



Administrationshandbuch DSS

erstellt durch

**ISB AG
Zur Gießerei 24
D-76227 Karlsruhe**

Inhaltsverzeichnis

1	Einleitung	1
2	Dokumentenhinweise	2
2.1	Legende	2
2.2	Glossar	2
3	Systemarchitektur	3
4	Installation, Deinstallation, Aktualisierung	4
4.1	Installation und Deinstallation der Software	4
4.2	Aktualisierung der Software	4
4.2.1	Aktualisierung des edoo ^{SVS} -Servers	4
4.2.2	Aktualisierung des edoo ^{SVS} -Clients	4
4.2.3	Aktualisierung der PostgreSQL-Datenbank	5
5	Konfiguration	6
5.1	Konfiguration des Betriebssystems	6
5.1.1	Einstellung von Datum und Uhrzeit	6
5.2	Konfiguration der PostgreSQL-Datenbank	6
5.2.1	Konfigurationsdatei postgresql.conf	6
5.2.2	Konfigurationsdatei pg_hba.conf	7
5.3	Konfiguration des edoo ^{SVS} -Servers	8
5.3.1	Konfigurationsdateien config.ini und config.local.ini	8
5.3.2	Konfigurationsdatei edoosys-server.cfg	10
6	Überwachung, Monitoring, Protokolle	11
6.1	Überwachung der Hardware	11
6.2	Überwachung des Betriebssystems	11
6.3	Überwachung der PostgreSQL-Datenbank	12
6.3.1	Monitoring über Systemtabellen der PostgreSQL-Datenbank	12
6.3.1.1	pg_stat_activity	12
6.3.1.2	pg_stat_all_tables	12
6.3.1.3	pg_statio_all_tables	12
6.3.2	Protokolldateien der PostgreSQL-Datenbank	12
6.4	Überwachung der edoo ^{SVS} -Anwendung	13
6.4.1	Protokolldateien der edoo ^{SVS} -Software	13
7	Regelmäßige Administrationstätigkeiten	14
7.1	Starten und Stoppen des Systems	14
7.1.1	Starten und Stoppen der Datenbank	14
7.1.1.1	Starten und Stoppen der Datenbank auf Windows-Systemen	14
7.1.1.2	Starten und Stoppen der Datenbank auf Linux-Systemen	15
7.1.2	Starten und Stoppen des edoo ^{SVS} -Servers	15
7.2	Betriebssystem-Updates	16
7.3	Update des Java Runtime Environment (JRE)	16
7.4	Updates der Datenbank und der Anwendung	16
7.5	Benutzerverwaltung	17
8	Bekannte Fehler und Probleme	18
8.1	Betriebssystem	18
8.2	PostgreSQL-Datenbank	18
8.2.1	edoo ^{SVS} -Server kann sich nicht mit der PostgreSQL-Datenbank verbinden	18
8.3	edoo ^{SVS} -Anwendung	18

8.3.1	edoo ^{SVS} -Server startet nicht.....	18
8.3.2	edoo ^{SVS} -Client kann sich nicht zum edoo ^{SVS} -Server verbinden.....	18
8.3.3	edoo ^{SVS} -Server kann sich nicht mit ZSS verbinden.....	19
8.3.4	Replikation einer Schule des edoo ^{SVS} -Servers schlägt fehl	19
9	Sicherung und Wiederherstellung	20
9.1	Sicherungs- und Wiederherstellungsstrategie.....	20
9.1.1	Systemausfall	20
9.1.2	Ausfall der Datenbank	20
9.1.3	Korrupte Daten	20
9.2	Sicherungs- und Wiederherstellungsarten	21
9.2.1	Replikation	21
9.2.1.1	Erstellen einer Replikations-Sicherung	21
9.2.1.2	Wiederherstellung der Schuldaten aus einer Replikations-Sicherung	23
9.2.2	Vollsicherung der Datenbank	24
9.2.2.1	Erstellen einer Vollsicherung der Datenbank	25
9.2.2.2	Wiederherstellung einer Vollsicherung der Datenbank	29
10	Dokumentenverweise	30
11	Anlagen.....	31
	Dokumentation dbctl.cmd.....	32
	Aufbau dbsettings.ini	33
	Aufbau ctlconfig.ini	34
	Anpassung ctlconfig.ini.....	34
	Mailversand	34

Abbildungsverzeichnis

Abbildung 1 Architektur edoo.sys	3
Abbildung 2 edoo.sys Update Mehrplatzclient	5
Abbildung 3 Update Mechanismus config.ini	5

1 Einleitung

Die Software **edoo^{SWS}** dient der Verbesserung sowie Vereinheitlichung der schulischen Datenbasis und bietet den Schulen eine moderne und umfassende, einheitliche Verwaltungs- und Planungssoftware für alle Verfahren der Schulverwaltung.

Die Anwendung **edoo^{SWS}** ist eine plattformunabhängige Java Anwendung im Windows Look and Feel auf Basis von Open Source Technologien. Das Design der Software verbindet dabei die Vorteile klassischer Desktop – und Webanwendungen.

Das vorliegende Handbuch richtet sich an das Administrationspersonal, welches für die Pflege, den Betrieb und die Konfiguration der Software **edoo^{SWS}** verantwortlich ist.

Es beschreibt zum einen die Administration von **edoo^{SWS}** in seinen verschiedenen Ausprägungen auf Server- und Client Seite und zum anderen alle wesentlichen Informationen, die für ein reibungsloses Zusammenspiel der verschiedenen **edoo^{SWS}** Komponenten von Bedeutung sind.

Dieses Handbuch ist als Ergänzung der Online-Hilfe der Anwendung und weiterer Handbücher konzipiert und setzt ein Basiswissen über die Terminologie und Struktur des **edoo^{SWS}** Systems voraus.

2 Dokumentenhinweise

2.1 Legende

Im folgenden Handbuch werden verschiedene Formatierungen und Hervorhebungen verwendet, die im Folgenden erklärt werden.

Tipp:

Hinweise zu den dargestellten Inhalten werden in einer blauen Box mit dem Zeichen für eine Information gekennzeichnet



Hinweis:

Wichtige Hinweise und typische Fehlerquellen werden in einer orangefarbenen Box mit dem Zeichen für Achtung gekennzeichnet

Quellcode oder
`user@host$ Befehle auf der Kommandozeile`
 werden in einer schwarzen Box und Maschinenschrift dargestellt

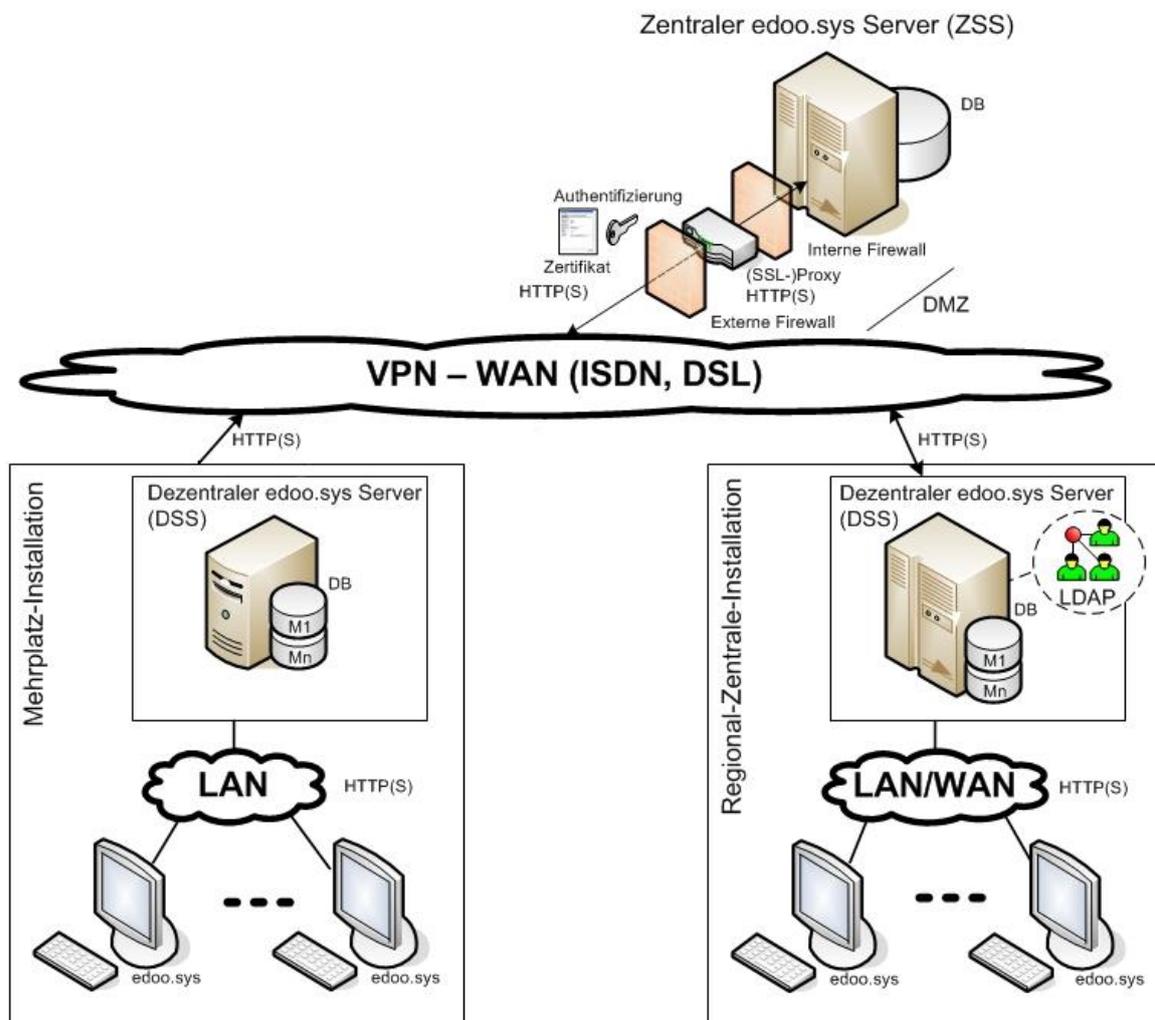
2.2 Glossar

Einige der im Dokument verwendeten Begriffe werden im nachfolgenden Glossar erläutert. Die Begriffe sind alphabetisch sortiert.

Begriff	Erklärung
DSS	Siehe edoo^{SVS} -Server
edoo^{SVS} -Server	Der Applikationsserver der edoo^{SVS} -Software. Er bildet zusammen mit dem edoo^{SVS} -Client die dezentrale Komponente und enthält große Teile der Anwendungslogik.
PostgreSQL	PostgreSQL ist ein weitverbreitetes Open Source-Datenbankmanagementsystem. Es wird in edoo^{SVS} als persistente Datenhaltungskomponente verwendet und dient somit der Speicherung der Anwendungsdaten.
PostgreSQL-Cluster	
PostgreSQL-Datenbank	
ZSS	Zentraler Schulserver, eine der zentral gehosteten Komponenten der edoo^{SVS} -Software.

3 Systemarchitektur

Einen abstrakten Gesamtüberblick über die Architektur des edoo^{SVS}-Systems in seinen verschiedenen Ausprägungen zeigt Abbildung 1.



M1,...,Mn: Mandanten (Dienststellen/Schulen)

Abbildung 1 Architektur edoo.sys

Das edoo^{SVS}-System gliedert sich in zwei Ebenen: die dezentralen Komponenten und die zentralen Komponenten.

Die dezentralen Komponenten sind die edoo^{SVS}-Clientsoftware und der dezentrale edoo^{SVS}-Server (DSS).

Die zentrale edoo^{SVS}-Komponente ist der Zentrale edoo^{SVS}-Server (ZSS). Der ZSS hat die zentrale Datenhaltung zur Aufgabe und ist ein Stellvertreter-Proxy für die bidirektionale Kommunikation zwischen edoo^{SVS} und anderen Systemen, z.B. zur Statistik-Abgabe.

Das vorliegende Administrationshandbuch bezieht sich ausschließlich auf den Betrieb des dezentralen edoo^{SVS}-Servers.

4 Installation, Deinstallation, Aktualisierung

4.1 Installation und Deinstallation der Software

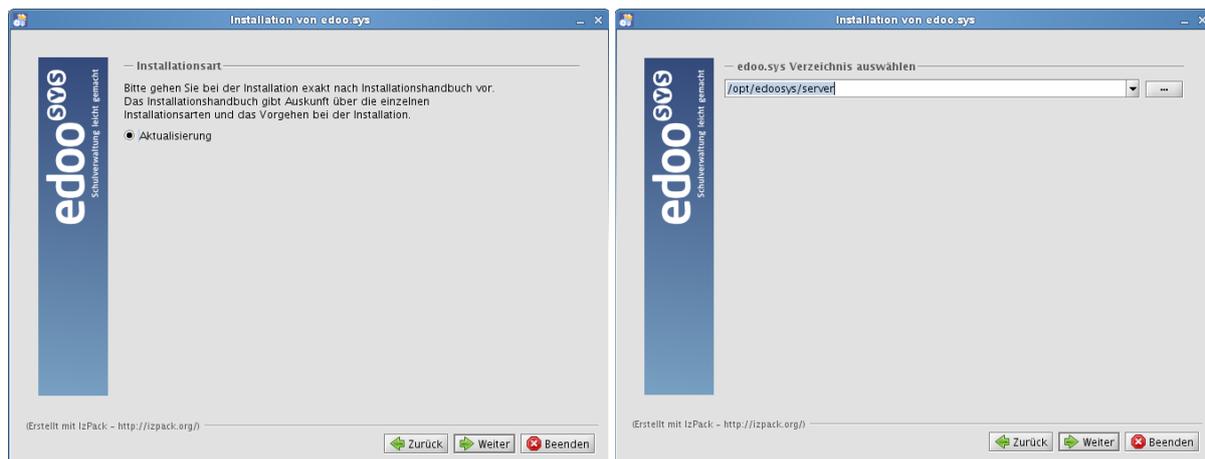
Alle Informationen zur Installation und Deinstallation des **edoo^{svs}**-Servers sowie eine detaillierte Schritt-für-Schritt-Anleitung sind im Installationshandbuch DSS [R1] enthalten.

4.2 Aktualisierung der Software

4.2.1 Aktualisierung des **edoo^{svs}**-Servers

Die Aktualisierung des **edoo^{svs}**-Servers erfolgt manuell und entspricht damit im Wesentlichen einer Neuinstallation, die im Installationshandbuch DSS [R1] detailliert beschrieben wird. Dabei wird im Gegensatz zur Neuinstallation das Update-Installer-Paket verwendet. Dieses trägt die Bezeichnung „..._rp_update_installer.zip“.

Der Installer ist daran zu erkennen, dass als Installationsmodus lediglich „Aktualisierung“ angeboten wird. Bei der Auswahl des Installationsverzeichnisses ist darauf zu achten, das bestehende **edoo^{svs}**-Server-Installationsverzeichnis anzugeben.



Rückfragen können mit „OK“ bestätigt werden.



Hinweis:

Bei der Neu-Konfiguration der Datenbank muss sichergestellt sein, dass zur bestehenden Datenbank „asv“ keine Verbindungen bestehen. Anderenfalls beendet sich die Installation und muss erneut aufgerufen werden.

4.2.2 Aktualisierung des **edoo^{svs}**-Clients

Die **edoo^{svs}** -Anwendung bietet einen automatischen Updatemechanismus für die Client-Komponente.

Um den Arbeitsaufwand für den Systembetreuer zu reduzieren genügt bei einer Mehrplatzinstallation im Falle einer Softwareaktualisierung die Aktualisierung des **edoo^{svs}**-Servers. Alle Mehrplatzclients, die sich mit diesem **edoo^{svs}**-Server verbinden, erhalten bei der nächsten Anmeldung automatisch eine Aktualisierung auf die neue Version.



Abbildung 2 edoo.sys Update Mehrplatzclient

Diese automatische Aktualisierung der Mehrplatzclients ist in der Konfiguration des edoo^{SVS}-Clients abschaltbar. Dies erfolgt in der Datei „config.local.ini“ im Unterverzeichnis „configuration“ der edoo^{SVS} - Software bzw. ist i.d.R. als Standard in der Datei „config.ini“ hinterlegt (siehe Kapitel 5.3.1).

```
# update
update.automatich=true
update.backup=true
update.server=http://127.0.0.1:8765/svp/update/
update.bundles.ignore=(launcher\\.jar)
```

Abbildung 3 Update Mechanismus config.ini

4.2.3 Aktualisierung der PostgreSQL-Datenbank

Die Aktualisierung der PostgreSQL-Datenbank auf die neueste, von der edoo^{SVS}-Software unterstützte, Version, entspricht einer Neuinstallation dieser:

1.	Eine Vollsicherung der Datenbank erstellen (siehe Kapitel 9.2.2.1.1) und den erstellten Dump lokal speichern.	
2.	Die PostgreSQL-Konfigurationsdateien der Datenbank sichern: <ul style="list-style-type: none"> • postgresql.conf • pg_hba.conf • pg_ident.conf 	host:~ #cp -a /var/lib/pgsql/data/...conf /tmp/
3.	Sowohl das Installations- als auch das Daten-Verzeichnis der PostgreSQL-Datenbank löschen.	host:~ #rm -r /usr/local/pgsql host:~ #rm -r /var/lib/pgsql/data
4.	Mit dem aktuellen Installationspaket eine Neuinstallation durchführen (siehe [R1]) <div style="border: 1px solid black; background-color: #ffffcc; padding: 5px; margin-top: 10px;"> <p>Tipp: Bei der Neuinstallation sollten die bisherigen Verzeichnisse für Installations- und Datenverzeichnis beibehalten werden.</p> </div>	
5.	Die in Schritt 1 erstellte Vollsicherung der Datenbank wiederherstellen (siehe Kapitel 9.2.2.2)	

5 Konfiguration

5.1 Konfiguration des Betriebssystems

5.1.1 Einstellung von Datum und Uhrzeit

Auf dem Host, auf dem der **edoo^{SVS}**-Server betrieben wird, sollte auf Betriebssystemebene ein NTP-Server konfiguriert sein, sodass jederzeit die aktuelle und korrekte Zeit als Systemzeit vorliegt. In der Anwendung **edoo^{SVS}** selbst können hierzu keine Einstellungen vorgenommen werden.

Ist kein NTP-Server vorhanden oder kann diese Einstellung aus anderen Gründen nicht vorgenommen werden, so sollte bei der Einstellung der Systemzeit auf eine automatische Umstellung von Sommer- auf Winterzeit geachtet werden.

5.2 Konfiguration der PostgreSQL-Datenbank

Die Konfiguration der PostgreSQL-Datenbank erfolgt über Konfigurationsdateien im Datenverzeichnis der jeweiligen Datenbank. Die wesentlichen Konfigurationsdateien sind

1. `postgresql.conf`
Die Datei „`postgresql.conf`“ enthält die meisten Einstellungsmöglichkeiten zur Datenbank. Hier lassen sich sowohl Parameter zur Beeinflussung der Performance als auch Logging und Monitoring konfigurieren.
2. `pg_hba.conf`
Die Datei „`pg_hba.conf`“ enthält alle Hosts, von denen ein Zugriff auf die PostgreSQL-Datenbank erfolgen darf. Sie dient damit als wesentlicher Baustein der Sicherheit der Datenbank.

5.2.1 Konfigurationsdatei `postgresql.conf`

Die Konfigurationsdatei „`postgresql.conf`“ befindet sich im Data-Verzeichnis der PostgreSQL-Datenbank und bringt eine Reihe von Einstellungen zur Beeinflussung des Systemverhaltens, z.B. zur Verbesserung der DB-Performance, mit sich.

Eine Änderung der Einstellungen bedingt teilweise einen Neustart des PostgreSQL-Servers (siehe Kapitel 7.1.1).

Tipp:

Bei einer Standard-Installation des **edoo^{SVS}**-Servers sind die Default-Einstellungen der PostgreSQL-Datenbank völlig ausreichend. Soll eine größere Anzahl an Schulen (Anzahl > 5) den gleichen **edoo^{SVS}**-Server verwenden, sollte eine Anpassung der Werte erfolgen.

Parameter	Wert(e)	Bedeutung
<code>max_connections</code>	100	Anzahl der Verbindungen, die die PostgreSQL-Datenbank akzeptiert.
<code>shared_buffers</code>		Anzahl der 8 KB-Blöcke, die PostgreSQL als Cache im Hauptspeicher zur Verfügung gestellt werden
<code>work_mem</code>		Speicher für PostgreSQL-interne Rechenoperationen wie z.B. Sortierungen.
<code>maintenance_work_mem</code>		Speicher für PostgreSQL-interne Optimierungsoperationen wie z.B. Indizierung oder Vacuum-Prozess.

Fsync	on	Legt fest, ob Daten auf persistenten Speicher, z.B. die Festplatte, geschrieben oder im Hauptspeicher gehalten werden.
checkpoint_segments		Maximale Anzahl an Segmenten im Write Ahead-Log zwischen zwei Checkpoints. Bei einem Checkpoint werden die im WAL befindlichen Änderungen in die eigentliche Tabellenstruktur der Datenbank geschrieben.
checkpoint_timeout		Maximale Dauer zwischen zwei Checkpoints.
effective_cache_size		Wert des gesamten der Datenbank zur Verfügung stehenden Caches.
track_activities	on	Aktiviert das Monitoring des in Laufzeit ausgeführten Kommandos jedes Serverprozesses
track_counts	on	Aktiviert Statistikerstellung über die Zugriffe auf Tabellen und Indizes
track_io_timing	on	Aktiviert Monitoring der Schreib-/Lesegeschwindigkeit
track_activity_query_size	1024	Legt die Anzahl der reservierten Bytes zur Überwachung der aktuell ausgeführten Kommandos fest
log_parser_stats	on/ off	Aktiviert Logging der Statistiken zum SQL-Statement Parser.
log_planner_stats	on/ off	Aktiviert Logging der Statistiken zum SQL-Statement Planner.
log_executor_stats	on/ off	Aktiviert Logging der Statistiken zum SQL-Statement Executor.
log_statement_stats	on/ off	Aktiviert Logging des gesamten SQL-Statements <div style="border: 1px solid black; background-color: #ffffcc; padding: 5px; margin-top: 10px;">  Hinweis: Diese Option kann nicht zusammen mit einer der anderen log_x_stats-Optionen aktiviert werden. </div>

5.2.2 Konfigurationsdatei pg_hba.conf

Die Konfigurationsdatei „pg_hba.conf“ befindet sich im Data-Verzeichnis der PostgreSQL-Datenbank. Sie enthält Einträge der Hosts und Benutzer, die Zugriff auf die PostgreSQL-Datenbank erhalten und schränkt den Zugriff auf die Datenbank auf diese benannten Komponenten ein. Die Einträge werden in der Form

TYPE	DATABASE	USER	ADDRESS	METHOD
------	----------	------	---------	--------

erstellt:

Parameter	Wert(e)	Bedeutung
TYPE	Local, host	Art des Zugriffs. local für lokale, host für remote-Zugriffe auf die Datenbank.
DATABASE	asv	Name der Datenbank, auf die Zugriff gegeben wird. „all“ für alle Datenbanken.
USER	asv	User, der für die Anmeldung verwendet werden darf. „all“ für alle User.
ADDRESS	IP-Adresse, z.B. 192.168.2.1/32	IP-Adresse des Hosts, von dem aus auf die zuvor genannte(n) Datenbank(en) zugegriffen werden darf.
METHOD	md5, trust,	Authentifizierungsmethode. „trust“, wenn dem Host vertraut und kein Passwort verlangt werden soll. „md5“ für Authentifizierung mit md5-Hash des User-Passworts.

Tipp:

Werden **edoo^{SVS}**-Server und PostgreSQL-Datenbank auf dem gleichen Host installiert, muss an der `pg_hba.conf` i.d.R. keine Änderung vorgenommen werden, da der lokale Zugriff standardmäßig gestattet ist.

Eine Änderung der „`pg_hba.conf`“-Datei bedingt immer einen Neustart des PostgreSQL-Servers (siehe Kapitel 7.1.1).



Hinweis:

Enthält die `pg_hba.conf` keinen Eintrag für den Host, auf dem der **edoo^{SVS}**-Server betrieben wird und der den Zugriff auf die Datenbank „`asv`“ gestattet, wird die **edoo^{SVS}**-Anwendung nicht starten. Es ist somit immer ein solcher Eintrag zu erstellen.

5.3 Konfiguration des **edoo^{SVS}**-Servers

Die Konfiguration des Servers erfolgt im Wesentlichen über einige wenige Konfigurationsdateien, die unterhalb des Installationsverzeichnis des **edoo^{SVS}**-Servers, unter Windows im Verzeichnis „`C:\Users\\AppData\Roaming\edoosys\configuration`“, abgelegt sind:

1. `config.ini` und `config.local.ini`
Die Datei „`config.ini`“ ist die Konfigurationsdatei der Anwendung. Sie ist im Verzeichnis `Root/configuration` zu finden und beinhaltet verschiedene Parameter der Systemkonfiguration (Detaillierung im Kapitel 5.3.1). Da in der „`config.ini`“-Datei keine Änderungen implementiert werden dürfen, werden diese über eine weitere Datei, die „`config.local.ini`“ ergänzt (siehe Kapitel 5.3.1).
2. `edoosys-server.cfg`
Die Datei „`edoosys-server.cfg`“ bestimmt das Laufzeitverhalten der Anwendung, gesteuert über Java Properties. Sie ist im Installationsverzeichnis des **edoo^{SVS}**-Servers zu finden. Einstellungen dieser Parameter bedingen einen Neustart der Anwendung.

5.3.1 Konfigurationsdateien `config.ini` und `config.local.ini`

Die Konfigurationsdatei „`config.ini`“ befindet sich im Unterverzeichnis „`configuration`“ des Installationsverzeichnisses des **edoo^{SVS}**-Servers und somit im Pfad `... \edoosys\server\configuration\config.ini`

Sie enthält vorgegebene Einstellungen zur Ausführung der Anwendung, die nicht verändert werden dürfen.

Müssen dennoch Konfigurationsparameter für den **edoo^{SVS}**-Server angepasst werden, so sind diese in einer separaten Datei „`config.local.ini`“ im gleichen Verzeichnis zu hinterlegen. Die Einstellungen der „`config.local.ini`“ überschreiben die in der „`config.ini`“ hinterlegten Einstellungen. Ein Beispiel für eine solche „`config.local.ini`“-Datei findet sich in den Anlagen.

Eine Änderung der Einstellungen über die Datei „`config.local.ini`“ bedingt einen Neustart des Servers (siehe Kapitel 7.1.2).



Hinweis:

Änderungen in der „`config.ini`“-Datei werden zur Folge haben, dass der **edoo^{SVS}**-Server nicht mehr startet. Diese darf somit in keinem Fall verändert werden.

Folgende Parameter können über die „config.local.ini“-Datei überschrieben werden:

Parameter	Wert(e)	Bedeutung
org.osgi.service.http.port	8765	Angabe des Ports für die Verbindungen über http
http.server.http.enabled	true	Aktivierung des http-Protokolls für Verbindungen
http.server.https.port	8763	Angabe des Ports für die Verbindungen über https
http.server.https.enabled	False	Aktivierung des https-Protokolls für Verbindungen
lf.log4j	true	
app.log.log4j.basedir	(=Installationsverzeichnis)	Verzeichnis, in dem die Log-Dateien des edoo ^{SVS} -Servers gespeichert werden.
update.backup	true	
installdir	Je nach Installation	Installationsverzeichnis des edoo ^{SVS} -Servers
remote.server.url	https://svp-prod-rlp.service24.rlp.de/svp	URL des ZSS
remote.server.keystore	certs/client.keystore	
remote.server.keystore.password	edoo	
remote.server.keystore.type	JKS	
replication.server.url	https://svp-prod-rlp.service24.rlp.de	URL des ZSS bzw. kaskadierten Replikationsservers (siehe Kapitel 9.2.1)
replication.server.cleanup	True	Löschen der auf den Replikationsserver hoch- oder vom Replikationsserver heruntergeladenen Dateien
app.type	dss	Installationsart
app.db.pass	verschlüsselt	Passwort zur Anmeldung an der Datenbank
app.db.user	verschlüsselt	User zur Anmeldung an der Datenbank
app.db.path	verschlüsselt	Pfad zur Datenbank
log4j.conf.file	log4j-debug.xml	Aktivierung des Debug-Logging für den edoo ^{SVS} -Server <div style="border: 1px solid black; padding: 5px; background-color: #ffffcc;">  <p>Hinweis: Eine Aktivierung des Debug-Logging kann die Erzeugung sehr großer Log-Dateien nach sich ziehen und sollte nur im Fehlerfall erfolgen.</p> </div>
app.login.type	ldap	Aktivierung des Logins über einen LDAP-Server
app.ldap.dn	cn={USER},ou=users,ou=system	DN zur Beschränkung der Login-fähigen User auf dem LDAP-Server
app.ldap.providerUrl	ldap://localhost:10389/	URL zum LDAP-Server, über den der Login erfolgen soll
app.proxy.host	z.B. proxy.de	Hostname des Proxyservers

app.proxy.port	z.B. 3128	Port des Proxy
app.proxy.type	z.B. http	Typ des Proxy

5.3.2 Konfigurationsdatei edoosys-server.cfg

Die Konfigurationsdatei „edoosys-server.cfg“ befindet sich direkt im Installationsverzeichnis des edoo^{SVS}-Servers. Sie enthält die Java-Parameter, mit denen der edoo^{SVS}-Server gestartet wird.

Eine Änderung der Einstellungen bedingt einen Neustart des Servers (siehe Kapitel 7.1.2).



Hinweis:

Alle in der nachfolgenden Auflistung nicht genannten Parameter der „config.ini“-Datei sollten nur nach vorheriger Rücksprache mit dem technischen Support verändert werden.

Eine falsche Änderung kann zu massiven Fehlern in der Anwendung bis hin zum Systemausfall führen.

Parameter	Wert(e)	Beispiel	Bedeutung
Xmx	1024m	-Xmx1024m	Maximal von Java für die edoo ^{SVS} -Anwendung genutzter Hauptspeicher (hier 1024 MB)
Xms	128m	-Xms128m	Minimal von Java für die edoo ^{SVS} -Anwendung genutzter Hauptspeicher (hier 128 MB)
XX:PermSize	128m	-XX:PermSize=128m	Initial von Java für die edoo ^{SVS} -Anwendung reservierter Hauptspeicher (hier 128 MB)

6 Überwachung, Monitoring, Protokolle

Im Rahmen des Betriebs der **edoo^{SVS}**-Anwendung ist es empfehlenswert, die einzelnen Komponenten nicht nur zu betreiben und im Fehlerfall entsprechend zu reagieren, sondern proaktiv kritische Parameter zu überwachen, um potenzielle Ausfälle zu vermeiden. Hierfür können bereits etablierte Monitoring-Systeme eingesetzt werden. Die zu überwachenden Parameter werden in den nachfolgenden Kapiteln beschrieben.

Darüber hinaus ist es im Fehlerfall wichtig, die einzelnen Protokolldateien der Anwendung zu kennen und auswerten zu können. Bekannte Fehlermeldungen und deren Behebung werden im Kapitel 8 beschrieben.

6.1 Überwachung der Hardware

Zu einer vollständigen Überwachung eines IT-Betriebs und so auch des Betriebs der **edoo^{SVS}**-Software, gehört immer auch die Überwachung der Hardware, auf der die Software betrieben wird. Hierbei sollte die Überwachung nicht auf eine bloße „still alive“-Überwachung beschränkt, sondern auch Parameter, die die Systemperformance beeinflussen können, darin aufgenommen werden:

- Up/Down Host
- CPU
 - Load
- RAM
 - Load
 - SWAP-Größe
 - Seitenfehler
 -
- Datenträger Performance
 - I/O
 - Queue/ Warteschlange
- Netzwerk
 - Bandbreite
 - Latenz zwischen Client und Server
 - Latenz zwischen Server und Datenbank
 - Latenz zwischen Server und zentralen Diensten

6.2 Überwachung des Betriebssystems

Auf der Ebene des Betriebssystems kann bereits eine Überwachung eingerichtet werden, die Aufschluss darüber gibt, ob die **edoo^{SVS}**-Anwendung und die zugehörigen Komponenten laufen und ob diese performant ausgeführt werden:

- Netzwerkstatus der Ports
 - 5432 (PostgreSQL-Datenbank)
 - 8765 (http-Port bzw. https-Port des **edoo^{SVS}**-Servers, siehe 5.3.1)
- Prozesse/ Dienste
 - PostgreSQL-Datenbank
 - **edoo^{SVS}**-Server

6.3 Überwachung der PostgreSQL-Datenbank

Die permanente Überwachung der PostgreSQL-Datenbank setzt sich zusammen aus der Überwachung auf Betriebssystemebene (siehe 6.2), die die Datenbank „von außen“ überprüft, wobei die „still alive“-Überwachung im Vordergrund steht, und der Überwachung der Datenbank selbst, die es ermöglicht, die Performance der Datenbank zu prüfen und ggf. zu optimieren.

In PostgreSQL werden diese statistischen Daten, die sich zur Überwachung der Datenbankperformance eignen in Systemtabellen protokolliert. Die Detaillierung lässt sich über die Parameter

- track_activities
- track_counts
- track_io_timing
- track_activity_query_size
- log_parser_stats
- log_planner_stats
- log_executor_stats
- log_statement_stats

der postgresql.conf steuern (siehe Kapitel 5.2.1).

6.3.1 Monitoring über Systemtabellen der PostgreSQL-Datenbank

Die Performance der PostgreSQL-Datenbank lässt sich gut anhand des Loggings in einige Systemtabellen bemessen. Diese können über SQL-Abfragen ausgewertet werden und sind somit sowohl mit etablierten Monitoring-Systemen als auch über einfache Shell-Skripte, die Protokolldateien erzeugen, nutzbar. Hierfür werden in PostgreSQL einige hilfreiche Views vorab definiert.

6.3.1.1 *pg_stat_activity*

Der View `pg_stat_activity` eignet sich, um sich die aktuellen Verbindungen gegen die Datenbank ausgeben zu lassen. Neben einigen Verbindungsdaten wie z.B. der Prozessnummer, die die Verbindung innehat, werden dabei auch Daten zur aktuellen Query mitgeführt (`current_query`, `query_start`), die es erlauben lang andauernde Datenbankabfragen zu erkennen.

6.3.1.2 *pg_stat_all_tables*

Der View `pg_stat_all_tables` gibt Aufschluss über die Zugriffe auf alle Tabellen der PostgreSQL-Datenbank. Es lässt sich darüber je Tabelle auswerten, wie häufig die Tabelle sequenziell durchlaufen werden musste (`seq_scan`), wie häufig der Index gescannt wurde (`idx_scan`) und wie häufig welche Datensatzoperationen (Insert, Update, Delete) auf der Tabelle durchgeführt wurden (`n_tup_...`).

Aus der Auswertung dieser Werte kann beispielsweise hervorgehen, wie häufig eine Tabelle beansprucht wird oder ob eine Tabelle (bei häufigen sequenziellen Läufen) einen zusätzlichen Index benötigt.

6.3.1.3 *pg_statio_all_tables*

Im View `pg_statio_all_tables` ermöglicht eine Übersicht auf das I/O-Verhalten und insbesondere auf das Caching der gesamten Datenbank. Hierfür kann beispielweise geprüft werden, wie häufig gelesene Blöcke (`heap_blocks_read`) im Cache gefunden wurden (`heap_blocks_hit`). Ist dies selten der Fall, kann die Datenbank-Performance evtl. über eine Cache-Erweiterung verbessert werden.

6.3.2 Protokolldateien der PostgreSQL-Datenbank

Die PostgreSQL-Datenbank legt, sofern nicht in der `postgresql.conf` anders festgelegt, ihre Protokolle im Unterverzeichnis „`pg_log`“ des Datenverzeichnisses ab. Die Detaillierung der Protokolle sowie deren

Ablage und Benennung können über die postgresql.conf gesteuert werden (siehe Kapitel 5.2.1). Dabei unterscheidet PostgreSQL in mehreren Detaillierungsstufen (debug5 bis panic).



Hinweis:

Eine Anpassung des Log Levels, also der Detaillierung, in der die Protokolle erzeugt werden, kann massive Auswirkungen auf den benötigten Speicherplatz für die Ablage der Protokoll-dateien haben. Es wird empfohlen, das Debug Logging nur im Falle einer konkreten Fehler-suche zu aktivieren.

6.4 Überwachung der edoo^{SVS}-Anwendung

Die edoo^{SVS}-Anwendung lässt sich über das Java Runtime Environment (JRE), in dem sie ausgeführt wird überwachen. Das JRE wird per default mit aktiviertem Monitoring ausgeführt. Die entsprechende API, Java Management Extensions (JMX), kann abgefragt werden, um eine Überwachung zu imple-mentieren.

6.4.1 Protokolldateien der edoo^{SVS}-Software

Die edoo^{SVS}-Anwendung besitzt ein eigenes Logging System. Die Protokolldateien sind im Unterver-zeichnis „logs“ des edoo^{SVS}-Installationsverzeichnis zu finden.

Folgende Protokolldateien werden von der Anwendung generiert:

- edoosys.log

Das globale edoo^{SVS} Protokoll. In Abhängigkeit vom konfigurierten Detaillierungsgrad der Prot- kollierung (Loglevel) kann bis auf Trace-Ebene jede Aktion in der Anwendung protokolliert wer- den. Der Loglevel kann in der Datei log4j-server.xml für Server eingestellt werden (siehe Kapitel **Fehler! Verweisquelle konnte nicht gefunden werden.**).

Tipp:

Das edoosys-Log wird im Fehlerfall für die Analyse und Behebung des Fehlers benötigt, es sollte daher jederzeit zur Verfügung stehen.

- error.log

Diese Datei beinhaltet Fehlermeldungen und Stack Traces der edoo^{SVS}-Anwendung

- svp-logins.log

Die Protokolldatei umfasst nur die An- und Abmeldevorgänge am edoo^{SVS}-Server.

Des Weiteren befindet sich im Unterverzeichnis „logs“ das Verzeichnis „old“. Hier werden die alten Pro- tokolldateien komprimiert und gepackt abgelegt. Der Name und das Datum der Datei lassen den jewei- ligen Ursprung erkennen. Die Aufbewahrungsfrist für diese Dateien kann konfiguriert werden.

7 Regelmäßige Administrationstätigkeiten

7.1 Starten und Stoppen des Systems

Das Gesamtsystem wird „von der Datenbank zum Client“ initialisiert. D.h. zunächst wird die Datenbank des edoo^{SVS}-Servers gestartet. Anschließend zunächst der edoo^{SVS}-Server selbst und daraufhin die Clients.



Sollte eine der Komponenten nicht starten, so sollten die Protokolle (siehe Kapitel 6) und anschließend die Konfiguration (siehe Kapitel 5) der jeweiligen Komponente geprüft werden.

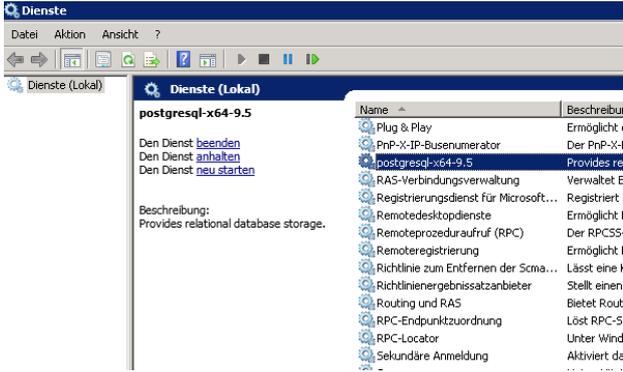
Soll das System gestoppt werden, bspw. im Rahmen von Wartungsarbeiten, so ist in umgekehrter Reihenfolge vorzugehen, d.h. zunächst sollten alle Clients gestoppt werden, anschließend der edoo^{SVS}-Server und schließlich die Datenbank heruntergefahren werden.

7.1.1 Starten und Stoppen der Datenbank

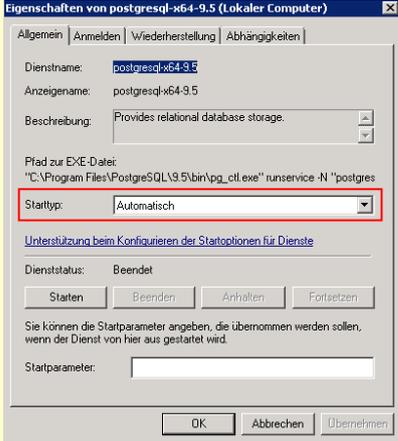
7.1.1.1 Starten und Stoppen der Datenbank auf Windows-Systemen

Wurde die Datenbank gemäß Installationshandbuch [R1] installiert, kann diese unter Windows-Systemen als Dienst gestartet und gestoppt werden.

<p>1. Über das Startmenü unter „Verwaltung“ den Punkt „Dienste“ auswählen.</p>	
--	--

<p>2. Aus der Liste einen Dienst „postgresql-...“ heraus-suchen</p>	
<p>3. Sofern sich der Dienst nicht im Zustand „Gestartet“ befindet, den Eintrag markieren und den Dienst über Klick auf  starten.</p>	

 **Hinweis:** Damit bei einem Neustart des Servers auch die PostgreSQL-Datenbank automatisch gestartet wird, muss der Starttyp des Dienstes auf „Automatisch“ konfiguriert sein. Ist dies nicht der Fall, kann es über die Eigenschaften des Dienstes geändert werden:



7.1.1.2 Starten und Stoppen der Datenbank auf Linux-Systemen

<p>1. Nach der Installation prüfen, ob es ein Init-Skript auf dem System gibt. Unter /etc/init.d muss eine Datei „postgresql-9.x“ liegen</p>	<pre>host:~ # ls /etc/init.d</pre>
<p>2. Prüfen, ob der gleichnamige Dienst läuft</p>	<pre>host:~ # service postgresql-9.5 status</pre>
<p>3. Sollte der Dienst nicht gestartet sein (Status ist ungleich „running“), den Dienst starten</p>	<pre>host:~ # service postgresql-9.5 start</pre>

7.1.2 Starten und Stoppen des edoo^{SVS}-Servers

Das Starten und Stoppen des edoo^{SVS}-Servers auf den unterschiedlichen Betriebssystem-Plattformen ist im Installationshandbuch Schulträger (DSS) [R1] detailliert beschrieben. Unter Windows kann der Server als Dienst gestartet und gestoppt werden.

Unter Linux erfolgt der Start über den Aufruf der Datei „edooSYS-server“ im Installationsverzeichnis des DSS. Gestoppt wird der Server über ein Beenden des entsprechenden Prozesses über das Betriebssystem.

7.2 Betriebssystem-Updates

Gerade im Microsoft Windows Umfeld ist es sehr zu empfehlen regelmäßig Patches und Sicherheitsupdates einzuspielen. Patches können Verbesserungen, Erweiterungen oder Fehlerbeseitigungen von Programmen enthalten. Hierbei sind natürlich die Release Notes der einzelnen Patches gründlich zu lesen. Aus **edoo^{SVS}** Sicht besonders relevant sind Patches, die die Java Laufzeitumgebung beeinflussen. In der Regel sind alle Patches vor dem Einspielen auf einem unabhängigen Testsystem zu verifizieren und nur bei Fehlerfreiheit auf den produktiven **edoo^{SVS}**-Systemen zu installieren.

Tipp:

Vor dem Einspielen von Updates und Patches, sollte die **edoo^{SVS}** Anwendung beendet werden (siehe Kapitel 7.1).

7.3 Update des Java Runtime Environment (JRE)

Die **edoo^{SVS}**-Anwendung läuft innerhalb eines Java Runtime Environment (JRE). Letzteres wird also essenziell in der richtigen Version benötigt, damit die Anwendung fehlerfrei und stabil laufen kann. Hierzu ist unter den Linux-Betriebssystemen ein regelmäßiges Update des JRE notwendig:

1.	Das aktuelle JRE-Installationspaket auf der Herstellerseite herunterladen.	http://www.oracle.com/technetwork/java/javase/downloads
2.	Das Paket gemäß der Herstellerinformationen installieren	
3.	Über Betriebssystemmechanismen sicherstellen, dass der Befehl „java“ auf die aktuelle Version zeigt.	host:~ #java --version
	Tipp: Dies unterscheidet sich von Distribution zu Distribution. Unter Suse Linux Enterprise empfiehlt sich ein „update-alternatives“. Unter anderen Distributionen kann eine Anpassung der PATH-Variable hilfreich sein.	



Hinweis:

Unter Windows darf das JRE **nicht** manuell aktualisiert werden. Hier wird die korrekte JRE-Version bei der Installation der **edoo^{SVS}**-Software mit installiert.

7.4 Updates der Datenbank und der Anwendung

In den Release-Zyklen der **edoo^{SVS}**-Anwendung, sollte eine Aktualisierung der Anwendung auf die aktuelle Version erfolgen (siehe Kapitel 4.2).



Hinweis:

Befindet sich die **edoo^{SVS}**-Anwendung auf einem anderen Versionsstand als die zentralen Dienste (ZSS), so kann i.d.R. nicht mit diesen kommuniziert werden. Eine Aktualisierung nach dem Release einer neuen Version ist also meist dringend erforderlich.

Eine regelmäßige Aktualisierung der Datenbankversion der PostgreSQL-Datenbank wird ebenfalls empfohlen. Die Aktualisierungszyklen sollten dabei den freigegebenen PostgreSQL-Versionen für die **edoo^{SVS}**-Software entsprechen. Die Aktualisierung erfolgt manuell (siehe Kapitel 4.2.3).

7.5 Benutzerverwaltung

Für die Basisadministration eines **edoo^{SVS}**-Systems kann die Benutzerverwaltung in zwei Kategorien eingeteilt werden. Auf der einen Seite betriebssystemspezifische Benutzer, auf der anderen Seite Benutzer in der **edoo^{SVS}**-Anwendung selbst. Spezielle Benutzer auf Betriebssystemseite sind in der Regel bei größeren DSS-Installationen zu empfehlen. Dieses ist natürlich auch in Abhängigkeit mit der vorhandenen Infrastruktur und den organisatorischen Rahmenbedingungen beim Betreiber zu sehen.

Für die Anwendung **edoo^{SVS}** selbst, kann man verschiedene Administratoren für die betrieblichen Aspekte von **edoo^{SVS}** unterscheiden. Diese Administratorenrollen müssen als organisatorische Rollen verstanden werden die für einen reibungslosen Betrieb von **edoo^{SVS}** erforderlich sind.

- **edoo^{SVS}**-Administrator

Der **edoo^{SVS}**-Administrator ist für die zentrale Verwaltung von **edoo^{SVS}** zuständig. Zu seinen Aufgaben gehört die Pflege von **edoo^{SVS}**, insbesondere die Erstellung und Pflege von Berichtsschablonen, Exportformatbeschreibungen, Zeugnisschablonen, Serienbriefen und Etiketten. Er verfügt zusätzlich über ein Plugin zum Veröffentlichen von diversen Anwendungsdaten.

- Systemadministrator

Der Systemadministrator ist für übergreifende Fragen, z.B. zur Installation und Themen, die den **edoo^{SVS}**-Server bzw. ZSS betreffen zuständig.

- Schuladministrator

Der Schuladministrator kümmert sich um die Administration einer Schule (beinhaltet z.B. die Durchführung der Datenübernahme aus den Altsystemen und die Verwaltung der Benutzer der zugehörigen Schule).

Informationen zur Erstanmeldung sind im Installationshandbuch des **edoo^{SVS}**-Servers [R1] beschrieben.

8 Bekannte Fehler und Probleme

8.1 Betriebssystem

8.2 PostgreSQL-Datenbank

8.2.1 edoo^{SVS}-Server kann sich nicht mit der PostgreSQL-Datenbank verbinden

Kann sich der edoo^{SVS}-Server nicht gegen die Datenbank verbinden, so ist i.d.R. die Netzwerkverbindung der beiden Komponenten gestört. In diesem Fall sollten folgende Punkte geprüft werden:

- **Netzwerkabel** (sind alle Kabel korrekt angeschlossen?)
- **Netzwerkverbindung** (haben Client und Server die korrekte Netzwerkkonfiguration, z.B. IP-Adressen?)
- **Erreichbarkeit des jeweils anderen Systems** (kann vom Client aus der Server „gepingt“ werden und umgekehrt?)
- **Komponenten** (laufen beide Komponenten, ist also insbesondere die PostgreSQL-Datenbank im Zustand „gestartet“?)
- **Firewall zwischen den Systemen und Systemfirewall** (ist der Port 8765, der für die Kommunikation verwendet wird, auf allen Firewalls freigegeben?)

In jedem Fall sollten zur Eingrenzung des Fehlers die Protokolle des edoo^{SVS}-Servers (siehe Kapitel 6.4.1) und der PostgreSQL-Datenbank (siehe Kapitel 6.3.2) eingesehen und auf Fehlermeldungen hin geprüft werden.

8.3 edoo^{SVS}-Anwendung

8.3.1 edoo^{SVS}-Server startet nicht

Startet der edoo^{SVS}-Server nicht, so sollten zunächst die Log-Dateien (siehe Kapitel 6.4.1) geprüft werden, diese geben i.d.R. Aufschluss über den Grund. Häufige Fehlerursachen sind:

- **JRE:** Der Server verwendet zum Starten ein falsches Java Runtime Environment (JRE). Dieses muss ggf. aktualisiert oder angepasst werden.
- **Konfigurationsdateien verändert:** Der Server hat Manipulationen an den Konfigurationsdateien festgestellt. Diese müssen rückgängig gemacht werden.
- **Nicht freigegebene Datenbankversion:** Der Server hat eine Datenbankversion vorgefunden, die nicht den vorgegebenen Versionen entspricht. Diese muss durch eine freigegebene Version ersetzt werden.

Zur Behebung des Problems sind die Fehlerursache zu beseitigen und anschließend ein erneuter Start des Servers zu versuchen.

8.3.2 edoo^{SVS}-Client kann sich nicht zum edoo^{SVS}-Server verbinden

Kann sich der edoo^{SVS}-Client nicht gegen den passenden Server verbinden, so ist i.d.R. die Netzwerkverbindung der beiden Komponenten gestört. In diesem Fall sollten folgende Punkte geprüft werden:

- **Netzwerkabel** (sind alle Kabel korrekt angeschlossen?)
- **Netzwerkverbindung** (haben Client und Server die korrekte Netzwerkkonfiguration, z.B. IP-Adressen?)
- **Erreichbarkeit des jeweils anderen Systems** (kann vom Client aus der Server „gepingt“ werden und umgekehrt?)

- **Komponenten** (laufen beide Komponenten, ist also insbesondere der edoo^{SVS}-Server im Zustand „gestartet“?)
- **Firewall zwischen den Systemen und Systemfirewall** (ist der Port 8765, der für die Kommunikation verwendet wird, auf allen Firewalls freigegeben?)

In jedem Fall sollten zur Eingrenzung des Fehlers die Protokolle des edoo^{SVS}-Clients und des edoo^{SVS}-Servers (siehe Kapitel 6.4.1) eingesehen und auf Fehlermeldungen hin geprüft werden. Die Protokolle des edoo^{SVS}-Clients befinden sich im Unterverzeichnis „log“ seines Installationsverzeichnis auf dem Client-Rechner.

8.3.3 edoo^{SVS}-Server kann sich nicht mit ZSS verbinden

Kann sich der edoo^{SVS}-Server nicht gegen den zentralen Schulverwaltungsserver (ZSS) verbinden, so ist i.d.R. die Netzwerkverbindung der beiden Komponenten gestört. In diesem Fall sollten folgende Punkte geprüft werden:

- **Netzwerkabel** (sind alle Kabel korrekt angeschlossen?)
- **Netzwerkverbindung** (ist die Netzwerkconfiguration, z.B. IP-Adressen, des edoo^{SVS}-Servers korrekt?)
- **Erreichbarkeit des jeweils anderen Systems** (kann vom edoo^{SVS}-Server aus der ZSS „gepingt“ werden bzw. ist dessen URL erreichbar?)
- **Komponenten** (laufen beide Komponenten, ist also insbesondere der edoo^{SVS}-Server im Zustand „gestartet“?)
- **Firewall zwischen den Systemen und Systemfirewall** (ist der Port, der für die Kommunikation verwendet wird, auf allen Firewalls freigegeben?)

In jedem Fall sollten zur Eingrenzung des Fehlers die Protokolle des edoo^{SVS}-Servers (siehe Kapitel 6.4.1) eingesehen und auf Fehlermeldungen hin geprüft werden.

8.3.4 Replikation einer Schule des edoo^{SVS}-Servers schlägt fehl

Ein Fehlschlag der Replikation des edoo^{SVS}-Servers äußert sich i.d.R. in folgender Fehlermeldung:



Tritt diese bei einem Replikationsversuch auf, ist zunächst zu prüfen, ob der betreffende Server direkt mit dem zentralen Replikationsserver oder aber mit einem kaskadierten Replikationsserver verbunden wird. Hierfür kann die URL des Replikationsservers aus den Konfigurationsdateien „config.ini“ bzw. „config.local.ini“ verwendet werden (siehe Kapitel 5.3.1).

In beiden Fällen sollten folgende Punkte geprüft werden:

- **Netzwerkabel** (sind alle Kabel korrekt angeschlossen?)
- **Netzwerkverbindung** (haben Client und Server die korrekte Netzwerkconfiguration, z.B. IP-Adressen?)
- **Erreichbarkeit des jeweils anderen Systems** (kann vom edoo^{SVS}-Server aus der Replikationsserver „gepingt“ werden?)
- **Firewall zwischen den Systemen und Systemfirewall** (ist der Port 8765, der für die Kommunikation verwendet wird, auf allen Firewalls freigegeben?)

Kann der Fehler reproduziert, jedoch die Ursache nicht festgestellt werden, so ist der Support der Anwendung zu kontaktieren.

9 Sicherung und Wiederherstellung

In den folgenden Kapiteln werden zum einen die Ausfallrisiken des **edoo^{SVS}**-Servers und der zugehörigen Datenbestände und die jeweils verfolgte Sicherungs- und Wiederherstellungsstrategie erläutert (siehe Kapitel 9.1). Zum anderen werden die einzelnen technischen Mechanismen und ihre Einrichtung beschrieben, die im Rahmen der Strategien zum Einsatz kommen (siehe Kapitel 9.2).

9.1 Sicherungs- und Wiederherstellungsstrategie

9.1.1 Systemausfall

Bei einem Ausfall des gesamten Systems werden die einzelnen Komponenten Betriebssystem, Datenbank und **edoo^{SVS}**-Server nach dem Installationshandbuch neu installiert.



Hinweis:

Der neu installierte **edoo^{SVS}**-Server muss in seiner Version dem **edoo^{SVS}**-Server entsprechen, unter dem das einzuspielende Backup der Datenbank erstellt wurde.

Wurde also seit dem letzten Backup ein Update des **edoo^{SVS}**-Servers gemacht, so muss die „alte“ Version installiert werden, damit die Version des Servers zum Stand der Datenbank passt.

Um eine identische Konfiguration des Systems gewährleisten zu können, wird bei der Installation die Installationsdokumentation berücksichtigt, die während der ursprünglichen Installation erstellt wurde (siehe [R1]).

Anschließend wird eine Vollsicherung der Datenbank in das System eingespielt (siehe Kapitel 0). Wurde eine veraltete Version des **edoo^{SVS}**-Servers verwendet (siehe Hinweisbox), so wird danach eine Aktualisierung des Servers vorgenommen (siehe Kapitel 4.2.1).

Anschließend kann bei Bedarf und bei getätigter Replikationssicherung, eine Rücksicherung dieser erfolgen (siehe Kapitel 9.2.1.2).

9.1.2 Ausfall der Datenbank

Bei einem Ausfall der Datenbank wird analog zum Systemausfall (siehe Kapitel 9.1.1) vorgegangen. Auf eine Neuinstallation des **edoo^{SVS}**-Servers kann dabei verzichtet werden, sofern sich die Einstellungen zur Verbindung mit der Datenbank durch deren Neuinstallation nicht ändern (Datenbank-Host, -Port, -User, -Passwort, -Name) und die Version des **edoo^{SVS}**-Servers zum Stand der Datenbank passt.

9.1.3 Korrupte Daten

Durch Benutzer- oder Anwendungsfehler kann es zu korrupten Datenbeständen kommen, bspw. durch falsche Eingaben bei der Erfassung von Daten. In diesem Fall wird der letzte fehlerfreie Stand einer Datensicherung wieder in das laufende System eingespielt. Dabei werden je nach Ausmaß des Fehlers entweder nur die Datenbestände einer einzelnen Schule oder aber des gesamten **edoo^{SVS}**-Servers auf einen alten Stand zurückgesetzt.

Im ersten Fall kann der Stand auf dem Replikationsserver (siehe Kapitel 9.2.1.2) zur Wiederherstellung verwendet werden.

Muss der Datenbestand des gesamten Servers wiederhergestellt werden, so sollte eine Vollsicherung der Datenbank eingespielt werden (siehe Kapitel 9.2.2.1).

9.2 Sicherungs- und Wiederherstellungsarten

9.2.1 Replikation

Die Sicherung und Rücksicherung über den Replikationsmechanismus erfolgt je Schule. Dabei werden die Daten einer Schule an einen Replikationsserver übertragen und können von diesem bei Bedarf auf den jeweiligen DSS zurückgeholt werden. Die Sicherung erfolgt zeitgesteuert und kann durch die Schule selbst über den **edoo^{SVS}**-Client gesteuert werden. Voraussetzung ist lediglich, dass auf dem **edoo^{SVS}**-Server ein Replikationsserver konfiguriert ist (siehe Kapitel 5.3.1). Ebenso kann eine Rücksicherung der Daten durch die Schule über den **edoo^{SVS}**-Client initiiert werden.



Hinweis:

Die Replikation der Schuldaten bietet keine Wiederherstellungsmöglichkeit im eigentlichen Sinne. Für die Rücksicherung ist es unbedingte Voraussetzung, dass die Schule auf dem jeweiligen DSS bereits angelegt und aktiv ist. Somit kann bei einem Systemausfall oder einem Ausfall der Datenbank das System nicht ausschließlich über die Replikation wiederhergestellt werden. Hierzu werden weitere Sicherungsmechanismen benötigt.

Die Replikationssicherung muss in der Konfigurationsdatei config.local.ini aktiviert und konfiguriert werden. Dies geschieht über die Parameter (siehe Kapitel 5.3.1):

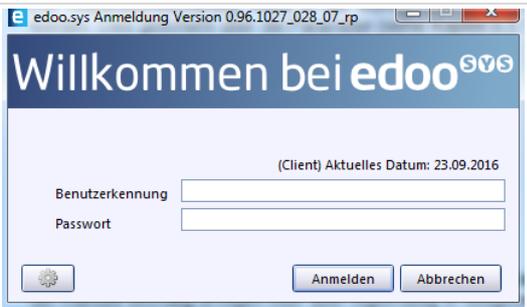
- remote.server.url
- remote.server.keystore
- remote.server.keystore.password
- remote.server.keystore.type
- replication.server.url
- replication.server.cleanup

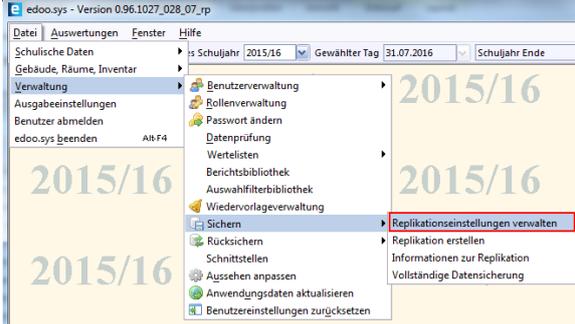
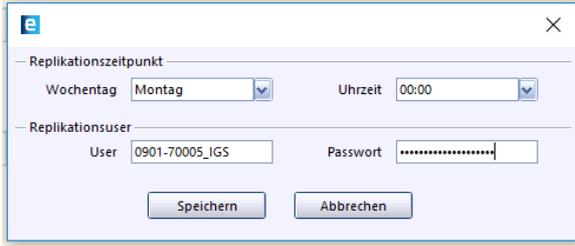
9.2.1.1 Erstellen einer Replikations-Sicherung

Das Erstellen einer Replikations-Sicherung wird über den **edoo^{SVS}**-Client konfiguriert. Es kann entweder manuell einmalig erfolgen oder aber regelmäßig in festgelegten Zyklen.

9.2.1.1.1 Automatische Replikationssicherung (zyklisch)

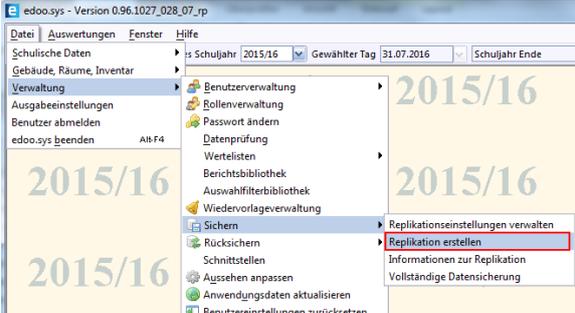
Um eine regelmäßige Replikation einzurichten, muss zunächst eine Anmeldung im **edoo^{SVS}**-Client erfolgen. Der angemeldete Benutzer braucht mindestens die Rechte eines Systemadministrators..

<p>1. Den edoo^{SVS}-Client über das Startmenü > Alle Programme > edoosys > edoosys Client starten.</p>	
<p>2. Am edoo^{SVS}-Client anmelden. Hierzu Benutzer und Passwort eingeben und auf „Anmelden“ klicken</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;">  <p>Hinweis: Der Benutzer benötigt für die Einrichtung der Replikationssicherung mindestens die Rechte eines Schuladministrators (siehe Kapitel 7.5).</p> </div>	

<p>3. Die Replikationseinstellungen über „Datei“ > „Verwaltung“ > „Sicherung“ > „Replikationseinstellungen verwalten“ öffnen</p>	
<p>4. Im sich öffnenden Einstellungsfenster die Optionen für die Replikation setzen. Es ist empfohlen täglich eine Replikationssicherung zu erstellen und diese nachts, außerhalb der Benutzeraktivitäten durchzuführen. Die Anmeldedaten am Replikationsserver entsprechen denen, die zuvor am ZSS für die Replikation erstellt wurden.</p>	
<p>5. Das Fenster mit „Speichern“ schließen.</p>	
<p>6. [optional] Zum Testen der Verbindung eine einmalige Replikations-Sicherung erstellen (siehe Kapitel 9.2.1.1.2).</p>	

9.2.1.1.2 Manuelle Replikations-Sicherung (einmalig)

Um eine einmalige Replikation durchzuführen, muss zunächst eine Anmeldung im edoo^{SVS}-Client erfolgen. Der angemeldete Benutzer braucht mindestens die Rechte eines Systemadministrators.

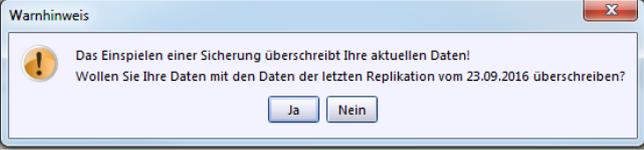
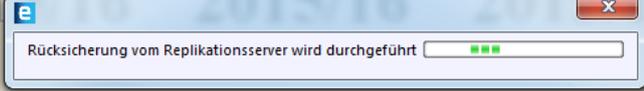
<p>1. Den edoo^{SVS}-Client über das Startmenü > Alle Programme > edoosys > edoosys Client starten.</p>	
<p>2. Am edoo^{SVS}-Client anmelden. Hierzu Benutzer und Passwort eingeben und auf „Anmelden“ klicken</p> <div data-bbox="255 1444 774 1624" style="border: 1px solid black; padding: 5px;"> <p>Hinweis: Der Benutzer benötigt für die Einrichtung der Replikationssicherung mindestens die Rechte eines Schuladministrators (siehe Kapitel 7.5).</p> </div>	
<p>3. Die manuelle Replikation über „Datei“ > „Verwaltung“ > „Sicherung“ > „Replikation erstellen“ starten.</p> <div data-bbox="255 1736 774 1960" style="border: 1px solid black; padding: 5px;"> <p>Hinweis: Sollte an dieser Stelle die Warnmeldung „Der Replikationsserver ist vorübergehend nicht erreichbar“ auftreten, so wird keine Sicherung erstellt. Diese muss dann zu einem späteren Zeitpunkt wiederholt werden (siehe Kapitel 8.3.4).</p> </div>	

<p>4. Aus der geöffneten Liste die zu replizierende Schule auswählen.</p>	
<p>5. Die Replikation wird gestartet.</p>	
<p>6. Nach erfolgter Replikation die Erfolgsmeldung mit „OK“ bestätigen.</p>	

9.2.1.2 Wiederherstellung der Schuldaten aus einer Replikations-Sicherung

Um eine Replikation auf den DSS rückzusichern, muss zunächst eine Anmeldung im edoo^{SVS}-Client erfolgen. Der angemeldete Benutzer braucht mindestens die Rechte eines Systemadministrators.

<p>1. Den edoo^{SVS}-Client über das Startmenü > Alle Programme > edoosys > edoosys Client starten.</p>	
<p>2. Am edoo^{SVS}-Client anmelden. Hierzu Benutzer und Passwort eingeben und auf „Anmelden“ klicken</p> <div data-bbox="255 1478 710 1680" style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>Hinweis: Der Benutzer benötigt für die Einrichtung der Replikationssicherung mindestens die Rechte eines Schuladministrators (siehe Kapitel 7.5).</p> </div>	
<p>3. Die Rücksicherung der Replikation über „Datei“ > „Verwaltung“ > „Rücksichern“ > „Rücksicherung vom Replikationsserver“ starten.</p>	

<p>4. Aus der geöffneten Liste die rückzusichernde Schule auswählen.</p> <div data-bbox="256 389 716 562" style="border: 1px solid black; padding: 5px; margin-top: 10px;">  <p>Hinweis: Es können nur Schulen rückgesichert werden, die bereits repliziert wurden, d.h. deren Status bei „Replikation“ gleich „v.“ ist.</p> </div>	
<p>5. Bei der folgenden Warnmeldung insbesondere prüfen, ob das korrekte Datum für die rückzusichernde Replikation gewählt wurde. Dann die Warnmeldung mit „Ja“ bestätigen.</p>	
<p>6. Die Rücksicherung wird gestartet.</p>	
<p>7. Nach erfolgter Rücksicherung die Erfolgsmeldung mit „OK“ bestätigen.</p>	

9.2.2 Vollsicherung der Datenbank

Um die vom edoo^{SVS}-Server verwendete PostgreSQL-Datenbank zu sichern, kann auf deren interne Sicherungstools zurückgegriffen werden:

- pg_dump
- pg_dumpall
- pg_restore

Diese werden bei der Installation der Datenbank mitinstalliert und bringen alle Funktionen mit, die für ein Backup und eine Wiederherstellung der Datenbank benötigt werden.

Um eine regelmäßige Vollsicherung der Datenbank zu erstellen, kann außerdem auf die Betriebssystemmechanismen

- Aufgabenplanung (Windows-Systeme)
- Cron (Linux-Systeme)

zurückgegriffen werden.



Hinweis: Die Vollsicherung einer Datenbank dient nur zur Wiederherstellung eines DSS nach einem Systemausfall. Diese erfolgt nach dem Neuaufsetzen des System selbst immer in 2 Schritten:

1. Einspielen der letzten Vollsicherung der Datenbank (siehe Kap. 9.2.2.1)
2. Rücksichern des aktuellen Replikationsdatenbestandes (siehe Kap. 9.2.1.2)



Achtung: explizit ist es nach einer Aktivierung des Replikationsmechanismus (siehe Kap. 9.2.1.1) nicht mehr möglich eine „lokale Rücksicherung“ durchzuführen. Rücksicherung können dann nur noch mittels des Replikationsmechanismus erfolgen (siehe Kap. 9.2.1.2).

9.2.2.1 Erstellen einer Vollsicherung der Datenbank

Eine Vollsicherung der Datenbank wird insbesondere im Katastrophenfall, d.h. bei einem vollständigen Ausfall des edoo^{SVS}-Servers benötigt, um diesen wiederherzustellen. Darüber hinaus kann sie beispielsweise beim Umzug der Datenbank auf einen anderen Datenbank-Server eingesetzt werden.

9.2.2.1.1 Manuelle Vollsicherung der Datenbank (einmalig)

Die Vollsicherung der Datenbank erfolgt über das PostgreSQL-eigene Tool „pg_dumpall“, das mit der PostgreSQL-Datenbank mitgeliefert wird. Das Tool findet sich im Unterverzeichnis „bin“ des Installationsverzeichnis der PostgreSQL-Datenbank und wird über die Kommandozeile (Bash, Shell, Eingabeaufforderung) aufgerufen.

Das Tool bietet die Möglichkeit die vollständige Datenbank des edoo^{SVS}-Servers sowie alle angelegten Benutzer etc. zu sichern.

Tipp:

Um nur die eigentliche Datenbank des edoo^{SVS}-Servers zu sichern, kann stattdessen das Tool „pg_dump“ verwendet werden. Für eine schnelle Wiederherstellung oder Migration bietet es sich aber an, den gesamten Datenbank-Cluster zu sichern.

Um sich alle relevanten Parameter des Tools anzeigen zu lassen, kann das Tool mit der Hilfe-Option aufgerufen werden:

```
pg_dumpall --help
```

Um eine Standard-Vollsicherung durchzuführen, ist es ausreichend, das Tool mit den zu verwendenden Verbindungsparametern für die Datenbank aufzurufen. Darüber hinaus ist es sinnvoll, das Zielverzeichnis bzw. den Namen der Backup-Datei anzugeben:

```
pg_dumpall -U postgres -w -h localhost -p 5432 -f .\backups\full.bak
```

Da „pg_dumpall“ kein Passwort mitgegeben werden kann, muss dieses in einer Datei „pgpass.conf“ abgelegt werden, damit die Verbindung zur Datenbank automatisch funktioniert. Die Datei hat folgenden Inhalt:

```
hostname:port:database:username:password
```

Um sich auf dem PostgreSQL-Server mit allen Datenbanken (dies ist für „pg_dumpall“ notwendig) mit dem User „postgres“ und dem Passwort „admin123“ zu verbinden, wird also folgender Inhalt in die Datei eingefügt:

```
localhost:5432:*:postgres:admin123
```

Die pgpass-Datei wird unter einem Pfad des Benutzers abgelegt, unter dem die Sicherung mit „pg_dumpall“ ausgeführt wird, mit dem das Tool also aufgerufen wird, dies ist:

- Unter Windows: %APPDATA%\postgresql\pgpass.conf
- Unter Linux: ~/.pgpass

9.2.2.1.2 Regelmäßige Vollsicherung der Datenbank (zyklisch) unter Windows

Unter den Windows-Betriebssystemen kann die systemeigene „Aufgabenplanung“ eingesetzt werden, um eine regelmäßige Vollsicherung der Datenbank zu erstellen.

Hierfür sollte ein ausführbares Skript erstellt und auf dem Server abgelegt werden, das eine Sicherung über „pg_dumpall“ durchführt. Ein Beispiel-Skript findet sich in den Anlagen.

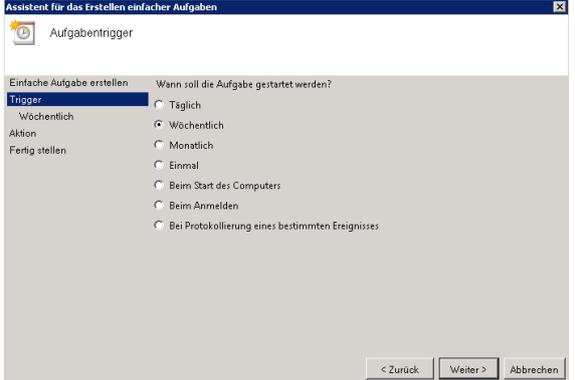
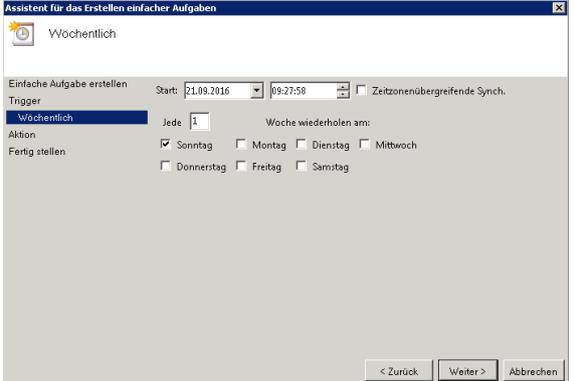
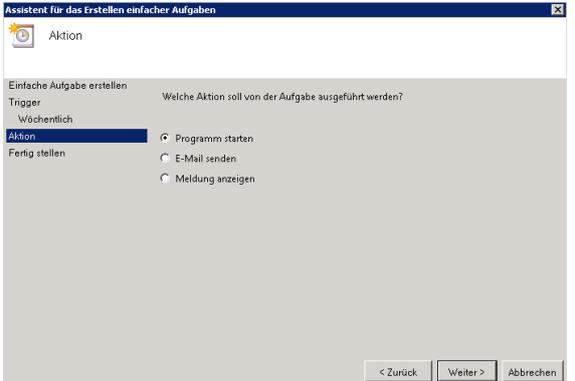
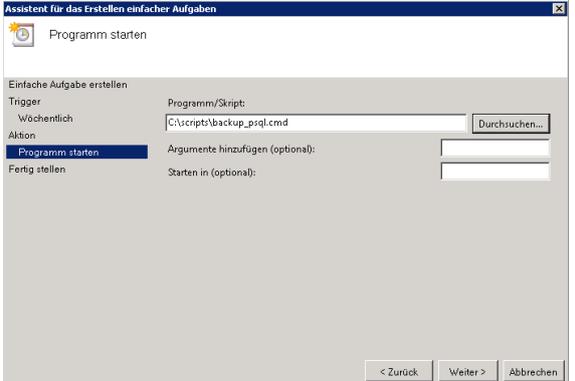


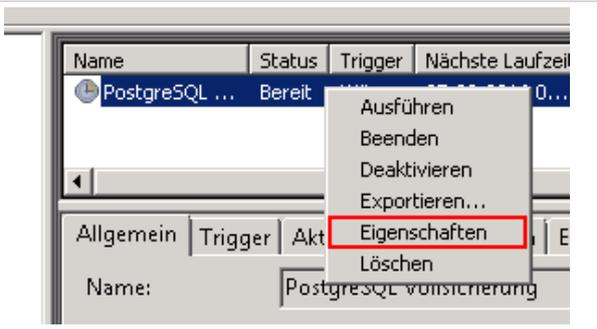
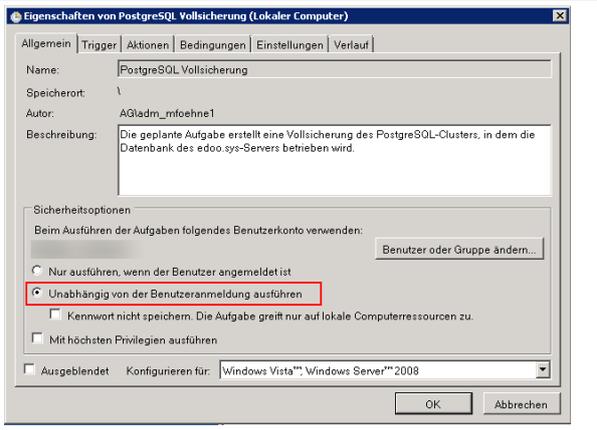
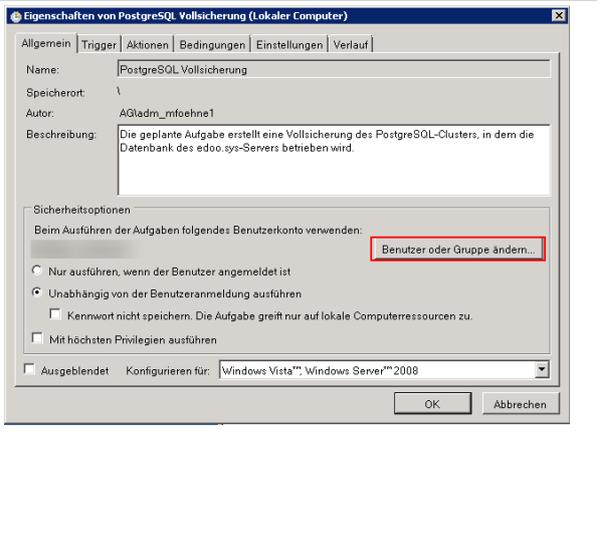
Hinweis:

Wie auch bei der manuellen Sicherung, benötigt der Benutzer, unter dem die Aufgabe, in der „pg_dumpall“ aufgerufen wird, ausgeführt wird, eine pgpass-Datei mit den Verbindungsdaten zur Datenbank (siehe Kapitel 9.2.2.1.1).

Um eine solche geplante Aufgabe anzulegen, muss zunächst die Windows-Aufgabenplanung geöffnet werden.

<p>1. Über das Startmenü „Alle Programme“ > „Zubehör“ > „Systemprogramme“ > „Aufgabenplanung“ ausführen.</p>	
<p>2. Den Eintrag „Aufgabenplanungsbibliothek“ mit Klick markieren. Anschließend mit Rechtsklick das Kontextmenü öffnen. Hier den Eintrag „Einfach Aufgabe erstellen...“ auswählen.</p>	
<p>3. Eine sprechende Bezeichnung und Beschreibung der Aufgabe eintragen und mit Klick auf „Weiter“ zum nächsten Schritt wechseln.</p>	

<p>4.</p>	<p>Die Häufigkeit der Ausführung auswählen und mit Klick auf „Weiter“ zum nächsten Schritt wechseln.</p> <div style="border: 1px solid black; background-color: #ffffcc; padding: 5px; margin-top: 10px;"> <p>Tipp: In der Regel ist eine wöchentliche Vollsicherung des DB-Clusters ausreichend, sofern zusätzlich auf weitere Sicherungs-Mechanismen wie bspw. die Replikation zurückgegriffen wird.</p> </div>	
<p>5.</p>	<p>Den Zeitpunkt der regelmäßigen Sicherung festlegen und mit Klick auf „Weiter“ zum nächsten Schritt wechseln.</p> <div style="border: 1px solid black; background-color: #ffffcc; padding: 5px; margin-top: 10px;"> <p>Tipp: Für die Sicherung werden Zeitpunkte empfohlen, an denen das System wenig verwendet wird, bspw. der Sonntag.</p> </div>	
<p>6.</p>	<p>Als Aktion „Programm starten“ auswählen und mit Klick auf „Weiter“ zum nächsten Schritt wechseln.</p>	
<p>7.</p>	<p>Unter „Programm/Skript“ über den Button „Durchsuchen“ das abgelegte Skript auswählen, das eine Vollsicherung der Datenbank erstellt. Mit Klick auf „Weiter“ und auf „Fertig stellen“ den Assistenten beenden.</p>	

8.	Die Aufgabe wird nun in der „Aufgabenplanungsbibliothek“ angezeigt. Mit Rechtsklick auf die Aufgabe die Eigenschaften öffnen.	
9.	Im Reiter „Allgemein“ unter „Sicherheitsoptionen“ die Option „Unabhängig von der Benutzeranmeldung ausführen“ auswählen.	
10.	<p>[optional] Den Benutzer, unter dem die Aufgabe ausgeführt werden soll über den Button „Benutzer oder Gruppe ändern...“ anpassen.</p> <div data-bbox="264 1189 778 1305" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>Tipp: Es wird empfohlen, einen eigenen Benutzer ausschließlich für die Sicherung zu verwenden.</p> </div> <div data-bbox="264 1350 778 1585" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>Hinweis: Wie auch bei der manuellen Sicherung, benötigt der Benutzer, unter dem die Aufgabe, in der „pg_dumpall“ aufgerufen wird, ausgeführt wird, eine pgpass-Datei mit den Verbindungsdaten zur Datenbank (siehe Kapitel 9.2.2.1.1).</p> </div>	
11.	Die Eigenschaften mit Klick auf „OK“ speichern.	

9.2.2.1.3 Regelmäßige Vollsicherung der Datenbank (zyklisch) unter Linux

Unter den Linux-Betriebssystemen kann das cron-System verwendet werden, um eine regelmäßige Vollsicherung der Datenbank zu erstellen.

Hierfür sollte ein ausführbares Skript erstellt und auf dem Server abgelegt werden, das eine Sicherung über „pg_dumpall“ durchführt. Ein Beispiel-Skript findet sich in den Anlagen.



Hinweis:

Wie auch bei der manuellen Sicherung, benötigt der Benutzer, unter dem die Aufgabe, in der „pg_dumpall“ aufgerufen wird, ausgeführt wird, eine pgpass-Datei mit den Verbindungsdaten zur Datenbank (siehe Kapitel 9.2.2.1.1).

Um einen cron-Job anzulegen muss eine Datei im Verzeichnis „/etc/cron.d“ angelegt werden. Diese sollte sprechend benannt werden, bspw. „edoo_full_backup“.

Die Datei hat folgenden Inhalt (ohne Anführungszeichen):

```
„Minute“ „Stunde“ „Tag im Monat“ „Monat“ „Tag der Woche“ „auszuführendes Kommando“
```

Um also immer sonntags um 23:00 das Skript aufzurufen, welches bspw. unter „~/full_backup.sh“ abgelegt wurde, wird folgender Inhalt in die Datei geschrieben:

```
00 23 * * 7 ~/full_backup.sh >/dev/null 2>&1
```

9.2.2.2 Wiederherstellung einer Vollsicherung der Datenbank

Die Wiederherstellung einer Vollsicherung der Datenbank, die über das PostgreSQL-eigene Tool „pg_dumpall“ erstellt wurde, kann mit „pg_restore“ wiederhergestellt werden. Das Tool findet sich im Unterverzeichnis „bin“ des Installationsverzeichnisses der PostgreSQL-Datenbank und wird über die Kommandozeile (Bash, Shell, Eingabeaufforderung) aufgerufen.

Um sich alle relevanten Parameter des Tools anzeigen zu lassen, kann das Tool mit der Hilfe-Option aufgerufen werden:

```
pg_restore --help
```

Um ein Backup in einen vorhandenen PostgreSQL-Datenbankcluster zurückzuspielen müssen die Verbindungsoptionen zum Cluster sowie die Dump-Datei, die wiederhergestellt werden soll, als Option angegeben werden.

```
pg_restore -h localhost -p 5432 -U postgres -w .\backup\full.bak
```



Hinweis:

Vor der Durchführung einer Wiederherstellung der Datenbank muss der DSS, sofern vorhanden in jedem Fall gestoppt werden (siehe Kapitel 7.1.2).



Achtung:

Wurde für diese Schule vor der Wiederherstellung der Replikationsmechanismus aktiviert, d.h. über diesen eine Replikationssicherung durchgeführt, so **muss** nach der oben beschriebenen Wiederherstellung der Datenbank **als zweiter Schritt noch die Rücksicherung der aktuellsten Replikationssicherung vorgenommen werden** (siehe Kap. 9.2.1.2).

Hinweis:

Erfolgt dieser Schritt nicht, so kann für diesen Datenbestand der Schule der Replikationsmechanismus nicht benutzt werden!

10 Dokumentenverweise

Name	Dokument / Link
[R1] Installationshandbuch DSS	
[R2] Anwenderhandbuch	edoosys_Anwenderhandbuch.doc

Weitere Informationen zur Schulverwaltungssoftware **edoo^{SVS}** finden Sie auch unter:

<http://www.edoosys.de/>

11 Anlagen

Dokumentation dbctl.cmd

Zur schulscharfen Sicherung von Daten steht zusätzlich das Kommandozeilenprogramm „dbctl.cmd“ zur Verfügung. Dieses befindet sich im Ordner „bin“ des DSS-Installationsverzeichnisses und kann über die Kommandozeile aufgerufen werden. Grundsätzlich dient das Programm zum Sichern, Löschen und Wiederherstellen der SVP-Mandanten.

Genereller Aufruf des Programmes:

<Pfad zum DSS-Installationsverzeichnis>bin\dbctl [options] [commands] [command options]

Folgende Parameter können hierbei verwendet werden:

Kategorie	Kurzform	Langform	Beschreibung	Beispiel
options	-h	--help, ?	Zeigt die Hilfe an	... bin\dbctl -h
options	-u	--user	Kennung des Systemadministrators. Default: sys	... bin\dbctl -u sys
options	-pi	--password-input	Passwort des Systemadministrators, muss über die Konsole eingegeben werden	... bin\dbctl -pi
options	-p	--password	Passwort des Systemadministrators als Kommandozeilenparameter	... bin\dbctl -p [passwort]
options	-b	--basedir	Das basisverzeichnis für Sicherungen. Default: .\svp_data	... bin\dbctl -b <Pfad>
options	-d	--dbconfig	Konfigurationsdatei in der die Datenbankverbindungsdaten abgelegt werden. Default: .\dbctl\dbsettings.ini	... bin\dbctl -d <Pfad>
options	-l	--log	Detail der Logausgabe (0-3). Default: 0	... bin\dbctl -l 2
command	s	store, backup	Sicherungsmodus. Die Daten der ausgewählten Schule(n) werden ins Basisverzeichnis gesichert. Pro Schule wird eine Sicherungsdatei erzeugt	... store [command options]
command options	-i	--include	Schulnummern der Mandanten die gesichert werden sollen. . kann dabei als Wildcard verwendet werden. Default: [.]	... store -i 6186 0064 04.
command options	-e	--exclude	Schulnummern der Mandanten die nicht gesichert werden sollen. . kann dabei als Wildcard verwendet werden	... store -i 04. -e 045.
command options	-if	--include file	Datei mit Schulnummern der Mandanten die gesichert werden sollen. Die Schulnummern müssen durch Leerzeichen oder Kommas getrennt werden oder es muss jede Schulnummer in einer neuen Zeile stehen.	... store -if <Pfad zur Datei>
command options	-ef	--exclude file	Datei mit Schulnummern der Mandanten die nicht gesichert werden sollen. Die Schulnummern müssen durch Leerzeichen oder Kommas getrennt werden oder es muss jede Schulnummer in einer neuen Zeile stehen.	... store -ef <Pfad zur Datei>

command	r	restore	Rücksicherungsmodus. Die Daten der ausgewählten Schule werden aus dem Basisverzeichnis zurückgesichert. Es kann immer nur eine Schule zurückgesichert werden.	... restore [command options] Achtung! Die Option -af muss immer mit angegeben werden.
command options	-af	--archiv-file	Die Sicherungsdatei, die zurückgesichert werden soll. Die Datei muss im Basisverzeichnis liegen.	... restore -af 20121221_6186_backup.sba
command options	-f	--force-remove	Achtung! Der angegebene Mandant wird im Falle von Fehlern teilweise gelöscht.	... restore -af 20121221_6186_backup.sba -f
command	d	delete, remove	Löschmodus. Die Daten der ausgewählten Schule(n) werden gelöscht. Bevor die Daten gelöscht werden, können sie in eine Sicherungsdatei gespeichert werden.	... delete [command options] Achtung! Der Parameter -i muss immer mit angegeben werden.
command options	-i	--include	Schulnummer(n) des Mandanten der gelöscht werden soll. Wildcards sind nicht möglich, es kann immer nur ein Mandant gelöscht werden.	... delete -i 6186
command options	-f	--force-remove	Achtung! Der angegebene Mandant wird im Falle von Fehlern teilweise gelöscht.	... delete -i 6186 -f
command options	-t	--test-remove	Die Daten des angegebenen Mandanten werden in einer Transaktion gelöscht. Die Transaktion wird anschließend zurück gerollt, so dass keine Daten gelöscht werden.	... delete -i 6186 -t
command options	-s	--slice	Es wird nur noch die angegebene Zeitscheibe des Mandanten gelöscht, nichtmehr der gesamte Mandant	... delete -i 6186 -s 2014
command	e	encrypt	Verschlüsselungsmodus. Die angegebene Datei wird verschlüsselt und kann nichtmehr entschlüsselt werden. Die Originaldatei wird umbenannt und mit der Endung .bak versehen	... encrypt [command options]
command options	-f	--file	Datei, die verschlüsselt werden soll.	... encrypt -f <Pfad zur Datei>

Tabelle 1: Übersicht Parameter dbctl.cmd

Alternativ können die Parameter auch in einer .ini-Datei abgelegt werden. Der Aufruf erfolgt dann in der Form: **<Pfad zum DSS-Installationsverzeichnis>bin\dbctl @/pfad/parameterdatei**. Wird nur @ ohne Pfad und Datei verwendet, so wird standardmäßig **.\dbctl\ctlconfig.ini** als Datei verwendet.

Der Ordner .dbctl muss hierfür vorher über die Kommandozeile angelegt werden (<Pfad der DSS-Instalation> md .dbctl). Es empfiehlt sich, hier die Dateien dbsettings.ini und ctlconfig.ini abzulegen.

Aufbau dbsettings.ini

In der Datei dbsettings.ini werden alle benötigten Datenbankverbindungsdaten eingetragen. Über den Parameter -d wird beim Aufruf des Programmes oder in der ctlconfig.ini-Datei der Pfad zur dbsettings.ini-Datei angegeben. Der prinzipielle Aufbau ist in nachfolgender Abbildung dargestellt. Die jeweiligen Parameter sind dabei den individuellen Gegebenheiten anzupassen:

```
# host IP oder DNS-Name
database.host=localhost
# PostgreSQL DB Port
database.port=5432
# PostgreSQL ASV Datenbank
database.name=asv
# DB User und Passwort
database.user=asv
database.password=asv
```

Aufbau ctlconfig.ini

In der Datei `ctlconfig.ini` können die Parameter für den Aufruf des Programmes gespeichert werden, so dass sie nicht jedesmal über die Kommandozeile eingegeben werden müssen. Liegt die Datei in dem zuvor beschriebenen Ordner `.dbctl`, so reicht zukünftig ein Aufruf über **<Pfad zum DSS-Installationsverzeichnis>bin\dbctl @**.

Ein beispielhafter Aufbau der Datei für eine Sicherung aller vorhandenen Mandanten ist in nachfolgender Abbildung dargestellt:

```
-u sys
-p !!EDOO!!
-d C:\Edoosys\.dbctl\dbsettings.ini
store -i .
```

Hierbei wird mit **-u** und **-p** der Systemadministrator authentifiziert, anschließend mit **-d** der Pfad zur `dbsettings.ini` angegeben und schließlich mit dem Kommando **store -i .** alle vorhandenen Mandanten gesichert.

Sofern das Basisverzeichnis nicht über den Parameter **-b** beim Aufruf geändert wird, befindet sich die Sicherung anschließend im Ordner **svp_data** im DSS-Installationsverzeichnis.

Anpassung `ctlconfig.ini`

Um den Mailversand für den beispielhaften Aufbau der Datei `ctlconfig.ini` aus Kapitel 6.2.2 einzurichten müssen lediglich die roten Zeilen ergänzt werden:

```
--log 2
-u sys
-p !!EDOO!!
-mh <Name / IP-Adresse des SMTP-Mailserver>
-mt <E-Mailadresse des Empfängers>
-d C:\Edoosys\.dbctl\dbsettings.ini
store -i .
```

Als Information in der Email erhalten Sie im Anschluss an die Sicherung Aussagen über die zur Sicherung ausgewählten Mandanten, die tatsächlich gesicherten Mandanten, den Status der Sicherung und die Dauer der Sicherung.

Mailversand

Zusätzlich zu den Funktionen Sichern, Löschen und Wiederherstellen der SVP-Mandanten bietet `dbctl.cmd` die Möglichkeit eines automatischen Emailversandes nach der Sicherung.

Hierzu stehen Ihnen die nachfolgenden Parameter zur Verfügung:

Kategorie	Kurzform	Langform	Beschreibung	Beispiel
options	-m	--mail	0 oder false: Es werden keine E-Mails versendet 1 oder true: Es werden E-Mails mit dem Status des Sicherungsvorganges versendet. Default: true	... bin\dbctl -m 0
options	-mh	--mail-host	Die IP-Adresse oder der qualifizierte Name (DNS-Name) des SMTP-Mailservers Default: localhost	... bin\dbctl -mh <Name>
options	-mp	--mail-port	Der Port des SMTP Mailservers Default: 25	... bin\dbctl -mp 25
options	-mu	--mail-user	Die optionale Kennung zur Anmeldung an den Mailserver	... bin\dbctl -mu <Kennung>
options	-ma	--mail-auth	Das optionale Passwort zur Anmeldung an den Mailserver	... bin\dbctl -ma <Passwort>
options	-mt	--mail-to	E-Mailadresse des Empfängers. Wird der Empfänger nicht angegeben, dann wird keine E-Mail versendet	... bin\dbctl -mt admin@...de
options	-mf	--mail-from	Absender der Email Default: dbctl@<aktueller vollständiger Geräteiname>	... bin\dbctl -mf backup@...de

Tabelle 2 - Parameter für den Mailversand