FUJITSU Software BS2000 APACHE

Version 2.4A
November 2019

Readme

All rights reserved, including intellectual property rights.
Technical data subject to modifications and delivery subject to availability. Any liability that the data and illustrations are complete, current or correct is excluded. Designations may be trademarks and/or copyrights of the respective manufacturer, the use of which by third parties for their own purposes may infringe the rights of such owner.

Fujitsu and the Fujitsu logo are trademarks or registered trademarks of Fujitsu Limited in Japan and other countries. BS2000 is a trademark of Fujitsu Technology Solutions GmbH in Germany.

1 General

This readme describes the BS2000 specific differences and extensions, and supplements the available online documentation (describing the Unix version) in the packages APACHE:httpd-d, APACHE:modprl-d, and APACHE:modphp-d, which otherwise serve as the reference documentation for the product.
It contains information about installing, configuring and using the components

APACHE:httpd	V2.4A
APACHE:modperl	V2.4A
APACHE:modphp	V2.4A

under the operating system BS2000

APACHE:httpd(BS2000) V2.4A is based on the official release version of the web server Apache httpd 2.4.41 from the Apache Software Foundation.[1]

APACHE:modperl(BS2000) V2.4A is based on version 2.0.10 of the Apache/Perl module mod_perl[2], as well as the Perl runtime environment version 5.24.4 (see subcomponent PERL (BS2000) V52.4A).[3]

APACHE:modphp(BS2000) V2.4A is based on version 5.6.40 of the PHP script language[4]

Version V2.4A, compared to V2.2A, is a rather evolutionarily enhanced delivery with numerous updated subcomponents. The most important changes are described below:

[1] This product contains software developed by the Apache Software Foundation for the Apache HTTP server project (https://httpd.apache.org/).

[2] mod_perl (https://perl.apache.org/) is an Apache Software Foundation project (https://www.apache.org/) and is covered by the Apache Software License (an Open Source license).

[3] Perl is Copyright (C) 1993-2018, by Larry Wall and others. It is free software and redistributed under the terms of the "Artistic License".

[4] This product includes PHP software, freely available from <https://www.php.net/software/>; PHP is distributed under the PHP License, version 3.01, an Apache-style license. PHP includes the Zend Engine, freely available at <https://www.zend.com>.

1.1 Updated HTTPD server

APACHE V2.4A contains an HTTPD server based on Apache 2.4.41, which brings with it especially new access control directives. For the thereby necessary changes see https://httpd.apache.org/docs/2.4/upgrading.html (or the corresponding included file /opt/apache24/manual/upgrading.html), which contains also further information regarding the other changes.

1.2 Updated Perl support

The Perl interpreter contained in the delivery unit PERL V52.4A (which is part of APACHE V2.4A) made a big version leap from 5.8 to 5.24; also, some further CPAN modules have been added. For details see the Readme file for PERL 52.4A.

2 Overview

The supply unit APACHE consists of several package components whose modules can be combined. Depending on the required functionality, the installation can be limited to the components actually required.

2.1 BS2000 package files

The packages are supplied in the form of SINLIBs or SKULIBs for the X86 platform (with X86 binaries instead of /390 binaries) which can be installed using the /START-POSIX-INSTALLATION command.

APACHE V02.4A

(APACHE base package: directories and shared libraries)

/390:	SINLIB.APACHE.024
X86:	SKULIB.APACHE.024

APACHE:httpd V02.4A

(APACHE secure web server)

/390:	SINLIB.APACHE.024.HTTPD
X86:	SKULIB.APACHE.024.HTTPD
all:	SYSSDF.APACHE.024 SYSSPR.APACHE.024

APACHE:httpd-d V02.4A

(online documentation about APACHE web server)

all:	SINLIB.APACHE.024.HTTPD-DOC

APACHE:modperl V02.4A

(PERL APACHE module mod_perl)

/390:	SINLIB.APACHE.024.MODPERL
X86:	SKULIB.APACHE.024.MODPERL

APACHE:modprl-d V02.4A

(online documentation about the PERL APACHE module)

all:	SINLIB.APACHE.024.MODPERL-DOC

APACHE:modphp V02.4A

(PHP5 APACHE module libphp5)

/390:	SINLIB.APACHE.024.MODPHP
X86:	SKULIB. APACHE.024.MODPHP

APACHE:modphp-d V02.4A

(online documentation about the PHP5 APACHE module)

all:	SINLIB.APACHE.024.MODPHP-DOC

The name "APACHE:modphp-d" stands for a package in the form of, for example, a SINLIB.APACHE.024.MODPHP-DOC which is installed via the POSIX-INSTALLER as product "APACHE" and package of product "MODPHP-D".

2.2 Dependencies of the packages

As some of the packages are mutually interdependent, the resulting package dependencies must be observed during the installation (and vice versa at deinstallation).

APACHE (Base package)
requires:
POSIX-SH
prerequisite for:
PERL APACHE:httpd APACHE:modperl APACHE:modphp

APACHE:httpd (Secure Web Server)
requires:
APACHE
prerequisite for:
APACHE:httpd-d APACHE:modperl APACHE:modprl-d
APACHE:modphp APACHE:modphp-d

APACHE:httpd-d (online docs for APACHE:httpd)
requires:
APACHE:httpd
prerequisite for: --

APACHE:modperl (Perl scripting modules for APACHE:httpd)
requires:
PERL APACHE:httpd
prerequisite for: --

APACHE:modprl-d (online docs for APACHE:modperl)
requires:
APACHE:httpd
prerequisite for: --

APACHE:modphp (PHP scripting modules for APACHE:httpd)
requires:
APACHE:httpd
prerequisite for: --

APACHE:modphp-d (online docs for APACHE:modphp)
requires:
APACHE:httpd
prerequisite for: --

Only the "APACHE" base package and the package "APACHE:httpd" is required for the Apache web server. However, if you wish to install the mod_perl module, the components APACHE, PERL and APACHE:httpd must be installed before you can install APACHE:modperl. In order to install all the supplied components, you should keep to the following sequence in order to make sure that the package dependencies are observed:

–	APACHE	(Base package with common run-time)
–	PERL	(Perl scripting language run-time)
–	APACHE:httpd	(Secure Web Server)
–	APACHE:httpd-d	(Online docs for APACHE:httpd)
–	APACHE:modperl	(Scripting modules for APACHE:httpd)
–	APACHE:modprl-d	(Online docs for APACHE:modperl)
–	APACHE:modphp	(PHP scripting modules for APACHE:httpd)
–	APACHE:modphp-d	(Online docs for APACHE:modphp)

Observe the dependencies in reverse order when deinstalling: before you can deinstall APACHE:httpd, the packages APACHE:httpd-d, APACHE:modperl, APACHE:modprl-d, APACHE:modphp and APACHE:modphp-d must be deinstalled.

All packages are optional: however, if the function of a package is to be used and the package is to be installed, the prerequisite packages must be installed first.

2.3 Additional software prerequisites

- See Release Notice APACHE(BS2000) V2.4A

2.4 POSIX installation

2.4.1 Properties of POSIX installation procedure

The installation procedure has the following properties and behaviors:

- The installation packages use the interaction with the user in order to make inquiries (to inquire configuration parameters or yes/no confirmations) and to display progress messages during the installation.

- If possible and if required, background processes (such as the Apache Web Server) are automatically started after a successful installation and stopped before deinstallation.

- Each installation package checks during the installation as to whether the dependencies to the prerequisite packages have been met. Should dependencies not be met, the installation is aborted with an error message.

- Each installation package checks during installation as to whether there is enough space on the selected file system for the package to be installed. It aborts with a "not enough space" error message before the actual installation begins.

- All configuration questions are asked at the beginning of the installation; the installation then runs as far as the end without inquiries (except for: %PLEASE ACKNOWLEDGE).

- In BATCH installation mode, standard responses are used, and background processes, such as the Apache Web Server, are not automatically started. Example parameter files for installing and deinstalling all APACHE packages in BATCH mode are in the installation library of the base package.

- If Apache Web Server add-on products (APACHE:modphp and APACHE:modperl) are installed, the installed module can be optionally added and activated, When deinstalling it is automatically deactivated and removed.

- Deinstallations partially test whether the package to be deleted is a prerequisite for another package (e.g., the APACHE base package is a prerequisite for APACHE:httpd and PERL and if it is deleted, these packages no longer operate correctly). The deinstallation can be aborted on request without endangering the mutual consistency of the packages.

2.4.2 Configuration files’ upgrade handling

The strategy to prevent losing configuration files modified by the user when upgrading to a new web server version is as follows:

- During the installation, configuration files are installed as original under another file name. With APACHE:httpd (Apache Web Server) these originals are in a subdirectory $IPATH/conf/original/... (see chapter 5.2).

- After the installation, the originals are copied to their final position $IPATH/conf/... if there is no user-adapted configuration file already at that position.

- At deinstallation, only the configuration files are removed which have not been modified in comparison to the original. All the changed copies remain at their position in $IPATH/conf/..., and are reused during reinstallation.

- If you wish to delete the product completely or install anew without the old configuration files, you must manually delete (after deinstallation of all of the components) the $IPATH directory structure (or delete specifically the files which you do not want to reuse).

- An automatic migration from an existing Apache 2.2 (or even 1.3) configuration to an Apache 2.4 configuration is not provided. Except for the changed access control directives one should be able to take over the configuration files from an Apache 2.2 installation without bigger changes.

3 Port assignments of the web server

The Apache web server assigns the following port numbers by default:

Port:	Protocol:	Explanation:
80/tcp	http	Hypertext Transfer Protocol
443/tcp	https	SSL/TLS secure http

(These default values can of course be changed in the configuration).

Depending on the security policy, you should first decide whether all the network connections to the web server should be generally permitted via the command

```
 /BCOPTION ADD-SERVER-PORTS=(80,443)
```

(See also /SHOW-BCAM-SERVER-PORTS).

Apache also makes connections to other computers on the following ports:

Port:	Protocol:	Explanation:
53/udp|tcp	dns	Domain name Services
20+21/tcp	ftp	When using mod_proxy_ftp
80/tcp	http	When using mod_proxy_http
443/tcp	https	When using mod_proxy_connect
8009/tcp	ajp13	Apache JServ Protocol 1.3 (when using a remote tomcat)

As a functioning DNS name resolution for Apache is very important, a functioning DNS name server (known to BCAM or generally permitted via the command

```
/BCOPTION ADD-REMOTE-SERVER-PORTS=(53)
```

should be named in the DNS resolver configuration file /etc/resolv.conf.

4 Installation

4.1 APACHE web server ID

The Apache Web Server requires for its operation a separate user ID which should be configured before installing the package APACHE:httpd. It is used for security purposes, as the individual web server processes should have as few privileges as possible and be optimally separated from the other existing user IDs. This prevents the server machine from being open to attacks from the network when using CGI programs. Only the Apache manager process which controls load control (creation and termination of the server processes), has more rights under the SYSROOT-ID, but it never comes into contact with the requests coming from the network.

The default name of the web server ID is "APACHE" but you can specify any other name during the interactive installation.
When installing in BATCH mode, if this ID does not exist at the time when installation begins, it is automatically created with suitable default values on the default pubset.

This web server ID should have the following attributes:

- The ADDRESS-SPACE-LIMIT for the ID (and also for SYSROOT) should be at least 96MB. (However, when using the SQL database interfaces to SESAM and ORACLE, see also the requirements for the respective databases e.g. 512MB for ORACLE users)

- The ACCOUNT of the web server ID or its default JOBCLASS should, if possible, permit the NO-CPU-LIMIT and START-IMMEDIATE privilege.

- The required ACCOUNT of the web server ID must be set as POSIX default account:

```
/MODIFY-USER-ATTRIBUTES -
/ USER-IDENTIFICATION=apache,-
/ ADDRESS-SPACE-LIMIT=96,-
/ ACCOUNT-ATTRIBUTES=-
/   *MODIFY(ACCOUNT=webserve,-
/               PRIVILEGE=*PAR(NO-CPU-LIMIT=*YES,-
/                              START-IMMEDIATE=*YES -
/                             ),-
/           POSIX-RLOGIN-DEFAULT=*YES)
```

- The default JOBCLASS of the web server ID should permit as LIMIT at least as many tasks as the APACHE server may create for child and children's child (CGI) processes.

- SYSROOT should also be allowed to use this JOBCLASS. If it is entered in /etc/default/APACHE24.httpd, the Apache monitor process (i.e. the SYSROOT web server task) starts with this JOBCLASS instead of the default job class.

- The POSIX home directory must not be /home/gast (or the default POSIX DIRECTORY; see /SHOW-POSIX-USER-DEFAULTS). As the ID is not designed as a standard login, you can select, e.g. the $IPATH base directory /opt/apache24.

- The numeric POSIX user ID must not be 0 or 100 (or the default POSIX USER NUMBER; see /SHOW-POSIX-USER-DEFAULTS). It should have a numeric value, which differs from all other POSIX user IDs if possible.

- The numeric POSIX Group ID must not be 0 or 100 (or the default POSIX GROUP NUMBER; see /SHOW-POSIX-USER-DEFAULTS). It should have a numeric value, which, if possible, differs from all other POSIX Group IDs. It is recommended to put this value in the file /etc/group with a name, e.g. "httpd".

As it is not easy to determine a free POSIX user or group number using BS2000 commands, this can be done very easily (having installed the base component APACHE) using the tool /opt/apache24/sbin/start-daemon.

The tool option "-=80-" determines an unused POSIX user ID / group ID pair larger than or equal to 80 (where the user ID is the same as the group ID) which is not used by any other ID.

Example for configuring the web server user ID:

```
============================================================
/EXECUTE-POSIX-CMD '/opt/apache24/sbin/start-daemon -=80-'
82

/ REMARK "Please insert this POSIX user/group id "
/ REMARK " in the commands below:"

/ADD-USER -
/  USER-IDENTIFICATION=apache,-
/  LOGON-PASS=C'<random text>',-
/  ADDRESS-SPACE-LIMIT=96,-
/  ACCOUNT-ATTRIBUTES= -
/             *PAR(ACCOUNT=webserve,CPU-LIM=*MAX,-
/                  PRIVILEGE=*PAR(NO-CPU-LIMIT=*YES,-
/                                 START-IMMEDIATE=*YES -
/                                ),-
/                  POSIX-RLOGIN-DEFAULT=*YES -
/                 ),-
/  MAIL-ADDR='APACHE web server Daemon'

/MOD-POSIX-USER-ATTR USER-ID=apache,-
/                    USER-NUM=82,-
/                    GROUP-NUM=82,-
/                    COMMENT='APACHE web server Daemon',-
/                    DIRECTORY='/opt/apache24',-
/                    PROGRAM=*SHELL

/EXECUTE-POSIX-CMD 'echo >>/etc/group httpd::82:'
============================================================
```

If the default pubset of the user id is different from the HOME pubset, the ADD-USER and MOD-POSIX-USER-ATTR commands must be issued a second time with ",PUBSET=XXXX" appended.

4.2 BS2000 job attributes

In order to operate background processes (in Unix also known as "daemons") such as web servers, it is recommended to configure a dedicated job class which provides the required attributes (NO-TIME-LIMIT, START=IMMED, LIMIT=...).

The job attributes can be modified further in the POSIX startup configuration file /etc/default/APACHE24.httpd:

- The standard job name for the APACHE server tasks is JOB-NAME="APACHE24". This name can be changed here (to a valid BS2000 job name).

- The preset JOB-CLASS is JOBCLASS="*STD", i.e. the default job class of the SYSROOT user. If a dedicated job class has been configured for background processes (and the SYSROOT and the APACHE IDs are allowed to use them), they can be specified here.

- If the default value ACCOUNT="*STD" is retained, when starting the web server task the "DEFAULT-ACCOUNT-# FOR REMOTE-LOGIN" is used for SYSROOT. Alternatively, a special valid ACCOUNT can be specified which, for example, was configured for background processes and permits the START=IMMED attribute.

- The START_APACHE_AT_POSIXSTART value decides whether the web server is to be automatically started or not at the next POSIX start. One of three values is allowed:

  - `START_APACHE_AT_POSIXSTART="yes":`
    starts APACHE with each POSIX start
  - `START_APACHE_AT_POSIXSTART="no":`
    No start, issues warning on console
  - `START_APACHE_AT_POSIXSTART="never":`
    No start, no message on console.

  This variable has the default value "no" in BATCH mode installations, and "yes" in interactive installations.

4.3 Apache directory structure

The main paths of the APACHE:httpd package:

Path:	Function:
$IPATH/	Base directory of the Apache installation
$IPATH/sbin/httpd	Apache Web Server Executable
/opt/bin/apache2ctl	Wrapper script to start and stop the server
$IPATH/conf/httpd.conf	Main configuration file of the Apache web server
$IPATH/conf/extra/httpd-*.conf	Included configuration parts
$IPATH/modules/	Directory with the loadable Apache modules
$IPATH/lib/lib*.so	Jointly used shared libraries
$IPATH/conf/conf.d/*.load	3rd -party "LoadModule" parameter files
$IPATH/conf/conf.d/*.conf	3rd party configuration files
$IPATH/logs/	Directory of Apache log files (Symlink) -> /var/opt/APACHE24.httpd/logs/
$IPATH/htdocs/	Standard document root of the web server
$IPATH/manual/	Directory tree for optional online documentation packages

4.4 POSIX installation in interactive mode

The packages are installed to the POSIX file system in the sequence of their above cross-dependencies with the POSIX installer, i.e. first of all the base package, then PERL and/or APACHE:httpd, then the further add-on packages for APACHE:httpd. The calling user ID must have user ID 0 in POSIX in order to install directories and files (e.g. TSOS or SYSROOT).

Example:

```
=============================================================
              BS2000 POSIX package installation

IMON support ?      : Y  (y) mandatory for official package
                         (n) private package (SINLIB...)

name of product     : APACHE
package of product  : HTTPD     (optional for certain products)

version of product  :           (format Vmm.n or mmn)

correction state    :           (format aso, optional)

installation userid:            (mandatory for no IMON support)


The definition of a installation path is optional
for this product. Please enter the full pathname of
the wanted installation directory:
/opt/apache24


install: DUE      help: F1     terminate: F2
=============================================================
```

The installation’s base directory (the default $IPATH being: "/opt/apache24") can be selected when installing the base package "APACHE". When installing the other APACHE packages (APACHE:httpd etc.) the same directory must be specified. However, retaining the standard value is recommended.
At the beginning of an interactive installation (i.e. /START-POSIX-INSTALLATION without FILE= specification) of the packages APACHE:httpd and the module packages APACHE:modperl and APACHE:modphp, the POSIX-INSTALLER asks a number of configuration questions and then begins with the actual installation. (Until this point, you can abort the installation without any danger by entering 'q' or K2 without changing anything on your system.)
Your answers are saved until the end of the installation and then used for configuration. The answers are subjected to a simple validity check, if possible, and any obviously incorrect entries are prompted for correction.

After installing the files in the POSIX file system and adapting the configuration files, the web server is started (optionally, see Prompt "AutoStartApache") in a SYSROOT /ENTER job. (Even if the installation was made under a TSOS-ID, the web server is started in a new task under SYSROOT in order to ensure secure operations under a non-TSOS-ID.)

4.5 POSIX installation in BATCH mode

In order to simplify the installation, an installation in BATCH mode is partially supported. While in an interactive installation a configuration is made during the installation, this is not possible in BATCH mode: the packages are installed with default configuration, but a subsequent configuration and server start must be done manually after the installation.

To make BATCH installation easier, the base package contains a library element S//LIB.APACHE.024(***PKGADD.APACHE.024***,S) which can be used as parameter file for the POSIX-INSTALLER call. Copy it to a file, edit it and comment the components that you don't want to install by prefixing a '#'.

You can then start the batch installation via a call such as:

```
  /START-POSIX-INSTALLATION INPUT-INTERFACE=-
  /  *FILE(FILE-NAME=PKGADD.APACHE.024)
```

As important information about problems can be displayed during the batch installation (low system limits, incorrect file permissions, non-configured IDs etc.), it is recommended to have the entire POSIX-INSTALLER output written to a file and then evaluate it.
The standard answers to the questions during the installation can already be modified and pre-assigned before starting the batch installation in the following 'S' library elements:

S//LIB.APACHE.024.HTTPD(response.APACHE-HTTPD.024,S):
ServerName, Port, ServerAdmin etc.

S//LIB.APACHE.024.MODPERL(response.APACHE-MODPERL.024,S):
EnablePerlModule? RestartApache?

S//LIB.APACHE.024.MODPHP(response.APACHE-MODPHP.024,S):
EnablePHPModule? RestartApache?

SINLIB.APACHE.024.MODPERL-DOC(response.APACHE-MODPERL-DOC.024,S):
RestartApache?

SINLIB.APACHE.024.MODPHP-DOC(response.APACHE-MODPHP-DOC.024,S):
RestartApache?

(Please note that the standard file permissions of the installation libraries (SHARE=YES, ACCESS=READ) do not permit editing library elements)

4.6 POSIX deinstallation

The deinstallation must also be made under a user ID with POSIX User ID 0 (e.g. TSOS or SYSROOT) in order to stop the web server and remove directories and files.

The interactive deinstallation is started by calling /START-POSIX-INSTALLATION, selecting "Delete Packages from POSIX" and marking the required product component; delete the packages in the reverse sequence of the installation.

Analog to batch installation, the base package also has a library element S//LIB.APACHE.024(***PKGRM.APACHE.024***,S) as example parameter file for the batch deinstallation with the POSIX-INSTALLER.

Note:
If, for a product upgrade, the deinstallation and reinstallation is **automated** via IMON, this occurs from the APACHE package viewpoint in batch mode, and that is why, with reinstallation, the variable START_APACHE_AT_POSIXSTART="**no**" is set in the parameter file /etc/default/APACHE24.httpd.
After the product upgrade, you must thus edit this file if necessary, set the value again to START_APACHE_AT_POSIXSTART="**yes**" and execute a web server start with

```
  # apache2ctl start
```

or under BS2000:

```
  /START-APACHE
```

5 Operations

5.1 Start / stop the web server

In normal Apache documentation, the use of the "apachectl" script is recommended to start, stop and restart the APACHE:httpd. This script can also be used with BS2000 Apache, but we recommend using the /opt/bin/apache2ctl wrapper instead.

The reasons are:

- Start under SYSROOT-ID:
  Even if the caller is TSOS, the web server monitor process is started under SYSROOT. This ensures that no part of the web server ever has increased TSOS privileges. (The actual web server processes are run with even fewer privileges under the 'APACHE' ID).

- Detach from current job attributes and from the terminal:
  the web server is started in a separate /ENTER job (its JOB-NAME and other attributes can be configured). It is thus detached from the terminal as well and can thus optionally already be started at the end of the installation.

As the restart function of the "apachectl" script can also start the server (if none is running), the restarts should be executed via the /opt/bin/apache2ctl wrapper as well if the caller is not already logged in as SYSROOT:

```
  # apache2ctl graceful
```

or under BS2000:

```
  /RESTART-APACHE *GRACEFUL
```

Is in /etc/default/APACHE24.http the variable START_APACHE_AT_POSIX_START set to “no” or “never”, then not only the Apache start is suppressed at POSIX startup, but also on invocation of “apache2ctl start” only a warning message is issued, but Apache is not started. However, when one invokes apache2ctl with parameter “START”, then after printing a warning message Apache is nevertheless started. When invoking with “graceful” or “restart” Apache is started without this warning message.

5.2 Configuration files of the Apache web server

If you would like to adapt the configuration files of the web server, you should know which files Apache evaluates in which sequence. The base directory of the installation (that you specified in the POSIX-INSTALLER) is known as $IPATH. The structure of the configuration files corresponds to the standard Apache layout of the Apache Software Foundation, but is different from that of many other Linux distributions.

- The main configuration file is $IPATH/conf/httpd.conf

- The list of the standard modules to be loaded is in $IPATH/conf/extra/httpd-modules.conf: you can bracket out unused Apache modules by placing a '#' in front of the affected line.

- The delivery status has a number of rarely used modules bracketed in order to avoid causing unnecessary loads on the main memory and CPU. Should you require the appropriate functionality, simply remove the comment characters and restart the web server.

- Additional configuration files are read in (via explicit "Include" directives in the $IPATH/conf/httpd.conf) from the directory $IPATH/conf/extra/httpd-*.conf.

- The load module directives of the optional add-on modules mod_php5 and mod_perl (and possible extensions of the user) are either bracketed or active in $IPATH/conf/conf.d/<name>.load, depending on the specification during the installation. These configuration files are read in from the $IPATH/conf/httpd.conf having loaded the standard Apache modules with the directive "Include conf/conf.d/*.load"

- The configuration directives of the optional add-on modules (and possible extensions of the user) are included by a directive "Include conf/conf.d/*.conf".

After each change on the list of the modules to be loaded, you should call "apache2ctl graceful" in order to have the web server reread the modified configuration.

5.3 SSL/TLS operation

The web server package installs private key and certificate of a Dummy-CA ("Snakeoil CA"), with which the subsequentially described utility mkcert.sh enables the creation of a test certificate.
This certificate is only used to run a web server with SSL before an official certificate has been applied for from a Trust Center. Each attempt to access your server via SSL results of course in a browser warning unless you install the dummy CA certificate in your browser (but this you should do at the most only in the test phase!)
Under no circumstance should you operate a productive server with this certificate!
The Dummy-CA certificate is 10 years valid, i.e. at the press date of this document until 2029.

The SSL configuration consists of the following files below the Apache configuration directory:
`/opt/apache24/conf/`:

path	**function**
`extra/httpd-modules.conf`	Determines whether SSL module is loaded
`extra/httpd-ssl.conf`	Configures SSL module if loaded
`ssl.crt/server.crt`	The SSL certificate for the web server
`ssl.key/server.key`	Private key for SSL certificate
`ssl.csr/server.csr`	SSL Certificate Signing Request
`ssl.crl/server.crl`	(Optional) certificate revocation list
`original/ssl.crt/snakeoil-ca-rsa.crt`	RSA certificate of the Dummy-CA

In order to apply for a correct certificate, you can create a CSR (*certificate signing request*), using the data of your machine, in the so-called PEM format which you transfer to the Trust Center when applying for an SSL certificate. A simple script (mkcert.sh) is supplied which helps you to create a CSR and creates a temporary snakeoil-signed dummy certificate.

Creating/Replacing the test certificate

Execute the following steps when you want to operate your server with an official SSL certificate (issued by a Certificate Authority (="CA") which your standard browsers trust) or with a test certificate issued by a Dummy-CA (see above):

- Invoke the "mkcert.sh" utility in order to create a CSR (*certificate signing request)*. The inquired values appear in the CSR and thus in your later official certificate. When the prompt "Common name (e.g. FQDN)" appears, enter your correct DNS host name (this value is compared against the host name in the https:// request by the client when making the SSL connection):

```
==============================================================
# cd /opt/apache24/support/
# ./mkcert.sh
SSL Certificate Generation Utility (mkcert.sh)
Copyright (c) 1998-2000 Ralf S. Engelschall, All Rights
Reserved.
...
STEP 2: Generating X.509 certificate signing request
You are about to be asked to enter information that will be
incorporated into your certificate request.
What you are about to enter is what is called a
Distinguished Name or a DN.
There are quite a few fields but you can leave some blank.
For some fields there will be a default value.
If you enter '.', the field will be left blank.
-----
1. Country Name           (2 letter code) [XY]:        DE
2. State or Province Name (full name)     [Snake Desert]:
                                                  Bavaria
3. Locality Name          (eg, city)      [Snake Town]:
                                                   Munich
4. Organization Name      (eg, company)   [Snake Oil,Ltd]:
                                                 Intranet
5. Organizational Unit Name (eg, section)[Webserver Team]:
6. Common Name            (eg, FQDN)   [www.snakeoil.dom]:
                                         hostname.dom.ain
7. Email Address       (eg, name@FQDN) [www@snakeoil.dom]:
                               WebMaster@hostname.dom.ain
8. Certificate Validity   (days)          [365]:       730
-----
…
Are you satisfied with the entered data, and do you want to in-
stall the generated private key and certificate as your new
server key and certificate? [y/n;N]: y
Certificate: /opt/apache24/conf/ssl.crt/server.crt
Private Key: /opt/apache24/conf/ssl.key/server.key
Signing Req: /opt/apache24/conf/ssl.csr/server.csr

==============================================================
```

You now have a certificate application /opt/apache24/conf/ssl.csr/server.csr as well as a private key /opt/apache24/conf/ssl.key/server.key and a temporary certificate /opt/apache24/conf/ssl.crt/server.crt (already issued on your machine, however from the non-trustworthy "Snakeoil CA"). Stop the web server completely and start it again so that the new certificate is activated:

```
========================================================
# apache2ctl stop
# apache2ctl start
========================================================
```

- Apply for an SSL certificate in PEM format by using the certificate application /opt/apache24/conf/ssl.csr/server.csr.

- If you have received the certificate issued, simply replace the temporary certificate /opt/apache24/conf/ssl.crt/server.crt via a copy of your official PEM certificate and start the server anew so that the new certificate is activated:

```
========================================================
# apache2ctl stop
# apache2ctl start
========================================================
```

- When replacing the certificate, do *not* use the "graceful" or "restart" functions, but stop the web server completely and start it anew as shown in the example above.

Deactivate SSL

We do not recommend deactivating the SSL interface as an encrypted SSL communication increases security and prevents transferred data from being tapped during transport and being misused.

If you nevertheless want to configure a pure HTTP server without SSL, simply carry out the following steps:

- Look in the configuration file "conf/extra/httpd-modules.conf" for the line:
  `LoadModule ssl_module modules/mod_ssl.so`
  and place a '#' character in front of the line.

- Restart the web server with
  ```
   # apache2ctl graceful
  ```
  or under BS2000:
  ```
   /RESTART-APACHE MODE=*GRACEFUL
  ```

the server is thus restarted without SSL support.

6 SDF support for APACHE:httpd

The supply unit APACHE also contains a SDF syntax file which is used to execute the start, restart, stop and status inquiry commands directly from TSOS or SYSROOT from the BS2000-ID as SDF commands. The implementation is contained in the procedure library SYSSPR.APACHE.024 and the SDF syntax file SYSSDF.APACHE.024.

List of SDF commands:

===

/START-APACHE
Requires *POSIX user ID 0*
Domain: NETWORK-MANAGEMENT

The Apache web server is started as SYSROOT /ENTER job. If a web server is already running, an error is reported.

===

/STOP-APACHE [MODE=<u>***FORCED**</u>|*GRACEFUL]
Requires *POSIX user ID 0*
Optional argument: (Default: **MODE=*FORCED**)

***FORCED**

A running web server is stopped forcefully (i.e. aborts all running transfers). If no web server is running, an error is reported.

*GRACEFUL

A running web server is stopped *gracefully* (i.e. all running transfers still continue within the configured *GracefulShutdownTimeout* time and only then is the Apache monitor process stopped). If no web server is running, an error is reported.

===

/RESTART-APACHE [MODE=***FORCED**|*GRACEFUL]
Requires *POSIX user ID 0*
Optional argument: (Default: **MODE=*FORCED**)

***FORCED**

A running web server is restarted forcefully (i.e. aborts running transfers). If no web server is running, a /START-APACHE is executed.

*GRACEFUL

A running web server is restarted *gracefully* (i.e. already running transfers still continue). If no web server is running, a /START-APACHE is executed.

===

==

/SHOW-APACHE-STATUS [INFORMATION=***STD**|*ALL]
Requires no privilege.
Optional argument: (Default=***STD**)
***STD**
The Apache status is displayed (overview of network and CPU consumption and currently running web server processes)
*ALL
The entire Apache status is displayed (additional details for the individual processes and information about the SSL session cache).

The command /SHOW-APACHE-STATUS assumes that the Apache web server has been configured so that an access to the mod_status output is possible under http://localhost/server-info.

==

/EXECUTE-APACHE-CMD PARAMETERS=<c-string-with-lower-case>
Alias: **/APACHE2CTL**
Requires *POSIX-UserId 0*
Argument: (No Default)
'c-string'
The POSIX command apache2ctl is invoked with the contents of the character string as its argument. This enables special forms of the command such as

```
/APACHE2CTL  PAR='-f conf/httpd-VHOST2.conf'
```

==

Prerequisite for using the /SHOW-APACHE-STATUS command is the configuration of the Apache module mod_status by:

- removing the comment character in front of
  `#Include conf/extra/httpd-info.conf`
  in the configuration file $IPATH/conf/httpd.conf as well as

- editing the configuration file $IPATH/conf/extra/httpd-info.conf (you can define the allowed host names, IP addresses or domain names to access the web server status page). The command /SHOW-APACHE-STATUS accesses the web server under the name "localhost", i.e. the two lines which regulate access for mod_info and mod_status should be changed to:

  Require ip 127

  It is not recommended to permit access for all IP addresses without any restrictions.

- If the web server is not listening on the standard port 80, the port number in the line
  `STATUSURL="http://localhost:80/server-status"`
  in the file $IPATH/sbin/apachectl must be modified to reflect the actual port number.

- If more details about the status of the web server are desired, you can add the "**ExtendedStatus on**" directive to the configuration file $IPATH/conf/extra/httpd-info.conf.
  With INF=*STD display, this shows additional network and CPU load details; with *ALL display, it also shows the requests’ URLs.

- After adapting the configuration, restart the web server with
  # **apache2ctl graceful**
  or under BS2000 with
  /**RESTART-APACHE MODE=*GRACEFUL**
  in order for the changes to become effective.

Example:

```
==============================================================
/SHOW-APACHE-STATUS INF=*STD

             Apache Server Status for localhost


 Server Version: Apache/2.4.41 (BS2000) OpenSSL/1.1.1c PHP/5.6.40
mod_perl/2.0.10 Perl/v5.24.4
 Server Built: Aug 22 2019 16:04:37
 -------------------------------------------------------------------------
 Current Time: Wednesday, 06-Nov-2019 19:23:16 MET
 Restart Time: Thursday, 24-Oct-2019 10:13:26 MDT
 Parent Server Config. Generation: 1
 Parent Server MPM Generation: 0
 Server uptime:  13 days 10 hours 9 minutes 50 seconds
 Server load: -1.00 -1.00 -1.00
 Total accesses: 1 - Total Traffic: 1 kB - Total Duration: 82
 CPU Usage: u16.81 s13.85 cu0 cs0 - .00264% CPU load
 8.62e-7 requests/sec - 0 B/second - 1024 B/request - 82 ms/request
 1 requests currently being processed, 3 idle workers
 __W_............................................................
 ................................................................
 ................................................................
 ..........................................................

 Scoreboard Key:
 "_" Waiting for Connection,  "S" Starting up,  "R" Reading Request,
 "W" Sending Reply,  "K" Keepalive (read),  "D" DNS Lookup,
 "C" Closing connection,  "L" Logging,  "G" Gracefully finishing,
 "I" Idle cleanup of worker,  "." Open slot with no current process
=========================================================
```

7 Additional tools

Some Open Source utilities are installed together with the APACHE base package, which can help you when configuring and operating the web server.

- /opt/apache24/support/iconv:
  (See https://www.gnu.org/software/libiconv/)
  The GNU iconv utility is used to convert text documents between numerous different source and target character sets. It is mainly useful for converting between an ASCII-based character set (ISO_8859-*, UTF*) and an EBCDIC-based character set (OSD_EBCDIC_DF*, but also national versions).
  The file is also linked as /opt/bin/giconv.
  The list of the supported character set names and aliases is displayed when calling "/opt/bin/giconv -l".

- /opt/apache24/support/openssl:
  (See https://www.openssl.org/)
  This tool can be used to display SSL certificate details. It is also required by the "mkcert.sh" script to create SSL certificates.
  The file is also linked as /opt/bin/openssl.

- /opt/apache24/support/sqlite3:
  (See https://www.sqlite.org/)
  This command line utility is used to create, edit and display SQLite3 databases, e.g. to use SQLite3-based user authentication (see How To further on).
  The file is also linked as /opt/bin/sqlite3.

- /opt/apache24/support/nc: (netcat)
  (See http://nmap-ncat.sf.net/)
  This tool is used internally used when the "apache2ctl status" function is used to display the web server status and no "lynx" utility was found in the search path ("lynx" is a text mode browser and can issue HTTP requests and convert the HTML response to normal text format, but it is not available as standard in POSIX).

The following Open Source utility is installed together with the web server component APACHE:httpd:

- /opt/apache24/sbin/cronolog:
  (See http://cronolog.org/)
  An improved logfile rotation utility used to automatically distribute the web server log file(s) to periodically changing output files, without the need for a web server restart. For example, an apache directive
  ```
  TransferLog "|/opt/apache24/sbin/cronolog logs/%Y/%m/%d/access_log"
  ```
  can distribute the output to an implicitly created daily directory hierarchy, e.g., on 31st December 2019:
  ```
  logs/2019/12/31/access_log
  ```
  and after midnight:
  ```
  logs/2020/01/01/access_log
  ```

8 EBCDIC character set support

The APACHE web server in BS2000 assumes by default that HTML documents are encoded in the OSD_EBCDIC_DF04_1 character set, and converts them internally to ISO_8859-1 for the browser. This configuration can be found in the configuration file /opt/apache24/conf/httpd.conf:

```
================================================
 …
 AddDefaultCharset ISO-8859-1
 <IfModule charset_lite_module>
   CharsetSourceEnc OSD_EBCDIC_DF04-1
   CharsetDefault   ISO-8859-1
   CharsetOptions ImplicitAdd
 </IfModule>
 …
================================================
```

This default can be adapted in the configuration when documents in other codes exist, or if they are to be converted to other character sets (in particular UTF-8 or ISO_8859-15) for browsers. If, for example, all HTML documents in the directory /opt/apache24/htdocs/utfdocs/ are encoded in the character set OSD_EBCDIC_DF04_15 (this character set contains among others the Euro character), and should be supplied to the browser as UTF-8 documents, the following configuration extension can be used:

```
================================================
<Directory /opt/apache24/htdocs/utfdocs/>
  <Files *.html>
    CharSetOptions ImplicitAdd
    CharsetSourceEnc OSD_EBCDIC_DF04_15
    CharsetDefault   UTF-8
    AddDefaultCharset UTF-8
  </Files>
</Directory>
================================================
```

Such a configuration can be created under a name such as /opt/apache24/conf/conf.d/utfdocs.conf and becomes effective by restarting the web server with "/opt/bin/apache2ctl graceful".

By the way, the character set names in the directive AddDefaultCharset, CharsetSourceEnc and CharsetDefault are evaluated by the underlying GNU libiconv library and utility. You can display the list of accepted character set names by invoking the GNU iconv utility ("/opt/bin/giconv -l") supplied with APACHE.

8.1 What must be done with incorrectly converted files?

However, if you have copied from an ASCII machine a set of UTF documents in ***text*** mode to the POSIX file system, then this is an invalid conversion: due to the "text mode", you implicitly applied a conversion ISO_8859-1 => OSD_EBCDIC_DF04_1, but the material was not really encoded in the character code ISO_8859-1, but in UTF-8. This incorrect conversion should be better corrected via renewed ***binary*** copying of the UTF-8 files and assigning attributes as "CharsetSourceEnc UTF-8". If need be you can use a little "white lie" and tell APACHE that the files exist in the character set OSD_EBCDIC_DF04_1 and are to be supplied as ISO_8859-1 to the browser. Then override for the browser the ISO_8859-1 character set indicator with an explicit type with UTF-8 character set:

```
=======================================================
<Directory /opt/apache24/htdocs/utfdocs/>
  <Files *.html>
    CharSetOptions ImplicitAdd
    CharsetSourceEnc OSD_EBCDIC_DF04_1
    CharsetDefault   ISO_8859-1
    # "White lie": the files are really UTF-8:
    ForceType text/html;charset=UTF-8
  </Files>
</Directory>
=======================================================
```

9 HowTos

9.1 How to use disk cache

The new disk_cache module in Apache2 can be used both as cache for the proxy module (sensible with forward and reverse proxy cases) as well as a generic access cache which, for example, saves dynamic resources in a disk cache thus reducing the web server load. The following example shows how the dynamic output of scripts from a specific directory (and if "cacheable") can be cached for a specific time on the server. If the same URL is requested again within the expiry time, it does not have to be calculated again, but can be fetched from the cache. As configuration you could save the following directives as /opt/apache24/conf/conf.d/mod_cache.conf:

```
================================================
# Example /opt/apache24/conf/conf.d/mod_cache.conf
# NOTE: The directory /var/opt/APACHE24.httpd/cache must
# be created and chown'ed to the APACHE user.
# The cache is filled but never cleaned by APACHE!
# Use the htcacheclean utility to clean it regularly.
<IfModule disk_cache_module>
  CacheEnable disk /php-scripts/
  CacheRoot     /var/opt/APACHE24.httpd/cache
  CacheDirLevels 3
  CacheDirLength 3
  CacheMaxFileSize 1000000
  CacheDefaultExpire      600
  CacheMaxExpire          7200
  ExpiresDefault          "now plus 2 hours"
</IfModule>
# The cache module should be used with care and can be
# used to circumvent Require directives.
# You should not enable caching for any content to which
# you wish to limit access by client host name,
# address or environment variable.
================================================
```

In addition, you must activate the "LoadModule" directives for the two modules cache_module and disk_cache_module in the configuration file /opt/apache24/conf/extra/httpd-modules.conf by removing the comment character, and then restart the web server via "/opt/bin/apache2ctl graceful".

In contrast to Apache V1.3 it is not the web server that handles the periodical cleanup work in the cache directory but a dedicated htcacheclean utility (its call syntax is documented in the description http://<yourhost>/manual/programs/htcacheclean.html) which you must execute periodically if you wish to use the disk cache.

9.2 How to handle Apache authentication with an SQLite database

In addition to the authentication from a text file (AuthUserFile Directive) and from a Berkeley-DB, Apache2 also offers user authentication against an SQL database. The advantage is the simple creation and maintenance of such an SQLite3-based (see www.sqlite.org) database, e.g. with PHP scripts. An example for configuring access protection based on user name and password is shown below. The SQL database used for authentication can be created and updated using PHP's pdo_sqlite interface.

```
=================================================
#Use the SQLite driver
DBDriver sqlite3

#Connection string: database name and login credentials
DBDParams "/opt/apache24/conf/user.passwd.sqlite"

#Authentication Section
<Directory /opt/apache24/htdocs/dbdprivate>
  AuthName "Protected Area"
  AuthType Basic
  AuthBasicProvider dbd file
  AuthType Digest
  AuthDigestProvider dbd file
  Require valid-user
  #SQL query to verify a user
  # The AuthDBDUserPWQuery specifies an SQL query
  # to look up a password for a specified user. The
  # query must take a single string (typically an SQL
  # varchar) argument (username), and must return
  # a single value (encrypted password).
  # Note: no single quotes are used around %s!
  AuthDBDUserPWQuery \
    "SELECT password FROM basic_authn WHERE username = %s"

  # The AuthDBDUserRealmQuery specifies an SQL query
  # to look up a password for a specified user and
  # realm. The query must take two string (typically
  # SQL varchar) arguments (username and realm) and
  # must return a single value (encrypted password).
  AuthDBDUserRealmQuery "SELECT password FROM digest_authn \
                         WHERE username = %s AND realm = %s"
</Directory>
=================================================
```

As a result of the following SQL code, which can be executed using the sqlite3 command line utility, two users can be entered as example: one for "Basic" authentication, the other for "Digest" authentication:

```
=================================================
# Example code to fill the SQLite database:
BEGIN TRANSACTION;
CREATE TABLE basic_authn (
  username varchar(32) NOT NULL,
  password varchar(32) NOT NULL,
  description varchar(64)
);
INSERT INTO "basic_authn" VALUES('sqlite3','cHUkyKx8z6QP.',
    'Basic Authentication user');

CREATE TABLE digest_authn (
  username varchar(32) NOT NULL,
  password varchar(32) NOT NULL,
  realm    varchar(32),
  description varchar(64)
);
INSERT INTO "digest_authn" VALUES('sqlite3',
  '415388677d084d30e591291feed51e61', 'Protected Area',
  'Digest Authentication user');

COMMIT;
=================================================
```

The encrypted password for basic authentication can be created using the htpasswd utility.

9.3 How to use the forward proxy

Like Apache 1.3, Apache2 offers proxy functionality. In Apache2 it is even more flexible in usage and configuration. Because of the danger resulting from "open proxies" in the Internet, the proxy functionality is however no longer contained in the standard configuration: whoever wants to set up a proxy should not do this by mistake, but taking all the risks into consideration.
The proxy module in Apache2 no longer automatically handles the cache functionality as it did in Apache 1.3, but the functionality is still available in the mod_disk_cache module and must be configured separately.

In order to use the BS2000 Apache as proxy forwarder e.g. for Internet resources, you can create a configuration similar to the following, e.g. as /opt/apache24/conf/conf.d/mod_proxy.conf:

```
=================================================
# Warning
#  Do not enable proxying with ProxyRequests until
#  you have secured your server. Open proxy servers
#  are dangerous both to your network and to
# the Internet at large.
ProxyRequests On
ProxyVia Full

<Proxy *>
  Require host .my.dom.ain

  AuthType Digest
  AuthName HTTP-Proxy
  AuthDigestProvider file
  AuthUserFile "/opt/apache24/user.passwd"
  Require valid-user
  # You can use the htdigest program to create
  #  the password database:
  #   htdigest -c "/opt/apache24/user.passwd" Proxy hugo
</Proxy>

# Use Company Firewall Proxy if not directly connected
# to the internet:
ProxyRemoteMatch .* http://proxy.my.dom.ain:3128

ProxyTimeout 300
AllowCONNECT 443 563

ProxyBlock .double-advertising.net
ProxyDomain .my.dom.ain

# Use the additional service of the mod_cache module
# to set up a caching proxy.
<IfModule cache_module>
  CacheEnable disk http://

  # For garbage collection, start htcacheclean
  # periodically or as daemon
  CacheRoot     /var/opt/APACHE24.httpd/proxy
  CacheDirLevels 3
  CacheDirLength 3

  CacheDefaultExpire    600
  CacheMaxExpire        7200
  ExpiresDefault        "now plus 2 hours"
</IfModule>
=================================================
```

9.4 How to use a reverse proxy

You can use the *Reverse* proxy functionality, for example, to allow access to intranet servers hidden behind the mainframe via the mainframe server’s web address.

```
# =============================================================
# Note
#  It is not necessary to enable proxying with "ProxyRequests on"
#  if you only want to use the ProxyPass directives!
ProxyPass        /phpdoc http://docserv.my.dom-ain/~docs/phpdoc
ProxyPassReverse /phpdoc http://docserv.my.dom-ain/~docs/phpdoc

# If the Host: header should be passed to the backend server:
#ProxyPreserveHost On
# =============================================================
```

10 PHP notes

10.1 PHP directory structure

The main paths of the APACHE:modphp package:

Pfad	Funktion
`$IPATH/php`	PHP base directory
`$IPATH/modules/libphp5.so`	The Apache-PHP module
`/opt/apache24/php/etc/php.ini`	Default-PHP config file (compiled-in path)
`$IPATH/php/bin/php`	"php" command line utility
`$IPATH/conf/conf.d/php5.load`	"LoadModule" parameter file for Apache PHP module
`$IPATH/conf/conf.d/php5.conf`	Apache configuration for PHP
`$IPATH/php/lib/php/extensions/no-debug-non-zts-20131226/*.so`	Optional loadable PHP modules

10.2 The PHP configuration file php.ini

The compiled-in default path for PHP configuration file is "/opt/apache24/php/etc/php.ini". In order to override this value, the php command line utility offers an option "-c <path>" to specify an alternative configuration file and the PHP module permits the specification of a directive "PHPIniDir <path>".
If you wish to use different configuration files for the PHP module and the php command line utility, remember that the command line utility looks for a php.ini in its execution directory first (i.e., $IPATH/php/bin/php.ini) and also offers an option "-c <path>" for specifying an alternative configuration file; the PHP module does not.
A number of optional function groups is available in the form of separately loadable extensions and can be activated or deactivated in the php.ini by adding or removing a semi-colon in the lines
extension = ***<extension name>***.so (at the end of the file).
In the delivery state, the SESAM and ORACLE database modules are deactivated (as they depend on the installation and configuration of the respective databases), all other PHP extensions are activated.

10.3 EBCDIC-/ASCII dependencies in PHP functions

Some of the PHP functions operate on EBCDIC strings, others on binary data, such as image files in GIF, PNG or JPEG format which are stored bit-identical in the same way as on an ASCII machine. The essential differences are summarized below:

- the standard EBCDIC character set in the POSIX universe is always **OSD_EBCDIC_DF04_1**: if a file is transferred in text mode via FTP, TELNET or remote copy (rcp) between an ASCII machine and POSIX, an implicit conversion OSD_EBCDIC_DF04_1 <-> ISO_8859-1 is always applied. This should be considered when transferring files encoded in a different character (it is sometimes better to transfer the file in binary mode and perform an explicit conversion using the GNU iconv utility afterwards, see below).

- Multibyte support and support for Japanese character sets are not supported in POSIX's PHP version.

- iconv() operations always operate on the specified character set. The iconv implementation normally available in POSIX is not used in APACHE as it only supports a few conversions; instead, APACHE and PHP use the GNU libiconv library. It knows the following BS2000 character set names:
  - OSD_EBCDIC_DF03_IRV aka. EDF03 aka. EDF03IRV
    7-bit ASCII equivalent EBCDIC standard character set
  - OSD_EBCDIC_DF04_1 aka. EDF04-1 aka. EDF04
    "Latin1" equivalent EBCDIC; **POSIX standard character set**
  - OSD_EBCDIC_DF04_15 aka. EDF04-15 aka. EDF04F
    "Latin9" equivalent EBCDIC character set
  - EDF03-DE aka. EDF03DE
    National 7-bit ASCII equivalent German
  - EDF03-DA,-EN,-IT,-FR,-SV aka. EDF03DA,EN,IT,FR,SV
    National 7-bit ASCII equivalent Denmark, England, Italy, France, Sweden
  - EBCDIC.DF.04-DRV aka. EDF04DRV
    "Latin1" equivalent alternative: German EDF03DE extension for missing characters from ISO_8859-1 (but not POSIX standard character set!)

- The first three of these character sets are also registered with the IANA (with the first named name).

- In addition to the named character sets the GNU libiconv library knows the normal ISO character set names and most of the numerous additional standardized character sets. The list of the character set names can be displayed by calling the GNU iconv utility supplied with APACHE: "/opt/bin/giconv -l" displays the available character set names. (The same character set names are also used for the configuration of conversions in the Apache web server.)

- If you want to convert in PHP an EBCDIC text file to UTF-8, the source character set must be specified as OSD_EBCDIC_DF04_1 or OSD_EBCDIC_DF04_15 (and not as ISO-8859-1 or ISO-8859-15).

- iconv_mime_decode_headers() and iconv_mime_decode() expect the input MIME headers in ASCII format, any character set name as EBCDIC string and returns an array with EBCDIC strings on success.

- iconv_mime_encode($fieldname,$fieldval[,$prefs_array]) expects the field name as EBCDIC-String, the field value in the character set as specified in the $prefs_array['input-charset'] (or as default ISO_8859-1), and returns a string in the character set specified in the $prefs_array['output-charset'] (or as default ISO_8859-1).

- image*() operations always operate on binary data.

- uuencode() or uudecode() and base64_encode() or base64_decode() convert ASCII binary data to EBCDIC uuencode text and vice versa.

- range() operations on strings are processed as in ASCII: the range ("a","z") results in the array of (only) lower case letters, not all characters with code values between X'81' and X'A9'.

- scanf(): character areas in scanf(), about: scanf("%[a-z]", ...) behave like in ASCII (in this example only lower case letters).

- trim():character areas in trim($var,"a..z") works like on ASCII machines (in this example only lower case letters).

- stripcslashes(): octal and hex constants are regarded as (EBCDIC converted) ASCII literals and processed accordingly (stripcslashes('{\133\x40}') results in => "{[@}" and not "{$ }")

10.4 Time zone specifications

The UNIX SystemV Rel 4 upon which POSIX is based uses time zone specifications different from the time zone database in PHP. For example, in POSIX the following environment variable must be set for the time zone in Germany (see /etc/TIMEZONE):

```
  TZ=MEZ-1MSZ-2,M3.5.0/02:00,M10.5.0/03.00
```

while in PHP it is:

```
  ini_set("date.timezone","Europe/Berlin");
```

(analog in php.ini). Depending on the date function used, it can be necessary to use both methods in order to set the correct time zone.

11 Oracle notes

The PHP Interpreter and the PHP module (both from APACHE:modphp) support the Oracle interface from ORACLE 12c Release 1 (12.1.0.2) for Fujitsu BS2000/OSD. In order to use it, the following configuration changes must be made:

- Assume $*ORAUID* to be the ID under which the ORACLE software was installed in BS2000

- There must be an Oracle installation under POSIX. You can get the directories ORACLE_BASE and ORACLE_HOME (determined by the installation) from the file $*ORAUID*.DEMO.P.ORAENV.

- The file $*ORAUID*.*APAC*.P.ORAENV isn't needed anymore since Oracle 11g.

- In the directory /opt/apache24/lib one must not create a symbolic link to the actual path of the libclntsh.so file below $ORACHE_HOME. A possibly existing link should be removed. The PHP modules oci8 and pdo_oci evaluate during search for the ORACLE library the environment variable LD_LIBRARY_PATH:

```
===============================================
# cd /opt/apache24/lib
# rm  -f  libclntsh.so*
===============================================
```

- With the specification of corresponding environment variables in the environment file $IPATH/sbin/envvars for the Apache webserver one can determine, which Oracle installation the PHP OCI8 module shall use for loading the Oracle modules:

```
 […]
 # This file is generated from envvars-std.in
 #
 if test "x$LD_LIBRARY_PATH" != "x" ; then
   LD_LIBRARY_PATH="/opt/apache24/lib:$LD_LIBRARY_PATH"
 else
   LD_LIBRARY_PATH="/opt/apache24/lib"
 fi
 #
 # Oracle environment
 ORACLE_BASE=<your oracle_base>;
 ORACLE_HOME=<your oracle_home>
 TNS_ADMIN=<your network admin path,
                        e.g. $ORACLE_HOME/network/admin>
 ORACLE_SID='APAC'
 CLN_BASE='200M'
 export ORACLE_BASE ORACLE_HOME
 export TNS_ADMIN ORACLE_SID CLN_BASE
 LD_LIBRARY_PATH="$LD_LIBRARY_PATH:$ORACLE_HOME/lib";
 export LD_LIBRARY_PATH
```

The environment variable TNS_ADMIN specifies the path, under which the Oracle Net parameter files sqlnet.ora and tnsnames.ora are searched.

- The file **tnsnames.ora** contains Oracle Net Alias names, with which a database can be addressed. If Oracle Net Alias names are used in the PHP scripts, these have to be defined in the tnsnames.ora file. This file must be created in the directory specified by TNS_ADMIN. If TNS_ADMIN is not supplied, the file is searched under $ORACLE_HOME/network/admin. It is recommended to use the TCP protocol for the communication with the Oracle database.

- In the PHP configuration file /opt/apache24/php/etc/php.ini the comment character must be removed from the lines which load the respective OCI interface to load the OCI functions on web server start:

  ```
  extension = oci8.so
  extension = pdo.so
  extension = pdo_oci.so
  ```

  Alternatively, the call dl('oci8.so'); or the two calls dl('pdo.so'); and dl('pdo_oci.so'); can be used in php scripts in order to load these functions only as required.

Two different PHP/Oracle interfaces exist in PHP: the direct OCI8 functions ("oci8") and the abstracted functions available on top of the PDO database abstraction layer ("pdo_oci"). For more information about this topic, see the PHP documentation in the package: APACHE:modphp-d or under https://www.php.net/ref.oci8 or https://www.php.net/pdo-oci

12 SESAM notes

The PHP Interpreter and the PHP module (both from APACHE:modphp) support the SESAM interface as of SESAM(BS2000) V8.0. In order to use it, the following configuration changes must be made:

- The specifications for the PHP Sesam module to find the database and the database handler must be entered in the file $IPATH/php/etc/php.ini:

  ```
  [Sesam]
  sesam.msgfile  = "$.SYSMES.SESAM-SQL.yyy"
  sesam.sesamlib = "$.SYSLNK.SESAM-SQL.yyy"
  sesam.blslib = "$.SYSLNK.CRTE"
  sesam.configuration = "$SESAM.CONF.AW"
  sesam.national_charset = "OSD_EBCDIC_DF04_15"
  magic_quotes_sybase = True
  ```

  The correct value in a particular case for the specification of the CRTE library sesam.blslib can be determined with the BS2000 command /SHOW-INST-PATH INSTALLATION-UNIT=CRTE,LOGICAL-IDENTIFIER=SYSLNK where one has to take the value for TARGET=A or TARGET=S.
  The option "magic_quotes_sybase = True" is used for the correct handling of single quotes when processing form data for SQL commands.

- In the PHP configuration file /opt/apache24/php/etc/php.ini, the comment character must be removed from the line which loads the SESAM interface if the SESAM functions are to be loaded on web server start:

  ```
  extension = sesam.so
  ```

  Alternatively, the call dl('sesam.so'); can be used in php scripts in order to load these functions only as required. However, this is not recommended due to the SESAM initialization overhead per call.

- Before first usage of the SESAM interface the module SEPHP contained in S//LIB.APACHE.024.MODPHP has to be copied e.g. with LMS or LMSCONV into the corresponding LMS library S//LNK.SESAM-SQL.yyy.

13 SQLITE notes

Both APACHE and PHP can use SQLITE3 databases. The web server can, for example, load authentication information from either a clear text file like in APACHE V1.3, or from Berkeley DB files, or from SQL files in SQLITE3 format.

The PHP module supports both the SQLITE2 and the SQLITE3 interfaces, but with a different php language interface. While the SQLITE2 functions can be addressed both in a database-specific manner:

```
$db = sqlite_open('/path/to/sqlite2dbfile', 0666, $sqliteerror);
```

as well as via the PDO database abstraction layer, the SQLITE3 functions can only be addressed via the modern PDO database abstraction layer:

```
$dbh = new PDO('sqlite:/path/to/sqlite3dbfile', $user, $pass);
```

More about this topic is found in the PHP documentation in the package:
APACHE:modphp-d or under https://www.php.net/

14 Restrictions

- At the end of the PHP module installation, the automatic activation and restart of the web server does not always succeed.
  If required, please issue the command (in BS2000):
  ```
  /RESTART-APACHE *GRACEFUL
  ```
  or (under POSIX):
  ```
  # apache2ctl graceful
  ```
  manually after completion of the APACHE:modphp installation in order to activate the module for good.

- When using bs2fs file systems with the Apache webserver care is necessary regarding performance aspects, because for the exact estimation of file sizes the files have to be copied from BS2000 to POSIX. This opens possibilities for Denial of Service attacks, i.e. the access should be enabled only for authenticated users if possible.

15 Literature

15.1 Online documentation

The online-documentation is available in the form of independently installable documentation packages. They can be installed or deinstalled as required. The documentation is activated by a web server restart which can be (optionally) executed automatically in the post-install phase of the respective package.

After the installation, the documentation packages can be addressed under the following URLs:

Component: **URL:**

APACHE:httpd-d (documentation of Apache web server directives):
http://***servername***/manual/

APACHE:modprl-d (documentation of the mod_perl module):
http://***servername***/manual/mod/mod_perl/

APACHE:modphp-d (documentation the PHP5 script language):
http://***servername***/manual/mod/mod_php5/

Furthermore, see the documentation pages of the respective OpenSource web sites at:

- https://httpd.apache.org/docs/2.4/,
- https://www.php.net/ und
- https://perl.apache.org/docs/

15.2 Apache Essentials: Install, Configure, Maintain

By Darren James Harkness (Author)
Paperback: 167 pages
Publisher: friends of ED; Corr. 2nd printing edition (May, 2004)
Language: English
ISBN-10: 1590593553
ISBN-13: 978-1590593554

15.3 mod_perl2 User's Guide Book

by Stas Bekmand Jim Brandt.
Published by OnyxNeon in August 2007.
Price: $34.95 (50% of this book's proceeds go to The Perl Foundation.)
Language: English
Date: August 2007
ISBN-10: 0-9779201-1-9
ISBN-13: 978-0-9779201-1-2
Companion Web Site at http://modperl2book.org/

15.4 PHP Cookbook (Paperback)

By David Sklar (Author), Adam Trachtenberg (Author),
Paperback: 796 pages
Publisher: O'Reilly Media, Inc.; 3rd edition (June 2014)
Language: English
ISBN-10: 144936375X
ISBN-13: 978-1449363758

15.5 POSIX (BS2000) - Basics for Users and System Administrators

User Guide

Target group
BS2000 system administrator, POSIX administrator, BS2000 user

Contents

- Introducing and working with POSIX
- BS2000 software products in the POSIX environment
- Install POSIX
- Administer POSIX control and filesystems
- Administer POSIX users
- BS2000 commands for POSIX