

Inhaltsverzeichnis	1
Kernmodul Installation Upgrade	2
Vorbereitung der Installation	2
Das Kernmodul	2
Ausbaustufen einer SuperX-Installation	3
Installation	3
Neuinstallation	3
Übersicht über Installationsschritte	4
Besonderheiten für verschiedene Betriebssysteme	4
Ubuntu / Debian	4
Pakete Ubuntu generell	4
Pakete Ubuntu DB-Server	4
Pakete Ubuntu Appserver	5
Pakete Ubuntu Webservers	5
RedHat / CentOS	5
Kurzanleitung Installation SuperX unter Linux	5
Installation und Pflege der SuperX-Datenbank	6
Einrichten des Datenbanksservers unter UNIX / LINUX	6
Zeichencodierung	6
Konfiguration der Zeichencodierung unter Suse Linux	6
Konfiguration der Zeichencodierung unter Ubuntu Linux	6
Zeichencodierung ändern	7
User superx - Kernmodul entpacken	7
Manuelle Installation von PostgreSQL	7
Neuinstallation Postgres vom Quellcode	7
Postgres-Zusätze installieren: pgcrypto	9
Postgres mit SSL Support	10
Postgres aus der Distribution Ubuntu oder Debian	10
Teil der Installation von Postgres unter Ubuntu/Debian	10
Postgres unter Redhat / Fedora / CentOS	11
Postgres unter Suse Linux	12
Postgres Performance-Tipps	12
Postgres-Nutzerkennungen mit Leserecht	13
Einspielen des Kernmoduls der SuperX-Datenbank	13
Update und Sichern der Datenbank	13
Ein Dump unter Postgres	14
Anpassen der Datenbankparameter db.properties	14
Installation und Pflege des Applikationsservers	15
Installation von Java	15
Einrichtung der Servlet-Engine	15
Steuerung des Servers: Die server.xml	15
Datenbankverbindung für DBFORMS	15
Die Datei conf/web.xml	15
Administrator und Manager	16
Einrichten der SuperX-Servlets unter Tomcat	16
Allgemeines zur web.xml	16
SuperXManager Servlet	17
ResponseCompression	17
filedCache	17
Session-Timeout	17
CSV-Encoding beim Export	17
Fehlerseiten	18
Start des Tomcat	18
Die Übertragung der Web Application	18
LDAP Anbindung	18
Zertifikat einspielen	18
Anpassen der superx_ldap.properties	19
Trotz LDAP lokalen Login aktivieren	19
Shibboleth Anbindung	19
Integration von Tomcat mit dem Apache	20
Umgang mit SSL Verschlüsselung	20
Erzeugen eines SSL Zertifikats	20
Erzeugen eines Zertifikat-Request für eine Zertifizierungsstelle	21
Importieren des Zertifikats in Java	21
Mailversand	22
Test- und Produktivsystem synchronisieren	23
Entladeparameter	23
Ausführung	25
Upgrade einer bestehenden SuperX-Installation	25
Patch einspielen	26
Upgraden des Kernmoduls	26
Standardvorgehen beim Upgrade	26
Upgrade Kernmodul Besonderheiten	27
Migrationsprojekte	27
Postgres: Wechsel auf der Zeichencodierung auf UTF-8	27
Migration von SuperX zu HISinOne / Edustore	28
Migration von Datenbank und Dateisystem	28
Einrichtung der BI Konnektoren	29
Tomcat aktualisieren	30

Kernmodul Installation Upgrade

Vorbereitung der Installation

Die SuperX-Datenbank läuft auf Windows- und Linux-Rechnern. Der SuperX-Client läuft im Browser. Für den Applikationsserver empfehlen wir in jedem Falle einen UNIX bzw LINUX-Server, da alle serverseitigen Scripte als Shellscripte konfiguriert sind. Alle aktuellen Linux-Distributionen enthalten bereits die benötigten Pakete Java, Tomcat und Postgres.



Noch ein Hinweis zur Zeichen-Codierung: Ab dem Kernmodul 5.0 wird UTF-8 empfohlen, nicht ISO. Achten Sie darauf, das jeweils passende SuperX-Paket herunterzuladen (im Dateinamen befindet sich entweder iso oder utf8). Weitere Hinweise siehe Kapitel zur [Zeichencodierung](#).



Für die Installation der Softwarevoraussetzungen liefern wir i.d.R. Beispiel für Ubuntu Linux, die, mit Einschränkungen, auch auf Debian übertragbar sind. Einzelne Kapitel beinhalten aber auch spezielle Hinweise für andere Distributionen (SuSE, AlmaLinux, Redhat)

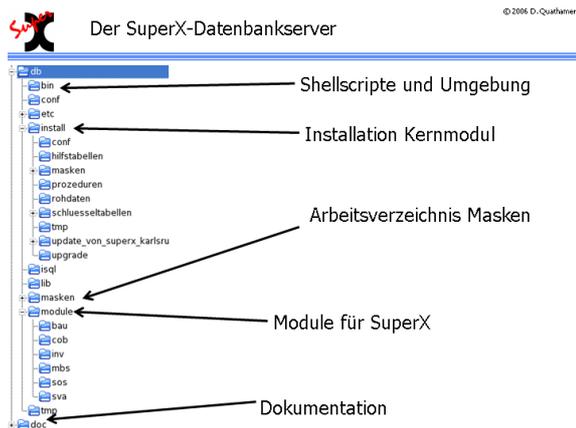
Das Kernmodul

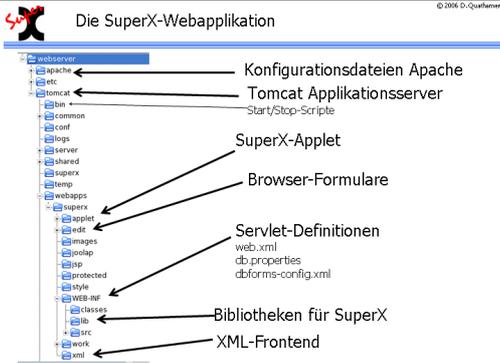
Das SuperX-Kernmodul beinhaltet alle zum Betrieb von SuperX unbedingt notwendigen Tabellen, Prozeduren und Abfragen; die wichtigsten Tabellen werden [unten](#) näher beschrieben.

Die folgende Tabelle zeigt die Ordnerstruktur des Kernmoduls auf einen Blick:

Pfad	Unterordner	Beschreibung
db		Die SuperX-Datenbankseite
	bin	Shellscripte
	etc	Beispiel-Initscripte für SuperX-DB-Dienste
	install	Installationsscripte
	module	Modulpfad
webserver	tomcat	Tomcat-Beispielimplementation (Tomcat 8.5)
	apache	Apache-mod_jk (binär für SuSE Linux + source)
	etc	Beispiel-Initscripte für SuperX-DB-Dienste

Die folgenden Abbildungen zeigen die Ordnerstruktur von jeweils Datenbank-Seite und Webserver-Seite.



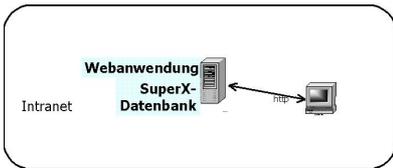


Der Datenbankserver kann auf einem anderen Rechner liegen als der Applikationsserver; es ist aber auch möglich, das gesamte SuperX auf einem Rechner zu installieren. Unter Linux können Sie einen Nutzer `superx` mit dem Verzeichnis `/home/superx` einrichten. Den von Ihnen gewählten Pfad bezeichnen wir als im Folgenden als `$$SUPERX_DIR`, und alle Verzeichnisse des Kernmoduls (`db.doc,webservice`) werden dort hineinkopiert.

Ausbaustufen einer SuperX-Installation

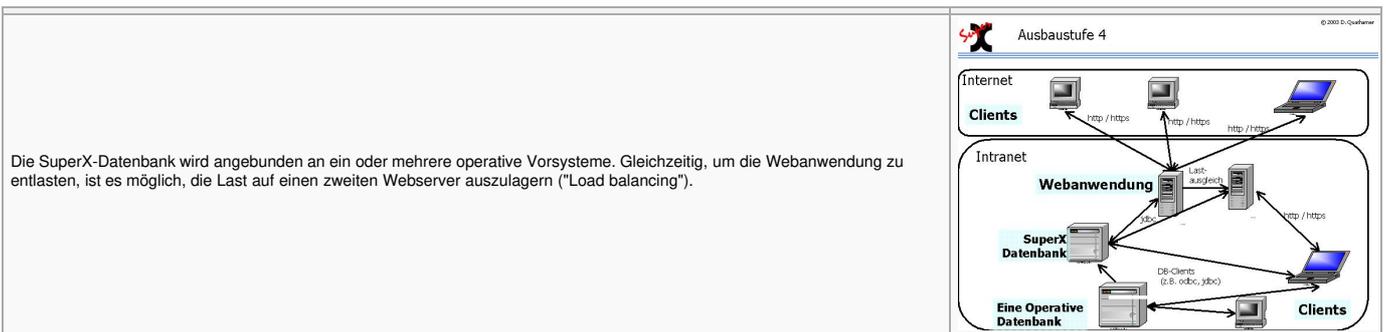
SuperX liefert eine datenbankbasierten Website zur Präsentation von Inhalten der Hochschule für die öffentliche Nutzung im Internet sowie für die interne Nutzung im Intranet. Nach einer Datenübernahme aus den operativen Systemen gilt es, eine effiziente Berichterstellung zu ermöglichen und Export- und Importschnittstellen zu bieten. Das System wird in mehreren Ausbaustufen realisiert, wichtig ist daher die Skalierbarkeit des Systems vom Prototypen bis zum Echtbetrieb.

Der Prototyp der Erstinstallation ist auf einem einzelnen System möglich:



Das zu realisierende System besteht aus drei Komponenten: der Datenbank, der Webanwendung und des Clients (3-tier-Application).

Falls die Last ansteigt, ist das System wie folgt skalierbar:



Die SuperX-Datenbank wird angebunden an ein oder mehrere operative Vorksysteme. Gleichzeitig, um die Webanwendung zu entlasten, ist es möglich, die Last auf einen zweiten Webserver auszulagern ("Load balancing").

Installation

Die Installationsschritte beziehen sich auf die Neuinstallation und das Upgrade. Für die Neuinstallation für Testzwecke gibt es eine [Kurzanleitung](#) unter Linux.

Neuinstallation

Bei der Neuinstallation können Sie einfach alle Komponenten in einen Pfad `$$SUPERX_DIR` kopieren und von dort die unten genannten Installationsschritte durchführen. Beim Update können Sie die Patchdatei in `$$SUPERX_DIR` entpacken; die "alten" Dateien werden ersetzt. Wenn Sie die Datenbank und den Applikationsserver auf getrennten Systemen betreiben, dann entpacken Sie am besten die Update-Datei in einem temporären Verzeichnis und kopieren dann die Ordner `/db` und `/webservice` auf die entsprechenden Rechner.



Wichtig: Ändern Sie bitte keinesfalls die Ordnerstruktur unterhalb von /db und /webserver; Sie können u.U. keine Updates ohne umfangreiche Anpassungen einspielen. Besonders bei der Inbetriebnahme des Systems ist es für die Fehlersuche unerlässlich, die Ordnerstruktur einzuhalten.

SuperX ist zwar ein sehr offenes System, aber gewisse Konventionen werden sich in Zukunft als nützlich erweisen, wenn verschiedene Hochschulen Daten und Skripte austauschen wollen. In jedem Fall empfehlen wir Ihnen immer erst dann manuelle Anpassungen, wenn die Anwendung oder das Script funktioniert – eine äußerst sinnvolle Heuristik für die Arbeit mit SuperX.

Übersicht über Installationsschritte

Das Kernmodul wird in drei Arbeitsschritten installiert:

- Installation und Einrichtung der Datenbank
- Installation eines Webservers mit Servlet-Engine
- Test im Browser

Die folgende Übersicht zeigt das Vorgehen bei der SuperX-Installation, darauf folgt eine Kurzanleitung für die Installationsmaßnahmen:

Schritt	Erläuterung
Kopieren und Vorbereiten des Kernmoduls nach \$SUPERX_DIR	Bringen Sie das SuperX-Kernmodul in ein Verzeichnis auf dem Rechner, am besten auf den Applikationsserver. Kopieren Sie die Datei \$SUPERX_DIR/db/bin/SQL_ENV.sam nach \$SUPERX_DIR/db/bin/SQL_ENV und passen Sie die Umgebungsvariablen an.
Installation der erforderlichen Software	<ul style="list-style-type: none"> • Installieren Sie auf dem Datenbankserver PostgreSQL • Installieren Sie auf dem Webserver Java die Datenbanktreiber, und setzen Sie die Umgebungsvariable JAVA_HOME für Tomcat in der Datei \$SUPERX_DIR/db/bin/SQL_ENV
Start des Datenbankservers	Starten Sie den Datenbankserver und spielen Sie danach die SuperX-Datenbank des Kernmoduls ein; die Zugangsparameter müssen Sie dem SuperX-Servlet bekannt geben.
Start des SuperX-Servlets	Gehen Sie in das Verzeichnis /webserver/tomcat/bin und starten Sie Tomcat, ggf. als Dienst
Test des Applikationsservers	Öffnen Sie die Datei http://-Rechnername-:8080/superx/ im Browser und testen Sie zuerst die Anmeldung
Freigabe des Webservers im Netz	Nun ist die Basisinstallation vom Kernmodul abgeschlossen, und Sie können mit der Anpassung für Ihre Einrichtung beginnen. Richten Sie User ein, und geben Sie die WWW-Adresse Ihres Webservers im Intra- oder Internet (in diesem Fall benötigen Sie auch die die Verschlüsselung) frei.
Einspielen der Module	Füllen Sie SuperX mit den einzelnen Modulen; bisherige SuperX-Anwender können ihr bisheriges System übernehmen.

Besonderheiten für verschiedene Betriebssysteme

Wir empfehlen den Einsatz von SuperX unter Linux. Für andere Betriebssysteme gelten hier und da Besonderheiten.

Ubuntu / Debian

Wenn Sie Ubuntu nutzen, können Sie auch den Tomcat und Postgres von Ubuntu nutzen. Dabei ist aber auf einiges zu achten. Bitte schauen Sie dafür bitte unter den Kapiteln für Übertragung der Webapplikation auf einen vorhandenen Tomcat unter Ubuntu und Postgres unter Ubuntu/Debian nach.



Bisher konnten wir auch noch keine Probleme mit OpenJDK feststellen, Sie benötigen also nicht unbedingt Oracle Java.

Pakete Ubuntu generell

Hier zum Kopieren eine Liste von Paketen für Ubuntu 22.04 Server, die immer (d.h. für DB- und Applikationsserver) benötigt werden:

```
apt-get update
apt-get upgrade -y
apt-get install -y openssh-server
apt-get install -y default-jre
apt-get install -y git
apt-get install -y recode
apt-get install -y zip
apt-get install -y openjdk-11-jdk
```

Noch ein Tipp:

Ubuntu deaktiviert standardmäßig die Suche in der Bash-History mit der PAGE-UP/PAGE-DOWN Taste. Sie können das systemweit einschalten in der Datei /etc/inputrc, indem Sie die Kommentarzeichen entfernen:

```
# alternate mappings for
"page up" and "page
```

Wenn Sie sich dann erneut einloggen sind Sie produktiver!

Pakete Ubuntu DB-Server

Aufbauend auf den [allgemeinen Ubuntu-Paketen](#) hier eine Liste von Paketen bzw. Konfigurationen für den DB-Server:

```
apt-get install -y postgresql
apt-get install -y postgresql-contrib
apt-get install -y gettext
apt-get install -y gettext libreadline8
```

```
apt-get install -y gettext libreadline-dev
apt-get install -y gettext zlib1g
apt-get install -y gettext zlib1g-dev
apt-get install -y gettext libxml2
apt-get install -y gettext libxml2-dev
apt-get install -y gettext libxslt1.1
apt-get install -y gettext libxslt1-dev
locale-gen de_DE.UTF-8
update-locale LANG=de_DE.UTF-8
```

Pakete Ubuntu Appserver

Aufbauend auf den [allgemeinen Ubuntu-Paketen](#) hier eine Liste von Paketen für den Applikationsserver:

```
apt-get install -y tomcat9
apt-get install -y sendmail
apt-get install -y s-nail
apt-get install -y curl
apt-get install -y mail
apt-get install -y dos2unix
apt-get install -y postgresql-client
```

Pakete Ubuntu Webserver

Wenn Sie einen separaten Webserver mit Apache betreiben wollen hier die Pakete:

```
apt-get install -y apache2
apt-get install -y libapache2-mod-jk
apt-get install -y socat
```

RedHat / CentOS

Bei RedHat bzw. CentOS muss man einige Pakete anders installieren. Bei Red Hat Enterprise Linux Server release 5.11 z.B. muss man das Paket recode von anderer Stelle installieren. Das Apache-Paket lautet "httpd".

Kurzanleitung Installation SuperX unter Linux

Die folgende Kurzanleitung gilt für den häufigsten Fall des Betriebs von SuperX: Unter Ubuntu/Debian Linux mit Postgres als DBMS im Verzeichnis /home/superx (also für die Kennung superx). Wenn Sie abweichende Varianten wählen folgen Sie den jew. Links, dort gibt es Alternativumgebungen.

Voraussetzungen	Postgres ist gestartet und läuft, der User existiert im Datenbanksystem hat das Recht, Datenbanken zu erzeugen, OpenJDK 8 oder 11 ist installiert. Auf dem Server sollte kein weiterer Server-Dienst auf den Ports 8005, 8009 und 8080 (Tomcat-Standard-Ports) laufen.
Entpacken	Entpacken Sie das Kernmodul in /home/superx tar -xzf kern<
Umgebungsvariablen	Gehen Sie in das Verzeichnis db/bin cd db/bin Kopieren Sie SQL_ENV.sam nach SQL_ENV cp SQL_ENV.sam SQL_ENV Wenn Sie die Datenbank superx unter Linux in /home/superx mit Postgres als DB-Server und Java im Verzeichnis /usr/lib/java installiert haben, brauchen Sie nichts ändern. Ansonsten passen Sie \$SUPERX_DIR, \$JAVA_HOME, \$DATABASE, \$SX_CLIENT, \$LANG und \$MAILPROG an Starten Sie das Script mit . SQL_ENV und fügen Sie den Aufruf am Ende der Datei ~/.bashrc ein: . ~/db/bin/SQL_ENV

Einspielen der Datenbank	Gehen Sie in das Install-Verzeichnis cd \$SUPERX_DIR/db/install Starten Sie das Script kernmodul_erzeugen.x Bei Fehlern kommt eine Meldung, Protokolle stehen in create.log Damit ist die db-Seite eingerichtet.
Steuerungsdatei für das Servlet: db.properties	Nun gehen Sie in das Verzeichnis cd \$SUPERX_DIR/webserver/tomcat/webapps/superx/WEB-INF Kopieren Sie db-postgres.properties bzw. db-informix.properties nach db.properties. Beispiel für Postgres: cp db-postgres.properties db.properties Staten Sie den PropAdmin mit propadmin.x Passen Sie hier den Servernamen, Datenbanknamen, Usernamen und Passwort an, und drücken Sie "Verbindung testen". Danach speichern Sie die Datei.

Start von Tomcat	Nun wechseln Sie in das Verzeichnis cd \$SUPERX_DIR/webserver/tomcat/bin Und starten tomcat mit startup.sh (Stop übrigens mit "shutdown.sh".)
Und los geht's...	Nach erfolgreichem Start rufen Sie im Browser auf: http://localhost:8080/superx/ Dort können Sie die Anwendung aufrufen. Geben Sie bei der Kennung superx ein, und als Passwort "anfang12"
Feineinstellung	Melden Sie sich in der Anwendung an und gehen Sie im Menü auf "Tabelle suchen", klicken einfach "Abschicken" und öffnen Sie die Tabelle hochschulinfo ; dort sollten Sie Ihre Hochschule auswählen und "Speichern" anklicken, damit Ihre Hochschulnummer gespeichert wird.

Installation und Pflege der SuperX-Datenbank

Zunächst muss aber der Datenbankserver eingerichtet werden. Derzeit laufen die Installationsskripte und auch alle Modulskripte nur unter UNIX /Linux.

Einrichten des Datenbankservers unter UNIX / LINUX

Der Datenbankserver läuft unter Informix (mind. Version 12.10) und PostgreSQL (mind. Version 11). Der Einsatz von Informix ist allerdings "deprecated", wir empfehlen den Umstieg auf Postgres.

Zeichencodierung

Bevor Sie das Kernmodul entpacken, sollten Sie sich vergewissern dass die Zeichencodierung des jew. Pakets mit der installierten übereinstimmt.

Die auf Ihrem System installierte Codierung erfahren Sie mit dem Befehl

```
echo $LANG
```

Mögliche Ausgaben sind de_DE@euro (oder die jew. Variante mit ISO) und de_DE.utf8 (je nach UNIX gibt es hier unterschiedliche Schreibweisen, z.B. auch de_DE.UTF-8). Wenn Sie sich nicht sicher sind, welche Codierung überhaupt installiert ist, können Sie mit

```
locale -a | grep de
```

eine Liste der deutschsprachigen Locales anzeigen.

SuperX unterstützte bis Version 3.5 nur die Locale de_DE@euro. Ab Version 4.0 ist auch UTF-8 möglich, ab 4.5 ist UTF-8 der Standard.

Mit dem SuperX-Kernmodul werden Skripte ausgeliefert, mit denen die Codierung von [Dateien](#)] flexibel geändert werden kann.

Wenn die Fehlermeldung:"psql: FATAL: conversion between LATIN9 and LATIN1 is not supported" auftritt, unterscheidet sich die in der Shell eingetragene locale mit der in der Datenbank eingetragenen locale. Das Problem lässt sich lösen mit setzen der Variable LANG auf "de_DE.8859-1". Wenn die Locale nicht verfügbar ist, muss man sie nachinstallieren (s.u.).

Konfiguration der Zeichencodierung unter Suse Linux

Da SuSE Linux "deutsche" Wurzeln hat, die die benötigten Locales de_DE@euro und de_DE.utf8 in der Standardinstallation bereits installiert. Sie können die Zeichencodierung in der SQL_ENV eintragen und in der \$HOME/.bashrc laden.

Hinweis für OpenSuse und Postgres: Die psql- und Java-Shell in OpenSuse 11.4 wertet nicht nur die Variable LANG aus, sondern auch "LC_ALL". Die Ursache dafür haben wir noch nicht gefunden. Im Zweifelsfall setzen Sie LC_ALL auf den gleichen Werte wie LANG.

Konfiguration der Zeichencodierung unter Ubuntu Linux

Mit locale -a | grep de sehen Sie alle deutschen installierten Locales. Wenn die ISO-Codierung fehlt, müssen Sie sie wie folgt nachinstallieren:

```
apt-get install language-pack-de-base language-pack-de locales
```

Danach prüfen Sie in der Datei /usr/share/i18n/SUPPORTED , ob die Locales auswählbar sind:

```
vi /usr/share/i18n/SUPPORTED
```

Hier sind nun alle möglichen Sprachen sichtbar. Wir benötigen die locale de_DE@euro ISO-8859-15

Dann muss man sie verfügbar machen:

```
vi /var/lib/locales/supported.d/de
```

hat z.B. den Inhalt:

```
de_DE.UTF-8 UTF-8
```

```
de_CH.UTF-8 UTF-8
de_BE.UTF-8 UTF-8
de_LI.UTF-8 UTF-8
de_LU.UTF-8 UTF-8
de_AT.UTF-8 UTF-8
de_DE@euro ISO-8859-15
```

Danach gibt man ein:

```
dpkg-reconfigure locales
```

Wenn Sie dann noch einmal `locale -a | grep de` eingeben, sollte die Locale `de_DE@euro` sichtbar sein.

Tipp: wenn Sie unter Debian/Ubuntu eine root-Shell benötigen, müssen Sie eingeben: `sudo -i`

Die Datei `.bashrc` wird unter Ubuntu Linux nicht beim Öffnen einer Login Session durchlaufen, Sie können diese aber in der `$HOME/.profile` laden:

```
...
if -n "$BASH_VERSION"; then
# .bashrc laden, wenn vorhanden:
if -f "$HOME/.bashrc"; then
. "$HOME/.bashrc"
fi
fi
...
```

Zeichencodierung ändern

Für die tägliche Arbeit ist es nützlich, das Unix-Programm `recode` zu installieren, dies wird von den [Konvertierungsskripten](#) genutzt. Bei OpenSuse ist das standardmäßig installiert, bei Ubuntu muss man es nachinstallieren. Bei Red Hat Enterprise Linux Server 5.* muss man zunächst die Paketquelle angeben:

Datei `/etc/yum.repos.d/rpmforge.repo` anlegen mit dem Inhalt:

```
# Name: RPMforge RPM Repository for Red Hat Enterprise 5 - dag
# URL: http://rpmforge.net/

[rpmforge]

name = Red Hat Enterprise $releasever - RPMforge.net - dag

#baseurl = http://apt.sw.be/redhat/el5/en/$basearch/dag
mirrorlist = http://apt.sw.be/redhat/el5/en/mirrors-rpmforge
#mirrorlist = file:///etc/yum.repos.d/mirrors-rpmforge
enabled = 1
protect = 0
#gpgkey = file:///etc/pki/rpm-gpg/RPM-GPG-KEY-rpmforge-dag
gpgcheck = 1
```

Danach gibt man ein

```
yum --nogpgcheck install recode
```

Danach ist `recode` verfügbar.

User superx - Kernmodul entpacken

Legen Sie einen User `superx` am einfachsten mit dem `home`-Verzeichnis `/home/superx` an.

Wenn wir im Folgenden `$SUPERX_DIR` sprechen, meinen wir `/home/superx`. Es ist natürlich auch jedes andere Verzeichnis möglich.

Es muss auf Betriebssystemebene sichergestellt werden, dass das Dateisystem Textdateien in der passenden Locale anlegt sind. Bei modernen UNIXen ist die Umgebungsvariable `$LANG` standardmäßig auf "UTF-8" gesetzt.

Setzen Sie die richtige Locale z.B. mit

```
LANG=de_DE.utf8; export LANG
```

Entpacken Sie die Datei `kernXX.tar.gz` im Verzeichnis `$SUPERX_DIR`.

Machen Sie eine Kopie der Datei `Date $SUPERX_DIR/db/bin/SQL_ENV.sam` und nennen Sie sie einfach `SQL_ENV`. In dieser Datei werden viele allgemeine [Konfigurationen](#) der Umgebung vorgenommen. Prüfen Sie, ob die in der `SQL_ENV` angegebene Locale (`LANG=de_DE...`) existiert.

Geben Sie der Datei ggf. Ausführungsrechte mit `chmod +x SQL_ENV`.

Manuelle Installation von PostgreSQL

SuperX ist seit Version 2.1 mit Postgres lauffähig. Die Distribution von Postgres für Unix findet sich unter www.postgresql.org.

Verschiedene Linux-Distributionen enthalten zwar bereits Postgres und müssen nicht "von Hand" installiert werden. Dies hat den Vorteil dass die Installation leicht ist und Sicherheitsupdates automatisch eingespielt werden können. Aber Vorsicht: die Distribution legt Postgres in anderen Verzeichnissen ab, als das Standardscript von Postgres.

Im folgenden wird die Installation vom Quellcode beschrieben.

Neuinstallation Postgres vom Quellcode

Voraussetzungen

Postgres läuft unter verschiedenen UNIX-Varianten, z.B. Linux, HP-UX oder MacOS X. Wir empfehlen Linux. Vor der Installation unter Linux sollte die Locale-Umgebungsvariable \$LANG auf den gewünschten Wert geändert werden (de_DE.utf8 oder de_DE@euro oder eine andere deutsche Locale (meist in /usr/lib/locale). Die aktuelle Locale wird bei der Installation von Postgres berücksichtigt und sorgt dafür, dass Datums- und Währungsformate korrekt sind.

Bei SuSE Linux ist es für ein Kompilieren der Postgres-Quellen erforderlich, dass die Pakete gcc, glibc, gettext, gettext-devel, readline, readline-devel, zlib und zlib-devel installiert sind.

Bei Ubuntu / Debian:

```
apt-get install -y gettext libreadline8 libreadline-dev zlib1g zlib1g-dev
```

Erzeugen Sie zunächst den User postgres mit dem Homeverzeichnis der Postgres-Installation, z.B. unter Linux mit

```
useradd -g users -d /usr/local/pgsql postgres
```

In der Download-Version von Postgres wird Postgres standardmäßig nach /usr/local/pgsql installiert. Als DBSpace muss man ein oder mehrere Verzeichnisse anlegen und mit initdb vorbereiten. Die SuperX-Datenbank lässt sich dann in einem eigenen DBSpace ablegen.

Zunächst müssen Sie sich als root anmelden. Wir gehen im folgenden davon aus, dass die Quellen von Postgres im Verzeichnis

```
/usr/src/packages/SOURCES
```

liegen (das Archiv z.B. von postgresql-7.3.4.tar.gz muss hier entpackt werden).

Dann gehen Sie in das Verzeichnis postgresql-7.3.4, und führen folgende Befehle aus:

Postgres-Installation "in short":

```
./configure --enable-nls
make
make install
mkdir /usr/local/pgsql/data
chown postgres /usr/local/pgsql/data
```

Wenn Sie SSL Support benötigen, müssen Sie noch den Parameter

```
--with-openssl
```

hinzufügen. Wenn Sie Postgres in einem anderen Verzeichnis als /usr/local/pgsql installieren wollen, müssen Sie den Parameter

```
--prefix=-Pfadname-
```

hinzufügen. Weitere Optionen fürs configure gibt die Zeile

```
./configure --help
```

Damit sind die Schritte, die als root auszuführen sind, beendet. Wir wechseln nun zur Kennung postgres mit

```
su - postgres
```

Vor der Initialisierung des DBSPACE sollte die Sprachumgebung des Users postgres korrekt sein. Für die bash wird in den meisten Distributionen die Umgebung generell in der Datei .bashrc bzw. .profile im Homeverzeichnis des Users postgres gesetzt; dort geben Sie den Pfad für das data-Verzeichnis an, und legen die Ausführprogramme von Postgres in den Datenpfad. Hier ein Beispiel für den Betrieb mit UTF-8:

```
export LANG=de_DE.utf8
#Zur Sicherheit für Postgres auch einzeln:
export LC_CTYPE=de_DE.utf8
export LC_COLLATE=de_DE.utf8
export LC_TIME=de_DE.utf8
export LC_NUMERIC=de_DE.utf8
export LC_MONETARY=de_DE.utf8
export LC_MESSAGES=de_DE.utf8
PATH=$PATH:/usr/local/pgsql/bin
export PGDATA=/usr/local/pgsql/data
export PGLIB=/usr/local/pgsql/lib
...
```

Wenn die Sprachumgebung stimmt, dann wird der DBSPACE vom User postgres initialisiert.

```
/usr/local/pgsql/bin/initdb -D $PGDATA
```

Durch initdb wird der DBSpace erzeugt. Wenn die Umgebung stimmt, dann wird Postgres für die deutsche Locale vorbereitet (Sortierung von Zeichen, Datums- und Währungsformate etc).

Ausgabe von initdb:

```
/usr/local/pgsql/bin/initdb -D $PGDATA
The files belonging to this database system will be owned by user "postgres".
This user must also own the server process.
The database cluster will be initialized with locale de_DE.utf8.
This locale setting will prevent the use of indexes for pattern matching
operations. If that is a concern, rerun initdb with the collation order
set to "C". For more information see the Administrator's Guide.
```

Dann müssen Sie die ip-Nummer des Rechners mit dem SuperX-Webserver (sowie von allen anderen Clients, die direkt auf die Datenbank zugreifen sollen) in die Datei /usr/local/pgsql/data/pg_hba.conf eintragen. In der Datei \$PGDATA/pg_hba.conf stehen die Verbindungsberechtigungen für der Server; hier müssen Sie mindestens dem User superx die Verbindungsrechte geben, z.B. mit folgender Zeile:

Auszug aus pg_hba.conf

```
host all all 127.0.0.1/32 trust
host all all 192.168.0.16/32 trust
```

Die obige Zeile gibt dem User superx Verbindungsrechte für alle Datenbanken auf dem lokalen Rechner 192.168.0.16.

Die Netzmaske "/32" schränkt die Regel einen Rechner ein (entspricht 255.255.255.255). Wenn Sie "/24" wählen, öffnen Sie die Netzmaske auf 255.255.255.0, d.h. bei obigem Beispiel alle Rechner im Netz 192.168.0.x.

Bitte beachten Sie, dass die Standardvorgabe nach der Installation von Postgres die ist, dass alle User auf dem aktuellen Rechner mit dem Datenbankserver verbinden dürfen. Dies sollten Sie natürlich ändern.

Wenn Sie statt "trust" den Wert "md5" eingeben, dann erfolgt eine Passwortabfrage. Dies ist für nächtliche Ladejobs nicht praktikabel. In diesem Falle müssen Sie das Passwort per Client übergeben, entweder mit einer Datei "~/.pgpass" mit dem Inhalt:

```
-Servername-:Port-:Datenbank-:Kennung-:Passwort-
```

z.B.

```
dbserver.hochschule.de:5432:superx:superx:anfang12
```

Alternativ kann man auch die Umgebungsvariable PGPASSWORD mit dem Passwort belegen, dies ist allerdings "deprecated" und wird in zukünftigen Versionen von Postgres unterbunden.

Weitere Parameter werden in der Konfigurationsdatei postgresql.conf definiert; wichtig ist die Einstellung, dass Postgres einen TCP-IP-Socket öffnet (Parameter listen_addresses=IP-Nr-) sowie der TCP-IP-Port (port = 5432 ist die Standardvorgabe). Die Anzahl der gleichzeitig offenen Verbindungen muss kleiner sein als die Anzahl, die Sie für das SuperX-Servlet definieren. Weitere Details zur Einrichtung von Postgres-Runtime-Parametern finden Sie im Admin-Handbuch der Postgres-Distribution.

Danach wird der Datenbankserver gestartet mit dem Befehl postmaster.

```
/usr/local/pgsql/bin/postmaster -i -D /usr/local/pgsql/data
```

Wir empfehlen, die Ausgabe von dem Prozeß in eine Logdatei zu schreiben, z.B. nach /var/log/postgresql.log. Legen Sie diese Datei als User root an, und machen Sie dann den User postgres zum Eigentümer. Ein Beispielscript ist folgendes (im Kernmodul zu finden unter \$SUPERX_DIR/db/install):

pgsql_start.x -Ein Beispielscript zum Start von Postgres

```
#!/bin/sh
PG_HOME=/usr/local/pgsql
export PG_HOME
PGDATA=$PG_HOME/data
export PGDATA
PGPORT=5432
export PGPORT
$PG_HOME/bin/pg_ctl -D $PGDATA -l /var/log/postgresql.log -o -i start
```

Um zu testen, ob die Locale richtig ist, gehen Sie als User postgres in die Shell:

```
psql template1
```

die Datenbank; dann geben Sie ein:

```
select 'aaa' union select 'bbb' union select 'äää' order by 1;
```

Bei richtiger Locale lautet die Ausgabe:

```
?column?
-----
aaa
äää
bbb
```

(3 rows)

Im Verzeichnis \$SUPERX_DIR/db/install befindet sich ein Shellsript check_sortierung_pg.x, das prüft, ob die aktuell in der Umgebung festgelegten Variablen zu korrekter Darstellung von Umlauten und Sortierung unter Postgres der gewünschte Ergebnis bringen. Das Script legt einen temporären DBSPACE an, führt darin einen Testselect aus und löscht den DBSPACE wieder, in der Logdatei check_sortierung.log steht dann das Ergebnis. In dem Script muss die Variable PG_HOME korrekt gesetzt sein, der Rest wird automatisch geprüft.

Dann erzeugen Sie den User superx für Postgres:

```
createuser superx
```

Dieser User muss Datenbanken erzeugen dürfen, braucht aber, wenn Sie als SuperUser bereits die Prozedursprache plpgsql in template1 installiert haben, kein Super-User sein bzw. bei Postgres 7.4 das Recht haben, andere User erzeugen zu dürfen. Aus Sicherheitsgründen empfehlen wir, den User superx, der standardmäßig auch der User ist, mit der die Webapplikation auf die Datenbank zugreift, nicht zum Super-User zu machen.

Wenn der User ein SuperUser sein soll, geben Sie ein:

```
createuser --superuser superx
```

ggfs.

```
ALTER USER superx WITH PASSWORD 'new_password';
```

Bei Änderungen der pg_hba.conf müssen Sie übrigens Postgres nicht neu starten, Sie können die Datei im laufenden Betrieb auch mit pg_ctl -D \$PGDATA reload neu laden.

Damit ist Postgres installiert und für die SuperX-Installation konfiguriert. Bei dieser Gelegenheit sollten Sie den Datenbankserver gleich als Dienst beim Systemstart einrichten.

Postgres-Zusätze installieren: pgcrypto

Neben dem Kernsystem von Postgres bietet es sich an, die vielen Zusatzmodule von Postgres zu nutzen. Die Installation erfolgt aus den Quellen der Kerndistribution. Wir zeigen dies am Beispiel von **pgcrypto**, einem Paket zur Verschlüsselung, das wir für die Verschlüsselung von Passwörtern gebrauchen:

In Postgres9 oder höher ist crypto defaultmäßig bereits mitinstalliert.

Nach dem ./configure (s.o.) der gesamten Postgres-Quellen gehen Sie als root in das Verzeichnis contrib/pgcrypto

Geben Sie ein:

```
gmake all
gmake install
```

Es werden Bibliotheken in /usr/local/pgsql/lib erzeugt. Das SQL-Skript zur Erzeugung der Crypto-Funktionen liegt in /usr/local/pgsql/share/contrib/pgcrypto.sql. Wenn Sie es in der SuperX-Datenbank installieren wollen, geben Sie dort ein:

```
psql superx < pgcrypto.sql
```

Wenn Sie es allen Datenbanken zur Verfügung stellen wollen, laden Sie die Funktionen nach template1:

```
psql template1 < pgcrypto.sql
```

Postgres mit SSL Support

Wenn die Postgres Binaries mit SSL Support erzeugt wurden, kann man den SSL Support leicht aktivieren:

- Erzeugen Sie ein öffentliches und ein privates Zertifikat
- in der Datei postgresql.conf den Schalter ssl = on setzen
- Das öffentliche Server Zertifikat nach \$PGDATA/server.crt kopieren
- Das private Zertifikat nach \$PGDATA/server.key kopieren.

Wenn Sie die Zertifikate wie unten in der Anleitung erzeugt haben lauten die Befehle z.B.:

```
cp /root/demoCA/cacert.pem /usr/local/pgsql/data/server.crt
cp /root/demoCA/private/cakey.pem /usr/local/pgsql/data/server.key
```

Achten Sie beim Kopieren darauf, dass die Dateirechte nur dem Eigentümer Leserecht geben.

Beim Serverstart wird ggf. ein PEM Passwort abgefragt.

Um den Zugriff zum Server per SSL zu steuern, können Sie in der Datei pg_hba.conf statt der Direktive "host" den Namen "hostssl" nutzen. Damit werden SSL Verbindungen erlaubt. Umgekehrt werden keine non-SSL Verbindungen erlaubt, wenn es keine "host" Direktive gibt.

In psql können Sie den Zugang testen, allerdings müssen Sie die Umgebungsvariable

```
PGSSLMODE=require
```

setzen und in der SQL_ENV speichern. Im Erfolgsfall bekommen Sie die Meldung:

```
psql (XXX)
SSL-Verbindung (Verschlüsselungsmethode: DHE-RSA-AES256-SHA, Bits: 256)
```

```
Geben Sie »help« für Hilfe ein.
superx=#
```

Damit die JDBC Klassen und die Shellscripte mit SSL verbinden, muss man das Zertifikat in der Java Runtime [bekannt](#) machen in der Datei webapps/superx/WEB-INF/db.properties den Parameter:

```
ssl=true
```

hinzufügen. Mit DOQUERY "select * from xdummy" kann man den Zugriff testen.

Postgres aus der Distribution Ubuntu oder Debian

Postgres sollte über die normale Paketverwaltung der Distribution installiert werden. Weiter [oben](#) wurde beschrieben wie Sie Postgres unter Debian/Ubuntu/Suse selbst kompilieren.

Vorbemerkung: Wenn Sie den DB-Server auf einem anderen Rechner betreiben, reicht es die Postgres-Clientpakete zu installieren:

```
apt-get install postgresql-client-common
apt-get install postgresql-client
```

Danach sind Kommandos wie **psql** verfügbar.

Die folgenden Anleitungen gelten nur für den Dienst als Server:

```
apt-get update
apt-get install postgresql postgresql-contrib

(wird für postgresql start/stop benötigt)
```

Unter Ubuntu finden Sie die Postgresinstallation z.B. unter /etc/postgresql/9.1/main. Achten Sie bitte darauf, dass Sie in der SQL_ENV den Pfad für \$PGPATH und \$PGDATA in der SQL_ENV Ihrer Postgresinstallation dementsprechend anpassen. \$PGDATA ist in meiner Beispielkonfiguration unter Ubuntu gleich dem Verzeichnis von \$PGPATH.

Wenn Sie auch den Tomcat von Ubuntu nutzen wollen, empfehlen wir Ihnen den Tomcatuser auch in der Datenbank für die SuperX DB Verbindung zu nutzen.

Postgres starten und stoppen Sie mit dem Befehl:

Starten:

```
/etc/init.d/postgresql start
```

Stoppen:

```
/etc/init.d/postgresql stop
```


Wenn Postgres automatisch mit dem Systemstart mit gestartet werden soll, machen Sie das mit dem folgenden Befehl:

```
chkconfig postgresql on
```

Start/Stopp Befehle für Postgres:

```
service postgresql
```

für kann folgendes verwendet werden:

- start : startet die Datenbank
- stop : stoppt die Datenbank
- restart : neustart der Datenbank – Nach Änderungen an der Konfiguration
- reload : reload pg_hba.conf Datei. Dabei läuft die Datenbank weiter.

Postgres unter Suse Linux

Sie installieren Postgres aus der Distribution mit

```
zypper -n in -l postgresql
zypper -n in -l postgresql-server
zypper -n in -l postgresql-contrib
```

Vor dem ersten Start müssen Sie die Sprachumgebung einrichten. Prüfen Sie die installierten Locales mit

```
locale -a | grep de
```

Die gewünschte Locale, in der Regel de_DE.utf8, tragen Sie dann in der Datei /etc/sysconfig/postgresql ein, bei der Variable

```
POSTGRES_LANG="de_DE.utf8"
```

Danach starten Sie den Server als root mit

```
rcpostgresql start
```

Der DBSPACE wird in /var/lib/pgsql/data angelegt, dort liegen dann die Dateien postgresql.conf und pg_hba.conf.

Damit der Server beim Booten hochfährt, geben Sie ein:

```
chkconfig --set postgresql on
```

Dann wechseln Sie in die Shell des Users postgres, und legen den User an mit

```
createuser --superuser superx
```

Postgres-Performance-Tipps

Der Optimierer unter Postgres läßt sich über die Kommandozeile mit

```
vacuumdb --analyze --verbose -f -d $DBNAME
```

starten und hilft bei regelmäßiger Anwendung, deshalb empfehlen wir, diesen Befehl als Cronjob jede Nacht oder einmal pro Woche auszuführen.

Wichtige Parameter sind die "Shared buffers" und der "Effective cache size". **Shared buffers** meinen nicht das gesamte zur Verfügung stehende RAM (das verwaltet das System), sondern den Arbeitsspeicher, der Postgres zum Zwischenspeichern benutzt, bevor Abfragen an den Kernel geschickt werden. Der Wert sollte nicht zu hoch gewählt werden, weil dann die Performance nachläßt. Faustregel: 6-15% des physischen RAM, man sollte aber auch in der Praxis viel probieren. Generell sollte man auf Datenbankservern mind. die Hälfte des verfügbaren physischen Rams für Postgres reservieren.

Beim Start des Postmasters lassen sich die "Shared buffers" zuweisen mit der Option

```
postmaster -o "-B 128"
```

Dabei wird das Shared Memory von (standardmäßig) 64*8192 Bytes auf 128*8192 Bytes erhöht. Man kann den Parameter aber auch in der postgresql.conf setzen.

Beispielkonfiguration Postgres-RAM bei DB-Server mit 1 GB RAM unter Suse Linux, in der Datei /etc/init.d/boot.local geben Sie ein:

```
echo 10737418240 > /proc/sys/kernel/shmmax #1024 MB RAM für PG
echo 2097152 > /proc/sys/kernel/shmall
echo 2 > /proc/sys/vm/overcommit_memory
```

Die Parameter lassen sich auch zur Laufzeit aus einer root-Shell setzen. Danach ersetzen Sie in der postgresql.conf die folgenden Parameter:

```
max_connections = 500
shared_buffers = 16384
max_fsm_pages = 50000
checkpoint_segments = 12
effective_cache_size = 32000
```

Danach starten Sie Postgres neu.

```
}}
```

Die checkpoint segments sollen Sie erhöhen, wenn Sie in den Postgres-Logs folgende Meldung bekommen:

LOG: Checkpoints passieren zu oft (alle xx Sekunden)

TIPP: Erhöhen Sie eventuell den Konfigurationsparameter "checkpoint_segments".

In der Postgres-Auslieferung sind checkpoint_segments=3 vorgegeben, bei großen Anwendungen sollten Sie großzügig erhöhen, z.B. 24.

Effective Cache Size sollte als Faustregel 25% des physischen RAM betragen.

Diese und weitere Performance-Tipps für das jeweilige Betriebssystem finden Sie im PostgreSQL Administrator's Guide im Abschnitt "Run-Time Configuration".

Leider lassen sich Transaktionen für Postgres nicht abschalten, für ein (passives) Berichtssystem wie SuperX wären Transaktionen unbedeutend.

Autovacuum wurde mit Postgres 8 eingeführt.

Postgres-Nutzerkennungen mit Leserecht

Falls Sie für Testzwecke z.B. einen User "alex" anlegen wollen, der alle Tabellen einer DB lesen kann, geht dies mit Postgres so:

Zunächst müssen Sie in der pg_hba.conf sicherstellen, dass Passwort-Anmeldung über IP4 möglich ist:

```
# TYPE DATABASE USER ADDRESS METHOD
host all all -IP-Nummernkreis- md5
```

Dann starten Sie Postgres neu oder führen einen RELOAD aus, z.B. unter Linux:

```
/etc/init.d/postgresql reload
```

Dann gehen Sie als Superuser auf den DB-Server, und führen Sie in der Ziel-Datenbank folgende SQLs aus:

```
CREATE USER alex WITH NOSUPERUSER PASSWORD 'password';
GRANT USAGE ON SCHEMA public TO alex;
GRANT SELECT ON ALL TABLES IN SCHEMA public TO alex;
ALTER DEFAULT PRIVILEGES IN SCHEMA public GRANT SELECT ON TABLES TO alex;
```

Der letzte Befehl sorgt dafür, dass auch neue Tabellen Berechtigung erhalten.

Wenn Sie das Passwort später ändern wollen schreiben Sie

```
alter USER alex WITH PASSWORD 'anfang12';
```

Einspielen des Kernmoduls der SuperX-Datenbank

Für die Installation haben wir unten eine Kurzanleitung vorbereitet. Das Kernmodul der Datenbank liegt exportiert vor und kann in das DBMS übernommen werden. Die nachfolgenden Installationschritte gehen davon aus, daß Sie keinen speziellen DBSpace für SuperX vorgesehen haben.

Das Installationsscript für die Datenbank befindet sich im Verzeichnis

```
$$SUPERX_DIR/db/install/kernmodul_erzeugen.x -ggf. mit Name des DBSpace-
```

Das Script läuft nur, wenn die Parameter in der Datei \$\$SUPERX_DIR/db/bin/SQL_ENV stimmen. Bei erfolgreichem Ablauf kommt eine Erfolgsmeldung, im Falle eines Fehlers wird die Fehler-Logdatei create.log angezeigt. Wenn ein Fehler auftritt, müssen sie die Datenbank vor einem erneuten Ablauf des Scriptes dropfen.

Danach können Sie mit psql superx (unter Postgres) testen, ob die Datenbank verfügbar ist.

Schließlich sollten Sie die Tabelle hochschulinfo anpassen und die Daten Ihrer Hochschule dort eingeben, insbesondere die Hochschulnummer (apnr-Wert in cifx mit key=36).

Update und Sichern der Datenbank

Vor dem Start der Update-Scripte sollte immer eine Sicherung der Datenbank erfolgen. Für Backups ist es notwendig, die Datenbank regelmäßig zu exportieren. Beide Datenbanken bieten entsprechende Werkzeuge. Es bietet sich an, einen cronjob einzurichten, der zuerst das Backup vornimmt, und dann die einzelnen Module nacheinander aktualisiert.

Ein Beispiel-Eintrag der crontab des users superx liegt in \$\$SUPERX_DIR/db/module/crontab.sam. Ein Beispiel-Update-Script liegt in \$\$SUPERX_DIR/db/module/update.x.sam. Der Eintrag in der crontab, der das Script werktags um 18:00 Uhr startet, sähe dann wie folgt aus:

```
# Täglicher SuperX-Update um 18 Uhr
#
00 18 * * 1-5 /home/superx/db/module/update.x >/home/superx/db/module/update.log 2>&1
```

Ein Beispielinhalt für das Script update.x ist Teil des Kernmoduls: update.x

Ein Beispielscript, das die Datenbank sichert, liegt in \$\$SUPERX_DIR/db/install/dump_it.x. Es erzeugt den Dump im Verzeichnis \$\$SUPERX_DIR/db/install, prüft die erfolgreiche Sicherung und verschickt ggf. eine Fehler-Mail. Wenn Sie das Script in einem Cronjob betreiben wollen, müssen Sie als ersten Parameter \$\$SUPERX_DIR übergeben.

Die Rücksicherung einer Datenbank ist mit dem Script `$$SUPERX_DIR/db/install/restore_it.x` möglich.

Ein Dump unter Postgres

Postgres lässt sich auch im laufenden Betrieb sichern.

```
pg_dump -f superx.sql superx
```

Noch kompakter ist der Dump als Binärfiler mit dem Parameter `--format=c`:

```
pg_dump -f $DBNAME.sql --format=c $DBNAME
```

Sie können den Dump mit dem Parameter `--inserts` versehen, dies vergrößert den Dump beträchtlich. Dies ist eine sehr vorsichtige Einstellung, aber der Dump ist dadurch maximal kompatibel zu verschiedenen Postgres-Versionen, außerdem tauchen keine Probleme mit Umbrüchen in langen Textfeldern auf.

Wenn Ihnen die resultierenden Dumps zu groß sind, können Sie in einem eigenen Dump auf die Inserts verzichten (s.o.).

Anpassen der Datenbankparameter db.properties

Wenn Sie die Verfügbarkeit des Datenbankservers getestet haben, dann müssen die Datenbankparameter in die Datei

```
$$SUPERX_DIR/webserver/tomcat/webapps/superx/web-inf/db.properties
```

übertragen werden, damit das Servlet eine Verbindung zur Datenbank herstellen kann. Ein Muster für Informix und eines für Postgres ist bereits im Kernmodul enthalten. Kopieren Sie die Datei `db-informix.properties` bzw. `db-postgres.properties` nach `db.properties`. Das voreingestellte Passwort lautet hier "anfang12".

Zur Erstellung und ggfs. Änderung dieser Datei gibt es ein Tool:

```
propadmin.x
```

Das Shellsript liest aus der Umgebungsvariable `$DB_PROPERTIES` (oder über den ersten Parameter) den Speicherort der `db.properties` ein; in der Regel muss das die obige Position sein, damit das Servlet die Datei findet. Ausnahmen gibt es nur, wenn SuperX über den jdbc-Cient auf eine andere Datenbank zugreifen soll.

Füllen Sie die Felder entsprechend des folgenden Beispiels (Postgres):

Hinweis für Postgres: Wenn Sie Postgres auf einem anderen Port als dem voreingestellten 5432 betreiben, müssen Sie im jdbc-Treiber als Connection-URL den Port wie folgt angeben:

```
connectionURL=jdbc:postgresql://localhost:Portnummer/superx
```

- Die Parameter für den **LogLevel** können auf einer Skala von fünf Stufen gewählt werden: FINEST bis SEVERE. Bei FINEST wird fast alles geloggt, bei SEVERE werden nur Fehler geloggt.
- Nur für Informix: Im Entwicklungsmodus werden alle SQL-Befehle von Abfragen einzeln an die Datenbank geschickt. Das dauert etwas länger, ermöglicht aber bessere Fehlermeldungen. Man kann diese Einstellung auch im laufenden Betrieb ändern.
- Die Parameter *im Cache* legen fest, wie viel Information gecached werden sollen. Standardmäßig wird nichts gecached, aber im Produktiveinsatz sollten hier die entsprechenden Parameter gewählt werden.
- In den Connection-Pool Angaben wird angegeben, wieviele Verbindungen maximal gleichzeitig vom Servlet zur Datenbank hergestellt werden sollen.

Durch Anklicken von **OK** wird die Datei `db.properties` (bzw. der Pfad zum Inhalt der Umgebungsvariable `DB_PROPERTIES`) erstellt, wobei das Passwort verschlüsselt wird. Vorher sollten Sie mit "Verbindung Testen" prüfen, ob eine Datenbankverbindung hergestellt werden kann. Wenn dies nicht klappt, sollten die Fehlermeldungen weiterhelfen.

Wenn Sie einen UNIX / LINUX-Server für Tomcat betreiben wollen, dann ist es möglich, daß Sie unter Linux keine graphische Java-Umgebung starten können. In diesem Fall müssen Sie das Kernmodul auf einem Rechner mit installiertem Java und graphischer Umgebung kopieren, das Programm dort aus der Konsole starten und die Parameter ändern (wichtig: der Rechner muss die gleiche Zeichenkodierung haben, also UTF-8). Danach kopieren sie die Datei `db.properties` mit `scp` / WinSCP auf den UNIX-Rechner. Alternativ können Sie die Parameter mit dem vi bearbeiten. Wenn der Propadmin ohne graphische Umgebung gestartet wird, kann lediglich das Passwort eingegeben werden.

Wenn Sie Tomcat auf einem anderen Rechner als dem Datenbankservers betreiben, müssen Sie die Startdateien `propadmin.bat` bzw. `propadmin.x` im Verzeichnis

```
$$SUPERX_DIR/webserver/tomcat/webapps/superx/WEB-INF
```

benutzen (in diesem Falle ist das Verzeichnis `$$SUPERX_DIR/db` nicht notwendig.)

Installation und Pflege des Applikationsserver

Die folgenden Anleitungen gehen davon aus, dass Sie als Installationspfade für den Applikationsserver `C:\superx\webserver` (unter win32) und `/home/superx/webserver` (unter UNIX / LINUX) gewählt haben. Sie können natürlich auch andere Pfade wählen, müssen dann aber die Pfade in dieser Dokumentation entsprechend umsetzen. Fehlende oder falsche Pfade bzw. Umgebungsvariablen sind in Java- und Datenbankprojekten eine wichtige Fehlerquelle (z.B. unter LINUX die Groß- / Kleinschreibung).

Installation von Java

Achtung: Für SuperX benötigen Sie serverseitig Java 1.8 oder Java 1.11 in der aktuellen Version. Demensprechend benötigen Sie auch Tomcat 8.* oder 9.*.

Bei Oracle gibt es seit 2019 nur noch kommerzielle Pakete vom Java / JDK, mit "Registrierungszwang". Unter Linux installieren Sie OpenJDK 1.8 mit den Paketquellen, also z.B. bei Ubuntu Linux:

```
apt-get install -y openjdk-8-jdk
```

Für andere System: der Download der quelloffenen OpenJDK-Pakete befindet sich für die jew. Versionen und Betriebssysteme hier:

- Java 11 und höher: <https://openjdk.org/projects/jdk/>
- Java 8: <https://wiki.openjdk.org/display/jdk8u/Main>

Einrichtung der Servlet-Engine

Die Servlet-Engine ermöglicht dem Applikationsserver das SuperX-Servlet auszuführen. Anders als andere Scriptsprachen (z.B. asp, PHP, Perl) für Webserver ist der Java-Code als Bytecode kompiliert; die Servlets werden normalerweise also nicht auf dem Webserver entwickelt und getestet, sondern auf einem eigenen Entwicklungsrechner.

Unsere Referenz für Servlet Engines ist seit 2003 [Tomcat](#), eine kostenlose und gleichzeitig umfassende Engine, die darüber hinaus auch recht leicht zu installieren ist und auf vielen gängigen Webservern läuft (Apache, IIS). Sie ist im Rahmen des Apache-Projektes frei verfügbar und distribuierbar. Tomcat ist von Oracle als Referenzimplementierung von Webapplikationen anerkannt, d.h. Sie sollten die Konfiguration mühelos auf andere Server übertragen können. Die Web-Applikation von SuperX läuft unter allen Tomcat-Versionen ab 8.*

Sie können und sollten Tomcat aus der jew. Distribution installieren, um die aktuellen Patches zu bekommen. Zu Test- und Entwicklungszwecken wird im Kernmodul der Tomcat 9.0.27 mitgeliefert. Wenn Sie das Kernmodul entpacken, ist Tomcat mitsamt dem SuperX-Kontext bereits installiert.

Sie können Tomcat auch direkt vom [Hersteller](#) herunterladen.

Die Installation von Tomcat für SuperX wird auf einer [eigenen](#) Website beschrieben. Die weitere Konfiguration ist im Folgenden beschrieben.

Steuerung des Servers: Die server.xml

Editieren Sie zunächst die Konfigurationsdatei

```
$$SUPERX_DIR/webserver/tomcat/conf/server.xml
```

Hier werden die Ports und Anbindungen der Tomcat-Implementation angepasst. Standardmäßig läuft Tomcat auf dem Port 8080, und die Apache-Anbindung auf dem Port 8009. Weiterhin muss der Port 8005 für den Shutdown frei sein. Die Apache-Anbindung ist weiter [unten](#) dokumentiert.

Wichtig beim Betrieb des Tomcat mit UTF-8-Codierung: Der jew. Connector muss das weitere Attribut `URIEncoding="UTF-8"` aufführen. Wenn z.B. der Connector 8080 genutzt wird, sieht das so aus:

```
<Connector port="8080" maxHttpHeaderSize="8192"
maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
enableLookups="false" redirectPort="8483" acceptCount="100"
connectionTimeout="20000" disableUploadTimeout="true"
URIEncoding="UTF-8"/>
```

Dies ist wichtig für den Ajax-Client.

Analog bei ISO Installationen:

```
...URIEncoding="ISO-8859-1"
```

Nach einer Änderung müssen Sie Tomcat neu starten.

Datenbankverbindung für DBFORMS

Die Datenbank-Verbindung für DBFORMS und Saiku wird in der Datei `$$SUPERX_DIR/webserver/tomcat/webapps/superx/WEB-INF/dbforms-config.xml` definiert. Am Ende steht:

```
<dbconnection conClass="org.apache.commons.dbcp.PoolingDriver"
name="jdbc:apache:commons:dbcp:default"
isIndi="false"
default="true"
defaultConnection="true"/>
```

Damit wird der Connection Pool genutzt, der in der `db.properties` definiert ist. Nach einer Änderung muss man Tomcat neu starten.

Die Datei conf/web.xml

In der Datei `$$TOMCAT_HOME/conf/web.xml` sie u.a. die serverweite "Welcome-Page" bzw. deren Reihenfolge., welche wiederum Dateien anzeigen, wenn der Anwender ein Verzeichnis ohne Dateinamen aufruft, z.B. <http://servername/superx/>:

index.html

index.htm

index.jsp

Wenn Sie z.B. die Reihenfolge so ändern, dass zuerst die Datei index.jsp angezeigt wird (sofern sie existiert), dann können Sie eigene "Homepages" definieren, die nicht von der SuperX-Distribution (z.B. bei Updates) überschrieben würden. Außerdem ist dies eine sinnvolle Sicherheitsmassnahme, weil so keine Directory Listings angezeigt werden.

Änderungen in der Datei web.xml in der Webanwendung (-Webanwendung-/WEB-INF/web.xml) überschreiben die Einträge in der serverweiten web.xml.

Weitere Konfigurationsmöglichkeiten (Server Side Includes etc). sind in dieser Datei dokumentiert.

Vergleiche auch den unten folgenden Abschnitt zur Einrichtung der SuperX-Servlets unter Tomcat.

Administrator und Manager

In der Tomcat Auslieferung gibt es Beispiel-Webanwendungen, bitte löschen Sie diese.

Einrichten der SuperX-Servlets unter Tomcat

Allgemeines zur web.xml

Die Datei \$WEBAPP/WEB-INF/web.xml enthält die Konfiguration der Webanwendung, z.B. maximale Zeilenanzahl bei Berichten. Diese Datei wird initial ausgeliefert, bei Upgrades des Kernmoduls aber nicht automatisch überschrieben. Im Zweifelsfall stellen Sie die Auslieferungsdatei wieder her, es Muster dafür befindet sich in der Datei \$WEBAPP/WEB-INF/web.xml.superxonly.

Eine Änderung in der web.xml macht einen Tomcat Neustart erforderlich. Achten Sie auf die wohlgeformte XML Notation, wenn die Datei nicht wohlgeformt ist, startet der Server gar nicht mehr. Nutzen sie daher am Besten einen XML-fähigen Editor.

Ab Kernmodul 4.9 benötigen Sie in der Datei am Ende vor dem ersten "" Eintrag folgende Konfiguration:

Jersey REST Service

com.sun.jersey.spi.container.servlet.ServletContainer

com.sun.jersey.config.property.packages

de.superx.sxrest

1

Jersey REST Service

/sxrest/*

spring

/oauth/*

Für SuperX-Standalone-Betrieb (also unabhängig von HisInOne) ist ab Kernmodul 5.0 der context-Parameter zentral

http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd" version="2.4">

SuperX

SuperX

superXStandalone

true

Anpassen der Datei db.properties mit den Datenbank-Zugangsdaten (siehe oben Anpassen der Datenbankparameter für das SuperX-Servlet) ist Voraussetzung dafür, dass der Applikationsserver auf die Datenbank zugreifen kann.

SuperXManager Servlet

In der web.xml werden im SuperXManager Servlet einige wichtige Konfigurationen vorgenommen. Zentrales Servlet ist der SuperXManager, für den ein Eintrag benötigt wird.

```
<servlet>
  <servlet-name>
```

Im folgenden werden einige Parameter beschrieben:

- XSL-Prozessor (bei Bedarf kann Saxon-als XSL-Prozessor definiert werden, wenn der folgende Eintrag aktiviert wird):

```
<init-param>
  <param>
```

- Max. Zeilenanzahl: Die maximale Anzahl von Datensätzen, die eine Abfrage zurückliefern sollte, sollte jetzt beim SuperX-Manager angegeben werden, nicht mehr beim SuperXDBServlet, im Normalfall reicht der Standardwert von 20000 der ohne init-param als default genommen wird

```
<init-param>
  <param>
```

- field1Cache

```
<init-param>
  <param>
```

- ResponseCompression

```
<init-param>
  <param>
```

- [Sessionbasierte Felddefaults](#) (Vorbelegung gleichnamiger Felder in unterschiedlichen Selektionsmasken)

```
<init-param>
  <param>
```

ResponseCompression

Default für Response Compression ist true, dann braucht der init-param auch nicht angegeben zu werden. Die RTWH Aachen nutzt allerdings einen ReverseProxy der mit der gzip-Kompression nicht klar kam, in diesem Fall kann man durch setzen des ResponseCompression-Initparams mit param-value=false die Kompression ausschalten.

field1Cache

Neu in SuperX3.5rc2 ist die Möglichkeit einen sogenannten field1Cache für Auswahllisten (Feldart 1) zu nutzen.

Wenn ein entsprechender init-param beim SuperXManager definiert ist, lädt sich der webservice beim Start Inhalte für die angegeben Felder der Feldart 1 (aus felderinfo), in denen es keine dynamischen Tags gibt (wie z.B. <

Als Param-value muss eine where-Bedingung für einen select auf die Tabelle felderinfo angegeben werden. Sie können das Beispiel tid>10000 belassen oder bei Bedarf bestimmte Felder auslassen, z.B.

```
tid > 10000 and name not in ('Haushaltsjahr','Semester').
```

Der Cache wird aktualisiert, wenn im SuperXManager-Servlet auf den Button "Server-Cache aktualisieren" geklickt wird oder der Webserver neu gestartet wird. Außerdem wird er jeden Morgen einmal automatisch aktualisiert. Felder die sich zusätzlich zu den nächtlichen Updates dynamisch ändern, sollten ausgeschlossen werden, damit sie immer aktuell aus der Datenbank geholt werden.

Session-Timeout

Ein weiterer Parameter für die gesamte Webapplikation, der aber nur im XML-Frontend ausgewertet wird, lautet (siehe Beispiel-web.xml in unserem Kernmodul, ganz am Ende der webapp-Deklaration). Dieser Wert beschreibt die "Lebenszeit" einer Anmeldung bei Inaktivität des Benutzers (in Minuten). Ein negativer Wert bedeutet, dass die Session nie beendet wird. Ein sinnvoller Wert ist z.B. 180 (3 Stunden). Je länger die Zeit, desto höher die Belastung des Servers.

CSV-Encoding beim Export

Ab Kernmodul 4.7 / HISinOne-BI 2017.12: Der CSV Export von Masken wird standardmäßig in der gleichen Kodierung vollzogen wie die des Servers. Sie können eine abweichende Codierung eingeben, wenn Sie z.B. den Server unter UTF-8 betreiben, aber den CSV Export für die Anwender/innen in ISO ausführen wollen, weil die meisten Anwender/innen unter Windows mit Excel arbeiten. In diesem Falle setzen Sie für das Servlet SuperXManager den init-param "CSV-Encoding" auf den Wert ISO-8859-1:

SuperXManager

```
de.superx.servlet.SuperXManager
```

```
...
```

```
CSV-Encoding
```

```
ISO-8859-1
```

Nach einer Änderung müssen Sie Tomcat neu starten.

Fehlerseiten

Sie können auch durch spezielle Fehlerseiten die normale Fehlerausgabe des SuperX-Servlets sperren.

Fügen Sie an das Ende der web.xml vor dem End-Tag "" z.B. folgende Elemente ein	...
	500 /error.htm

Die ist die Voreinstellung bei Neuinstallation von SuperX, ältere Installationen müssen dies ggf. nachholen.

Sie können auf verschiedene Fehler-Codes sowie Exception-Types eigene Fehlerseiten definieren. Details dazu finden Sie in der Dokumentation Ihres Applikationservers.

Start des Tomcat

Vor dem Start müssen die Umgebungsvariablen der Datei \$SUPERX_DIR/db/bin/SQL_ENV geladen werden.

Die Umgebungsvariable JAVA_HOME muss korrekt gesetzt sein

- Das aktuelle Verzeichnis sollte im PATH sein (ggfs. /etc/profile oder .profile bzw. bashrc PATH=PATH\$:.:export PATH
- Melden Sie sich ab und wieder an
- \bin\startup.sh ausführen (zum Beenden shutdown.sh).

Das Terminal-Fenster zeigt den Port an, auf dem Tomcat läuft, z.B. 8080; um die Engine zu testen, kann man einen Webbrowser (zur Not auch lynx) starten und die Seite [http://localhost:8080/...](http://localhost:8080/) aufrufen.

Beendet wird Tomcat mit dem Befehl: shutdown.bat für MS-DOS bzw. shutdown.sh für UNIX.

Die Übertragung der Web Application

Wenn Sie die SuperX-Webapplikation auf einem vorhandenen Tomcat installieren wollen, folgen Sie dem Leitfaden auf der Seite

http://www.superx-projekt.de/doku/kern_modul/tomcat/f_TomcatusLinuxDistributionen.htm

LDAP Anbindung

Es ist möglich, die Authentifizierung von Usern über eine LDAP-Datenbank laufen zu lassen.

Die User müssen aber auf jeden Fall in SuperX auch existieren und entsprechende Gruppenzugehörigkeiten und Rechteinstellungen haben (Tabellen userinfo, groupinfo, user_group_bez, user_masken_bez, group_masken_bez, user_sachgeb_bez, group_sachgeb_bez, user_sichten etc).

Zertifikat einspielen

Falls Sie ein eigenes Zertifikat für die Verschlüsselung Ihres LDAP-Servers einsetzen, muss dies importiert werden in die Java Runtime.

Danach müssen Sie das Zertifikat auch Tomcat bzw. der JVM bekannt machen, die Tomcat startet. Erweitern Sie die Umgebungsvariable CATALINA_OPTS um den Parameter:

```
-Djavax.net.ssl.trustStore=<
```

also z.B.

-Djavax.net.ssl.trustStore=/usr/local/tomcat/conf/cacerts

Anpassen der superx_ldap.properties

Kopieren Sie die Datei superx/WEB-INF/superx_standalone_ldap.properties.sam nach superx_standalone_ldap.properties und bearbeiten Sie diese – wenn SuperX beim Serverstart diese Sie finden auch in der catalina.out einen Hinweis „Passwortkontrolle via LDAP wird aktiviert“. D.h. es wird nicht mehr auf die Tomcat eigene LDAP security-Technik gesetzt wie bis Ke

In der sam-Datei ist eine Verbindung zu einem öffentlichen LDAP-Testserver vorkonfiguriert. Dort gibt es einen User „einstein“ mit Passwort „password“. Wenn Sie in SuperX einen Use

```
# SuperX LDAP Konfiguration für Passwortkontrolle
#Falls SuperX standalone betrieben wird, können hier LDAP Parameter hinterlegt werden
# Beispielparameter verwenden
# https://www.forumsys.com/tutorials/integration-how-to/ldap/online-ldap-test-server/
# Beispieluser auf diesem System einstein Passwort=password
# Wenn Sie in Superx einen User mit Kennung einstein anlegen, können Sie die Anmeldung via LDAP # testen
LdapUrl=ldap://ldap.forumsys.com:389
#
#Base Definition für die LDAP Suche, kann ggfs. um Gruppen erweitert werden, wie
# ou=superx_user,dc=example,dc=com
LdapBase=dc=example,dc=com
#
#Falls für den LDAP Zugriff ein ServiceUser (mit eigenem Passwort) benötigt wird, kann dies hier angegeben werden
#LdapServiceUserDN=cn=read-only-admin,dc=example,dc=com
#LdapServiceUserPassword=password
#
# Identifying Attribute mit dem ein User mittels seiner Kennung gefunden wird (z.B. uid)
# z.B. bei Suchstring uid={0} einfach nur uid
LdapIdentifyingAttribute=uid
```

Optional können Sie bei Bedarf zusätzlich einen komplexeren Searchfilter definieren, z.B.

```
LdapSearchFilter=(&(uid=${0})(status=aktiv))
```



Passwortänderung: Falls ein Benutzer in SuperX dazu aufgefordert wird sein Passwort zu ändern, muss die Konstante „Passwortgültigkeit (Tage)“ angepasst werden. Diese Konstante dazu einfach auf einen möglichst hohen Wert

Trotz LDAP lokalen Login aktivieren

Wenn für einen Testbenutzer oder durch andere Gegebenheiten für einen Benutzer keine LDAP Kennung zur Verfügung steht, kann ab Kernmodul 5.2 für die index.jsp der Paramete

```
.../superx/index.jsp?login=superx
```

Damit wird dann für die verwendete Kennung das Passwort aus der userinfo von SuperX geprüft.

Shibboleth Anbindung

Es war möglich, die Authentifizierung von Usern über ein Shibboleth-Login laufen zu lassen.

Die User müssen aber auf jeden Fall in SuperX auch existieren und entsprechende Gruppenzugehörigkeiten und Rechteinstellungen haben (Tabellen userinfo, groupinfo, user_group_b)

Nach der Anmeldung in Shibboleth wird die Kennung an SuperX weitergeleitet und der User ist in SuperX angemeldet. Dazu muss unterhalb von der Webapplikation superx/ die inde

Integration von Tomcat mit dem Apache

Die Servlet-Engine Tomcat verfügt zwar über einen kleinen "eingebauten" Webserver, doch für den Echtbetrieb sollte man aus Performance-Gründen einen der marktgängigen Webserv

Umgang mit SSL Verschlüsselung

An mehreren Stellen können Sie mit SSL Verschlüsselung in Berührung kommen:

- Beim Datenbankzugriff mit SSL
- Beim Verschlüsseln im [Webserver](#)
- Beim Zugang zu einem SSL-verschlüsselnden LDAP Server

Im folgenden eine Anleitung zum Erzeugen und Bereitstellen von Zertifikaten.

Erzeugen eines SSL Zertifikats

Auf dem Rechner, der verschlüsseln soll, muss das Paket Openssl installiert sein. Ist dies der Fall, kann man als User root ein Zertifikat erzeugen.

Wir führen alle Schritte als user root durch, und gehen z.B. davon aus, dass wir uns im Verzeichnis /root befinden.

Zunächst muss ein Zertifikat erzeugt werden:

```
openssl req -new -nodes -newkey rsa:2048 -keyout mydomain.key -out mydomain.csr
```

Sie die jeweiligen Angaben (Land, Organisation etc.). Beim "Common Name" muss der DNS-Servername des Servers angegeben werden. Ein Challenge Passwort ist erst einmal nicht

Der private Schlüssel liegt in /root/mydomain.key

Der öffentliche Schlüssel hat das RSA Format. In Java Runtimes und im Browser wird ggf. nur ein X.509 Zeirtifikat erlaubt. Dazu müssen Sie das öffentliche Zertifikat zuerst in das e

```
openssl x509 -in mydomain.key -out capub.crt -outform DER
```

Es wird die Datei /root/capub.crt erzeugt.

Alternatives Vorgehen unter Ubuntu mit Ziel Zertifikat für Postgres:

```
openssl req -x509 -newkey rsa:4096 -keyout key.pem -out cert.pem -days 365
```

Eine passphrase ist nicht hilfreich, wenn Server automatisch neu gebootet werden können soll.

So kann man eine Entfernen:

```
openssl rsa -in [file1.key] -out [file2.key]
```

<https://knowledge.digicert.com/solution/SO5292.html>

Erzeugen eines Zertifikat-Request für eine Zertifizierungsstelle

Sie können ein neues Zertifikat erstellen und direkt einen Request für eine Zertifizierungsstelle erzeugen. Der Schlüssel sollte mind. 2048 bit haben. Das geht mit dem Befehl:

```
openssl req -new -nodes -newkey rsa:2048 -keyout mydomain.key -out mein_request.csr
```

Es wird ein Request in die Datei ./mein_request.csr geschrieben.

Letzteren übergeben Sie an die Zertifizierungsstelle. Achten Sie dabei darauf, dass die Angabe bei "Common Name" exakt dem Domainnamen entspricht. Wie uopenssl req -new -nod

Importieren des Zertifikats in Java

Wenn Client Programme wie z.B. Java auf einen SSL verschlüsselten Server zugreifen, der über ein selbst signiertes Zertifikat verfügt, dann muss man das Zertifikat in den TrustStore

Es gibt einen systemweiten Truststore , wenn Sie das systemweiten TrustStore von Java verwenden, liegt dies in \$JAVA_HOME/jre/lib/security/cacerts. Dann würde der Befehl lauten:

```
sudo keytool -import -alias myssl -file /root/capub.crt -keystore $JAVA_HOME/jre/lib/security/cacerts
```

Ggf. müssen Sie hier das Passwort des keystore eingeben (Standard wenn nicht verändert: changeit) . Danach kommt die Sicherheitsabfrage

```
Diesem Zertifikat vertrauen? [Nein]: Ja  
Zertifikat wurde zu Keystore hinzugefügt.
```

Wenn Sie für Tomcat einen speziellen Truststore definieren, z.B. durch den Tomcat Start Parameter

```
-Djavax.net.ssl.trustStore=/usr/local/tomcat/conf/cacerts
```

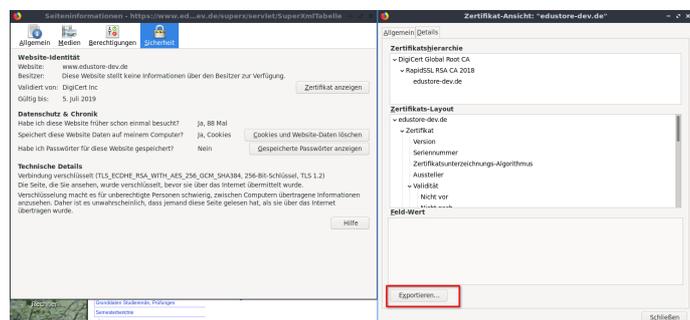
dann müssen Sie den Zielpfad für den Import entsprechend anpassen:

```
sudo keytool -import -alias myssl -file /root/capub.crt -keystore /usr/local/tomcat/conf/cacerts
```

In der HISinOne-BI kann der TrustStore in der globalen Konfiguration im Parameter KEYSTORE=... gesetzt werden. Das Vorgehen wäre hier analog.

Achtung: Wenn Sie eine eigene Datei als keystore nutzen wollen, gibt es in manchen Tomcat/Java Konstellationen Probleme, z.B. bei Tomcat 7.0.22 und jdk1.7.0_51. Verwenden Sie

Falls Sie das Zertifikat des Ziel-Servers nicht zur Hand haben, können Sie es auch direkt herunterladen. Gehen Sie im Browser auf die entsprechende URL und klicken Sie vor der



Die Zieldatei bekommt eine Endung `".crf"`. Alternativ geht auch die Linux Kommandozeile:

```
openssl s_client -connect meinserver.de:443 -servername meinserver.de:443 < /dev/null | sed -ne '/-BEGIN CERTIFICATE-/-END CERTIFICATE-/' < /dev/null | sed -ne '/-BEGIN CERTIFICATE-/-END CERTIFICATE-/' >
```

Falls Das Zertifikat in x509 format benötigt wird, bitte folgenden Befehl verwenden:

```
openssl x509 -in <(openssl s_client -connect ads.hs-karlsruhe.de:636 -prexit 2<(openssl s_client -connect ads.hs-karlsruhe.de:636 -prexit 2>
```

Danach können Sie das Zertifikat importieren mit

```
$JAVA_HOME/bin/keytool -importcert -alias myssl -keystore $JAVA_HOME/jre/lib/security/cacerts -trustcacerts -file ./my-ca.crf
```

Der Befehl keytool ist recht flexibel, man kann damit auch Zertifikate anschauen (-list) oder löschen (-delete). Details liefert die Ausgabe von keytool -help.

Mailversand

Zum Mailversand von Logmails installieren Sie ein UNIX Mailprogramm. Unter Ubuntu Linux ist dies z.B. s-nail, das ein Binary namens "s-nail" bereitstellt:

```
apt-get install -y s-nail
```

Standardmäßig sucht s-nail den SMTP Server auf localhost, typischerweise muss ein externer Mailserver angebunden werden. Dazu legen sie im Homeverzeichnis des Benutzers eine

```
set smtp="smtp://smtp.meinmailserver.de:587"
set from="superx@localhost"
set smtp-auth=login
set smtp-auth-user="meinemailkennung"
set smtp-auth-password=meinpasswort
```

Hinweis: es gibt auch andere Mailclients wie mutt oder mailx, wir haben s-nail gewählt weil die Syntax der obigen .mailrc-Datei und z.B. bzgl. den Anfügens von Attachments (z.B. für

Sie können den Mailversand dann testen z.B. mit

```
echo test | s-nail -s Testbetreff supervisor@meinmailserver.de
```

Sie können auch Mailverteiler einrichten, mit dem Befehl alias:

```
alias sxadmin supervisor1@meinmailserver.de supervisor2@meinmailserver.de
```

Sie können den Alias als Adressaten verwenden, z.B. mit

```
echo test | s-nail -s Testbetreff sxadmin
```

Test- und Produktivsystem synchronisieren

An einigen Hochschulen gibt es ein Testsystem in dem Entwicklungen getestet werden und ein Produktivsystem in dem die stabilen Entwicklungen dann übertragen werden. Die Schwi

Über die Bordmittel der Laderoutinen (Unload / Copy / Upload) können also Konfigurationen übertragen werden.

Details zu den Laderoutinen jeweils für

- HISinOne-BI: https://wiki.his.de/mediawiki/index.php/Komponentenverwaltung_der_HISinOne-BI
- SuperX: [Modulverwaltung](#)

Im Kontext der Säulenübertragung in HISinOne kann neben der Copy-Funktion auch SVN / Git genutzt werden, Sie müssen dazu nur die Dateien

```
webapps/superx/WEB-INF/conf/edustore/db/install/rohdaten/unl/*.unl
```

aus dem SVN-Ignore /GIT-Ignore entfernen, auf dem Entwicklungssystem committen und auf der Zielsäule updaten.

Entladeparameter

Über die Entladeparameter können Sie steuern, welche Daten übertragen werden sollen. Hier gibt es default-Werte, die Sie in der KERN_ENV selber anpassen können. Falls Sie kein

UNLOAD_USERRIGHTS

Hier handelt es sich um die Benutzerrechte aus dem Kernmodul. Es werden also die Benutzer, Gruppen, Sichtenrechte... übertragen. Spezielle Rechte wie die Kamerale Rechte des

UNLOAD_FIN_USER_KAM

Hierbei handelt es sich nur um die Kamerale Rechte des FIN Moduls.

UNLOAD_KONSTANTEN

Hierbei handelt es sich um die Konstanten.

UNLOAD_UNLOAD_PARAMS

Hierbei handelt es sich um die Entladeparameter.

UNLOAD_REPOSITORY

Hierbei handelt es sich um die Repository Variablen

UNLOAD_HOCHSCHULINFO

Hierbei handelt es sich um die Tabelle Hochschulinfo mit Informationen wie Name, Anschrift... der Hochschule.

UNLOAD_THEMENBAUM

Hierbei handelt es sich um die Menüstruktur des Informationssystems.

UNLOAD_MASKEN

Hierbei handelt es sich um alle Berichte.

UNLOAD_STYLESHEETS

Hierbei handelt es sich um die Styles der Berichte. Achtung: Es werden nur die Datenbankeinträge übertragen keine Dateien aus dem Dateisystem. Die Stylesheets müssen daher ext

UNLOAD_MAKROS

Hier werdne die Makros übertragen.

UNLOAD_CAPTIONS

Hierbei handelt es sich um die Beschreibungen von Feldern, Erläuterungstexten ...

UNLOAD_SICHTEN

Heirbei handelt es sich um die Sichten

UNLOAD_MAN_CATALOGUE

Hierbei handelt es sich um den Zahlenkatalog des MAN Moduls.

UNLOAD_MAN_ZAHL_WERT

Hierbei handelt es sich um eigene Werte der Hochschule für das MAN Modul.

UNLOAD_KENN_ZAHL_WERT

Hierbei handelt es sich um eigene Werte der Hochschule für das KENN Modul.

Ausführung

Wie üblich muss in der SQL_ENV die DB Verbindung eingerichtet werden und auch die Entladeparameter. Über die Scripte kern_unload.x werden dann die Tabellen aus dem jeweils

Upgrade einer bestehenden SuperX-Installation

Der Update eines bestehenden SuperX ist nicht trivial: Es kursieren verschiedene SuperX-Versionen, und das System ist offen für Änderungen durch den Benutzer. Deshalb müssen d

Stattdessen sollte Sie immer das passende Upgrade- bzw. "Patch"-Paket herunterladen. Die von Ihnen genutzte Version (zu finden in der Datei \$SUPERX_DIR/db/install/VERSION) gibt

Patch einspielen

Sie können einen Patch von der Seite <http://download.superx-projekt.de/> herunterladen. Dabei ist zu beachten welches System und welche Codierung benötigt wird. Informationen über

Die zip-Datei gibt es in drei Varianten:

- SuperX klassisch (UTF-8): Die Ordnerstruktur in \$SUPERX_DIR mit db/. und webservice/...
- SuperX klassisch (ISO): Die Ordnerstruktur in \$SUPERX_DIR mit db/. und webservice/..., aber im ISO Format
- SuperX webapps (UTF-8): Die Ordnerstruktur in \$WEBAPP mit \$SUPERX_DIR in \$WEBAPP/WEB-INF/conf/edustore. Diese Struktur wird in HISinOne-BI sowie in lokalen Entwicklungssystemen genutzt.

Wählen Sie die Datei, die zu Ihrer Installation paßt, und legen Sie sich für die Patches ein Verzeichnis auf dem SuperX Server an, z.B. /home/superx/patches. In dieses Verzeichnis l

Um Patches in SuperX einzuspielen gibt es das Script \$SUPERX_DIR/db/bin/patch_apply.x. Das Script starten Sie direkt aus dem Patchordner, in dem der zu installierende Patch liegt

```
patch_apply.x -PatchFile-
```

Ein Beispiel:

```
patch_apply.x patch_2011-06-01_superx_iso.zip
```

Dabei wird in dem Verzeichnis der Patch entpackt und ausgeführt. Je nach Patch muss auch ein Tomcat Neustart erfolgen, ein entsprechender Hinweis befindet sich in der Patch-Do

Weitere Hinweise finden Sie im Verzeichnis der [Shellscripte](#).

Upgraden des Kernmoduls

Standardvorgehen beim Upgrade

Beim Upgrade des Kernmoduls gibt es ab Version 4.0 ein standardisiertes Vorgehen. Hier das Vorgehen für Kernmodul 4.x oder höher "in short":

Beenden Sie Tomcat, und machen Sie eine Sicherung der Datenbank mit

```
cd $SUPERX_DIR/db/install
dump_it.x
```



Um sicher zu gehen empfehlen wir anschließend von dem gesamten SUPERX_DIR ein Backup anzulegen, bevor Sie mit dem Upgrade beginnen.

Laden Sie das Kern- Patch-Paket für Ihr DBMS (Postgres) und Codierung (iso, utf-8) herunter, in der Regel kern5.1_superx_utf8_POSTGRES_patch.tar.gz, und entpacken Sie es in \$

```
cd $SUPERX_DIR/db/install/upgrade
kern_env_upgrade.x
./../bin/SOL_ENV
kern_upgrade.x
```

Danach starten Sie Tomcat neu.

Upgrade Kernmodul Besonderheiten

Java 7 wird nicht mehr unterstützt, und Java 17 oder 21 werden noch nicht unterstützt, wir empfehlen Java 11. Beachten Sie die Hinweise zu [Java](#).

Beim Umstieg von Kernmodul 4.x auf 4.9 gibt es noch eine wichtige Änderung: in der Datei `$SUPERX_DIR/db/bin/SQL_ENV` muss die Variable `CATALINA_OPTS` um den Passus `"-DSuperX-DB-PROPERTIES-SET=true -L`

Außerdem müssen Sie beim Umstieg aufs Kernmodul 4.9 die Datei `$WEBAPP/WEB-INF/web.xml` erweitern um die neue REST-Schnittstelle.

Migrationsprojekte

Es gibt verschiedene Szenarien zur Migration von SuperX, hierzu werden Empfehlungen gegeben.

Postgres: Wechsel auf der Zeichencodierung auf UTF-8

Unter Postgres bietet es sich an, von Postgres ISO-Codierung zu Postgres-UTF-8 Codierung zu wechseln. Für die Umsetzung der Zeichencodierung von Dateien gibt es in Postgres e

Exportieren Sie die ISO-Datenbank mit `pg_dump` im Format "plain text". Es wird eine sql-Datei erzeugt, im folgenden Beispiel für die Datenbank mit dem Namen "\$DBNAME":

```
pg_dump -f $DBNAME.sql $DBNAME
```

Falls Sie direkt beim Dump eine zip-Datei erzeugen wollen, nutzen Sie folgendes Script:

```
vpq_dump $DBNAME | gzip >$DBNAME.sql.gz 2>dump.err
```

Wechseln Sie dann in eine Postgres-Installation, die UTF-8 unterstützt, und erzeugen Sie die Datenbank neu:

```
createdb --encoding=UTF-8 $DBNAME
```

Dann importieren die Datenbank mit:

```
psql $DBNAME < $DBNAME.sql
```

Danach müssen Sie alle Dateien unterhalb von `$SUPERX_DIR` von ISO nach UTF-8 konvertieren. Bitte fertigen Sie zunächst eine Sicherung des Dateisystems unterhalb von `$SUPER`.

Zum Konvertieren sich die Shellscrippte vom SuperX Kernmodul 4.0 oder höher an. In short:

```
cd $SUPERX_DIR
sx_list_isofiles.x . >iso.txt
sx_recode_isofiles.x iso.txt
```

Am Ende müssen Sie noch in der Datei `$SUPERX_DIR/db/bin/SQL_ENV` die Variable `LANG` auf die UTF-8-Codierung setzen. Führen Sie dazu

```
locale -a | grep de
```

aus, um die verfügbaren Codierungen zu ermitteln, und setzen Sie dann den entsprechenden Wert, z.B.:

```
LANG=de_DE.utf8
export LANG
```

Außerdem ergänzen Sie im Parameter `CATALINA_OPTS` den Schalter `-Dfile.encoding=UTF-8`, z.B.

```
CATALINA_OPTS="-Xmx700M -XX:MaxPermSize=200m -Djava.awt.headless=true -DSuperX-DB-PROPERTIES-SET=true -Dfile.encoding=UTF-8"
```

Danach laden Sie einmal die Datei `SQL_ENV` neu:

```
.$SUPERX_DIR/db/bin/SQL_ENV
```

Danach müssen Sie in der Datei `$SUPERX_DIR/tomcat/conf/server.xml` den Parameter `URIEncoding="UTF-8"` ergänzen, z.B.

```
<Connector port="8080" protocol="HTTP/1.1"
connectionTimeout="20000"
redirectPort="8443" URIEncoding="UTF-8" />
```

Starten Sie dann noch Tomcat neu. Damit die die Migration nach UTF-8 beendet.

Migration von SuperX zu HISinOne / Edustore

Migration von Datenbank und Dateisystem

In HISinOne / Edustore ist folgendes fest vorgegeben:

- Das DBMS ist Postgres
- Die Zeichencodierung ist UTF-8

Vor einer Migration zu HISinOne / Edustore müssen Sie also o.g. zuerst erledigen. Eine Anleitung finden Sie [oben](#).

- Wenn die Migration zu Postgres / UTF-8 gelungen ist, müssen Sie wie folgt vorgehen:
- Entpacken Sie das HISinOne Release auf dem Server, und richten Sie den Oisserver ein (databases.xml etc).
- Bei der Neuinstallation von HISinOne müssen Sie normalerweise eine leere Postgres-Datenbank "eduetl" anlegen, dies in Ihrem Falle nicht notwendig. Sie verlinken die oben migrierte Datenbank in der databases.xml

- Sie starten in der BI-Administration den Upgrade der jeweiligen Module, beginnend mit dem Kernmodul. Dies können Sie über den Browser realisieren, oder über Shell- bzw. ANT Skripte

Sie können nach der Migration der Datenbank die Konnektoren auf BI-Technik umstellen. Sie können alternativ auch die bisherigen SuperX Shellscrippte weiterhin nutzen. Im HISinOne-

SUPERX_DIR	oben	superx/WEB-INF/conf/edustore
WEBAPP	webservertomcat/webapps/superx	superx
Kernmodul	\$(SUPERX_DIR)/db/install	superx/WEB-INF/conf/edustore/db/install

Bitte passen Sie daher Ihre [SQL_ENV](#) entsprechend an.

Einrichtung der BI Konnektoren

Die Technik der BI-Konnektoren ist identisch mit der von SuperX, allerdings lassen sich BI Konnektoren auch in der Komponentenverwaltung von HISinOne ausführen. Wenn Sie die r

org.postgresql.Driver

jdbc:postgresql:

sospos

soshost

5432

sospos

anfang12

y

Das analoge Beispiel in \$SOS_LOAD_PFAD/db-sos_pg.properties:

```
driverName=org.postgresql.Driver
connectionName=superx
connectionURL=jdbc:postgresql://localhost:5432/sospos
```

```
logLevelSQL=FINEST  
connectionPassword=sx_des\#-1\#108\#-75\#72\#-110\#-117\#-34\#-94\#116\#120\#-97\#-28\#-54\#-55\#11\#8
```

Sie übertragen also die Connection Angaben in die databases.xml und starten den Tomcat neu.

Hinweis: Wenn Sie andere Vorsysteme als Postgres oder Informix nutzen (z.B. MACH unter SQL-Server), müssen Sie auch den jdbc-Treiber der Anwendung vom Rohdatenpfad (z.B. \$FIN_LOAD_PFAD/lib/sqljdbc.jar) nach

Weitere Infos:

https://wiki.his.de/mediawiki/index.php/Komponentenverwaltung_der_HISinOne-BI

https://wiki.his.de/mediawiki/index.php/Komponenten_Update-HISinOne-BI#Skriptgesteuerter_Update

Tomcat aktualisieren

Wenn Sie den Tomcat benutzen, welcher mit SuperX ausgeliefert wird, wird dieser nicht automatisch vom System geupgradet. Dies muss manuell gemacht werden. Bei kleineren Vers

Dieser Leitfaden zur Aktualisierung des Tomcats ist ein Vorschlag von uns, wie Sie den Tomcat aktualisieren können. Bitte sichern Sie zuvor das Dateisystem von SuperX und die D

Bitte laden Sie sich hier: <http://tomcat.apache.org/> die Version des Tomcats herunter, welche Sie verwenden möchten.

Nun beenden Sie den Tomcat und benennen das Verzeichnis \$SUPERX_DIR/webserver/tomcat nach z.B. \$SUPERX_DIR/webserver/old_tomcat um (eventuell verwenden Sie noch ein C