct_wiki`page`“ is the „working horse“ or „swiss knife“ for most of all HTML documents.
- Standard processing is:
 - Read the file from the DocFolder (tail of URL)
 - Read the pagetemplate from the skin folder
 - Insert the customer file into the pagetemplate
 - Process active code in the merged document
 - Deliver resulting document

```
<h3>Manuals</h3>
```

```
<h4>Linux</h4>
```

```
#[BookIndex -pattern "works/de/book.manual.linux" ]#
```

```
#[DocLink -href  
"/coop/works/de/wikipage/linux.manual.xubuntu1204.book/_i  
ndex.html?  
outname=Xubuntu_12_04_Installation&pagetemplate=pt_manual  
_1&skin=skin.vimacon.usermanual" -title "Linux XUbuntu  
Installationen" -span 0 ]#
```

```
#[DocLink -href  
"/coop/works/de/wikipage/plotter.usermanual/_cover.html"  
  
]#
```


- There is only one kind of „active code“
- A procedure call to a procedure in the processor or code which the processor loaded.
- The call is the name of the procedure which is then biased by the connection namespace.
- The call may have options as a „-key value“ list
- It is enclosed in „#[]#“
- Active code may be used in pagetemplates and the payload page

- Special features of the processor:
 - Branding by URL parameters
 - pagetemplate and skin can be overridden
 - Online editing (JS editor required)
 - Build a book from single documents
 - PDF export (via wkhtmltopdf)
 - ZIP export of „stand alone“ documents

- Several navigation elements
- Image, ImageLink, DocLink, UploadFileLink
- SetVariable, URLParameter, ThisSkin, Title
- Time, Date, URLTail
- SlideIndex, SlideTransport, ListOfSlides
- Conditional
- FileIndex, DataGrid, Editor_Button, PDF_Button, Calendar, TodoList, ShowBook, ShowBookTOC, WikiAttachments
- and many more

- The current state demonstrates that most of the web based tools and features, mentioned in the beginning, can be realized with this framework.
- 2 versions of InfoServer are currently in use.
- A large code cleanup is underway.
- Some features are realized only in parts.
- Search function is very primitive.
- Limitations and performance are not yet evaluated.

- The vision is, that one day a system becomes reality that can easily be rolled out by an hobbyist and that provides the features needed for a small business, a community or even a family and friends.

