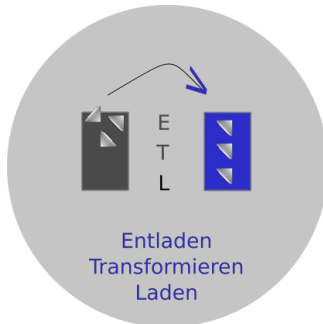


Inhaltsverzeichnis	1
Moduldokumentation ETL	2
Aktuelle Version und Verfügbarkeit	2
Nutzen	2
Perspektivisch	3
Installation	3
Voraussetzungen	3
Vorgehen bei Neuinstallation- kurz und knapp	4
Vorgehen beim Upgrade	4
Dokumentation und Ressourcen	4
Benutzungsanleitungen	4
Administrative Anleitungen	4
Versionshistorie	4
Version 0.4	4
Version 0.3	5
Version 0.2	5
Version 0.1b	5

Moduldokumentation ETL



Das SuperX-Modul ETL liefert keine neuen Auswertungen, sondern erweitert das SuperX-Kernmodul um komfortablere Gestaltung und Bedienung von Laderoutinen im Browser oder per Kommandozeile.

Aktuelle Version und Verfügbarkeit

Aktuelle Version:
0.3

Das Modul ist frei verfügbar und kann [hier](#) heruntergeladen werden. Support bekommen Sie [hier](#).

Nutzen



Unser ETL-Modul macht, anders als im Bild, Ihre Transfer Routinen leicht, schnell & komfortabel.

Viele Hochschulen entwickeln derzeit eigene Masken und Funktionalitäten (Sichten etc.), was direkt zu den neuen Wünschen führt.

1. Die Quellcodes in einem "menschenslesbaren" Format zu speichern und zu versionieren.
2. Die Quellcodes von einer Säule (z.B. Entwicklungssäule) zu einer anderen "Säule" (z.B. Produktivsystem) zu transferieren.

Zu diesem Zweck wurde im neuen SuperX-Modul Laderoutinen (ETL-Modul) eine Browseroberfläche geschaffen, die folgende Aktivitäten ermöglicht:

- Übernahme von "normalen" Laderoutinen in die Datenbank
- XML-Exporte und Importe von Konfigurationen
 - Masken und anhängige Tabellen (Metadaten, Stylesheets, Menüs)
- Ausführungssteuerung von Laderoutinen im Browser

Perspektivisch

Perspektivisch soll dieses SuperX-Modul auch weitere Funktionalitäten bieten, z.B.

- Säulenübertragung von Konfigurationen (Masken, Sichten, Repository-Variablen etc.) via Browser oder Kommandozeile
- Ausführung und Kontrolle von Laderoutinen im Browser
- Ladeschritte wiederverwertbar machen (z.B. Entladen Prüfungen in Hauptladeroutine Studierende und Unterladeroutine "Lade Prüfungen aus POS")
- Zusammenlegung der Install/Upgrade Jobs in einen Job
- Idempotenz der Ladejobs, und Fernsteuerung der Schritte mit Administrationstools wie Ansible
- Laderoutinen in Schleifen, z.B. einen Bericht über mehrere Organisationseinheiten ausführen, nach Excel Speichern, und per Mail versenden

Derzeit arbeiten Laderoutinen weitgehend unsichtbar, Sie "sehen" höchstens einen Fortschrittsbalken, und bei Bedarf schwer lesbare Logausgaben. Es wird häufig bemängelt dass die Laderoutinen intransparent sind. Das wollen wir verbessern, die Laderoutinen sollen sich "lesen lassen wie ein Roman":

1. Es wird zur Laufzeit beschrieben, was passiert. Da das "Buch" meistens im Keller steht, benötige ich Fernzugriff (-> Logausgabe im Browser)
2. Es gibt Kapitel und Unterkapitel, die auch einzeln lesbar bzw. lauffähig sind (-> Teile der Laderoutinen einzeln starten)
3. Es gibt für jedes Kapitel Leitfragen bzw. "Soll"-Zustände, die wir erreichen wollen (-> Füllstatus von Tabellen etc.)
4. Bei jedem Kapitel weiß ich, ob es eher bis zum Ende lesen sollte oder jederzeit abbrechen kann (-> stoppen oder weitermachen, auch im Fehlerfall)
5. Ich kann auch wählen ob ich ein Kapitel genau lese oder nur überfliege (-> Loglevel)
6. Auch kleine Unterbrechungen sollten möglich sein, d.h. ich stoppe das Lesen / die Laderoutine, und
 - schaue mir Details in Ruhe an (-> Tabelleninhalte)
 - Ich kann mir Seitenzahlen und Kapitelnummern notieren als Lesezeichen, wo ich nochmal nachschauen kann (-> Ladeschritt-Nummern oder Identifier, über die ich sie ansteuern kann)
7. Es werden für jede Seite Metadaten zur Laufzeit geschrieben, z.B. Laufzeiten, Datenmengen -> Verbessertes Logging
8. Teile des Codes lassen sich einfach herauskopieren, um z.B. ein Einzelschritt-Debugging zu leisten.
9. Kapitel die mich nicht interessieren kann ich überspringen -> Ladeschritte deaktivieren
10. Ich kann eigene Notizen hinter jedes Unterkapitel heften -> Laderoutinen um eigene Mini-Laderoutinen erweitern, z.B. Mailversand oder hochschulspezifische Anpassungen
11. Ich möchte auch ein "Autorensystem" haben, um eigene Bücher (->Laderoutinen) zu schreiben/zu bauen. Die Möglichkeiten sollten von Kettle Jobs bis hin zu eigenen Java-Klassen reichen, die über ein Interface eingebunden werden können
 - Dafür benötige ich Werkzeuge, wie Stift und Papier (->ETL-Ladeschritte, z.B. Upload von Excel oder CSV)
12. Zeitgesteuerter Aufruf analog zu Cron/Quartz

Installation

Voraussetzungen

- Sie benötigen das Kernmodul 5.0 bzw. HISinOne-BI 2021.12/2022.06



Derzeit ist das ETL Modul in HISinOne 2022.12 und 2013.06 **nicht** nutzbar. Teile des ETL Codes sind im Rahmen der "neuen Komponentenverwaltung" ins BI Kernmodul gewandert, sind aber auf einem älteren Stand. Wir arbeiten an einer

Standardisierung, bis dahin können Sie mit den [Kommandozeilen-Scripten](#) arbeiten, mit dem JDBC_CLASSPATH auf die alte SuperX-jar.

Vorgehen bei Neuinstallation- kurz und knapp

- Laden Sie das Modul herunter und speichern Sie es in \$SUPERX_DIR.
- Achten Sie dabei darauf, daß Sie die richtige Version laden (Zeichencodierung ISO oder UTF-8, Datenbanksystem Postgres oder Informix)
- Entpacken Sie das Modul mit

```
tar -xvzf Dateiname
```

- Weitere Anleitungen finden Sie im [Modul ETL Administrationshandbuch](#).

Vorgehen beim Upgrade

Wenn Sie bereits das ETL-Modul installiert haben, müssen Sie wie folgt vorgehen:

- Laden Sie das ETL-Modul herunter und speichern Sie es im Verzeichnis \$SUPERX_DIR auf dem Server, und entpacken Sie die Datei
- Führen Sie den Patch in der Shell im Verzeichnis \$SUPERX_DIR/db/module/etl/upgrade mit

```
etl_upgrade.x
```

aus.

Danach bitte Tomcat neu starten, und dann ist das Modul einsatzbereit.

Dokumentation und Ressourcen

Benutzungsanleitungen

- Ein [Lehrfilm](#) zum Exportieren und Importieren von Masken
- Ein [Benutzungshandbuch](#)

Administrative Anleitungen

- [Installation und Konfiguration](#)
- Eine Datenbankbeschreibung (Tabellen, Prozeduren, Masken) im [html](#)- und [rtf](#)-Format.

Versionshistorie

Version 0.4

15.5.2024

- Maskenexport

- Beim Exportieren gibt es jetzt einen "Herunterladen"-Button, der ein XML produziert, das unbeeindruckt vom jeweils verwendeten Betriebssystem und Browser exakt dem Format entspricht, das auch die Kommandozeile erzeugt. Die ist für Versionierungstools wichtig, z.B. den git-Abgleich. #3
- Auch Metadaten zu Stylesheets mit Kernmodul 5.0 werden nun exportiert und geladen #2
- Kopieren-Button funktioniert auch unter Firefox, inkl. Tooltip #2
- Kompatibilität zu HISinOne-BI verbessert.
 - Das ETL Modul war mit HISinOne-BI 2023.12 oder höher nicht kompatibel, wg. geänderter Klassen mit gleichem Namen im Kernmodul. #4
 - Beim Tomcat Start unter HISinOne-BI werden alle Jobs gelöscht, so dass das ETL Modul hier nicht nutzbar war. Dies wurde korrigiert.

Version 0.3

15.08.2023

- Anpassungen ans Kernmodul 5.0
- XML Export von Masken in Firefox und Chrome identisch (vorher war Leerzeichen-Differenz)

Version 0.2

27.8.2021

- Laderoutinen zum Entladen und Hochladen von Masken enthalten auch
 - Themenbaum-Eintrag
 - Zugehörige Stylesheets
 - Zugehörige Stylesheet-Feld-Vorbelegungen
- Paket im Download-Bereich, Administrations-Dokumentation

Version 0.1b

6.1.2020

Die Features:

- Laderoutinen zum Entladen und Hochladen von Masken auf der Basis von git-freundlichem XML
- Kommandozeilenaufwurf des Entladens und Hochladens