SuperX-Adminstrationshandbuch Kernmodul



www.MemText.de

- Daniel Quathamer danielq@memtext.de
- Meikel Bisping mbisping@memtext.de

http://www.superx-projekt.de Version 3.0final Stand 4.6.2007

> Download als PDF-Datei Lehrfilm zur Installation von Postgres Lehrfilm zur Installation des Kernmoduls

Sun, Sun Microsystems, Solaris, Java, JavaServer Web Development Kit, JDBC und Java-Server Pages sind eingetragene Warenzeichen von Sun Microsystems, Inc. UNIX ist ein eingetragenes Warenzeichen von X/Open Company, Ltd. Windows, WindowsNT, Win32, VBScript und Office 2000 sind eingetragene Warenzeichen von Microsoft Corp. Linux ist eingetragenes Warenzeichen von Linus Torvalds. Informix Dynamic Server, Informix Client SDK und Intersolv JDBC Driver sind eingetragene Warenzeichen der IBM Corp. HIS SOS, POS, SVA, MBS, BAU, LSF und COB sind Produkte der HIS GmbH. Alle weiteren Produktnamen sind Warenzeichen der jeweiligen Hersteller.

Dieses Produkt beinhaltet Software, die von der Apache Software Foundation (http://www.apache.org/) entwickelt wurde.

SuperX wird unter der deutschen Variante der GPL-Lizenz von dem Land Nordrhein-Westfalen, vertreten durch die FernUniversität Hagen, diese wiederum vertreten durch die Geschäftsstelle der Initiative CampusSource bei der FernUniversität Hagen, Feithstraße 142, D-58084 Hagen vertrieben (www.campussource.de). Details zu den Lizenzbedingungen finden Sie im Kernmodul-Archiv (/lizenz.txt) oder unter http://www.campussource.de/lizenz/. Ergänzende Hinweise finden Sie auf der Projekthomepage unter http://www.superxprojekt.de.

LizenzPostgreSQL Database Management Systemstgres:(formerly known as Postgres, then as Postgres95)		
Portions Copyright (c) 1996-2001, The PostgreSQL Global Development Group		
Portions Copyright (c) 1994, The Regents of the University of California		
Permission to use, copy, modify, and distribute this software and its documentation for any purpose, without fee, and without a written agreement is hereby granted, provided that the above copyright notice and this paragraph and the following two paragraphs appear in all copies.		
IN NO EVENT SHALL THE UNIVERSITY OF CALIFORNIA BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS, ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN IF THE UNIVERSITY OF CALIFORNIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.		
THE UNIVERSITY OF CALIFORNIA SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE SOFTWARE PROVIDED HEREUNDER IS ON AN "AS IS" BASIS, AND THE UNIVERSITY OF CALIFORNIA HAS NO OBLIGATIONS TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.		

Lizenz | Copyright 2002 Sun Microsystems, Inc. All Rights Reserved.

Java

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

-Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

-Redistribution in binary form must reproduct the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of Sun Microsystems, Inc. or the names of contributors may be used to endorse or promote products derived from this software without specific prior written permission.

This software is provided "AS IS," without a warranty of any kind. ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE HEREBY

EXCLUDED. SUN AND ITS LICENSORS SHALL NOT BE LIABLE FOR ANY DAMAGES OR LIABILITIES SUFFERED BY LICENSEE AS A RESULT OF OR RELATING TO USE, MODIFICATION OR DISTRIBUTION OF THE SOFTWARE OR ITS DERIVATIVES. IN NO EVENT WILL SUN OR ITS LICENSORS BE LIABLE FOR ANY LOST REVENUE, PROFIT OR DATA, OR FOR DIRECT, INDIRECT, SPECIAL, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF THE USE OF OR INABILITY TO USE SOFTWARE, EVEN IF SUN HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You acknowledge that Software is not designed, licensed or intended for use in the design, construction, operation or maintenance of any nuclear facility.

1	Einführung	10			
1.1 Sicherheitsaspekte					
1.2 Erforderliche Hardware					
	1.2.1 Datenbankserver				
	1.2.2 Webserver	12			
	1.3 Erforderliche Software	12			
	1.4 Das Kernmodul	13			
•	1.5 Ausbaustufen einer SuperX-Implementierung	15			
2	Installation	10 16			
	2.1.1 Übersicht über Installationsschritte	17			
	2.1.2 Besonderheiten für verschiedene Betriebssysteme	18			
	2.1.2.1 Windows / Cygwin	19			
	2.1.2.2 AIX / HP-UX	19			
	2.1.2.3 Noch nicht getestete Betriebssysteme	19			
	2.1.3 Kurzanleitung: Das Vorgehen -kurz und knapp für Linux-Systeme	20			
	2.1.4 Installation und Pflege der SuperX-Datenbank	21			
	2.1.4.1 Einrichten des Datenbankservers unter UNIX / LINUX	21			
	2.1.4.1.1 User superx - Kernmodul entpacken	21			
	2.1.4.1.2 Informix	22			
	2.1.4.1.2.1 Systemvoraussetzungen	22			
	2.1.4.1.2.2 Konfiguration	23			
	2.1.4.1.3 Installation von PostgreSQL	26			
	2.1.4.1.3.1 Neuinstallation (am Beispiel derVersion 7.3.4)	27			
	2.1.4.1.3.2 Postgres-Zusätze installieren: pgcrypto	30			
	2.1.4.1.3.3 Installation von Postgres unter Windows	31			
	2.1.4.1.3.4 Native Windows-Version (nur PowerGres, Postgres 8.0 oder höher)	31			
	2.1.4.1.3.5 Postgres unter Cygwin	31			
	2.1.4.1.3.6 Cygwin für SuperX	33			
	2.1.4.1.3.7 Postgres-Performance-Tipps	34			
	2.1.4.1.4 Datenbankverbindung über einen eingeschränkten User für mehr Sicherheit	35			
	2.1.4.1.5 Automatischer Start des Datenbankservers als Dienst	36			
	2.1.4.1.5.1 Einrichtung der Dienste	36			
	2.1.4.1.5.2 Aktivierung der Dienste	37			
	2.1.4.2 Einspielen des Kernmoduls der SuperX-Datenbank	38			
	2.1.4.3 Update und Sichern der Datenbank	38			
	2.1.4.3.1 Ein Dump unter Informix	39			
	2.1.4.3.2 Ein Dump unter Postgres	39			
	2.1.4.4 Anpassung der DB-Parameter für Chentanwendungen	39			
	2.1.4.4.1 Unter WIN32 auf den Informix-Server zugreifen: iLogin	40			
	2.1.4.4.1.1 SuperX (Informix) unter Win32 als ODBC-Datenquelle einrichten	40			
	2.1.4.4.2 Einrichtung des ODBC-Treibers für den Postgres-Server	42			
	2.1.4.4.5 Antindung des Access-Frontends an die ODBC-Quelle	44			
	2.1.4.4.4 Anpassen der Datenbankparameter für das SuperX-Serviet	45			
	2.1.4.4.5 Datenbankverbindung und Steuerung von DBForms	48			
	2.1.5 Installation und Pilege des webservers				

2.1.5.2	Einrichtung der Servlet-Engine	50
2.1.5.2.1	Steuerung des Servers: Die server.xml	51
2.1.5.2.2	Administrator und Manager	53
2.1.5.2.3	Einrichten des SuperX-Servlets unter Tomcat	53
2.1.5.2.4	Start des Tomcat	54
2.1.5.2.5	Die Übertragung der Web Application	55
2.1.5.2	2.5.1 Übertragung der Webapplikation auf einen vorhandenen Tomcat unter Wi	indows
	55	
2.1.5.2	2.5.2 Übertragung der Webapplikation auf einen vorhandenen Tomcat 5.5	56
2.1.5.2.6	Das SuperXManager-Servlet	
2.1.5.2.7	Verbesserung der Performance	57
2.1.5.2.8	Einrichtung einer SSL-Verbindung in Tomcat	57
2.1.5.2	2.8.1 Signierung eines Zertifikats in Tomcat	58
2.1.5.2.9	Zusätzliche Verschlüsselung im Applet durch Public-Private-Key-Kontrolle	58
2.1.5.2.10) Tomcat als Dienst unter Linux	59
2.1.5.2.1	Tomcat als Dienst unter Windows einrichten (nur WINNT/2000 und Tomcat	3.x)60
2.1.5.2.12	2 Steuerung für das Applet: Die superx.properties	61
2.1.5.2.13	3 Steuerung des XML-Frontends: PageComponents	61
2.1.5.2.14	Einrichtung des Webservers bei mehreren Mandanten.	
2.1.5.2.15	5 Einrichtung von DBFORMS bei mehreren Mandanten	
2.1.5.3	Integration von Tomcat mit dem Apache	67
2.1.5.3.1	Installation des Apache-Tomcat-Connectors	
2.1.5.3.2	Umleitung von Requests vom Apache zu Tomcat	
2.1.5.3.3	Einrichtung von Load Balancing	69
2.1.5.3.4	Einrichten von SSL beim Apache 1.3.x unter Linux	69
2.1.5.3.5	Einrichten von SSL beim Apache 2.x unter Linux	
2.1.6 Anpa	ssungen auf den Client-Rechnern	72
2.1.6.1	Installation der Java-Runtime	
2.1.6.1	1.1.1 Installation des Applets unter Windows	
2.1.6.1.2	Manuelle Anpassungen der Policy	
2.1.6.1.3	Installation des Applets unter UNIX / Linux	
2162	Bei Problemen mit dem Start des Applets	
2163	Leeren des Browser-Cache	
2.1.6.4	Leeren des Java - Cache	79
2.1.0.1 2.2. Undate ei	ner bestehenden SuperV Installation	70
2.2.1 Kurza	anleitung zum Upgrade von Version 2.0 nach 2.1	
2.2.1.1	Der Datenbankupgrade	80
2.2.1.2	Upgrade des Webservers	80
2.2.2 Kurza	anleitung zum Upgrade von Version 2.1 nach 3.0	
2.2.2.1	Der Datenbankupgrade	
2.2.2.2	Upgrade des Webservers	
2.2.3 Upgra	ade von SuperX Karlsruhe auf SuperX V2.0	
2.2.3.1	Erzeugen der Tabellen	
2.2.3.2	Erzeugen der Prozeduren	
2.2.3.3	Upgrade des Servlets und Applets	
2.2.3.4	Ändern der Masken	
2.2.3.5	Masken-Anpassung für den Java-Client	
2.2.3.5.1	Felderinfo	
=.=.0.0.1		

2.2.3.5.2 Maskeninfo	85
2.3 Checkliste Sicherheitsmassnahmen	
2.3.1 SSL-Verschlüsselung mit Zertifikat von Trustcenter	85
2.3.2 Applet ggfs. deaktivieren	86
2.3.3 Public-Private-Key-Kontrolle von Applet-Befehlen	
2.3.4 Datenbankverbindung über einen eingeschränkten Datenbank-User	
2.3.5 Einstellungen zur Passwortsicherheit	
2.3.6 Servertrennung für maximale Sicherheit	
2.3.7 Keine Verwendung von Standardkennungen	88
2.3.8 Keine Anzeige von internen Details bei Fehlermeldungen	88
2.3.9 Directory-Listing in Tomcat/Apache abschalten	
2.3.10 Sperren der DBFORMS-Komponente	
2.3.11 Kontrone von (Temernanen) Anmeldungen	
2.3.12 Logging von Aktivitäten im Adminibereich (dotorins)	
2.3.13 Entremen von temporaren Dateien	
2.1. Des SuperV Clientraket	
2.4 Das SuperA-Chentpaket	
2.4.2 Werkzeuge	
2.4.3 Installation von Entladescripten auf Basissystemen	
2.4.4 Entladen aus Windows-Datenbanken: Access	
3 Administration des Kernmoduls: HowTo	
3.1 Die SuperX-Administrationswerkzeuge	
3.1.1 Übersicht über Scripte unter UNIX	93
3.1.1.1 Die Umgebungssteuerung: SQL_ENV	94
3.1.1.2 Allgemeine Scripte	96
3.1.1.2.1 DOSQL	
3.1.1.2.2 DOQUERY	
3.1.1.2.3 sx_transform	
3.1.1.2.4 Propadmin	
3.1.1.3 Umgang mit Tabellen	97
3.1.1.3.1 sx_unload_table	
3.1.1.3.2 sx_upload_table	
3.1.1.3.3 sx_upload_records	
3.1.1.3.4 sx_schema	
3.1.1.4 Modulverwaltung	
3.1.1.4.1 module_scripts_create.x	
3.1.1.4.2 module_install.x	
3.1.1.4.3 module_drop.x	
3.1.1.4.4 module_update.x	
3.1.1.4.5 module_eti.x	
3.1.1.5. Maskan Varwaltung	
3.1.1.5 Masken Suchen	101
3.1.1.5.7 Fine Maske sichern und entladen	101
3.1.1.5.2 Eine Maske seitern und entiduen.	102
3.1.1.5.4 Fine Maske löschen	102
3 1 1 6 Änderungen an einer Maske vornehmen	
5.1.1.6 Anderungen an einer Muske vorheimen	105

3.1.2 Administration mit Abfragen im XML-Frontend	104
3.1.2.1 Das Organigramm bearbeiten	104
3.1.2.2 Den Themenbaum bearbeiten	
3.1.2.3 Userverwaltung	106
3.1.2.3.1 Einzelne Benutzer löschen, neu anlegen und Stammdaten ändern	106
3.1.2.3.1.1 Neuer Benutzer	107
3.1.2.3.1.2 Benutzer löschen	107
3.1.2.3.1.3 Benutzer bearbeiten	108
3.1.2.3.2 Gruppen anlegen, löschen und Stammdaten verwaltung	109
3.1.2.3.2.1 Neue Gruppe anlegen	109
3.1.2.3.2.2 Gruppe bearbeiten	110
3.1.2.3.2.3 Gruppe löschen	111
3.1.3 Das Access-Frontend	112
3.1.4 Weitere Tools	112
3.1.4.1 SQLWorkbench	112
3.2 Einen User betreuen	
3.2.1 Neuen User einrichten	114
3.2.2 Passwort vergessen	115
3.2.3 User-Rechte ändern	115
3.2.4 User löschen	115
3.3 Einstellungen zur Passwortsicherheit	116
3.4 Eine Gruppe betreuen	
3.4.1 Neue Gruppe einrichten	117
3.4.2 Gruppen-Rechte ändern	117
3.4.3 Eine Gruppe löschen	
3.5 Verwaltung und Rechtevergabe von Sichten	
3.5.1 Bearbeitung von Sichten	
3.5.2 Berechtigung für Sichten	
3.6 (Abtrage-)Masken entwickeln	
3.6.2 Maskenverwaltung mit MS Access	120
3.6.2 Effizientes Debugging	121
2.6.4 Delumentation von Abfragen Classen	
3.6.4 1 Allgemeine Sehlüggelwörter	
3.6.4.2 Der Spezielfell Meskenfelder	
3.6.5 Maskan für das XML Frontand vorbereiten	
3.6.5.1 Erzeugen eines Stylesheets	124
3.6.5.7 Zuordnung einer Macke zu einem Stylesheet	124
3.6.5.3 Annassung an Lesegeräte	
3.6.5.4 Einschränkungen des YML Erontende	120
3.6.5.5 Frweiterungen des XML-Frontends	128
3 6 5 5 1 Navigationsspalten im XMI -Frontend	120
36552 Export yon Abfragen nach RTF PDF und XI S	120
3.7 Installation von Modulen	121
3.7.1 Architektur von SuperX-Modulen	
3.7.2 Modulscripte im Kernmodul	
3.7.3 Installation eines Moduls: Allgemeines Vorgehen	
3.7.3.1 Dateitransfer mit scp/rsvnc	

	3.7.3.	2 Modulupdate in mandantenfähigen Installationen	134
	3.8 Übe	rwachung und Performance	
	3.8.1	Überwachung und Performance der Webanwendung	
	3.8.1.	1 Steuerung des SQL-Logging im SuperXManager	136
	3.9 Dow	vnloads einrichten und verteilen	
	3.9.1	Tabellenstruktur	
	3.9.2	Berechtigung für Downloads	
	3.9.3	Masken zur Erzeugung und Verteilung von Downloads	
	3.9.3.	1 Download suchen	
	3.9.3.	2 Download bearbeiten: Metadaten und Dateien	139
	3.9.3.	3 User- und Gruppenrechte auf Downloads	140
	3.9.3.	4 Stichworte für Downloads	140
4	Bestar	dteile des Kernmoduls: Die Referenz	
	4.1 Die	Userverwaltung	
	4.1.1	Verwaltung einzelner User	141
	4.1.1.	1 Tabelle userinfo	141
	4.1.1.	2 Tabelle user_masken_bez	
	4.1.1.	3 Tabelle sachgebiete	142
	4.1.1.	4 Tabelle sachgeb_maske_bez	143
	4.1.1.	5 Tabelle user_sachgeb_bez	143
	4.1.1.	6 Tabelle user_institution	144
	4.1.2	Gruppenverwaltung	145
	4.1.2.	1 Tabelle groupinfo	145
	4.1.2.	2 Tabelle user_group_bez	146
	4.1.2.	3 Tabelle group_masken_bez	146
	4.1.2.	4 Tabelle group_sachgeb_bez	146
	4.1.3	Zugriffsprotokollierung	147
	4.1.3.	1 Die Tabelle protokoll	147
	4.1.3.	2 Die Tabelle proto_funktion	148
	4.2 Das	Organigramm	
	4.2.1	Die Tabelle Organigramm	149
	4.2.2	Füllen des Organigramms	151
	4.2.3	Die Prozedur sp_user_orga	151
	4.2.4	Die Prozedur sp_user_orga_child	151
	4.3 Die	SuperX-Auswertungen	152
	4.3.1	Die Tabelle Maskeninfo	
	4.3.1.	1 SQL-Scripte	
	4.3.1.	2 Aufbau der Ergebnistabelle	154
	4.3.1.	3 Verbindung zur Tabelle felderinfo	155
	4.3.2	Tabelle Felderinfo	
	4.3.2.	1 Dialogsteuerung	
	4.3	.2.1.1 Angabe einer DB- Tabelle	159
	4.3	2.1.2 Angabe einer Stored Procedure	160
	4.3	2.1.3 Angabe eines SQL-Ausdrucks	160
	4.3	2.1.4 Hinweis für Dialogart 1 und 2	160
	4.3.2.	2 Vorgabewerte für die Felder	
	4.3	2.2.1 Konstanten	161
	4.3	.2.2.2 SQL-Ausdrücke	

	4.3.3		
4.3.4		Die Tabelle themenbaum	
	4.3.5	Verkettung von Masken: Die Tabelle macro_masken_bez	164
	4.4 Ein	zelne Schlüsseltabellen	
	4.4.1	Die Tabelle schluessel	
	4.4.2	Die Schlüsseltabelle aggregierung	
	4.4.3	Die Schlüsseltabellen cif und cifx	165
	4.4.4	Die Schlüsseltabelle trans_inst	
4.4.5		Weitere Schlüsseltabellen	169
	4.4.5	.1 Tabelle hochschulinfo	169
	4.5 Mo	dulbeschreibung	
5	Hinw	eise für Entwickler/innen	
	5.1 Kor	npilieren der Java-Quellen	
	5.1.1	Kompilieren mit Bordmitteln des JDK	
	5.1.2	Kompilieren mit dem Jakarta-Build-Tool ant	
	5.1.3	Entwicklung mit Jedit	
	5.2 Erz	eugung der SuperX-Hilfe im Javahelp-Format	
	5.3 Ver	sionshistorie	

1 Einführung

Das Berichtssystem SuperX ist ein sog. *Data Warehouse* für Bildungseinrichtungen, d.h. beliebig viele Datenquellen werden unter einer einheitlichen Auswertungsschnittstelle zur Verfügung gestellt. Da jede Hochschule unterschiedliche Datenquellen besitzt und in SuperX übernehmen will, bereiten wir für jede Datenquelle ein Modul vor, z.B. ein COB-Modul oder ein SOS-Modul. Bei Bedarf können Anwender auch eigene Module für proprietäre Datenquellen erzeugen und SuperX so erweitern.

Die Module enthalten die wichtigsten Prozeduren, Tabellen und Abfragen für die jeweilige Datenquelle. Der Startpunkt ist das Kernmodul. Eine Kurzanleitung für die Installation ist vorbereitet (S. 20).

Zur Geschichte von SuperX

SuperX wurde in den 90er Jahren an der Universität Karlsruhe von der Projektgruppe Abakus unter der Leitung von Herbert W. Roebke entwickelt. *SuperX* stand damals für: System zur Unterstützung von Planung und Entscheidung des Rektorats durch Information, Controlling und Simulation. In der damaligen Version, die im Folgenden als das "alte SuperX" bezeichnet wird, bestand das System aus einer SuperX-Datenbank (Informix) und einem Win32 / SGI / Mac-Client. Im Zuge der Verbreitung von WWW-basierten Frontends wurde im Jahr 2000 in Karlsruhe der Client nach Java portiert. Dieses SuperX-Applet wurde an der Universität Duisburg weiterentwickelt und aus Performance- und Sicherheitsgründen in eine Applet-Servlet-Anwendung (3-tier) geändert. Da SuperX sich vor allem dann als nutzbar erwiesen hat, wenn das Berichtssystem auf die Bedürfnisse der Anwender (in der Regel Hochschulen) zugeschnitten werden kann, ist die neue SuperX-Anwendung ein Open Source-Projekt, d.h. Anwender können die Datenbank und den Client für ihre Zwecke ändern. Lizenzrechtlich basiert SuperX auf der CampusSource-Lizenz, einer Variante der GPL (http://www.campussource.de/lizenz).

Um die Installation und die Weiterentwicklung von SuperX überschaubar zu halten, hat die Projektgruppe SuperX in Duisburg Ende 2001 beschlossen, die Datenbank in Module zu zerlegen. Das vorliegende Kernmodul soll sicherstellen, dass das gesamte System selbst nach Änderung von anderen Modulen weiterhin lauffähig und übertragbar bleibt.

Die vorliegende Dokumentation wird außerdem deutlich machen, dass die "neue" Architektur einige Änderungen an der SuperX-Datenbank erfordert. Bisherige Anwender der Karlsruher SuperX-Anwendung erhalten eine spezielle Anleitung für das Update (S. 82).

Falls es bei der Implementation des Kernmoduls zu Problemen kommt, können Sie sich unter www.superx-projekt.de informieren. Oder mailen Sie uns direkt: danielq@memtext.de bzw. mbisping@memtext.de.

1.1 Sicherheitsaspekte

Da SuperX für den Einsatz in großen Netzen konzipiert wurde, sind folgende Schutzmechanismen implementiert:

- Benutzer- und Paßwortkontrolle (S. 114)
- SHA-Verschlüsselung von Passwörtern
- Zugriffsprotokollierung (S. 147)
- Benutzerspezifische Einschränkung des Angebots an Abfragemasken (S. 152)
- Benutzerspezifische Einschränkung der einsehbaren Institutionen in Datenbankanfragen (S. 149)
- Getrennte Datenhaltung (operative Systeme SuperX)

- Abschottung der Datenbank gegenüber fremden Zugriffen (z.B. mit ODBC) durch Applet-Servlet-Architektur. Auch Client-Anwendungen wie das Informix Client SDK werden nicht eingesetzt.
- Verschlüsselte Verbindung von Client und Servlet (https), Möglichkeit der Zwischenschaltung eines Apache-Webservers (ggf. in der DMZ).
- Zusätzliche Verschlüsselung der in der Anwendung eingebaute Applet/Servlet-Kommunikation

Nur das SuperX-Servlet auf dem Webserver und die SuperX-Datenbankadministratoren auf der Serverseite haben einen direkten Zugriff auf die SuperX-Datenbank. Alle anderen Zugriffsmöglichkeiten für Benutzer können ausgeschlossen werden, d.h. kein Zugriff mit anderen SQL-Frontend-Programmen wie ISQL, DBACCESS (Unix) oder ODBC (Windows, Mac).

1.2 Erforderliche Hardware

Im Minimalbetrieb ist das gesamte SuperX-System auf einem Desktop-PC installierbar, z.B. auf einem SuSE-Linux 8.1-PC; dies reicht für den Testbetrieb im Intranet mit wenigen Usern vollkommen aus.

Für den Einsatz im Echtbetrieb unterscheiden wir mindestens drei Komponenten:

- Ein Datenbankserver
- Ein Java-basierter Web- und Applicationserver
- Die SuperX-Clients

Für jede Komponente gibt es unterschiedliche Empfehlungen.

1.2.1 Datenbankserver

Wir empfehlen generell Intel-Architektur mit Linux als Betriebssystem, da dies relativ kostengünstig ist und immer weitere Verbreitung findet. Zukünftige Versionen von SuperX werden verstärkt Java 1.4 und bash-2.x-Scripte auf dem DB-Server einsetzen, beides läuft sicher unter Linux.

DB-Server	• Intel-Architektur
	• 2GHZ-Prozessor, z.B. AMD Athlon oder Intel Xeon
	(Dual-Processor-System für Informix-Betrieb besonders
	sinnvoll)
	• 512 kb L2-Cache
	• 1 GB DDR-Ram
	RAID-Controller
	• Festplatte IDE oder SCSI, 7200 U/Min, UDMA/133
	1-6 GB Kapazität für DB

Wir empfehlen die Hochleistungsserver aus den aktuellen Produktpaletten von HP, Dell, Sun oder vergleichbaren Herstellern, die Firma Novell/SuSE zertifiziert auch Hardware für Linux. Für den produktiven Einsatz empfehlen wir einen Mittelklasse-Server mit dem Betriebssystem Linux. SuperX benötigt an einer größeren Hochschule (>10.000 Studierende, viele HIS-Systeme) erfahrungsgemäß 1-2 GB Platz für den DB-Server.

1.2.2 Webserver

Wirt empfehlen auch hier Intel-Architektur und Linux als Betriebssystem für Neuanschaffungen. Wenn bei SuperX nur das Applet als Frontend verwendet wird, sind die Anforderungen sehr gering: Ein Server mit 1- 2 GHZ-Prozessor, 256 MB DDR-RAM reichen völlig aus.

Der Webserver benötigt keinen Plattenplatz, aber eine leistungsfähige CPU. Die RAM-Anforderungen auf dem Webserver sind geringer, da wir mit Conncetion Pool arbeiten. Der SuperX-Webserver läßt sich auch ideal auf bereits konfigurierten Webservern (z.B. Homepages von Hochschulverwaltungen) einsetzen, sofern nicht bereits Tomcat oder ein anderer Application Server darauf läuft.

Wenn das XML-Frontend eingesetzt werden soll, empfiehlt sich eine etwas leistungsfähigere Architektur, ggf. sogar der Betrieb von zwei Webservern im software-basierten Lastausgleich ("load balancing" via Tomcat).

Web-Server	• Intel-Architektur
	• 2GHZ-Prozessor, z.B. AMD Opteron, Athlon oder Intel
	Xeon
	• 512 kb L2-Cache
	• 2 GB DDR-Ram
	• RAID-Controller
	• Festplatte IDE oder SCSI, 7200 U/Min, UDMA/133
	Kaum Festplatten-Kapazität notwendig

Generell gilt natürlich die Devise: So viel CPU-Taktfrequenz und RAM wie Sie sich leisten können.

Je nach Architektur der Umgebung sollte der Webserver 2 Netzwerkkarten enthalten, eine davon ist mit dem DB-Server verbunden, die andere mit dem Firewall-Rechner. Alternativ kann der Webserver auch in einer DMZ stehen, dann reicht eine Netzwerkkarte.

1.3 Erforderliche Software

Die SuperX-Datenbank läuft auf Windows- und Linux-Rechnern. Der SuperX-Client läuft auf allen Plattformen, die die Java-Runtimeumgebung (1.4.x) anbieten. Für den DB-Server empfehlen wir in jedem Falle einen UNIX bzw LINUX-Server, da alle serverseitigen Scripte als Shellscripte konfiguriert sind. SuSE Linux (Server-System mit Linux-Kernel, Version 2.2.16 oder höher, z.B. SuSE Linux 7.3) versteht sich gut mit Informix, aber alle aktuellen Linux-Distributionen enthalten bereits Java, Tomcat und Postgres

Beim Informix-Datenbankserver sollten Sie darauf achten, daß SuperX in einem eigenen Online-System läuft.

Die erforderliche Software für den Betrieb des Kernmoduls:

Software	Win32-Systeme	Linux / AIX	MacOS X
DB-Server			
SuperX-Datenbank	PostgreSQL 7.2.X oder höher (in Ver- bindung mit Cygwin)	Informix Dynamic Server f. Unix 7.3 oder höher bzw. PostgreSQL 7.2.X oder höher	PostgreSQL 7.2.X oder höher -
Webserver			
Webserver	 Apache oder IIS 4.0 / 5.0 oder (beliebig) 	Apache oder(beliebig)	(beliebig)
Servlet-Engine	• Tomcat 4.x	• Tomcat 4.x	(beliebig, sollte aber Verschlüsselung bie- ten)
Java	JDK 1.4.x	JDK 1.4.x	JDK 1.3.x-i (nur für MacOS X, nicht mehr für MacOS 8.x oder 9.x verfügbar)
Client			
Webbrowser	IE4, NS4 und höher oder andere, sollten aber Verschlüsselung bieten	NS4 und höher oder andere, sollten aber Verschlüsselung bieten	NS4 und höher oder andere, sollten aber Verschlüsselung bieten
Java-Runtime	JRE 1.4.x	JRE 1.4.x	JRE 1.3.x-i (nur für MacOS X, nicht mehr für MacOS 8.x oder 9.x verfügbar)
JavaHelp (Teil des Kernmoduls)	Jh1.1.1	Jh1.1.1	Jh1.1.1

Der Datenbankserver von SuperX läuft im Kernmodul 2.03 unter PostgreSQL; rudimentäre Funktionen des Kernmoduls sind bereits implementiert, das Projekt befindet sich nun im Beta-Stadium.

Im Kernmodul ist eine kleine Website enthalten, die eine Schritt-für-Schritt-Anleitung für die Installation des Clients für die gängigen Webbrowser enthält; diese finden Sie unter \$SUPERX_DIR/webserver/tomcat/webapps/superx/applet/install/

(Auf der Seite install.htm wird der aktuelle Browser per JavaScript festgestellt und auf eine entsprechende

Installationsseite weitergeleitet.)

1.4 Das Kernmodul

Das SuperX-Kernmodul beinhaltet alle zum Betrieb von SuperX unbedingt notwendigen Tabellen, Prozeduren und Abfragen; die wichtigsten Tabellen werden unten (S. 141) näher beschrieben.

Pfad		Beschreibung
db		Die SuperX-Datenbankseite
	bin	Shellscripte
	etc	Beispiel-Initscripte für SuperX-DB-Dienste
	install	Installationsscripte
	isql	isql-Formulare, Scripte und Berichte
	masken	Entladene Masken
	module	Modulpfad
doc		Dokumentation
src		Java-Quellen de.superx.*
webserver	tomcat	Tomcat-Beispielimplementation (Tomcat 3.2.2)
	apache	Apache-mod_jk (binär für SuSE Linux 8-9+ source)
	etc	Beispiel-Initscripte für SuperX-DB-Dienste

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

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





Der Datenbankserver kann auf einem anderen Rechner liegen als der Webserver; es ist aber auch möglich, das gesamte SuperX auf einem Rechner zu installieren. Je nach Hardware- oder Softwarevoraussetzungen kann dies ein WinNT/2000- oder Linux-Rechner sein. Unter Windows können Sie z.B. ein Verzeichnis C:\superx erstellen; unter Linux sollten 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,webserver) werden dort hineinkopiert.

Die Rohdaten der Module liegen in einem eigenen Unterverzeichnis rohdaten unterhalb des Modul-Verzeichnissesⁱ. Ggf. ist es zweckmäßig, aus dem Modulpfad einen symbolischen Link auf den Entladepfad vorzunehmen, z.B. im Pfad db/module/sva geben Sie ein:

ln ---symbolic <<Tatsächlicher Entladepfad>> rohdaten

1.5 Ausbaustufen einer SuperX-Implementierung

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 Aufbaustufen realisiert, wichtig ist daher die Skalierbarkeit des Systems vom Prototypen bis zum Echtbetrieb.

Das zu realisierende System besteht aus drei Komponenten: der Datenbank, der Webanwendung und des Clients (3-tier-Application). Die folgende Abbildung zeigt eine typsiche Beispielarchitektur:

Die Clients im Intranet greifen direkt oder über die Webanwendung auf die Datenbank zu. Die Clients im Internet greifen über den Browser (http oder für Verschlüsselte Zugänge https) auf die Inhalte zu.



Durch diese Architektur wird verhindert, dass WWW-Clients direkten Zugriff zur Datenbank haben. Bei mittlerer Last ist diese Architektur ausreichend.

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

Die SuperX-Datenbank Internet wird angebunden an ein Clients oder mehrere operative http / https http / https http / http Vorsysteme. Gleichzeitig, um die Webanwen-Intranet dung zu entlasten, ist es Last ausale Webanwendung möglich sein, die Last auf einen zweiten Webttp / https server auszulagern SuperX ("Load balancing"). Datenbank DB-Clients (z.B. odbc, jdbc) **Eine Operative** Clients

Datenbank

2 Installation

Die Installationsschritte beziehen sich auf die Neuinstallation und das Update. Für die Neuinstallation gibt es eine Kurzanleitung (S. 20) unter Linux.

2.1 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 Datei update-versionsnr-.tar bzw. update-versionsnr-.exe in \$SUPERX_DIR entpacken; die "alten" Dateien werden ersetzt, und die erforderlichen Schritte werden in der Datei update-versionsnr-.readme.txt beschrieben. Wenn Sie die Datenbank und den WWW-Server auf getrennten Systemen betreiben, dann entpacken Sie am besten

die Update-Datei in einem temporären Verzeichnis und kopieren dann die Ordner /db und /webserver 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 Scripte 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 derart komplexen Systemen wie SuperX.

2.1.1 Übersicht über Installationsschritte

Das Kernmodul wird in drei Arbeitsschritten installiert:

- 1. Installation und Einrichtung der Datenbank
- 2. Installation eines Webservers mit Servlet-Engine
- 3. Installation der Java Runtime auf den Clients

Die folgende Übersicht zeigt das Vorgehen bei der SuperX-Installation, darauf folgt eine Kurzanleitung für die Installationsmaßnahmen (S. 20):

Schritt	Erläuterung
Kopieren und Vorberei- ten des Kernmoduls nach \$SUPERX_DIR	Bringen Sie das SuperX-Kernmodul (S. 14) in ein Verzeich- nis auf dem Rechner, am besten auf den Datenbankserver; ggf. können Sie die Verzeichnisse /doc und /webserver auf einen anderen Rechner verschieben. Unter Windows können Sie z.B. ein Verzeichnis C:\superx erstellen und unter Linux einen Nutzer superx mit dem Ver- zeichnis /home/superx einrichten und alle Verzeichnisse des Kernmoduls (db, doc, webserver) dort hineinkopieren. Bei Be- trieb unter Windows muss das gesamte db-Verzeichnis auf einen UNIX-Rechner verschoben werden. Kopeiren Sie die Datei \$SUPERX_DIR/db/bin/SQL_ENV. sam nach \$SUPERX_DIR/db/bin/SQL_ENV und passen Sie die Umgebungs-
Installation der erforder- lichen Software	 variablen an. Installieren Sie auf dem Datenbankserver Informix (S. 21) oder PostgreSQL (S. 26) Installieren Sie auf dem Webserver Java die Datenbanktreiber (S. 50), und setzen Sie die Umgebungsvariable JAVA_HOME für Tomcat (S. 50) in der Datei \$\$UPERX_DIR/db/bin/SQL_ENV Installieren Sie auf den Clients die Java-Runtime und die Javahilfe
Start des Datenbankser- vers	Starten Sie den Datenbankserver und spielen Sie danach die SuperX-Datenbank (S. 38) des Kernmoduls ein; die Zugangsparameter (S. 45) müssen Sie dem SuperX-Servlet bekannt geben.
Start des SuperX-Servlets Test des Webservers	Gehen Sie in das Verzeichnis /webserver/tomcat/bin und star- ten Sie Tomcat, ggf. als Dienst Öffnen Sie die Datei http://< <rechnername>>:8080/superx/ im Browser und testen Sie zuerst die Anmeldung im XML- Frontend und dann im Applet</rechnername>
Freigabe des Webservers im Netz Einspielen der Module	Nun ist die Basisinstallation vom Kernmodul abgeschlossen, und Sie können mit der Anpassung für Ihre Einrichtung be- ginnen. 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. Füllen Sie SuperX mit den einzelnen Modulen; bisherige SuperX-Andender können ihr bisheriges System überneh-
Freigabe des Webservers im Netz Einspielen der Module	Browser und testen Sie zuerst die Anmeldung im XML- Frontend und dann im Applet Nun ist die Basisinstallation vom Kernmodul abgeschloss und Sie können mit der Anpassung für Ihre Einrichtung b ginnen. Richten Sie User ein, und geben Sie die WWW- Adresse Ihres Webservers im Intra- oder Internet (in diese Fall benötigen Sie auch die die Verschlüsselung) frei. Füllen Sie SuperX mit den einzelnen Modulen; bisherige SuperX-Andender können ihr bisheriges System überneh- men.

2.1.2 Besonderheiten für verschiedene Betriebssysteme

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

2.1.2.1 Windows / Cygwin

Unter Windows lassen sich derzeit der Applikaitonsserver von SuperX und Postgres betreiben. Der Datenbankserver läßt sich nur betreiben, wenn vorher die Unix-Emulation Cygwin installiert wird und wenn Postgres als Datenbanksystem gewählt wird.

Die aktuelle Cygwin-Distribution erhalten Sie von www.cygwin.com, das genaue Vorgehen haben wir bei der Installation von PostgreSQL (S. 31) beschrieben. Erfahrungsgemäß ist die Postgres-Version in Cygwin aktueller als in einer "normalen" Linux-Distribution.

2.1.2.2 AIX / HP-UX

Typischerweise (aber nicht notwendigerweise) fehlen auf AIX- oder HP-UX-Systemen eine aktuelle X11- und Java-Installation, so daß sich bei diesen Betriebssystemen nur der Datenbankserver betreiben läßt. Dabei ist zu beachten, dass auch der jdbc-Client (Umgebungsvariable \$SX_CLIENT) nicht benutzbar ist. Die Shellscripte wurden unter AIX (mit bash oder ksh als Terminalprogramm) getestet.

2.1.2.3 Noch nicht getestete Betriebssysteme

Folgende Betriebssysteme wurden bisher noch nicht als Plattfomen für SuperX getestet:

- Solaris
- MacOS X

2.1.3 Kurzanleitung: Das Vorgehen -kurz und knapp für Linux-Systeme

Lehrfilm zur Installation des Kernmoduls

Voraussetzungen	Postgres bzw. Informix IDS ist gestartet und läuft, der User exis-
	tiert im Datenbanksystem hat das Recht, Datenbanken zu erzeu-
	gen, Java ist installiert. Auf dem Server sollte kein weiterer Ser-
	ver-Dienst auf den Ports 8005, 8009 und 8080 (Tomcat-
	Standard-Ports) laufen.
Entpacken	Entpacken Sie das Kernmodul in /home/superx
	tar -xzvf kernmodul< <versionsnr>>.tar.gz</versionsnr>
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 Daten-	Gehen Sie in das Install-Verzeichnis
bank	cd \$SUPERX_DIR/db/install
	Starten Sie das Script
	kernmodul_erzeugen.x < <ggf. dbspace="" des="" mit="" name="">></ggf.>
	Bei Fehlern kommt eine Meldung, Protokolle stehen in
	create.log Damit ist die dh Seite eingerichtet
Stowerse adata; fiir	Danni Ist die do-Sene eingerichtet.
Steuerungsdatei für	INUN genen Sie in das verzeichnis
das Serviet:	Konjeren Sie dh postaroa properties bzw. dh
db.properties	Kopicien Sie do-postgres.properties bzw. do-
	Informix.properties Hach db.properties
	perspire fur Posigies:
	Staten Sie den Pron Δ dmin mit
	propadmin.x
	Passen Sie hier den Servernamen. Datenbanknamen. Usernamen
	und Passwort an und drücken Sie "Verbindung testen"
	Danach speichern Sie die Datei
Connection Pool für	Editieren Sie die Datei
dhfarme	CIDERTY DIR /undranzion /tomast /conf/contrary und nassen Sie am
ubiti IIIS	Endo im Abachnitt (December Devene and a With Amagen and Ala
	Linuc III Auschlinu <kesourceparams name="jobc/superx"> ule Ver-</kesourceparams>
	omdungsparameter an, die Sie oden auch dem SuperA-Serviet
	gegeben haben (also driverClassName und url).

Steuerungsdatei für	Nun gehen Sie in das Verzeichnis
das Applet:	\$SUPERX_DIR/webserver/tomcat/webapps/superx/applet
superx.properties	cd \$SUPERX_DIR/webserver/tomcat/webapps/superx/applet
	Kopieren Sie die Datei superx-postgres.properties.sam bzw.
	superx-informix.properties nach
	superx.properties
	cp superx-postgres.properties superx.properties
	Editieren Sie die Datei, und tragen Sie bei SxTitle den Hoch-
	schulnamen ein.
	Wenn Sie Informix benutzen, muss bei SxDB "Informix" stehen,
	bei Postgres "Postgres"
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 das Applet und das XML-Frontend aufrufen.
	Geben Sie bei der Kennung superx ein, und als Passwort
	"anfang12"
Feineinstellung	Melden Sie sich im XML-Frontend an und gehen Sie im Menü
8	auf "Tabelle suchen", klicken einfach "Abschicken" und öffnen
	Sie die Tabelle hochschulinfo (S. 169): dort sollten Sie Ihre
	Hochschule auswählen und "Sneichern" anklicken damit Ihre
	Hochschulnummer gespeichert wird
	nochschumunnner gespeichert wird.

2.1.4 Installation und Pflege der SuperX-Datenbank

Die SuperX-Datenbank liegt als exportierte Datei in dieser Distribution vor und kann einfach importiert werden. Zunächst muss aber der Datenbankserver eingerichtet werden. Derzeit laufen die Installationsscripte und auch alle Modulscripte nur unter UNIX /Linux / Cygwin. Bei Betrieb von SuperX unter Windows muss also das gesamte Verzeichnis db auf einen UNIX-Rechner kopiert und betrieben (S. 26) werden. Deshalb empfehlen wir für den Anfang einen Linux-Rechner, da hier die Java-Unterstützung kein Problem ist.

2.1.4.1 Einrichten des Datenbankservers unter UNIX / LINUX

Der Datenbankserver läuft unter Informix (mind. Version 7.31) und PostgreSQL (mind. Version 7.2).

2.1.4.1.1 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 natülrich auch jedes andere Verzeichnis möglich.

Es muss auf Betriebssystemebene sichergestellt werden, dass das Dateisystem Textdateien im ISO-Format LATIN-1 anlegt (und nicht UTF-8). Bei RedHat-Linux z.B. wird die Umgebungsvariable \$LANG auf diesen Wert gesetzt. Prüfen mit echo \$LANG, welche Sprache eingestellt ist. Wenn nicht de_DE.ISO-8859-1 oder de_DE@euro eingestellt ist, prüfen Sie unter Linux im Verzeichnis /usr/lib/locale, welche der beiden Locales existiert (ggfs. nachinstallieren).

Setzen Sie eine der beiden Locales z.B. mit $LANG=de_DE@euro;$ export LANG.

Entpacken Sie die kernmodul-XX.tar.gz imVerzeichnis \$SUPERX_DIR.

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

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

2.1.4.1.2 Informix

SuperX unter Informix läuft derzeit unter UNIX und LINUX. Für den Datenbankserver unter Windows NT benötigen Sie in jedem Fall einen UNIX / LINUX-Rechner für die Shellscripte in den Modulen. Das Vorgehen ist im Abschnitt Konfiguration (S. 26) beschrieben..

2.1.4.1.2.1 Systemvoraussetzungen

Da die meisten Hochschulen bereits Informix-Datenbanken einsetzen, sind hier keine Hinweise zur Installation nötig. Da SuperX ein beliebtes System für Linux-basierte Systeme ist, hier nur ein paar kurze Hinweise für Informix unter Linux.

Informix für Linux lässt sich ab Version 7.3 unter Linux installieren (wir haben SuSE Version 7.3-8.1 und RedHat 8/9 getestet). Gemäß Anleitung von IBM/Informix geht man so vor:

- 1. Als root anmelden
- 2. User und Gruppe informix anlegen
- 3. Die Umgebungsvariable export INFORMIXDIR=/home/informix setzen
- 4. Dann die Informix- sql-CD einlegen und mounten
- 5. Das tar-Archiv sql.tar nach /home/informix entpacken
- 6. ./installsql starten (Seriennummer etc eingeben)
- 7. Dann die IDS_2000-CD einlegen und mounten
- 8. Dann startet man unter SUSE Linux oder RedHat9 (bei RedHat kann man den Hinweis, dass die Installation vom user informix gemacht werden sollte ignorieren) rpm -i --relocate /opt/informix=/home/informix /mnt/IDS_2000/IDS.RPM

Unter **RedHat 8.0** existiert ein Bug im RPM-Programmⁱⁱ, deshalb kann man Informix nicht nach /home/informix, sondern nur nach /opt/informix installieren und vorher die Umgebungsvariablen setzen: RPM_INSTALL_PREFIX=/opt/informix INFORMIXDIR=/opt/informix Danach startet man die Installation mit rpm -i ids.rpm

Damit ist der IDS installiert.

Die Bibliothek libpthread muss richtig eingebunden werden. Überprüfen kann man das Einbinden der Bibliotheken über

ldd \$INFORMIXDIR/bin/oninit

Es muss erscheinen (vielleicht über einen symbolischen Link):

```
libpthread.so.0 => /lib/i686/libpthread.so.0
...
```

9. für RED-Hat 8 und 9

. . .

Wenn beim oninit die Fehlermeldung erscheint

oninit: relocation error: /var/lib/libpthread.so.0: symbol __on_exit, version GLIBC_2.0 not defined in file libc.so.6 with link time reference .

Dann muss man unter RedHat noch eine Bibliothek ändern:

I. Als root bennenen Sie den alten Link um:

mv /var/lib/libpthread.so.0 /var/lib/libpthread.alt.

II. Dann erzeugen Sie einen neuen Link:

ln -s /lib/i686/libpthread.so.0 /var/lib/libpthread.so.0

Um die menübasierten Tools von Informix (dbaccess, onmonitor) zu nutzen, muss man falls eine entsprechende Fehlermeldung erscheint . die ältere libncurses.so.4 einbinden. Man prüft, wo libncurses.so.4 auf der Platte liegt und erstellt einen symbolischen Link.

ln -s /usr/lib/libncurses.so.5 /usr/lib/libncurses.so.4

2.1.4.1.2.2 Konfiguration

Die Kofiguration des IDS geschieht im onmonitor über das Menü Mode->Parameters oder direkt in der Textdatei "onconfig", für unser Beispiel onconfig.superx. Die Pfade zu \$INFORMIXDIR müssen ggf. angepaßt werden, die Voreinstellung ist oft "/usr/informix". Wichtig ist außerdem der DBSpace (zu prüfen mit onstat -d).

Zum Betrieb von SuperX hier nur einige Angaben zur empfohlenen Größe: Für das Kernmodul selbst würden 100 MB ausreichen, wenn Sie aber als erstes das SOS-Modul installieren möchten, sollten Sie nicht unter 400 MB starten (Parameter ROOTSIZE in onconfig.superx s.u.).

Wir empfehlen, das Logging auszuschalten, da SuperX keine Dialog-Anwendung ist und durch die Prozeduren sehr viel Logging anfallen würde. Selbst bei ausgeschaltetem Logging entstehen noch sehr viele Eintragungen, deshalb sollten Sie als Log Archive Tape Device /dev/null angeben.

Für die Rohdaten aus den operativen Systemen gibt es ein eigenes Verzeichnis, z.B.

\$SUPERX_DIR/db/module/<<modulname>>/rohdaten. Aus Platzgründen, und um sich den ftp-Transfer zu ersparen, bietet es sich unter UNIX an, hier NFS-Laufwerke einzurichten.

Falls Sie noch keine onconfig Datei für SuperX haben, erstellen Sie eine Kopie von

/home/informix/etc/onconfig.std und nennen Sie sie onconfig.superx.

Wenn man den DB-Space in einem Cooked-File ablegen will, kann man z.B. als root eine leere Datei /var/informix/rootdbs erstellen (z.B. leere Datei mit vi). Beim DB-Space müssen Sie darauf achten, dass der Benutzer und die Gruppe informix Schreibrechte auf den Cooked File bzw. die Datenpartition haben. Dieser Pfad muss dann als Parameter für den DB-Space in der onconfig.superx angegeben werden. ROOTNAME rootdbs ROOTPATH /var/informix/rootdbs

In der onconfig-Datei für SuperX sind die Parameter DBSERVERNAME (wir empfehlen superx_host) und DBSERVERALIAS (wir empfehlen superxdb) wichtig.

Entsprechend dieser zwei Parameter ergänzen Sie die Datei sqlhosts.

\$INFORMIX Beispiel:	X/etc/sqlhosts	Die Datei n Zugriff (sta ten Rechne	nit den Hostnamen für Shared Memory- att miles geben Sie den in /etc/hosts definier- ernamen an) und für TCP-Zugriff.
#*********	*****	* * * * * * * * * *	* * * * * * * * * * * * * * * * * * * *
#Servername superx_host superxdb	Protokoll onipcshm onsoctcp	Host miles miles	Service superx_shm superx_server

Machen Sie eine Ergänzung in /etc/services

/etc/services | Der SuperX-Service mit Portnummer

<mark>s</mark>uperx_server 1542/tcp

Unter Informix für Windows NT befindet sich die onconfig unter %INFORMIXDIR%\etc\onconfig, die sqlhosts wird in der Registry unter HKEY_LOCAL_MACHINE oder besser über das Programm setnet32 geändert.

Wichtig ist die Eintragung eines DBSERVERALIAS, über den das Servlet die Verbindung aufbaut. Der Port des Service in /etc/services wird ebenfalls benötigt.

Diese Parameter werden in der Datei db.properties (S. 45) vom SuperX-Servlet benötigt.

Es muss sichergestellt werden, dass einige Umgebungsvariablen beim Start initialisiert werden. Je nach UNIX-Art geschieht das in der .profile oder. .bashrc im Home-Verzeichnis der Benutzer informix und superx. (Im Zweifelsfall ausprobieren)

Damit man die Umgebungsvariabeln nur an einer Stelle zu pflegen braucht, empfiehlt es sich, dem User Informix Leserechte auf die Datei \$SUPERX_DIR/db/bin/SQL_ENV zu geben und diese in der .profile bzw. .bashre der beiden User aufzurufen.

Wichtig für den Ein-	#SX_CLIENT=pgsql;
satz unter Linux /	SX_CLIENT=dbaccess
University SOL ENV	SUPERX_DIR=/home/superx; export SUPERX_DIR
Unix: die SQL_ENV	INFORMIXDIR=/home/informix; export INFORMIXDIR
unter Informix	INFORMIXSERVER=superx_host; export INFORMIXSERVER
	ONCONFIG=onconfig.superx; export ONCONFIG
	echo
	echo "ONCONFIG: "\$ONCONFIG
	echo "INFORMIXSERVER: " \$INFORMIXSERVER
	CLIENT_LOCALE=de_de.8859-1; export CLIENT_LOCALE
	DB_LOCALE=de_de.8859-1; export DB_LOCALE
	SERVER_LOCALE=de_de.8859-1; export SERVER_LOCALE
	TERMCAP=\$INFORMIXDIR/etc/termcap; export TERMCAP
	TERM=ansi; export TERM
	#Terminal für TeraTerm Pro auf Win32-Systemen:pctcp
	PATH=\${PATH}:\${SUPERX_DIR}/db/bin:\$INFORMIXDIR/bin; export
	PATH
	DBDELIMITER=^ export DBDELIMITER
	DBDATE=DMY4.; export DBDATE
	DBMONEY=.; export DBMONEY
	TERMINAL=`tty`; export TERMINAL
	#Wenn auf DB-Server auch Webserver / Tomcat läuft
	#Beispiel für Suse Linux 7.3-Installation:
	export JAVA_HOME=/usr/lib/jdk1.3.1

Stellen Sie sicher, dass die Zeile #SX_CLIENT=pgsql; mit dem Gatterzaun auskommentiert ist und die Zeile SX_CLIENT=dbaccess nicht;

In dieser Datei werden auch die Pfade und Parameter für das Laden der Daten aus den operativen System festgelegt. Sie wird von den Entladescripten und von den Cronjobs benutzt.

Für Informix ist es generell günstiger, unter Unix / Linux mit einem ANSI-Terminal zu arbeiten. Beachten Sie allerdings, daß bei dieser Einstellung kein xterm verfügbar ist und Sie somit keine graphischen Java –Anwendungen, z.B. den propadmin (S. 45), auf dem Datenbankserver starten können.

Die Umgebungsvariablen DBTEMP und PSORT_DBTEMP sind eigentlich nicht mehr notwendig; wenn es Probleme beim Sortieren und Auslagern auf temporäre Datenträger gibt, dann sollte man diesen Pfad ebenfalls setzen. Die onconfig.superx liegt unter \$INFORMIXDIR/etc und muss unbedingt als Parameter die Zeile

DBSPACETEMP dbtemp # Default temp dbspaces

enhalten, wobei der Name abtemp im onmonitor frei gewählt werden kann.

Ist die Umgebung korrekt eingerichtet, dann startet man den IDS mit

Erstmaliger Start des oninit -ivy IDS

Weitere nützliche Kommandozeilen-Befehle für Informix

oninit	startet den Datenbankserver und bringt ihn in onli-
	ne-Modus
onstat	Zeit die aktuellen Prozesse des DB-Servers an
oninit -s	Von Offline nach quiescent
onmode -m	Von quiescent nach online
onmonitor	Zeigt aktuellen Status sowie ein Menü zur Admi-
	nistration an
tail -200 \$INFORMIXDIR/online.log	Zeigt das Ende der Logdatei an
ipcs -m	Anzeigen von Shared Memory für die Datenbank
oncheck -pt < <daten-< td=""><td>Zeigt die Extents einer Tabelle an</td></daten-<>	Zeigt die Extents einer Tabelle an
bank>>:< <tabellenname>></tabellenname>	
oncheck -ce	Zeigt den genutzten Speicherbedarf der Extents für
	jeden dbspace an

Dann kann man die Datenbank als User superx einspielen (s.u.).

Für den Ablauf der UNIX-Scripte zu den Masken (S. 93) (sx_select_mask, sx_insert_mask etc.) und für Cron-Jobs müssen die Parameter in der Datei *\$SUPERX_DIR/db/bin/SQL_ENV* stimmen.

Hinweis für Datenbankserver unter AIX oder anderen Linux / Unix-Derivaten: Beachten Sie, daß die Scripte nur dann lauffähig sind, wenn auf dem Datenbankserver unter /bin/bash die bash Version 2.x oder höher liegt (bzw. gelinkt ist). Die Scripte von SuperX erwarten die bash-Shell im Verzeichnis /bin; wenn dies nicht der Fall ist, sollte die Datei sh z.B. von /usr/bin nach /bin kopiert oder gelinkt werden. Unter Ubuntu Linux 6.10 beispielweise ist die Standardshell nach /bin/dash gelinkt, dies müssen Sie für SuperX ändern.

Exkurs: SuperXInformix kann auch auf einem UNIX-Rechner installiert werden, der nur
als Client auf einen anderen Server zugreift, z.B. einen Informix-Server
unter Windows NT. Dazu müssen die sqlhosts-Einträge auf beiden Rech-
nern übereinstimmen, und der Port des Service muss in /etc/services ste-
hen. Der Zugriff auf einen Remote-IDS geht dabei nicht über Shared
Memory, sondern über tcp.
Im Verzeichnis db/bin des Kernmoduls steht die Datei
SQL_ENV_fuer_remote_entladen.sam
Diese Beispieldatei für Informix zeigt, wie man den SuperX-Rechner
als Client auf eine anderen Datenbankserver nutzen kann.
Dies ist nützlich, da so die Entladescripte nicht auf dem DB-Server
des operativen Systems laufen, sondern auch auf dem SuperX-Rechner.

2.1.4.1.3 Installation von PostgreSQL

Lehrfilm zur Installation von Postgres

SuperX ist seit Version 2.1 mit Postgres 7.2 oder höher lauffähig. Die Distribution von Postgres für Unix findet sich unter www.postgresql.org. Eine Version für Windows befindet sich im Cygwin-Paket, dass Sie von unserem www.cygwin.com beziehen können. Eine allgemeine Anleitung befindet sich unter http://www.postgresql.org/idocs/index.php?install-upgrading.html, Spezialitäten für Cygwin finden Sie unten (S. 31).

Verschiedene Linux-Distributionen enthalten zwar bereits Postgres und müssen nicht "von Hand" installiert werden, doch spätestens beim Datenbankupdate ohne die jeweils neue Distribution rächt sich dies: Die Distribution legt Postgres in anderen Verzeichnissen ab, und das Installationsscript von Postgres passt dann nicht mehr. Dieses Problem besteht unter Red Hat Linux und SuSE Linux. Wir empfehlen daher, die in die Distribution "eingebaute" Version von Postgres zunächst zu deinstallieren.

2.1.4.1.3.1 Neuinstallation (am Beispiel derVersion 7.3.4)

Voraussetzun- Postgres läuft unter verschiedenen UNIX-Varianten, z.B. Linux, HP-UX

gen oder MacOS X. Wir empfehlen für den Einstieg Linuxⁱⁱⁱ. Vor der Installation unter Linux sollte die Locale-Umgebungsvariable \$LANG auf de_DE@euro oder eine andere deutsche Locale (meist in /usr/lib/locale) stehen – die aktuelle Locale wird bei der Installation von Postgres berücksichtigt und sorgt dafür, dass Datums- und Währungsformate korrekt sind.
Bei SuSE Linux 7.x bis 10.x 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.

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 Postgres-7.2-Version ist es wichtig, den Datenbankserver für internationale Sprachumgebungen zu konfigurieren, deshalb bietet es sich an, die Installation nicht als Binary- sondern als Quell-Installation vorzunehmen^{iv}. Bei höheren Versionen von Postgres ist dies nicht mehr nötig.

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 postgresq1-7.3.4, und führen folgende Befehle aus:

Postgres 7.38.x	./configureenable-localeenable-nls
Installation	make
	make install
in short	mkdir /usr/local/pgsql/data
	chown postgres /usr/local/pgsql/data

Wenn Sie Postgres 7.2.x installieren, müssen beim ./configure der Parameter ---enable-multibyte=IATIN1 gesetzt werden, in Postgres 7.3 oder höher ist dies defaultmäßig bereits eingebaut sind.

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

su - postgres

Vor der Initalisierung 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 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:

```
...

export LANG=de_DE@euro

#Zur Sicherheit für Postgres auch einzeln:

export LC_CTYPE='de_DE@euro'

export LC_COLLATE='de_DE@euro'

export LC_TIME='de_DE@euro'

export LC_NUMERIC='de_DE@euro'

export LC_MONETARY='de_DE@euro'

export LC_MESSAGES='de_DE@euro'

PATH=$PATH:/usr/local/pgsql/bin

export PGDATA=/usr/local/pgsql/lib

...
```

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

```
Initialisierung des /usr/local/pgsql/bin/initdb –D $PGDATA
DBSPACE
```

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 "post-
gres".
This user must also own the server process.The database cluster will be initialized with locale de_DE@euro.
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.

Hinweis für SuSE-
AnwenderWenn Sie Postgres als Binärpaket aus der Distribution von SuSE
8.x und 9.x verwenden, müssen Sie beachten, dass der
DBSPACE beim ersten Start des Postmaster automatisch in
/var/lib/pgsql/data angelegt wird. Wenn dabei die Umgebungs-
variablen nicht auf die deutsche Locale gesetzt sind, wird ein
amerikanischer Zeichensatz benutzt (Default-Einstellung). Bevor
Sie also das Init-Script z.B. im Runlevel-Editor des YAST starten,
sollten Sie mit dem Sysconfig-Editor (im Yast: System->Editor
für Sysconfig-Dateien) die Variable POSTGRES_LANG (im Yast: Su-
che nach "POSTGRES") auf die deutsche Locale de_DE@euro set-
zen. Noch ein Fallstrick in SuSE 9.1: Wegen eines Bugs im Yast
funktioniert die Suche im Sysconfig-Editor nur im Textmodus.

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
 host
 all
 superx
 192.168.0.16
 255.255.255
 trust

 pg_hba. conf
 -</

Die obige Zeile gibt dem User superx Verbindungsrechte für alle Datenbanken auf dem lokalen Rechner 192.168.0.16. Wenn Sie den Datenbankserver für Subnetze freigeben wollen, müssen Sie bei der IP-Nr. einen 0 für das Subnetz setzen, und bei der Netzmaske ebenfalls, also z.B.

192.168.0.0 255.255.255.0

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.

Weitere Parameter werden in der Konfigurationsdatei postgresql.conf definiert; wichtig ist die Einstellung, dass Postgres einen TCP-IP-Socket öffnet (Parameter tcpip_socket=true bei Postgres 7.x, listen_addresses=<<IP-Nr.>> bei Postgres 8.0) sowie der TCP-IP-Port (port = 5432 ist die Standardvorgabe). Die Anzahl der gleichzeitig offnenen Verbindungen muss kleiner sein als die Anzahl, die Sie für das SuperX-Servlet (S. 45) definieren. Weitere Details zur Einrichtung von Postgres-Runtime-Parametern finden Sie im Admin-Handbuch der Postgres-Distribution. Außerdem sollen Sie beim Betriebssystem SuSE 9.1 oder höher den IPV6-Eintrag für "localhost" (::1) in /etc/hosts auskommentieren.

/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/postgres.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 <code>\$SUPERX_DIR/db/install</code>):

pgsql start.x	#!/bin/sh
Fin Beisnielscrint zum	PG_HOME=/usr/local/pgsql
Ctart man Dastana	export PG_HOME
Start von Postgres	PGDATA=\$PG_HOME/data
	export PGDATA
	PGPORT=5432
	export PGPORT
	<pre>\$PG_HOME/bin/pg_ctl -D \$PGDATA -l /var/log/postgres.log -o -i</pre>
	start

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

Öffnen Sie mit
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
ddd
(3 rows)

Im Verzeichnis \$SUPERX_DIR/db/install befindet sich ein Shellscript 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 ^v. 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.

SuperX benötigt die Prozedursprache **plpgsql**. Wenn Sie als SuperUser die Prozedursprache installieren wollen, geben Sie in der Shell ein:

createlang plpgsql

Damit ist Postgres installiert und für die SuperX-Installation konfiguriert. Bei dieser Gelegenheit sollten Sie den Datenbankserver gleich als Dienst (S. 36) beim Systemstart einrichten.

2.1.4.1.3.2 Postgres-Zusätze installieren: pgcrypto

Neben dem Kernsystem von Postgres bietet es sich an, die vielen Zusatzmodule von Postrges 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:

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

Geben Sie ein:

qmake all

gmake install

Es werden Bibliotheken in /usr/local/pgsql/lib erzeugt. Das SQL-Script zur Erzeugung der Crypo-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

2.1.4.1.3.3 Installation von Postgres unter Windows

Für die Installation von Postgres unter Windows existiert seit Postgres 8.0 eine Möglichkeit, Postgres nativ zu betreiben. Dies empfehlen wir. Aus historischen Gründen haben wir auch den Betrieb von Postgres unter Cygwin dokumentiert.

Für den Betrieb von SuperX wird aber auf jeden Fall die Shell-Umgebung von Cygwin benötigt. Dies wird in einem dritten Abschnitt erläutert.

2.1.4.1.3.4 Native Windows-Version (nur PowerGres, Postgres 8.0 oder höher)

Seit längerem gibt es eine kostenpflichtige Windows-Version von Postgres unter dem Namen PowerGres. Mit der Version 8.0 läuft auch das "normale" Postgres nativ (d.h. ohne die Unix-Emulation Cygwin) unter Windows, allerdings nur unter Win2000 und WinXP (nur XP Professional, nicht XP Home). Dies bietet erheblich mehr Komfort bei der Installation und Stabilität beim Betrieb. Für SuperX müssen Sie aber in jedem Fall cygwin installieren (s.u.), da die SuperX Scripte nur unter Unix / bash laufen.

Laden Sie die neueste Version von Postgres (Win) herunter.

- Installieren Sie als Administrator das msi-Paket, z.B. im Verzeichnis C:\Programme\PostgreSQL\8.0betal. Achten Sie darauf, daß alle Pakete installiert werden, auch **pgadmin III** (ältere pgadmins, odbc- oder jdbc-Treiber funktionieren nicht).
- Der User, der postgres startet, muss ein normaler User sein(z.B. "postgres"), kein Administrator; er muss vorher unter Windows angelegt sein. Er ist auch der Eigentümer der Datenbank **templa-te1** (der Superuser).
- Postgres sollte als Dienst installiert werden
- Beim Anlegen des Datenbank-Cluster legen sie die deutsche Locale an, und als Zeichenformat LATIN1 (nicht unicode). Das Dateisystem muss NTFS sein.
- psql & co dürfen für den Betrieb von SuperX beim User nicht in den Windows-PATH gesetzt werden (z.B. C:\Programme\PostgreSQL\8.0-x\bin), stattdessen nehmen wir die Cygwin-Applikationen (s.u.).
- in C:\Programme\PostgreSQL\8.0-x\data\postgresql.conf muss man statt früher tcpip_socket = true den Parameter listen adress ='IP-Adresse'
- In der Datei pg_hba.conf ist die Standardanmeldung anders als unter Unix auf md5 (nicht trust) gesetzt; wenn Sie nicht ständig das User-Passwort eingeben wollen, sollten Sie den entsprechenden Passus auf "trust" setzen.

Damit ist Postgres konfiguert, Sie können den Dienst jederzeit in der Computerverwaltung über das Applet "Dienste" neu starten. Normalerweise startet Postgres dann auch beim Systemstart automatisch.

2.1.4.1.3.5 Postgres unter Cygwin

Neben der nativen Postgres-Installation (die wir empfehlen) gibt es auch die Möglichkeit, Postgres unter Cygwin zu betrieben. Insgesamt eignet sich eine unter Cygwin kompilierte Postgres-Installation unter nur für den Testbetrieb, denn bei der Sortierung werden Umlaute falsch eingeordnet und es wird sehr großzügig mit der Prozessorlast umgegangen: Wenn Postgres-Prozesse laufen, dann ist die Performance des Rechners für andere Anwendungen weitgehend gesperrt.

Aber auch bei der nativen Postgres-Installation unter Windows benötigen Sie für Postgres und SuperX unter Windows die UNIX-Shell-Emulation **cygwin**. Cygwin bietet rudimentäre UNIX-Funktionen wie z.B. die "bash", aber keine UNIX-typischen Dateirechte (z.B. Ausführungsrechte für User, Gruppen oder Andere). Außerdem unterstützt Cygwin (unseres Wissens) keine Locales, und unter Win98 haben wir keine stabile Installation hinbekommen. In den Mailinglisten wurden häufiger Probleme mit Win98 berichtet, unter WinME, Win2000 und Windows XP haben wir Cygwin erfolgreich getestet.

Das folgende Beispiel arbeitet mit Postgres 7.4.x. Postgres ist als Paket im Installer von Cygwin auswählbar.

Für die Installation muss man eine Windows-Kennung benutzen, die Rechte für "Standardbenutzer" reichen aus (es sei denn Cygwin soll als Dienst laufen). Außerdem: Wenn Sie planen, Daten bzw. entladene Datenbank-Exporte zwischen verschiedenen Rechnern hin- und herzuschieben, sollten Sie darauf achten, dass Sie immer die gleiche Kennung benutzen. Sie können z.b. superx nehmen. Die Windows Kennung, unter der man Cygwin installiert, wird nämlich nach Cygwin durchgereicht.

Vorgehen:

1. Die setup-Datei setup.exe der Unix-Emulation Cygwin von http://www.cygwin.com herunterladen und starten

Dann je nach Belieben direkt aus dem Internet installieren oder zunächst herunterladen und dann install from local directory (alle Komponenten ausgewählt lassen) anklicken (wir empfehlen letzteres Vorgehen, da das Online-Cygwin-Paket ständig aktualisiert wird).

- 2. Als Installationspfad sollten Sie unbedingt einen Pfad wählen, der keine Leerzeichen enthält, z.B. c:\cygwin).
- 3. Bei der Frage, für welchen User Cygwin installiert werden soll, wählen Sie "All users", und beim Standard-Dateiformat wählen Sie Unix.
- 4. Bei der Auswahl der Pakete sollten Sie wie folgt vorgehen: Bei den Shells mauss auf jeden Fall die bash ausgewählt sein. Zusätzlich zu den Defaults müssen lediglich Base -> TextUtils, Database -> Postgres, Admin -> cron, net->openssh und Libs -> libint und libint1 manuell ausgewählt werden. Ein Mailprogramm (mutt, mail) sollte auch installiert werden. Wenn Sie Postgres selbst aus den Quellen installieren wollen, dann wählen Sie natürlich nicht Postgres aus.
 - Danach einmal starten, das home-Verzeichnis wird angelegt
- 5. Das Cygwin-/bin Verzeichnis muss in der Umgebungsvariable PATH vor den Windows Programm-Verzeichnissen liegen, denn die sort.exe von Cygwin muss benutzt werden, nicht die von Windows. Prüfen Sie außerdem im Verzeichnis /bin, ob die bash.exe existiert - dies muss der Fall sein.
- 6. Wenn Sie Postgres nativ installiert haben, dann können Sie jetzt aufhören. Der folgende Teil gilt nur für Postgres unter Cygwin:

IPC-Daemon starten

ipc-daemon2 &

Danach ist Postgres bereits installiert. Wenn Sie Postgres selbst aus den Quellen installieren, dann gehen Sie in das Verzeichnis mit den Quellen von postgresql. Die Installationsschritte entsprechen der Linux-Installation, außer dass Sie beim configure auch —enable-odbc eingeben sollten. Wenn entsprechende Fehlermeldungen erscheinen, müssen Sie noch dafür sorgen, dass (am Beispiel einer Installation von Cygwin in c:\cygwin) C:\cygwin\usr\local\pgsql\lib\pq.dll im PATH ist.

7. Nach der Installation Cygwin neu starten; danach muss unter cygwin ein User installiert werden. Geben Sie dazu ein

Unter Win95/98/ME muss man das Passwort in /etc/passwd noch verschlüsseln; ersetzen Sie den Passus "use crypt" durch die Ausgabe von dem Befehl crypt <<Ihr Passwort>>

8. Zur Initialisierung von Postgres folgendes eingeben:

ipc-daemon2 &

initdb -D /usr/local/pgsql/data

in /usr/local/pgsql/data/postgresql.conf #tcpip_socket=false

we gnehmen und auf true setzen

Zum Start des Postmaster eine Batchdatei z.B. pgsql_start.x anlegen mit dem Inhalt:

pgsql_start.x #! /bin/sh ipc-daemon2 & pg_ctl -D /usr/local/pgsql/data -l /var/log/postgres.log -o -i start Danach gibt man ein: chmod +x pgsql_start.x ./pgsql_start.x

Der Postmaster startet dann, und die Logdatei /var/log/postgres.log wird gefüllt.

Den erfolgreichen Start von Postgres kann man prüfen, indem man psql templatel eingibt.

Den postmaster beendet man wie unter UNIX mit

```
pg_ctl stop -D /usr/local/pgsql/data
```

Die Installation (S. 20) des Kernmoduls kann danach vorgenommen werden; bei der Umgebungsvariable JAVA_HOME müssen Sie die Windows-Installation von Java verwenden (/cgydrive/<<Windows-Laufwerk>>/<<Pfad zum JDK>>).

Noch ein kleiner Hinweis: Wenn Sie sich von entfernten Rechnern auf dem Cygwin-Server anmelden wollen, müssen Sie den ssh-Daemon installieren (s.u.).

2.1.4.1.3.6 Cygwin für SuperX

Für die Modulscripte von SuperX wird die leistungsfähige Scripting-Umgebung Cygwin benötigt (unter Windows / DOS gibt es nichts Vergleichbares!). Gleichzeitig bleiben dadurch SuperX-Distributionen plattformübergreifend, durch geringe Anpassungen erreichen wir, dass Scripte unter Unix auch unter Cygwin laufen. Sie installieren also zunächst wie oben beschrieben Cygwin, allerdings <u>ohne</u> die Pakete Postgres und IPC-Daemon zu installieren. Bei nativem Windows-Betrieb muss der oben bei Cygwin genannte cygipc-Dienst nicht installiert und gestartet werden. Im Folgenden ein paar Anpassungen für die Bash unter Cygwin.

Beachten Sie, dass in der Konfigurationsdatei <code>\$SUPERX_DIR/db/bin/SQL_ENV</code> die Umgebungsvariable <code>PGHOST</code> gesetzt sein muss, und dass der Pfad für die Binaries von Postgres angepasst werden muss.

```
case $SX CLIENT in
   Auszug aus
der SQL_ENV
                  psql)
   für Cygwin
                         export PGDATESTYLE=German
 und Postgres
                         O_DESCR=$SUPERX_DIR/db/conf/unldescr_postgres_copy.xml
        (nativ)
                         export O DESCR
                         PGPORT=5432
                         export PGPORT
                         #Bei Betrieb von Postgres unter Win muss für psql
                         #unter cygwin die Umgebungsvariable PG_HOST gesetzt sein
                         #Sonst versucht er eine Socket Connection
                         PGHOST=localhost
                         export PGHOST
                         #Prüfen ob der PATH erweitert werden muss
                         PGPATH=/bin
                         case $PATH in
                         *$PGPATH*)
                                ;;
                         *)
                                export PATH=$PATH:$PGPATH
                                #echo "PATH erweitert"
                                ;;
                         esac
                 ;;
```

Hinweis für ältere SuperX-Versionen: Der alte jdbc-Treiber pgjdbc2.jar im Verzeichnis %SUPERX_DIR%\webserver\tomcat\webapps\superx\WEB-INF\lib muss gelöscht und durch den mitgelieferten Treiber pg74.214.jdbc3.jar ersetzt werden. Entsprechende Verweise in der Datei \$SUPERX_DIR/db/bin/SQL_ENV (Umgebungsvariable JDBC_CLASSPATH) müssen entsprechend geändert werden.

Wenn Sie auch einen SSH-Zugriff aus dem Rechner ermöglichen wollen (dies empfehlen wir u.a. wg. der Dateiübertragung mittels rsync), müssen Sie den SSH-Dämon unter Cygwin starten. Dazu müssen Sie zunächst eine Cygwin-Shell öffnen, und dort eingeben:

ssh-host-config

Es werden einige Dateien generiert, und außerdem werden ein paar Einstellungen abgefragt. Bei dem Fragen zum Account für den SSH-Daemon antworten Sie mit "no", d.h. der aktuelle Cygwin User startet den Dämon (dieser ist ohnehin kein Admin-User). In diesem Falle lässt sich cygwin aber nicht als Dienst einrichten.

Danach startenSie den SSH-Server mit /usr/sbin/sshd

Danach können Sie sich mit Putty auf dem Server einloggen.

2.1.4.1.3.7 Postgres-Performance-Tipps

Der Optimierer unter Postgres läßt sich uber 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.

Beim Start des Postmasters läßt sich das verfügbare RAM zuweisen mit der Option

postmaster -o "-B 128"

Dabei wird das Shared Memory von (standardmäßig) 64*8192 Bytes auf 128*8192 Bytes erhöht. Auch diese Maßnahme empfehlen wir. Generell sollte man auf Datenbankservern die Hälfte des verfügbaren physischen Rams für Postgres reservieren.

Beispielkonfiguration	in der Datei /etc/init.d/boot.local geben Sie ein:
Postgres-RAM bei DB-	echo 536870912 > /proc/sys/kernel/shmmax #512 MB RAM für PG
Server mit 1 GB RAM	echo 2097152 > /proc/sys/kernel/shmall
untor Suco Linux	echo 2 > /proc/sys/vm/overcommit_memory
unter Suse Linux	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
	<pre>max_fsm_pages = 50000</pre>
	checkpoint_segments = 6
	effective_cache_size = 65536
	Danach starten Sie Postgres neu.

Diese und weitere Perfomance-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.

2.1.4.1.4 Datenbankverbindung über einen eingeschränkten User für mehr Sicherheit

Zur Erhöhung der Sicherheit ist es möglich, dass die Datenbankverbindung von Tomcat zur Datenbank mit einem eingeschränkten User durchgeführt wird. Richten Sie dazu einen entsprechenden User in Ihrer Datenbank ein und geben Sie diesen beim Propadmin bei eingeschränkter User an. Der erste im Propadmin auszufüllende User muss weiterhin umfassende Rechte auf alle Tabellen haben, weil er auch bei Modulinstallationen/-updates verwendet wird. Das Minimum, was der eingeschränkte User haben muss sind select-Rechte auf alle Tabellen, insert-Rechte auf die Tabelle protokoll und update-Rechte auf userinfo.

Sobald Sie Ihre db.properties mit dem Propadmin bearbeitet haben, können Sie praktisch die Minimal nötigen Rechte vergeben, in dem Sie einmal das Skript

sx_restrictedconnmanager.x false aufrufen.

Nach einem Tomcat-Neustart findet sich in der catalina.out nach "Aufbau von Datenbank-

ConnectionPool (..) .. OK" ein Hinweis:

eingeschränkter Datenbankuser für Verbindung: truelfalse

Wenn Sie Funktionen wie User/Gruppe/Maske einrichten/löschen etc. im XML-Frontend benutzen wollen, müssen zusätzliche Kernmodultabellen freigeschaltet werden:

- protokoll
- userinfo

- groupinfo
- user_institution
- user_sachgeb_bez
- user_masken_bez
- group_sachgeb_bez
- group_masken_bez
- user_group_bez
- user_pw
- user_sichten
- user_sichtarten
- group-sichten
- group_sichtarten
- felderinfo
- maskeninfo
- maske_system_bez
- masken_felder_bez
- sachgeb_maske_bez
- organigramm
- themenbaum

Am einfachsten können Sie dies erledigen, indem Sie das Skript

sx_restrictedconnmanager.x true aufrufen.

2.1.4.1.5 Automatischer Start des Datenbankservers als Dienst

Nach erfolgreicher Installation des Datenbankservers muss der Server als Dienst eingerichtet werden. Wir haben das Vorgehen für die Betriebssysteme RedHat 8.0 und SuSE Linux 7.x-8.x beschrieben (für Debian ebenfalls, aber diese Scripte haben wir noch nicht getestet).

2.1.4.1.5.1 Einrichtung der Dienste

Im Kernmodul befinden sich unter /home/superx/db/etc die Vorlagen für den DB-Server. Die Ordnerstruktur entspricht dem Linux-Rechners auf oberster Ebene. Kopieren Sie die Dateien als root in die entsprechenden Verzeichnisse, z.B. bei Redhat-Linux

\$SUPERX_DIR/db/etc/init.d/superx_db.redhat

nach
/etc/init.d/superx_db
Ebenso verfahren Sie mit den Dateien in \$SUPERX_DIR/db/etc/sysconfig.
Dann machen Sie die User informix / postgres zu Eigentümern der Dateien.
Die Variablen, die ggf. angepasst werden müssen, sind
SUPERX_USER
JAVA_HOME und andere Variablen aus \$SUPERX_DIR/db/bin/SQL_ENV
(wenn Sie SuperX in einem anderen Verzeichnis als /home/\$SUPERX_USER installiert haben, müssen Sie die

Pfade zu TOMCAT_START und TOMCAT_STOP entsprechend anpassen).

Dann erzeugen Sie als root die leere Datei

/var/log/superx.log

und machen den User superx zum Eigentümer

chown superx:users /var/log/superx.log

Analog verfahren Sie mit

- /var/log/informix und machen den user informix zum Eigentümer bzw.
- /var/log/postgres und machen den user postgres zum Eigentümer

Dann

 kopieren Sie die Datei \$SUPERX_DIR/db/etc/home_informix/start.sh in das Homeverzeichnis von Informix,

und machen den user informix zum Eigentümer bzw.

• kopieren Sie die Datei \$SUPERX_DIR/db/etc/home_postgres/start.sh in das Homeverzeichnis von postgres,

und machen den user postgres zum Eigentümer.

Kontrollieren Sie, ob die Datei start.sh Ausführungsrechte besitzt.

2.1.4.1.5.2 Aktivierung der Dienste

Zur Aktivierung der Dienste für den Runlevel 3 führen Sie jeweils folgende Schritte durch; erzeugen Sie einen symbolischen Link für das Script superx-db im Runlevel 3 und 5

Redhat/Mandrake:

ln -s /etc/rc.d/init.d/superx_db /etc/rc.d/rc3.d/S90superx_db
ln -s /etc/rc.d/init.d/superx_db /etc/rc.d/rc5.d/S90superx_db
ln -s /etc/rc.d/init.d/superx_db /etc/rc.d/rc3.d/K90superx_db
ln -s /etc/rc.d/init.d/superx_db /etc/rc.d/rc5.d/K90superx_db

SuSE 8.x:

```
ln -s /etc/init.d/superx_db /etc/init.d/rc3.d/S98superx_db
ln -s /etc/init.d/superx_db /etc/init.d/rc5.d/S98superx_db
ln -s /etc/init.d/superx_db /etc/init.d/rc3.d/K98superx_db
ln -s /etc/init.d/superx_db /etc/init.d/rc5.d/K98superx_db
```

Debian,LSB:

- ln -s /etc/init.d/rc/superx_db /etc/init.d/rc3.d/S98superx_db
- ln -s /etc/init.d/rc/superx_db /etc/init.d/rc5.d/S98superx_db
- ln -s /etc/init.d/rc/superx_db /etc/init.d/rc3.d/K98superx_db
- ln -s /etc/init.d/rc/superx_db /etc/init.d/rc5.d/K98superx_db

Danach können Sie als root testen, ob die Scripte laufen, indem Sie als root /etc/init.d/superx_db start zum Starten der Datenbank ausführen, und sowie /etc/init.d/superx_db stop zum Stoppen der Datenbank.

Etwaige Fehlermeldungen stehen in Logdatei /var/log/informix.log, postgres.log bzw. superx.log

2.1.4.2 Einspielen des Kernmoduls der SuperX-Datenbank

Für die Installation haben wir eine Kurzanleitung (S. 20) 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 (S. 94). Bei erfolgreichem Ablauf kommt eine Erfolgmeldung, im Falle eines Fehlers wird die Fehler-Logdatet create.log angezeigt. Wenn ein Fehler auftritt, müssen sie die Datenbank vor einem erneuten Ablauf des Scriptes droppen.

Danach können Sie mit dbaccess superx (unter Informix) bzw. 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 (appr-Wert in cifx mit key=36).

2.1.4.3 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:

Beispieleintrag in der
crontab des users
superx# 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:

Beispiel-Updatescript	#!/bin/sh			
für SuperX•	#This is the central update script for SuperX.			
	. /home/superx/db/bin/SQL_ENV			
update.x	LOG=\$SUPERX_DIR/db/module/superx_update.log			
(Auszug)	#Stop Tomcat			
	\$SUPERX_DIR/webserver/tomcat/bin/shutdown.sh >\$LOG 2>&1			
	#Dump Database			
	\$SUPERX_DIR/db/install/dump_it.x >>\$LOG 2>&1			
	#Now the Modules are updated:			
	\$SOS_PFAD/sos_update.x >>\$LOG 2>&1			
	\$COB_PFAD/cob_update.x >>\$LOG 2>&1			

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.

2.1.4.3.1 Ein Dump unter Informix

Die Datenbank lässt sich mit dem Kommando dbexport -o <Pfad> superx exportieren und sichern. Beachten Sie aber, dass durch das Servlet eine (oder mehrere) Verbindungen zur Datenbank geöffnet ist. Deshalb muss das Servlet beendet werden oder die Datenbank muss vom User Informix einmal auf quiescent und dann wieder auf online gesetzt werden, damit eventuell noch ablaufende SuperX-Prozesse beendet werden.

2.1.4.3.2 Ein Dump unter Postgres

Postgres lässt sich auch im laufenden Betrieb sichern.

In unserem Dump-Script wird der Dump mit dem Parameter "--inserts" versehen. Dies ist eine sehr vorsichtige Einstellung, aber der Dump ist dadurch maximal kompatible 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, z.B. mit

pg_dump -f superx.sql superx

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

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

2.1.4.4 Anpassung der DB-Parameter für Clientanwendungen

Zunächst ist es wichtig, eine Verbindung vom Webserver zum Datenbankserver zu bekommen. Dazu gibt es verschiedene Werkzeuge.

Beschreibung SuperX-Kern-Modul V3.0 © Projektgruppe SuperX

2.1.4.4.1 Unter WIN32 auf den Informix-Server zugreifen: iLogin

Wenn Sie Tomcat und den Webserver auf einem WIN32-Rechner betreiben wollen, dann ist es sinnvoll, zunächst die Datenbankverbindung zu überprüfen. Um von Windows-Rechnern auf Informix-Datenbanken zugreifen zu können, muss man dem Rechner den Service bekannt machen. Dazu muss man in der Datei winnt\system32\drivers\etc\services^{vi} den Port für den Service angeben, z.B. die Zeile

```
1542/tcp
superx_server
hinzufügen.
```

Nun können Sie Parameter für den Zugang von WIN32-Rechnern auf den Datenbankserver überprüfen. Der beste Weg dafür ist das Werkzeug iLogin, das von Informix in den Client-SDKs mitgeliefert wird. Die folgende Abbildung zeigt ein Beispiel für die Parameter beim iLogin

Die Parameter sind oben	Login Parameters				
bereits erläutert. Ein erfolgreicher iLogin ist	Server	super×246			
Voraussetzung für das weitere Vorgehen!	Hostname	sg24m×-6			
	Servicename	superx_server			
	Protocolname	onsoctcp			
	Username	superx			
	Password	****			
	Stores Database	superx			
	Fill in desired values.				
	Server, Host, Service, Protocol, User and Password fields will be read from Registry if left blank.				
	Stores7 will be used if Database field is left blank.				
	<u>0</u> k	<u>C</u> ancel			

2.1.4.4.1.1 SuperX (Informix) unter Win32 als ODBC-Datenquelle einrichten

Für den regulären SuperX-Betrieb ist dieser Schritt nicht unbedingt erforderlich. Wenn Sie allerdings unter Win32 direkt auf die Datenbank zugreifen möchten, z.B. um Microsoft Access als Frontend einzusetzen, müssen Sie SuperX als ODBC-Quelle einrichten. Für die Informix-Datenbank gibt es eigene Treiber für den ODBC-Zugriff (für IDS 7.31 gibt es Intersolv 3.10 oder 3.11). Diesen Treiber muss man sich zunächst von www.informix.com besorgen. Meist sind die Treiber Teile des Informix Client SDK; für den reinen ODBC-Zugriff reicht es vollkommen aus, bei der Installation Custom zu wählen und nur den ODBC-Treiber zu installieren.

Zur Installation:

Systemsteuerung -> (Win 2000: Verwaltung)-> Datenquellen (ODBC)->System-DSN -> Hinzufügen

Für IDS 7.31 kann man als **ODBC-Treiber** z.B.den Intersolv 3.11-Treiber wählen. Dieser befindet sich im Informix Client SDK 2.40 (der 3.10-Treiber geht auch, der ist im Informix Client SDK 2.02). Für den IDS 9.21 benötigt man den Treiber Informix 3.33, der Teil des Client SDK 2.60 ist. Version 3.34 läuft ebenfalls.

Der Datenquellen-Name ist superx.

Name ECDCMusic minisuperx minisuperxSG24M× MQIS	Driver Microsoft Access Driver (*.mdb) Microsoft Access-Treiber (*.mdb) INTERSOLV 3.10 32-BIT INFORMIX 9 SQL Server	A <u>d</u> d <u>R</u> emove <u>C</u> onfigure
sg24mx spx superx_UB SuperX_MDB	INTERSOLV 3.10 32-BIT INFORMIX 9 INTERSOLV 3.10 32-BIT INFORMIX 9 INFORMIX 3.33 32 BIT INFORMIX 3.33 32 BIT Microsoft Access-Treiber (*.mdb)	
An ODB the indic on this n	C System data source stores information about ated data provider. A System data source is v achine, including NT services.	how to connect to isible to all users

User DSN System DSN File DSN Drivers Tracing Connection Pooling 4	hout l
Informix ODBC Driver Setup	
General Connection Environment Advanced About Informix ODBC	Ľ
	ΠL
Data Source Name superx	
Description	ΙΓ
- OK Abbrechen Übernehm	

Beschreibung SuperX-Kern-Modul V3.0 © Projektgruppe SuperX

Als Datenbank-	Informix ODBC Driver Setup	<u>? ×</u>
Name die SuperX-	General Connection Environment	Advanced About Informix ODBC
Datenbank ange-		
ben. Für die Ver-	Server Name	superx_246_sys3
bindung die rech-	Host Name	sys3-6
ten Parameter ein-	Service	superx_server
geben (Achtung:	Protocol	onsoctop 💌
Beispielangaben	Options	
für Duisburg);	Database Name	superx 🔻
wichtig sind der	User Id	superx
Hostname, der	Password	*****
Service-Name	Apply & Test Connection	
(s.u.) und der Ser-		
ver.	OK	Abbrechen Ü <u>b</u> ernehmen

Get-DB-List from Informix kann man deaktivieren. Manche ODBC-Treiber erlauben es in den erweiterten Optionen (Environment), eine DB-Locale zu definieren; wir empfehlen, diese auf Use Server Database Locale zu setzen.

Vorsicht Der Informix-ODBC-Treiber 3.8 darf auf keinen Fall benutzt mit ODBC werden, er ist extrem fehlerhaft und kann zum Datenverlust führen.

Beachten Sie ggf., dass diese Installation unter NT/ Win 2000 kennungsabhängig ist.

2.1.4.4.2 Einrichtung des ODBC-Treibers für den Postgres-Server

Der ODBC-Treiber für Postgres ist vom Postgres-Projekt verfügbar (www.postgresql.org). Er ist in der 8.0-Distribution von Postgres bereits enthalten. Der Treiber lässt sich leicht installieren, indem Sie in der Systemsteuerung über Verwaltung -> ODBC-Datenquellen eine Datenquelle einrichten, z.B. mit dem Namen superx.

Der Datenquellen-Name ist superx, der Datenbankname ebenfalls. Bei Server geben Sie den Hostnamen oder die IP-Nummer ein, und rechts den Port. Die Kennung ist hier z.B. superx.

PostgreSQL ODBC Driver (psqlODBC) Setup				
Data Source	superx	Descript ion		
Database	superx			
Server	localhost	Port	5432	
User Name	superx	Passuord	*****	
	Г	Opt ions		
Save	Cancel	Datasource	Global Manage DSN	
	L			

Im Dialog "Options-> Datasource" müssen einige Einstellungen vorgenommen werden:

Setzen Sie die Data Type Options wie rechts angezeigt. Das Kreuz bei Bools as Char ist notwendig, weil Access oder andere Frontends sich mit Postgres bei Binären Datentypen nicht vertragen. Boolean-Felder werden mit "1" oder "0" codiert. Bei Max Varchar geben Sie 255 ein (sonst macht Access aus allen VARCHAR(255)-Feldern Memo-Felder), und Max LongVarchar mindestens 30.000. Der Rest ist ok.

Advanced Options (superx) 1/2)				
Page 1 Page 2				
🔽 Disable Genetic Optimizer	🔲 ConnLog (C:\psqlodbc_xxxx.log)			
🔲 KSQO(Keyset Query Optimizatior	🥅 Parse Statements			
🔽 Recognize Unique Indexes	🥅 Cancel as FreeStmt (Exp)			
🔲 Use Declare/Fetch	∏ HyLog (C:\нуlog_xxxx.log)			
Unknown Sizes © Haximum C Don't Know	C Longest			
Data Type Options Text as LongVarChar Tunknowns as LongVarChar V Bools as Char				
-Miscellanous				
Max Varchar 255 Max Li	ongVarChar 30159			
Cache Size 100 SysTa	ble Prefixes dd_;			
Defaults	OK Apply Cancel			

Auf der zweiten Seite sind die Defaults korrekt.

Die Linefeed-	Advanced Options (superx) 2/2				
der Scripte in SuperX-	Page 1 Page 2				
Textfeldern notwendig.	F Read Only	🥅 Rou Versioning			
	🥅 Show System Tables	🔲 Disallou Premature			
	🔽 LF <-> CR/LF conversion	🔲 True is -1			
	🥅 Updatable Cursors (Experimental)	🔲 Server side prepare			
	🖵 bytea as LO				
Int8 As					
	● default ⊂ bigint ⊂ numeric ⊂ var	char⊙double⊙int4			
	● >= 6.4 ○ 6.3	C 6.2			
	-OID Options				
	🥅 Shou Column 🕅	Fake Index			
	Connect Settings				
	OK	Apply Cancel			

2.1.4.4.3 Anbindung des Access-Frontends an die ODBC-Quelle

Wenn Sie die SuperX-Datenbank als ODBC-Quelle unter dem Namen superx eingerichtet haben, dann können Sie das im SuperX-Clientpaket (S. 90) unter \$SUPERX_DIR/tools/access/superx_frontend_sam.mdb enthaltene Access-2000-Frontend benutzen. Bei der Datei handelt es sich um ein Muster, vor dem ersten Gebrauch kopieren Sie sie bitte nach \$SUPERX_DIR/db/superx_frontend.mdb und arbeiten Sie nur mit letzterer Datei - so können Sie sichergehen, dass Ihr Access-Frontend nicht bei einem SuperX-Update überschrieben wird.

Beim ersten Öffnen der Datei sind die Tabellen noch nicht verknüpft. Sie müssen zunächst Das Formular Setup aufrufen, den Namen der ODBC-Quelle (s.o.) eintragen, und "Erzeuge Kernmodul-Verknüpfungen" drücken. Wenn der Informix-Treiber dies unterstützt, sollte vorher die Option "Passwort speichern" aktiviert werden, ansonsten muss man für jede Tabellenverknüpfung das Passwort eingeben.

Die Datenquelle wird eingegeben, und die Kernmodul-Tabellen können so verknüpft werden. Die Verknüpfungen haben nach Access-Voreinstellung den Namen "superx_tabellenname" und werden automatisch umbenannt zu "tabellenname".

1.	ormular			_	. [C
· [1		l l		· ·
Date	enbanl u Iquellen-Name (DSN)	k-Verbin ber ODBC Superx Erzeuge Kernme Verknüpfunge	dung Jadul-	Millkommen zumSetup des Access-Frontend der SuperX- Datenbank. Die müssen die SuperX- Datenbank in der Systemsteuerung als ODBC- Quelle einbinden. Tragen Sie dann den Namen der Datenquelle links ein, und	50°
N.				drücken Sie den linken Button. Die Verknüpfungen zu den Kernmodul-Tabellen werden Jann erzeugt.	-
Achtung: Duro Datenbank ni beenden. Die Setun-Bo	ch die Schwäc cht sehr stabil utinen wurden	hen des ODBC-Tr . Manchmal läßt sir umit dem Informix-1	eibers ist die Ve ch die Anwendu Freiber 3 33 und	rbindung zur Informix- ng nur durch den Taskmanager 3.4 sowie mit allen Postores-	
Die Gelup Fil	uunen wurden		rtenser 5.55 unu rtioniort dofinitiu	nicht Bei Problemen mit	

Falls der Setup so nicht funktioniert, müssen die Tabellen "von Hand" verknüpft und umbenannt werden. Die Funktionalität des Access-Frontends ist dadurch nicht beeinträchtigt. Bei Tabellen ohne Primary Key muss allerdings ein eindeutiger Datensatzbezeichner angegeben werden, sonst ist die Tabelle schreibgeschützt.

Bitte beachten Sie, dass die ODBC-Treiber von Informix recht instabil sind und die Anwendung sich manchmal nur durch den Taskmanager beenden läßt. Wir mussten bei Access 2000 und 2002 feststellen, dass einige Formulare nach einiger Zeit nicht mehr geschlossen werden können, und Visual-Basic-Routinen mit der ominösen Fehlermeldung "Dieser Vorgang wird den aktuellen Code in den Unterbrechungsmodus zurücksetzen" beenden. Dieser Fehler ist bei Microsoft dokumentiert, aber die vorgeschlagene Lösung hat bei uns nicht funktioniert^{vii}. Eine funktionierende Lösung fanden wir in Access-Foren^{viii}.

2.1.4.4.4 Anpassen der Datenbankparameter für das SuperX-Servlet

Wenn Sie die Verfügbarkeit des Datenbankservers getestet haben (z.B. über das Utility iLogin von Informix), dann müssen die Datenbankparameter (S. 21) 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 Shellscript (S. 96) 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-Client auf eine andere Datenbank zugreifen soll. Starten Sie das Tool von einer Konsole bzw. Eingabeaufforderung das Tool mit dem Befehl propadmin.bat

bzw.

propadmin.x (unter Linux).

Füllen Sie die Felder entsprechend des folgenden Beispiels:

Nehmen wir z.B. für
Informix die Parameter
beim ILogin (oben er-
läutert). Wenn die rech-
te Abbildung eine kor-
rekte Einstellung an-
zeigt,

Login Parameters		×		
Server	super×246			
Hostname	sg24mx-6			
Servicename	superx_server			
Protocolname	onsoctcp			
Username	superx			
Password	****			
Stores Database	superx			
Fill in desired values.				
Server, Host, Service, Protocol, User and Password fields will be read from Registry if left blank.				
Stores7 will be used if Database field is left blank.				
<u>O</u> K <u>C</u> ancel				

...dann übernehmen Sie die Parameter wie folgt für die db.properties:

📉 DB-Properties Adm	in 3.0 🎱			_ O X
DB-Properties	s Admin für /home/	superx/webserver/tomc	at/webapps/super	x/WEB-INF/db.properties
Driver :	PostgreSQL	🔹 (mögliche Dat	enbanksysteme für S	SuperX)
Driver Class:	org.postgresql.Driver		(muss im CLAS	SSPATH stehen)
Connection URL:	jdbc:postgresql://loca	lhost/superx		
Username:	superx P	assword: *******	Verbindung test	en
Eingeschränkter Us	er: security	Passwort	Verb.testen	select-Rechte auf alle Tabellen
Log Level SQL FINEST	•			
Log Level XML FINEST	•			
Entwicklungsmodus Øm Entwicklungsmodu Das dauert etwas länge	● an ○ aus Is werden alle SQL-Be er, ermöglicht aber be	fehle von Abfragen einzeln ssere Fehlermeldungen.)	an die Datenbank g	eschickt.
Masken, die im Cache s	sein sollen select tid fr	rom maskeninfo where 1=0		
User, die im Cache seir	n sollen select tid from	userinfo where 1=0		
Der Apache ConnectionPool verwaltet die Anzahl benötigter Verbindungen dynamisch. min/max idle gibt an wieviele Connections ständig bereit gehalten werden sollen. maxActive gibt an wieviele Connections maximal gleichzeitig aktiv sein sollen.				
minIdle 5 maxIdle maxActive				
		Speichern		

Der Port 1542 ergibt sich aus dem Service für SuperX, der oben bereits beschrieben wurde. 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:<<Portnumer>>/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 (S. 135).

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 (S. 136) ändern.

Die Parameter *im Cache* legen fest, wie viel Information gecached werden 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 LATIN1). 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 Datenbankserver 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.)

2.1.4.4.5 Datenbankverbindung und Steuerung von DBForms

Neben der normalen Properties-Konfiguration muss außerdem der Verbindungsparameter für die Servlets von DBFORMS gesetzt werden.

Die zentrale Steuerungsdatei heißt dbforms-config.xml und liegt im Verzeichnis

SUPERX_DIR/webserver/tomcat/webapps/superx/WEB-INF. Dort liegt bereits ein Muster mit dem Namen kern_dbforms-config_pg.xml für Postgres bzw. kern_dbforms-config_ids.xml für Informix. Diese Datei wird bei der Installation automatisch kopiert nach dbforms-config.xml.

Die Datenbankverbindung wird in der server.xml konfiguriert. und am Ende der Datei die Connection-Attribute angeben. Die Parameter sind identisch mit denen, die Sie in der db.properties (S. 45) angeben.

Detaillierte Dokumentation zum Connection Logging sowie allgemein zu DBForms (leider nur in Englisch) finden Sie im beigefügten DBForms-Handbuch im Verzeichnis doc/dbforms des Kernmoduls.

Wenn Sie die DBFORMS-Komponente nicht benötigen bzw. aus Sicherheitsgründen für eine externe Website abschalten wollen, gehen Sie wie folgt vor:

Aktion	Code
Sperren Sie das dbforms-Servlet in der	
Datei	===================================</td
<pre>\$SUPERX_DIR/webserver/tomcat</pre>	<servlet
/webapps/superx/WEB-	<servlet-name>control</servlet-name>
INF/web.xml, indem Sie die rechts blau	<servlet-class>org.dbforms.servlets.Controller</servlet-class>
markierten Kommentarzeichen um die	
entsprechenden Elemente setzen.	<init-param></init-param>
	<pre><param-name>maxUploadSize</param-name></pre>
	<pre><param-value>80000</param-value></pre>
	>
	=== DbForms FileServlet ====================================</td
	>
	<servlet

	<servlet-name>file</servlet-name>
	<servlet-class>org.dbforms.servlets.FileServlet</servlet-class>
	<load-on-startup>2</load-on-startup>
	>
	=== Controller Servlet and FileServlet Mappings======-
	<servlet-mapping
	<servlet-name>control</servlet-name>
	<url-pattern>/servlet/control</url-pattern>
	<servlet-mapping></servlet-mapping>
	<servlet-name>file</servlet-name>
	<url-pattern>/servlet/file</url-pattern>
	>
Fügen Sie an das Ende der web.xml	
vor dem End-Tag "" folgende	<error-page></error-page>
Elemente ein	<error-code>500</error-code>
	<location>/error.htm</location>
Ändern Sie am Ende der Datei	
SSUPERY DIR/webserver/tomcat	
/webapps/superx/WEB-	
INE/dbforms_config yml heim	<dbconnection <="" id="superx" isjndi="true" td=""></dbconnection>
Element dhconnection den Attributnamen	name="java:/comp/env/jdbc/superx1"/>
"name" auf einen nicht existenten Namen	
z B "superv1"	
Starten Sie Tomcat neu, und prüfen Sie	
in der Logdatei	
\$SUPERX_DIR/webserver/tomcat	
/logs/catalina.out, ob der Tom-	
cat-Start erfolgreich war.	

Durch diese Maßnahme sind der DBFORMS-Komponente keine Datenbankverbindungen mehr möglich, und das Ausspähen geschützter Dateien in Tomcat-Systemverzeichnissen durch das Control-Servlet ist nicht mehr möglich.

Eine Abschaltung der DBFORMS beeinträchtigt in keiner Weise die "normalen" Funktionen zur Berichtserstellung von SuperX.

2.1.5 Installation und Pflege des Webservers

Die Servlet-Engine Tomcat verfügt zwar über einen kleinen "eingebauten" Webserver, doch für den Echtbetrieb sollte man aus Performance-Gründen einen der marktgängigen Webserver nutzen (z.B. Apache, IIS), der auch Verschlüsselung bietet. Für den Echtbetrieb empfehlen wird die Installation eines Apache 1.3.x auf Linux-Basis – meist ist dieser in der Linux-Distribution bereits integriert. Der Apache läßt sich sehr gut mit dem Tomcat verbinden (siehe Tomcat User's Guide im Kernmodul unter \$SUPERX_DIR/doc/tomcat/doc). Bei der Linux-Installation gehen wir davon aus, dass alle Maßnahmen unter der Kennung superx erfolgen, und dass der User superx Zugriffsrechte auf die Datenbank hat. Beim Kopieren des Archivs sollten Sie darauf achten, dass der User superx auf die Scriptdateien Ausführungsrechte besitzt.

Die folgenden Anleitungen gehen davon aus, dass Sie als Installationspfade für den Webserver 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). Aus diesem Grunde haben wir ins Stammverzeichnis des webservers eine html-Datei erstellt (

\$SUPERX_DIR/webserver/index.htm), von der aus die Parameter und Pfade schrittweise überprüft werden können.

2.1.5.1 Installation von Java und Datenbanktreibern

Der Webserver muss Java-fähig sein, damit er Servlets ausführen kann.

- Arbeitsschritte:
- 1. Java (S. 12) Development Kit JDK 1.4.x installieren
 - am besten alles ins Verzeichnis c:\Programme\jdk1.4.x bzw. /usr/local/jdk1.4.x
- 2. Die Umgebungsvariable JAVA_HOME (S. 54) setzen und das bin-Verzeichnis der Java-Installation in den PATH legen. Die Umgebungsvariable CLASSPATH sollte mindestens "." enthalten, aber auf keinen Fall einen älteren XML-Parser (z.B. xerces 1.0).
- 3. Nur für Informix-Anwender: Laden Sie den jdbc-Treiber von Informix (oder das Informix Client SDK) herunter, installieren Sie das Produkt und kopieren Sie die Datei ifxjdbc.jar nach \$SUPERX_DIR/webserver/tomcat/webapps/superx/WEB-INF/lib.
- 4. Nur für Unix/Linux: Auf Betriebssystemebene muss sichergestellt werden, dass das Dateisystem Textdateien im ISO-Format anlegt (und nicht UTF-8). Bei RedHat-Linux z.B. wird die Umgebungsvariable \$LANG auf diesen Wert gesetzt. Stattdessen muss \$LANG auf de_DE.ISO-8859-1 oder de_DE@euro (installierte Locales finden Sie unter Linux im Verzeichnis /usr/lib/locale) stehen.
- 5. Testen Sie die Datenbankverbindung mit dem Werkzeug propadmin (S. 45).
- 6. Ablauf mit einem einfachen java-Servlet testen

2.1.5.2 Einrichtung der Servlet-Engine

Die Servlet-Engine ermöglicht dem Webserver 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.

Es gibt verschiedene Anbieter von Servlet-Engines, teilweise kostenpflichtig, z.B.

- Unify ServletExec
- Allaires Jrun
- Apaches JServ
- Apache / Jakartas Tomcat

Im Kernmodul ist der Tomcat 4.1.27 mitgeliefert. Tomcat ist von SUN 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 der 3er^{ix}, 4er und 5er-Generation (5.5.x wurde noch nicht getestet).

Tomcat ist eine kostenlose und gleichzeitig umfassende Engine, die darüber hinaus auch recht leicht zu installieren ist und auf vielen gängigen Webservern läuft (Apache^x, IIs, Netscape). Sie ist im Rahmen des Apache-Projektes frei verfügbar und distribuierbar, deshalb ist Tomcat in der SuperX-Distribution bereits enthalten. Wenn Sie das Kernmodul (S. 14) entpacken, ist Tomcat mitsamt dem SuperX-Kontext bereits installiert. Sie müssen nur noch ein paar Schritte durchführen.

2.1.5.2.1 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 (S. 68) dokumentiert.

Die Datenbank-Verbindung für DBFORMS wird in der server.xml definiert. Diese sähe dann z.B. für Postgres auf dem Rechner "miles" auf Port 5432 so aus:

Der SuperX-Kontext in der server.xml

```
<Context path="/superx" docBase="superx" debug="0"
                 reloadable="true" crossContext="true">
          <Logger className="org.apache.catalina.logger.FileLogger"
                     prefix="localhost_superx_log." suffix=".txt"
              timestamp="true"/>
         <Environment name="maxExemptions" type="java.lang.Integer"</pre>
                      value="15"/>
          <Parameter name="context.param.name" value="context.param.value"
                     override="false"/>
          <Resource name="jdbc/superx" auth="Container"
          type="javax.sql.DataSource"/>
<ResourceParams name="jdbc/superx">
  <parameter>
    <name>factory</name>
    <value>org.apache.commons.dbcp.BasicDataSourceFactory</value>
  </parameter>
  <parameter>
    <name>driverClassName</name>
    <value>org.postgresql.Driver</value>
  </parameter>
  <parameter>
    <name>url</name>
    <value>jdbc:postgresql://localhost/superx</value>
  </parameter>
  <parameter>
    <name>username</name>
    <value>superx</value>
  </parameter>
  <parameter>
    <name>password</name>
    <value>anfang12</value>
  </parameter>
  <parameter>
    <name>maxActive</name>
    <value>7</value>
  </parameter>
  <parameter>
    <name>maxIdle</name>
    <value>5</value>
  </parameter>
  <parameter>
    <name>maxWait</name>
    <value>-1</value>
  </parameter>
  <parameter>
<name>removeAbandoned</name>
<value>true</value>
 </parameter>
<parameter>
<name>removeAbandonedTimeout</name>
<value>10</value>
</parameter>
</ResourceParams>
</Context>
```

Die von Ihnen anzupassendne Werte sind jeweils **fett** gedruckt. Leider verlangt **dbforms**, dass das Passwort im Klartext eingegeben wird; achten Sie daher darauf, dass die Datei nur für den User selbst lesbar ist. Alternativ können Sie (zumindest bei Postgres) über die hba_conf steuern, wie der User sich ohne Passwort authentifizieren kann.

2.1.5.2.2 Administrator und Manager

Die Voreinstellungen in Tomcat 4 sind für einen ersten Testbetrieb bereits vorbereitet. Bearbeiten Sie lediglich die Datei conf\tomcat-users.xml

Im folgenden Beispiel wird der User superx mit dem Passwort "anfang12" als Admin und als Manager eingetragen.

Tomcat Users:<?xml version='1.0' encoding='utf-8'?>Administrator und
Manager<toncat-users>
<role rolename="tomcat"/>
<role rolename="tomcat"/>
<role rolename="role1"/>
<role rolename="manager"/>
<role rolename="admin"/>
<user username="tomcat" password="tomcat" roles="tomcat"/>
<user username="role1" password="tomcat" roles="tomcat"/>
<user username="superx" password="tomcat" roles="role1"/>
<user username="superx" password="admin12" ro-
les="tomcat, admin, manager"/>

</tomcat-users>

Natürlich ist dieses nur ein Beispiel für eine erste Testimplementation, nicht für einen produktiven Server geeignet.

2.1.5.2.3 Einrichten des SuperX-Servlets unter Tomcat

Anpassen der Datei **db.properties** (S. 45) mit den Datenbank-Zugangsdaten (siehe **Anpassen der Da-tenbankparameter für das SuperX-Servlet** (S. 45))

Die web.xml

Schließlich muss man in der Datei

\$SUPERX_DIR/webserver/tomcat/webapps/superx/web-inf/web.xml

das Servlet mit entsprechenden Parametern eintragen. Für den Zugriff auf die

Superx-Datenbank müsste der Eintrag wie folgt aussehen.

Auszug aus der Web.xml Der Parameter max_rows ganz amEnde legt fest, wie viele Datensätze ein Servlet maximal an den Client ausliefert. Wenn Ihr Organigramm z.B. mehr als 3000 Sätze enthält, dann sollten Sie diesen Wert höher setzen.

Ein weiterer Parameter für die gesamte Webapplikation, der aber nur im XML-Frontend ausgewertet wird, lautet <session-timeout> (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.

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

Fügen Sie an das Ende	
der web.xml vor dem	<error-page></error-page>
End-Tag ''''	<error-code>500</error-code>
z.B. folgende Elemente	<location>/error.htm</location>
em	

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 Applikationsservers.

2.1.5.2.4 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

• Unter WIN32:

•

Das geht unter MS-DOS als Kommandozeile set JAVA_HOME=c:\jdk1.4x oder man macht einen Eintrag als Systemvariable (Systemsteuerung – System – Erweitert – Umgebungsvariablen) neue Systemvariable JAVA_HOME, Wert c:\jdk1.4x

(wenn nur die Runtime installiert ist, ist das Verzeichnis evtl. c:\programme\javasoft\jre\1.4x)

- <tomcat-Basisverzeichnis>\bin\startup.bat ausführen (zum Beenden shutdown.bat)
- Falls unter Windows 98/ME eine Meldung kommt, dass der Umgebungsspeicher nicht ausreicht, muss man über start->Ausführen folgende Zeile eingeben: command.com /p /e:4096
 - Unter UNIX / LINUX: Setzen Sie entweder in der /etc/profile oder in der Datei .profile bzw. .bashrc im Heimverzeichnis des Users superx bzw. bei Betrieb von Datebank und Webserver auf einem Rechner in der Datei \$SUPERX_DIR/db/bin/SQL_ENV die Zeile ein:

export JAVA_HOME=/usr/lib/java (als Beispiel für eine Java-Installation unter SUSE Linux 9.1)

• 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
- <tomcat-Basisverzeichnis>\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 ... aufrufen.

- 1. Testen, ob der SuperX-Kontext unter Tomcat verfügbar ist: http://localhost:8080/superx/
- 2. Testen, ob Sie sich auf der SuperX-Datenbank anmelden können http://localhost:8080/superx/xml/
- 3. Testen, ob das Applet läuft http://localhost:8080/superx/applet/

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

2.1.5.2.5 Die Übertragung der Web Application

Wenn Sie die SuperX-Webapplikation auf einem vorhandenen Tomcat installieren wollen, müssen Sie alle Libraries (*.jar) von der SuperX-Distribution kopieren und ältere Versionen, die bereits vorhanden sind, löschen (Wichtig!). Bitte beachten Sie, dass Sie beim Einsatz eines eigenen Tomcat Bibliotheken aus "unserem" Tomcat nachinstallieren müssen, die sich um Verzeichnis \$CATALINA_HOME/common/endorsed/

befinden. Außerdem dürfen Sie auf dem Datenbankserver nicht "unseren" Tomcat löschen, selbst wenn er nicht gebraucht wird: Die Java-Bibliotheken und die properties-Dateien werden auch von Scripten auf dem Datenbankserver benötigt.

Sie kopieren nun das gesamte Verzeichnis \$SUPERX_DIR/webserver/tomcat/webapps/superx in das webapps-Verzeichnis des Tomcat 4. Dann müssen Sie z.B. nach web_tomcat3.xml sichern, und die Datei web_tomcat4.xml.sam nach web.xml kopieren. Gegebenenfalls müssen Sie dann Steuerungsparameter (S. 53) in der Web-Application in der Datei WEB-INF/web.xml prüfen (z.B. maxRows oder das sql- bzw. connection-Logging, Session-Timeout).

Als die SuperX-Webapplikation entwickelt wurde, war Tomcat noch in einem recht "jungen" Entwicklungsstadium, so dass die SuperX-Webapplikation einige Funktionalitäten manuell umgesetzt hat, z.B. Authentifizierung, Logging und Connection Pooling. Deshalb können derzeit die z.B. in Tomcat 4 und 5 genutzten Funktionalitäten nicht ohne weiteres eingesetzt werden. In einer zukünftigen Version werden wir die neueren Funktionen aus Tomcat 5 (z.B. Balancing, LDAP-Anbindung) direkt übernehmen und die alten SuperX-Mechanismen ablösen. In der aktuellen Version ist dies allerdings noch nicht der Fall.

Der Vorteil ist allerdings, dass SuperX kompatibel mit allen Web-Application-Servern ist, die die Servletspezifikation 2.2 oder höher umsetzen.

2.1.5.2.5.1 Übertragung der Webapplikation auf einen vorhandenen Tomcat unter Windows

Der Betrieb von Tomcat 4.x-5.x unter Windows ist grundsätzlich möglich. Wenn Tomcat unter **cygwin** installiert wird, entsprechen alle Schritte dem obigen Vorgehen unter Linux. Wenn Tomcat aus dem exe-Installer als Dienst installiert wird, dann müssen zwei Unterschiede beachtet werden:

- Die Standardausgabe von Tomcat geht nicht nach logs\catalina.out, sondern stdout_<<Datum>>.log bzw. stderr_<<Datum>>.log.
- Der Pfad zur Logging-Datei für DBFORMS wird in der Datei WEB-INF/log4j.properties festgelegt. Hier wird die Pfadangabe nicht relativ zum Statup-Verzeichnis von Tomcat gegeben, sondern absolut, z.B.

log4j.appender.logFile.File=C:/tmp/dbforms.log

2.1.5.2.5.2 Übertragung der Webapplikation auf einen vorhandenen Tomcat 5.5

Die Übertragung der Webanwendung auf die Referenzimplementation Tomcat 5.5 bringt einige Neuerungen mit sich, die für die Funktion beachtet werden müssen (alle Verzeicnisangaben relativ zur Installationsverzeichnis von Tomcat):

- Die Webapplikation (S. 55) selbst kann wie gewohnt in webapps abgelegt werden
- Als Muster für die web.xml muss die Datei WEB-INF/web_tomcat55.xml.sam nach web.xml umbenannt und ggf. angepasst werden
- Die Einrichtung der JNDI-Datenquelle für DBFORMS geschieht nicht mehr in der conf/server.xml, sondern in der Datei webapps/superx/META-INF/context.xml, ein Beispiel liegt im gleichen Verzeichnis in der Datei context.xml.sam.

2.1.5.2.6 Das SuperXManager-Servlet

Mit dem SuperXManager-Servlet kann man verschiedene Einstellungen vornehmen. Es kann von Admins aufgerufen werden unter der Adresse: http://rechnername:port/superx/servlet/SuperXManager

Server-Cache

SuperX cacht zur Performanceverbesserungen einige Dinge im Webserver, dazu gehören Erläuterungen und Übersetzungen und für's XML-Frontend auch: User,Userrechte und Sichten und auch Abfragen wenn in der db.properties eingetragen.

Falls Sie bei Entwicklungsarbeiten Änderungen an diesen Dingen gemacht haben und im XML-Frontend arbeiten, müssen Sie einmal den "Server-Cache aktualisieren". Hinweis: Neue Sichten können z.B. auch durch ein Update der COB-Daten erfolgen, wenn neue alternative Hierarchien dazukommen.

Entwicklungsmodus

Im Entwicklungsmodus werden alle SQL-Befehle von Abfragen einzeln an die Datenbank geschickt. Das dauert länger, ermöglicht aber bessere Fehlermeldungen. Die Standardeinstellung ist in den db.properties hinterlegt. Sie kann hier bei Entwicklungsarbeiten umgestellt werden.

Mit Logdateien leeren können Sie die superx_*.log-Dateien im Tomcat/logs-Verzeichnis leeren. Dies kann bei Entwicklungsarbeiten praktisch sein, wenn Sie nur die Logs eines bestimmten Vorgangs haben möchten. Außerdem werden noch verschiedene SQL/XML-Loginformationen für Entwicklungszwecke angezeigt.

2.1.5.2.7 Verbesserung der Performance

Die Tomcat-Performance läßt sich durch Zuweisung von mehr RAM steigern. Dazu muss lediglich die Umgebungsvariable JAVA_OPTS gesetzt werden, z.B. mit JAVA_OPTS= "-Xmx300M -Djava.awt.headless=true" export JAVA_OPTS

Hierdurch werden 300 MB RAM dem Tomcat zugewiesen. Die Umgebungsvariable wird außerdem auch von diversen SuperX-jdbc-Clients berücksichtigt. Dies ist z.B. sinnvoll, wenn größere Tabellen beoder entladen werden. Der Passus " -Djava.awt.headless=true" muss immer dabei sein, wenn Tomcat auf einem UNIX-System ohne graphische Konsole aus aufgerufen wird.

Die Performance von Tomcat läßt sich weiterhin durch den Lastausgleich in Kombination mit dem Apache Webserver steigern. Beim Tomcat 3.2.x Die Konfiguration wird in der Datei conf/workers.properties vorgenommen.

Die bereits vorhandenen Beispieleinträge sollten die Konfiguration des Lastausgleich erläutern. Weitere Details zur workers.properties finden Sie in der Anleitung zur Anbindung an den Apache (S. 67). Die verschiedenen Howtos in der Tomcat-Distribution erläutern Details zur Apache-Anbindung.

2.1.5.2.8 Einrichtung einer SSL-Verbindung in Tomcat

Wenn Tomcat mit einer SSL-Verschlüsselung arbeiten soll, dann sollte von vornherein das JDK 1.4.x installiert werden, weil dies die notwendigen Bibliotheken dazu enthält. Eine bestehende JDK-1.3.Installation kann aber genutzt werden, wenn die Verschlüsselungs-Klassen von Java (jsse.jar, jnet.jar,jcert.jar) nach kopiert werden. Es gibt ferner die Möglichkeit, die Verschlüsselung vom Webserver (Apache oder IIS) vornehmen zu lassen (siehe Tomcat-Dokumentation in tomcat-docs/tomcat-sslhowto.htm); da SuperX aber kaum statische html-Seiten enthält, empfehlen wir immer,die Verschlüsselung direkt in Tomcat einzurichten. Dies ist relativ einfach, wenn kein öffentlich bekanntes und signiertes Zertifikat genutzt werden soll.

Erzeugen Sie auf dem Webserver einen Keystore mit dem Befehl

Windows:%JAVA_HOME%\bin\keytool -genkey -alias tomcat -keyalg RSAUNIX:\$JAVA_HOME/bin/keytool -genkey -alias tomcat -keyalg RSA

Die Parameter werden erfragt; wichtig ist, dass der erste Eintrag (Vor- und Nachname, COMMON NAME CN) der DNS-Name des Werbservers ist, z.B. superx.verwaltung.uni-duisburg.de. Als Passwort geben Sie beide Male "changeit" an. Draufhin wird ein Zertifikat erzeugt und in der Datei .keystore im Homeverzeichnis des Benutzers angelegt (unter Windows im Profiles-Verzeichnis, unter UNIX im home-Verzeichnis).

Beschreibung SuperX-Kern-Modul V3.0 © Projektgruppe SuperX Das persönliche Zertifikat können Sie durch einen kommerziellen Zertifizierungsserver publizieren; zu Testzwecken können Sie auch ein selbsterstelltes Zertifikat erzeugen:

keytool -selfcert -alias tomcat -validity <<Anzahl der Tage>>

Danach ändern Sie die Datei stomCAT_HOME/conf/server.xml, indem Sie die Passage mit der SSL-Verschlüsselung ent-kommentieren und den normalen Port (8080) auskommentieren. Danach ist das Servlet über https://localhost:8443 statt http://localhost:8080 erreichbar. Sie müssen alle Links entsprechend ändern und in der Datei superx.properties die Zeile

superx.properties SxServerURL=https://localhost:8443/superx/servlet/SuperXDBServlet mit ssl statt SxServerURL=http://localhost:8080/superx/servlet/SuperXDBServlet aktivieren.

Das Zertifikat können Sie löschen, indem Sie auf der Kommandozeile eingeben: keytool -delete -alias tomcat

2.1.5.2.8.1 Signierung eines Zertifikats in Tomcat

Bei selbst signierten Zeritfikaten erscheint im Browser immer eine Sicherheitswarnung. Um dies zu vermeiden, muss man ein öffentliches Zertifikat von einem Trust Center erwerben. Dies kann man im Apache eintragn (s.u.), aber auch direkt im Tomcat, wenn Sie keinen Apache nutzen^{xii}:

- public key + private key erzeugen und die im keystore-file ablegen (der private key wird dabei mit passwd verschlüsselt): keytool -genkey -keyalg RSA -alias tomcat -keystore xxx.jks
- 2. certificate request generieren --> Datei server.csr und an die CA schicken: keytool -certreq -keyalg RSA -file server.csr -keystore xxx.jks
- 3. Den von der CA signierten public key = Serverzertifikat zurückbekommen --> Datei server.cer
- 4. Zuerst das Zertifikat der CA (z.B. UTN-USERFirst-Network Applications, http://www.usertrust.com) downloaden und in den keystore einspielen: keytool -import -file UTN.cer -alias tomcat -keystore xxx.jks
- 5. Dann das neue Serverzertifikat in den keystore einspielen: keytool -import -file server.cer -alias tomcat -keystore xxx.jks
- 6. in der Tomcat-server.xml auf die keystore-Datei verweisen: keystoreFile="<<Pfad zur xxx.jks-Datei>>" keystorePass="passwd" Wichtig: dasselbe passwd einsetzen wie unter 1. zum Verschlüsseln des private key benutzt wurde
- 7. Restart Tomcat
- 8. https-Verbindung zum Server, Zertifikat überprüfen vertrauenswürdig?

2.1.5.2.9 Zusätzliche Verschlüsselung im Applet durch Public-Private-Key-Kontrolle

Zur Erhöhung der Sicherheit im **SuperX-Applet** ist es möglich, eine DSA-public/private-Key-Kontrolle zu installieren. Dabei wird jeder SQL-Befehl, der vom Applet ans Servlet geschickt wird, mit dem einen Key signiert und im Servlet wird mit Hilfe des anderen, nur dort bekannten Keys kontrolliert, ob der ankommende SQL eine gültige Signatur aufweist.

Zur Installation eines zufällig erzeugten Key-Paars brauchen Sie auf dem Datenbankserver in der Shell nur die SQL_ENV aufzurufen und anschließend das Kommando

sx_keymanager.x install abzuschicken. Mit sx_keymanager.x delete könnten Sie ggfs. das Schlüsselpaar wieder entfernen und mit sx_keymanager.x check prüfen, ob ein Schlüsselpaar installiert ist. Wenn Sie Tomcat auf einem separaten Rechner betreiben, brauchen Sie hier kein Script ausführen, es recht, dort das jeweilige Kernmodul-Paket zu entpacken. Bei mandantenfähigen Installationen müssen Sie das Script sx_keymanager.x install für jeden Mandanten einzeln ausführen.

Wenn Sie Tomcat neu starten, können Sie in den Logdateien (normalerweise

\$SUPERX_DIR/webserver/tomcat/logs/catalina.out) kontrollieren, ob die public/private key Kontrolle aktiv ist oder nicht.

Nach der Meldung zum Aufbau des Datenbank-Connectionpools kommt ein einsprechender Hinweis. Aufbau des ConnectionPool (....) .. OK public/private key aktiv

Im SuperX-Applet können Sie den Info-Button anklicken, in der erscheinenden Infobox wird angegeben, ob public/private key Kontrolle aktiv ist oder nicht.

2.1.5.2.10 Tomcat als Dienst unter Linux

Die Implementation von Tomcat als Dienst ist unverzichtbar, damit der Serve rbeim Hochfahren automatisch startet. Wir haben Konfigurationsscripte und Startscripte mitgeliefert, die Sie recht leicht anpassen können.

Im Verzeichnis \$SUPERX_DIR/webserver/etc befinden sich Musterdateien, um einen Dienst unter SUSE oder RedHat Linux daraus zu machen. Kopieren Sie die Inhalte des Verzeichnisses etc als root auf den Webserver ins Verzeichnis /etc, und passen Sie /etc/sysconfig/superx_webserver entsprechend Ihrer Umgebung an. Schließlich muss ein symbolischer Link von /etc/init.d/superx_webserver nach (usr)/bin/rcsuperx_webserver gelegt werden.:

```
ln -symbolic /etc/init.d/superx_webserver /bin/rcsuperx_webserver
```

Danach kann man den Dienst im Runlevel-Editor des YAST aktivieren (Runlevel 3 und 5). Der Dienst muss vor dem Webserver, aber nach dem Start des Datenbankservers gestartet werden. Der Dienst selbst wird vom User superx gestartet, und kann jederzeit mit

rcsuperx_webserver restart

neu gestartet werden.

Unter RedHat Linux gibt es ebenfalls Werkzeuge für die Einrichtung der Runlevel, ggf. kann man auch manuell symbolische Links einrichten, wie beim Start des Datenbankservers beschrieben (S. 36). Außerdem muss ggf. die Umgebung vor dem Start des Tomcat geladen werden, z.B. durch Aufruf der SQL_ENV. Wichtig ist, dass beim Start des Tomcat als Dienst die Variable JAVA_HOME korrekt gesetzt ist und die Variable LANG auf eine deutsche Locale zeigt. Letzteres ist bei RedHat nicht standard-mäßig vorgesehen.

Die Einrichtung des Tomcat als Dienst ist auch für Windows-Server möglich, wie im folgenden gezeigt wird.

2.1.5.2.11 Tomcat als Dienst unter Windows einrichten (nur WINNT/2000 und Tomcat 3.x)

Tomcat muss auf Windown-NT/2000-Rechnern nicht in einer DOS-Box laufen, sondern kann auch als Dienst laufen. Die Installer von Tomcat 4 sehen unter Windows NT /2000/XP eine Installation als Dienst vor. Nur beim Tomcat 3.2.2 muss der Dienst manuell eingerichtet werden.

Unter NT 4 läuft der Tomcat-Dienst nur mit dem JDK 1.2.2, unter Win2000 sollte man wegen eines Bugs im Java SDK 1.3.0 das Java SDK 1.3.1 oder höher installieren. Die Variable JAVA_HOME zeigt dann auf dieses Verzeichnis. Für die Einrichtung muss man bei Windows folgendes machen:

- Man fügt der Computerverwaltung Tomcat als Dienst hinzu, indem man in einer DosBox vom <tomcat>/bin-Verzeichnis aus "jk_nt_service -I tomcat ...\conf\wrapper.properties" ausführt.
- Dann kann man den Dienst über die Systemsteuerung -> Dienste starten (besser: auf automatisch setzen), und theoretisch läuft Tomcat auch dann, wenn kein User auf dem Rechner angemeldet ist. Aus der DOS-Box kann man den Dienst auch mit net start tomcat starten.
- Die Deinstallation des Dienstes erfolgt über jk_nt_service -R tomcat

Der Dienst wird in der Systemsteuerung des Rechners aufgeführt, und das Ergebnis sieht unter Win2000 wie folgt aus:

Rechts sehen Sie die Eigenschaf-	Eigenschaften vo	n tomcat (Lokaler Computer) ? 🔀
ten des tomcat-Dienstes unter	Allgemein Anme	lden Wiederherstellen Abhängigkeiten
NT-Server. Bei dem Starttyp	Dienstname:	tomcat
können Sie automatisch wählen,	Anzeigename:	tomcat
und die weiteren Registerkarten	Rosobroibung	
sind nicht gefüllt. Der Dienst	Beschleibung.	
lässt sich mit den Start/ Unterbre-	Pfad zur EXE-Da C:\Programme\1	ater: Fomcat\bin\jk_nt_service.exe tomcat
chungsbuttons manuell neu star-	,	
ten:	Starttyp:	Automatisch
	Dienstatatum	Gastatat
	Dienststatus:	Gestarret
	Starten	Beenden Anhalten Fortsetzen
	Sie können die 9 wenn der Dienst	Startparameter angeben, die übernommen werden sollen, von hier aus gestartet wird.
	Startparameter:	

ΟK

Abbrechen

2.1.5.2.12 Steuerung für das Applet: Die superx.properties

Das SuperX-Applet greift u.a. auf eine Datei **superx.properties** zu , um zu erfahren, mit welchem Datenbanksystem gearbeitet wird (Informix/Postgres).

Für diese Datei gibt es im Kernmodul ein Muster

\$SUPERX_DIR/webserver/tomcat/webapps/superx/applet/superx.properties.sam

das Sie nach superx.properties kopieren und wie folgt anpassen:

Die Adresse des Servlets wird normalerweise automatisch ermittelt, bei Netzwerkproblemen kann sie jedoch auch fest angegeben werden, dazu # vor SxServerURL entfernen und localhost ggfs. durch IP-Nummer/Rechnername ersetzen.

In der SuperX-Properties wird außerdem das Datenbanksystem (Variable SxDB) festgelegt, sowie das Logging (Variable logToKonsole). Bei der Installation von SuperX sollten Sie das Logging auf "all" setzen, im Echtbetrieb sollten Sie das Logging wie beim Servlet auf "none" setzen.

```
Ein Beispiel
             # Die Adresse des Servlets wird normalerweise automatisch ermit-
    für die
             telt,
             # bei Problemen kann sie hier fest angegeben werden, dazu # vor
    superx.
             SxServerURL entfernen und localhost ggfs. durch IP-
properties:
             Nummer/Rechnername ersetzen
             #SxServerURL=http://IP:8080/superx/servlet/SuperXDBServlet
             # Der Logging-Level logToKonsole kann eingeschaltet werden: "no-
             ne", "errors"=fehler, "all"=alles)
             logToKonsole=errors
             #Wird im Applet in Titelleiste angezeigt
             SxTitle=Testhochschule
             #Das verwendete DB-System, möglich sind "Informix" und
             "Postgres"
             SxDB=Postgres
             # HTML-Format Parameterleiste, Feldname in BOLD
             face=arial, helvetica size=-2>
             SxParamEnd=</font></body></html>
             SxParamBoldOn=<b>
             SxParamBoldOff=</b>
```

Wenn der Webserver mehrere Mandanten in unterschiedlichen Datenbank bedient, muss es für jeden Mandanten eine superx.properties geben, die den zusätzlichen Parameter MandantenID enthält, z.B. MandantenID=7200 (vergl. entsprechendes Kapitel).

Schließlich muss man noch darauf achten, dass ggfs. der Tomcat-Port in der Firewall (standardmäßig Port 8080, evtl. noch 8007 und 8443) freigegeben ist.

Die Homepage von SuperX liegt standardmäßig unter http://<IP-Nummer des Servers>:8080/superx/

2.1.5.2.13 Steuerung des XML-Frontends: PageComponents

Das XML-Frontend generiert aus XML-Datenströmen die Oberfläche im html-Format. So lässt sich die Oberfläche von SuperX beliebig mit XSLT anpassen, Details dazu finden Sie im SuperX-Entwicklerhandbuch. Die Neuentwicklung des XML-Frontends mit DHTML-Techniken erlaubt es, wahlweise den Themenbaum als Javascript-"Baum" anzuzeigen (SuperX Kernmodul ab Version 3.0rc7) oder als normales html-Menü (SuperX Kernmodul bis Version 3.0rc6). Wenn der Javascript-Baum nicht genutzt werden soll, kann dieser wie folgt ausgeschaltet werden:

Abschaltung des	Editieren Sie die Datei
Javascript-Baums	\$SUPERX_DIR/webserver/tomcat/webapps/superx/xml/pageComponents
	_html_final.xsl
	und setzen Sie folgende Anweisung aktiv:
	<xsl:template name="showJavascriptMenue"></xsl:template>
	<xsl:text>false</xsl:text>

Nach einem Tomcat-Neustart wird wieder das alte html-Menü angezeigt.

Viele Hochschulen, die SuperX mit LDAP-Anmeldung nutzen oder anderweitig konfigurieren, wollen weitere Steuerungmöglichkeiten über das Aussehen des Menüframes nutzen. Sie können auch steuern, wie der Fuss des linken Navigationsframes aussehen soll; standardmäßig werden folgende Links angezeigt:

Hyperlinks unter dem Themenbaum



Editieren Sie Ihre Datei "pageComponents_html_final.xsl":

Fügen Sie die folgenden Einträge in pageComponents_html_final.xsl vor dem Tag am Ende

</xsl:stylesheet>ein:

Ausblenden des Pass-	<xsl:template name="showPasswordChangeLink"></xsl:template>
wort-Ändern-Links•	<xsl:text>false</xsl:text>
wort-Andern-Emiks.	
Ausblenden des Log-	<xsl:template name="showLogoutLink"></xsl:template>
in/Logout-Links.	<xsl:text>false</xsl:text>
III/Logout-Links.	
Ausblenden des	<xsl:template name="showAppletLink"></xsl:template>
Applet-Links:	<xsl:text>false</xsl:text>

Sie können unter diesem Bereich noch weitere HTML-Elemente einbauen. Dafür gibt es ein in der Auslieferung von SuperX befindliches leeres Template menue fuss:

```
Einblenden weiterer
Links oder Texte:
Beispiel <xsl:template name="menue_fuss">
SuperX an der Universität xy</xsl:template>
```

Dies können Sie in pageComponents_html_final.xsl mit beliebigen Elementen füllen.

2.1.5.2.14 Einrichtung des Webservers bei mehreren Mandanten.

Wenn die Servlet-Engine mehrere SuperX-Mandanten in unterschiedlichen Datenbanken bedienen soll,

muss es unter \$SUPERX_DIR/webserver/tomcat/webapps/superx/WEB_INF eine Datei mandanten.cfg geben.

Darin müssen die MandantenIDs (typischerweise Hochschulnummern) der einzelnen Mandanten aufgeführt sein. (Jeweils eine ID pro Zeile). Zusätzlich muss es dann nicht eine db.properties geben, sondern für jeden Mandanten eine nach dem Schema db_XXXX.properties, wobei XXXX für die MandantenID steht.

Bearbeiten mit dem propadmin können Sie die einzelnen db_XXXX.properties Dateien, indem Sie ins Verzeichnis \$SUPERX_DIR/webserver/tomcat/webapps/superx/WEB_INF wechseln und dann den propadmin^{xiii} starten mit

propadmin.x ./db_XXXX.properties.

Nach dem Start von Tomcat können Sie in den Logdateien (meist catalina.out oder localhost.log) kontrollieren, ob für jeden Mandanten ein Datenbank-ConnectionPool aufgebaut wurde.

Unter \$SUPERX_DIR/webserver/tomcat/webapps/superx sollte es für jeden Mandanten ein Unterverzeichnis mit dem Namen der MandantenID geben.

z.B.

\$SUPERX_DIR/webserver/tomcat/webapps/superx/7200 \$SUPERX_DIR/webserver/tomcat/webapps/superx/7300 \$SUPERX_DIR/webserver/tomcat/webapps/superx/7400

In jedes der Mandantenunterverzeichnisse müssen einige Dateien und Verzeichnisse

reinkopiert werden, da Tomcat mit symbolischen Links Probleme hat.

Rufen Sie im Verzeichnis

\$SUPERX_DIR/webserver/tomcat/webapps/superx

das Skript copytoMandantenDir.x MANDANTENID auf (z.B. copytoMandantenDir.x 7200).

Falls noch nicht vorhanden wird ein Unterverzeichnis 7200 angelegt und alle Dateien dort hinkopiert. Wechseln Sie dann in das Mandantenverzeichnis 7200.

Im Unterverzeichnis applet muss die superx.properties angelegt werden, wie im vorherigen Abschnitt beschrieben. Zusätzlich muss die MandantenID in der superx.properties angegeben werden, z.B.

MandantenID=7200

Ebenso muss in Unterverzeichnis xml in der Datei anmeldung.htm, die MandantenID als versteckter Parameter mit übergeben werden, z.B.

<input type="hidden" name="MandantenID" value="7200">

Die einzelnen Mandanten können SuperX dann mit der Url

http://rechnername:8080/superx/MANDANTENID

aufrufen, z.B.

http://www.plgr-bw.de:8080/superx/7200

Beschreibung SuperX-Kern-Modul V3.0 © Projektgruppe SuperX

2.1.5.2.15 Einrichtung von DBFORMS bei mehreren Mandanten

Wenn auch die Administrationsabfragen von DBFORMS genutzt werden sollen, müssen die Datenbankverbindungen in zwei Steuerungsdateien eingetragen werden: der server.xml für die Datenbankverbindung, und der doforms-config.xml für die dbforms-Anbindung.

Die Vorbereitung des Tomcat-Servers für den Einsatz von dbforms wurde im Abschnitt zur server.xml (S. 51) erläutert. Für den Einsatz mehrerer Mandanten müssen die Datenquellen in der Datei \$SUPERX_DIR/webserver/tomcat/conf/server.xml eingetragen werden.

Diese sähe dann z.B. für die Mandanten 7200 und 7300, deren Datenbanknamen unter Postgres pg7200 und pg7300 lauten, so aus:

<Context path="/superx" docBase="superx" debug="0" Der mandantereloadable="true" crossContext="true"> fähige SuperX-<Logger className="org.apache.catalina.logger.FileLogger" Kontext in der prefix="localhost_superx_log." suffix=".txt" server.xml timestamp="true"/> <Environment name="maxExemptions" type="java.lang.Integer"</pre> value="15"/> <Parameter name="context.param.name" value="context.param.value" override="false"/> <Resource name="jdbc/mandant 7200" auth="Container" type="javax.sql.DataSource"/> <Resource name="jdbc/mandant_7300" auth="Container" type="javax.sql.DataSource"/> <!--Mandant 7200 wird angebunden: <ResourceParams name="jdbc/7200"> <parameter> <name>factory</name> <value>org.apache.commons.dbcp.BasicDataSourceFactory</value> </parameter> <parameter> <name>driverClassName</name> <value>org.postgresql.Driver</value> </parameter> <parameter> <name>url</name> <value>jdbc:postgresql://localhost/pg7200</value> </parameter> <parameter> <name>username</name> <value>superx</value> </parameter> <parameter> <name>password</name> <value>anfang12</value> </parameter> <parameter> <name>maxActive</name> <value>7</value> </parameter> <parameter> <name>maxIdle</name> <value>5</value> </parameter> <parameter> <name>maxWait</name> <value>-1</value> </parameter> <parameter> <name>removeAbandoned</name> <value>true</value> </parameter> <parameter> <name>removeAbandonedTimeout</name> <value>10</value> Beschreibung SuperX-Kern-Modul V3.0 </parameter> </ResourceParams> © Projektgruppe SuperX

```
<!--Mandant 7300 wird angebunden:
<ResourceParams name="jdbc/7300">
  <parameter>
    <name>factory</name>
    <value>org.apache.commons.dbcp.BasicDataSourceFactory</value>
  </parameter>
  <parameter>
    <name>driverClassName</name>
    <value>org.postgresql.Driver</value>
  </parameter>
  <parameter>
    <name>url</name>
    <value>jdbc:postgresql://localhost/pg7300</value>
  </parameter>
  <parameter>
    <name>username</name>
    <value>superx</value>
  </parameter>
  <parameter>
    <name>password</name>
    <value>anfang12</value>
  </parameter>
  <parameter>
    <name>maxActive</name>
    <value>7</value>
  </parameter>
  <parameter>
    <name>maxIdle</name>
    <value>5</value>
  </parameter>
  <parameter>
    <name>maxWait</name>
    <value>-1</value>
  </parameter>
  <parameter>
<name>removeAbandoned</name>
<value>true</value>
 </parameter>
<parameter>
<name>removeAbandonedTimeout</name>
<value>10</value>
</parameter>
</ResourceParams>
</Context>
```

Es werden also die Tags **<Resource... />** und **<ResourceParams... />** für jeden Mandanten dupliziert und konfiguriert. Jeder Mandanten-Datenquelle ist dann für dbforms über den Namen der Ressource ansprechbar, also in diesem Beispiel 7200 und 7300.

Diese Datenquellen müssen dann wie im Abschnitt zu dbforms (S. 48) erläutert in der Datei \$SUPERX_DIR/webserver/tomcat/webapps/superx/WEB-INF/dbforms-config.xml beim Tag <dbConnection .../> eingetragen werden:

Das Ende der Datei	•••
dbforms-config.xml	Hier endet Moduldefinition
	<pre><dbconnection id="7200" isjndi="true" na-<="" pre=""></dbconnection></pre>
bei mandantenfahigem	me="java:/comp/env/jdbc/mandant_7200"/>
Betrieb	<dbconnection id="7300" isjndi="true" na-<="" th=""></dbconnection>
	me="java:/comp/env/jdbc/ mandant_7300"/>

Wichtig ist, dass die Mandanten-ID mit der in der mandanten.cfg (s.o.) übereinstimmt.

2.1.5.3 Integration von Tomcat mit dem Apache

In Systemumgebungen, in denen bereits ein Webserver wie Apache läuft, bietet es sich an, den SuperX-Tomcat mit dem Webserver zu verbinden. Der Webserver kann so konfiguriert werden, dass Aufrufe zu http://<<Servername>>/superx direkt an Tomcat weitergeleitet werden können. Dies hat auch den Vorteil, dass die ungewöhnlichen Ports von Tomcat (8080 bzw. 8443) nicht in der Firewall freigegeben werden müssen. Außerdem kann die Verschlüsselung vom Apache durchgeführt werden, und es kann ein Load-Balaning eingeführt werden (Lastausgleich zwischen 2 Tomcat-Servern, gesteuert vom Apache). Wir empfehlen daher generell die Anbindung von Tomcat an den Apache für einen Produktivbetrieb.

2.1.5.3.1 Installation des Apache-Tomcat-Connectors

Der Apache-Tomcat-Connector für Apache 1.3.x und Apache 2.x heißt mod_jk und ist ein Apache-Modul, das via DSO in eine vorhandene Apache-Installation gelinkt werden kann. Das mod_jk kann man herunterladen z.B. von http://jakarta.apache.org (im Downloadbereich unter Sources, die aktuelle Version ist 1.2.6). Unter SuSE Linux 8.2 und höher befindet sich das mod_jk im Paket apache-tomcat-connectors. Ein unter SuSE Linux 9.0 für den Apache 1.3.28 kompiliertes mod_jk liegt im Kernmodul unter \$SUPERX_DIR/webserver/apache/lib), die Quellen liegen in \$SUPERX_DIR/webserver/apache/src/jakartatomcat-connectors-jk-1.2-src-current.tar.gz.

Bei manchen Systemen ist es sinnvoll, den mod_jk selbst zu kompilieren. Zunächst müssen der Apache 1.3.x bzw. 2.x und das apxs-Tool installiert sein (apxs ist unter SuSE Linux Teil der *devel*-Package für Apache). Nun entpackt man die mod jk-Quellen z.B. im Verzeichnis /usr/src/apache/

Dann geht man als root in das Verzeichnis

/usr/src/apache/jakarta-tomcat-connectors-jk-1.2.6-src/jk/native

und gibt ein (\$JAVA_HOME und /usr/sbin/apxs müssen ggf. angepasst werden):

Unter Apache 1.3.x:

./configure --with-java-home=\$JAVA_HOME --enable-EAPI --with-apxs=/usr/sbin/apxs

Unter Apache 2.x:

./configure --with-java-home=\$JAVA_HOME --with-apxs=/usr/sbin/apxs2

Dann gilt für beide:

make

Nutzer von RedHat 9.x beachten bitte folgende Fußnote^{xiv}.

Danach ist das Modul kompiliert und wird in das Modulverzeichnis des Apache kopiert (z.B.

/usr/lib/apache). Für Apache 1.3:

cp ./apache-1.3/mod_jk.so /usr/lib/apache

bzw. für Apache 2.x:

cp ./apache-2.0/mod_jk.so /usr/lib/apache2

Danach kann die Konfiguration des mod_jk beginnen, was im folgenden Kapitel beschrieben ist.

2.1.5.3.2 Umleitung von Requests vom Apache zu Tomcat

Die Konfiguration des Apache zur Anbindung an Tomcat ist im tomcat-apache-howto dokumentiert, der sich in jeder Download-Version des offiziellen Tomcat 4.1.27 befindet (webapps/doc).

Die Umleitung von Requests vom Apache zum Tomcat kann auch auf zwei Rechnern geschehen, z.B. um den Apache-Server in der DMZ und den Tomcat-Server im Intranet zu betreiben. Wir empfehlen letzteres aus Sicherheitsgründen, beachten Sie aber dabei, dass auch die Verbindung vom Apache-Server zum Tomcat via mod_jk verschlüsselt wird, z.B. über einen ssh-Tunnel.

In der SuperX-Distribution sind die Tomcat-spezifischen Dateien für die Anbindung an den Apache 1.3.x via mod_jk bereits enthalten, es müssen lediglich ein paar Anpassungen gemacht werden:

- Teil der SuperX-Distribution ist ein Konfigurationsbeispiel mit dem Namen \$SUPERX_DIR/webserver/tomcat/conf/superx_mod_jk.conf.sam. dieses können Sieumbenennen nach superx_mod_jk.conf, und in der Datei den Pfad für das mod_jk-Modul (mod_jk.so) anpassen. Ausserdem kann der Logging-Level festgelegt werden (Werte: "debug", "warning", "error", im Echtbetrieb empfehlen wir "error").
- In der Datei \$SUPERX_DIR/webserver/tomcat/conf/workers.properties muss der Parameter workers.tomcat_home

auf den richtigen Pfad gesetzt werden (wenn Sie SuperX in /home/superx installiert haben, brauchen Sie hier keine Änderungen vornehmen).

Ausserdem muss der richtige Pfad für workers.java_home gesetzt werden, sowie der Pfad-Demiliter ps für das Betriebssystem ("/" für Unix, "\" für Win, ":" für Mac)

Wenn Sie den Apache auf einem separaten Rechner betrieben, dann müssen Sie beim Parameter worker.ajp13.host nicht "localhost", sondern den Rechnernamen / IP-Nr. des Tomcat-Servers eintragen.

• Danach fügen Sie am Ende der Apache-Konfigurationsdatei (unter SuSE Linux z.B. /etc/httpd/httpd.conf) die Zeile

Include /home/superx/webserver/tomcat/conf/superx_mod_jk.conf)

ein. Außerdem müssen Sie ggf. die Umgebungsvariable DirectoryIndex auf index.htm setzen (nicht nur index.html), da die Startseite in den Verzeichnissen immer index.htm heißt. Danach starten Sie Apache neu (apachectl restart).

- In der Datei \$SUPERX_DIR/webserver/tomcat/conf/server.xml auf dem Tomcat-Rechner kann dann der http-Connector 8080 auskommentiert werden, und der Apache-Connector ajp13 kann benutzt werden; standardmäßig geht dieser über den Port 8009. Dieser Connector ist bei Auslieferung von SuperX aktiviert.
- Dann starten Sie zuerst Tomcat neu, und dann den Apache. Danach müsste auf dem Webserver das Verzeichnis superx gemounted sein, und alle Anfragen mit der Endung *.jsp bzw. in das serv-let-Verzeichnis gehen zu Tomcat.
- Danach müssen Sie ggf. in der Datei \$SUPERX_DIR/webserver/tomcat/webapps/superx/applet/superx.properties den Port für Tomcat (Vorbelegung: ":8080") rausnehmen (mit "#" auskommentieren).

Sobald Tomcat so an den Apache angebunden ist, kann auch die Verschlüsselung (S. 69) über den Apache laufen. Wenn das Apache-Modul mod_ssl installiert und konfiguriert ist, dann werden auch automatisch anhand des Präfixes http://... und https://... Anfragen an Tomcat weitergeleitet.

2.1.5.3.3 Einrichtung von Load Balancing

Mit dem eingerichteten mod_jk lässt sich recht einfach ein Lastausgleich zwischen mehreren Tomcats oder eine Trennung von Apache und Tomcat auf zwei Server implementieren. Die Konfiguration findet statt im Tomcat-Konfigurationsverzeichnis \$SUPERX_DIR/webserver/tomcat/conf.

Dazu muss in der Datei workers.properties ein zweiter Worker eingerichtet werden, siehe Beispieldatei workers.properties.lb.sam in der SuperX-Distribution. Außerdem muss in der server.xml auf dem Rechner, auf dem der Apache läuft, ein zweiter AJP-Connector eingerichtet werden, z.B. am Port 8010.

Auszug aus
der server.xml des
Apache-Rechners<!-- Define an AJP 1.3 Connector on port 8009 -->
<Connector className="org.apache.ajp.tomcat4.Ajp13Connector"
port="8009" minProcessors="5" maxProcessors="75"
acceptCount="10" debug="0"/><Connector className="org.apache.ajp.tomcat4.Ajp13Connector"
port="8010" minProcessors="5" maxProcessors="75"
acceptCount="10" debug="0"/>

Der AJP-Connector am Port 8010 muss dann auf dem zweiten Tomcat-Server eingetragen werden (nur dieser, nicht der 8009er).

Danach kann in der mod_jk-Konfigurationsdatei ein Lastausgleich eingerichtet werden (siehe Beispieldatei superx_mod_jk_lb.conf.sam).

2.1.5.3.4 Einrichten von SSL beim Apache 1.3.x unter Linux

Achtung:Diese Anleitung funktioniert bei Debian, Ubuntu und Suse Li-
nux bis 10.0 (mit OpenSSL 0.9.7g); bei Suse Linux 10.1 (mit
OpenSSL 0.9.8a) klappt dies leider nicht mehr, für Hinweise zur
Korrektur sind wir dankbar.

Der Apache Version 1.3.x benötigt für den SSL-Betrieb das Modul mod_ssl^{xv}, im Apache 2.x ist das SSL-Modul bereits Bestandteil des "Kern"-Apache. Mit Hilfe des Openssl-Paketes sowie können Schlüssel für den Server erzeugt werden. Im Folgenden erläutern wir das Vorgehen unter SuSE Linux 8-9, für andere Distributionen müssen Sie ggf. die Verzeichnisnamen anpassen. Für die Installation verwenden wir zunächst ein selbst signiertes Zertifikat, was zwar den Nachteil hat, dass die Anwender vor dem Aufruf der Webseite eine Warnung erhalten ("Diese Seite stammt aus einer nicht vertrauenswürdigen Quelle..."), der Vorteil ist aber, dass das Vorgehen relativ einheitlich ist und später bei Bedarf leicht um ein öffentliches Zertifikat erweitert werden kann. Wenn die Verschlüsselung mit einem selbst signierten Zertifikat funktioniert, dann ist der Rest relativ einfach.

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 (bitte passen die die Verzeichnisnamen jeweils an Ihr OpenSSL-Paket an):

/usr/share/ssl/misc/CA.sh -newca

Sie geben ein Passwort ein und die jeweiligen Angaben (Land, Organisation etc.). Beim "Common Name" muss der DNS-Servername angegeben werden.

Das öffentliche CA-Zertifikat liegt nun in /root/demoCA/cacert.pem und der private Schlüssel liegt in /root/demoCA/private/cakey.pem.

Wenn Sie Ihren Besuchern das CA-Zertifikat zum Download anbieten möchten, müssen Sie dieses zuerst in das entsprechende DER-Format konvertieren:

openssl x509 -in demoCA/cacert.pem -out capub.crt -outform DER

Es wird die Datei /root/capub.crt erzeugt. Danach müssen Sie für den Server-Dienst http ein weiteres Zertifikat erzeugen. Mit

/usr/share/ssl/misc/CA.sh -newreq

erzeugen Sie ein neues Zertifikat, das Passwort sollte auf keinem Fall dem obigen Server-Zertifikat entsprechen.

Wenn Sie Ihr Zertifikat bei einer Zertifizierungstelle signieren lassen möchten, müssen Sie die Zertifizierungsanfrage aus der Datei newreq.pem mit folgendem Befehl in eine separate Datei speichern. Senden Sie auf gar keinen Fall die Datei newreq.pem zur Zertifizierungsstelle, da diese zusätzlich Ihren privaten Schlüssel enthält.

openssl req -text -in newreq.pem -out request.pem

Wie und in welchem Format Sie die Anfrage an die von Ihnen ausgewählte Zertifizierungsstelle senden müssen, erfahren Sie von der entsprechenden Zertifizierungsstelle.

Mit dem weiter oben erstellten CA-Zertifikat können Sie Ihr http-Zertifikat folgendermaßen selbst signieren:

/usr/share/ssl/misc/CA.sh -sign

Es wird eine Datei newcert.pem erzeugt. Nachdem Sie nun ein signiertes Zertifikat für Ihre Anwendung erstellt haben, müssen Sie dieses nur noch in das entsprechende Verzeichnis kopieren und in der Konfigurationsdatei eintragen. Der Apache erwartet den privaten Schlüssel in einer separaten Datei, in solchen Fällen können Sie den privaten Schlüssel wie folgt extrahieren

openssl rsa -in newreq.pem -out newkey.pem

Nun bereiten wir den Neustart des Apache mit ssl-Modul vor. Die Einbindung mit Load;odule und Add-Module muss bei den meisten Distributionen nicht manuell gemacht werden.

Apache 1.3.x-SSL-
Verschlüsselung unter
SuSE LinuxUnter SuSe Linux müssen Sie zunächst eine Umgebungsvariable
setzen. Schreiben Sie in der Datei /etc/sysconfig/apache:
HTTPD_SEC_MOD_SSL=yes
Und starten Sie danach einmal das Script SuSEconfig.

Im Apache muss nun in der Steuerungsdatei httpd.conf der Pfad zum privaten und öffentlichen Schlüssel angegeben werden. Das folgende Beispiel geht davon aus, dass der öffentliche CA-Schlüssel auf der

Website des Users superx (Modul public_html des Apache) unter /home/superx/public_html steht, und dass der private Schlüssel des Servers vom User root im Verzeichnis /root/demoCA erzeugt wurde.

Auszug aus der	<virtualhost <<ipnr.="" des="" webservers="">>:443></virtualhost>	
Anache-	ServerName < <ihr dns-servername="">></ihr>	
	# SSL Engine Switch:	
Konfigurations-	# Enable/Disable SSL for this virtual host.	
datei httpd.conf	SSLEngine on	
	# SSL Cipher Suite:	
	SSLCipherSuite	
	ALL: !ADH: !EXPORT56:RC4+RSA: +HIGH: +MEDIUM: +LOW: +SSLv2: +EXP: +eNUL	
	# Server Certificate:	
	SSLCertificateFile /home/superx/public_html/capub.crt	
	# Server Private Key:	
	SSLCertificateKeyFile /root/demoCA/private/cakey.pem	

Danach müssen Sie in /etc/sysconfig/apache die Systemvariable HTTPD_START_TIMEOUT auf einen sinnvollen Wert setzen, z.B. 10. Sie haben dann beim Start des Apache 10 Sek. Zeit, dass CA-Passwort einzugeben.

Wenn sie wünschen, dass der Apache beim Booten ohne Passwort-Abfrage startet, dann müssen Sie das CA-Passwort löschen und die Leserechte für den privaten Schlüssel ändern (nur root und der Apache-Daemon haben Leserecht)^{xvi}. Dies ist allerdings ein Sicherheitsrisiko; der Server wird leichter kompromittierbar, wenn ein Hacker auf den Rechner kommt und die Datei lesen kann, kann er den Schlüssel missbrauchen. Unserer Erfahrung nach ist aber nur dieser Weg gangbar, denn bei einem Reboot nach Stromausfall würde der gesamte Webserver sonst nicht laufen!

Wir geben als root im Verzeichnis /root/demoCA/private ein:

openssl rsa -in cakey.pem -out cakey2.pem

(1x mit der Passphrase bestätigen).

Dann wird ein Schlüssel ohne Passphrase erzeugt. Wenn wir diesen dann wiederum in

/etc/httpd/httpd.conf eintragen:

#SSLCertificateKeyFile /root/demoCA/private/cakey.pem SSLCertificateKeyFile /root/demoCA/private/cakey2.pem

Dann startet der Apache ohne Passwortabfrage. In diesem Fall kann man auch die Variable HTTPD_START_TIMEOUT auf 1 zurücksetzen.

Zum Abschluss können Sie bei einem selbst signierten Zertifikat die oben erstellte Datei /root/capub.crt auf den Webserver kopieren und mit folgendem Link auf Ihrer Webseite verfügbar machen:

CA-Zertifikat

Die Anwender können dann mit Klick auf Link das Zertifikat importieren und somit im Browser speichern, so dass die Warnung, dass die Quelle nicht vertrauenswürdig ist, nicht mehr kommt. Wir haben auch den Eindruck, dass das Applet dann schneller arbeitet.

2.1.5.3.5 Einrichten von SSL beim Apache 2.x unter Linux

Die SSL-Einrichtung beim Apache2.x entspricht weitgehend derjenigen beim Apache 1.3.x, die Pfade zu den log-, lib- und etc-Konfigurationsdateien haben jedoch generell eine "2" angehängt, z.B. die der Pfad /var/log/apache2. Die Konfiguration des Servers befindet sich (bei der Standardeinstellung von Apache2) nicht mehr in /etc/httpd, sondern in /etc/apache2. Unter Suse Linux befindet sich die Susespezifische Konfiguration nicht mehr in /etc/sysconfig/apache, sondern in der Datei /etc/sysconfig/apache2. In dieser Datei wiederum wird die Variable HTTPD_SEC_MOD_SSL nicht mehr benutzt. Das Modul **ssl** ist nicht mehr separat zu installieren, sondern bereits im Lieferumfang des Apache2 enthalten, das Modul muss nur in den entsprechenden LoadModule und Include-Abschnitten geladen werden.

Apache2 mit SSL	SuSE-typisch wird die Konfiguration in einer Datei im Ver-
unter SuSE Linux	zeichnis /etc/sysconfig abgelegt, nämlich in apache2. Dort set-
	zen Sie in der Direktive
	APACHE_CONF_INCLUDE_FILES="/home/superx/webserver/tomcat/conf/
	<pre>superx_mod_jk.conf /etc/apache2/vhosts.d/myhost-ssl.conf"</pre>
	die Tomcat-Anbindung (S. 67) und den Virtuellen SSL-Host.
	Letzteren konfigurieren Sie am besten, indem Sie die Vorlage
	/etc/apache2/vhosts.d/vhost-ssl.template kopieren, z.b. wie o-
	ben nach myhost-ssl.conf.
	Weiter unten in /etc/sysconfig/apache2 setzen Sie die Direktive APACHE_SERVER_FLAGS="SSL"
	Damit werden in verschiedenen anderen conf-Dateien die Ab-
	fragen <ifdefine ssl=""> positiv aufgelöst und die jeweiligen Direk-</ifdefine>
	tiven darin werden aktiviert.
	Nach dem Ändern der Datei /etc/sysconfig/apache2 müssen Sie
	als User root das Script SuSEconfig ausführen.

Die Kommentare in httpd.conf bzw. unter SuSE in /etc/sysconfig/apache2 sollten selbsterklärend sein. Außerdem werden die einzelnen Konfigurationsparameter nicht mehr in einer großen httpd.conf gesammelt, sondern in separaten conf-Dateien. Bei virtuellen Hosts zum Beispiel befinden sich die Konfigurationen in Dateien mit der Endung *.conf im Verzeichnis vhosts.d. Der Startpunkt ist aber immer die httpd.conf (standardmäßig in /etc/apache2).

Wenn Sie keine Virual Hosts nutzen, dann können Sie den Abschnitt, der im Konfigurationsbeispiel /etc/apache2/vhosts.d/vhost-ssl.template beschrieben ist (und sich im Übrigen mit der Spezifikation aus derApache 1.3.-Konfiguration (S. 69) deckt) auch in der Datei /etc/apache2/default-server.conf einfügen.

2.1.6 Anpassungen auf den Client-Rechnern

Der Vorteil von java-basierten Webclients ist es, dass prinzipiell keine Installationen auf den Clients notwendig sind, und dass sie plattformübergreifend arbeiten. Man muss natürlich das Java-Plugin instal-
Installation

liert haben. Darüberhinaus hat sich bei verschlüsselten Verbindungen gezeigt, dass die Performance erheblich gesteigert wird, wenn das Zertifikat des Webservers importiert wird (s.o.).

Das eigentliche SuperX-Applet wird bei jedem Aufruf (je nach Cacheing des Browsers) neu geladen; der Umstieg auf neue Versionen des Applets ist also ohne lokale Installationen möglich. Eine Bedienungsanleitung zum Java-Client findet sich unter

\$SUPERX_DIR/doc/benutzerhandbuch

2.1.6.1 Installation der Java-Runtime

Für die Installation der Java-Dateien auf den Clients sind einige Maßnahmen vorgesehen; wir haben für Win32- und Linux-Clients einige Scripte geschrieben, die diesen Vorgang automatisieren. Scripte für andere Betriebssysteme sind in Vorbereitung. Am einfachsten ist es, den Anweisungen, die der Kernmodul-Distribution beiliegen (http://localhost:8080/superx/applet/install/) zu folgen.

2.1.6.1.1.1 Installation des Applets unter Windows

Im Kernmodul liegt unter

\$SUPERX_DIR/webserver/tomcat/webapps/superx/applet/install/

eine Mini-Installationssite, die nach der erfolgreichen Installation des Servers von den Clients aufgerufen werden kann. Der Browser wird ausgelesen, und je nach Browser werden unterschiedliche Installationsscripte angeboten. Dabei handelt es sich um jsp.- javascript- und VBS-Scripte.

Wenn man zur Installation auf den Java-Link auf der Applet- Startseite klickt, dann startet eine jsp-Anwendung, die versucht, den Browser und die Umgebung zu diagnostizieren und dann den richtigen Installationsschritt vorzuschlagen. Wenn Sie nicht wollen, dass Java jedesmal von Sun heruntergeladen werden muss, können Sie die

j2re-1_4_1_02-windows-i586-i.exe

auch von SUN laden und dann in das Verzeichnis

/webserver/tomcat/webapps/superx/applet/install/download legen - SuperX nimmt dann diese; Sie können auch eine beliebige 1.4-jre nehmen, müssen Sie dann aber exakt wie oben umbenennen.

Rechner mit dem Java 1.3 können entweder updaten oder über die Installationssite und Javascript auf die alten Startseiten umgeleitet werden (superx.html und superx_ns6.html), die immer noch existieren. Wenn SuperX in einer verschlüsselten Verbindung arbeitet oder wenn der Client auch Joolap einsetzt, dann muss auf Java 1.4 gewechselt werden.

Weiterhin kann es sich bei verschlüsselten Verbindungen vorkommen, dass der erste Aufruf der htmlSeite sehr langsam ist. Hier kann es sich als notwendig erweisen, das Zertifikat des Webservers lokal auf dem Client zu installieren, oder den Webserver auf den Clients als vertrauenswürdige Site einzutragen (s.u.). Die Installation des Zertifikats ist Browserabhängig. Details zum IE finden Sie unter http://www.uni-duisburg.de/Verwaltung/Dez2/superx/ZertifikatIEmanuell.htm.

2.1.6.1.2 Manuelle Anpassungen der Policy

Bei einigen Windows-Umgebungen (z.B. mit Netscape 6.1, ohne IE, oder mit Windows XP) läßt sich die Policy nicht scriptgesteuert installieren. Man muss dann die Policy dialogisch einrichten. Starten Sie dazu die Anwendung policytool, die sich im Lieferumfang der Java-Runtime befindet. Wenn Sie die Anwendung z.B. unter C:\Programme\Java\JRE\1.4.1_02\bin\policytool.exe installiert haben, dann starten Sie die Anwendung mit Doppelklick und gehen wie folgt vor:

Die AWT-Permission "AccessClipboard"	Permissions Add New Permission:	X
muss gesetzt werden.	AWTPermission accessClipboard Actions: Signed By:	java.awLAWTPermission accessClipboard
	ок	Cancel
Die Runtime-	Permissions	×
Die Runtime- Permission "queuePrint-	Permissions Add New Permission:	×
Die Runtime- Permission "queuePrint- Job" muss gesetzt wer-	Permissions Add New Permission: RuntimePermission	Java lang.RuntimePermission
Die Runtime- Permission "queuePrint- Job" muss gesetzt wer- den.	Permissions Add New Permission: RuntimePermission GueuePrintJob Actions: Signed By:	Java.lang.RuntimePermission queuePrintJob

Danach klicken Sie auf "Done" und speichern die Policy im Home-Verzeichnis Ihrer Windows-Kennung, z.B. c:\dokumente und einstellungen\<<Ihre Kennung>>\.java.policy

2.1.6.1.3 Installation des Applets unter UNIX / Linux

Die Installtionssite von SuperX erkennt, ob es sich um einen Linux-Browser handelt. Die Anwender werden zum Download auf die Seiten von Sun verwiesen.

Unter UNIX / Linux werden zunächst die Dateien der Java-Runtime bzw. des JDK 1.4.x der Firma SUN installiert (s.o.). In Mozilla 1.4 oder höher bzw. Netscape 6.x oder 7.x wird das SuperX-Applet am besten unterstützt.

Bei der Installation des Browsers ist zu beachten, daß im Plugins-Verzeichnis der Browserinstallation ein symbolischer Link auf die libjavaplugin_oji.so von der Java-Runtime gelegt wird, z.B. für Netscape

ln -s /usr/lib/java2/jre/plugin/i386/ns610/libjavaplugin_oji.so
/opt/netscape/plugins/libjavaplugin_oji.so

Bei Mozilla 1.x und Java 1.4.1 muss u.U. ein spezielles Java-1.4.1 verwendet werden, da das "normale" Java 1.4.1 mit einem anderen Compiler erzeugt wurde. Für Mozilla ist es zwingend notwendig, eine mit dem gcc 3.2.2 kompiliertes Paket zu verwenden, das auf der Website von Blackdown^{xvii} zu beziehen ist. Bei Java 1.4.2 soll dieses Problem laut Aussage von SUN nicht mehr bestehen, da auch dieses mit dem gcc 3.2.2 kompiliert ist. Wir haben das Vorgehen bei einem vorinstallierten Java von Sun in SuSE Linux 8.x-9.0 getestet^{xviii}, hier gab es keine Probleme. Bei RedHat 9.0 klappte es allerdings nicht.

Installation

Nach der Installation von Java lautet der obige Befehl (wenn sowohl Java als auch Mozilla in /usr/local installiert sind):

ln -s /usr/local/jdk1.4.1/jre/plugin/i386/mozilla/javaplugin_oji.so /usr/local/mozilla/plugins/

👸 SuperX, SuperX	
II-3 Info Anmel₫en Passwort Clone Kopieren Drucken Hilfe Info	Surger
Themenauswahl Maske Tabelle	
	Erläuterung
Abfragen Administration Benutzer von SuperX Institutionsrechte Maske kopieren Maske löschen Masken von SuperX Neue Maske Nutzungsprotokolle (intern)	er gegenwärtigen Benutzer von und deren Sachgebietsrechte
Abfrage suchen	
Oeffnen	
26 Sätze gefunden	
Java Applet Window	

Bei erfolgreicher Anmeldung erscheint folgendes Fenster:

2.1.6.2 Bei Problemen mit dem Start des Applets

Wenn es Probleme mit dem Start des Applets gibt, kann dies verschiedene Ursachen haben.

Unter Netscape ist aufgefallen, dass bei verschlüsselter Verbindung auf dem Server die Datei <code>\$superx-dir/webserver/tomcat/webapps/superx/applet/superx_help/superx.hs</code> im gleichen Verzeichnis auch mit dem Namen <code>superx_de_DE.hs</code> existieren muss.

Eine weitere Ursache können **Sicherheitseinstellungen** sein. Fügen Sie Ihren SuperX-Server zur Liste der vertrauenswürdigen Sites hinzu.

Hier als Beispiel die Einstellung für den Duisburger SuperX-Server im InternetExplorer.

Im InternetExplorer und Extras / Internetoptionen, Registerkarte "Sicherheit" Punkt Vertrauenswürdige Sites. Auf "Sites" klicken.



Danach gibt man wie gezeigt den SuperX-Server ein und klickt auf Hinzufügen und OK.

Im **lokalen Netz** kann es durch den **Proxy** zu Problemen kommen. Man sollte daher den Proxy-Server für lokale Adressen umgehen.

Im InternetExplorer geht das folgendermaßen:

Zunächst wählt man im IE-Menü Extras-> Internetoptionen aus und wechselt zur Registerkarte Verbindungen. Dann klickt man auf LAN-Einstellungen.

Installation

Sofern "Proxyserver verwenden" aktiviert ist, sollte man den Menüpunkt "Proxyserver für lokale Adressen umgehen" ebenfalls aktivieren.

Einstellungen für lokales Netzwerk (LAN) 🛛 📲	'X
Automatische Konfiguration Die automatische Konfiguration könnte die manuellen Einstellungen überlagem. Deaktivieren Sie sie, wenn Sie die Verwendung der manuelen Einstellungen garantieren möchten.	
Automatische Suche der Einstellungen	
🗖 Automatisches Konfigurationsskript verwenden	
Adlesse -	
Proxyserver	
Proviserver verwenden	
Agresse: xy.uni-duisburg.de Agschluss: 3128 Erweitert	
Proxyserver für lokale Adressen umgehen	
· · · · · · · · · · · · · · · · · · ·	
OK Abbreche	n

Nach Veränderungen der Einstellungen ist es generell sehr sinnvoll den Cache zu leeren (S. 77).

2.1.6.3 Leeren des Browser-Cache

Wenn ein neues SuperX-Applet auf dem Webserver installiert wird, ist es möglich dass die Clients dies nicht sofort mitbekommen. Je nach Java-Version und Betriebssystem unterscheiden Sie sich Wege, den Browser-Cache zu leeren. Unter Windows mit Java 1.4.x wird der Browser-Cache geleert, bei Windows mit Java 1.5.x oder unter Linux wird der Java-Cache (S. 79) geleert. Im Zweifelsfall löschen Sie beide Caches.

Beim Browser-Cache sind die Einstellungen des Browsers maßgeblich. Beim **Internet Explorer** gehen Sie in das Menü "Extras"->"Internetoptionen"

In der Registerkarte "Allgemein" sehen Sie im Abschnitt "Temporäre Internetdateien" den Button "Dateien löschen"; klicken Sie darauf, und löschen Sie alle Inhalte. Danach klicken Sie auf "Einstellungen"...

Internetop	tioner			? 🗙
Verb	indünge	n P	rogramme	Erweitert
Allgeme	ein	Sicherheit	Datenschutz	Inhalte
Startseil	e			
	Sie kö	nnen die Seite änd	ern, die als Startseite a	ngezeigt wird.
	Adres:	se: /isapi/redir.	dll?prd=ie&pver=6&a	r=msnhome
	Ak	tuelle Seite	itandardseite	eere Seite
Tempor	äre Intei	netdateien		
e	Seiten spezie anzeig	, die Sie im Internet llen Ordner gespeic en zu können.	besucht haben, werde hert, um sie später sch	en in einem neller
	Cook	ies löschen Da	teien löschen) Ein	stellungen
	Der Or haben Seiten	dner "Verlauf" enth , um einen schnelle zu ermöglichen.	iält Links zu Seiten, die n Zugang zu kürzlich t	Vie besucht besuchten
	Tage, aufbev	die die Seiten in ''V vahrt werden:	erlauf" 20 🤤 🔍 Ve	erlauf'' leeren
Farbe	en	Schriftarten	Sprachen	Eingabehilfen
			K Abbrecher) Übernehmen

...und markieren Sie den Knopf "Bei jedem Zugriff auf die Seite". Dann drücken Sie "OK".

Einstellungen	?	×
Neuere Versionen der gespeicherten Seiten suchen: Bei jedem Zugriff auf die Seite Start von Internet Explorer Advinatisch Nie		
Temporäre Internetdateien		
Aktueller Ort: C:\Dokumente und Einstellungen\superx\Lokale Einstellungen\Temporary Internet Files\		
Anteil des Speicherplatzes auf dem Datenträger:		
1249 📚 MB		
Ordner verschieben) Dateien anzeigen Objekte anzeige	n]
OK Abbrech	ien	J

Starten Sie den Browser dann neu.

Bei **Netscape/Mozilla** befindet sich die Einstellung im Menü "Edit" (deutsch "Bearbeiten") -> "Preferences" (deutsch "Einstellungen").

Hier drücken Sie den Button "Clear Cache" ("Cache leeren") und kreuzen dann unten den Button "Every time I view the page" ("Bei jedem Zugriff auf die Seite") an.

Category		Cache
	^	Gathe
🖃 Navigator		Set Cache Options
History		The cache keeps copies of frequently visited web pages on your bard disk.
Languages		(Clicking Reload always shows you the latest version of a page.)
Helper Applica		Cache: 50 MB Clear Cache
- Smart Browsing		Costo Eolderi
-Internet Search		
- Tabbed Browsing		
Downloads	≣	Cache files will be stored in a subfolder named "Cache" of the directory you
		speciry, Restart Mozilia for changes to take effect.
		Compare the page in the cache to the page on the network:
		 Every time I view the page Once per session
🗄 ChatZilla		O W the page is out of date O Never
Advanced		
- Scripts & Plugins		Link Prefetching
-Keyboard Navi		- Prefetch web pages when idle, so that links in web pages designed
Cache		for prefetching can load faster.
Proxies		
-HTTP Networking		
-Software Inst		
		OK Cancel <u>H</u> elp

Starten Sie den Browser dann neu.

2.1.6.4 Leeren des Java - Cache

Bei der Java-Runtime Java 1.5.x unter Windows sowie bei der Java Runtime 1.4.x unter Linux wird ein separater, vom Browser unabhängiger Cache genutzt, der manuell geleert werden muss. Löschen Sie also alle Inhalte in den Pfaden:

Unter Windows:

c:\Dokumente und Einstellungen\<<Kennung>>\Anwendungsdaten\ sun\java\deployment\cache\javapi\v.1.0\jar*

Unter Linux:

/home/<<Kennung>>/.java/deployment/cache/javapi/v1.0/jar/*

Danach starten Sie den Browser neu.

2.2 Update 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 die Dateien unterhalb von \$SUPERX_DIR gesichert werden, und die Datenbank muss vorher exportiert werden. Der Bereich Webserver sollte ganz überspielt werden.

Wenn Sie SuperX nur in Verbindung mit dem COB-Modul betreiben, empfehlen wir die Neuinstallation, da sich auch das COB-Modul geändert hat. Sichern Sie ggf. benutzerspezifische Änderungen in der Tabelle aggregierung.

© Projektgruppe SuperX

Beschreibung SuperX-Kern-Modul V3.0

2.2.1 Kurzanleitung zum Upgrade von Version 2.0 nach 2.1

Die folgende Kurzanleitung zeigt die wesentlichen Schritte für den Upgrade von 2.0 nach 2.1.

- 1. Stoppen Sie tomcat über \$SUPERX_DIR/webserver/tomcat/bin/shutdown.sh
- 2. Sichern Sie den alten Tomcat mv \$SUPERX_DIR/webserver/tomcat \$SUPERX_DIR/webserver/tomcat3
- 3. Sichern Sie das gesamte Verzeichnis \$SUPERX_DIR
- 1. Entpacken Sie das neue Kernmodul unter \$SUPERX_DIR cd \$SUPERX_DIR tar -xzvf kernmodul2.1.tar.gz

Wenn Sie Datenbank- und Webserver auf unterschiedlichen Rechnern betreiben, müssen Sie das Kern-

modul auf beiden Rechnern entpacken.

Die alten Scripte werden dadurch überschrieben, nicht aber die vorhandenen Properties-Dateien (unsere properties-Dateien sowie auch das Access-Frontend haben immer die Endung .sam für sample).

2.2.1.1 Der Datenbankupgrade

- 1. Gehen Sie in das Verzeichnis \$SUPERX_DIR/db/bin
- 2. Kopieren Sie die Datei $sol_env.ach sol_env.alt$
- 3. Kopieren Sie SQL_ENV.sam nach SQL_ENV
- $\text{4. } \" \text{Ubertragen Sie relevante Parameter aus sol_env.alt nach sol_env, z.B. sx_client, java_home etc. } \\$
- 5. Sourcen Sie die neue SQL_ENV mit . SQL_ENV
- 6. Gehen Sie in das Verzeichnis \$SUPERX_DIR/db/install
- 7. Sichern Sie die Datenbank mit dump_it.x (Bei Informix müssen Sie vorher alle Datenbankverbindungen, d.h. auch Tomcat, beenden).
- 8. Gehen Sie in das Verzeichnis upgrade
- 9. Starten Sie das Script kernmodul_upgrade20_to_21.x, und prüfen Sie auf Fehler in der Logdatei upgrade.log^{xix}
- 10.Ggf. müssen Sie das Script macro_ids.sql bzw. macro_pg.sql von Hand starten, wenn die Tabelle macro_masken_bez nicht das Feld nummer enthält.
- 11.Entfernen Sie aus dem Themenbaum den "alten" Ast Administration. Sie erkennen diesen daran, dass darunter weniger Masken sind als unter dem neuen Ast.
- 12.Richten Sie bei Bedarf das neue Access-Frontend (S. 40) ein (Kopieren von \$SUPERX_DIR/db/superx_frontend_sam.mdb nach superx_frontend.mdb).

2.2.1.2 Upgrade des Webservers

- 1. Kopieren Sie die Dateien superx.properties und db.properties cp \$SUPERX_DIR/webserver/tomcat3/webapps/superx/WEB-INF/db.properties \$SUPERX_DIR/webserver/tomcat/webapps/superx/WEB-INF/db.properties cp \$SUPERX_DIR/webserver/tomcat3/webapps/superx/applet/superx.properties \$SUPERX_DIR/webserver/tomcat/webapps/superx/applet/superx.properties
- 2. Nur bei Informix: Kopieren Sie die Datei \$SUPERX_DIR/webserver/tomcat3/lib/ifxjdbc.jar nach

\$SUPERX_DIR/webserver/tomcat/webapps/superx/WEB-INF/lib

3. Starten Sie Tomcat neu, und testen Sie die Anmeldung

Installation

4. Mit Tomcat 4 ist auch eine neue web.xml im Verzeichnis

\$SUPERX_DIR/webserver/tomcat/webapps/superx/WEB-INF/web.xml installiert worden
Wenn Sie die Logging-Parameter oder die maxRows geändert haben, dann sollten Sie die Änderungen
manuell von der tomcat3-Datei nachtragen.

Das gleiche Verfahren gilt nicht nur für das Kernmodul, sondern auch für die einzelnen Module. Die Update-Scripte liegen jeweils in

\$SUPERX_DIR/db/module/<<Modulname>>/<<Modulname>>_modul_upgrade<<Versionsnr>>.x

2.2.2 Kurzanleitung zum Upgrade von Version 2.1 nach 3.0

Die folgende Kurzanleitung zeigt die wesentlichen Schritte für den Upgrade von 2.1 nach 3.0.

- 4. Stoppen Sie tomcat über \$SUPERX_DIR/webserver/tomcat/bin/shutdown.sh
- 5. Sichern Sie den alten Tomcat mv \$SUPERX_DIR/webserver/tomcat \$SUPERX_DIR/webserver/tomcat_alt
- 6. Sichern Sie das gesamte Verzeichnis \$SUPERX_DIR
- 2. Entpacken Sie das neue Kernmodul unter \$SUPERX_DIR cd \$SUPERX_DIR tar -xzvf kernmodul3.0.tar.gz
- 7. Wenn Sie Datenbank- und Webserver auf unterschiedlichen Rechnern betreiben, müssen Sie das Kernmodul auf beiden Rechnern entpacken.

Die alten Scripte werden dadurch überschrieben, nicht aber die vorhandenen Properties-Dateien (unsere properties-Dateien sowie auch das Access-Frontend haben immer die Endung .sam für sample).

2.2.2.1 Der Datenbankupgrade

- 1. Gehen Sie in das Verzeichnis $SUPERX_DIR/db/bin$
- 2. Kopieren Sie die Datei $\operatorname{SQL}_{\operatorname{ENV}}$ nach $\operatorname{SQL}_{\operatorname{ENV}}$ alt
- 3. Kopieren Sie SQL_ENV.sam nach SQL_ENV
- 4. Übertragen Sie relevante Parameter aus SQL_ENV.alt nach SQL_ENV, z.B. SX_CLIENT, JAVA_HOME, DATABASE, MAILPROG.
- 5. Sourcen Sie die neue SQL_ENV mit . ${\tt SQL_ENV}$
- $6. \ Gehen \ Sie \ in \ das \ Verzeichnis \ \texttt{SUPERX_DIR/db/install}$
- 7. Sichern Sie die Datenbank mit dump_it.x
- 8. Gehen Sie in das Verzeichnis upgrade
- 9. Starten Sie das Script kernmodul_upgrade21_to_30.x, und prüfen Sie auf Fehler in der Logdatei upgrade.log
- 10.Entfernen Sie aus dem Themenbaum den "alten" Ast Administration. Sie erkennen diesen daran, dass im Klammern (alt) dehinter steht. Wenn Sie eigene Masken dort erzeugt haben, müssen Sie diese in den neuen Ast verschieben.

11.Richten Sie bei Bedarf das neue Access-Frontend (S. 40) ein (Kopieren von

$\texttt{SUPERX_DIR/db/superx_frontend_sam.mdb} nach \texttt{superx_frontend.mdb}.$

2.2.2.2 Upgrade des Webservers

- 1. Kopieren Sie die Dateien superx.properties und db.properties cp \$SUPERX_DIR/webserver/tomcat_alt/webapps/superx/WEB-INF/db.properties \$SUPERX_DIR/webserver/tomcat/webapps/superx/WEB-INF/db.properties cp \$SUPERX_DIR/webserver/tomcat_alt/webapps/superx/applet/superx.properties \$SUPERX_DIR/webserver/tomcat/webapps/superx/applet/superx.properties
- 2. Nur bei Informix: Stellen Sie sicher, dass sich eine aktuelle ifxjdbc.jar unter \$SUPERX_DIR/webserver/tomcat/webapps/superx/WEB-INF/lib befindet. Achtung: Es ist eine neuere Version als für SuperX 2.1 nötig!
- 3. Benennen Sie die Datei \$SUPERX_DIR/webserver/tomcat/webapps/superx/WEB-INF/kern_dbforms_config_<<DB-Kürzel>>.xml um nach \$SUPERX_DIR/webserver/tomcat/webapps/superx/WEB-INF/dbforms_config.xml
- 4. Tragen Sie in der Datei \$SUPERX_DIR/webserver/tomcat/conf/server.xml im Kontext superx die JDBC-Parameter nach (url, username, password, maxIdle).
- 5. Starten Sie Tomcat neu, und testen Sie die Anmeldung im XML-Frontend und im Applet
- 6. Mit der neuen Webanwendung ist auch eine neue web.xml im Verzeichnis \$SUPERX_DIR/webserver/tomcat/webapps/superx/WEB-INF/web.xml installiert worden Wenn Sie die maxRows geändert haben, dann sollten Sie die Änderungen manuell von der tomcat-Datei nachtragen.

Der Upgrade von SuperX-Karlsruhe (Win32-Client und Informix-Datenbank) gestaltet sich anders und wird im folgenden (S. 82) beschrieben.

2.2.3 Upgrade von SuperX Karlsruhe auf SuperX V2.0

Das Kernmodul der SuperX-Version 2 ist abwärtskompatibel mit der SuperX-Anwendung aus Karlsruhe, wenn ein paar Änderungen an der Datenbank vorgenommen werden. Bevor das neue Servlet und das neue Applet eingesetzt werden, sollten die notwendigen Tabellen erzeugt worden sein. Der Upgrade ist leider nicht automatisch machbar, weil frühere SuperX-Installationen sehr unterschiedlich sind. Es ist daher möglich, dass einige Scripte mit Fehler abbrechen.

2.2.3.1 Erzeugen der Tabellen

Sie erzeugen bzw. erweitern Tabellen und Prozeduren in der SuperX-Datenbank. Diese Änderungen sollten nicht im laufenden Betrieb von SuperX vorgenommen werden, da die neuen Tabellen, z.B. bis auf userinfo und user_institutionen vom alten SuperX benutzt werden. Außerdem sollten Sie die Regel beherzigen, zunächst den bestehenden Stand der Datenbank mit dbexport zu sichern.

Bisherige SuperX-Benutzer können mit der vorliegenden SuperX-Version ihre Masken weiterverwenden. Dazu müssen die Masken und Sachgebiete in den Themenbaum übernommen werden. Dazu liegt ein Script im Verzeichnis

Installation

\$SUPERX_DIR/db/install/update_von_superx_karlsruhe/einmal_starten/ themenbaum_import_superxalt.sql

Außerdem können bisherige SuperX-Benutzer mit der vorliegenden SuperX-Version ihr Organigramm weiterverwenden. Dazu müssen neben den Masken (S. 152) die Tabellen Bereiche, Einrichtungen, Institutionen und Abteilungen in das Organigramm übernommen werden. Dazu liegt ein Script im Verzeichnis

\$SUPERX_DIR/db/install/update_von_superx_karlsruhe/einmal_starten/ organigramm_import_superxalt.sql

Die Tabelle hochschulinfo darf nur mit einem Datensatz, dem Namen und der Nummer der Hochschule gefüllt sein. Das Script fügt auch die Lehreinheiten unterhalb der Hochschule unter dem Knoten "Lehreinheiten" hinzu; dadurch können Sie mit dem Admin-Tool die Lehreinheiten einfacher in das Organigramm übernehmen. Es bleibt dabei Ihnen überlassen, ob Sie die Lehreinheiten in die Institutionsstruktur einfügen oder nicht.

Die Tabelle userinfo wird um zwei deskriptive Felder erweitert: "name" für den Benutzernamen, und "info" für die Beschreibung der Person. Außerdem wird das Feld passwd_sha hinzugefügt, und als Default das Passwort "anfang12" gesetzt.

```
$SUPERX_DIR/db/install/update_von_superx_karlsruhe/einmal_starten/
userinfo_import_superxalt.sql
```

Die Tabelle user_institution im "alten" SuperX wurde erweitert um die Felder gueltig_seit, guel-

tig_bis und lehre; das Script

\$SUPERX_DIR/db/install/update_von_superx_karlsruhe/einmal_starten/ user_institution_import_superxalt.sql

konvertiert die Tabelle in das neue Format.

Die Tabelle felderinfo im "alten" SuperX wurde geändert: das Feld default lautet nun defaultwert; dadurch ist SuperX mit anderen Datenbanken kompatibel, z.B. PostgreSQL; das Script

\$SUPERX_DIR/db/install/update_von_superx_karlsruhe/einmal_starten/

felderinfo_import_superxalt.sql

konvertiert die Tabelle in das neue Format. Achtung: Der darin enthaltene Passus, das in der Tabelle felderinfo auch das Feld default gelöscht wird, macht SuperX inkompatibel zum Win32-Client aus Karlsruhe. Andererseits funtionieren ohne droppen des Feldes die Scripte sx_select_mask und sx_insert_mask nur noch mit eigenen Abfragen, nicht mit Download-Versionen anderer Hochschulen.

Der Tabelle maskeninfo im "alten" SuperX wurde das Feld hinweis hinzugefügt. Die ermgöglicht es, zu Ergebnistabellen Legenden zu liefern, die im Kopf der Tabelle angezeigt werden; das Script \$SUPERX_DIR/db/install/update_von_superx_karlsruhe/einmal_starten/ maskeninfo_import_superxalt.sql

konvertiert die Tabelle in das neue Format.

Die Tabellenübernahme - Kurzanleitung:

Starten Sie im DBACCESS nacheinander die folgenden Scripte im Verzeichnis

\$SUPERX_DIR/db/install/update_von_superx_karlsruhe/einmal_starten/

(Sicherheitsabfragen können Sie ggf. mit Ja beantworten):

themenbaum_import_superxalt.sql
organigramm_import_superxalt.sql
userinfo_import_superxalt.sql
user_institution_import_superxalt.sql
felderinfo_import_superxalt.sql
maskeninfo_import_superxalt.sql
macro.sql

Sie erzeugen so die Tabellen Themenbaum, Organigramm und erweitern die anderen Tabellen.

2.2.3.2 Erzeugen der Prozeduren

Der Themenbaum benötigt keinerlei Prozeduren, aber das Organigramm benötigt die Prozeduren sp_user_orga.sql sp_user_orga_child.sql

Die Prozeduren liegen im Verzeichnis

\$SUPERX_DIR/db/install/update_von_superx_karlsruhe/prozeduren/

Der Dateiname beginnt wie in SuperX üblich mit proc_. Erzeugen Sie diese Prozeduren im DBACCESS.

2.2.3.3 Upgrade des Servlets und Applets

Das SuperX-Servlet ist nicht allein updatebar, bei einem Upgrade muss das gesamte Verzeichnis \$SUPERX_DIR/webserver gesichert, gelöscht, und dann neu entpackt werden, wie in der Installationsanleitung des Webservers (S. 53) beschrieben. Danach müssen die Dateien db.properties sowie superx.properties vom "alten" Webserver zum neuen Webserver kopiert werden.

2.2.3.4 Ändern der Masken

Die alten SuperX-Masken sind problemlos im neuen SuperX lauffähig. Falls Sie schon den Karlsruher SuperX-Client (WIN32-Anwendung) in Betrieb haben, müssen Sie beachten, dass sich der alte Client teilweise anders als der Java-Client verhält. Umgekehrt ist das neue SuperX-Kernmodul mit Version 2.0 nicht mehr kompatibel mit dem alten Client. Beide Clients können also lediglich für Spezialabfragen parallel betrieben werden (z.B. aus 'historischen' Gründen). Beim Anpassen der alten SuperX-Abfragen für SuperX Version 2.0 müssen die folgenden Punkte beachtet werden.

2.2.3.5 Masken-Anpassung für den Java-Client

Die Masken werden je nach Client auf unterschiedliche Arten aufgebaut. Für die Gestaltung der Masken sind die Tabellen *maskeninfo* und *felderinfo* verantwortlich.

2.2.3.5.1 Felderinfo

- Das Feld default wurde umbenannt in defaultwert
- Bei Mehrfachauswahl-Feldern muss als Datentyp immer char gewählt werden
- Bei "char(xx)"-Deklarationen zeigt der Java-Client keine Textfelder an. Man muss die Typdeklaration ändern auf "char"
- Attribut Art: 6 fällt als Zahl weg. Statt 6 wählt man 5 (nur Anzeige) 0 ist die einzige Art, bei der nur ein Feld angegeben wird, ansonsten immer 2; zur Not select semester, semester
- Bei Auswahl Köpfe oder Fälle funktioniert nur der Datentyp sql, nicht char oder integer, bei beiden Clients
- Bei Eingabe eines Datums muss zwingend der Datentyp date gesetzt werden. Leider muss man dies für den alten client im select_stmt der Tabelle maskeninfo mit date("<<feld>>") abfangen

2.2.3.5.2 Maskeninfo

- Der Tabelle wurde das Feld hinweis hinzugefügt, dass es ermöglicht, Ergebnistabellen zu dokumentieren.
- Bei Ergebnistabellen im neuen Client wird nicht mehr automatisch gerundet round(<<Ausdruck>>,2)
- Bei der Vorgabe des Tabellenlayouts (XIL-Proplist) sind unbedingt Leerzeichen zwischen den Attributen erforderlich. Im Zweifelsfall muss man die Beschreibung von einer funktionierenden Maske kopieren.

Auch der Aufruf des Organigramms in den Masken muss geändert werden. Die folgende Zeile im Feld

select_stmt der Tabelle maskeninfo

```
execute procedure sp_ch110_institut (/* <<Institution>> +1 */ -1, <<UserID>>);
  muss ersetzt werden durch
  and a muse and a child (<UserID>>)
```

```
execute procedure sp_user_orga_child (<<UserID>>, <<Organigramm-Stand>>, 0,
    /* <<Institution>> +1 */ -1,<<erlaubt>>);
```

2.3 Checkliste Sicherheitsmassnahmen

Um die Datensicherheit zu verbessern, empfehlen wir folgende Massnahmen:

2.3.1 SSL-Verschlüsselung mit Zertifikat von Trustcenter

Generell sollten Sie SuperX immer mit SSL-Verschlüsselung betreiben, egal ob über Tomcat oder Apache.

Es wird an anderer Stelle (S. 69) beschrieben, wie Sie ein Zertifikat selbst erstellen können, dies sollte nur zu Testzwecken dienen. Lassen Sie stattdessen ein persönliches Zertifikat durch einen kommerziellen Zertifizierungsserver publizieren. Akkreditierte Anbieter von qualifizierten Zertifikaten gemäß Deutschem Signaturgesetz sind die AuthentiDate International AG, verschiedene Bundesnotarkammern, D- TRUST (Bundesdruckerei-Gruppe), DATEV, Deutsche Post, TC Trustcenter, T-Systems und S-Trust Sparkassen-Finanzgruppe.

2.3.2 Applet ggfs. deaktivieren

Falls Sie das Applet nicht benötigen löschen Sie das Verzeichnis tomcat/webapps/superx/applet.

Außerdem wird die Sicherheit erhöht, wenn ein Zugriff auf das nur vom Applet benutzte Servlet SuperXDBServlet unterbunden wird.

Bearbeiten Sie dazu Ihre tomcat/webapps/superx/WEB-INF/web.xml

Um si

2.3.3 Public-Private-Key-Kontrolle von Applet-Befehlen

Das Applet genügt von seiner Anlage her nicht mehr den modernen Sicherheitsanforderungen und wird langfrisitg durch das XML-Frontend ersetzt. Zur Erhöhung der Sicherheit ist es möglich, eine DSApublic/private-Key-Kontrolle (S. 58) zu installieren. Dabei wird jeder Befehl, der vom Applet ans Servlet geschickt wird, mit dem einen Key signiert. Im Servlet wird mit Hilfe des anderen, nur dort bekannten Keys kontrolliert, ob der ankommende Befehl eine gültige Signatur aufweist.

Im SuperX-Applet können Sie den Info-Button anklicken, in der erscheinenden Infobox wird angegeben, ob public/private key Kontrolle aktiv ist oder nicht.

2.3.4 Datenbankverbindung über einen eingeschränkten Datenbank-User

Zur Erhöhung der Sicherheit ist es möglich, dass die Datenbankverbindung von Tomcat zur Datenbank mit einem eingeschränkten User durchgeführt wird. Dies wird von ZENDAS (Zentrale Datenschutzstelle der baden-württembergischen Universitäten) für den Produktivbetrieb nachdrückliche empfohlen.

Richten Sie dazu einen entsprechenden eingeschränkten User in Ihrer Datenbank ein und geben Sie diesen beim Propadmin bei *eingeschränkter User* an. Der erste im Propadmin auszufüllende User muss weiterhin umfassende Rechte auf alle Tabellen haben, weil er auch bei Modulinstallationen/-updates verwendet wird. Das Minimum, was der eingeschränkte User haben muss sind select Rechte auf alle Tabellen, insert-Rechte auf die Tabelle user_pw und protokoll (sowie bei Postgres auf die zugehörige Sequence protokoll_protokoll_id_seq) und update-Rechte auf userinfo.

Sobald Sie Ihre db.properties mit dem Propadmin bearbeitet haben, können Sie praktisch die Minimal nötigen Rechte vergeben, in dem Sie einmal das Skript

sx_restrictedcommanager.x false aufrufen. Rufen Sie dieses Skript erneut auf nach Modulneuinstallationen oder Upgradepatches.

Installation

Nach einem Tomcat-Neustart findet sich in der catalina.out nach "Aufbau von Datenbank-ConnectionPool (..) .. OK" ein Hinweis:

eingeschränkter Datenbankuser für Verbindung: true|false

Wenn Sie Funktionen wie User/Gruppe/Maske einrichten/löschen etc. im XML-Frontend benutzen wollen, müssen zusätzliche Kernmodultabellen freigeschaltet werden: protokoll userinfo groupinfo user_institution user sachgeb bez user masken bez group_sachgeb_bez group_masken_bez user_group_bez user_pw user_sichten user_sichtarten group-sichten group_sichtarten felderinfo maskeninfo maske_system_bez masken felder bez sachgeb_maske_bez organigramm themenbaum

Am einfachsten können Sie dies erledigen, indem Sie das Skript

sx_restrictedconnmanager.x true aufrufen.

Exkurs:

Wenn Sie die höchste Sicherheit wollen, aber der Zuständige für die Userverwaltung trotzdem das

XML-Frontend benutzen können soll, könnten Sie folgendermaßen vorgehen:

- Richten Sie für den regulären SuperX-Betrieb einen eingeschränkten User mit minimalen Rechten ein, wie oben beschrieben und deaktivieren Sie alle Datenbankformulare, indem Sie nach jedem Modulupgrade das Verzeichnis tomcat/webapps/superx/edit leeren.
- Erzeugen Sie einen weiteren eingeschränkten Datenbankuser, der zusätzlich die Kernmodultabellen bearbeiten darf.
- Richten Sie einen zweiten Tomcat ein, der mit diesem zweiten eingeschränkten Datenbankuser arbeitet.
- Sorgen Sie (z.B. per Firewall) dafür, dass nur der für die Userverwaltung zuständige Mitarbeit Zugriff auf den zweiten Tomcat hat.

2.3.5 Einstellungen zur Passwortsicherheit

Bitte folgen Sie den Empfehlungen von Zendas zur Passwortsicherheit (S. 116). Die Zentrale Datenschutzstelle der baden-württembergischen Universitäten macht folgende Empfehlung:

Passwortgültigkeit (Tage)	90-180
Passwort Groß- u. Kleinb.	1
Passwort erfordert Ziffer	1
Passwortlänge (Minimum)	8

Diese Einstellungen können in der Konstanten-Tabelle geändert werden.

Der SuperX-Administrator kann erzwingen, dass der Benutzer sein Passwort ändern muss, indem er im XML-Frontend den entsprechenden User bearbeitet und bei "User muss Passwort ändern" ein Häkchen setzt.

2.3.6 Servertrennung für maximale Sicherheit

Für maximale Sicherheit empfiehlt es sich physikalisch getrennte Server für Apache, Tomcat und die Datenbank zu betreiben. Des Weiteren sollten die Server durch eine Firewall abgeschottet werden, normalerweise steht der Apache-Server in der DMZ.

2.3.7 Keine Verwendung von Standardkennungen

Verwenden Sie nach Möglichkeit nicht die Standardkennungen (superx, admin und testuser), die bei Auslieferung im SuperX-Kernmodul enthalten sind. Richten Sie eine bzw. mehrere neue Administrator-Kennungen an und arbeiten mit diesen. Die Standardkennungen superx, admin, testuser sollten aus der SuperX-Tabelle userinfo gelöscht werden.

2.3.8 Keine Anzeige von internen Details bei Fehlermeldungen

Schalten Sie den Entwicklungsmodus aus, damit keine detaillierten Fehlermeldungen bei SQL-Abfragen nach außen angezeigt werden.

Verwenden Sie dazu das grafische Tool Propadmin (propadmin.x) oder setzen Sie in \$SUPERX_DIR/webserver/tomcat/webapps/superx/WEB-INF/db.properties "developmentMode" auf "false".

Für das SuperX-Applet setzen Sie in

 $\texttt{SUPERX_DIR/webserver/tomcat/webapps/superx/applet/superx.properties} \ \texttt{,logToKonsole"} auf \ \texttt{,none"}.$

Eine weitere Verschleierungstechnik stellt das Aktivieren einer eigenen allgemeinen Fehlermeldung dar. Dadurch soll verhindert werden, das weitere Details – wie zum Beispiel der Java-Stacktrace – angezeigt wird. Richten Sie dafür eine Html-Datei error.htm ein, in der könnte zum Beispiel stehen:

"Es ist ein Fehler aufgetreten. Bitte wenden Sie sich an superx@universitaet.de"

Installation

Editieren Sie anschließend \$SUPERX_DIR/webserver/tomcat/webapps/superx/WEB-INF/web.xml und fügen an das Ende der Datei (also vor dem Endtag </web-app>) folgenden Abschnitt ein:

```
<prror-page>
<prror-code>500</pror-code>
<location>/error.htm</location>
</pror-page>
```

2.3.9 Directory-Listing in Tomcat/Apache abschalten

Kontrollieren Sie, dass die Funktion Directory-Listing sowohl im Apache als auch im Tomcat abgeschaltet ist.

In Tomcat muss in der Datei tomcat/conf/web.xml im folgenden Abschnitt der Eintrag für listings auf false gesetzt werden.

<servlet>

```
<servlet-name>default</servlet-name>
```

```
<servlet-class>org.apache.catalina.servlets.DefaultServlet</servlet-class>
```

<init-param>

<param-name>debug</param-name>

<param-value>0</param-value>

- </init-param>
- <init-param>

<param-name>listings</param-name>

<param-value>false</param-value>

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

<load-on-startup>1</load-on-startup>

</servlet>

2.3.10Sperren der DBFORMS-Komponente

Die DBFORMS-Komponente des XML-Frontends dient nur der Datenbankadministration und kann daher in Produktivsystemen mit WWW-Anbindung deaktiviert (S. 48)werden. Damit sind von außen keine Datenbankverbindungen mehr möglich.

Eine Abschaltung der DBFORMS beeinträchtigt in keiner Weise die "normalen" Funktionen zur Berichtserstellung von SuperX. Wenn die DBFORMS-Komponente benötigt wird, kann eine SuperX-Installation der betreffenden Module in einem separaten Tomcat auf einem lokalen oder besonders geschützten System stattfinden, wo die DBFORMS dann freigeschaltet werden können.

2.3.11 Kontrolle von (fehlerhaften) Anmeldungen

Kontrollieren Sie die Tabelle protokoll auf Häufung von Fehlanmeldungen (proto_fkt_id=2), z.B. per cron-job.

Wenn Sie zusätzlich erfolgreiche Anmeldungen loggen wollen, setzen Sie den Eintrag "Erweitertes Protokoll" in der konstanten-Tabelle auf apnr=1 und starten Sie Tomcat neu.

2.3.12Logging von Aktivitäten im Adminbereich (dbforms)

Das Logging für DBFORMS wird in der Datei \$SUPERX_DIR/webserver/tomcat/webapps/superx/WEB-INF/log4j.properties festgelegt. Passen Sie die Datei entsprechend Ihrer Erfordernisse an.

2.3.13Entfernen von temporären Dateien

Entfernen Sie temporäre Dateien, die sich auf dem Webserver befinden (z.B. mit Endung ~ oder #Untitled#). In \$SUPERX_DIR/db/bin steht das Skript remove_tmp.x zur Verfügung. Es entfernt automatisch alle Dateien mit den Endungen ~, tmp und bak sowie #Untitled#-Dateien aus dem aktuellen Verzeichnis und dessen Unterverzeichnissen. Optional kann auch ein Pfad angegeben werden, in dem die Dateien gelöscht werden sollen, z.B.: remove_tmp.x /home/superx/webserver/tomcat/webapps/superx

2.3.14Kontrolle des Referers

Zur Steigerung der Sicherheit kann eingestellt werden, dass SuperX bei aufgerufenen Links kontrolliert, ob auch ein bestimmter Referer im Request-Parameter enthalten ist.

Um diese Funktion zu aktivieren, fügen Sie den folgenden Abschnitt als zusätzlichen Parameter zum SuperXManager-Servlet in der tomcat/webapps/superx/WEB-INF/web.xml hinzu:

```
<init-param>
<param-name>referer_start</param-name>
<param-value>https://superx-webserver/superx</param-value>
</init-param>
```

Es ist allerdings zu berücksichtigen, dass einige Browser erlauben, die Übermittlung des Referrers zu deaktivieren. Auch Content-Filter (sowohl im Browser als auch auf Proxys) können entsprechend eingestellt werden. Ein Aufruf mittels IP-Nummer statt Rechnername würde ebenfalls dann fehlschlagen.

2.4 Das SuperX-Clientpaket

Wenn Sie nicht das gesamte Kernmodul inkl. Tomcat benötigen, sondenr nur eine kleines Paket, um auf SuperX zuzugreifen, haben wir ein SuperX-Clientpaket "geschnürt", das die wichtigsten Werkzeuge zur die Arbeits mit dem Server beinhaltet. Insdbesondere Windows-Anwender können dieses Paket benutzen, um mit SuperX zu arbeiten, z.B. Masken entwickeln, Tabellen entladen etc. Dazu enthält das Paket ein paar Werkzeuge.

Das Client-Paket wird außerdem für das Entladen aus HIS-Systemen unter Windows genutzt.

2.4.1 Installation

Laden Sie das Paket superxclient<<Versionsnummer>>.zip und speichern Sie es lokal auf der Festplatte. Wenn Sie das Clientpaket direkt auf dem SuperX-Datenbankserver installieren wollen, entpacken Sie das Paket in \$SUPERX_DIR (Achtung: Die Versionsnummern von SuperX-Kernmodul und SuperX-Clientpaket müssen dann übereinstimmen!). Ind iesem Falle können Sie die Umgebungsvariablen in SQL_ENV übernehmen.

Wenn das Clientpaket auf einem anderen Rechner laufen soll, entpacken Sie dieses Archiv in einem separaten Verzeichnis, z.B. c:\superx, und benennen Sie die Dateien um; unter Windows:

db\bin\sql_env_superxclient.bat.sam nach db\bin\sql_env.bat

```
bzw. unter Unix:
```

```
db/bin/SQL_ENV_superxclient.sam nach db/bin/SQL_ENV
```

Passen Sie die Parameter in der Datei an, und sorgen Sie dafür, daß die Datei beim Aufruf der Shell ausgeführt wird (unter DOS autoexec.bat, unter Linux / bash die ~/.bashrc).

Folgende Parameter müssen Sie wahrscheinlich anpassen:

JAVA_HOME (Der Pfad zur JRE)

SUPERX_DIR (das Verzeichnis, in dem Sie den Client entpackt haben)

Folgende Parameter sind wichtig, aber für SuperX-User meist korrekt vorbelegt:

JDBC_CLASSPATH (der Pfad zu Ihrem jdbc-Treiber, Postgres und hsqldb 1.7.1 sind bereits vorhanden, Informix (ifxjdbc.jar) muss manuell nachinstalliert werden)

DB_PROPERTIES (der Pfad zu den Datenbankparametern)

Starten Sie zunächst den propadmin mit

propadmin.bat (DOS)

bzw.

propadmin.x (Linux)

Dort geben Sie die Parameter für den DB-Zugriff ein. Das Passwort wird verschlüsselt gespeichert. Danach sind die Kommandozeilen-Werkzeuge verfügbar.

2.4.2 Werkzeuge

Im Clientpaket befinden sich verschiedene Werkzeuge. Einerseits können Sie mit dem Clientpaket aus Windows-Datenbanken (S. 92) entladen, und andererseits befinden sich im Ordner Tools die Anwendungen Jedit (S. 204)sowie die sqlWorkbench (S. 112) sowie das Access-Frontend (S. 121). Diese Tools dienen zur Abfragenentwicklung. Details dazu finden Sie im SuperX-Entwicklerhandbuch.

2.4.3 Installation von Entladescripten auf Basissystemen

Nach der Installation des SuperX-Datenbankservers müssen die Datenquellen für SuperX eingerichtet werden. Dies sind andere relationale Datenbanken, für die in der Regel Entladescripte mit den SuperX-Modulen ausgeliefert werden. Die Scripte wiederum müssen eingerichtet werden. Die jeweiligen Anleitungen finden Sie in den einzelnen Modulen. Ein Spezialfall ist das Entladen aus Access, das im folgendne dokumentiert wird.

2.4.4 Entladen aus Windows-Datenbanken: Access

Beim Entladen aus Windows-Datenbanken wie z.B. Access wird der SuperX-jdbc-Client verwendet. Dieser wiederum befinden sich im SuperX-Clientpaket (S. 90). Zur Anbindung des jdbc-Client an eine Access-Datenqelle kann auch der propadmin (S. 96) genutzt werden. Wichtig bei Access ist dabei die Einrichtung der Benutzerberechtigung.

Der Arbeitsgruppenadministrator ist in Access 2000 etwas versteckt, deshalb muss man zunächst nach der Datei wrkgadm.exe im Office Ordner suchen. Anschließend die Datei aufrufen und unter "Beitreten" die Arbeitsgruppen-Informationsdatei angeben.

Die **mdw** brauchen Sie wahrscheinlich für die Berechtigung, dh. Sie müssen in Access unter Extras/Sicherheit/Arbeitsgruppenadministrator diese Datei angeben.

Wenn Sie zum Entladen auf einem Windows-basiertes Datenbank-System ohne eigenen Datenbankserver (z.B. Microsoft Access) zugreifen wollen, müssen Sie auf dem Windows-Rechner eine Applikation starten, die die ODBC-Verbindung als Serverapplikation verfügbar macht; in SuperX nutzen wir dazu den Rmi-JDBC-Treiber der Firma ObjectWeb, der Teil des Kernmoduls ist. Installieren Sie dazu das SuperX-Kerndmoul oder den SuperX-Client Version 2.1 oder höher auf dem Windows-Rechner, und starten Sie den RMI-Server in der DOS-Box mit \$SUPERX_DIR\db\bin\start_rmiodbc_server.bat.

Danach können Sie von einem entfernten Rechner mit dem Propadmin darauf zugreifen, indem Sie den Datenbanktreiber rmiOdbc auswählen; die Verbindungsparameter sind dabei nach dem folgenden Muster anzugeben:

jdbc:rmi://<rmiHostName[:port]>/<jdbc-url>

Der Treibername ist org.objectweb.rmijdbc.Driver. Wenn Sie zum Beispiel auf den Windows-Rechner mit dem Namen cobserver zugreifen wollen und dort COB unter Access installiert ist, und die odbc-Quelle dort cob6 heisst, dann lautet der Pfad:

jdbc:rmi://cobserver/jdbc:odbc:cob6

Wen der RMI-Server permanent laufen soll, müssen Sie die Sicherheitshinweise der Firma ObjectWeb beachten.

Wenn Sie den RMI-Server nicht nutzen wollen, können Sie auch auf dem jeweiligen Windows-Rechner entladen. Bei installiertem und konfiguriertem Clientpaket (S. 90)entpacken Sie das jeweilige Modul (.z.B. das SOS-Modul sos_modul<<Versionsnr>>_pg.tar.gz) im gleichen Verzeichnis. Dann gehen Sie in der DOS-Shell nach db\module\sos\rohdaten, und konfigurieren die Umgebung in sos_env.bat (Vorlage liegt im gleichen Verzeichnis mit der Endung .sam). Danach können Sie das DOS-Entladescript sos_unload.bat ausführen. Die Rohdaten werden dann nach unl geschrieben und müssen dannauf den SuperX-Server kopiert werden.

3 Administration des Kernmoduls: HowTo

Im folgenden werden zentrale Arbeitsschritte beim Betrieb von SuperX beschrieben. Für einen Blick auf den Hintergrund sollten Sie sich ggf. die Bestandteile (S. 141) anschauen.

Zuächst zeigen wir, wie die Frontends funktionieren, und dann beschreiben wir die Werkzeuge für die Administration von SuperX.

SuperX verfügt über unterschiedliche Benutzeroberflächen, hier "Frontends" genannt. Das SuperX-Applet dient dem allgemeinen Berichtswesen und liefert vordefinierte Ergebnistabellen. Die Installation des Applets auf den Clients ist in der Installationsanleitung (S. 73) beschrieben. Die Funktionsweise des Applets ist ausführlich in dem Benutzerhandbuch dokumentiert.



Das XML-Frontend (S. 124) liefert komplexe Berichte, die aus mehreren Ergebnistabellen zusammengestellt werden, und die flexibel für verschiedene Ausgabegeräte und -formate aufbereitet werden können. Im Gegen-(S. 124) satz zum Applet sind keinerlei Installationsschritte notwendig, es genügt ein html4-fähiger Browser. Derzeit ist das XML-Frontend noch im Betastadium.



Joolap bietet die Möglichkeit, multidimensionale Auswertungen zu machen und Statistiken flexibel den eigenen Bedürfnissen anzupassen. Joolap wird mit einer eigenen Dokumentation ausgeliefert.

3.1 **Die SuperX-Administrationswerkzeuge**

Die Verwaltung des Organigramms und des Themenbaums sowie grundlegende User- und Gruppenverwaltung lässt sich mit Hilfe eines graphischen Administrationswerkzeugs SuperXAdmin (Betaversion 1.0) sowie über ein Access-Frontend erledigen. Es gibt neben der Shell-Zugang über UNIX zwei Administrationswerkzeuge für das Kernmodul: Browser-basierte Formulare im XML-Frontend (S. 104), die auf die DBFORMS-Technologie zurückgreifen. Außerdem wurde ein Access-Frontend entwickelt, dass über ODBC-Verknüpfung einen direkten Zugriff auf die SuperX-Tabellen liefert. Das Browser-basierte Frontend hat den Vorteil, dass es auch über eine http-Verbindung arbeitet und somit höhere Sicherheitsstandards erfüllt. Das Access-Frontend eignet sich besser für die direkte Bearbeitung einzelner Tabellen und für die Entwicklung von Abfragen. Die Funktionalität ist ansonsten identisch, deshalb wird im folgenden nur die Oberfläche des Browser-Frontends beschrieben. Lediglich die Abfragenbearbeitung mit dem Access-Frontend (S. 121) wird gesondert dargestellt.

3.1.1 Übersicht über Scripte unter UNIX

Für die Administration des DataWarehouse sind Shellscripte vorbereitet, die flexible Werkzeuge zur Datenbankadministration bereitstellen. Die Shellscripte werden in den Update-Scripten aufgerufen, können aber auch zur manuellen Administration benutzt werden. Die wichtigsten Bereiche sind die Masken-Verwaltung und die Ladescripte im Umgang mit Tabellen sowie allgemeine Scripte.

Alle Scripte befinden sich unter \$SUPERX_DIR/db/bin, deshalb muss dieser Pfad inder Umgebungsvariable PATH enthalten sein. Die Scripte wurden unter UNIX entwickelt (ohne Endung oder Endung .x), einige davon sind auch nach DOS portiert worden (erkennbar an der Endung .bat).

Einige Scripte lauten "sx_auto_"..., dies bedeutet, dass die Scripte ohne Sicherheitsabfrage ausgeführt werden.

Voraussetzung für den Ablauf der Scripte ist die Eintragung der korrekten Umgebungsvariablen in \$SUPERX_DIR/db/bin/SQL_ENV bzw. \$SUPERX_DIR\db\bin\sql_env.bat. Wenn der Client jdbc verwendet wird, muss ausserdem die korrekte DB_PROPERTIES gesetzt sein.

3.1.1.1 Die Umgebungssteuerung: SQL_ENV

Das Script \$SUPERX_DIR/db/bin/SQL_ENV steuert die Umgebung und ist für den Betrieb der Scripte unverzichtbar. Einige Variablen sind vorbelegt, Beispiele sind auf Kommentar gesetzt. Da die Umgebung von dem System abhängt, muss jeder Anwender die Werte manuell pflegen. Bei einem Update des SuperX-Kernmoduls wird diese Datei nicht überschrieben, lediglich sein SQL_ENV.sam im gleichen Verzeichnis. Von dort müssen relevante Änderungen dann in die "richtige" SQL_ENV manuell übernommen werden. Informix- und Postgres-spezifische Variablen sind in dem Kapitel zur Installation und Konfiguration (S. 21) der Datenbankserver beschrieben.

Folgende Variablen sind auf jeden Fall zubelegen:

SUPERX_DIR	Der Installationspfad von SuperX
DATABASE	Das Datenbanksystem ("POSTGRES" / "INFORMIX")
DBNAME	Der Name der Datenbank (standardmäßig "superx").
SX_CLIENT	Die Clientanwendung (bei Postgres "psql", bei Informix
_	"dbaccess"). Ein client namens "jdbc" ist generisch und dient
	dem Zugriff auf beliebige DB-Systeme, für die idbc-Treiber
	existieren.
	Der idbc-Client wurde bisher mit Informix. Postgres und
	hsaldb getestet – die idbc-Treiber für Informix und Postgres
	werden mitgeliefert und dürfen auf keinen Fall durch andere
	ersetzt werden
MAILPROG	Das Mailprogramm unter UNIX z B mutt oder mail: dies
	muss sich im $PATH$ des users superv befinden
LOCMAII /FRRORMAII	Die superv-weite Mailadresse an die Logdateien von ETL-
LOOMAIL/EXKORMAIL	Scrinten geschickt werden
IAVA HOME	Installationspfad der Java-Runtime Das Unterverzeichnis hin
JAVA_HOME	muss in den PATH aufgenommen werden
IAVA OPTS	Java-Runtime-Ontionen z B RAM
JAVA_0115	Bei Finsetz unter Cygwin muss gaf folgende Finstellung
	gemacht worden:
	$YATAN PATH=`cropath =-path =-windows r^2$
	"\$JAVA HOME"/jre/lib/endorsed/xalan2-6-0.jar`
	JAVA_OPTS="-Xmx200M -Djava.awt.headless=true +
	-Xbootclasspath/p:"\$XALAN_PATH
DB_PROPERTIES	Der Pfad zur DB_PROPERTIES (S. 45), standardmäßig
	\$SUPERX_DIR/webserver/
	tomcat/webapps/superx/WEB-INF/db.properties
MANDANTID	Mandantennummer (Hochschulnummer) bei mandantefahi-
D ¹ A 1 A A A A A A A A A A	gen Installationen
Die folgenden Umgebungsvariablen	sind nur fur den JDBC-Client sowie für Postgres relevant:
LOGGING_PROPERTIES	Logging-Parameter für den jdbc-Client. Voreingestellt ist
	"WARNING", mehr Ausgaben erhalt man mit "FINE"
PG_HOST	Name des Postgres-Servers (fur Postgres unter Windows)
Die folgenden Umgebungsvariablen	werden wahrscheinlich nicht geandert (sollten sie auch nicht):
DBDELIMITER	Standardmaßig "A"
PATH	Der PATH wird erweitert um das Verzeichnis
IDRC CLASSDATH	./: your text (u) uit
JDDC_CLASSIAIN	men
YMI CLASSDATH	Der Pfad zu den XML-Tools (Valan, Verges & co)
AWIL_CLASSI ATH	Dei I Iau Zu uch AiviL-10015 (Aalall, Actees & CO).

Die Datei sollte unter UNIX in jedem Aufruf der shell "gesourced" werden, z.B. durch den Befehl: . ~/db/bin/SQL_ENV

(Leerzeichen zwischen Punkt und Tilde!) in der Datei ~/.bashrc.

Wenn Sie unter Windows den jdbc-Client nutzen, dann müssen Sie die Datei als erstes in der DOS-Shell aufrufen, bzw. in definierten Tasks am Anfang aufrufen.

3.1.1.2 Allgemeine Scripte

Zum Absetzen beliebiger SQL-Kommandos werden die Befehle DOSQL und DOQUERY genutzt.

3.1.1.2.1 DOSQL

Shellvariablen setzen und SQL-Anweisung(en) in der der Datei (als Parameter) in der SuperX-Datenbank ausfuehren.

SyntaxDOSQL "Dateiname mit sql-Anweisung (en) " header
(true, false) (optional) Ausgabedatei (optional)BeispielDOSQL "/home/superx/db/isql/test.sql" false "^" output.txt

Das Ergebnis kann mit Feldüberschriften (header=true) in eine Datei Ausgabedatei ausgegeben werden

3.1.1.2.2 DOQUERY

Shellvariablen setzen und eingegebene SQL-Anweisung (als Parameter) in der SuperX-Datenbank ausfuehren.

SyntaxDOQUERY "sql-Anweisung" header (true, false) (optional) Delimi-
ter (optional) Ausgabedatei (optional)BeispielDOQUERY "select name from userinfo" false "^" output.txt

Das Ergebnis kann mit Feldüberschriften (header=true) in eine Datei Ausgabedatei ausgegeben werden.

3.1.1.2.3 sx_transform

Transformiert eine xml-Datei mit einer übergebenen XSL-Datei und gibt das Ergebnis in einen Ausgabekanal aus (stdout oder Datei). Dabei wird der in SuperX integrierte XML-Parser Xerces und der XML-Prozessor Xalan benutzt.

```
Syntaxsx_transform.x IN:<xml-Datei> -XSL:<xsl-Datei> -<br/>OUT:<Ausgabedatei> -method:<Ausgabeformat (text,<br/>xml,html,rtf,pdf)>(optional) <Parameter>(optional)Beispielsx_transform.x -IN:myxml.xml -XSL:myxsl.xsl -OUT:output.htm -<br/>method:html
```

Als Parameter kann ein spezielles Ausgabeformat gewählt werden, z.B. TEXT (siehe Xalan-Doku). Bei rtf wird der RTF-Construktor Jfor auferufen, bei pdf wird FOP aufgerufen. Die *.fo-Datei wird nach tmp.fo geschrieben und dann nach pdf transformiert. Wir gehen also nich davon aus, dass .fo-Dateien die Eingabequellse darstellen.

3.1.1.2.4 Propadmin

Der PropAdmin ist ein kleines Werkzeug, um den Zugriff auf jdbc-Datenbanken zu testen und die Verbindungsparameter (S. 45) in einer übergebenen properties-Datei zu sichern. Wenn keine graphische Umgebung eingerichtet ist, müssen Sie die alle Verbindungsparameter manuell in die db.properties eintragen. Nur das Passwort kann mit dem propadmin bearbeitet werden.

(Musterdateien für Postgres und Inofrmix liegen vor in

 $SUPERX_DIR/webserver/tomcat/webapps/superx/WEB-INF/db-postgres.properties bzw. db-$

96

informix.properties). Wenn als weiterer Parameter kein Dateiname übergeben wird, dann wird die Umgebungsvariable DB_PROPERTIES ausgewertet.

Syntax UNIX

propadmin.x -nogui (optional) <Dateipfad zu db.properties>(optional) Syntax DOS | propadmin.bat <Dateipfad zu db.properties>(optional)

Wenn die Default-Dateiencodierung der aktiven Locale für die Passwort-Verschlüsselung nicht ausreicht, wird eine Fehlermeldung ausgegeben. Unter Windows / DOS ist die Vorbelegung Cp1252 bei deutscher Codepage ausreichend, unter Unix wird die deutsche Locale (S. 27) benötigt.

3.1.1.3 Umgang mit Tabellen

In SuperX werden ständig Tabellen erstellt / geladen / entladen. Zu diesem Zweck wurden Shellscripte entwickelt.

3.1.1.3.1 sx unload table

Entlädt die Inhalte der Tabelle nach <<Dateiname>>(optional) oder <<name>>.unl

Svntax | sx_unload_table.x <<name>> <<Dateiname>> (optional) Beispiel | sx_unload_table.x userinfo

3.1.1.3.2 sx_upload_table

Löscht die Inhalte der Tabelle <<name>>, und lädt die Inhalte einer Datei in die Tabelle mit sx upload records (S. 97). Wenn kein Dateiname übergeben wurde, wird als Name <<name>>.unl angenommen.

> Syntax | sx_upload_table.x <<name>> <<Dateiname>> (optional) **Beispiel** sx_upload_table.x userinfo

3.1.1.3.3 sx_upload_records

Lädt die Inhalte einer Datei in die Tabelle, ohne vorherige Inhalte zu löschen. Wenn kein Dateiname übergeben wurde, wird als Name <<name>>.unl angenommen.

> Svntax | sx_upload_records.x <<name>> <<Dateiname>> (optional) Beispiel | sx_upload_records.x userinfo

Bei Postgres als DB-System wird eine Java-Klasse (de.superx.bin.UnlFileConverter) aufgerufen, die die Unload-Datei entsprechend einer Spezifikation aufbereitet (siehe \$SUPERX_DIR/db/conf/unldescr*).

Wenn der jdbc-Client benutzt wird, können umfangreiche Parameter übergeben werden (Import mit Spaltenüberschriften, Ausgabe von Fehlerprotokollen). Vergleichen Sie die Kommentare im Script.

3.1.1.3.4 sx_schema

Entlädt das Schema einer Tabelle in einem vorgegebenen Format.

Syntax	sx_schema.x sx_schema <tabelle> <format< th=""></format<></tabelle>
J	(pg ids ansi xml HIS))>(optional) <ausgabedatei> (optional)</ausgabedatei>
Beispiel	sx_schema.x userinfo ids myschema.sql
Die Formate	Die Formate sind entweder sql-Scripte für die jeweiligen Daten-
	banktypen (Postgres, Informix, ANSI), die aus der Umgebungs-
	variable DATABASE ausgelesen werden, oder xml bzw. ein
	xml-Format in Anlehnung an die Datenbank-DTD der HIS
	GmbH.

3.1.1.4 Modulverwaltung

Bisherige SuperX-Implementationen sind an den Hochschulen entstanden und haben dementsprechend eine große Vielfalt von Update-Scripten, die jeweils die Vorlieben und Bedingungen der jewieligne Hochschule wiederspiegeln. Daraus ergibt sich für "Neulinge" ein sehr verwirrendes Bild. Außerdem gestaltet sich der Entwurf eines Moduls recht aufwändig, weil die ETL-Funktionen (Extraction -> Transformation -> Loading) manuell programmiert werden müssen. Eine weiteres Problem ist, daß SuperX zukünftig auf zwei Datenbankplattformen lauffähig sein muss, Informix und Postgres.

Das Ergebnis ist: SuperX ist (auf Datenbankseite) sehr fehleranfällig, schwer wartbar und praktisch nicht updatebar.

Mit SuperX Version 2.1 wurde die Verwaltung der Module (Installieren / Aktualisieren / sichern) in zentrale Shellscripte verlagert, die sich ebenfalls in *\$SUPERX_DIR/db/bin* befinden. Die Shellscripte sind dabei nur die operativen "Hüllen" um die eigentlichen SQL-Scripte. Diese wiederum werden zum Teil "von Hand" erzeugt (um z.B. hochschulspezifische Erweiterungen oder Anpassungen vorzunehmen), und zum Teil automatisch aus einer zentralen Steuerdatei

(\$SUPERX_DIR/db/module/<<Modulname>>/conf/<<Modulname>>.xml) jeweils für Postgres und Informix erzeugt.

3.1.1.4.1 module_scripts_create.x

Das Script erzeugt die Installationsdateien für ein Modul, jeweils für Postgres und Informix, nach dem Schema

<<Modulname>>_<<Scriptaktion>>_<<Kürzel für Datenbanksystem>>.sql

Z.B. wird für das BAU-Modul aus der Datei \$BAU_PFAD/conf/bau.xml das Script bau_load_pg.sql erzeugt, das die Rohdaten unter Postgres lädt, oder die Datei bau_trans_ids.sql für das Script, das die Bau-Tabellen unter Informix transformiert

Syntaxmodule_scripts_create.x <<Modulename>> <<Modulpfad>> <<Daten-
banksystem(optional)>> <<Versionsnr.(optional)>>Beispielmodule_scripts_create.x BAU \$BAU_PFAD INFORMIX 1.0

Im Grunde handelt es sich um XML-Transformationen. Die Stylesheets für dieses Script befinden sich im Verzeichnis <code>\$SUPERX_DIR/db/conf</code>, und die XML-Datei für das Module in

\$SUPERX_DIR/db/module/<<Modulname>>/conf. Wenn die Datei nicht gefunden wird, bricht das Script ab.

Die folgende Abbildung zeigt die Arbeitsweise:

Das Script

module_scripts_create.x erzeugt eine Reihe von Scripten, die das Modul installieren / aktualisieren / deinstallieren. Außerdem werden htmlbzw. rtf-Dokumentationen erzeugt sowie Administrationsformulare für **dbforms**.



3.1.1.4.2 module_install.x

Installiert ein Modul <<Modulname>> in der Datenbank, wobei die Installationsdateien sich im <<Modulpfad>> befinden.

Syntaxmodule_install.x <<Modulename>> <<Modulpfad>>Beispielmodule_install.x BAU \$BAU_PFAD

3.1.1.4.3 module_drop.x

Löscht die Komponenten eines Moduls <<Modulname>> in der Datenbank, wobei die Installationsdateien sich im <<Modulpfad>> befinden.

Syntaxmodule_drop.x <<Modulename>> <<Modulpfad>>Beispielmodule_drop.x BAU \$BAU_PFAD

3.1.1.4.4 module_update.x

Installiert eine neue Version eines Moduls <<Modulname>> in der Datenbank, wobei die Installationsdateien sich im <<Modulpfad>> befinden.

Syntaxmodule_update.x <<Modulename>> <<Modulpfad>>Beispielmodule_update.x BAU \$BAU_PFAD

3.1.1.4.5 module_etl.x

Aktualisiert ein Modul <</Modulname>> in der Datenbank, wobei die Installationsdateien sich im <</Modulpfad>> befinden.

Syntaxmodule_etl.x <<Modulename>> <<Modulpfad>>Beispielmodule_etl.x bau \$BAU_PFAD

Die folgende Abbildung zeigt, wie die Komponenten zusammenhängen (klicken Sie auf die Grafik, um sie zu vergrößern):



Bei Fehlern im ETL-Prozeß wird die Sicherung wiederhergestellt, und eine Mail an den Administrator verschickt. Außerdem werden die übernommenen Daten überprüft; wenn z.B. Schlüssel fehlen oder Rohdaten falsch zu sein scheinen, wird dies als Attachment an die Log- oder Fehlermail angehängt.

In der Praxis wird dieses Script nicht direkt von Cronjobs ausgeführt, sondern von einem Shellscript, das vorher die Umgebung einrichtet. Das folgende Beispiel zeigt das Update-Script für Bau unter Informix:

```
bau_update.x #!/bin/sh
. /home/superx/db/bin/SQL_ENV
DBMONEY=,
export DBMONEY
ERRORMAIL="bau-admin@hochschule.de"
export ERRORMAIL
LOGMAIL=$ERRORMAIL
export LOGMAIL
module_etl.x bau $BAU_PFAD >$BAU_ERRORDAT 2>&1
```

Weil Buisy mit "," als Dezimaltrenner arbeitet, wird ausnahmsweise DBMONEY auf "," gesetzt. Außerdem ist es möglich, für jedes Modul unterschiedliche Mailadressen zuzuweisen. Die Mailadressen werden in der SQL_ENV (S. 94) eingetragen.

Im allgemeinen ETL-Prozeß wird standardmäßig auch die Tabelle protokoll in einem festzulegendem Rhythmus (Konstante Löschung Protokoll (Tage)) gelöscht. Beim Vorgabewert 90 werden bei jeder ETL-Routine Einträge, die älter als 90 Tage sind, gelöscht. Zusätzlich lassen sich im ETL-Prozeß hochschulspezifische Scripte ausführen (und überwachen). Dazu müssen Dateien mit einem gewissen Namensschema im Stammverzeichnis des Moduls vorhanden sein. Es gibt einen vereinfachten und einen erweiterten Modus für hochschulspezifische Transformationen.

Einfacher Modus	Wenn im Modulpfad die Datei "preparation.sql" existiert, wird sie nach dem LOAD-Schritt ausgeführt. Wenn im Modulpfad die Datei "finalize.sql" existiert, wird sie nach dem TRANS-Schritt ausgeführt.
Erweiterter Modus:	Wenn im Modulpfad Dateien nach dem Schema
Mandantenspezifische	< <modulname>>_<<etl-schritt>>_<<mandatennr.>>.sql</mandatennr.></etl-schritt></modulname>
Scripte	exisitieren,w erden diese jeweils nach dem "normalen" ETL-
~	Schritt ausgeführt.
	Wenn also z.B. die Datei
	cob_trans_70.sql
	existiert und in der SQL_ENV die Umgebungsvariable \$MANDANTID
	auf "70" steht, dann wird das Script nach dem normalen Trans-
	Schritt ausgeführt und nach L_cob_trans_mandant_70.log geloggt.

Der erweiterte Modus erlaubt die beliebige Anpassung eines Modus an eigene Bedürfnisse, z.B. Schlüsselumsetzung o.ä. Gleichzeitig erlaubt er einen echten mandentenfähigen Betrieb der ETL-Scripte.

3.1.1.5 Masken-Verwaltung

Die Masken-Verwaltung ist detailliert im **Entwicklerhandbuch SuperX** beschrieben. Hie rnur ein paar Hinweise zur Verwaltung der Masken. Zum Erzeugen und Verändern von Masken gibt es unter UNIX eine Kommandoschnittstelle, die auf dem Gebrauch folgender Skripte beruht. Die Skripte stehen unter dem Verzeichnis

\$SUPERX_DIR/db/masken

und erzeugen oder verwenden Dateien in dem gegenwärtigen Arbeitsverzeichnis. Nach dem Einspielen der Datenbank sollten Sie darauf achten, den Dateien Ausführungsberechtigung (chmod 750 sx_*) zu geben.

3.1.1.5.1 Eine Maske suchen

Wenn Sie eine Maske suchen, sollten die die Felder tid oder name in der Tabelle maskeninfo durchsuchen. Das folgende Script macht dies automatisch:

sx_search_mask

Aufruf:	sx_search_mask <string></string>
Aktion:	sx_search_mask sucht die Masken, deren Name <string> enthält</string>
Ausgabe: .	tid, name der gefundenen Masken

3.1.1.5.2 Eine Maske sichern und entladen

Um eine Maske zu sichern, müssen Sie die entsprechenden Einträge in den Tabellen

- 1. felderinfo,
- 2. masken_felder_bez,
- 3. maskeninfo,
- 4. sachgeb_maske_bez,
- 5.maske_system_bez

selektieren und sichern. Für dies gibt es das Script sx_select_mask.

|--|

Aufruf:	sx_select_mask <tid></tid>
Aktion:	sx_select_mask entlädt alle Metadaten aus den Tabellen maskeninfo, felderin-
	fo, masken_felder_bez, sachgeb_maske_bez, maske_system_bez zur Maske
	mit tid = $\langle TID \rangle$.
Ausgabe:	Fünf Dateien:
	1. <tid>_felderinfo.unl,</tid>
	2. <tid>_masken_felder_bez.unl,</tid>
	3. <tid>_maskeninfo.unl,</tid>
	4. <tid>_sachgeb_maske_bez.unl,</tid>
	5. <tid>_maske_system_bez.unl</tid>

3.1.1.5.3 Eine Maske neu einfügen

Um eine Maske neu einzufügen, müssen Sie die entsprechenden Einträge in den Tabellen

- 6. felderinfo,
- 7. masken_felder_bez,
- 8. maskeninfo,
- 9. sachgeb_maske_bez,
- 10.maske_system_bez

einfügen. Dafür gibt es das Script sx_insert_mask.

	SA_IIISCI t_IIIIISK
Aufruf:	<pre>sx_insert_mask <tid> [<neue tid="">] [j]</neue></tid></pre>
Aktion:	sx_insert_mask lädt den Inhalt der fünf Dateien
	1. <tid>_felderinfo.unl,</tid>
	2. <tid>_masken_felder_bez.unl,</tid>
	3. <tid>_maskeninfo.unl,</tid>
	4. <tid>_sachgeb_maske_bez.unl,</tid>
	5. <tid>_maske_system_bez.unl</tid>
	in die jeweiligen Tabellen der SuperX-Datenbank. Mit "j" wird die Sicher-

sx_insert_mask

heitsabfrage umgangen.	

Falls <neue TID> nicht angegeben wird, werden die Metadaten wieder mit der alten TID in die Datenbank eingespielt (=Update).

Falls <neue TID> angegeben wird, werden die Metadaten mit der neuen TID in die Datenbank eingespielt (=Insert). Dabei werden alle TIDs in den abhängigen Tabellen angepasst. So können Masken sehr einfach kopiert werden. Eine neue TID bekommt man durch die Wahl der nächsten Zehnerzahl, die größer als die größte vorkommende Nummer ist. Die größte vorkommende Nummer erhält man durch Ausführung des folgenden SQL-Ausdrucks mit Hilfe des Kommandos DBACCESS:

select max(tid) from maskeninfo;



Um den Austausch von Abfragen innerhalb der Hochschulen zu erleichtern ("Abfragen-Pooling" über die SuperX-Website), sollten die Masken immer im Nummernkreis xxxx0000 bis xxxx9990 liegen, wobei xxxx der von der HIS verwandten Hochschulnummer entspricht. Die Zehnerschritte ergeben sich daraus, dass die dazwischen liegenden Nummern für die Maskenfelder (Tabelle felderinfo) reserviert sind^{xx}.

Wie im Abschnitt Userverwaltung (S. 141) beschrieben, kann die neue Maske Benutzern oder Gruppen zugänglich gemacht werden.

3.1.1.5.4 Eine Maske löschen

Um eine Maske zu löschen, müssen Sie die Einträge in den oben genannten Tabellen entfernen. Dafür gibt es das Script sx_delete_mask

	sx_delete_mask
Aufruf:	sx_delete_mask <tid></tid>
Aktion:	sx_delete_mask löscht alle Metadaten aus den Tabellen maskeninfo, felder-
	info,masken_felder_bez,sachgeb_maske_bezundmaske_system_bez
	zur Maske mit tid = <tid>.</tid>

3.1.1.6 Änderungen an einer Maske vornehmen

- 1. Selektieren der Metadaten der betreffenden Maske: sx_select_mask <TID>
- 2. Editieren der fünf Metadaten-Dateien "<TID>_..."
- 3. Abspeichern der neuen Metadaten: sx_insert_mask <TID>

3.1.2 Administration mit Abfragen im XML-Frontend

Die Masken des XMI-Frontends erscheinen bei der Anmeldung von Benutzern, die Administratorrecht ehaben (z.B. voreingestellte User **superx** und **admin**).

aufruft. Nach der Anmeldung erscheint die folgender Themenbaum:

```
http://rechnername:8080/superx/xml/
```

im XML-Frontend	SuperX
	Administration
Benutzerverwaltung	Benutzer
	Gruppe suchen
	Institutionsrechte
	Nutzungsprotokolle (intern)
	User einrichten
	User löschen
	User suchen
Organigramm	Institutionen suchen
Masken und Felder	Masken
	Beschriftungen suchen
	Felder
	Feld erzeugen
	Feld kopieren
	Feld suchen
	Maske kopieren
	Maske löschen
	Maske suchen
Sichten	Sicht suchen
	Stylesheet suchen
Tahallan allaamain	Themenbaum-Eintrag suchen
i abenen angemenn	Tabelle suchen

Nach Anklicken eines Unterpunkts (wie Institution suchen) erscheint auf der rechten Seite ein Dialog zur Suche des jeweiligen Eintrags.

3.1.2.1 Das Organigramm bearbeiten

Meist wird da sOrganigramm aus anderen HIS-Systemen gefüllt, z.B. HISCOB. Wenn die Hochschule das Organigramm allerdigns selbst pflegt, gibt es die Möglichkeit, die Einsträge in einem einfachen Browser-Formular zu bearbeiten. Wenn man den Punkt **Institution suchen** anklickt und da sformular abschickt, erscheint z.B. folgendes Bild:

Administration des Kernmoduls: HowTo

Ohne Einschränkung werden alle Institutionen im Organigramm angezeigt. Mit dem rechten Button "Bearbeiten" gelangen Sie in eine Bearbeitungsmaske.

Export: Druckversion

Institution suchen

Stand: 01	.01.2003					
Nummer	Name (kurz)	Name	Übergeordnet	Lehre	Art	Bearbeiter
1510	Abteilung für Elektrotechnik u	Abteilung für Elektrotechnik u.Informationstechnik	1500	1		1
1530	Abteilung für Informatik, Info	Abteilung für Informatik, Informations-/Medientech	1500	1		1
1520	Abteilung für Maschinenbau	Abteilung für Maschinenbau	1500	1		1
1540	Abteilung für Materialtechnik	Abteilung für Materialtechnik	1500	1		1
2200	AkZent (Akad.Beratungszentrum)	AkZent (Akad.Beratungszentrum)	2	0		1
420	Audiovisuelles Medienzentrum	Audiovisuelles Medienzentrum	4	0		1
420	Audiovisuelles Medienzentrum	Audiovisuelles Medienzentrum	100000011	0		3
6440	Beauftragte und Personalvertre	Beauftragte und Personalvertretungen	6	0		1

Die Bearbeitungsmaske ermöglicht die Änderung der Bezeichnung (Drucktext wird normalerweise nicht angezeigt) und der übergeordneten Institution ("Parent") sowie der Gültigkeit.

ormular Organigramm	verwalten - Mozilla	-
		sort
		Sup
Organigramm	verwalten In diesem Formular können Sie Instit	utionen verwalte
Tid	1.420	
Key_apnr	1540	
Parent	Fakultät 5 – Ingenieurwissenschaften	
Drucktext	Abteilung für Materialtechnik	
Name	Abteilung für Materialtechnik	
Ebene	3	
Lehre	1	
Gueltig_seit	01.10.2001	
Gueltig_bis	31.12.2999	
Orgstruktur	[NULL]	
Sneichern << En	ster - Vorberiger Nächster > Letzter >> Löschen	Neu

Hier kann man nun den Namen, Drucktext, die Key-Apnr, Ebene, das Lehrekennzeichen, ggfs. Kennzeichen Orgstruktur und den Gültigkeitszeitraum bearbeiten.

Wenn man den Button Neu anklickt, erscheint der gleiche Dialog, bei dem man den Namen, key-apnr etc. der neuen Organisationseinheit eingeben kann.

Anklicken des löschen-Buttons entfernt eine Organisationseinheit aus dem Organigramm.

Wenn eine Organisationseinheit verschoben werden soll, z.B. Philosophie von Fachbereich 1 nach Fachbereich 6, geht dies über die Zuweisung des "Eltern"-Elements.

Wenn Sie alle Änderungen gemacht haben, können Sie diese durch Anklicken des speichern-Buttons in die Datenbank übernehmen.

3.1.2.2 Den Themenbaum bearbeiten

Wenn man den Punkt Themenbaum-Eintrag suchen anklickt und das Formular abschickt, erscheint z.B. folgendes Bild:

Es erscheint eine Liste mit Einträgen im Themenbaum. Sie können jeden Eintrag bearbeiten.

Einträge, die mit Masken verknüpft sind, können direkt zur Masken-Bearbeitung verlinken.

Export: Druckversion

Themenbaum-Eintrag suchen

Stand: 01.01.2003

otalia, 01/01/2000				
Name	Maske	Übergeordn. Knoten	Bearbeiten	Maske bearbeiten
Abfragen			3	
Administration		Abfragen	1	
Benutzer		Administration	3	
Benutzer im Detail	Benutzer im Detail	Benutzer	3	٦
Benutzer suchen	User suchen	Benutzer	1	1
Benutzer von SuperX	Benutzer von SuperX	Benutzer	3	٦
Benutzerdaten ändern	Benutzerdaten ändern	Benutzer	٦	1
Beschriftung suchen	Beschriftungen suchen	Masken	1	٦
Feld erzeugen	Feld erzeugen	Felder	٩.	1
Feld kopieren	Feld kopieren	Felder	3	3
Feld löschen	Feld löschen	Felder	4	٩

Das folgende Bild zeigt die Bearbeitungsmaske. Es können Bezeichnungstexte und übergeordnete Elemente geändert werden. Beachtne Sie, dass nach jeder Änderung in der jeweiligen Spalte rechts auf "Speichern" geklickt werden muss.

a romatai memenoaan -	nozmu					SuperX	
aum ^{In diesem Form}	ular können Sie den Themenba muss der "	aum t Then	pearbeiten. Bitte be nenbaum neu gelad	eachten en werd	Sie: bei Ände en (SuperXMa	rungen Inager)	
Name	Maskeninfo_id		Parent		Gueltig_seit	Gueltig_bis	
Benutzer im Detail	71040-Benutzer im Detail	~	Benutzer	~	01.01.190	01.01.300	Speichern
Benutzer suchen	71030-User suchen	~	Benutzer	~	01.01.190	01.01.300	Speichern
Benutzer von SuperX	750-Benutzer von SuperX	~	Benutzer	~	01.01.190	01.01.300	Speichern
Benutzerdaten ändern	71010-Benutzerdaten ändern	~	Benutzer	~	01.01.190	01.01.300	Speichern
Beschriftung suchen	70170-Beschriftungen suchen	~	Masken	~	01.01.190	01.01.300	Speichern
Feld erzeugen	70600-Feld erzeugen	~	Felder	~	01.01.190	01.01.300	Speichern
Feld kopieren	70630-Feld kopieren	~	Felder	~	01.01.190	01.01.300	Speichern
Feld löschen	70620-Feld löschen	~	Felder	~	01.01.190	01.01.300	Speichern
Feld suchen	70570-Feld suchen	~	Felder	~	01.01.190	01.01.300	Speichern
Felder		~	Masken	~	01.01.190	01.01.300	Speichern
Gruppe suchen	71050-Gruppe suchen	~	Benutzer	~	01.01.190	01.01.300	Speichern
Institution suchen	71080-Institution suchen	~	Administration	~	01.01.190	01.01.300	Speichern
Institutionen		۷	Administration	~	01.01.190	01.01.300	Speichern
Institutionsrechte	950-Institutionsrechte	~	Benutzer	~	01.01.190	01.01.300	Speichern
Maske kopieren	70350-Maske kopieren	۷	Masken	*	01.01.190	01.01.300	Speichern
Maske löschen	70370-Maske löschen	~	Masken	~	01.01.190	01.01.300	Speichern

Die Bezeichnungen von Maksen werden hier nicht vorgenommen, sondern nur in der Tabelle maskeninfo.

Ein Eintrag kann in der jeweiligen Zeile durch Anklicken von löschen entfernt werden.

Wenn Sie eine neue Kategorie wie Administration, Studierende oder Haushalt oder neue Masken einhängen wollen, wählen Sie unten *Neu*.

3.1.2.3 Userverwaltung

In SuperX lassen sich User- und Gruppenrechte komfortabel durch das XML-Frontend einrichten. Ausführliche Informationen zu den Tabellen und Relationen finden Sie im Kapitel Userverwaltung (S. 141).

3.1.2.3.1 Einzelne Benutzer löschen, neu anlegen und Stammdaten ändern

Wenn Sie im Bereich Administration den Bereich Benutzer wählen, sehen Sie folgende Oberfläche:

Themenbaum-Menü zur	Administration	
Userverwaltung	Benutzer	
	Benutzer im Detail	
	Gruppe einrichten	
	Gruppe löschen	
	Gruppe suchen	
	User einrichten	
	User löschen	
	User suchen	

3.1.2.3.1.1 Neuer Benutzer

Wenn Sie einen neuen Benutzer einrichten wollen, klicken Sie auf *User einrichten*. Anschließend werden Sie nach Angaben zur Kennung für den neuen Benutzer gefragt:

Die User-tid wird auto-	User einrichten	
matisch hochgezählt.	User ennichten	
Die Benutzerkennung	Bitte schränken Sie Ihre Auswahl ein:	
ist der Login-Text, und	User-tid 12	Benutzerkennung superx12
eine Gruppe kann aus-		
gewählt werden. Der	Email	
Name der Person muss	Max. Login-Versuche <mark>5</mark>	
angegeben werden.	Administrator Onein 💌	
Wenn Sie das Feld	Info	
"Inst-Rechte" leer las-	InstRechte	Stand: 20.06.2005 Aktualisieren
sen, hat der User Rechte	gültig von 20.06.2005	gültig bis 31.12.3000
auf alle Institutionen.	Nur Lehre O-alle 💌	

Die Gültigkeit kann ebenfalls eingeschränkt werden. Das Klappmenü "nur Lehre" wird bei den Orgranigramm-Rechten ausgewertet (obsolet mit 3.0).

Klicken Sie zum Abschluss auf "Abschicken". Der Benutzer wird dann mit dem Standardstartpasswort "anfang12" angelegt.

3.1.2.3.1.2 Benutzer löschen

Wenn Sie einen Benut- zer löschen wollen,	User löschen
wählen Sie im Themen-	Bitte schränken Sie Ihre Auswahl ein:
baum die Abfrage "User	
löschen" und dort die	zu loschende Kennung superx11 (wert)
Kennung in der Com-	Benutzerkennung (bestätigen) superx11
bobox.	

Bestätigen Sie Ihre Auswahl einmal, indem Sie die Kennung eintippen.

107

Beschreibung SuperX-Kern-Modul V3.0 © Projektgruppe SuperX

3.1.2.3.1.3 Benutzer bearbeiten

Im Formular **User suchen** erhalten Sie je nach Einschränkung eine Liste mit Benutzern.

Wir schränken z.B. ein auf die Gruppe Admi- nistratoren.	20.04.2005 hilfe] über
	User suchen
	Bitte schränken sie Ihre Auswahl ein:
	Kennung Gruppe Administratoren Name Institution Zurücksetzen
Es erscheinen zwei U- ser, die voreingestellten Administratoren. Sie können sich Details zur Person ansehen, oder die Person bearbeiten.	Export: Druckversion XMLTextRTF Export: Druckversion XMLTextRTF Duscharter Suppr: Administratoren ; Stand: 01.01.2003 Kennung Name Email Ansehen Bearbeiten admini Jane Doe Superx John Doe Export: Disgesamt 2 Sätzen.
In der Bearbeitungs- maske können Sie Ken- nung und Namen än- dern, sowie das Pass- wort auf "anfang12" zurücksetzen. Danach müssen Sie unten "Spei- chern" anklicken. Sie können die Grup- penzugehörigkeit zu- ordnen, und sie können dem User Rechte auf einzelne Sachgebiete, Masken und Institutio- nen geben.	User: John Doe Tid: 4 Kennung Superx Name John Doe E-mail [NULL] Administrator/in ? Pohlerhafte Login-Versuche O von max. 5 Passwort (verschlüsselt) 0533a66a3e9bea16131; Zurücksetzen auf anlang12' Curpe Sachgebietsrechte @ Administratoren @ Administration; Masken-Verwaltung; Userverwaltung
	Gruppen Markierte Gruppe speichern Markierte Gruppe löschen Neue Gruppe Dezernenten V Neue Gruppe einflügen Masken Neue Maske Veru Maske einflügen
	Institution Güitig von Güitig bis Clehreinheit sonstige Fächer Clehreinheit sonstige Fächer C

Bei den Gruppen, Masken und Institutionen handelt es sich um Unterformulare, d.h. Sie brauchen Änderungen hier nicht zu speichern.
3.1.2.3.2 Gruppen anlegen, löschen und Stammdaten verwaltung

Im Bereich *Gruppenverwaltung* erhält man die Möglichkeit, Gruppen zu löschen, neue Gruppen anzulegen und Stammdaten zu ändern.

3.1.2.3.2.1 Neue Gruppe anlegen

Wenn man eine neue Gruppe einrichten will, wählt man im Menü Administration->Benutzer->Gruppe einrichten.



Sie können auch direkt im Tabellenformular arbeiten, im Menü "Tabelle suchen"-> Tabelle groupinfo.

Hier können Sie Grup-	📕 Formular Gri	uppen verwalten - Mozilla				
penbezeichnungen än-						X
dern, Gruppen löschen						SuperX
(Vorsicht!) und neu an-	Grupp	en	In dieser	n Formular I	zönnen Sie I	Gruppen
legen.	verwa	lten	in alcoci	in onnaidh i	ve	rwalten.
Wenn Sie neue Gruppen	Tid	Name				
anlegen, müssen Sie die	2	Administratoren		Speichern	Löschen	
Nummern (tid) selbst	0	Dezernenten		Speichern	Löschen	
festlegen, die Daten-	1	Rektorat/Kanzler		Speichern	Löschen	
bank zählt nicht hoch.	Neu					
	<		1111			

3.1.2.3.2.2 Gruppe bearbeiten

Im Formular **Gruppe suchen** erhalten Sie je nach Einschränkung eine Liste mit Gruppen.

Wir schränken z.B. ein auf die Gruppe(n) des		SuperX				
Users superx.	20.04.2005	hilfe über				
	Gruppe suchen					
	Bitte schränken sie Ihre Auswahl ein:					
	Benutzer superx - John Doe 💌 Name					
	Abschicken	etzen				
Es erscheint ein Eintrag mit der Gruppe Admi-	Export: Druckversion XML Text RTF					
nistratoren, die zwei	Gruppe suchen					
User enthält. Sie können	Benutzer: superx - John Doe ; Stand: 01.01.2003					
die Gruppe bearbeiten.	Name Anzahl Benutzer Bearbeiten Administratoren 2 1					

In der Bearbeitungs- maske können Sie den	Gruppe	n bearbeiten	
Namen der Gruppe än-	Gruppe: Admir	stratoren	
dern. Danach müssen	Tid: 2 Name	Administratoren	
Sie unten "Speichern"			
anklicken.		Sachgebiet Benutzer von S löschen: Feld s	Masken-Rechte uperX; Feld ansehen; Feld bearbeiten; Feld erzeugen; Feld uchen: Institutionsrechte: Institution suchen: Maske ansehen:
Sie können die Grup-		 Administration Maske bearbeit Sicht suchen; Searbeiten (int) 	n; Maske löschen; Maske suchen; Nutzungsprotokolle (intern); itylesheet suchen; Tabellenfeld bearbeiten (char); Tabellenfeld ; Tabellenfeld bearbeiten (text); User-Institutionen
penzugehörigkeit zu-	Sachgebiete	O Masken-Verwaltung Verwaltung Reschriftungen suchen; Theme	suchen; Feld kopieren; Maske erzeugen; Maske kopieren; Tabel nbaum-Eintrag suchen
ordnen, und sie können		O Userverwaltung Y Gruppe suchen;	User einrichten; User löschen; User suchen
der Gruppe Rechte auf		Markiertes Sachgebiet speichern Markiertes S	uchgebiet löschen
einzelne Masken geben.			
Außerdme können Sie		• 💿 Nutzungsprotokolle (intern) 💌	
der Gruppe weitere User	Masken	Markierte Maske speichern Markierte Maske lo	schen
zufügen / enfernen.		Neue Maske	Neue Maske einfügen
	User	 admin - Jane Doe superx - John Doe Markierten User speichern Markierten User lös 	chen

Bei den Gruppen, Masken und Usern handelt es sich um Unterformulare, d.h. Sie brauchen Änderungen hier nicht zu speichern.

3.1.2.3.2.3 Gruppe löschen

Im XML-Frontend im Menü Administration -> Benutzer -> Gruppe löschen können Sie eine Gruppe löschen und die jeweiligen Rechte für Sachgebiete und Masken entfernen. Auch die Zuordnungen von Usern zur Gruppe (nicht aber die User selbst) werden gelöscht.

Sie müssen lediglich den Namen der Gruppe auswählen und einmal zur Sicherheit bestätigen, indem Sie den Namen eintippen. Mit Abschicken werden die Einträge entfernt.

Gruppe löschen

Bitte schränken Sie Ihre Auswahl ein:

zu löschende Gruppe	Testgruppe 🛛 👻	
Name (bestätigen)	Testgruppe	_

Aboobioka	- er er
AUSLIIILKE	311

Tabelle

3.1.3 Das Access-Frontend

Die Access-Datenbank enthält die Tabellen des Kernmoduls als Verknüpfungen und ermöglicht so ein leichtes Administrieren der Datenbank. Die Installation ist in der Installationsanleitung für ODBC-Quellen (S. 44) beschrieben. Die folgende Abbildung zeigt das Hauptmenü:





Probleme mit der Bedienung von Access gibt es immer dann, wenn Tabellen keine Primärschlüssel haben oder wenn die Felder mit den Primärschlüsseln nicht gefüllt sind. Mit der Version 2.1 erhalten alle Tabellen in SuperX (außer Datentabellen und Hilfstabellen, weil diese normal nicht manuell bearbeitet werden) Primärschlüssel. Wenn es dennoch Probleme gibt, empfehlen wir die Java-basierte SQLWorkbench (S. 112).

Das Access-Frontend ist insbesondere für die Änderung von Masken und Feldern (S. 121) gut geeignet.

3.1.4 Weitere Tools

Durch die odbc- und jdbc-Treiber können beliebige Datenbankfrontends eingesetzt werden. Gute Erfahrungen gerade mit Tabellen ohne Primärschlüssel haben wir mit der SQLWorkbench von Thomas Kellerer gemacht. Exemplarisch für andere jdbc-Clients haben wir dieses Programm näher beschrieben.

3.1.4.1 SQLWorkbench

Die SQLWorkbench von Thomas Kellerer arbeitet mit dem jdbc-Treiber jeweils von Postgres oder Informix. Beim ersten Aufruf der Workbench können Sie Profile für Treiber und Datenbanken eingeben. Musterprofile für viele gängige Datenbanksysteme liegen vor. Leider ist der Informix-Treiber nicht dabei, deshalb muss dieser "von Hand" registriert werden. Gehen Sie dazu über File->Connect in das Feld "Manage Drivers". Dort können Sie einen Namen vergeben und die jdbc-Parameter übertragen. Die folgende Abbildung zeigt ein Beispiel:



Im Dialog File -> Connect können Sie dann eine Datenquelle eintragen, und die Verbindungsparameter vervollständigen (Username, Passwort). Autocommit sollten Sie immer einschalten.

Die SQLWorkbench ist ein hervorragendes Administrations- und Entwicklungswerkzeug, daher haben wir die Version 94 in das SuperX-Clientpaket 30final integriert. Darin sind die Profile und Treibereinstellungen für Informix, Postgres und Joolap bereits voreingestellt, Sie müssen lediglich Servernamen und Ports ändern.

Interessant ist der Datenbank-Explorer (Tools -> Database Explorer), der es ermöglicht, die Datenbank nach Tabellen / Prozeduren etc. zu durchsuchen. Wenn eine Tabelle ausgewählt ist, kann sie auch über die Registerkarte "Data" editiert werden. Achten Sie darauf, dass Sie das Feld **Max. Rows** auf einen sinnvollen Wert setzen, z.B. 2000. Die SQLWorkbench ist gerade für die Arbeit mit Tabellen ohne Primärschlüssel geeignet, weil jede Äderung intern als Update formuliert wird. Der Nachteil ist, dass nicht mehrere Zellen über Zwischenablage geändert / eingefügt werden können.

Sehr praktisch für die Entwicklung von SQL-Abfragen ist die Möglichkeit, zu jeder Tabelle einen select-String zu formulieren. Markieren Sie die Tabelle im Database Explorer, und gehen Sie über das Kontextmenü auf Show table data, und wählen Sie ein Editorfenster aus. Der Select-String wird dann angezeigt.

🗑 Database Explorer - [superxBN_miles]											
User superx 💌											
Tablaa)roooduroo		arah tahlaa								
	Toteuules		sarun tables								
TABLE 🔻	- 🏀 🏀 🤅	\$	COLUMN_N	JAME	DATA_TYPE		ΡK	NULLA	DEFAULT	REMARKS	
NAME		_	lehr		char(10)		NO	YES			
kan lehrangeh	ot		stg		char(3)		NO	YES			
kan studienerf	ola	-	vertfg		char(3)		NO	YES			
kasse ha	oig		schwerpunkt		char(2)		NO	YES			
kasse_bu			pversion		smallint		NO	YES			
kasse_bus			kz_fach		char(1)		NO	YES			
kasse_bas_pro	л.		abschluss		char(2)		NO	YES			
kasse_zps			text		char(200)		NO	YES			
kasse_zwa	0.010		regel		smallint		NO	YES			
koepie_oder_i	aeiiz		fach_zaehler		smallint		NO	YES			
koepte_oder_taelle tid			int		NO	YES					
konstanten semester_von		1	int		NO	NO					
konto_summe			semester_bis		int		NO	NO			
lenr_stg_ap	B Save d	ata a	e								
ienrein_aimoa	4 <u>0</u> 04 <u>v</u> c 4										
ienrein_tach_i	<u>С</u> ору D	ata I	o Clipboard	Ctrl-Y							
lehrein_zahl_k	A Eind			Ctrl-F							
lehreinh_umst	00 <u>-</u>										
lehreinheit_fb	🖷 Find <u>N</u> e	EXT		Utrl-G							
lehreinheiten	⊲2 Sp <u>o</u> ol o	lata									
lehreinheiten	🛱 Print										
macro maske	Contraction of the second seco										
maske syster	Print pr	eviež	<u>v</u>								
masken_felde	Drop										
maskeninfo	Observatedate				Otatawanta						
mbs_geldgebe	Show table	data			Statement 1						
mbs_konto		1			Statement 2						
mbstellen		Statement 3									
mittelverteilung			Ctotomont 4								
nachricht			B3333333333333		otatement 4					8	•
notebn	Þ	•	Columns	Sou.	Statement 5	dexes	Refe	erences	Reference	d by Trigger	s

Seit den Versionen 93 lassen sich bei Informix auch Felder vom Typ text anzeigen und editieren. Das Tool bietet außerdem eine Makrofunktion, und in neueren Versionen auch ETL-Funktionen über einen "Data Pumper", was es natürlich für SuperX besonders interessant macht. Weitere Tipps und Hilfen erhalten Sie im (gelungenen, aber englischen) Benutzerhandbuch der Workbench.

3.2 Einen User betreuen

Jeder Benutzer von SuperX sollte ein geheimes Paßwort benutzen, welches nicht einfach erraten werden kann. Paßwörter wie Vornamen, Stellung im Beruf o.ä. dürfen unter keinen Umständen verwendet werden. Zum Ändern des Paßworts kann im Applet und im XML-Frontend ein Paßwortänderungsdialog aufgerufen werden.

3.2.1 Neuen User einrichten

Im Kernmodul befindet sich eine Abfrage "User einrichten", mit der Sie einen User einrichten und ggf. auf bestimmte Institutionsrechte oder Gültigkeitszeiträume einschränken können. Außerdem können Sie den User einem Sachgebiet oder einer Gruppe zuordnen.

Bearbeitungsformulare zur Benutzerverwaltung befinden sich im XML-Frontend (S. 104). Ausführliche Informationen finden Sie im Kapitel UserverwaltungTabellen (S. 141). Hier eine Anleitung für die direkte Änderung in der Datenbank:

- 1. Erstellen eines Eintrags in der Tabelle userinfo. Neue Tid merken.
- 2. Setzen des Startpassworts "anfang12". Mit dem Befehl update userinfo set passwd_sha="0533a66a3e9bea16f3139bfe4f6ce50ced591dea" where tid=<Neue Tid> Der User muss aufgefordert werden, sein Passwort beim ersten Start zu ändern.
- 3. Dem User Rechte für Institutionen geben durch Einträge in die Tabelle user_institution.

Ggfs. Gruppenzugehörigkeit eines Benutzers festlegen.

- 4. (Eintrag in Tabelle user_group_bez siehe Abschnitt Userverwaltung)
- 5. Falls durch die Gruppenrechte noch nicht abgedeckt: Dem User Rechte für ganze Sachgebiete und/oder einzelne Abfragen geben (Einträge in die Tabellen user_sachgeb_bez bzw. u-ser_masken_bez)

Sie können einen User zwingen, sich nach der Anmeldung ein neues Passwort zu geben. Dazu wird dem Feld kennwort in der Tabelle userinfo der Wert "ändern" gegeben. Es erscheint dann nach der ersten Anmeldung eine Aufforderung, das Passwort zu ändern.

3.2.2 Passwort vergessen

Den Befehl

update userinfo set passwd_sha="0533a66a3e9bea16f3139bfe4f6ce50ced591dea" where benutzer="<Kennung des Benutzers">

ausführen. Dadurch erhält der Benutzer wieder das Startpasswort "anfang12", was er nach erfolgreichem Anmelden wieder sofort ändern sollte. Des weiteren kann der SuperX-Adminstrator im XML-Frontend ein beliebiges Passwort für den Benutzer vergeben und die Checkbox "User muss Passwort ändern" aktivieren, damit der er bei der nächsten Anmeldung ein neue Passwort vergeben muss.

3.2.3 User-Rechte ändern

- Rechte für Institutionen gibt oder entfernt man durch Hinzufügen/Löschen von Einträgen in der Tabelle user_institution
- Gruppenzugehörigkeit wird über die Tabelle user_group_bez definiert, ggfs. dort Einträge hinzufügen oder löschen
- individuelle Rechte für Sachgebiete und/oder Masken über die Tabellen user_sachgeb_bez bzw. user_masken_bez anpassen

Die Stammdaten (Name, email, etc) befinden sich in der Tabelle userinfo.

3.2.4 User löschen

Es gibt im Kernmodul eine Abfrage "User löschen" (im Themenbaum unter Administration -> Benutzerverwaltung). Wenn Sie den User "von Hand" löschen wollen:

Die tid des Benutzers aus der Tabelle userinfo heraussuchen.

Folgende Befehle ausführen.

delete from user_masken_bez where userinfo_id=<tid des Users>; delete from user_sachgeb_bez where userinfo_id=<tid des Users>; delete from user_institution_bez where userinfo=<tid des Users>; delete from userinfo where tid=<tid des Users>;

3.3 Einstellungen zur Passwortsicherheit

Bei der Installation des SuperX-Kernmoduls werden in die Tabelle konstanten vier Einträge zur Einstellung der Passwortsicherheit gemacht.

Name der Konstante	Kommentar	default-Wert
Passwortgültigkeit (Tage)	Gibt an, für wie viele Tage	180
	das Passwort gültig sein soll.	
	Sobald die Gültigkeit abgelau-	
	fen ist, muss der Anwender	
	bei der nächsten Anmeldung	
	ein neues Passwort vergeben.	
Passwort Groß- u. Kleinb.	Müssen Groß- und Klein-	1
	buchstaben im Passwort vor-	
	kommen (0=nein, 1=ja)	
Passwort erfordert Ziffer	Müssen Ziffern im Passwort	1
	vorkommen (0=nein, 1=ja)	
Passwortlänge (Minimum)	Geben Sie hier die minimale	8
	Passwortlänge an	

Beim Upgrade einer älteren Kernmodul-Installation (vor Kernmodul 3.0rc7) sind folgende Vorbelegungen aktiv:

Name der Konstante	default-Wert
Passwortgültigkeit (Tage)	360
Passwort Groß- u. Kleinb.	0
Passwort erfordert Ziffer	0
Passwortlänge (Minimum)	6

Die Zentrale Datenschutzstelle der baden-württembergischen Universitäten (Zendas) macht folgende Empfehlung:

Passwortgültigkeit (Tage)	90-180
Passwort Groß- u. Kleinb.	1
Passwort erfordert Ziffer	1
Passwortlänge (Minimum)	8

Der SuperX-Administrator kann erzwingen, dass der Benutzer sein Passwort ändern muss, indem er im XML-Frontend den entsprechenden User bearbeitet und bei "User muss Passwort ändern" ein Häkchen setzt. Neue User werden in der Maske "User einrichten" defaultmäßig so eingestellt, dass sie ihr Passwort nach der ersten Anmeldung ändern müssen.

3.4 Eine Gruppe betreuen

Sie können in SuperX durch Einträge in den Tabellen zur Gruppenverwaltung einzelne Gruppen anlegen, mit Leserechten für Abfragen und Institutionen versehen. Die zugehörigen Tabellen werden in der Gruppenverwaltung (S. 145) erläutert.

3.4.1 Neue Gruppe einrichten

Ausführliche Informationen zur Gruppenverwaltung finden Sie im Kapitel Userverwaltung (S. 141).

1. Erstellen eines Eintrags in der Tabelle groupinfo. Neue Tid merken.

2. Der Gruppe Rechte für Sachgebiete und/oder einzelne Masken geben (Einträge in die Tabellen group_sachgeb_bez bzw. group_masken_bez)

3.4.2 Gruppen-Rechte ändern

Gruppenrechte für Sachgebiete und einzelne Masken werden in den Tabellen group_sachgeb_bez bzw. group_masken_bez festlegt. Dort ggfs. Einträge machen oder löschen.

Der Gruppenname kann in der Tabelle groupinfo geändert werden.

3.4.3 Eine Gruppe löschen

Die tid der Gruppe aus der Tabelle groupinfo heraussuchen.

Folgende Befehle ausführen.

delete from group_masken_bez where groupinfo_id=<tid der Gruppe>; delete from group_sachgeb_bez where groupinfo_id=<tid der Gruppe>; delete from groupinfo where tid=< tid der Gruppe>;

3.5 Verwaltung und Rechtevergabe von Sichten

SuperX-Sichten sind hierarchische Zusammenstellungen von Dimensionen, z.B. von alternativen Kostenstellenhierarchien. Die Sichten können in einem eigenen Formular verwaltet werden, außerdem können die Berechtigungen für Sichten eingeschränkt werden.

Nach der Anmeldung als Administrator im XML-Frontend können Sie im Themenbaum die Abfrage **Administration** -> **Sicht suchen** wählen, und abschicken.

Im Kernmodul ist nur	Sicht su	chen				
eine Sicht enthalten,	orone ou	onon				
eine alternative Hierar-	Stand: 01.01.200	3				
chie, die den Themen-	Name	Beschreibung	Art	Bearbeiten	User- und Gruppenrechte	
enie, die den Themen-	Themenbaum		Themenbaum-Sicht	1	3	
baum aufbaut.						
	Datensatz 1 - 1	von insgesamt 1	Satz			

3.5.1 Bearbeitung von Sichten

Sie können mit Klick auf den "Bearbeiten"-Button die Sicht in einem Datenbank-Formular bearbeiten. Die folgende Abbildung zeigt das Formular.

	Formular Sichte	n - Mozilla <2>
Sichten		In diesem Formular können Sie Sichten bearbeite
tid	11	
Systeminfo	Administration 💌	Querverweis zur Systeminfo
Art	Themenbaum-Sicht	
Туре	10	10 ist standardsicht, 20 ist alt.Hierarchie
Interner Name	memtext_themenbaum	Eindeutiger Datensatzbezeichner
Name	Themenbaum	Bezeichnung der Sicht, die für User angezeigt wird
Beschreibung		
Sortiernummer	0	kann für Sortierungen benutzt werden
Quelle	<sql>> select name,tid,parent from themer</sql>	Angabe einer Tabelle mit key,parent,name,gueltig_seit,gueltig_bis od Prozedur, die mit sp_ anfängt
Alt_hier_id		id der alt.Hier in angegebenen Quelltabelle, null bei regulärer Hierard
Treecfgtable		Tabelle mit Infos zu TreeView aus Cob
Treecfgid		id des benutzen trees aus Cob-Tabelle trees null bei regulärer Hierac
User_rechte	1	sollen User Rechte berücksichtigt werden, derzeit von Org.Sichten ausgewertet
Standbutton	0	soll der Stand geändert werden können
Attribut1		bei Bedarf noch Attribute der Sicht hinterlegt werden, auf die man bei Bedarf Einschränkungen fahren kann
Attribut2		bei Bedarf noch Attribute der Sicht hinterlegt werden, auf die man bei Bedarf Einschränkungen fahren kann
Attribut3	No Dat:	bei Bedarf noch Attribute der Sicht hinterlegt werden, auf die man bei Bedarf Einschränkungen fahren kann
Attribut4	[No Dat:	bei Bedarf noch Attribute der Sicht hinterlegt werden, auf die man bei Bedarf Einschränkungen fahren kann
Max. Zeilen XML	[No Dat:	Max. Zeilen im Klappmenü
Gueltig_seit	Jan 1, 1900	
Gueltig_bis	Jan 1, 3000	
Aktiv	1	
Speichern	< Erster < Vorheriger Nächster >	Letzter >> Löschen Neu

Wenn Sie Sichten für die Anwender ausblenden wollen, ist es nicht ausreichend, die Datensätze zu löschen - sie würden beim nächsten Update aus dem Quellsystem wieder eingespielt. Stattdessen sollten Sie die Sichten in der Konstante "Aktiv" ganz unten im Formular auf "0" setzen und dann im SuperXManager den Cache aktualisieren.

Weitere Details zur Anpassung von Sichten finden Sie im SuperX-Entwicklerhandbuch.

3.5.2 Berechtigung für Sichten

Um die Berechtigung von Sichten zu vereinfachen, werden mehrere Sichten in SuperX zu "Sichtarten" zusammengefasst. Sie können Berehcitgungen auf beiden Ebenen vergeben.

Mit Klick auf den Button "User- und Gruppenrechte" können Sie die Rechte für die Sicht / Sichtart vergeben. Wie beid er Userverwaltung finden Sie hier einige Unterformulare: User bzw. Gruppen, die die Sicht sehen dürfen. Sie können User bzw. Gruppen hinzufügen oder entfernen.

Im unteren Teil des

Formulars können Sie

te für ganze Sichtarten

festlegen.

User- und Gruppenrech-



Wie bei der Userverwaltung handelt es sich um Unterformulare, d.h. Sie brauchen die Änderungen jeweils nicht zu manuell zu speichern.

3.6 (Abfrage-)Masken entwickeln

Die Abfragemasken liefern die Daten aus den Basissystemen an das SuperX-Frontend aus. Einige Abfragen zur Administration sind im Kernmodul enthalten, die Abfragen zu den Basissystemen sind in den jeweiligen Modulen enthalten. Die Abfragen in der Administration erlauben es, neue Masken anzulegen, zu kopieren und zu löschen. Im Folgenden finden Sie allgemeine Hinweise für die Verwaltung der Masken.

Die Masken lassen sich über UNIX, über Access und in Zukunft über ein Java-Frontend administrieren. Unter UNIX geschieht dies über Scripte. Für Windows-Nutzer gibt es ein Access-Frontend, das sich derzeit im Betatest befindet.

3.6.1 Maskenverwaltung im SuperX-Applet oder XML-Frontend

Die Masken lassen sich im SuperX-Applet verwalten, weitergehende Möglichkeiten bietet aber das XML-Frontend (Möglichkeit der Editierung von großen text-Feldern bei Postgres als Datenbanksystem). Nach der Anmeldung haben Administratoren das Recht, Masken zu löschen, zu kopieren und erzeugen. Die einzelnen Felder der Masken lassen sich direkt in der Datenbank oder z.B. mit MS Access (S. 121) verändern. Im Applet sind nur grundlegende Verwaltungsoperationen möglich. Sie sind als Ersatz für die UNIX-Scripte (S. 93) gedacht.

Folgende "Abfragen" zur Maskenverwaltung gibt es im Sachgebiet Administration:

Darunter im Ast "Felder" gibt es noch folgende Abfragen: Darüberhninaus gibt es (nur unter

Postgres) die Masken zur Pflege von Masken bzw. Feldern

- Maske erzeugen
- Maske kopieren
- Maske löschen
- Feld kopieren
- Feld löschen
- Maske suchen
- Feld suchen
- Maske erzeugen. Hier kann eine neue Maske erzeugt werden, und die wichtigsten Zuordnungen der Maske werden angegeben, z.B. Sachgebiet, Themenbaum-Parent etc. Die Felder der Maske selbst in den dazugehörigen Tabellen (z.B. maskeninfo) werden nicht gefüllt oder im Applet administriert. Dazu dienen die Datenbank-Frontends selbst. (s.u.).
 Bei der Nummer der Maske (tid) sollten Sie das Nummernschema von SuperX einhalten, um in Zukunft Abfragen-Pooling (S. 103) zu ermöglichen.
- Maske kopieren. Wie im UNIX Script (S. 102) wir eine Maske in eine neue Maske kopiert, und alle zugehörigen Tabellen werden aktualisiert. Zusätzlich wird auch der Eintrag im Themenbaum gemacht.
- Maske löschen. Wie im UNIX-Script (S. 103) werden Masken aus allen dazugehörigen Tabellen entfernt. Zusätzlich wird auch der Eintrag im Themenbaum gelöscht. Zur Sicherheit muss die Nummer der Maske manuell eingegeben werden.
- Maske suchen. Sie können Masken suchen und im XML-Frontend komfortabel editieren. Schränken Sie Ihre Auswahl auf ein Sachgebiet ein, und drücken Sie "Abschicken". Sie erhalten eine Liste mit "Treffern", und rechts befinden sich jeweils Buttons zum ansehen bzw. editieren einer Maske. Die Maske läuft nur unter Postgres, weil Informix kein direktes Bearbeiten von Blob-Feldern mit sql unterstützt.

Feld suchen. Sie können analog zu "Maske suchen" auch Felder suchen und bearbeiten.

Die Abfragen sind selbsterklärend; das Erzeugen neuer Masken, Löschen vorhandener Masken und Kopieren vorhandener Masken ist nur für Userkennungen möglich, die in der Tabelle userinfo im Feld administration den Wert 1 haben. Natürlich sollten die Abfragen sehr vorsichtigt benutzt werden, sie sind die einzigen Abfragen in SuperX, die tatsächlich Änderungen an der Datenbank vornehmen können.

3.6.2 Maskenverwaltung mit MS Access

Das Access-Frontend (S. 44) ermöglicht die bequeme Änderung von Abfragen (für die Eingabe <u>neuer</u> Masken und Felder empfehlen wir eher die Abfragen im normalen Themenbaum). Es befindet sich im SuperX-Clientpaket (S. 90) in tools\access\superx_frontend_sam.mdb. Benennen Sie die Datei um nach superx_frontend.mdb. Danach können Sie unter *Masken* die einzelnen Masken von SuperX anwählen und öffnen. Sie erhalten im Formular maskeninfo ein Formular, das Eingaben oder Änderungen in der Tabelle maskeninfo ermöglicht.



Das select_stmt ist ein großes Textfeld und läßt sich besser durch Drücken der _____-Taste in einem separaten Fenster bearbeiten. Leider werden Tabulatoren im normalen Windows-Editor nicht korrekt dargestellt, deshalb befinden sich rechts noch zwei Buttons, mit denen Sie Masken in Word^{xxi} editieren können.

Mit dem Button is öffnen Sie das select_stmt in Word, und können dort Änderungen vornehmen. Mit dem Button is speichern Sie die Änderungen in der Datenbank, und Word wird geschlossen. Bitte beachten Sie, dass Sie die Dateien in Word nicht speichern müssen. Analog können Sie verfahren, wenn Sie das Feld xil_proplist bearbeiten. Um in Access sicherzustellen, dass Feldänderungen wirklich in der Datenbank gespeichert werden, sollten Sie sich einen Button zum Speichern von Datensätzen in die Access-Symbolleiste setzen (Extras -> Anpassen -> Befehle -> Datensatz speichern in eine häufig benutzte Symbolleiste ziehen).

Mit dem Button Felderinfo gelangen Sie zu den Feldern dieser Maske. Sie können die Felder dort bearbeiten. Bein Hinzufügen neuer Felder müssen Sie allerdings die jeweiligen tids manuell in die Tabelle masken_felder_bez eintragen.

Analog funktioniert die Bearbeitung der individuellen Stylesheets für eine Maske.

3.6.3 Effizientes Debugging

Ein großer Nachteil des 'alten' SuperX war die unvollständige Übermittlung von Fehlermeldungunen bei der Ausführung von Abfragen. Mit dem neuen SuperX gestaltet sich die Entwicklung von Abfragen wesentlich einfacher. Es gibt mehrere Wege, Abfragen zu enwickeln und zu debuggen.

Als Entwickler sollten Sie sich eine einene Servlet-Engine mit Tomcat lokal installieren (S. 50) und das Logging in der Datei

\$SUPERX_DIR/webserver/tomcat/webapps/superx/web-inf/web.xml

aktivieren (S. 53) (Achtung: dies sollten Sie nur bei lokalen Tomcats tun, nicht im Echtbetrieb, da sonst die Performance leidet). Durch das Logging können Sie genau sehen, an welcher Stelle eine Abfrage abbricht, u.U. übermittelt der JDBC-Treiber auch die Fehlermeldung. Sie können die Protokollierung auch in eine Datei umleiten. So können Sie auch einzelne SQL-Statements aus der LOG-Datei kopieren und in dbaccess von Hand ablaufen lassen.

Viele Abfragen in SuperX arbeiten mit temporären Tabellen. Diese sind zwar unter Informix kennungsund sitzungsabhängig ist, aber es kann beim Abbruch einer Abfrage passieren, daß temporäre Tabellen erhalten bleiben. Beim nächsten Start der Abfrage führt dies also zu einer Fehlermeldung, wenn die temporäre Tabelle neu erzeugt werden soll. So kann es also passieren, dass kein User eine Abfrage mehr ausführen kann, oder gar dass Tomcat abstürzt. In diesem Falle muss man Tomcat einmal beenden und wieder neu starten. Auch deshalb ist es im Echtbetrieb also ratsam, für die Entwicklung von Abfragen einen "eigenen" Tomcat lokal zu installieren.

Ein weiterer einfacherer Weg des SQL-Debugging besteht darin, sich die Fehlermeldungen im Applet anzeigen zu lassen; dazu muss das Logging in der Datei superx.properties eingeschaltet werden. Danach können Sie sich die SQL-Statements in der Java-Konsole anschauen. Diese können Sie (unter Windows) in der Systemsteuerung -> Java Plugin x.x aktivieren. Unter Netscape 6.x mit Linux erreichen Sie die Java-Konsole über das Menü Tools -> Java Console. Sie können auch das SuperX-Applet auspacken (jar -xvf superx.jar) und dann SuperX als Anwendung starten mit java superx >logdatei.txt. In diesem Falle sehen Sie die Fehlermeldungen direkt auf der Konsole oder in der DOS-Box bzw. in der angegebenen Logdatei.

3.6.4 Dokumentation von Abfragen: Glossare

Die Statistiken in SuperX ist nicht immer für Außenstehende "selbsterklärend", und insbesondere bei Kennzahlen und kondensierten Werten sollten die Konzepte mit einem Glossar versehen sein.

Die Frontends von SuperX bieten drei Möglichkeiten der Dokumentation:

- Dialogelemente auf den Masken können mit einem "Tool-Tip" versehen werden, d.h. bei Mausbewegung über den Button wird eine Erläuterung angezeigt.
- Ergebnistabellen können mit einem Glossar versehen werden, das die in der Tabelle benutzten Begriffe auf einer zweiten Seite erläutert.
- Umfangreichere Hilfetexte sind über die kontextabhängigen Hilfetexte zu einer Maske und Ergebnistabelle verlinkt. Dies ist in der Entwickleranleitung (S. 205) dokumentiert.

Für die ersten beiden Dokumentationsarten wird in SuperX die Tabelle sx_captions gepflegt, die Felderläuterungen und allgemeine Schlüsselörter dokumentieren. Die Dokumentation ist sogar mehrsprachig möglich.

3.6.4.1 Allgemeine Schlüsselwörter

Allgemeine Schlüsselwörter sind der Tabelle sx_captions definiert, man erkennt sie, daran dass die Spalte id gefüllt ist (table_name, field_name und record_no hingegen leer)

tid	id	table_name	field_name	record_no	locale	contents_short	contents_long	sachgebiete_id
1	studiengang				de	Studiengang	Studiengänge	16
							definieren	
							sich durch	
							das Fach, die	
							Vertiefungs-	
							richtung,	
							durch	
							Haupt- oder	
							Nebenfach	
							sowie den	
							Abschluss.	
2	studiengang				en	Subject /	A combina-	16
						Degree	tion of sub-	
							ject and de-	
							gree as well	
							as the major-	
							minor dis-	
							tinction	
3	stud_general				de	Studierende		
						allgemein		
4	stud_general				en	students (ge-		
						neral)		

Im Beispiel wird der Tag studiengang definiert.

Dieser Tag wird an beliebiger Stelle (Maskennamen, Überschriften, select_stmt, XIL-Proplist, XSL-Dateien, etc) durch den Eintrag contents_short der aktuellen Locale ersetzt.

3.6.4.2 Der Spezialfall Maskenfelder

Für die Erläuterung von Maskenfeldern können kurze und längere Hilfetexte hinterlegt werden. Die kurzen Texte dienen als Beschriftung des Feldes (überscheiben als den "Feldnamen"), und die langen Texte erscheinen als Tool-Tip bei Mausbewegung auf den Button. Im Ausdruck werden die Maskenfelder wahlweise auf einer separaten Seite dokumentiert. Damit nicht für jedes einzelne Maskenfeld ein Eintrag gemacht werden muss, kann ein Hilfetext über seinen Namen auch mehreren Maskenfeldern zugeordnet werden; in diesem Fall ist die Spalte record_no leer.

Für Felder aus der Tabelle felderinfo schaut SuperX nach, ob in der Tabelle sx_captions ein Eintrag für die Tabelle felderinfo, field_name studiengang und record_no = 10050 oder null vorhanden ist

Im folgenden Beispiel ist ein Maskenbutton "Studiengang" erläutert, der in dieser Weise und bei dem Feld Nummer 10050 dokumentiert sein soll.

tid	id	table_name	field_name	record_no	locale	contents_short	contents_long	sachgebiete_id
9		felderinfo	studiengang	10050	de	Grundständiger	Ein Stu-	16
						Studiengang	diengang im	
							grundstän-	
							digen Stu-	
							dium	
10		felderinfo	studiengang	10050	en	Degree program		16

Wenn Sie den Erläuterungstext bei allen Feldern mit dem Namen "studiengang" erscheinen lassen wollen, dann müssen Sie das Feld record_no leer lassen.

3.6.5 Masken für das XML-Frontend vorbereiten

Das XML-Frontend arbeitet mit den vorhandenen Masken und stellt dort grundlegende Funktionen zur Verfügung. Darüber hinaus bietet das Frontend die Möglichkeit, einzelne Abfragen individuell zu gestalten. Hierzu sind allerdings grundlegende XML-Kenntnisse erforderlich. Außerdem gibt es für den Betrieb gewisse Einschränkungen.

Ein großer Vorteil des XML-Frontends ist, dass Anwender sich ihre Bericht im XML-Format herunterladen können und ohne Datenbankkenntnisse ihre Berichte "maßschneidern" können.

Außerdem arbeitet das XML-Frontend asynchron, d.h. die neuen Servlets können (bislang über die URL) von beliebigen Stellen aus aufgerufen werden. Es ist z.B. damit möglich, auf beliebte Bericht mit gesetzten Parametern einen Bookmark zu legen.

3.6.5.1 Erzeugen eines Stylesheets

Zunächst muss für das Ergebnis ein neues Stylesheet erzeugt werden. Als Vorlage für Masken können Sie das Muster-Stylesheet

\$SUPERX_DIR/webserver/tomcat/webapps/superx/xml/maske_html_ns.xsl

(bzw. maske_html_ie.xsl mit speziellen Tags für den Internet Explorer von Microsoft) verwenden, für Ergebnistabellen können Sie das Muster-Stylesheet

\$SUPERX_DIR/webserver/tomcat/webapps/superx/xml/tabelle_html.xsl

verwenden. Speichern Sie das Stylesheet unter einem anderen Namen im gleichen Verzeichnis ab, und ändern Sie das Stylesheet. Dann fügen Sie das Stylesheet in die Tabelle sx_stylesheets ein.

▦	III sx_stylesheets : Tabelle						
	tid	filename	caption	description	relation	userag	contenttype
▶	1	tabelle_html.xsl	Generisches St	Generisch	table		text/html; charset=ISO-8859-1
	2	tabellenfeld_bearbeiten.xsl	Generisches St	Generisch	table		text/html; charset=ISO-8859-1
	3	maske_html.xsl	Generisches St	Generisch	mask		text/html; charset=ISO-8859-1
	4	maske_html_tabfeld.xsl	Generisches St	Generisch	mask		text/html; charset=ISO-8859-1
	5	tabelle_html_11570.xsl	Berichtsblatt Ko	Kurze Zusamm	table		text/html; charset=ISO-8859-1
	7	tabelle_fo_rtf.xsl	RTF	Export in Textve	table		application/msword
	6	tabelle_fo_pdf.xsl	PDF	Export in PDF (table		application/pdf
*							
Da	Datensatz: II I I III V VON 7						

Das Beispiel zeigt einige Styleheets, das erste ist bereits Teil des Kernmoduls, das fünfte befindet sich im COB-Modul. Zu den Feldern:

- **filename** kennzeichnet den Dateinamen relativ zum Verzeichnis \$SUPERX_DIR/webserver/tomcat/webapps/superx/xml.
- caption dient als Kurzüberschrift, die im Ergebnisblatt als Button angezeigt wird.
- description stellt einen Erläuterungstext für den Button dar.
- **relation** bezieht sich auf die Beziehung des Stylesheets; mögliche Werte sind "mask" für eine Maske und "table" für Tabelle.
- **useragent** beitet die Möglichkeit, ein Stylesheet für spezielle Lesegeräte anzubieten, z.B. WAP-Handys oder Braille-Zeilen.
- **contenttype** entspricht dem useragent und kennzeichnet den content-type, der dem Lesegerät im http-header übermittelt werden soll. Möglich sind derzeit die obigen Varianten (svg oder excel sind in Vorbereitung).

3.6.5.2 Zuordnung einer Maske zu einem Stylesheet

Konkret arbeitet SuperX so: Wenn einer Abfrage ein oder mehrere Stylesheets zugeordnet sind, dann werden die in der Reihenfolge angezeigt, in der sie definiert sind. Wenn kein Stylesheet definiert ist, dann wird das Standard-Stylesheet von SuperX benutzt: maske_html_ie.xsl bzw maske_html_ns.xsl für Masken sowie tabelle_html.xsl für Tabellen.

Die Zuordnung eines Stylesheets geschieht in der Tabelle sx_mask_style. Der Tupelidentifier des Stylesheets wird in der Tabelle sx_mask_style im Feld stylesheet_id eingetragen.

Das Beispiel zeigt, daß	▦	⊞ sx_mask_style : Tabelle				
die beiden oben be-		tid	maskeninfo_id	stylesheet_id	ord	
schriebenen Stylesheets	►	i	11690	1	1	
der Maske 11690 zuge-		2	11690	2	2	
ordnet werden.	*					
	Da	tensatz:	I ◀ 1	▶ ▶ ▶ ▶ von	2	

Das Feld ord kennzeichnet die Reihenfolge der anzubietenden Stylesheets. Wir sehen hier, dass zuerst das generische Standard-Stylesheet angezeigt wird, und dann das Stylesheet Nr.2.

Defaultmäßig sind die Stylesheets für html (Druckversion in neuem Fenster), xml und text in jeder Ergebnistabelle enthalten. Die Stylesheets für rtf und pfg müssen in der obigen Tabelle zugeordnet werden dies ist sinnvoll, da die Standard-Stylesheets zunächst mit der in Frage kommenden Maske erprobt werden muss. Im PDF-Format z.B. muss man die Spaltenbreite nach der Textlänge bestimmen. Und das RTF-Fromat ist aufgrund des "experimentellen" Status von Jfor ebenfalls noch prüfungsbedürftig. In OpenOffice Version 1.1.x ist der erzeugte RTF-Code zum Beispiel unansehnlich, in Microsoft Word dagegen besser.

3.6.5.3 Anpassung an Lesegeräte

Der Vorteil von XML-Berichten ist, dass sie sich an individuelle Lesegeräte anpassen lassen. So können Sie die Standardoberfläche automatisch für das jeweilige Lesegerät anpassen und dadurch ganz individuelle Designs erzielen, z.B. auch für barrierefreie Angebote.

Das folgende Beispiel zeigt dies anhand des textbasierten HTML-Browsers **lynx**, der sich (zumindest am Anfang) gut zum Testen für barrierefreie Angebote eignet.

Klicken Sie jeweils auf die Grafik, um sie zu vergrößern.



Wir gehen auf das XML-Frontend, und erhalten die Anmeldemaske. Die Frame-Tags ignorieren wir.

Die rechte Abbildung

zeigt die SuperX-

lynx.

Homepage in einer

Konsole im Browser

miles - PuTTY	
	SuperX
FRAME: menue	
FRAME: maske	
Logo von SuperX	
Anmeldung	
superx	

Anmelden	
Anmelden	
asswort-Eingabefeld) Text Eingeben. Rauf- und	Runter-Pfeiltasten: Verlassen
Normale Tasten für Texteingabe verwenden.	
trl-U zum Löschen des ganzen Texts in einem Fe	eld, [Entf] um einen Buchstaben z

Nach erfolgreicher Anmeldung erscheint das Menü aus dem Themenbaum. Wir wählen hier als Beispiel die Abfrage **Benutzer von SuperX**.

Nun wird die Maske von dieser Abfrage angezeigt. Bei Kombinationsfeldern gehen wir auf das Feld, und drücken die Return-Taste. Es erscheinen die Auswahleinträge. Zum Abschluss gehen wir auf "Abschicken".

Es erscheint die Ergebnisanzeige. Dies sieht natürlich noch nicht besonders gut aus, weil textbasierte Browser und Tabellen sich nicht gut vertragen. Via Stylesheet lassen sich aber ganz übersichtlich Darstellungen entwerfen.

niles - PullY		<u> </u>	1
	Anmeldung (p	1 of 2	2
SuperX-Logo			
Abmelden			
Willkommen John Doe			
-Administration			
-Benutzer			
-Nutzungsprotokolle (intern)			
-Benutzer von SuperX			
-User einrichten			
-User löschen			
-Masken			
-Maske erzeugen			
-Maske kopieren			
-Maske löschen			
-Maske suchen			
-Felder			
-Feld erzeugen			
Leertaste für nächste Seite			
ile: Auf/Ab: andere Seite im Text. Rechts: Ver	rweis folgen; Links: zur	ück.	
lfe O)ptionen P) Druck G)ehe zu M) Hauptseite	Q) Beenden /=Suche <-=H	istory	y

niles - PuTTY	
	Eingabemaske
	Zur SuperX-Homepage
Benutzer von SuperX	
Benutzer	
Gruppe	
Sachgebie Administratoren	
Thema [Dezernenten]	
Administr Rektorat/Kanzler	
Archiv-Re	
Abschicken	
	and the second
swahlliste) Eingabetaste drücken und Pfe:	iltasten zur Optionswahl werwenden.
16. Obstants BL Daugh Challe an BL Haust	and a Resident (administration



Das Beispiel zeigt, dass durch XML und XSL keine Grenzen bei der Gestaltung von Benutzeroberflächen für SuperX existieren. Die obigen Stylesheets befinden sich als Muster im Verzeichnis

\$SUPERX_DIR/webserver/tomcat/webapps/superx/xml.

Wein kleiner Tipp noch für lynx: Wenn Sie das produzierte html überprüfen wollen, dann starten Sie lynx wie folgt:

lynx -trace http://localhost:8080/superX/xml/

Eine Logdatei lynx.trace wird in das aktuelle Verzeichnis geschrieben.

3.6.5.4 Einschränkungen des XML-Frontends

Das XML-Frontend arbeitet zwar ähnlich wie das Applet, aber es gibt ein paar wichtige Unterschiede. Die Felder einer Maske werden in einem Durchgang aufgebaut, während das Applet die Maskenfelder interaktiv füllt. Dies führt zu folgenden Einschränkungen:

- Es ist im XML-Frontend nicht möglich, im relation-Feld in Felderinfo dynamisch auf den Inhalt eines anderen Feldes (mit <<Feldname>>) zu verweisen.
- Es gibt keine Möglichkeit, den Organigramm-Stand zu verändern
- Der Institut-Button zeigt auch bei der art 4 nur die Einträge an, bei denen "lehre" = 1 gesetzt ist, d.h. die Lehreinheiten und Fakultäten. Alles andere würde zu einer starken Verlangsamung führen (gilt nur bei Organigrammen von über 500 Einträgen).
- Die Mehrfachauswahl ist in html über ein spezielles Listenfeld möglich, dass derzeit aber nur der Internet Explorer unterstützt, nicht Mozilla oder Netscape.
- Das XML-Frontend ist recht langsam und nach unserer Erfahrung auch recht instabil; zukünftige SuperX-Versionen werden sich dieses Problems annehmen.

3.6.5.5 Erweiterungen des XML-Frontends

Das XML-Frontend bietet gegenüber dem Applet einige Erweiterungen, die insbesondere für aufwändiger gestaltete Webapplikationen nützlich sind:

- Die Ergebnissseiten werden nicht komplett geladen, sondern im Rahmen von frei definierbaren Intervallen, z.B. 30 Datensätze pro Seite. Am Seitenende wird dann eine Navigationsmöglichkeit gelifert (Vorherige Seite / Nächste Seite). Der Intervall wird in \$SUPERX_DIR/webserver/tomcat/webapps/superx/WEB-INF/web.xml definiert (Parameter maxOffset).
- Die Ergebnisseiten können verlinkt werden, über spezielle Navigationsspalten (s.u.).
- In Feldern können Links zu anderen Masken definiert werden (Feldart 15).

3.6.5.5.1 Navigationsspalten im XML-Frontend

Wenn die Ergebnistabelle an das XML-Frontend übergeben wird, dann können spezielle Spalten für die Navigation eingesetzt werden. Die Spaltennamen werden im letzten select des select_stmt einer Makse übergeben.

nexttable	Link auf eine andere SuperX-Tabelle; der Inhalt des Feldes wird dann um den Pfad zum Servlet, (optional auch den String der Sessionid) und den Passus "SuperXmlTabelle?tid=" ergänzt, d.h. dem Servlet wird als erster Parameter die maskeninfo-tid übergeben. So wird z.B. aus dem Inhalt: 20010&id=2044
nextpage	 der Link http://<url der="" webapplikation="">>/servlet/SuperXmlTabelle?tid=20010&id=2044</url> Die Ergebnisseite wird dann um einen Button regänzt. Link auf eine andere SuperX-Tabelle wie nexttable, es wird nur ein anderes Laon und ein anderer Target genutzt
nextmask	Link auf eine andere SuperX-Maske; der Inhalt des Feldes wird dann um den Pfad zum Servlet, (optional auch den String der Sessionid) und den Passus "SuperXmlMaske?tid=" ergänzt. So wird z.B. aus dem Inhalt: 20010&id=2044 der Link http:// <url der="" webapplikation="">>/servlet/SuperXmlMaske?tid=20010&id=2044</url>
nextdelete	Die Ergebnisseite wird dann um einen Button ergänzt. Link auf eine andere SuperX-Maske; Im Unterschied zu nextmask wird hier ein anderes Icon gewählt: Die Ergebnisseite wird dann um einen Delete-
nextedit	Button Ergänzt. Link auf ein DBForms-Formular relativ zur URL des Servlets. die Ergebnis-
nextmail	Feldinhalte werden um einen Mailto-Tag ergänzt. z.B. info@superx-projekt.de
url	<pre>wird zu <a info@superx-projekt.de"="" mailto:"=""> info@superx-projekt.de Feldinhalte werden um einen href-Tag (sowie wenn nötig um ein "http" er- gänzt. z.B. www.superx-projekt.de</pre>
nextlink	wird zu www@superx-projekt.de Link auf eine externe Seite oder eine andere SuperX-Tabelle; anders als bei nexttable wird ein frei wählbarer textueller Link angegeben, wobei der Voll- text des Links und der eigentliche Linkt durch ein Trennzeichen "l" getrennt sind
	So wird z.B. der Feldwert "Erläuterungenlhttp://www.erlaeuterungen.de" wie folgt ersetzt: Erläuterungen Wenn nach dem Trennzeichen keine externe Web-Adresse angeboten wird (erkennbar am vorangestellten "http:"), dann wird der Inhalt des Feldes um den Pfad zum Tabellen-Servlet ergänzt: So wird z.B. aus dem Inhalt: Details zur Hochschule 20010&id=2044 der Link
	Details zur Hochschu- le

3.6.5.5.2 Export von Abfragen nach RTF, PDF und XLS

Im XML-Frontend können Abfragen direkt nach html (Druckversion), XML, RTF (-> z.B. Word), PDF oder XLS (->Excel) exportiert werden. Die zugehörigen Stylesheets lauten:

html	<pre>\$SUPERX_DIR/webserver/tomcat/webapps/superx/xml/tabelle_html_p.xls</pre>
(Druck-	
version)	
RTF	\$SUPERX_DIR/webserver/tomcat/webapps/superx/xml/tabelle_fo_rtf.xsl
PDF	<pre>\$SUPERX_DIR/webserver/tomcat/webapps/superx/xml/tabelle_fo_pdf.xsl</pre>
XLS	\$SUPERX_DIR/webserver/tomcat/webapps/superx/xml/tabelle_xls.xsl

Die PDF- und RTF-Konverter arbeiten mit der OpenSource-Bibliothek FOP, der Excel-Konverter mit POI. Die Vorlagen können als Grundlage für eigene Stylesheets verwendet werden. Wir verweisen hier auf der SuperX-Entwicklerhandbuch.

Der PDF- und RFT-Export funktioniert zwar technisch, aber leider sehen die Ergebnisse oft nicht "schön" aus, da die Berichte in SuperX generell über die Seitenbreite hinaus gehen. Wir empfehlen daher, die Exporte nur bei speziell geeigneten Berichten (mit weniger Spalten) zu verwenden. Außerdem gibt es für Volltexte keine Silbentrennung.

Der XLS-Export wurde mit MS Excel (95-2003) und OpenOffice (1.1.3-2.x) getestet. Da die Produkte automatisch auf Seitenbreite skalieren können, sieht der Export hier deutlich besser aus.

Außerdem können gurdlegende Lauyoutelemente wie Kopf- und Fußzeilen und Seitenzahlen individuell angepasst werden, ohne zwingend XSLT-Kenntnisse zu haben. Schauen Sie einfach in die für das Seitenformat in die Datei <code>\$SUPERX_DIR/webserver/tomcat/webapps/superx/xml/pageComponents.xsl;</code> dort wird im Abschnitt <xls-pageformat> definiert, welche Einträge wie gefüllt werden sollen.

Des realts Deisniel	- <xsl:template name="xls_PageFormat"></xsl:template>				
Das lecille Beispiel	- <xls_pageformat></xls_pageformat>				
zeigt die Vorgabewerte	fontSize="12" klappt leider noch nciht <xls_header-left></xls_header-left>				
des Seitenformats einer	<xls_header-center>SuperX</xls_header-center> - <xls_header-right font="Times" fontstyle="normal"></xls_header-right>				
Excel-Datei. Die Kno-	<xsl:call-template name="today"></xsl:call-template> 				
ten und Attributnamen	<xls_footer-left></xls_footer-left> <xls_footer-center></xls_footer-center>				
sollten selbsterklärend	<xis_footer-right_font="times" fontstyle="Italic">Seite \$PAGE von \$NUMPAGES <xis_autobreaks>true</xis_autobreaks></xis_footer-right_font="times">				
sein.	wenn nicht gewünscht, dann bitte ganzen Tag entfernen <xls_fitheight>100</xls_fitheight>				
	soll Tabelle in Höhe auf X Seiten passen? Default ist 100, statt eingebaut 1 <xls_fitwidth>1</xls_fitwidth>				
	soll Tabelle in Breite auf eine Seite passen? <xks landscape="">true</xks>				
	<pre></pre>				
	<pre>- modulch ist auch "letter"></pre>				

Sie können den gesamten Knoten kopieren in die hochschulspezifische pageComponents_final.xsl-Datei, sie überlagert automatisch die "normale" pageComponents.xsl und ist sofort in allen Berichten aktiv. So können Sie z.B. im Berichtskopf den Hochschulnamen und die Abteilung eintragen.

3.7 Installation von Modulen

Das Kernmodul enthält außer Adminnistrationsabfragen und Tabellen keinerlei Inhalte. Die Inhalte werden in Form von Modulen hinzugefügt. Dazu gibt es vorgefertigte Installationsscripte (S. 98).

3.7.1 Architektur von SuperX-Modulen

Die folgende Abbildung zeigt die Architektur von Modulen ab Beispiel vom HISCOB-Modul:



Die Abbildung zeigt, dass ein Modul eigene Komponenten nutzt, aber auch auf Teile des Kernmoduls zugreift, z.B. das Orgranigramm - dies macht SuperX zu einem integrierten System. Neben dem Organigramm sind alle anderen Komponenten des Kernmoduls natürlich betroffen, z.B. Themenbaum, Userrechte.

Die Ordnerstruktur eines Moduls spiegelt die Komponenten des Systems wieder. Es gibt je ein Verzeichnis für datentabellen, schluesseltabellen und hilfstabellen.

Die Installation eines Moduls ist in der Dokumentation des jeweiligen Moduls näher beschrieben. Module, die auf dem Kernmodul 2.1 oder höher basieren, haben einen einheitlichen Aufbau.

3.7.2 Modulscripte im Kernmodul

Seit Version 2.1 werden die Datenbankschemata und Scripte der Module in einem einheitlichen Format zusammengestellt und in einer Datei *\$SUPERX_DIR/db/module/\$MODULNAME/conf/\$MODULNAME.xml* gespeichert. Das XML-Format hat den Vorteil, dass die Scripte dynamisch für Postgres und Informix erzeugt werden können, und dass die Scripte vereinheitlicht werden. Aus dieser Datei werden die Scripte erzeugt, die das Modul jeweils für Postgres und Informix installieren / updaten /aktualisieren / überprüfen und entfernen. Die folgende Abbildung zeigt das Vorgehen:

Aus der xml-Datei werden die jeweiligen Scripte für die Installation, den Update, die Extraktions-, Transformations- und Ladescripte (ETL) und die Deinstallation erzeugt.



Die Modul-Scripte liegen als Shellscripte im Verzeichnis \$SUPERX_DIR/db/bin, und sind an anderer Stelle (S. 98) im Detail erläutert.

module_install.x < <name>> <<pfad>></pfad></name>	Installiert ein Modul, d.h. erzeugt die Tabel-
	len, Views, Funktionen und Abfragen. Die
	Abfragen werden in den Themenbaum einge-
	hängt.
module_etl.x < <name>> <<pfad>></pfad></name>	Aktualisiert das Modul, d.h. lädt die Rohdaten,
	Transformiert sie, und aggregiert die Hilfsta-
	bellen. Fehler werden in eine Log-Datei ge-
	schrieben, diese kann dann per Mail an einen
	Admin versandt werden. Wenn das Script er-
	folgreich durchläuft, wird die Tabelle sys-
	teminfo aktualisiert.
	Das Script wird nicht direkt als Cronjob aufge-
	rufen, sondern von einem Shellscript, das die
	Umgebungsvariablen und Parameter setzt, z.B.
	cob_update.x.
module_drop.x < <name>> <<pfad>></pfad></name>	Entfernt das Modul bzw. die Tabellen, Views,
	Funktionen und Abfragen.
module_update.x < <name>> <<pfad>></pfad></name>	Ein Modulupdate wird durchgeführt, d.h. eine
	neue Version des Moduls wird installiert
	(nicht zu verwechseln mit dem regelmäßigen
	Update im Sinne eines ETL-Prozesses).
module_scripts_create.x	Erzeugt via XSL-Transformation die jeweili-
< <name>></name>	gen Scripte, die in den obigen Scripten aufge-
< <pfad>></pfad>	rufen werden, z.B. bau_install_ids.sql für das
< <datenbanksystem(optional,< th=""><th>Installationsscript des Baumoduls beim Infor-</th></datenbanksystem(optional,<>	Installationsscript des Baumoduls beim Infor-
INFORMIX, POSTGRES)>>	mix Dynamic Server.
< <versionsnr.>></versionsnr.>	

Neben den operativen Scripten erzeugt module_scripts_create.x auch html-Dateien zur Dokumentation eines Moduls in

\$SUPERX_DIR/db/module/<<Modulname>>/conf/<<Modulname>>.html

(auch als rtf-Datei zu Einbindung in Modul-Dokumentationen) sowie zur Schnittstelle in

\$SUPERX_DIR/db/module/<<Modulname>>/rohdaten/<<Modulname>>_unload.html

Darüber hinaus werden auch DBForms-Formulare erzeugt.

3.7.3 Installation eines Moduls: Allgemeines Vorgehen

Das Vorgehen bei der Installation eines Moduls ist standardisiert.

- 1. Entpacken Sie das Modul in \$SUPERX_DIR
- 2. Einrichtung der Umgebung in SQL_ENV (S. 94); für jedes Modul sollten Beispielvariablen in SQL_ENV_<<Modulname>>.sam stehen. Bitte legen Sie hier auch Email-Adressen für log- und Fehlermails an.
- 3. Entladen der Rohdaten; auch hier müssen Umgebungsvariablen auf dem Quellrechner angepasst werden

(<<Modulname>>_ENV)

- 4. Kopieren der Rohdaten nach <<Modulpfad>>/rohdaten. Neuere SuperX-Module haben dafür vorgefertigte Scripte mit dem Namen <<Modulname>>_copy.x
- 5. Bei sehr großen Datenmengen bietet es sich an, die Rohdaten zunächst auf überschaubare kleine Dateien zu kürzen. Das Script

<<Modulname>>_shrink.x

kürzt alle "*_neu"-Tabellen auf 100 Zeilen. Sie können dies später mit <<Modulename>>_unshrink.x rückgängig machen.

- 6. Installieren Sie das Modul mit <<Modulname>>_erzeugen.x
- 7. Wenn die Installation erfolgreich war, können Sie das Modul aktualisieren mit <<Modulname>>_update.x (ggf. mit Parametern) d.h. die ETL-Prozesse (S. 99) werden gestartet (s.u.).
- 8. Wenn das Modul erfolgreich aktualisiert ist, wird eine Prüfprozedur gestartet, die die Daten plausibilisiert. Fehler und Warnungen finden Sie in der Datei \$<<Modulname>>_ERRORDAT.

3.7.3.1 Dateitransfer mit scp/rsync

Für den Transfer der Rohdaten wird in SuperX die dateibasierte Schnittstelle genutzt. Unter UNIX läßt sich dieser Transfer vollends automatisieren, indem die Programme scp oder rsync auf der Basis des OpenSSH-Pakets genutzt werden^{xxii}. Beide setzen auf das ssh-Protokoll 2 auf und stellen somit einen verschlüsselten Dateitransfer sicher.

In den jeweiligen Modulen wird im Verzeichnis rohdaten eine Beispieldatei mit dem Namen <<MODULNAME>>_ENV.sam ausgeliefert, die Sie umbenennen können nach <<MODULNAME>>_ENV. Darin werden am Ende der Datei die Parameter zum Kopieren festgelegt, also die Userkennung REMOTE_USER, der Hostname REMOTE_HOST, und die Methode des Kopierens (COPY_METHOD) sowie die jeweiligen Zielpfade. Diese Umgebungsvariablen werden von dem jeweiligen Script <<modulname>>_copy.x benutzt. Damit die Passworteingabe entfällt, muss man wie folgt vorgehen:

Loggen Sie sich zunächst testweise einmal ein. Wenn Sie z.B. vom COB-Server auf den SuperX-Server kopieren wollen, loggen Sie sich als user cob auf cobhost ein mit ssh superx@superxhost

Beim ersten Mal müssen Sie die Sicherheitsabfrage mit "yes" bestätigen.

Erzeugen Sie auf dem Quellrechner einen öffentlichen Schlüssel mittels ssh-keygen -t rsa, wobei man eine leere Passphrase vergibt (Achtung: mögliche Sicherheitslücke!). Der öffentliche Teil dieses Schlüssels (~/.ssh/id_rsa.pub) muss auf dem Zielrechner in die Datei ~/.ssh/authorized_keys eingefügt werden, ggf. muss die Datei neu erzeugt werden.

Wenn z.B. auf dem COB-Server unter der Kennung cob ein Key wie folgt erzeugt wurde:

 Beispieleintrag eines
 ssh-rsa AAAAB3Nza...[hier viele kryptische Zeichen]

 Public Keys
 ...pg6VkCc= cob@cobhost

Dann wird genau diese Zeile in der Datei /home/superx/.ssh/authorized_keys angefügt (die Datei kann mehrere PublicKeys enthalten, ein Eintrag pro Absatz).

Danach sollte z.B. der Login vom cobhost als user cob mit ssh superx@superxhost ohne Passworteingabe klappen. Wenn nicht, schalten Sie das Logging mit ssh -v superx@superxhost ein. Eine Möglichkeit ist, dass die PublicKey-Authentifizierung in der Konfigurationsdatei des SSHD (normal /etc/ssh/ssh_config) abgeschaltet ist.

Sie können außerdem noch einschränken, von welchem Host die obige Authentifizierung ermöglicht wird. Dazu setzen Sie den Parameter "from=*.uni-xy.de" davor, z.B.

Einschränkung	from="*.uni-xy.de" ssh-rsa AAAAB3Nza[hier viele kryptische
"from"	Zeichen]pg6VkCc= cob@cobhost
in authorized_keys	

Wenn Sie die Kopiermethode scp benutzen, und die obige "authorized_keys"-Metohde mit PublicKey nicht nutzen wollen, können Sie auch mit Private Keys arbeiten (siehe SSH-Doku). Dazu können sie in der *_ENV-Datei in dem Parameter SCP_OPTS den Verweis auf den private Key setzen.

SCP_OPTS in *_ENV: SCP_OPTS="-p -B -i /home/cob/.ssh/superx_key" Beispiel COB_ENV export SCP_OPTS

3.7.3.2 Modulupdate in mandantenfähigen Installationen

Der Modulupdate in mandantenfähigen Installtion findet in einer SuperX-Installation statt, allerdings werden die einzelnen Scripte mit unterschiedlichen Umgebungsvariablen, wie sie in

```
\texttt{SQL\_ENV.} <\!\!<\!\!\texttt{MANDANTID}\!\!>\! definiert \ ist, \ z.B. \ \texttt{SQL\_ENV.FHRO}.
```

In der SQL_ENV.<<MANDANTID>> werden unterschiedliche Pfade für den jeweiligen *_LOAD_PFAD gesetzt, wobei in der Regel die Mandandid ein Unterverzeichnis vom "normalen" LOAD_PFAD ist. So ist z.B. beim COB-Modul folgender Pfad anzusetzen:

Normale SuperX-Installation:

COB_LOAD_PFAD=\$SUPERX_DIR/db/module/cob/rohdaten

Mandantenfähige SuperX-Installation:

COB_LOAD_PFAD=\$SUPERX_DIR/db/module/cob/rohdaten/FHRO

Unterhalb von FHRO befindet sich noch einmal die Entladeroutine sowie das Unterverzeichnis unl mit den Rohdaten. Dieses Verzeichnis FHRO kann der Einfahheit halber auch ein symbolischer Link auf den gemounteten COB-Rechner sein.

Durch Setzen der Mandantennummer in der Umgebungsvariable MANDANTID in der jeweiligen SQL_ENV des Mandanten werden die ETL-Scripte anders ausgeführt: Die Logdateien werden jeweils mit der Mandantennummer versehen (z.B. L_cob_updateFHRO.log), damit die Übersicht nicht verloren geht und der gleichzeitige Update mehrerer Mandanten in eine rsuperX-Installation möglich ist.

Außerdem können weitreichende Steuerungsmechanismen im Modulupdate eingesetzt werden: Nach jedem ETL-Schritt können optional mandantenspezifische Scripte aufgerufen werden. Diese müssen folgende Namenskonvention einhalten:

<<Scriptname>>_<<MANDANTID>>.sql

Also für eine hochschulspezifische Transformation im COB-Modul des Mandanten FHRO wird eine Datei namens

```
cob_trans_FHRO.sql
```

mit entsprechenden SQL-Anweisungen angelegt.

3.8 Überwachung und Performance

SuperX besteht aus verschiedenen Komponenten, die jeweils eigene Überwachungsmerkmale und Performance-Mechanismen besitzen.

3.8.1 Überwachung und Performance der Webanwendung

Die Webanwendung baiert auf Tomcat, und die Logdateien des Tomcat liegen standardmäßig im Verzeichnis \$SUPERX_DIR/webserver/tomcat/logs. Die Logdateien im Einzelnen:

- Logging von Tomcat: catalina.out bzw. localhost.xxx.out
- Logging der SuperX-Webanwendung jeweils in superx_default.log (statt "default" ggf. die Mandantenid) für allgemeines SQL-Logging, und superx_default_xml.log für das Logging der XML-Ausgabe des XML-Frontends.
- dbforms.log für Logging der DBForms-Komponente

Alle Logging-Ausgaben lassen sich flexibel an verschiedenen Stellen steuern:

• Das Tomcat-Logging lässt sich in der Datei \$SUPERX_DIR/webserver/tomcat/common/classes/log4j.properties steuern

- Das Ausmaß des Loggings der SuperX-Webanwendung: Im propadmin (S. 45) wird der Logging-Level für die SQL-Ausgabe sowie für die XML-Ausgabe festgelegt.
- Das Logging für DBFORMS wird in der Datei \$SUPERX_DIR/webserver/tomcat/webapps/superx/WEB-INF/log4j.properties festgelegt.
- Das Logging der java-bezogenen SuperX-DB-Anwendung wird in der Datei \$SUPERX_DIR/db/conf/logging.properties gesteuert.

Die SQL-Scripte der SuperX-Abfragen können in der Java-Konsole des Applets sowie in der o.g. superx_default.log eingesehen werden. Bitte beachten Sie dabei, dass bei SQL-Fehlern nur im Entwicklungsmodus die genaue Stelle des Auftretens ermittelt werden kann.

3.8.1.1 Steuerung des SQL-Logging im SuperXManager

Bei der Auslieferung von superX werden alle SQL-Befehle einzeln an den Server übertragen und ausgeführt, um im Falle von Fehlern schnell die Position ermitteln zu können. Dieser Modus nennt sich "Entwicklungsmodus" und ist im propadmin (S. 45) die Voreinstellung, kann aber an dieser Stelle auch geändert werden. Die Änderung wäre nach dem Start des Tomcat aktiv.

Einige Parameter des propadmin lassen sich auch im laufenden Betrieb umstellen. Im SuperX-Manager-Servlet können Sie z.B. den Entwicklungmodus umschalten.

Gehen Sie dazu im XML-Frontend auf ein Bearbeitungsformular von Benutzerrechten und klicken Sie oben rechts auf den Link Su-	SuperX Nach Änderungen sollten Sie den Cache aktualisieren SuperXManager	
perXManager		
Der Schalter Entwick-	SuperX Manager	
lungsmodus an/aus	SuperX 3.0rc3 (build 25.01.2006 22:06)	
lässt sich auch im lau-		
fenden Betrieb umstel-	Server-Cache aktualisieren	
len. Klicken Sie dazu	Es werden Enzumerungen und Übersetzungen (capitons) gedaant. Fürs XML-Frontend auch Üser, Üserrechte und Sichten,	
auf an/aus und dann auf	Entwicklungsmodus ist AN @ an @ aus []hemehmen	
"Übernehmen". Auch	(Im Entwicklungsmodus werden alle SQL-Befehle von Abfragen einzeln an die Datenbank geschickt. Das dauert länger, ermöölicht aber besserer Fehlermeldungen)	
die Tomcat-logs lassen		
sich hier leeren. Unten	Loq-Dateien leeren	
zeigt der Manager je- weils den letzten SQL vor bzw. nach Freemar-	letzter Masken sql (vom XML-Frontend) for FreeMarker-Transformation Abfrage Maske User suchen (71030) durchführen 09:32:52 Abfrage	xml versi<br <ergebnisse isessionid=</ergebnisse
ker-Transformation an, sowie den letzten XML- Strom.		

In Produktionsumgebungen emfpehlen wir, den Entwicklungsmodus abzuschalten, da die Abfragen dann 25-50% schneller laufen.

3.9 Downloads einrichten und verteilen

SuperX bietet die Möglichkeit, beliebige Dateien über die Webapplikation an Anwender auszuliefern, z.B. um einen Downloadbereich einzurichten. Die Downloads können einzelnen Usern oder Gruppen sowie Institutionen und Themen zugeordnet werden.

Die Download-Dateien werden in dem geschützten Verzeichnis der Webapplikation in \$SUPERX_DIR/webserver/tomcat/webapps/superx/WEB-INF/downloads abgelegt. Gleichzeitig werden der Dateiname und diverse andere Metadaten in der Tabelle sx_downloads gespeichert. Wenn ein Anwender einen Download abruft, dann wird die Datei im SuperX-Servlet geladen und über http(s) ausgeliefert.

Die Auslieferung von Dateien wird defaultmäßig **protokolliert** und kann über die Maske "Downloadstatistik" abgerufen werden. Sie können diese Funktionalität (z.B. aus Datenschutzgründen) sperren, indem Sie die Konstante "DOWNLOAD_PROTOKOLL" statt auf "1" auf "0" setzen - damit werden keine Download-Aktivitäten in SuperX protokolliert (was aber nicht bedeutet, dass dies auch im Webserver-Log nicht mehr passiert, die dortige Protokollierung sowie die Tomcat-eigene Protokollierung ist davon unabhängig).

3.9.1 Tabellenstruktur

Feldname	Feldtyp	Grö- ße	De- fault	Not Null	Beschreibung	
tid	SERIAL	4		true	Primärschlüssel	
name	CHAR	255		false	Titel	
ch110_institut	CHAR	10		false	Kostenstelle/Institut	
bezugsdatum	DATE	4		false	(für Ermittlung Bezugsjahr,- Monat oder Sem.)	
importdatum	DATE	2		false	Datum des Imports in die SuperX-Datenbank	
kommentar	TEXT	32000		false	Kommentar für Website (Datenlegende o.ä.).	
kommen- tar_www	CHAR	255		false	Verweis auf andere Website für längere und gelay- outete Kommentare oder Dokumentationen.	
contenttype	CHAR	50		false	Mime-Type der Datei (pdf, html etc).	
datei	CHAR	255		true	Pfad zum geschützten Verzeichnis (relativ zu \$SU- PERX_DIR/webserver/tomcat/webapps/superx/WEB- INF/downloads)	
gueltig_seit	DATE	2		false	Soll Download angezeigt werden von	
gueltig_bis	DATE	2		false	Soll Download angezeigt werden bis	

Es gibt eine Tabelle sx_downloads angelegt mit folgenden Feldern:

Desweiteren gibt es eine Tabelle sx_keywords zur Erhebung der Stichworte:

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
tid	SERIAL	4		false	Tupelidentifier
name	CHAR	255		false	Stichwort
parent	INTEGER	4		false	Übergeordnetes Stichwort Wird derzeit noch nicht ausgewertet.

Die Zuordnung zwischen Download und Stichwort findet in der Tabelle download_keyw_bez statt:

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
keyword_id	INTEGER	4		false	
download_id	INTEGER	4		false	

3.9.2 Berechtigung für Downloads

Die Berechtigungen für die Downloads werden über die SuperX-Gruppen- bzw. Userrechte verwaltet. Dazu werden eigene Tabellen user_download_bez und group_download_bez erzeugt, für die auch Pflegeformulare (S. 140)existieren. Die Institutions-Berechtigung wird auch Bordmitteln von SuperX realisiert, d.h. die Anwender erhalten über ihre Zuordnung zur jeweiligen Kostenstelle in der Tabelle user_institution das Recht für die Kostenstelle und alle jeweils untergeordneten Kostenstellen.

Einzelne vorgefertigte Masken sind bereits eingerichtet und werden im Folgenden beschrieben.

3.9.3 Masken zur Erzeugung und Verteilung von Downloads

Im XML-Frontend finden Sie die Download-Masken im Themenbaum-Ast "Administration".

3.9.3.1 Download suchen

Mit der Maske "Download suchen" können sie einzelne Downloads einrichten, bearbeiten oder löschen.

In der Suchmaske kön-	Download suchen	
nen Sie verschiedene		
Parameter einschränken.	Bitte schränken Sie Ihre Auswahl ein:	
Wenn ein Stichwort	Stichwort	
oder eine Kostenstelle	Institution	
ausgewählt wird, dann	Jahr 💗 Monat 👽	<u> </u>
werden alle Downloads	Suchwort	
mit diesem oder unter-		
geordnetem Stichwort/		
Kostenstelle gefunden.	Abschicken Zurücksetzen	

Das Freitext-Feld Suchwort bezieht sich auf den Namen des Downloads.

Die Ergebnistabelle	Download suchen
zeigt die Downloads.	Institution: FB ABV gesamt ; Stand: 01.01.2003
Wenn Sie als Administ-	Name Kommentar , adap postheiten User- und Stichworte
rator gekennzeichnet	Test
sind (Feld administrati-	
on in userinfo steht	Datensatz 1 – 1 von insgesamt 1 Satz.
auf "1"), dann können	
Sie die Downloads nicht	
nur laden, sondern auch	
bearbeiten sowie zu	
Usern/Gruppen bzw.	
Themen zuordnen.	

3.9.3.2 Download bearbeiten: Metadaten und Dateien

In der Bearbeitungsmaske erscheinen die oben (S. 137)beschriebenen Felder nebst Erläuterungen.

Downloads		In die	esem Formular können Sie Downloads bearbeiten.
tid 1 Name Test			
Zugehörige Kostenstelle		v	
Bezugsdatum 01.12.2006	_		Worauf bezieht sich der Download (für Ermittlung Bezugsjahr Monat oder Sem.)
Importdatum 01.11.2006			Datum des Imports in die SuperX-Datenbank
Kommentar			Kommentar für Website (Datenlegende o.ä.).
Hyperlink zu Kommentar-Site			Verweis auf umfangreichere Anleitungen und Kommentare, beginnend mit "http://" oder relativ zu http://superx-server:Port/superx/servlet
MimeType application/	/pdf		Mime-Type der Datei (für pdf z.B. "application/pdf", html etc
D ^{atel} Aktuelle I Neue Dat	Datei: test.pdf ei: Browse…		Pfad zur Datei (relativ zu \$SUPERX_DIR/webserver/tomcat/webapps/superx/WEB-INF/)
Gültig von 01.01.1900	_		
Gültig bis 31.12.3000			
Speichern Erster V	orheriger Nächster Letzter K	opieren Löschen Neu	

Sie können, müssen aber nicht, einem Download einer einzelnen Kostenstelle zuordnen. Hierarchische Anordnungen werden dabei suchbar, d.h. wenn ein Anwender in der Insitutions-Sicht des Organigramms eine Kostenstelle auswählt, dann werden alle Downloads mit untergeordneten Kostenstellen ebenfalls gefunden.

Sie können Dateien Hochladen, inden Sie in der Zeile Datei eine neue Daten festlegen. Ansonsten wird darüber der aktuelle Dateiname festgelegt. Wichtig ist, dass der Dateiname in dem Verzeichnis \$SUPERX_DIR/webserver/tomcat/webapps/superx/WEB-INF/downloads eindeutig ist. Außerdem funktioniert der Browser-basierte Upload nur mit kleinen Dateien, größere Dateien sollten Sie manuell in das Verzeichnis \$SUPERX_DIR/webserver/tomcat/webapps/superx/WEB-INF/downloads kopieren.

Sie können auch Datensätze kopieren, allerdings werden nur die Metadaten werden kopiert, sie müssen dann eine neue Datei hochladen.

3.9.3.3 User- und Gruppenrechte auf Downloads

Mit der Schaltfläche unter "User- und Gruppenrechte" sehen Sie das Bearbeitungsformular.

Sie können jeweils ein-	📕 Formular Downloads - Mozilla	
zelne User oder Grup- pen zuordnen, die Funk-		SuperX
tionalität entspricht der	Downloads In diesem Formular können Sie die Lese-Ber	rechtigung für Downloads bearbeiten
Berechtigung für	Name des Download Test	Bezeichnung des Download, die für User angezeigt wird
Sichten (S. 118).	tid 1 User, die diesen Download sehen dürfen: Markierten User löschen Neuen User Ver Ver Ver Ver Ver Ver Ver Ver Ver V	Interne Nummer Hier können Sie einzelne User für den Download berechtigen.
	Gruppen, die diesen Download sehen dürfen: Markierte Gruppe löschen Neue Gruppe Dezementen1 Neue Gruppe einfügen	Hier können Sie einzelne Gruppen für den Download berechtigen.
	< <erster <="" nächster="" orheriger=""> Letzter >></erster>	

3.9.3.4 Stichworte für Downloads

Mit der Schaltfläche unter "Stichworte zuordnen" sehen Sie das Bearbeitungsformular.



4 Bestandteile des Kernmoduls: Die Referenz

Das Kernmodul besteht aus der Userverwaltung, der Maskenverwaltung sowie aus dem Organigramm. Die wichtigsten Tabellen des Kernmoduls sind im folgenden aufgeführt.

Die Basisdaten und zusätzlichen Schlüssel der weiteren Module werden nach folgender Konvention nach SuperX übernommen:

Tabellenname	<basisdatenbank>_<name basisdatenbank="" der="" in="" tabelle=""></name></basisdatenbank>
Beispiele	"cob_busa" oder "mbs_inst"

4.1 Die Userverwaltung

Die User- und Gruppenverwaltung geschieht über eine Reihe von Tabellen, die User, Gruppen, Institutionen, thematische Sachgebiete und einzelne Abfragen in Beziehung setzen. Die Tabellen werden im folgenden dokumentiert. Client-Formulare zur Benutzerverwaltung finden Sie hier (S. 106).

4.1.1 Verwaltung einzelner User

Die Userverwaltung beruht auf einige Tabellen, die Stammdaten und Beziehungen zu anderen Tabellen (z.B. Masken) abbilden.

4.1.1.1 Tabelle userinfo

Die Tabelle userinfo enthält die Angaben zur Person, d.h. email, Nutzerkennung, Passwort etc.

	7	
Feld	Erläuterung	Beispiel
tid	Id-Nummer	1
benutzer	Nutzerkennung für Anmeldung	testuser
kennwort	Passwort alter Client	frsgrgr
name	Name der Person	Ein Testuser ohne
		Name
max_versuch	Maximale Logins	5
akt_versuch	Aktuelle Login-Versuche	0
email	Email-Adresse	test@uni.de
administrator	Administrator-Rechte (1=ja, 0=nein)	0
archiv_recht	Leserechte auf Archiv (1=ja, 0=nein)	0
	[im alten Client]	
passwd_sha	Passwort neuer Client (SHA-1-verschlüsselter HexCo-	
	de)	
info	Beschreibung des Users (Sachgebiet, Fachgebiet)	SG 22

Tabelle userinfo

Diese Tabelle kann manuell gepflegt werden und bildet die Grundlage für die Benutzerverwaltung in SuperX. Die Tabelle wird mit mehreren Tabellen verknüpft, z.B. user_group_bez.

Die Benutzerpassworte werden sha-verschlüsselt gespeichert. Sie können auch externe Daten in diese Tabelle füllen, z.B. aus einer anderen Benutzerverwaltung. Die Passwort-Verschlüsselung müssen Sie allerdings selbst vornehmen. Unter PostgresSQL kann dieses Verfahren elegant automatisiert werden, dort gibt es mit dem Paket pgcrypto (S. 30) eine Möglichkeit, SHA-Digests scriptgesteuert zu erzeugen. Die Syntax lautet:

select encode(digest('<<Klartext-Passwort>>', 'sha1'), 'hex');

4.1.1.2 Tabelle user_masken_bez

Rechte einzelner User für einzelne (Abfrage-)Masken werden in der Tabelle user_masken_bez gespeichert:

Feld	Erläuterung	Beispiel
userinfo_id	Nummer des Benutzers	1
	(entspricht dem Feld tid in der Tabelle userinfo)	
maskeninfo_id	Nummer der Maske	10050
	(entspricht dem Feld tid in der Tabelle maskeninfo	
	s.u.)	

Tabelle user	_masken_	_bez
--------------	----------	------

Der *testuser*, der den tid-Eintrag 1 in der Tabelle userinfo hat, bekommt Zugriffsrechte für die (Abfrage-)Maske mit der Nummer 10050 (Studierende allgemein).

Falls Sie Einzelrechte für (Abfrage-)Masken vergeben wollen, machen Sie einen Eintrag in der Tabelle user_masken_bez.

4.1.1.3 Tabelle sachgebiete

Damit man nicht jedem Benutzer für jede (Abfrage-)Maske einzeln Rechte geben muss, gibt es sog. Sachgebiete.

In SuperX gibt es u.a. die Sachgebiete

- Studierende/Prüfungen
- Personal/Stellen
- Gebäude/Räume/Flächen
- Kennzahlen
- Haushalt.

Diese Sachgebiete finden sich in der Tabelle sachgebiete.

Tabelle sachgebiete

Feld	Erläuterung	Beispiel
tid	Nummer des Sachgebiets	1
name	Name des Sachgebiets	Studierende/Prüfungen

4.1.1.4 Tabelle sachgeb_maske_bez

In der Tabelle sachgeb_maske_bez wird die Beziehung von (Abfrage-)Masken zu Sachgebieten festgelegt.

Feld	Erläuterung	Beispiel
sachgebiete_id	Nummer des Sachgebiets	1
	(entspricht dem Feld tid in der Tabelle sachgebie-	
	te)	
maskeninfo_id	Nummer der Maske	10050
	(entspricht dem Feld tid in der Tabelle maskeninfo	
	s.u.)	

Tabelle sachgeb_maske_bez

Das Beispiel bedeutet, dass die Abfrage Studierende Allgemein (Nr. 10050) zum Sachgebiet Nr.1 (Studierende/Prüfungen) gehört.

4.1.1.5 Tabelle user_sachgeb_bez

Man kann dann einem Benutzer Zugriffsrechte für ein komplettes Sachgebiet (z.B. Studierende/Prüfungen) geben. Dazu macht man einen Eintrag in die Tabelle user_sachgeb_bez.

	C	
Feld	Erläuterung	Beispiel
userinfo_id	Nummer des Users	1
	(entspricht dem Feld tid in der Tabelle userinfo)	
sachgebiete_id	Nummer des Sachgebiets	1
	(entspricht dem Feld tid in der Tabelle sachgebie-	
	te)	

Tabelle user_sachgeb_bez

Das Beispiel bedeutet, dass der User Nr. 1 (*testuser*) alle Masken aufrufen darf, die zum Sachgebiet Nr. 1 (Studierende/Prüfungen) gehören. Da über die Sachgebietseintragungen Rechte verwaltet werden, sollte hier pro Maske nur eine Eintragung erfolgen (im Gegensatz zum alten Client).

4.1.1.6 Tabelle user_institution

Weiterhin kann festgelegt werden, für welche Institutionen ein Benutzer Informationen erhalten darf. Die Tabelle user_institution ordnet die Leserechte einer Person auf die Institutionen im Organigramm zu. Wenn ein User alle Institutionen einsehen darf, dann ist es notwendig, einen Datensatz zum User mit dem Wert 0 im Feld ch110_institut hinzufügen (Gültigkeitszeitraum beachten!). Der Übersichtlichkeit halber sollte man im Feld Name dann "Alle" eintragen.

In der Downloadversion des Kernmoduls enthält diese Tabelle zwei Beispielsätze: der User "admin" hat die Leserechte auf einen Fachbereich und auf alles.
Feld	Erläuterung	Beispiel
userid	Nummer des Benutzers	1
	(entspricht dem Feld tid in der Tabelle userinfo)	
name	Name der Institution	TestFB
	(entspricht dem Feld name in der Tabelle organi-	
	gramm)	
	Eingaben nicht unbedingt erforderlich	
ch110_institut	Nummer der Institution	11
	(entspricht dem Feld key_apnr in der Tabelle organi-	
	gramm)	
hierarchie	Hierarchieebene (0,1,2,3)	
	[wird in Zukunft für COB-Abfragen benutzt]	
gueltig_seit	Gültigkeit der Rechte: Anfang	1.1.2001
gueltig_bis	Gültigkeit der Rechte: Ende	31.12.2001
lehre	Freigabe einer Lehrheit bei lehrbezogenen Abfragen	null
	Wenn ein User z.B. keine Rechte für die Lehreinheit	
	Physik hat, sondern nur für das untergeordnete Institut	
	A, kann man hier beim Eintrag der Lehreinheit Physik	
	das Feld auf 1 setzen, damit der User bei lehrbezogenen	
	Abfragen (z.B. Mittelberechnung) trotzdem die Rechte	
	für die Lehreinheit erhält. In allen anderen Fällen muss	
	hier null stehen.	

Tabelle user_institution

Der *testuser* erhält im Beispiel das Recht den TestFB (ch110_institut - key_apnr=11) im Zeitraum vom 1.1.2001 bis zum 31.12.2001 einzusehen.

Die Masken, die einen Organigramm-Button enthalten, arbeiten mit den Prozeduren sp_user_orga bzw. sp_user_orga_child, die die Institutionen zusammensuchen und in eine temporäre Tabelle schreiben.

4.1.2 Gruppenverwaltung

Man braucht nicht jedem User einzeln Rechte für (Abfrage-)Masken geben, sondern kann dies auch über die Zugehörigkeit zu einer Gruppe erledigen.

4.1.2.1 Tabelle groupinfo

Die Gruppennamen werden in der Tabelle groupinfo festgelegt.

Tabelle groupinfo

Feld	Erläuterung	Beispiel
tid	Nummer der Gruppe	0
name	name der Gruppe	Dezernenten

4.1.2.2 Tabelle user_group_bez

Die Zugehörigkeit eines Users zu einer Gruppe wird in der Tabelle user_group_bez festgelegt.

Feld	Erläuterung	Beispiel
userinfo_id	Nummer des Users	1
	(entspricht dem Feld tid in der Tabelle userinfo)	
groupinfo_id	Nummer der Gruppe	1
	(entspricht dem Feld tid in der Tabelle groupinfo)	

Tabelle user_group_bez

Der testuser (Nr. 1) wird zur Gruppe 1 (Dezernenten) gerechnet.

Die Gruppenrechte werden in den Datenbank-Tabellen group_masken_bez und group_sachgeb_bez gespeichert.

4.1.2.3 Tabelle group_masken_bez

In der Tabelle group_masken_bez wird festgelegt, auf welche (Abfrage-)Masken die Gruppe Zugriff haben soll.

Tabelle group_masken_bez

Feld	Erläuterung	Beispiel
groupinfo_id	Nummer der Gruppe	0
	(entspricht dem Feld tid in der Tabelle groupinfo)	
maskeninfo_id	Nummer der Maske	10050
	(entspricht dem Feld tid in der Tabelle maskeninfo	
	s.u.)	

Das Beispiel besagt, dass die Gruppe der Dezernenten (und damit alle User, die dieser Gruppe zugeordnet sind), Zugriffsrechte für die (Abfrage-)Maske 10050 (Studierende allgemein) hat.

4.1.2.4 Tabelle group_sachgeb_bez

Wie bei einzelnen Usern kann man auch bei Gruppen Zugriffsrechte auf ganze Sachgebiete festlegen. Dies geschieht in der Tabelle group_sachgeb_bez.

Tabelle group_sachgeb_bez

Feld	Erläuterung	Beispiel
groupinfo_id	Nummer der Gruppe	0
	(entspricht dem Feld tid in der Tabelle groupinfo)	
sachgebiete_id	Nummer des Sachgebiets	1
	(entspricht dem Feld tid in der Tabelle sachgebie-	
	te)	

Das Beispiel zeigt die Freigabe des Sachgebiets 1 (Studierende/Prüfungen) für die Gruppe der Dezernenten. Die Reihenfolge der Berücksichtugng von Rechten ist wichtig. Werden innerhalb eines Sachgebiets Einzelrechte auf eine Abfrage vergeben, dann müssen bei jeder neuen Abfrage in diesem Sachgebiet wieder Einelrechte vergeben werden.

4.1.3 Zugriffsprotokollierung

Alle fehlgeschlagenen Anmeldeversuche an die Datenbank SuperX werden protokolliert (proto_fkt_id=2). Die Protokollierung dient nur der Überwachung der Autorisierung des Anmeldenden; darüber hinaus findet keinerlei Aufzeichnung von Benutzeraktivitäten o. ä. statt. Die maximale Anzahl der Anmeldeversuche ist für jeden Benutzer einstellbar (Tabelle ,userinfo', Feld ,max_versuch') und ist standardmäßig auf 5 Versuche eingestellt. Wird die maximale Anzahl überschritten, so wird die Benutzerkennung gesperrt. Der SuperX-Administrator könnte sich zusätzlich ein per Cronjob aufzurufendes Skript einrichten, dass bei Häufung von fehlgeschlagenen Anmeldungen ein Warnemail verschickt.

4.1.3.1 Die Tabelle protokoll

Attributname	ame Bedeutung	
protokoll_id	ID des Protokollsatzes	serial
proto_fkt_id	Nummer der Protokollfunktion (siehe Tabelle proto_funktion)	smallint
userinfo_id	Benutzer-ID (aus Tabelle userinfo)	integer
ip_adresse	IP-Adresse des Benutzers	char(16)
client_name Rechnername des Benutzers bzw. Fehlermeldung, wenn keine Netzverbindung möglich war (WINSOCKET -Fehler etc.)		char (255)
zeitpunkt Zeitpunkt des protokollierten Ereignisses		datetime year to second

Die Tabelle Protokoll enthält die Protokollsätze der Zugriffe auf SuperX.

4.1.3.2 Die Tabelle proto_funktion

Diese Tabelle enthält Funktionen, die protokolliert werden.

Feld	Bedeutung	Тур
proto_fkt_id	Nummer der	smallint
	Protokoll-	
	funktion	
proto_fkt	Name der	char(20)
	Protokoll-	
	funktion	

Inhalt der Tabelle proto_funktion:

proto_fkt	Bedeutung
LOGIN	Benutzer ist angemeldet
LOGIN_FAIL	falsches Paßwort eingegeben
LOGIN_LOCKED	Kennung gesperrt (Versuche> max_versuch)
LOGOUT	Benutzer hat sich abgemeldet
CH_PASSWD_OLD	Paßwort im Paßwortänderungsdialog eingegeben
CH_PASSWD_NEW	Neues Paßwort im Paßwortänderungsdialog ver- geben
CH_PASSWD_FAIL	Falsches Paßwort im Paßwortänderungsdialog eingegeben

4.2 Das Organigramm

Das Organigramm stellt eine integrierende Sicht für verschiedene Datenquellen zusammen und ist somit die Voraussetzung für eine integrierte Betrachtung. Alle Einrichtungen, Institutionen und Projekte sind im Organigramm hierarchisch angeordnet.

Die folgende Abbildung zeigt ein Beispiel für ein Organigramm:

Wie im Themenbaum können Sie durch einen hierarchischen Baum navigieren.

😤 Auswahl organisatorische Einheit
Stand: 7.1.2002 anderer Stand
GMU Duisburg GMU Duisburg Fakultät 1 - Gesellschaftswissenschaften Cehreinheit Erziehungswissenschaft Lehreinheit Geographie u. ihre Didaktik Lehreinheit Sozialwissenschaft Cehreinheit Sozialwissenschaft Fakultät 2 - Geisteswissenschaften Fakultät 3 - Wirtschaftswissenschaft Fakultät 4 - Naturwissenschaften Fakultät 5 - Ingenieurwissenschaften Lehreinheit Ostasienwissenschaft Lehreinheit sonstige Fächer
Suchen
OK Abbrechen Leeren

Das Organigramm wird vom Java-Applet aus der Tabelle organigramm unter Berücksichtigung der Userrechte aufgebaut.

4.2.1 Die Tabelle Organigramm

Das Kernmodul enthält bei Auslieferung das Organigramm der Universität Duisburg als Beispiel. Die Datensätze können nach der Installation als Vorlage dienen.

Feld	Erläuterung	Beispiel
tid	Interne Nummer	
key_apnr	Institutionennummer (z.B. im MBS)	"1"
parent	key_apnr der übergeordneten Institution	"0"
drucktext	Kurzer Text	Einr. Forsch. und Lehre
name	Name der Institution	Einrichtungen Forschung
		und Lehre
ebene	Hierarchieebene	1
lehre	Ist diese Institution relevant für Auswertungen	1
	im Bereich Lehre (1=ja, 0=nein); das Feld darf	
	nicht leer sein	
gueltig_seit	Beginn des Gültigkeitszeitraums	1.10.2001
gueltig_bis	Ende des Gültigkeitszeitraums	31.12.2999
orgstruktur	In diesem Feld wird festgelegt, dass es ich bei	
	einem Eintrag um eine besondere Institution	
	handelt.	
	Wenn es sich um einen Fachbereich handelt,	
	trägt man 20 ein, wenn es sich um eine Lehrein-	
	heit handelt 30.	

Bei der Gültigkeit bestehen einige Abhängigkeiten. Z.B. müssen bei Lehreinheiten als Anfang/Ende jeweils die Semestertermine genommen werden (also Lehreinheit alt gültig_bis 30.9.2001, Lehreinheit neu gültig_seit 1.10.2001).

Generell sollten Einrichtungen, die unbegrenzt gültig sind, im Feld gueltig_bis das Datum "31.12.2999" haben.

Erläuterung des parent-Felds:

Das Parent-Feld gibt die key_apnr der übergeordneten organisatorischen Einheit an. Das root-Element des Baums besitzt als parent einen null-Wert. Das Organigramm darf demenstprechend nur ein Element haben, dass keinen parent besitzt. Hier sollte grundsätzlich die Hochschulnummer eingetragen werden die auch in anderen HIS-Programmen verwendet wird. In Duisburg ist dies z.B. der Basiseintrag key_apnr = "70" (*GMU Duisburg*). Es ist praktisch der oberste "Knoten" im Baum. Dann werden alle Einträge in der Tabelle organigramm gesucht, die parent="70" haben – also direkte Kinder des Basiseintrags.

Hier findet sich u.a. *Einrichtungen Forschung und Lehre* (key_apnr="7", parent="70"). Dieser Eintrag hat wiederum u.a. folgende Kinder:

Fakultät 1 (key_apnr="1100", parent="7") *Fakultät 2* (key_apnr="1200", parent="7") *Faktultät 3* (key_apnr="1300", parent="7") usw.

Die weiteren Äste des Baums werden rekursiv abgefragt.

4.2.2 Füllen des Organigramms

Das Organigramm kann von Anwendern, die das "alte" SuperX bereits nutzen, relativ einfach importiert (S. 82) werden. Bei Neuinstallation von SuperX kann man (falls vorhanden) von der Institutionentabelle in MBS ausgehen (im SuperX-MBS-Modul lautet diese Tabelle mbs_inst). Der Imort würde lauten

alter table organigramm modify (tid serial); insert into organigramm select 0, inst_nr, uebinst_nr, lname1, lname2, "", 0, key_von, key_bis, orgstruktur from inst; alter table organigramm modify (tid integer, key_apnr char(10), parent char(10));

Danach könnte man diese recht "flache" Hierarchie nachbearbeiten bzw. nicht gewünschte Unterorganisationen streichen. Nachträglich müssen alle Organisationseinheiten, die oberhalb einer Lehreinheit liegen (z.B. Fakultäten), sowie die Lehreinheiten selbst, auf lehre = 1 stehen.

4.2.3 Die Prozedur sp_user_orga

Die Prozedur sp_user_orga sucht die Institutionen, die ein User sehen darf, und bereitet sie in einem temporären Organigramm auf (siehe Organigramm (S. 149)).

sp_user_orga(userid integer default -1, p_datum date default today, p_lehre smallint) Liefert für angegebenen Stand alle org. Einheiten zurück, die ein Benutzer sehen darf,

p_lehre=0	alle org. Einheiten, für die ein Benutzer Rechte hat
p_lehre=1	nur org. Einheiten aus dem Bereich Lehre, für die der Benutzer Rechte hat
p_lehre =2	Benutzer darf alle org. Einheiten im Bereiche Lehre sehen

4.2.4 Die Prozedur sp_user_orga_child

Die Prozedur sp_user_orga_child generiert die Tabelle tmp_ch110institut, die wiederum in der Abfrage aufgerufen wird und die alle Institutionen enthält, die ein User sehen darf und ausgewählt hat, d.h. den aktuellen "Ast" des Users im Organigramm.

sp_user_orga_child(userid integer default -1, p_datum date default today, p_lehre smallint, p_key_apnr integer, p_erlaubt smallint)

Liefert für einen angegebenen Stand alle Untereinheiten einer org. Einheit, die ein Benutzer einsehen darf (S. 144).

p lehre=1 nur die für den Bereich Lehre,	p_lehre=0	alle,]
1- ,	p_lehre=1	nur die für den Bereich Lehre,	

Aufruf steht im select_stmt (Tabelle maskeninfo), Änderungen dort

$p_{erlaubt} = 0$	Benutzer darf Einheit nicht komplett einsehen, nur ein oder mehrere un-
	tergeordnete Einheiten (z.B. nicht der gesamte FB6 – nur Geographie)
$p_{erlaubt} = 1$	Benutzer darf die gewählte Einheit mit allen Untereinheiten einsehen.
	p_erlaubt wird vom Applet gesetzt.

Beispielaufruf in einem SQL-Script:

execute procedure sp_user_orga_child (<<USERID>>, <<Organigramm-Stand>>, 0, /* <<Institution>>,
<<erlaubt>>)

Variablen in << >> werden vom Applet vor der Ausführung z.B. wie folgt ersetzt

User1, Fachbereich 6 (Interne Nummer = 6), Stand 1.5.2002, den der User komplett einsehen darf: execute procedure sp_user_orga_child (1, "1.5.2002", 0, "6", 1)

Achtung: Diese Prozedur ist in PostgreSQL bisher noch nicht unter Berücksichtugng der Userrechte in user_institution implementiert.

4.3 Die SuperX-Auswertungen

Im folgenden werden die grundlegenden Tabellen für die Verwaltung der SuperX-Auswertungen bzw. Abfragen erläutert.

4.3.1 Die Tabelle Maskeninfo

Basis einer SuperX-Abfrage ist ein Eintrag in der Tabelle maskeninfo. Eigene Masken müssen immer in einem definierten Nummernkreis liegen (z.B. >=10.000, <20000)) und Zehnerzahlen sein (z.B. 10050).

Tabelle maskeninfo

Feld	Erläuterung	Beispiel
tid	Interne Nummer	10050
name	Name der (Abfrage-)Maske	Studierende Allge-
		mein
select_stmt	SQL-Audrücke, die die Abfrage durchführen	SQL-Statement (s)
xil_proplist	beschreibt den Aufbau der Ergebnistabelle (s.u.)	
chart_xtitel	für graphische Darstellung der Ergebnisse	
	[derzeit nur im alten Client]	
chart_ytitel	für graphische Darstellung der Ergebnisse	
	[derzeit nur im alten Client]	
erlaeuterung	Erklärungstext zur Maske	
cleanup_stmt	SQL-Ausdruck nach select_stmt, z.B. um temporäre	drop table tmp_stud;
	Tabellen wieder zu löschen	
default_file	[wird derzeit nur vom alten Client genutzt]	
macro	[wird derzeit nur vom alten Client genutzt]	
breite	Breite der Maske in Pixel	
hoehe	Höhe der Maske in Pixel	
ampel	[wird derzeit nur vom alten Client genutzt]	
hilfe	Kennzeichen, ob Java-Hilfetext vorliegt (1=ja, 0=nein)	
hinweis	Erläuterungstext zur Ergebnistabelle, wird im Kopf	< <sql>> select</sql>
	angezeigt	erlaeuterung from
		koepfe_oder_faelle
		where apnr =
		"< <köpfe fäl-<="" oder="" td=""></köpfe>
		le>>"

4.3.1.1 SQL-Scripte

Die für die Suchanfrage einer Maske notwendigen SQL-Ausdrücke sind in der Tabelle maskeninfo im Blob-Feld select_stmt abgelegt.

Ein kleines Beispiel soll die Besonderheiten der SuperX-Suchanfragen erläutern.

Feld	Eintrag
name	Auslastung
select_stmt	select lehreinheit, export, auslastquote from auslas-
	tung where jahr = < <jahr>></jahr>
	/*and lehreinheit = < <lehreinheit>> */</lehreinheit>
	into temp tmp_auslastung with no log;
	select * from tmp_auslastung order by 1;
cleanup_stmt	drop table tmp_auslastung;

Es handelt sich hierbei um eine Maske zur Bestimmung der Auslastung einer Lehreinheit. Auf der Auswahlmaske gibt es 2 Felder: Jahr und Lehreinheit. Jahr ist ein obligatorisches, Lehreinheitein fakultatives Eingabefeld.

Für jedes Eingabefeld gibt es im select_stmt eine Variable << >>, die beim Auswerten der SQL Anweisungen durch den Inhalt des Feldes ersetzt wird. <<Jahr>> wird durch das vom User gewählte Jahr ersetzt. Handelt es sich wie bei <<Lehreinheit>> um ein fakultatives Eingabefeld, so kann der Feldinhalt leer sein. In diesem Fall wird zusätzlich der Teil der SQL-Anweisung auskommentiert (/*...*/), in dem die entsprechende Variable vorkommt (zwischen 2 Kommentarklammern (/*...*/) <u>muss genau eine</u> Feldvariable stehen!). Falls eine Lehreinheit vom User ausgewählt wird (z.B. 50000=Psychologie) wird die Zeile and lehreinheit =50000 mit ausgeführt. Wenn keine Lehreinheit ausgewählt wurde, bleibt sie unberücksichtigt und man erhält einen Gesamtwert über alle Lehreinheiten.

Der Variablen <<UserID>> kommt eine Sonderbedeutung zu: Sie ist im Applet als verborgenes Feld vorhanden. Für <<UserID>> wird die Nummer des Benutzers eingesetzt (vgl. sp_user_orga_child, s.o.).

Wichtig:Die **letzte** SQL-Anweisung muss ein select-Ausdruck sein, der das Ergebnis der Suchanfrage liefert. Das Ergebnis steht in unserem Beispiel in der temporären Tabelle tmp_auslastung. Diese Tabelle muss nach der Ausführung des select-Ausdrucks noch entfernt werden. Dafur gibt es das Feld cleanup_stmt, dessen Inhalt nach Ausführung von select_stmt ausgewertet wird.

4.3.1.2 Aufbau der Ergebnistabelle

Das Suchergebnis wird in einer Ergebnistabelle auf einer speziellen Suchergebnismaske dargestellt. Die Definition der Ergebnistabelle geschieht durch besondere Tags, die im Feld xil_proplist gespeichert werden.

Wichtig ist, dass für die Ergebnistabelle die Anzahl der selektierten Felder größer sein darf als die Anzahl der COLUMNS in XIL-List sein, aber nicht umgekehrt. Am einfachsten ist es, die Tabellendefinition einer bestehenden Maske zu kopieren und dann anzupassen.

Bestandteile des Kernmoduls: Die Referenz

Ein Beispiel für	^XIL List\
	sizable_columns horizontal_scrolling\
die Abfrage Auf-	white_space_color=COLOR_WHITE\
nahmekapazität im	arop_ang_aelete movable_columns llxeg_columns=1 \ min_heading_height=50 \
aktuellen Studien-	Column CID=0 heading_text="Lehreinheit / Studiengang" center_heading\ row_selectable heading_platform readonly\
ichm	width=35 text_size=50\
Jam.	Column CID=1 heading_text="Aufnahme-\\n kap. o. Ber.\\n Schwundquote" center_heading row_selectable col_selectable rightJust heading_platform readonly\ width=14\
	Column CID=2 heading_text="Aufnahme-\\n kap. m. Ber.\\n Schwundquote" center_heading row_selectable col_selectable rightJust heading_platform readonly\ width=14\
	Column CID=3 heading_text="Studierende\\n im 1. FS\\n im Studienj. " center_heading row_selectable col_selectable rightJust heading_platform readonly\ width=11\
	Column CID=4 heading_text="1. FS / \\n Aufn.kap o.\\n Schwund in %" center_heading\ row_selectable col_selectable rightJust heading_platform readonly\ width=14\
	Column CID=5 heading_text="1. FS / \\n Aufn.kap m.\\n Schwund in %" center_heading\ row_selectable col_selectable rightJust heading_platform readonly\ width=14\
	Column CID=6 heading_text="Studier.\\n in RSZ\\n im WS" center_heading\ row_selectable col_selectable rightJust heading_platform readonly\ width=8\
	Column CID=7 heading_text="Studier.\\n gesamt\\n im WS" center_heading\ row_selectable col_selectable rightJust heading_platform readonly\ width=8 \ @@@

Die Attribute zu den Felder werden wie folgt interpretiert:

Attribut	Erläuterung	
heading_text	Spaltenüberschrift	
center_heading	Zeilenausrichtung der Überschrift zentriert	
row_selectable	Zeile ist selektierbar (z.B. für Kopie in Zwischenablage)	
col_selectable	Spalte ist selektierbar	
rightjust	Zeilenausrichtung der Zellen rechtsbündig	
heading_platform	wird z.Zt. nicht genutzt	
readonly	Nicht editierbar	
height	Höhe der Zelle in Pixel (default ist 12)	
width	Breite der Zelle in Pixel (default ist Auto)	

4.3.1.3 Verbindung zur Tabelle felderinfo

Die einzelnen Felder auf einer Maske (z.B. Semester, Lehreinheit, Haushaltsjahr, Köpfe oder Fälle) sind in der Tabelle felderinfo abgelegt. Gibt es zu einer Maske der Nummer <TID> n Felder, so haben diese in der Tabelle felderinfo die Nummern:

 $<\!\!\text{TID}\!\!>\!\!,<\!\!\text{TID}\!\!>\!\!+1,\,...,<\!\!\text{TID}\!\!>\!\!+n\text{-}1$

So gehören z.B. zur Maske 10050 "Studierende allgemein", die folgenden Einträge in der Tabelle felderinfo

Bestandteile des Kernmoduls: Die Referenz

tid	name
10050	Köpfe oder Fälle?
10051	Semester
10052	Organisatorische Einheit
10053	Abschluss
10054	bis Fachsemester
10055	Hörerstatus
10056	Hochschulzugangsberechtigung
10057	Staatsangehörigkeit
10058	Aggregierung?

Gibt es mehr als 10 Felder auf einer Maske, so entfällt die entsprechende Zehnerzahl als Nummer für eine andere Maske, man sollte aber sicherheitshalber eine entsprechenden Eintrag in maskeninfo machen.

4.3.2 Tabelle Felderinfo

In der Tabelle felderinfo sind die einzelnen Auswahlfelder einer Maske abgelegt (s.o.).

Die Lage der Felder auf der Maske wird durch die Attribute x und y bestimmt. Der Ursprung des Koordinatensystems eines Fensters ist die obere linke Ecke, die y-Achse geht nach unten. Die Felder werden mit dem Offset (z.B. 60,60) platziert.

Tabelle felderinfo	
Erläuterung	Beispiel
Interne Nummer	
Name des Feldes auf der Maske	
Nummer des Feldes auf der Maske	
x-Position auf der Maske	
y-Position auf der Maske	
$\mathbf{y} = 0$: Gehe in die nächste Zeile	
y = -1: Bleibe in der aktuellen Zeile	
sonst: Nehme y als absoluten (auf die Maske bezogenen) y-Wert	
Die Feldnamen werden durch SuperX rechtsbündig angeordnet.	
Die dafür vorgesehene Breite wird mit dem Attribut buttonbreite	
definiert und in Pixel angegeben. Der Wert 100 reicht in den	
meisten Fällen aus.	
buttonbreite = 0 : Übernehme die entsprechenden Werte vom	
vorhergehenden Feld	
Die Breite des Eingabefeldes wird mit feldbreite bestimmt. Häu-	
fige Wer1e sind 70 für kurze Felder (z.B. Rechnungsjahr) oder	
200 für längere Felder (z.B. Institution).	
feldbreite = 0 : Ubernehme die entsprechenden Werte vom vor-	
hergehenden Feld	
Für die Anzahl der Zeilen des Eingabeteldes gibt es das Attribut	
zeilenanzahl. 1. Es konn höchtens ein Went im Einschafeld en	
Zenenanzani = 1 : Es kann nochtens ein wert im Eingabereid an-	
gegeben werden.	
gegeben werden Eingabefeld besitzt Scroll Leiste Damit das	
Fingabefeld nicht zu groß wird, reicht zeilenanzahl – 3 aus	
Achtung : Bei Mehrfachauswahlfeldern die als Werte integer-	
Schlüssel zurückgeben sollen, muss beim folgenden Attribut typ	
der Wert sol eingetragen werden. In SuperX werden die einzel-	
nen Werte mit "." getrennt. z.B.	
select * from groupinfo where tid in < <gruppe>></gruppe>	
wird aufgelöst nach:	
select * from groupinfo where tid in (1,4,6);	
Wenn der Typ nicht sql, sondern Character ist, geben Sie als Typ	
char ein; der Select wird dann aufgelöst nach:	
select * from groupinto where tid in ('1', '4', '6');	
ni SQL-Ausulucken wild zwischen numerischen und alphanume-	
mit Hochkommas versehen werden. Aus diesem Grund gibt as in	
SuperX verschiedene Typen von Feldern.	
tyn = integer. Es werden für dieses Feld in dem SOI - Ausdruck	
der Suchanfrage keine Hochkommas eingesetzt	
typ = char : Im select stmt werden automatisch Hochkommata	
	Tabelle felderinfo Erläuterung Interne Nummer Name des Feldes auf der Maske Nummer des Feldes auf der Maske y-Position auf der Maske y-Doistion auf der Maske y = 0: Gehe in die nächste Zeile y = -1: Bleibe in der aktuellen Zeile sonst: Nehme y als absoluten (auf die Maske bezogenen) y-Wert Die Feldnamen werden durch SuperX rechtsbündig angeordnet. Die dafür vorgesehene Breite wird mit dem Attribut buttonbreite definiert und in Pixel angegeben. Der Wert 100 reicht in den meisten Fällen aus. buttonbreite = 0 : Übernehme die entsprechenden Werte vom vorhergehenden Feld Die Breite des Eingabefeldes wird mit feldbreite bestimmt. Häufige Wer1e sind 70 für kurze Felder (z.B. Rechnungsjahr) oder 200 für längere Felder (z.B. Institution). feldbreite = 0 : Übernehme die entsprechenden Werte vom vorhergehenden Feld Für die Anzahl der Zeilen des Eingabefeldes gibt es das Attribut zeilenanzahl. zeilenanzahl. zeilenanzahl z 1: Es können mehrere Werte im Eingabefeld angegeben werden. gegeben werden. zeilgabefeld nicht zu groß wird, reicht zeilenanzahl = 3 aus. Achtung: Bei Mehrfachauswahlfeldern, die als Werte intesper-Schlüssel zurückgeben sollen, muss beim fol

	um den Variablenwert gesetzt. Bei Mehrfachauswahlfeldern gibt	
	es eine Sonderbehandlung (s.o.).	
	typ = sql : Das Ergebnis des Feldes liefert einen SOL-Ausdruck.	
	Im select stmt werden keine Hochkommas eingefügt.	
	typ = date : Es werden für dieses Feld in dem SOL-Ausdruck der	
	Suchanfrage date('') eingesetzt, bei PostgreSOL lautet die	
	Funktion date val('')	
	typ = decimal: Der eingegebene Wert (mit "," als Dezimaltrenn-	
	zeichen) wird im SOL-Ausdruck der Suchanfrage mit " " als De-	
	zimaltrennzeichen versehen	
laenge	zur Zeit nicht genutzt	
obligatorisch	Fingabe kann zwingend (obligatorisch –1) oder freiwillig sein	
obligatorisen	(obligatorisch = 0)	
ort	In den operativen Systemen wird off mit Schlüsselwerten (z B	
alt	In den operativen Systemen wird oft hitt Schusserwerten (Z.B.	
	in SuperV pur intern diese Nummern vervendet. Nach außen	
	in SuperA nur intern diese Nummern verwendet. Nach auben	
	die Felden neuen enterschieden menden scheine Neuenenen. Für	
	die Feider muss unterschieden werden, ob eine Nummemauspra-	
	gung existent. Daner glot es verschiedene Arten von Feldern:	
	art = 0: SuperX verwaltet nur den wert des Eingabereides. In	
	diese Felder kann man immer direkt werte eingeben. Die Eingabe	
	uber einen Dialog ist wahlweise möglich (siehe Abschnitt	
	Dialogsteuerung (S. 159)).	
	art = I : Eine Dialogbox wird geöffnet, die eine Liste mit Aus-	
	wahlmöglichkeiten anzeigt. Die Datenquelle ist eine Tabelle oder	
	ein SQL-Ausdruck, wobei die erste Spalte unsichtbar ist und den	
	Rückgabewert der Dialogbox liefert. In diese Felder kann man	
	nicht direkt, sondern nur über einen Dialog Werte eingeben.	
	art = 2 : Genau wie art = 1. Die Auswahlwerte des Dialogs kön-	
	nen jedoch nur durch Angabe einer Stored Procedure bestimmt	
	werden.	
	art = 3 : Feld zur Auswahl einer Datei mit Hilfe des plattformspe-	
	zifischen Dateiauswahldialogs. Eine direkte Eingabe ist nicht	
	möglich, das Feld ist nur einzeilig. Wird in Version 2.0 des	
	Applets noch nicht umgesetzt.	
	art = 4 : Es handelt sich um ein Feld zur Auswahl einer Institution	
	oder Person. Dazu wird ein spezieller Dialog geöffnet, der die	
	hierarchische Struktur der Hochschule widerspiegelt. Dabei wer-	
	den alle Institutionen angezeigt, für die der Benutzer Zugriffs-	
	rechte hat.	
	art = 5 : SuperX verwaltet nur den Wert des Eingabefeldes. Im	
	Unterschied zu art = 0 ist eine direkte Eingabe nicht möglich. Die	
	Art kann für die Gestaltung von Kommentarzeilen genutzt wer-	
	den.	
	art = 6 : wie art=4, aber es erscheinen nur alle Institutionen im	
	Bereich Lehre, für die der Benutzer Rechte hat (lehre=1 in der	
	Tabelle organigramm)	
	art = 7 : wie art=4, aber es erscheinen alle Institutionen im Be-	

	reich Lehre (lehre=1 in der Tabelle organigramm) ohne Rech-	
	teeinschränkung	
	art = 8: Das Feld dient nur als Label, es erscheint kein Eingabe-	
	feld	
	art = 12: Sicht, in Spalte relation muss ein SQL stehen, der die	
	tids der gewünschten Sichten aus der Sichtentabelle liefert	
relation	wird für Dialogsteuerung benötig, s.u.	
attribut	Feld in der DB-Tabelle, die man in relation angegeben hat	
	(s. Dialogsteuerung)	
default_wert	Vorgabewerte für den Feldinhalt (s.u.)	

4.3.2.1 Dialogsteuerung

Überwiegend erfolgt die Eingabe in die Felder dialoggesteuert, das heißt der Anwender kann aus einem Dialog mögliche Eingabewerte auswählen. In der Tabelle felderinfo kann zu jedem Feld definiert werden, wie die möglichen Auswahlwerte des Dialoges lauten. Um selbsterklärend zu sein, sollte der Anwender nur Klartext sehen. Die entsprechende Nummernausprägung verwendet SuperX nur intern. Die Bestimmung der Auswahlwerte eines Dialogs geschieht entweder durch Angabe einer DB-Tabelle und eines dazugehörigen Attributs, durch Angabe einer Stored Procedure oder eines SQL-Ausdrucks. Soll die Feldeingabe mit Hilfe eines Dialoges **nicht** möglich sein, so müssen die Einträge für relation und attribut leer bleiben.

4.3.2.1.1 Angabe einer DB- Tabelle

Soll z.B. auf einer Studierenden-Maske das Semester ausgewählt werden können, so steht in der Tabelle felderinfo:

Feld	Eintrag
name	Semester
relation	semester
attribut	eintrag

In der SuperX-Datenbank gibt es dazu die Tabelle semester.

Feld	Erläuterung	Beispiel
tid	Interne Nummernausprägung des Semesters	20011
eintrag	Semester als Klartext	SS 2001
sem_beginn	Datum des Semesterbeginns	01.04.2001
sem_ende	Datum des Semesterendes	30.09.2001

4.3.2.1.2 Angabe einer Stored Procedure

Statt einer DB- Tabelle kann auch eine Stored Procedure angegeben werden. Diese wird beim Maskenaufbau ausgeführt und liefert als Rückgabewerte die Auswahlwerte des Dialogs.

Damit SuperX zwischen einer Tabelle und einer Stored Procedure unterscheiden kann, muss der Name der Stored Procedure mit "sp_" beginnen. Bei Feldern mit art = 2 ist diese Konvention nicht erforderlich.

Häufiges Beispiel sind Felder, deren Eingaben aus der CIF (Central Information File) kommen. Hier die Tabelle felderinfo für ein Feld, mit welchem die Dienstart ausgewählt werden soll (in der CIF: hochschulallgemeiner Schlüssel der Nummer 107):

Feld	Eintrag
name	Dienstart
relation	sp_cif(0,107)
attribut	

4.3.2.1.3 Angabe eines SQL-Ausdrucks

Die Ergebnisse des angegebenen SQL-Ausdrucks sind die Auswahlwerte des Dialogs. Genau wie bei der Definition von Vorgabewerten für Felder muß der SQL-Ausdruck mit "<<SQL>>" beginnen. Beispiel:

Feld	Eintrag
name	Etage
relation	< <sql>> select distinct</sql>
	geschossnr, druck from baupc-
	geschoss order by 1;
attribut	

4.3.2.1.4 Hinweis für Dialogart 1 und 2

Für Felder bei denen eine Nummernausprägung intern verwendet wird (art = 1, 2) ist zu beachten, dass die Stored Procedure bzw. der SQL-Ausdruck 2 Werte (Nummernausprägung und Klartext) zurückliefern muss. Bei Angabe einer Tabelle müssen entweder 2 Attribute angegeben werden (attribut = A1,A2) oder man gibt nur das Attribut für den Klartext an. In diesem Fall geht SuperX davon, dass das Attribut für die Nummernausprägung "tid" heißt. Die Nummernausprägung muss zuerst angegeben werden. Man kann natürlich auch zugleich Nummernausprägung und Klartext in dem Dialog darstellen:

<<SQL>> select geschossnr, druck || "(" || geschossnr || ")" from baupc-geschoss;

4.3.2.2 Vorgabewerte für die Felder

Beim Öffnen einer Maske können dem Anwender Vorgabewerte angeboten werden, Dabei handelt es sich entweder um konstante Werte oder um Ergebnisse eines SQL-Ausdrucks:

4.3.2.2.1 Konstanten

Ein Feld für die Eingabe eines Rechnungsjahres soll den fest vorgegebenen Wert "2002" besitzen. In der Tabelle felderinfo steht:

Feld	Eintrag
name	Rechnungsjahr
default	2002

4.3.2.2.2 SQL-Ausdrücke

Viel flexibler ist die Definition des Vorgabewertes mit Hilfe eines SQL-Ausdrucks. Damit kann sowohl auf Werte aus der Datenbank als auch auf das aktuelle Datum zugegriffen werden.

Damit SuperX zwischen Konstanten und SQL-Ausdrücken unterscheiden kann, beginnen letztere mit "<<SQL>>" (Leerzeichen nicht vergessen !). Im folgenden Beispiel lautet der

SQL-Ausdruck für das Vorjahr:

Feld	Eintrag
name	Rechnungsjahr
default	< <sql>> select (year(today) -1) "" from xdummy;</sql>

Die Tabelle xdummy ist eine Tabelle mit einem Satz. Sie dient lediglich dazu, den o.g. SQL-Ausdruck syntaktisch korrekt zu machen.

Für Felder bei denen eine Nummernausprägung intern verwendet wird (art = 1, 2) ist zu beachten, dass der SQL-Ausdruck 2 Werte (Nummernausprägung und Klartext) zurückliefern muss.

Achtung: Vorgabewerte können nicht für Institutions-Felder (art = 4,6,7) angegeben werden. Für mehrzeilige Felder (zeilenanzahl> 1) können lediglich SQL-Ausdrücke angegeben werden.

4.3.3 Tabelle systeminfo

Die Tabelle systeminfo enthält für einzelne Sachgebiete/System, das Datum des letzten Datenupdates.

Tabelle systeminfo

Feld	Erläuterung	Beispiel
tid	interne Nummer	6
name	Name des Systemteils	Personal/Stellen
datum	Datum des letzten Datenupdates	14.1.2002

Dier Eintrag aus der Tabelle systeminfo wird über die Tabelle maske_system_bez mit der Tabelle maskeninfo verknüpft; so lassen sich die Abfragen den Systemen zuordnen.

4.3.4 Die Tabelle themenbaum

Nach der Anmeldung erhält der User eine Reihe von Auswertungen zur Auswahl in Form eines sog. "Themenbaums". Der Themenbaum wird dynamisch generiert aus der Tabelle themenbaum, die alle Auswertungen und deren hierarchischen Zusammenhang enthält, und den spezifischen Rechten, die der Benutzer hat. Die folgende Abbildung zeigt einen Ausschnitt aus einem Beispiel-Themenbaum.

👹 SuperX, SuperX	
Anmelden Passwort Clone Kopieren Drucken Info	5.2
Themenauswahl Maske Tabelle	
7.1.2002 anderer Stand	Erläuterung
 Abfragen ➡ → ☐ Evaluierung ➡ → ☐ Gebäude,Räume,Flächen ➡ → ☐ Gesamtüberblick ➡ → ☐ Haushalt ➡ → ☐ Kennzahlen ➡ → ☐ Kostenrechnung ➡ → ☐ Personal/Stellen ➡ → ☐ Studierende / Prüfungen ➡ → ☐ Absolventen (Zeitreihe über 6 Jahre) ➡ Absolventen nach Fachsemestern ➡ Absolventen nach Fachsemestern (mit Medianber.) 	Übersicht über Absolventen über einen Zeitraum von 5 Jahren
Abfrage suchen	
Oeffnen	
Anmeldung erfolgreich	

Das Java-Applet erzeugt aus der Tabelle themenbaum unter Berücksichtigung der Userrechte die graphische Oberfläche. Hier werden die Themen und Sachgebiete gesammelt und strukturiert. So kann aus der relativ einfachen Zuordnung von Themen (bzw. Masken) und Sachgebieten eine relativ komplexe Hierarchie gebildet werden.

Feld	Erläuterung	Beispiel
tid	Interne Nummer	2
name	Name der Maske bzw. des Sachgebiets	Absolventen
maskeninfo_id ID der Maske		10140
	(entspricht dem Feld tid in der Tabelle maskeninfo)	
	Bei Sachgebieten bleibt dieses Feld leer.	
parent	ID der übergeordneten Maske bzw. des Sachgebiets	1
gueltig_seit	Beginn des Gültigkeitszeitraums	1.1.2001
gueltig_bis	Ende des Gültigkeitszeitraums	1.10.3000
erlaeuterung	[Wird nicht benutzt]	

Tabelle Themenbaum

Der hierarchische Aufbau der Tabelle über das Feld parent entspricht dem der Tabelle organigramm. Die folgende Tabelle zeigt ein paar Beispieleinträge.

tid	name	maskeninfo_id	parent	gueltig_seit	gueltig_bis	erlaeuterung
5	Kennzahlen		1	01.01.1900	01.01.3000	
8	Studierende		2	01.01.1900	01.01.3000	
6	Gesamtüberblick		5	01.01.1900	01.01.3000	
4	Gebäude,Räume,Flächen		1	01.01.1900	01.01.3000	
3	Personal/Stellen		1	01.01.1900	01.01.3000	
2	Studierende / Prüfungen		1	01.01.1900	01.01.3000	
1	Abfragen			01.01.1900	01.01.3000	
88	Kostenrechnung		1	01.01.1900	01.01.3000	
81	Zeitreihen		2	01.01.1900	01.01.3000	
91	Evaluierung		1	01.01.1900	01.01.3000	
92	Prüfungen		2	01.01.1900	01.01.3000	
7	Haushalt		1	01.01.1900	01.01.3000	
47	Flächenarten für Institutionen	10010	4	01.01.1900	01.01.3000	
9	Studierende nach Hörerstatus	10040	8	01.01.1900	01.01.3000	
10	Studierende allgemein	10050	8	01.01.1900	01.01.3000	

Die Maske *Studierende nach Hörerstatus* hat als übergeordneten Knoten das Thema *Studierende*, und dies wiederum das Thema *Studierende/Prüfungen*.

4.3.5 Verkettung von Masken: Die Tabelle macro_masken_bez

Im SuperX können einzelne Masken zu einem Bericht kombiniert werden. Dazu wird ein Makro definiert, und die einzelnen Auswertungen werden dem Makro zugeordnet.

Feld	Erläuterung	Тур
maskeninfo_id1	Makro-Auswertung	integer
maskeninfo_id2	Dem Makro zugeordnete Auswertungen	integer
nummer	Ordnungsnummer der Zuordnung	

Die Makrofunktionalität ist in der Version 2.02 vom SuperX-Kernmodul wie folgt umgesetzt:

- 1. Sie erzeugen eine Maske mit allen relevanten Feldern. Diese Maske benötigt kein select_stmt und keine xil_proplist, sie dient gewissermaßen als "Hülle" für die eigentlichen Abfragen.
- 2. Dann ordnen Sie die vorhandenen Abfragen diese Maske in der obigen Tabelle zu. Mit dem Feld nummer legen Sie die Reihenfolge fest. Achten Sie darauf, daß die Feldnamen in der Makro-Maske mit denen in den Unter-Masken identisch sind.
- 3. Der Makromechanismus läuft im XML-Frontend automatisch ab. Die Stylesheet-Einstellungen der Makro-Maske überlagern die der etwa vorhandenen Einzel-Masken.

Ein Beispiel im COB-Modul ist das Makro zum Berichtsblatt MSWF NRW.

4.4 Einzelne Schlüsseltabellen

Für die Abfragen aus den verschiedenen Basissystemen gibt es ein paar regelmäßig wiederkehrende Anfragen, z.B. Aggregierung nach Quartalen und Halbjahren. Deshalb sind diese im Kernmodul angelegt.

4.4.1 Die Tabelle schluessel

Die Tabelle schluessel enthält ausschließlich Metadaten zum Betrieb von SuperX; sie enthält schluessel und Erläuterungstexte zu den einzelnen Funktionalitäten in SuperX, z.B. zu den Feldtypen auf einer Maske etc.

Feld	Erläuterung	Тур
id	Tupelidentifier	integer
variable	Art der Variable / des Schlüssels	char(50)
wert	Codierung oder SQL-Ausdruck	char(255)
beschreibung	Kurzer Erläuterungstext	char(255)
typ	Variablentyp / Schlüsseltyp	char(255)
erlaeuterung	Langer Erläuterungstext	char(255)

Die Tabelle besitzt folgende Struktur

Die folgende Tabelle zeigt die Metadaten zum Aufbau von Feldern einer Maske. Die "Werte" sind wiederum die Schlüssel, die in der Tabelle Felderinfo als Attibute gefüllt werden.

▦	🖩 schluessel : Tabelle					
	id	variable	wert	beschreibung typ	erlaeuterung	
►	1	Feldart	0	0-nur Text, direl		
	2	Feldart	1	1-Nummer+Tex	Eine Dialogbox wird geöffnet, die eine Liste mit Au	
	3	Feldart	2	2-wie 1, aber m		
	4	Feldart	3	3-Dateiauswahl	Feld zur Auswahl einer Datei mit Hilfe des plattforr	
	5	Feldart	4	4-Institutionsdia	Es handelt sich um ein Feld zur Auswahl einer Ins	
	6	Feldart	5	5-nur Text, nur		
	7	Feldart	6	6-Institutionsdia	wie art=4, aber es erscheinen nur alle Institutioner	
	8	Feldart	7	7-Institutionsdia	wie art=4, aber es erscheinen alle Institutionen im	
	9	Feldart	8	8-Erläuterungsfi		
	10	Feldtyp	integer	integer		
	11	Feldtyp	char	char		
	12	Feldtyp	sql	sql		
	14	obligatorisch	0	0-Nein		
	15	obligatorisch	1	1-Ja		
*						

4.4.2 Die Schlüsseltabelle aggregierung

Die Tabelle aggregierung wird in Abfragen verwendet, um nach bestimmten Markmalen zu zu summieren oder zusätzliche Kriterien einzufügen.

Die Abfrage Nutzungsprotokolle (intern) benutzt die Tabelle z.B., um auf Zeiträume (Halbjahre, Quartale) einzuschränken.

Die Tabelle besitzt folgende Struktur

Feld	Erläuterung	Тур
tid	Tupelidentifier	integer
ord	Sortiernummer	smallint
name	Beschreibung	char(30)
kategorie	Kategorie	char(30)
wert	numerischer Wert oder sql-Ausdruck	char(255)

4.4.3 Die Schlüsseltabellen cif und cifx

Die Schlüsseltabelle cif ist Bestandteil des Kernmoduls und enthält Schlüssel, die in verschiedenen operativen Systemen verwendet werden. Die Tabelle cifx ist eine analoge Schlüsseltabelle, die auch alphanumerische Ausprägungen enthält.

Die Tabellen sind das "Herzstück" des Data Warehouse, und möglichst alle Schlüssel sollen darin enthalten sein. Die Art des Schlüssels wird dirch den Wert "key" bestimmt, und prinzipiell ist es möglich, hochschulspezifische und allgemeine Schlüssel zu pflegen. So gibt es in der cifx z.B. den Schlüssel:

Schlüsselname	ch35_ang_abschluss		
Key	35		
Bedeutung	Hochschulspezifischer Schlüssel für die angestrebte Abschluss-		
	prüfung aus SOS		

Der jeweilige Wert für des Schlüssels steht im feld apnr, und die Kurz- und Langbeschreibungen stehen in den Felder kurz, druck, lang_1 usw.

Feld	Erläuterung	Тур
tid	Interne Nummer	serial
hs	Hochschul-Nr. (0=Hochschulübergreifend)	integer
key	Schlüsselgruppe	smallint
apnr	Schlüssel	integer
d_akt_von	Datum von	date
d_akt_bis	Datum bis	date
kurz	Kurzbeschreibung	char(10)
druck	Drucktext	varchar(30)
lang_1	Langbeschreibung 1	char(50)
lang_2	Langbeschreibung 2	char(50)
lang_3	Langbeschreibung 3	char(50)

Folgende Schlüsselgruppen sind z.B. in SuperX enthalten (Schlüssel in der cifx sind gesondert gekennzeichnet):

key	hs	Bedeutung	Schlüsseltabelle	Herkunft System	Herkunft Tabelle
12	0	Staat			
13	<>0	Familienstand	cifx		
27	<>0	Grund Beurlaubung			
30	0	Studienfach	cifx		
30	<>0	Studienfach	cifx		
35	0	HS-Abschluss	cifx		
35	<>0	HS-Abschluss	cifx		
36	0	Hochschule	cifx	SVA / COB	k_hochschule
39	<>0	Vertiefungsrichtung	cifx		
40	<>0	Studientyp			
62	<>0	Grund Exmatrikulation			
86	0	Dienstverhaeltnis			
90	<>0	Fakultaet fuer Wahlen			
95	0	Anrede / Titel	cifx		
106	0	Beurlaubungsgrund	cifx		
*107	0	Dienstart			
108	0	Amt-	cifx		
		/Dienstbezeichnung			
*109	0	BVL-Gruppe			
110	<>0	Besch.stelle			
*115	0	Haushaltsvermerk			
*116	0	Stellenart			
120	0	Bewährungs-, Zeitauf-			
		stieg			
212	<>0	Geldgeber	cifx		
258	0	Stellung in der HS			
*259	0	Stellenkategorie			
260	0	Grund Ausschei-	cifx		
		dung/Befristung			
261	0	Grund fuer das Beset-	cifx		
		zungsende			
268	0	Staatspruefung-			
		Abschluss			
*270	0	Besetzungsabweichung			
284	<>0	Kapitel	cifx		
286	0	Arbeitszeit			
*291	0	Personalkategorie			
305	0	Sperrkennzeichen			
500	0	Mittelschoepfung	cifx		
501	0	Staat	cifx		

Die vorgegebenen Schluessel sind hier mit * gekennzeichnet und dürfen nicht in die cif geladen werden. Die Tabelle wird im Zuge der Aufnahme von weiteren Modulen (z.B. SOS und SVA) weiter gefüllt.

4.4.4 Die Schlüsseltabelle trans_inst

Die organisatorischen Einheiten des Organigramms werden in SuperX gebündelt; mitunter stammen aus den zugrundeliegenden Basissystemen sehr viel detailliertere bzw. "tiefere" Institutionen, die in SuperX nicht unbedingt von Interesse sind, z.B. die Kostenstellen aus HISCOB. Diese Einrichtungen werden in der Tabelle trans_inst auf übergeordnete Insitutionen des Organigramms projiziert. Die Tabelle hat folgende Struktur:

Feld	Erläuterung	Тур
tid	Tupelidentifier	serial
inst_nr	Institutionen- bzw. Kostenstellen-Nummer	char(10)
ch110_institut	Übergeordnete bzw. zugeordnete Institution im Su-	char(10)
	perX-Organigramm	
name	Name der Institution	char(200)
gueltig_von	Gültigkeit der Projektion: Datum von	date
gueltig_bis	Gültigkeit der Projektion: Datum bis	date

Bei der Übernahme von Daten aus einem Basissystem wie COB wird dann der Datentabelle das Feld der Institution im SuperX-Organigramm hinzugefügt, in dem die Projektion abgebildet wird: Zum Beispiel die Kostenstelle 1200144 (Werkstatt) wird für den Zeitraum vom 1.1.2001 bis 1.4.2002 der SuperX-Institution 12001 (Lehreinheit Psychologie) zugeordnet:

inst_nr	1200144
ch110_institut	12001
name	Werkstatt
gueltig_von	1.1.2001
gueltig_bis	1.4.2002

Im Ladescript eines Basissystems wird diese Zuordnung dann übertragen:

```
Auszug aus dem Lade-
script für HISCOB update cob_busa
set chl10_institut = (select chl10_institut from trans_inst M
where M.inst_nr = cob_busa.instnr and
M.d_gueltig_von <= date ("01." || cob_busa.monat || "." ||
cob_busa.jahr) and
M.d_gueltig_bis >= date ("01." || cob_busa.monat || "." ||
cob_busa.jahr))
where instnr is not null;
```

Neben der eigentlichen Kostenstelle "Werkstatt" finden wir also bei obigem Beispiel im Feld ch110_institut die SuperX-Institution "Lehreinheit Psychologie". In allen Statistiken zur Lehreinheit, die auf cob_busa beruhen, wird also die Werkstatt stillschweigend hinzugezählt.

4.4.5 Weitere Schlüsseltabellen

4.4.5.1 Tabelle hochschulinfo

Die Tabelle hochschulinfo enthält die Nummer und den Namen der eigenen Hochschule.

Der Schlüssel der Hochschule wird in der Tabelle cif bzw. cifx benutzt, um hochschuleigene Schlüssel von allgmeinen Schlüsseln abzugrenzen.

Sie können die Hochschulinfo in einem DBFORM pflegen; gehen Sie dazu im XML-Frontend auf "Tabelle suchen"-> hochschulinfo. Sie erhalten ein DBFORM mit einem Datensatz:

Wählen Sie Ihre Hochschule aus. Wenn Ihre Hochschule in dem Klappmenü nicht enthalten ist, erfragen Sie die Hochschulnummer bei HIS und tragen sie sie manuell mit einem SQL-Tool in die Tabelle ein.

				X
				7
				SuperX
ochschulinfo	In diesem Fo ordnen	omular können Infosizu Ihrer Sie Ihre Hochschule im Feld	Hochschule speichern Hochschulnummer (Hi	. Bitte S1 zu.
Hochsch	ulnummer (HIS)	9000-sonst.deutsch.Hachschule	-	
	Name	Hochschule		
	Adresse	Musterstr. 65 47046 Musterstadt		
	Kapitel	06220		
	Superx server	sysrs380		
	r chschulinfo Hachsch	rehschulinfo In diesem Fo ordnen Hochschulnummer (HIS) Name Adresse Kapitel Superx_server	Antiperson and the service of the se	In diesem Formular können Infos zu Ihrer Hochschule speichern ordnen Sie Ihre Hochschule im Feld Hochschulnummer (HI Hochschulnummer (HIS) Hochschulnummer (HIS) 9000-sonst.deutsch.Hochschule Name Hochschule Adresse Musterstr. 65. 47046 Musterstadt Kapitel [06220 Superx_servert sysr380

4.5 Modulbeschreibung

SuperX-Modul

SuperX-	2
Version	
Modulversion	3.0
Sachgebiet	Administration
Systeminfo	Administration
Thema	Administration
Themenbaum-	Abfragen
Ast	

Übersicht

Tabellen

- aggre_bland
- aggregierung

- cif

- cifx

- db_forms
- db_tabellen
- db_version
- download_group_bez
- download_keyw_bez
- download_user_bez
- felderinfo
- fm_templates
- graphicformat
- group_masken_bez
- group_sachgeb_bez
- group_sichtarten
- group_sichten
- groupinfo
- hochschulinfo
- konstanten
- lehr_orga_bez
- lehreinheit_inst
- macro_feld_wert
- macro_masken_bez
- maske_system_bez
- masken_felder_bez
- maskeninfo
- nachricht
- organigramm
- organigramm_bak
- proto_funktion
- protokoll
- sachgeb_dbform_bez
- sachgeb_maske_bez
- sachgeb_sichtarten
- sachgeb_sichten
- sachgebiete
- schluessel
- sichten
- sx_captions
- sx_downloads
- sx_keywords
- sx_mask_style
- sx_repository
- sx_stylesheets

- systeminfo
- themenbaum
- themenbaum_bak
- trans_inst
- user_einstellungen
- user_group_bez
- user_institution
- user_masken_bez
- user_pw
- user_sachgeb_bez
- user_sichtarten
- user_sichten
- user_startfields
- user_startpage
- userinfo
- xdummy
- Joolap-Blueprints

Views

Funktionen / Prozeduren

Masken

Datenbankschema

Tabellen

aggre_bland

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
tid	INTEGER	4		false	
bland	CHAR	2		false	
text	CHAR	40		true	

Primärschlüssel c_aggre_bland

- tid

aggregierung

Modulübergreifende Tabelle für Aggregierungen in Ergebnisabfragen.

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
tid	INTEGER	4		true	
ord	SMALLINT	2		false	

name	CHAR	50	false	
kategorie	CHAR	30	false	
wert	CHAR	255	false	

Primärschlüssel co_aggregierung

- tid

cif

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
tid	INTEGER	4		true	
hs	INTEGER	4		true	
key	SMALLINT	2		true	
apnr	INTEGER	4		true	
d_akt_von	DATE	4		false	
d_akt_bis	DATE	4		false	
kurz	CHAR	10		false	
druck	CHAR	100		false	
lang_1	CHAR	100		false	
lang_2	CHAR	150		false	
lang_3	CHAR	250		false	

Primärschlüssel c_cif

- tid

- Index: unique **cif_ind** auf cif (hs, key, apnr);

cifx

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
tid	INTEGER	4		true	
hs	INTEGER	4		true	
key	SMALLINT	2		true	
apnr	CHAR	10		true	
d_akt_von	DATE	4		false	
d_akt_bis	DATE	4		false	
kurz	CHAR	10		false	
druck	CHAR	100		false	
lang_1	CHAR	100		false	
lang_2	CHAR	150		false	
lang_3	CHAR	250		false	

Primärschlüssel c_cifx_tid

- tid

- Index: unique **cifx_ind** auf cifx (hs, key, apnr);

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
tid	INTEGER	4		true	
name	CHAR	100		false	
tablename	CHAR	100		false	
form_path	CHAR	255		false	
caption	CHAR	200		false	
description	CHAR	255		false	

db_forms

Primärschlüssel pk_db_forms

- tid

db_tabellen

Ein Verzeichnis der Tabellennamen nach dem Schema der HIS GmbH.

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
db_einfuegema	CHAR	6			
rke					
tabellen_id	SERIAL			true	
name	CHAR	18			
protokollierung	SMALLINT				

Primärschlüssel c_db_tabellen

- tabellen_id

- Index: unique uxdb_tabellen_name auf db_tabellen (name);

- Index: xdb_tabellen_efm auf db_tabellen (db_einfuegemarke);

db_version

Ein Verzeichnis der Modul-Versionen nach dem Schema der HIS GmbH.

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
db_einfuegema	CHAR	6			
rke					
his_system	CHAR	20			
version	CHAR	8			
kern_system	CHAR	1			

Primärschlüssel c_db_version

- his_system

- version

- Index: xdb_version_efm auf db_version (db_einfuegemarke);

download_group_bez

Zuordnung Gruppen zu Download-Adressen

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
groupinfo_id	INTEGER	4		false	
download_id	INTEGER	4		false	

Primärschlüssel c_downloadgroup_b

- groupinfo_id

- download_id

download_keyw_bez

Zuordnung Stichwörter zu Download-Adressen

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
keyword_id	INTEGER	4		false	
download_id	INTEGER	4		false	

Primärschlüssel c_downloadthm_b

- keyword_id

- download_id

download_user_bez

Zuordnung Gruppen zu Download-Adressen

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
userinfo_id	INTEGER	4		false	
download_id	INTEGER	4		false	

Primärschlüssel c_downloaduser_b

- userinfo_id

- download_id

felderinfo

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
tid	INTEGER	4		true	
name	CHAR	50		true	
nummer	INTEGER	4		true	
X	INTEGER	4		false	
У	INTEGER	4		false	
buttonbreite	INTEGER	4		false	
feldbreite	INTEGER	4		false	
zeilenanzahl	INTEGER	4		false	
typ	CHAR	15		false	
laenge	INTEGER	4		false	

obligatorisch	INTEGER	4	false	
art	INTEGER	4	false	
relation	text		false	
attribut	CHAR	100	false	
defaultwert	CHAR	255	false	

Primärschlüssel c_felderinfo

- tid

fm_templates

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
id	char	200		true	
content	text			true	
description	char	200		false	
comment	char	200		false	
version	integer		1	false	

Primärschlüssel pk_fm_libs

- id

graphicformat

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
id	CHAR	100		true	
charttype	CHAR	255		true	
caption	CHAR	255		false	
width	INTEGER	4		false	
height	INTEGER	4		false	
captionx	CHAR	255		false	
captiony	CHAR	255		false	
linex	INTEGER	4		true	
liney	INTEGER	4		true	
showvalues	INTEGER	4		true	
moreattribs	TEXT			false	

Primärschlüssel c_graphicformat

- id

group_masken_bez

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
groupinfo_id	INTEGER	4		false	

maskeninfo_id	INTEGER	4	false	
	_	_		

 $Prim \"arschlüssel \ c_group_masken_bez$

- groupinfo_id

- maskeninfo_id

group_sachgeb_bez

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
groupinfo_id	INTEGER	4		false	
sachgebiete_id	INTEGER	4		false	

 $Prim \"arschlüssel \ c_group_sachgeb_bez$

- groupinfo_id

- sachgebiete_id

group_sichtarten

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
groupinfo_id	INTEGER			false	
sichtart	CHAR	255		false	

Primärschlüssel pk_group_sichtart

- groupinfo_id

- sichtart

group_sichten

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
groupinfo_id	INTEGER			false	
sicht_name_int	CHAR	200		false	
ern					

Primärschlüssel pk_group_sicht

- groupinfo_id

- sicht_name_intern

groupinfo

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
tid	INTEGER	4		false	
name	CHAR	30		false	

Primärschlüssel c_groupinfo

- tid

hochschulinfo

FeldnameFeldtypGrößeDefaultNot Null	Beschreibung
-------------------------------------	--------------

name	CHAR	100	true	
adresse	CHAR	200	false	
hs_nr	INTEGER	4	true	
kapitel	CHAR	5	true	
superx_server	CHAR	100	true	
kennung	CHAR	8	false	
passwd	CHAR	16	false	

Primärschlüssel c_hochschulinfo

- hs_nr

konstanten

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
tid	INTEGER	4		true	
apnr	INTEGER	4		false	
beschreibung	CHAR	25		false	

Primärschlüssel co_konstanten

- tid

lehr_orga_bez

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
lehr_id	CHAR	10		false	
orga_id	CHAR	10		false	
name	CHAR	50		false	
gueltig_seit	DATE	4		false	
gueltig_bis	DATE	4		false	

lehreinheit_inst

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
key_apnr	CHAR	10		false	
parent	CHAR	10		false	
drucktext	CHAR	50		false	
lehr	CHAR	10		false	
gueltig_seit	DATE	4		false	
gueltig_bis	DATE	4		false	

macro_feld_wert

Feldname	Feldtyp	Größe	Default		Ν	ot Null	Beschreibung
				5			

macro	INTEGER	4		true	
sortnr	INTEGER	4		true	
feldname	CHAR	255		true	
alias	CHAR	255		true	
value	CHAR	255		false	
value_caption	CHAR	255		false	
feldstand	CHAR	255		false	
feldsicht	CHAR	255		false	
active	INTEGER	4	1	true	

Primärschlüssel pk_macro_feld_wert

- macro

- sortnr

- active

macro_masken_bez

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
maskenin-	INTEGER			true	
fo_id1					
maskenin-	INTEGER			true	
fo_id2					
active	INTEGER		1	true	
sortnr	SMALLINT	2		false	
schleifenrelati-	CHAR	255		false	
on					
schleifenfeld-	CHAR	255		false	
name					
alias	CHAR	255		false	
schleifenfstand	CHAR	255		false	
schleifenfsicht	CHAR	255		false	
aktion	CHAR	255		false	

Primärschlüssel c2_macro_masken_bez

- maskeninfo_id1

- maskeninfo_id2

- active

- sortnr

maske_system_bez

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
maskeninfo_id	INTEGER	4		true	

systeminfo_id	INTEGER	4		true	

Primärschlüssel c_maske_system_bez

- maskeninfo_id

- systeminfo_id

masken_felder_bez

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
maskeninfo_id	INTEGER	4		false	
felderinfo_id	INTEGER	4		false	

Primärschlüssel c_masken_felder_bez

- felderinfo_id

- maskeninfo_id

maskeninfo

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
tid	INTEGER	4		true	
name	CHAR	255		true	
select_stmt	text			false	
xil_proplist	text			false	
chart_xtitel	CHAR	50		false	
chart_ytitel	CHAR	50		false	
erlaeuterung	CHAR	255		false	
cleanup_stmt	CHAR	240		false	
default_file	CHAR	200		false	
macro	SMALLINT	2		false	
breite	INTEGER	4		false	
hoehe	INTEGER	4		false	
ampel	SMALLINT	2		false	
hilfe	SMALLINT	2		false	
hinweis	text			false	

Primärschlüssel c_maskeninfo

- tid

nachricht

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
aktuell	CHAR	255		false	

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
tid	INTEGER	4		true	
key_apnr	CHAR	10		false	
parent	CHAR	10		false	
drucktext	CHAR	50		false	
name	CHAR	150		false	
ebene	INTEGER	4		false	
lehre	SMALLINT	2		true	
gueltig_seit	DATE	4		true	
gueltig_bis	DATE	4		true	
orgstruktur	INTEGER	4		false	

organigramm

- Index: unique **i_orga** auf organigramm (key_apnr, gueltig_seit, gueltig_bis);

- Index: ix130_3 auf organigramm (parent);

organigramm_bak

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
tid	INTEGER	4		true	
key_apnr	CHAR	10		false	
parent	CHAR	10		false	
drucktext	CHAR	50		false	
name	CHAR	150		false	
ebene	INTEGER	4		false	
lehre	SMALLINT	2		false	
gueltig_seit	DATE	4		true	
gueltig_bis	DATE	4		true	
orgstruktur	INTEGER	4		false	

- Index: unique i_orga_bak auf organigramm_bak (key_apnr, gueltig_seit, gueltig_bis);

proto_funktion

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
proto_fkt_id	SMALLINT	2		true	
proto_fkt	CHAR	20		true	

Primärschlüssel c_proto_fkt

- proto_fkt_id

protokoll

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
protokoll_id	SERIAL	4		true	
proto_fkt_id	SMALLINT	2	true		
--------------	----------	------	------	--	
userinfo_id	INTEGER	4	true		
ip_adresse	CHAR	16	true		
client_name	VARCHAR	255	true		
zeitpunkt	DATETIME	3594	true		

Primärschlüssel **c_protokoll**

- protokoll_id

sachgeb_dbform_bez

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
sachgebiete_id	INTEGER	4		false	
dbform_id	INTEGER	4		false	

Primärschlüssel c_sachgeb_dbform_bez

- dbform_id

- sachgebiete_id

sachgeb_maske_bez

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
sachgebiete_id	INTEGER	4		false	
maskeninfo_id	INTEGER	4		false	

 $Prim \"arschlüssel \ c_sachgeb_maske_bez$

- maskeninfo_id

- sachgebiete_id

sachgeb_sichtarten

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
sachgebiete_id	INTEGER			false	
sichtart	CHAR	255		false	

Primärschlüssel pk_sg_sichtart

- sachgebiete_id

- sichtart

sachgeb_sichten

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
sachgebiete_id	INTEGER			false	
sicht_name_int	CHAR	200		false	
ern					

Primärschlüssel pk_sg_sicht

- sachgebiete_id

- sicht_name_intern

sachgebiete

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
tid	INTEGER	4		false	
name	CHAR	50		false	

Primärschlüssel c_sachgebiete

- tid

schluessel

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
id	INTEGER	4		true	
variable	CHAR	50		false	
wert	CHAR	255		false	
beschreibung	CHAR	255		false	
typ	CHAR	255		false	
erlaeuterung	CHAR	255		false	

Primärschlüssel c_schluessel

- id

sichten

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
tid	SERIAL	4		true	
parent	CHAR	255		false	
systeminfoid	INTEGER		0	true	
art	CHAR	40		false	
type	INTEGER		10	true	
name_intern	CHAR	200		false	
name	CHAR	200		false	
beschreibung	CHAR	255		false	
sortnr	INTEGER		0	true	
quelle	CHAR	255		false	
alt_hier_id	CHAR	150		false	
treecfgtable	CHAR	255		false	
treecfgid	CHAR	150		false	
label	SMALLINT		0	true	
user_rechte	SMALLINT	2	1	true	

rechtequelle	CHAR	255		false	
sesamkey	CHAR	100		false	
standbutton	SMALLINT	2	0	true	
attribut1	CHAR	255		false	
attribut2	CHAR	255		false	
attribut3	INTEGER			false	
attribut4	INTEGER			false	
xmlmaxentries	INTEGER	4		false	
gueltig_seit	DATE	4	01.01.1900	true	
gueltig_bis	DATE	4	01.01.3000	true	
cacheapplet	CHAR	255		false	
cachexml	CHAR	255		false	
aktiv	SMALLINT		1	false	

Primärschlüssel c_sichten30tid

- tid

- Index: unique ix_name_intern auf sichten (name_intern);

sx_captions

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
tid	SERIAL	4		true	
id	CHAR	200		false	
table_name	CHAR	200		false	
field_name	CHAR	200		false	
record_no	INTEGER	4		false	
locale	CHAR	10		false	
contents_short	CHAR	255		false	
contents_long	text			false	
equalitystatus	SMALLINT	2		false	
sachgebiete_id	INTEGER	2		false	

Primärschlüssel c_sx_captions

- tid

- Index: ix_sx_captions auf sx_captions (field_name);

- Index: ix_sx_captions2 auf sx_captions (table_name);

- Index: ix_sx_captions3 auf sx_captions (record_no);

- Index: ix_sx_captions4 auf sx_captions (locale);

sx_downloads

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
tid	SERIAL	4		true	
name	CHAR	255		false	
ch110_institut	CHAR	10		false	
bezugsdatum	DATE	2		false	
importdatum	DATE	2		false	
kommentar	TEXT	20000		false	
kommentar_url	CHAR	255		false	
contenttype	CHAR	50		false	
datei	DISKBLOB	255		true	
gueltig_seit	DATE	1	01.01.1900	true	
gueltig_bis	DATE	1	01.01.3000	true	

Download-Adressen

Primärschlüssel pk_sx_downloads

- tid

- Index: ix_sx_down_inst auf sx_downloads (ch110_institut);

- Index: ix_sx_down_dbat auf sx_downloads (bezugsdatum, gueltig_seit, gueltig_bis);

sx_keywords

Stichwörter für Download-Adressen

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
tid	SERIAL	4		false	
name	CHAR	255		false	
parent	INTEGER	4		false	

Primärschlüssel c_sx_keywords

- tid

sx_mask_style

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
tid	INTEGER	4		true	
maskeninfo_id	INTEGER	4		false	
stylesheet_id	INTEGER	4		false	
ord	SMALLINT	2		false	

Primärschlüssel c_mask_style

- tid

sx_repository

Eine Sammlung hochschulspezifischer Konstanten und SQL-Ausdrücke

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
----------	---------	-------	---------	----------	--------------

tid	SERIAL	4		true	
id	CHAR	200		true	
content	TEXT	200		false	
caption	CHAR	200		false	
comment	TEXT	4		false	
version	SMALLINT	10		false	
art	CHAR	200		false	
art2	CHAR	200		false	
art3	CHAR	200		false	
sachgebiete_id	INTEGER	2		false	
sort1	INTEGER	2		false	
sort2	INTEGER	2		false	
sort3	INTEGER	2		false	
geaendert_am	DATE	4		false	
aktiv	CHAR	1		false	
gueltig_seit	DATE	1	01.01.1900	true	
gueltig_bis	DATE	1	01.01.3000	true	

Primärschlüssel pk_sx_repository

- tid

sx_stylesheets

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
tid	INTEGER	4		true	
filename	CHAR	255		false	
caption	CHAR	255		false	
description	CHAR	255		false	
relation	CHAR	10		false	
useragent	CHAR	255		false	
contenttype	CHAR	200		false	

Primärschlüssel c_stylesheets

- tid

systeminfo

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
tid	INTEGER	4		true	
name	CHAR	50		true	
datum	DATE	4		false	

Beschreibung SuperX-Kern-Modul V3.0 © Projektgruppe SuperX

Primärschlüssel c_systeminfo

- tid

themenbaum

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
tid	INTEGER	4		false	
name	CHAR	50		true	
maskeninfo_id	INTEGER	4		false	
parent	INTEGER	4		false	
gueltig_seit	DATE	4		false	
gueltig_bis	DATE	4		false	
erlaeuterung	CHAR	240		false	

Primärschlüssel c_themenbaum

- tid

themenbaum_bak

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
tid	INTEGER	4		false	
name	CHAR	50		true	
maskeninfo_id	INTEGER	4		false	
parent	INTEGER	4		false	
gueltig_seit	DATE	4		false	
gueltig_bis	DATE	4		false	
erlaeuterung	CHAR	240		false	

trans_inst

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
tid	INTEGER	4		true	
instnr	CHAR	10		false	
ch110_institut	CHAR	10		false	
name	CHAR	200		false	
gueltig_von	DATE	4		false	
gueltig_bis	DATE	4		false	

Primärschlüssel c_trans_inst

- tid

user_einstellungen

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
tid	INTEGER	4		true	

arbeitsver-	CHAR	255	false	
zeichnis				

Primärschlüssel c_user_einstellungen

- tid

user_group_bez

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
userinfo_id	INTEGER	4		false	
groupinfo_id	INTEGER	4		false	

Primärschlüssel c_user_group_bez

- groupinfo_id

- userinfo_id

user_institution

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
userid	INTEGER	4		true	
name	CHAR	50		false	
ch110_institut	CHAR	10		true	
hierarchie	SMALLINT	2		false	
gueltig_seit	DATE	4		true	
gueltig_bis	DATE	4		true	
lehre	SMALLINT	2		false	

Primärschlüssel c_user_institution

- ch110_institut

- gueltig_bis

- gueltig_seit

- userid

user_masken_bez

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
userinfo_id	INTEGER	4		false	
maskeninfo_id	INTEGER	4		false	

$Prim \ddot{a} rschlüssel \ c_user_masken_bez$

- maskeninfo_id

- userinfo_id

user_pw

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
userinfo_id	INTEGER			true	
pw_gueltig_bis	date			false	
last_login	DATETIME	3594		false	
old_sha1	CHAR	40		false	
zeitpunkt1	DATETIME	3594		false	
old_sha2	CHAR	40		false	
zeitpunkt2	DATETIME	3594		false	
old_sha3	CHAR	40		false	
zeitpunkt3	DATETIME	3594		false	
old_sha4	CHAR	40		false	
zeitpunkt4	DATETIME	3594		false	
old_sha5	CHAR	40		false	
zeitpunkt5	DATETIME	3594		false	
old_sha6	CHAR	40		false	
zeitpunkt6	DATETIME	3594		false	
old_sha7	CHAR	40		false	
zeitpunkt7	DATETIME	3594		false	

Primärschlüssel pk_user_pw

- userinfo_id

user_sachgeb_bez

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
userinfo_id	INTEGER	4		false	
sachgebiete_id	INTEGER	4		false	

Primärschlüssel c_user_sachgeb_bez

- sachgebiete_id

- userinfo_id

user_sichtarten

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
userinfo_id	INTEGER			true	
sichtart	CHAR	255		true	

Primärschlüssel pk_user_sichtart

- userinfo_id

- sichtart

user_sichten

Feldname Feldtyp Größe Default	Not Null	Beschreibung
--------------------------------	----------	--------------

userinfo_id	INTEGER		true	
sicht_name_int	CHAR	200	true	
ern				

Primärschlüssel pk_user_sicht

- userinfo_id

- sicht_name_intern

user_startfields

In angepassten SuperX-XML-Frontends kann man hier Maskenfelder für Startseiten hinterlegen, zu denen der User automatisch nach Login weitergeleitet wird.

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
tid	serial			true	
userid	INTEGER			true	
fieldname	CHAR	255		true	
value	CHAR	255			
active	integer			true	

Primärschlüssel c_user_startfields

- tid

- Index: **ix_ussf** auf user_startfields (userid);

user_startpage

In angepassten SuperX-XML-Frontends kann man hier Startseiten hinterlegen, zu denen der User automatisch nach Login weitergeleitet wird.

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
tid	serial			true	
userid	INTEGER			true	
starturl	CHAR	255			
encrypted	smallint		0		
notolderthan	INTEGER				
startmask	integer				
alturl	char	255			
stylesheet	CHAR	255			
stylesheet2	CHAR	255			
contenttype	CHAR	255			
extra1	CHAR	255			
extra2	CHAR	255			
active	integer			true	

Primärschlüssel c_user_startpage

- tid

- Index: **ix_ussp** auf user_startpage (userid);

userinfo

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
tid	INTEGER	4		true	
benutzer	CHAR	30		true	
kennwort	CHAR	16		false	
name	CHAR	50		false	
max_versuch	SMALLINT	2		true	
akt_versuch	SMALLINT	2		true	
email	CHAR	255		false	
administration	SMALLINT	2		true	
archiv_recht	SMALLINT	2		true	
passwd_sha	CHAR	40		false	
info	CHAR	255		false	

Primärschlüssel c_userinfo

- tid

xdummy

Feldname	Feldtyp	Größe	Default	Not Null	Beschreibung
c	CHAR	1		false	

Joolap-Blueprints

Views

View	hochschulen
Felder	
sql	DBMS-SpezifischDBMS-Spezifisch;

View	schluessel_feldart
Felder	
sql	DBMS-SpezifischDBMS-Spezifisch;

View	schluessel_feldtyp
Felder	
sql	SELECT id, variable, wert::char(15), beschreibung, typ, er-
	laeuterung FROM schluessel where variable='Feldtyp';

View	schluessel_oblig	
------	------------------	--

Felder	
sql	DBMS-SpezifischDBMS-Spezifisch;

View	schluessel_org
Felder	
sql	select wert as orgstruktur, beschreibung from schluessel
	where variable='orgstruktur';

View	user_groups
Felder	
sql	select U.tid, U.benutzer, G.tid, G.name from userinfo U,
	groupinfo G, user_group_bez B where U.tid=B.userinfo_id
	and G.tid=B.groupinfo_id and U.akt_versuch <=
	U.max_versuch;

View	user_inst_org
Felder	
sql	SELECT userid, O.name, U.ch110_institut, U.hierarchie,
	U.gueltig_seit, U.gueltig_bis, U.lehre, O.orgstruktur FROM
	user_institution U, organigramm O where
	O.key_apnr=U.ch110_institut ;

Funktionen / Prozeduren

Funktion / Pro-	current
zedur	
Parameter	
Script	Create or replace Function current() RETURNS ti-
Ĩ	<pre>mestamptz AS `select now() `LANGUAGE `SQL';</pre>
Rückgabe	- wert Typ: timestamptz
Kommentar	Das aktuelle Datum und die Uhrzeit als Timestamp

Funktion / Pro-	date
zedur	
Parameter	- datum text (Ein Datumswert als String (German locale))
Script	Create or replace Function date(text) RETURNS date AS `select to_date(\$1,''DD.MM.YYYY'')' LANGUAGE `SQL';
Rückgabe	- wert Typ: date
Kommentar	Das Datum als date

Funktion / Pro-	date_str
zedur	
Beschreibung	Liefert einen date als String (deutsche Locale)
Parameter	- wert date (Ein date)
Script	CREATE or replace FUNCTION date_str(date) RETURNS text AS `select day(\$1) `'.'' month(\$1) `'.'' year(\$1) ;' LANGUAGE `SQL';
Script	CREATE procedure date_str(datum date) RETURNING varchar(255); return '' datum; end procedure;
Rückgabe	- wert Typ: text
Kommentar	Der String

Funktion / Pro-	date_val
zedur	
Parameter	- datum text (Ein Datumswert als String (German locale))
Script	<pre>\$SUPERX_DIR/db/install/prozeduren/proc_date_val_</pre>
	pg.sql
Script	create procedure date_val(datstr char(100)) re-
	turning date; define p_date date; let
	p_date=date(datstr);
	re;
Rückgabe	- wert Typ: date
Kommentar	Das Datum als date

Funktion / Pro-	day
zedur	
Parameter	- datum date (Ein Datumswert)
Script	Create or replace Function day(date) RETURNS
1	smallint AS `select extract(day from
	<pre>\$1)::smallint `LANGUAGE `SQL';</pre>
Rückgabe	- wert Typ: smallint
Kommentar	Der Tag als Smallint

Funktion / Pro-	day
zedur	
Parameter	- datum timestamp (Ein Datumswert)
Script	Create or replace Function day(timestamptz)
1	RETURNS smallint AS `select extract(day from
	<pre>\$1)::smallint `LANGUAGE `SQL';</pre>
Rückgabe	- wert Typ: smallint
Kommentar	Der Tag als Smallint

Funktion / Pro- hour	Eventuation / Dross hours
----------------------	---------------------------

zedur	
Parameter	- datum time (Ein Datumswert)
Script	Create or replace Function hour(time) RETURNS smallint AS `select extract(hour from \$1)::smallint `LANGUAGE `SQL';
Rückgabe	- wert Typ: smallint
Kommentar	Die Stunde als Smallint

Funktion / Pro-	hour
zedur	
Parameter	- datum timestamp (Ein Datumswert)
Script	Create or replace Function hour(timestamptz)
1	RETURNS smallint AS `select extract(hour from
	<pre>\$1)::smallint `LANGUAGE `SQL';</pre>
Rückgabe	- wert Typ: smallint
Kommentar	Die Stunde als Smallint

Funktion / Pro-	minute
zedur	
Parameter	- datum time (Ein Datumswert)
Script	Create or replace Function minute(time) RETURNS
1	smallint AS `select extract(minute from
	<pre>\$1)::smallint `LANGUAGE `SQL';</pre>
Rückgabe	- wert Typ: smallint
Kommentar	Die Minute als Smallint

Funktion / Pro-	minute
zedur	
Parameter	- datum timestamp (Ein Datumswert)
Script	Create or replace Function minute(timestamptz)
±	RETURNS smallint AS `select extract(minute from
	<pre>\$1)::smallint `LANGUAGE `SQL';</pre>
Rückgabe	- wert Typ: smallint
Kommentar	Die Minute als Smallint

Funktion / Pro-	month
zedur	
Parameter	- datum date (Ein Datumswert)
Script	Create or replace Function month(date) RETURNS
1	smallint AS `select extract(month from
	<pre>\$1)::smallint `LANGUAGE `SQL';</pre>

Rückgabe	- wert Typ: smallint
Kommentar	Der Monat als Smallint

Funktion / Pro-	month
zedur	
Parameter	- datum timestamp (Ein Datumswert)
Script	Create or replace Function month(timestamptz)
Ĩ	RETURNS smallint AS `select extract(month from
	<pre>\$1)::smallint `LANGUAGE `SQL';</pre>
Rückgabe	- wert Typ: smallint
Kommentar	Der Monat als Smallint

Funktion / Pro-	now
zedur	
Parameter	
Script	create procedure now() returning datetime year
1	to second; return CURRENT; end procedure;
Rückgabe	- wert Typ: timestamp
Kommentar	Jetzt

Funktion / Pro-	nvl
zedur	
Beschreibung	Prüft, ob ein Float-Wert null ist; wenn ja, dann wird der
	zweite Parameter (ein Integer) zurückgegeben.
Parameter	- wert float (Ein Float-Wert)
	- altwert integer (Ein Integer)
Script	Create or replace Function nvl(float,integer) RETURNS float AS ' Declare wert float; altwert integer; BEGIN wert := \$1; altwert:=\$2; if wert=null then wert = altwert; end if; return altwert; END ' LANGUAGE 'plpgsql';
Rückgabe	- wert Typ: float
Kommentar	Der Float-Wert

Funktion / Pro-	nvl
zedur	
Beschreibung	Prüft, ob ein decimal-Wert null ist; wenn ja, dann wird der
	zweite Parameter (ein Integer) zurückgegeben.
Parameter	- wert float (Ein decimal-Wert)
	- altwert integer (Ein Integer)
Script	Create or replace Function nvl(decimal, integer)
	RETURNS decimal AS ' Declare wert decimal; alt-
	<pre>wert integer; BEGIN wert := \$1; altwert:=\$2; if</pre>
	wert=null then wert = altwert; end if; return

	altwert; END ` LANGUAGE `plpgsql';
Rückgabe	- wert Typ: float
Kommentar	Der decimal-Wert

Funktion / Pro-	nvl
zedur	
Beschreibung	Liefert einen String und einen Ersatz-String, wenn der erste
	Parameter null ist.
Parameter	- wert1 varchar (Ein String, der null sein kann)
	- wert2 varchar (Ein String, der zurückgegeben wird, wenn
	der erste null ist)
Script	CREATE or replace FUNCTION nvl(varchar,varchar) RETURNS text AS `select \$1::text where \$1 is not null union select \$2::text where \$1 is null;' LANGUAGE `sql';
Rückgabe	- wert Typ: text
Kommentar	Der String

Funktion / Pro-	nvl
zedur	
Beschreibung	Liefert einen String und einen Ersatz-String, wenn der erste
	Parameter null ist.
Parameter	- wert1 char (Ein String, der null sein kann)
	- wert2 char (Ein String, der zurückgegeben wird, wenn der
	erste null ist)
Script	CREATE or replace FUNCTION nvl(char, char)
	null union select \$2text where \$1 is null.
	LANGUAGE `sql';
Rückgabe	- wert Typ: text
Kommentar	Der String

Funktion / Pro-	nvl
zedur	
Beschreibung	Liefert einen String und einen Ersatz-String, wenn der erste
	Parameter null ist.
Parameter	- wert1 text (Ein String, der null sein kann)
	- wert2 text (Ein String, der zurückgegeben wird, wenn der
	erste null ist)
Script	CREATE or replace FUNCTION nvl(text,text)
	RETURNS text AS `select \$1::text where \$1 is not

	null union select \$2::text where \$1 is null;'
	LANGUAGE `sql';
Rückgabe	- wert Typ: text
Kommentar	Der String

Funktion / Pro-	second
zedur	
Parameter	- datum time (Ein Datumswert)
Script	Create or replace Function second(time) RETURNS
1	smallint AS `select extract(second from
	<pre>\$1)::smallint `LANGUAGE `SQL';</pre>
Rückgabe	- wert Typ: smallint
Kommentar	Die Sekunde als Smallint

Funktion / Pro-	second
zedur	
Parameter	- datum timestamp (Ein Datumswert)
Script	Create or replace Function second(timestamptz)
1	RETURNS smallint AS `select extract(second from
	<pre>\$1)::smallint `LANGUAGE `SQL';</pre>
Rückgabe	- wert Typ: smallint
Kommentar	Die Sekunde als Smallint

Funktion / Pro-	sp_bool_to_smallint
zedur	
Beschreibung	Konvertiert boolean nach smallint
Parameter	- wert boolean (Boolean-Wert)
Script	\$SUPERX_DIR/db/install/prozeduren/proc_bool_to_s mallint_pg.sql
Rückgabe	- wert Typ: smallint
Kommentar	1 oder 0

Funktion / Pro-	sp_filter_orga
zedur	
Beschreibung	Filtert das Organigramm von oben nach unten für das
	XML-Frontend.
Parameter	- wert integer (Maximale Anzahl Elemente)
Script	<pre>\$SUPERX_DIR/db/install/prozeduren/proc_sp_filter</pre>
	_orga_pg.sql
Script	<pre>\$SUPERX_DIR/db/install/prozeduren/proc_sp_filter</pre>
-	_orga_ids.sql
Rückgabe	- wert Typ: boolean
Kommentar	Keine Rückgabe, die Tabelle tmp_organigramm ist gefiltert.

Funktion / Pro-	sp_get_benutzer
zedur	
Beschreibung	Liefert einen String mit Benutzernamen
Parameter	- ausgangswert integer (userinfo_id)
Script	<pre>\$SUPERX_DIR/db/install/prozeduren/proc_get_benut</pre>
1	zer_pg.sql
Script	\$SUPERX_DIR/db/install/prozeduren/proc_get_benut
-	zer_ids.sql
Rückgabe	- wert Typ: text
Kommentar	Der Name doer die Kennung

Funktion / Pro-	sp_get_dbform_right
zedur	
Beschreibung	Prüft ob Anwender Recht für DBFORM hat
Parameter	- p_form char (Formularname (String))
	- p_userid integer (Die Userid)
Script	<pre>\$SUPERX_DIR/db/install/prozeduren/proc_get_dbfor</pre>
	m_right_ids.sql
Script	<pre>\$SUPERX_DIR/db/install/prozeduren/proc_get_dbfor</pre>
L L	m_right_pg.sql
Rückgabe	- wert Typ: smallint
Kommentar	1=ja,0=nein

Funktion / Pro-	sp_get_keylist_str
zedur	
Beschreibung	Liefert einen String aus einem Unterselect
Parameter	- ausgangswert varchar (Ein String, der null sein kann)
	- ausgangsfeld varchar (Ein String, der zurückgegeben wird,
	wenn der erste null ist)
	- schluesseltab varchar (Ein String, der zurückgegeben
	wird, wenn der erste null ist)
	- schluesselfeld varchar (Ein String, der zurückgegeben
	wird, wenn der erste null ist)
	- datentab varchar (Ein String, der zurückgegeben wird,
	wenn der erste null ist)
	- where varchar (Ein String, der zurückgegeben wird, wenn
	der erste null ist)
Script	<pre>\$SUPERX_DIR/db/install/prozeduren/proc_get_keyli</pre>
	st_str_pg.sql
Script	\$SUPERX_DIR/db/install/prozeduren/proc_get_keyli
	st_str_ids.sql

Rückgabe	- wert Typ: text
Kommentar	Der String

Funktion / Pro-	sp_get_subquery_str
zedur	
Beschreibung	Liefert einen String aus einem Unterselect
Parameter	- ausgangswert varchar (Ein String, der null sein kann)
	- ausgangsfeld varchar (Ein String, der zurückgegeben wird,
	wenn der erste null ist)
	- schluesseltab varchar (Ein String, der zurückgegeben
	wird, wenn der erste null ist)
	- schluesselfeld varchar (Ein String, der zurückgegeben
	wird, wenn der erste null ist)
	- datentab varchar (Ein String, der zurückgegeben wird,
	wenn der erste null ist)
	- where varchar (Ein String, der zurückgegeben wird, wenn
	der erste null ist)
Script	<pre>\$SUPERX_DIR/db/install/prozeduren/proc_get_subqu</pre>
	ery_str_pg.sql
Script	\$SUPERX_DIR/db/install/prozeduren/proc_get_subqu
	ery_str_ids.sql
Rückgabe	- wert Typ: text
Kommentar	Der String

Funktion / Pro-	sp_lehr_inst
zedur	
Beschreibung	erzeugt die Tabelle lehreinheit_inst, die zu jeder Institution,
	unterhalb einer Lehreinheit die zug. Lehreinheit abspeichert.
Parameter	
Script	\$SUPERX_DIR/db/install/prozeduren/proc_sp_lehr_i
1	nst_ids.sql
Script	<pre>\$SUPERX_DIR/db/install/prozeduren/proc_sp_lehr_i</pre>
-	nst_pg.sql
Rückgabe	
Kommentar	Erzeugt die Tabelle lehr_inst, ansonsten keine Rückgabe-
	werte

Funktion / Pro-	sp_table_exists
zedur	
Beschreibung	Prüft, ob Tabelle existiert
Parameter	- tabelle char(200) (Der Tabellenname)
Script	<pre>\$SUPERX_DIR/db/install/prozeduren/proc_sp_table_</pre>
	exists_pg.sql

Script	<pre>\$SUPERX_DIR/db/install/prozeduren/proc_sp_table_</pre>
	exists_ids.sql
Rückgabe	- wert Typ: boolean
Kommentar	Wahr / Falsch

Funktion / Pro-	sp_trunc_string
zedur	
Beschreibung	Liefert einen String mit vorgegebener Länge
Parameter	- ausgangswert text (Der String beliebiger Länge)
	- laenge integer (Die maximale Länge)
Script	<pre>\$SUPERX_DIR/db/install/prozeduren/proc_trunc_str</pre>
	ing_pg.sql
Script	<pre>\$SUPERX_DIR/db/install/prozeduren/proc_trunc_str</pre>
1	ing_ids.sql
Rückgabe	- wert Typ: text
Kommentar	Der gekürzte String

Funktion / Pro-	sp_update_sequence
zedur	
Beschreibung	Aktualisiert sequence-Felder in Postgres-Tabellen
Parameter	- tabname varchar(255) (Tabellenname)
Script	\$SUPERX_DIR/db/install/prozeduren/proc_sp_update
Rückgabe	- wert Typ: varchar(255)
Kommentar	Letzter SQL-Ausdruck

Funktion / Pro-	sp_user_lehr
zedur	
Beschreibung	Script für die Sichten-Fkt.
Parameter	
Script	\$SUPERX_DIR/db/install/prozeduren/proc_user_lehr _ids.sql
Rückgabe	
Kommentar	

Funktion / Pro-	sp_user_lehr_child
zedur	
Beschreibung	Script für die Sichten-Fkt.
Parameter	
Script	\$SUPERX_DIR/db/install/prozeduren/proc_user_lehr

	_child_ids.sql
Rückgabe	
Kommentar	

Funktion / Pro-	sp_user_orga
zedur	
Beschreibung	Erzeugt für eine übergebene Institution eine Tabelle mit
	Kind-Institutionen (tmp_organigramm) für einen User, die für
	den Themenbaum verwendet wird
Parameter	- userid integer (Die ID des Users)
	- datum date (Der geforderte Stand des Organigramms)
	- lehre integer (Filter auf Lehre? (1=ja,0= nein))
Script	<pre>\$SUPERX_DIR/db/install/prozeduren/proc_sp_user_o</pre>
_	rga_ids.sql
Script	<pre>\$SUPERX_DIR/db/install/prozeduren/proc_sp_user_o</pre>
-	rga_pg.sql
Rückgabe	
Kommentar	Erzeugt die Tabelle tmp_organigramm, ansonsten keine
	Rückgabewerte

Funktion / Pro-	sp_user_orga_child
zedur	
Beschreibung	Erzeugt für eine übergebene Insitution eine Tabelle mit
	Kind-Institutionen (tmp_ch110_institut) für einen User, die in
	Abfragen für Joins verwendet wird
Parameter	- userid integer (Die ID des Users)
	- datum date (Der geforderte Stand des Organigramms)
	- lehre integer (Filter auf Lehre? (1=ja,0= nein))
	- key_apnr char(10) (Übergebene Institution)
	- erlaubt smallint (erlaubt-Kennzeichen)
Script	<pre>\$SUPERX_DIR/db/install/prozeduren/proc_sp_user_o</pre>
	rga_child_ids.sql
Script	<pre>\$SUPERX_DIR/db/install/prozeduren/proc_sp_user_o</pre>
	rga_child_pg.sql
Rückgabe	
Kommentar	Erzeugt die Tabelle tmp_ch110_institut, ansonsten keine
	Rückgabewerte

Funktion / Pro-	sp_user_orga_child2
zedur	
Beschreibung	liefert Die Kinder einer ausgwählten Inst. aus dem Organi-
	gramm , die der user zum gegebenen Zeitpunkt sehen darf.

	Das Ergebnis steht in tmp_ch110_institut2 Diese Funktion
	gibt es nur wegen eines Bugs in Postgres: Temporäre Tabel-
	len werden in Funktionen nicht gelöscht! Die Funktion wird
	also nur gebraucht, wenn in einem Script die
	sp_user_orga_child 2x aufgerufen werden soll.
Parameter	- userid integer (Die ID des Users)
	- datum date (Der geforderte Stand des Organigramms)
	- lehre integer (Filter auf Lehre? (1=ja,0= nein))
	- key_apnr char(10) (Übergebene Institution)
	- erlaubt smallint (erlaubt-Kennzeichen)
Script	<pre>\$SUPERX_DIR/db/install/prozeduren/proc_sp_user_o rga_child2_pg.sql</pre>
Rückgabe	
Kommentar	Erzeugt die Tabelle tmp_ch110_institut2, ansonsten keine
	Rückgabewerte

Funktion / Pro-	sp_user_orga2
zedur	
Beschreibung	liefert einen Ausschnitt aus dem Organigramm , den der u-
	ser zum gegebenen Zeitpunkt sehen darf. Das Ergebnis steht
	in tmp_organigramm2 Dies Funktion gibt es nur wegen eines
	Bugs in Postgres: Temporäre Tabellen werden in Funktion-
	nen nicht gelöscht! Die Funktion wird also nur gebraucht,
	wenn in einem Script die sp_user_orga_child 2x aufgerufen
	werden soll.
Parameter	- userid integer (Die ID des Users)
	- datum date (Der geforderte Stand des Organigramms)
	- lehre integer (Filter auf Lehre? (1=ja,0= nein))
Script	\$SUPERX_DIR/db/install/prozeduren/proc_sp_user_o
	rga2_pg.sql
Rückgabe	
Kommentar	Erzeugt die Tabelle tmp_organigramm2, ansonsten keine
	Rückgabewerte

Funktion / Pro-	sp_user_themen
zedur	
Beschreibung	Erzeugt für einen übergebenen User den Themenbaum
Parameter	- userid integer (Die ID des Users)
	- datum date (Der geforderte Stand des Themenbaums)

Script	<pre>\$SUPERX_DIR/db/install/prozeduren/proc_sp_user_t</pre>
1	hemen_pg.sql
Script	<pre>\$SUPERX_DIR/db/install/prozeduren/proc_sp_user_t</pre>
1	hemen_ids.sql
Rückgabe	
Kommentar	Erzeugt die Tabelle tmp_themenbaum, ansonsten keine
	Rückgabewerte

Funktion / Pro-	string_not_null
zedur	
Beschreibung	Liefert einen Leerstring, wenn Parameter null ist (für
	Stringverkettungen)
Parameter	- wert char (Ein String)
Script	CREATE or replace FUNCTION string_not_null(char) RETURNS text AS `select \$1::text where \$1 is not null union select `''' where \$1 is null;' LANGUAGE `SQL';
Script	<pre>create procedure string_not_null(p_str char(255)) returning char(255); return nvl(p_str,''); end procedure;</pre>
Rückgabe	- wert Typ: char
Kommentar	Der String

Funktion / Pro-	string_not_null
zedur	
Beschreibung	Liefert einen Leerstring, wenn Parameter null ist (für
	Stringverkettungen)
Parameter	- wert text (Ein String)
Script	CREATE or replace FUNCTION string_not_null(text) RETURNS text AS `select \$1::text where \$1 is not null union select `''' where \$1 is null;' LANGUAGE `SQL';
Rückgabe	- wert Typ: text
Kommentar	Der String

Funktion / Pro-	string_not_null
zedur	
Beschreibung	Liefert einen Leerstring, wenn Parameter null ist (für
	Stringverkettungen)
Parameter	- wert varchar (Ein String)
Script	CREATE or replace FUNCTION string_not_null(varchar) RETURNS text AS `select \$1::text where \$1 is not null union select `''' where \$1 is null;' LANGUAGE `SQL';
Script	<pre>create procedure string_not_null(p_str var- char(255)) returning varchar(255); return</pre>

	<pre>nvl(p_str,''); end procedure;</pre>
Rückgabe	- wert Typ: varchar
Kommentar	Der String

Funktion / Pro-	string_not_null
zedur	
Beschreibung	Liefert einen Leerstring, wenn Parameter null ist (für
	Stringverkettungen)
Parameter	- wert integer (Eine Ganzzahl)
Script	CREATE or replace FUNCTION string_not_null(integer) RETURNS text AS `select \$1::text where \$1 is not null union select `''' where \$1 is null;' LANGUAGE `SQL';
Rückgabe	- wert Typ: text
Kommentar	Der String

Funktion / Pro-	suchen_ersetzen
zedur	
Parameter	- Text text (Ein String)
	- alt text (Alter String der ersetzt werden soll.)
	- neu text (Neuer String)
Script	Create or replace Function su- chen_ersetzen(text,text,text) returns text as ` Declare zs text; Declare k integer; Declare s1 text; Declare alt text; Declare neu text; BEGIN s1:=\$1; alt:=\$2; neu:=\$3; k := position(alt in s1); While k > 0 loop zs := substring(s1, 1, k - 1); s1 := zs neu substring(s1, k + Length(alt), Length(s1) - k - Length(alt) + 1); k := position(alt in s1); END LOOP; return s1; END; `LANGUAGE `plpgsql';
Rückgabe	- wert Typ: text
Kommentar	Der Zielstring

Funktion / Pro-	time_stamp
zedur	
Parameter	- datumzeit text (Ein Timestamp als String (German locale))
Script	Create or replace Function time_stamp(text) RETURNS timestamptz AS `select to_timestamp(\$1,''DD.MM.YYYY HH.MI.SS'')' LANGUAGE `SQL';
Rückgabe	- wert Typ: timestamptz
Kommentar	Der Timestamp

Funktion / Pro-	timestamp_str
zedur	
Beschreibung	Liefert einen timestamp als String (deutsche Locale)
Parameter	- wert timestamptz (Ein timestamp)
Script	CREATE or replace FUNCTION times- tamp_str(timestamptz) RETURNS text AS `select to_char(\$1,''DD.MM.YYYY HH24:MI:SS `');' LANGUAGE `SQL';
Rückgabe	- wert Typ: text
Kommentar	Der String

Funktion / Pro-	timestamp_to_date
zedur	
Beschreibung	Konvertiert von timestamp nach date.
Parameter	- wert timestamtz (Ein timestamtz-Wert)
Script	CREATE or replace FUNCTION times- tamp_to_date(timestamptz) RETURNS date AS `select to_date(day(\$1) ''.'' month(\$1) `'.'' year(\$1),''DD.MM.YYYY'')' LANGUAGE `SQL';
Rückgabe	- wert Typ: float
Kommentar	Der Float-Wert

Funktion / Pro-	today
zedur	
Parameter	
Script	Create or replace Function today() RETURNS date
1 I	AS `select current_date `LANGUAGE `SQL';
Rückgabe	- wert Typ: datum
Kommentar	Das aktuelle Datum als Timestamp

Funktion / Pro-	today
zedur	
Beschreibung	Today() -Funktion für IDS
Parameter	
Script	<pre>\$SUPERX_DIR/db/install/prozeduren/proc_today_ids .sql</pre>
Rückgabe	- wert Typ: date
Kommentar	Heutiges Datum

Funktion / Pro-	today
zedur	
Beschreibung	to_number -Funktion für IDS

Parameter	- ausgangswert char (Der String beliebiger Länge)
	- text char (Die maximale Länge)
Script	\$SUPERX_DIR/db/install/prozeduren/proc_to_number
-	_ids.sql
Rückgabe	- wert Typ: date
Kommentar	Zahl

Funktion / Pro-	unescape_semicolon
zedur	
Beschreibung	Ersetzt das Zeichen '^' durch ';'
Parameter	- Text text (Ein String)
Script	<pre>create function unescape_semicolon(sqltext char(32000)) returning char(32000); begin return replace(sqltext,"^",";"); end; end function;</pre>
Script	Create or replace Function unesca- pe_semicolon(text) RETURNS text AS `select trans- late(\$1,''^'','';'') ` LANGUAGE `sql';
Rückgabe	- wert Typ: text
Kommentar	Der Zielstring

Funktion / Pro-	val
zedur	
Beschreibung	Liefert einen String als Zahl (analog Basic-Funktion val).
	Hat gegenüber to_number den Vorteil, dass man die Länge
	des Strings nicht kennen muss. Dafür klappt es nur bei Inte-
	ger-Werten.
Parameter	- wert text (Eine String-Repräsentation einer Ganzzahl)
Script	CREATE or replace FUNCTION val(text) RETURNS in-
	teger AS `select
	<pre>to_number(\$1,repeat(''9'',length(\$1)))::integer</pre>
	` LANGUAGE `SQL';
Script	create procedure val(p_str char(255)) returning
1	<pre>integer; return p_str::integer; end procedure;</pre>
Rückgabe	- wert Typ: integer
Kommentar	Die Zahl

Funktion / Pro-	val
zedur	
Beschreibung	Liefert einen String als Zahl (analog Basic-Funktion val).
	Hat gegenüber to_number den Vorteil, dass man die Länge
	des Strings nicht kennen muss. Dafür klappt es nur bei Inte-

	ger-Werten.
Parameter	- wert char(255) (Eine String-Repräsentation einer Ganz-
	zahl)
Script	CREATE or replace FUNCTION val(char(255)) RETURNS integer AS `select to_number(\$1,repeat(''9'',length(\$1)))::integer ` LANGUAGE `SQL';
Rückgabe	- wert Typ: integer
Kommentar	Die Zahl

Funktion / Pro-	val
zedur	
Beschreibung	Liefert einen String als Zahl (analog Basic-Funktion val).
	Hat gegenüber to_number den Vorteil, dass man die Länge
	des Strings nicht kennen muss. Dafür klappt es nur bei Inte-
	ger-Werten.
Parameter	- wert varchar(255) (Eine String-Repräsentation einer
	Ganzzahl)
Script	CREATE or replace FUNCTION val(varchar(255))
	RETURNS integer AS `select
	<pre>to_number(\$1,repeat(''9'',length(\$1)))::integer</pre>
	` LANGUAGE `SQL';
Rückgabe	- wert Typ: integer
Kommentar	Die Zahl

Funktion / Pro-	week
zedur	
Parameter	- datum date (Ein Datumswert)
Script	Create or replace Function week(date) RETURNS
1	smallint AS `select extract(week from
	<pre>\$1)::smallint `LANGUAGE `SQL';</pre>
Rückgabe	- wert Typ: smallint
Kommentar	Die Woche als Smallint

Funktion / Pro-	week
zedur	
Parameter	- datum timestamp (Ein Datumswert)
Script	Create or replace Function week(timestamptz) RETURNS smallint AS `select extract(week from \$1)::smallint `LANGUAGE `SQL';
Rückgabe	- wert Typ: smallint
Kommentar	Die Woche als Smallint

Funktion / Pro-	weekday

zedur		
Parameter	- datum date (Ein Datumswert)	
Script	Create or replace Function weekday(date) RETURNS text AS `select to_char(\$1,''Dy'')::text `LANGUAGE `SQL';	
Rückgabe	- wert Typ: text	
Kommentar	Der Wochentag (Mon, Tue) als String (englisch)	

Funktion / Pro-	weekday
zedur	
Parameter	- datum timestamp (Ein Datumswert)
Script	Create or replace Function weekday(timestamptz)
1	RETURNS text AS `select weekday(date(\$1))::text
	`LANGUAGE `SQL';
Rückgabe	- wert Typ: text
Kommentar	Der Wochentag (Mon, Tue) als String

Funktion / Pro-	year
zedur	
Parameter	- datum date (Ein Datumswert)
Script	Create or replace Function year(date) RETURNS
1	smallint AS `select extract(year from
	<pre>\$1)::smallint `LANGUAGE `SQL';</pre>
Rückgabe	- wert Typ: smallint
Kommentar	Das Jahr als Smallint

Funktion / Pro-	year
zedur	
Parameter	- datum timestamp (Ein Datumswert)
Script	Create or replace Function year(timestamptz)
1	RETURNS smallint AS `select extract(year from
	<pre>\$1)::smallint `LANGUAGE `SQL';</pre>
Rückgabe	- wert Typ: smallint
Kommentar	Das Jahr als Smallint

Masken

- Maske Benutzerdaten ändern Nummer 71010
- Maske Benutzer im Detail Nummer 71040
- Maske Benutzer suchen Nummer 71030
- Maske Benutzer von SuperX Nummer 750
- Maske Beschriftung suchen Nummer 70170

- Maske Download abrufen Nummer 71180
- Maske Downloadstatistik Nummer 71200
- Maske Download suchen Nummer 71190
- Maske Feld ansehen Nummer 70580
- Maske Feld bearbeiten Nummer 70640
- Maske Feld kopieren Nummer 70630
- Maske Feld löschen Nummer 70620
- Maske Feld suchen Nummer 70570
- Maske Gruppe einrichten Nummer 71120
- Maske Gruppe löschen Nummer 71140
- Maske Gruppe suchen Nummer 71050
- Maske Institutionsrechte Nummer 950
- Maske Institution suchen Nummer 71080
- Maske Maske ansehen Nummer 70520
- Maske Maske bearbeiten Nummer 70700
- Maske Maske kopieren Nummer 70350
- Maske Maske löschen Nummer 70370
- Maske Maske suchen Nummer 70510
- Maske Passwort ändern Nummer 70060
- Maske Sicht suchen Nummer 70160
- Maske Stylesheet suchen Nummer 71000
- Maske Tabellenfeld bearbeiten (char) Nummer 70540
- Maske Tabellenfeld bearbeiten (int) Nummer 70560
- Maske Tabellenfeld bearbeiten (Text) Nummer 70550
- Maske Tabelle suchen Nummer 71100
- Maske Themenbaum-Eintrag suchen Nummer 71090
- Maske User einrichten Nummer 70970
- Maske User-Institutionen Nummer 70460
- Maske User löschen Nummer 70990

5 Hinweise für Entwickler/innen

SuperX enthält verschiedene Formen von Scripten: Das Laden und die Übernahme der Basisdaten sowie die Erzeugung der Hilfstabellen wird von Shell-Scripten erledigt, wie in der Installationsanleitung der jeweiligen Module dokumentiert. Die Abfragen sind in der Datenbank in der Tabelle maskeninfo sowie felderinfo; Änderungen sind im Howto (S. 119) dokumentiert. Die Erzeugung von Hilfedokumenten für die Abfragen ist im Abschnitt Javahelp (S. 205) beschrieben.

Das Applet und Servlet wurde in Java programmiert. Änderungen werden im Folgenden beschrieben.

5.1 Kompilieren der Java-Quellen

Das Java-Applet und das Servlet sind im Quellcode verfügbar. Für die Entwicklung nutzen wir das Build-Tool Ant, es können aber auch andere Entwicklungsumgebungen eingesetzt werden. Die SuperX-Quellen haben folgende Struktur:

de.superx.applet	Klassen des SuperX-Applets
de.superx.dbadmin	Klassen des SuperX-Admintools
de.superx.servlet	Klassen des SuperX-Servlet
de.superx.bin	Kommandozeilen-Klassen für den SuperX-Client
de.superx.util	Gemeinsam benutzte Dateien
de.memtext.*	Gemeinsam benutzte Utilities der Fa. memtext
images	Gemeinsam benutzte Grafiken
com.sun.help	Javahelp-Klassen
javax.help.	Ebenfalls Javahelp-Klassen

Auf dem Webserver wird im Verzeichnis

\$SUPERX_DIR/webserver/tomcat/webapps/superx/WEB-INF/lib

die Datei superx<<Versionsnr.>>.jar abgelegt. Diese enthält alle Klassen. Beim Zugriff über Tomcat ist die Datei automatisch im CLASSPATH, beim SuperX-Client via jdbc muss diese Datei manuell, z.B. über die Datei SQL_ENV (S. 21), im CLASSPATH sein (Umgebungsvariable JDBC_CLASSPATH).

Das Applet und das Admintool soll wegen WWW-Einsatz möglichst "leicht" sein, deshalb wird es separat kompiliert. Lediglich die Klasse de.superx.servlet.SuperX_el.class wird sowohl vom Applet als auch vom AdminTool benötigt. Das Archiv heißt jeweils superx.jar für das Applet, und SuperXDBAdmin.jar für das Admintool. Beide werden nach

\$SUPERX_DIR/webserver/tomcat/webapps/superx/applet
kopiert.

Beachten Sie beim Kompilieren, dass das Applet und das AdminTool bei vielen Browsern im Cache gehalten wird (selbst wenn der Browser immer nach aktuellen Versionen suchen soll). Sie sollten nach neuem Kompilieren sicherheitshalber immer den Cache löschen und den Browser einmal beenden. Alternativ können Sie das Applet auch lokal aus dem Browser starten (also nicht über http://), Sie müssen lediglich eine korrekt eingestellte superx.properties mit gültiger SxServerURL im gleichen Verzeichnis haben.

5.1.1 Kompilieren mit Bordmitteln des JDK

Aufgrund der Komplexität der eingebundenen Klassen ist ein Build mit normalen Bordmitteln des JDK zwar möglich, aber viel zu umständlich. Der Build läuft voreingestellt nur mit ANT (s.u.) und unter Linux.

5.1.2 Kompilieren mit dem Jakarta-Build-Tool ant

Wir empfehlen, Applet und Servlet mit dem im Kernmodul enthaltenen Werkzeug ant zu kompilieren, das bereits in dem SuperX-Kernmodul enthalten ist. Sämtliche Quellen lassen sich von der Konsole aus mit dem Sun JDK 1.4.x und ANT kompilieren. Folgende Pfade sind für Entwickler wichtig:

Javadoc-Dateien zum	\$SUPERX_DIR/doc/apidoc
gesamten SuperX-	
Paket	
Quellcode des	\$SUPERX_DIR/webserver/tomcat/webapps/superx/WEB-
SuperX-Applets	INF/src/de/superx/applet
Quellcode des	\$SUPERX_DIR/webserver/tomcat/webapps/superx/WEB-
SuperX-Servlets	INF/src/de/superx/servlet
Quellcode des	\$SUPERX_DIR/webserver/tomcat/webapps/superx/WEB-
SuperX-Admintools	INF/src/de/superx/dbadmin
Ant-Pfad zur	\$SUPERX_DIR/webserver/tomcat/webapps/superx/WEB-
build-xml	INF/src/build.xml
Ant-Shellscript für	\$SUPERX_DIR/webserver/tomcat/webapps/superx/WEB-
den Build	INF/src/build_it.x

Die Quellen lassen sichmit einem JDK-Compiler der Generation 1.4.x oder höher kompilieren. Zum Kompilieren des Servlets muss die Bibliothek für Servlets im Classpath enthalten sein; dies ist bei der normalen SuperX-Distribution der Fall.

Bei einer eingerichteten Umgebung für SuperX brauchen Sie die build.xml nicht anpassen. Bei benutzerspezifischen Einstellungen passen Sie die Einträge zum CLASSPATH, zur SuperX-Version und zu SUPERX_DIR an. Für die Versionierung wird der Filter-Mechanismus in ant genutzt, d.h. jedesVorkommen des Strings "@version@" wird durch den aktuellen Wert ersetzt, der in der ant-Property VERSION gesetzt ist.

Zur Nutzung von ant wechseln Sie in der Konsole in das Verzeichnis \$SUPERX_DIR/webserver/tomcat/webapps/superx/WEB-INF/src und rufen Ant wie folgt auf:

Kompilieren der Klassen des Applets Kompilieren und Erzeugen des Applets Dokumentieren des Applets Kompilieren des gesamten Pakets Erzeugen der superx<<Versionsnr>>.jar Dokumentieren des Servlets Kompilieren der Klassen des Admin-Tools Kompilieren und Erzeugen des Admin-Tools Dokumentieren des Admin-Tools build_it.x compileApplet build_it.x distApplet build_it.x docApplet build_it.x compileServer build_it.x distServer build_it.x docServlet build_it.x compileAdmin build_it.x distAdmin build_it.x docAdmin

Bei der Distribution des Applets ist im ant-Script folgende Nachbearbeitung vorgesehen: zunächst werden nicht benötigte Klassen aus der superx.jar entfernt, um das Applet möglichst klein zu halten. Das OpenSource-Tool obfuscator wird aufgerufen und die resultierende superx.jar wird an die richtige Stelle kopiert (\$SUPERX_DIR/webserver/tomcat/webapps/superx/applet). Danach ist eine Signierung des Applets vorgesehen. Die Syntax ist in dem Kommentar des targets distApplet in der build.xml beschrieben:

Geben Sie auf der Kommandozeile ein:

Befehlsfolge zum Sig-
nieren des Appletskeytool -genkey -alias superx_applet -keyalg RSA
keytool -selfcert -alias superx_applet -validity 365

Als Passwort wählen Sie das, das in der build.xml vorgesehen ist. Der Wert hinter Validitiy beschreibt den Gültigkeitszeitraum des Zertifikats (in Tagen).

Wenn Sie das Zertifikat erneuern wollen, müssen Sie es zunächst löschen mit keytool -delete -alias superx_applet

Das gleiche Vorgehen gilt für das Admin-Tool.

5.1.3 Entwicklung mit Jedit

Als Entwicklungsumgebung empfehlen wir Eclipse von IBM oder den plattformübergreifend verfügbaren OpenSource-Editor jedit (www.jedit.org). Er unterstützt via Pugins die Java-Entwicklung. Für SuperX benötigen Sie die folgenden Plugins:

Plugins für Jedit 📡 Console-Plugin

- JBrowse
- JCompiler
- AntFarm
- XML
- XSLT

Im Clientpaket sind diese Plugins bereits enthalten.

Sie starten das Plugin AntFarm, und geben als Build-File die Datei

\$SUPERX_DIR/webserver/tomcat/webapps/superx/WEB-INF/src/build.xml an. Die Targets werden dann übersichtlich angezeigt und können sofort ausgeführt werden.

5.2 Erzeugung der SuperX-Hilfe im Javahelp-Format

Die SuperX-Hilfe besteht aus einem Archiv im Javahelp-Format. Die Hilfetexte sind in den Modulen erzeugt und können problemlos integriert werden. Falls Sie eigene Hilfetexte einbinden wollen, müssen Sie wie folgt vorgehen:

- 1. Erzeugen Sie html-Seiten mit der Hilfe (html 3.2)
- 2. Binden Sie die Dateien in die Mapping-Datei ein

(\$SUPERX_DIR/webserver/tomcat/webapps/superx/applet/javahelp/map.jhm)

3. Falls die Hilfeseiten kontextabhängig abrufbar sein sollen, müssen die Titel der Mapping-Einträge folgenden Konventionen folgen:

- Allgemeine Beschreibungen der Abfragen lauten A<<TID>>.htm
- Beschreibungen der Masken lauten M<<TID>>.htm
- Beschreibungen der Ergebnistabellen lauten T<<TID>>.htm

Am Anfang ist es hilfreich, die vorhandenen Hilfetexte als Vorlage zu benutzen.

Die Javahilfe kann auch komfortabler mit dem Memtext-Autorensystem aus einer Word-Datei erzeugt werden. Details dazu siehe http://studio.memtext.de.

5.3 Versionshistorie

3.0 final (05/2007)

Entwickler/innen | Meikel Bisping, Daniel Quathamer, Christoph Litz

- Viele neue Sicherheitsfeatures (Passwort-Policy etc.)
- Verbessertes XML-Frontend: Excel-Export, Baummenüs etc.
- Verbesserte Administrationsmasken (DBFORMS) zur Userverwaltung etc.

3.0 beta (04/2005)

Entwickler/innen | Meikel Bisping, Daniel Quathamer

- Neue Stored Procedures für Postgres- Organigramm-Auswertung
- Abbildung alternativer Hierarchien und Anbindung an Userverwaltung
- Mandantenfähigkeit
- Einsatz von Freemarker als Template Engine für Masken-Scripte (und damit Java-Unterstützung der Scripte), erste Libraries für Postgres und Informix-unabhängigen Code
- Glossare und Felderläuterungen abrufbar
- Einsatz von dbforms 2.5 als Formular-Engine; erste Administrationsformulare
- Komplettes Refactoring des XML-Frontend inkl. Cacheing, Organigramm-Darstellung, XSL-Mechanismen

2.1 (04/2004)

Entwickler/innen | Meikel Bisping, Daniel Quathamer, Marlies Winterstein

- SSL/Apache-Anbindung dokumentiert, Musterdateien für mod_jk fertig
- XML-Frontend liefert html,- XML und text-Export sowie rtf und pdf; anderer Authentifizierungsmechanismus (Cookie)
- Stylesheet-Verwaltung verbessert, erste Beispielstylesheets für lynx.
- Unter Postgres im XML-Frontend sind Masken zur Administration von Masken und Usern fertiggestellt
- Bugfixes im XML-Frontend (Pflichtfelder abfangen, Feldinhalte einlesen)
- Admin-Tool 0.93 mit vielen Bugfixes (insbes. für Postgres)
- Applet: Implementation von Organigramm-"Sichten" (noch nicht dokumentiert); viele kleine Verbesserungen, z.B. beim Anmeldedialog, Tabellendarstellung, Bedienung.
- Tomcat4-Anbindung
- Datenbankschema in XML auf der Basis von Apache Torque-DTD und HIS-DTD (ergänzt um eigene Tags für ETL-Prozesse). Die Datenbank selbst wurde um die Tabellen db_version und db_tabellen ergänzt, um HIS-konform zu arbeiten. Aus den XML-Dateien werden die Modulscripte und Dokumentationen erzeugt.

- Neue Stored Procedures für Anmeldedialog (sp_user_themen). Makros werden im Applet jetzt ausgeblendet.
- Neues Installationsscript für Datenbank; diverse Anleitungen für RedHat,- SuSE Linux und Cygwin.
- Installationsscripte für Modulinstallation, Aktualisierung, Deinstallation (alpha); neue Shellscripte zum ETL-Prozeß
- jdbc-Client für Kommandozeile fertiggestellt (DOS und UNIX)
- Neue Kommandozeilen-Scripte unter DOS und Unix:
 - Maskenverwaltung
 - Tabellenextraktion / Upload
 - Datenbankschemata von Tabellen
 - XML-Transformation
 - Konvertierung von Rohdaten-Dateien nach der Maßgabe von Import/Exportspezifikationen (S. 97)
- Postgres 7.3 oder 7.4 wird unterstützt
- Verbessertes Access-Frontend: Formulare für alle relevanten Tabellen des Kernmoduls

2.01 (06/2003)

Entwickler | Meikel Bisping, , Marlies Winterstein, Daniel Quathamer

- Integration der Javahilfe ins Applet
- Signierung des Applets -> Keine Client-Installation außer JRE mehr notwendig
- Aufbau der Package de.superx.*
- Update auf JRE 1.4 in html-Aufrufseiten
- Java-Installationswebsite für versch. Browser verbessert (JSP-Seite mit Anpassung für Netscape 6.x /7.x, Mozilla 1.3.x, IE 5.x,6.x)

2.0 (03/2002)

Entwickler | Marlies Winterstein, Meikel Bisping, Daniel Quathamer

- Einbettung von kontextabhängigen Hilfeseiten mit Javahelp
- Stabilität und Performance im Netzbetrieb durch Connection Pooling
- Entwurf eines Werkzeugs zur Administration von Organigramm und Userrechten ("SuperX-Admintool")
- Fertigstellung eines Prototypen zur Administration via MS Access 2000
- Beliebig "tief" verschachtelbares und zeitabhängiges Organigramm
- Modularisierung von SuperX
- Baumstruktur im Organigramm eingebaut
- Baumstruktur im Themenbaum
- Stored Procedures für Abbildung der hierarchisches Struktur des Organigramms
- Modularisierung von SuperX vollzogen
- Applet-Servlet-Struktur
- XML-Frontend mit Makrofunktion
- -Möglichkeiten sicherer Verbindungen Servlet-Applet
- Fertigstellung eines Prototypen des Kernmoduls auf der Basis von PostgreSQL 7.2

1.0 (04/2001)

Entwickler | Reiner Behr (Uni Karlsruhe)

- Portierung des Win32-Client nach Java
- Datenbankzugriff über jdbc
- Ergebnistabellen optimiert (Sortierung / Löschung von Spalten, Druckfunktion)

ⁱⁱBug: 158974 RPM INSTALLATION USING RELOCATE OPTION FAILS ON REDHAT LINUX V8 {NBS}

ⁱⁱⁱ Bezüglich der Postgres Installation ist Herrn Wehling von der Uni Köln aufgefallen, daß die Postgres 7.2 Versionen unter dem neuen Redhat 9 nicht kompilierbar sind.

Wenn man in

/install/postgres-7.2.4/src/backend/commands/copy.c

oben folgende Zeile einbaut, geht es:

#include <errno.h>

^{iv} Unter Suse Linux 8.x, 9.x und RedHat 8.x und Fedora ist der Postgres-Datenbankserver für internationale Umgebungen installiert und daher problemlos lauffähig. Siehe allerdings den Hinweis zur Sysconfig unter SuSE.

^v In den 7.x-Postgres-Versionen ist dies offensichtlich ein Bug: Wenn ein User Datenbanken und Benutzer anlegen darf, dann wird er von Postgres als "Superuser" klassifiziert und darf deshalb auch Scriptsprachen installieren. In Postgres 8.x wurde dies korrigiert.

 $^{\rm vi}$ Unter Win 98 / Me befindet sich diese Datei im Verzeichnis c:\windows

vii http://support.microsoft.com/default.aspx?scid=kb;DE;304548

viii Ändern Sie mit regedit folgenden Schlüssel:

$HKEY_LOCAL_MACHINE \verb|SOFTWARE\|Microsoft\] tet\|VERSION\| Engines\|ODBC$

Setzen Sie dort den (dezimalen) Wert ConnectionTimeout (z.B. 600) auf 0

Siehe http://www.ms-office-forum.net/forum/archivethread-111477.html oder

http://www.ms-office-forum.net/forum/showthread.php?s=&postid=438543#post438543

^{ix} Bitte beachten Sie, daß die mitgelieferte web.xml nur mit Tomcat 4 lauffähig ist. Ein Muster für eine Tomcat-3-taugliche web.xml steht im Verzeichnis \$SUPERX_DIR/webserver/tomcat/webapps/superx/WEB-INF/web_tomcat3.xml.sam.Kopie-ren Sie diese vor dem Start von Tomcat 3 nach web.xml.

^x Eine hervorragende Einführung zum Einsatz des Apache mit Tomcat findet sich in der Zeitschrift iX, 2/2001, S.48ff.

^{xi} Aufgrund von Exportrestriktionen in den USA sind diese Archive nicht Teil der Tomcat-Distribution, Anwender können die Klassen von http://java.sun.com/products/jsse herunterladen.

^{xii} Vielen Dank für diese Anleitung an Herrn Behnke, Uni Bonn. Siehe auch http://www.junlu.com/msg/48529.html

^{xiii} Im letzten Release waren versehentlich noch veraltete eine veraltete propadmin.x und propadmin.bat im Verzeichnis \$SUPERX_DIR/webserver/tomcat/webapps/superx/WEB-INF vorhanden. Falls der Fehler NoClassDefFound auftaucht, löschen Sie diese zwei Dateien.

^{xiv} Bei RedHat 9.x kann es Probleme geben. Wenn Sie die Fehlermeldung "make[1]: *** [mod_jk.la] Error 1" erhalten, müssen Sie den Aufruf wie folgt ändern:

make LIBTOOL=/etc/httpd/build/libtool

^{xv} http://www.modssl.org

^{xvi} Auch in der offiziellen Doku von mod_ssl wird dies empfohlen, mit dem Zusatz, dass nur root und der apache-daemon

diese Datei lesen darf.

http://www.modssl.org/docs/2.0/#FAQ-nopass

^{xvii} www.blackdown.org

xviii Der Pfad des vorinstallierten Plugins mit der JRE lautet bei SuSE

/usr/lib/SunJava2/jre/plugin/i386/ns610-gcc32

ⁱ in älteren Versionen des SuperX-Kernmoduls laden die Rohdaten unter db/rohdaten. Dies hat sich als unpraktisch erwiesen

^{xix} Das Schwierigste ist der Update der Datenbank. Dies läßt sich zwar automatisch durchführen, aber wegen der heterogenen Datenbankstrukturen kann es zu Fehlern kommen. Am sichersten ist es, die Schritte im Update-Script \$SUPERX_DIR/db/install/update/kernmodul-upgrade20_to_21_ids.sql (für Postgres upgrade20_to_21_pg.sql) einzeln durchzuführen.

^{xx} Aus historischen Gründen liegen die Nummern aus Karlsruhe im Bereich 0-9990, aus Duisburg im Bereich 10000-19990.
^{xxi} Warum ausgerechnet Word? Das Access-Frontend ist in Visual-Basic-for-Applications programmiert, und nach unserer Erfahrung ist dies der am meisten verfügbare Editor mit VBA-Unterstützung, wenn auch Access (als Teil von MS Office) installiert ist. Der Editor WordPad z.B. bietet keine VBA-Schnittstelle. Uns war außerdem eine ausgefeilte Such- und Undo-Funktion wichtig. Theoretisch könnte man in der mitgelieferten Dokumentvorlage editblob.dot im gleichen Verzeichnis auch Autotexte und Makros hinterlegen. Daher: Auch wenn es ungewöhnlich ist, Word als IDE zu benutzen: nach unserer Erfahrung ist es recht praktisch. Fehlt nur noch die farbige Syntaxunterstützung...

^{xxii} Details zu rsync siehe Dr. Boris Pasternak, Dr. Uwe Meyer-Gruhl (2003). Der Gleich-Macher. Dateien mit Rsync synchronisieren. c't 10/2003, S. 116ff.