Inhaltsverzeichnis	1
Kettle für Fortgeschrittene - Monitoring, Embedding	2
Kursbeschreibung	2
Monitoring/Logging von Kettle Jobs	3
Mailversand in Kettle	3
Mailversand Beispiel 1	3
Mailversand mit Variablen	5
naliversalio hiir ausgelagenen valiablen	° 7
Versionierung von Jobs	, 9
Entfernen der Connection-Flemente	9
Automatisierung mit Groovy	10
Jobmanagement	12
Eigenes BI-Modul	12
Ladejob-Maske erstellen und ausführen	13
Tomcat Konfiguration	14
Job registrieren	14
Ladejob-Masken erstellen Ladejob-Masken nutzen	15 17
Kitchen	17
Home-Verzeichnis	
Imgebungsvariablen	
Kettle.properties	18
Shared.xml	18
Shared.xml in SuperX bzw. HISinOne-BI	19
Ausiuniung, Logging und Femiennelaung	Uع ۱۹
Ausführungsskript inkl. Mailversand	19
Parameter übergeben	20

Kettle für Fortgeschrittene - Monitoring, Embedding

Pentaho Da	ata Integration
99	Version 7.0
	General Availability Release - 7.0.0.0-25 Build Date: November 6, 2018 03:38:38
	Copyright (C) 2007 - 2016 Fertaho Corporation. All rights reserved. Licensed under the Apoche License. Version 2.0 (the "License"); you may not use this application and all files except in compliance with the License. You may obtain a copy of the License at
@pentaho	http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law or agenet to in writing, software distributed writer he Locesse a distributed on an 7-8.3 (5° FA315, WITHOUT WARRANTES OR CONDITIONS OF ANY KIND, ether express or implied. See the Loces for the specific marging governing termissions and
A Hitachi Group Company	imitations under the License.

Kettle für Fortgeschrittene - Monitoring, Embedding

Universität Wuppertal, 8.12.2022

Kettle / Pentaho ist ein Produkt von Hitachi Vantara

Kursbeschreibung

Der Online-Kurs führt Inhalte aus der Kettle-Grundlagenschulung fort und adressiert folgende Themen

- Monitoring von Kettle Jobs
 - Mailversand
 - Logging und Fehlerbehandlung
- Embedding Kettle
 - in HISinone-BI
 - Standalone unter Linux
- Versionierung von Quellcodes

Download Kursmaterial:

- Per Browser: https://superx-rocks.de/git/Memtext/Kettle-Schulung
- Per git:

git clone https://superx-rocks.de/git/Memtext/Kettle-Schulung.git

Nach dem Download Klonen laden Sie Ihre SQL_ENV, und führen aus:

./rsync_to_h1.x

Damit wird das Modul in Ihrer BI eingespielt.

Mailversand in Kettle

Mailversand Beispiel 1

Unser erster Beispieljob versendet einfach nur eine Mail. Hier eine Gesamtübersicht der Schritte im Job:

iobmonit ▷ ▼ □	or_mail ≌ ▷ ይ	6. 8	100% -			
	START	- 8 →	DUMMY	Success-Logmail sende	m Success	

Der Job führt nur einen Dummy-Step aus, und verschickt dann eine Mail. Dazu gibt es im Reiter "Design" einen Step:



In dem Dialog im Reiter "Adresses" spezifizieren Sie die Absender- und Empfängeradresse. Danach geben Sie Daten zum Mailserver an:

	jow man dotano	
Name of mail job entry:	Success-Logmail senden	
Addresses Server EMail Message A	ttached Files	
SMTP Server		
SMTP Server:	smtp.mailserver.de	۲
Port:	587	۹
Authentication		
Use authentication?		
Authentication user:	schulung@superx-projekt.de	۹
Authentication password:	•••••	۹
Use secure authentication?		
Secure connection type	SSL	

Unser Beispiel-Mailserver verlangt Zugangsdaten zum Mailversand, diese tippen wir einfach ein. Danach wird Betreff und Inhalt der Mail definiert:

	Job mail details – S	×
Name of mail job entry:	Success-Logmail senden	
Addresses Server EMail Message	Attached Files	
Message Settings		
Include date in message? Only send comment in mail body?		
Use HTML format in mail body? Encoding	UTF-8	
Manage priority		
Priority	Normal V	
Importance	Normal	
Sensitivity	Normal	
Message		
Subject:	Erfolg	
Comment	Hier die Protokolle vom Job	
	•	
() Help	OK Cancel	

Wir versenden eine einfache Testmail mit voreingestellten Kettle-Protokolldaten. Dann starten wir den Job:

	100% -				
START	DUMMY	Success-Lo	gmail senden	Success	
Execution Results	sh matrice) 🎦 Matrice)				23
Execution Results	ob metrics	Result	Reason	Filename	23
Execution Results History [] Logging [] Job Job / Job Entry v jobmonitor mail	ob metrics 🔁 Metrics	Result	Reason	Filename	23
Execution Results Ø History (Logging) [] Job / Job Entry • jobmonitor_mail Job; jobmonitor_mail	bb metrics 🖹 Metrics Comment Start of job execution	Result	Reason	Filename	23
Execution Results O History [Logging]= Jo Job / Job Entry • jobmonitor_mail Job: jobmonitor_mail START	ob metrics 🖹 Metrics Comment Start of job execution Start of job execution	Result	Reason start start	Filename	2)
Execution Results @ History [Logging]] Job / Job Entry = jobmonitor_mail Job: jobmonitor_mail START START	bb metrics Retrics Comment Start of job execution Start of job execution Job execution finished	Result	Reason start start	Filename	2:

Der Job war erfolgreich, und es müßte tatsächlich eine Mail ankommen:



Damit haben wir ein einfaches Beispiel zum Mailversand abgeschlossen.

Mailversand mit Variablen

Im obigen Beispiel haben wir die Angaben zum Mailversand "hartcodiert" in den Dialog eingetragen. Da es vermutlich mehrere Jobs mit Mailversand gibt, ist es sinnvoll die immer wiederkehrenden Angaben zum Mailserver etc. in Variablen auszulagern. Wir verfeinern also obiges Beispiel, indem wir dem Mailversand eine Transformation vorschalten, die die Variablen definiert und setzt. Hier eine Gesamtübersicht des Jobs:

8	Spoon - jobmonitor_mail2	- e ×
File Edit Wew Action Tools Help		
8 8 8 8 8 9 *		Connect
Dires Obsign	🔁 jobmoniter, mail2-12	
Entries Q tra C to 1	▷ * □ ▷ Ø B, 🚟 100% *	
👻 🛅 General	1	- 1
💥 Transformation		- 1
🗀 Mail		
* 🛅 file management		
C HTTP	START set_variables2 Soccess-Logmail senden	
Conditions		
🖿 Scripting	1	
🖿 Bulk loading		
+ 🛅 Big Data		
Rentaho MapReduce		
🖿 Modeling		- 1
- E XML		
R XSL Transformation		

Da in der Regel solche Angaben in einer Datenbank vorgehalten werden, wählen wir in der Transformation einen Table-Input Step:

36	Spoon - set_variables2	- e ×
File Edit View Action Tools Help		
0 2 8 8 8 8 *		Connect
View Design	💝 jobmonitor_mail2 🛛 💢 set_variables2 🕴	
Steps Q C		
Property Input		- 1
RSS Input		
S3 CSV Input		
C SAP Input		
🖳 SAS Input		
Salesforce Input		
Table input		
📄 Text file input		
🕞 XBase input	 maiiparams Set maiiparams 	
R XML Input Stream (StAX)		
😰 Yami Input		
Dutput		
a Day Transformer		

Der Step selektiert die jew. Parameter, das Beispiel funktioniert unter Postgres und kann so leicht auf die eigene DB angepaßt werden.

*		$-\infty \times$
File Edit View Action Tools Help		
	c	onnect
View Design	C jobmonitor_mail2 IX set_variables2 X	
Steps Q Se 80	▷▼Ⅱ□●次▷ 其品』 100% ▼	
Property Input	Table input	- e ×
🛐 RSS Input	Step name mailparams	
S3 CSV Input	Connection eduet V Edit New	Wizard
SAP Input	501	
Calesfant Inna	Get SQL select s	tatement
Table input	select 'Kettle-Schulung' as sender_name, 'none' as encr,	ľ
Text file input	'schulung@superx-projekt.de' as sender_address, 'schulung@superx-projekt.de' as auth_user,	
R XBase input	'smtp.mailserver.de' as server_name,	
🔗 XML Input Stream (SbAX)	'587' az server_port	
🕎 Yaml Input		
> 🛅 Output		
Transform	Line 6 Column 17	
112 21.1	Enable lazy conversion	
MARD M	Replace variables in script?	
- ALAN	insert data from step	~
ATTO MIN	Execute for each row?	
AL INT	Limit size 0	0
	O Help OK Preview Cancel	

Die Parameter werden dann über den "Set Variables"-Step zur Laufzeit im gesamten Job zur Verfügung gestellt:

🛪 Spoon - set v	variables2		Set Environme	ent Variables	- 0 ×
File Edit View Action Tools Help		St	ep name : Set mailparame		
		Apply	formatting		
🔁 View 🖉 Design 💦 💱 jobmonitor_mail2 🗱 💢 set	variables2 8 Field	d values:			
Stens 0 0 1- 8- ▷ • Ⅱ □ ● Ø ▷ □	K 🖓 🖻 🗟 📰 100% 👻 🔶 F	Field name Variable name	Variable scope type	Default value	
	1 s	sender_name sender_name	Valid in the Java Virtual Machine	-1	
Property Input	2 a	auth_pw auth_pw	Valid in the Java Virtual Machine	-1	
S RSS Input	38	auth_user auth_user	valid in the Java Virtual Machine	-1	
P S3 CSV Input	40	sender address sender address	Valid in the Java Virtual Machine	.1	
🔄 SAP Input	6 s	server name server name	Valid in the Java Virtual Machine	-1	
🖳 SAS Input	7 s	server_port server_port	Valid in the Java Virtual Machine	-1	
Salesforce Input					
Table input					
Text file input maile	arams Set mailparams				
R XBase input					
👰 XML Input Stream (StAX)					
Y Yami Input					
Output					
Transform					
Utility					
> T Flow					
HERE MIKE	0	Help	OK Can	Get Fields	

Der folgende Dialog zum Mailversand wir dann statt "hartcodiert" mit Verweisen auf die Variablen versehen:

Spoon - jobmonitor_mail2	8	Job mail details – 🗸	×
	Name of mail job entry:	Success-Logmail senden	
	Addresses Server EMail Message A	Attached Files	
🛿 Welcome! 🛟 jobmonitor_mail2 🕴	Destination		
	Destination address:	danielq@memtext.de	>
	Cc:	()	>
	BCc:	· · · · · · · · · · · · · · · · · · ·	>
	Sender		
START set_variables2 Success-Logmail senden	Sender name	\${sender_name}	>
	Sender address:	\${sender_address}	•
	Reply to		
	Contact norman:		4
	Contact person.		Ľ.
	contact prone.		J.
	() Help	OK Cancel	

Im folgenden Reiter zum Mailserver sind die Angaben dann analog vorzunehmen. Der Job sollte danach genauso funktionieren wie das erste Beispiel.

Mailversand mit ausgelagerten Variablen

Im obigen Beispiel haben wir "sensitive" Angaben zum Mailversand (Benutzername, Passwort) "hartcodiert" in die Datenbank geschrieben. Hier ist es sinnvoll, diese Angaben aus Job und Datenbank herauszunehmen. Kettle bietet die Möglichkeit, zentrale bzw. systemspezifische Variablen in eine Datei "kettle.properties" auszulagern, Details dazu siehe unten.



Das verschlüsselte Passwort und die Zugangskennung wird dann als Variable übergeben.

Spoon - set_variables3	元	Table input	- 0 ×
	Step name	mailparams	
	Connection	eduetl 🗸	Edit New Wizard
Welcome! Composition mail3 Image: set_variables3 Image: set_vari	sqL		Get SQL select statement
	<pre>select 'Retle-Schulung' as sender_nam 'none' as enor, 'schulungBuuers-preikt.de' as sender 's NAM.105M' as auth_user,' Sathphaliscivelnde as server_name, '587' as server_port</pre>	, _address,	[
mailparams Set mailparams	Line 6 Column 17 Enable lazy conversion Replace variables in script?		
	Insert data from step		~
	Execute for each row?		
	Limit size	•	
	() Help	OK Preview Cancel	

Das Passwort-Feld ist maskiert, die Eingabe ist

\${MAILPW}



Bei einer Änderung der kettle.properties müssen Sie Spoon neu starten.

Loglevel und -dateien

Kettle bietet diverse Möglichkeiten zum Logging und zur Fehlerbehandlung:

- Logausgabe von Spoon / Kitchen
 - In Jobs kann man diese verfeinern mit dem "Write to log"-Step
- Eine Transformation kann
 - die Logausgabe in einer dezidierten Logdatei speichern bzw. erweitern
 - im Aufruf eines Jobs in eine Fehlerbehandlung laufen

Das folgende Beispiel zeigt die Nutzung, im Job wird eine Logdatei angelegt, und dann schreiben die einzelnen Transformationen in diese Logdatei. Diese wird dann am Ende per Mail als Anhang verschickt - egal ob es mit Fehler oder erfolgreich war:



Im ersten Schritt wird die Logdatei "ladejob.log" angelegt bzw. geleert, wenn sie schon existiert:

*	Spoon - jobm	onitor_logausgabe		- 2 X
File Edit View Action Tools Help				
		4	Write to file	- 0 ×
View Design	🛟 jobmonitor_logausgabe 🛙	Job entry name:	Logdatei leeren	
Explorer Q 🛛 🖬 🗄		File		
👻 🚞 Jobs		File name:	\${Internal.Job.Filename.Directory}/ladejob.log	Browse
👻 🛟 jobmonitor_logausgabe		Create parent folder:		
Database connections		Append file		
Job entries		Content		
Hadoop clusters		Encoding	1175-9	
🛅 Slave server	START Logdatei leere		logaurgabe	
		Iexu	Logausgabe	
				I
		() Help	OK Cancel	

Die erste "Set variables"-Transformation wird im Reiter "Logging" so konfiguriert, dass die Ausgabe in die Logdatei "ladejob.log" geloggt wird.



Ergänzend wollen wir auch die gesamt-Logausgabe des Jobs um eine Zeile erweitern, die die übergebene Mailkennung aus der kettle.properties ausgibt:

iobmonitor_logausgabe % ▷ ▼ □ ▷ □ □			(
	in .	Write to log	- 0 ×
	Job entry name	Write To Log	
	Log level	Basic	~
START Logdater leeren set_variables	Log subject	Mailversand Konfiguration	•
, Sector	Log message	Mailuser: \$(auth_user)	I
Ť.			
Write To Log			
write to Log	() Help	OK Cancel	

Danach wird eine Transformation erstellt, die Postgres-spezifische Systemvariablen

- selektiert und
- das Ergebnis in die Logdatei schreibt

				Co	nnect	
🔁 jobmonitor_logausgabe 🛛 💥 collect_logs 🖾	α.	Table input				- 0 ×
	Step name	get_pg_env				
	Connection	eduetl		Edit	New	Wizard
	SQL			Get SQ	L select st	atement
get_pg_env	SELECT trim(name) as name, tri where category='Preset Options order by 1;	m(setting) as setting FROM	pg_se	ettings		•
¥	Line 1 Column 0					
	Enable lazy conversion					
write pg Log File	Insert data from step					~
	Execute for each row?					
	Limit size	0				•
	() Help	DK Preview Car	ncel			

Da diese Operation aus der Datenbank liest und ggf. Fehler bewirken kann, wird neben der normalen Erfolgs-Ausgabe ein "Fehlerkanal" eingerichtet, der eine entsprechende Fehler-Mail verschickt:



Im Reiter "Attached files" wird die Logausgabe des Jobs angehängt:



Unser Job arbeitet von nun an mit einer Fehler- und Logbehandlung.

Mit den "Logleveln" können Sie wählen wie detailliert geloggt wird. Möglich sind:

- Nothing
- Error
- Minimal
- Basic (dies ist der Default)
- Detailed
- Debug
- Row level (Achtung: erzeugt viele Logeinträge)

Versionierung von Jobs

Kettle Jobs und Transformationen sind XML-Dateien und lassen sich mit Versionskontrollsystemen (z.B. git) versionieren. Es gibt allerdings ein paar Fallstricke:

- Kleinste Layoutänderungen in Spoon führen mitunter zu großen Änderungen im Quellcode, so dass Sie solche Änderungen eher in separate Commits trennen sollten, und den Layout-Comment in der Commit Message z.B. als "reine Layout-Änderung" kennzeichnen.
- Die Connection-Angaben müssen entfernt werden. Wie das geht zeigen wir nun.

Entfernen der Connection-Elemente

Kettle für Fortgeschrittene - Monitoring, Embedding

Beim Speichern eines Jobs wird immer die Connection mitgespeichert. Daher muss sie vor dem Commit ins git gelöscht werden. Hierzu ist der gesamte Tag sowohl im Job, als auch in den Transformationen zu löschen.



Die Connection-Information in den einzelnen Schritten bleibt jedoch enthalten.

```
<entry>
   <name>Update kenn_stelle_hhpl</name>
   <description/>
   <type>SQL</type>
   <sql>update kenn_stelle_hhpl set summe &#x3d;&#xa;&#x28;se
   <useVariableSubstitution>F</useVariableSubstitution>
   <sqlfromfile>F</sqlfromfile>
   <sqlfilename/>
    <sendOneStatement>F</sendOneStatement>
   <connection>eduetl</connection>
    <parallel>N</parallel>
   <draw>Y</draw>
   <nr>0</nr>
   <xloc>688</xloc>
   <yloc>112</yloc>
 </entry>
</entries>
```

Automatisierung mit Groovy

Sie können mit dem Java-Tool Groovy und einem von uns mitgelieferten Script die Connection-Elemente automatisch entfernen. Gehen sie dazu wie folgt vor:

- Stellen Sie sicher dass am Arbeitsplatz-Rechner Java installiert ist
- Laden Sie Groovy herunter, und entpacken Sie es in einem Ordner Ihrer Wahl (am besten ohne Leerzeichen im Pfadnamen).
- Fügen Sie den bin-Pfad der Groovy-Installation in Ihren PATH an,
 - unter Windows z.B.:
 - in der Systemsteuerung -> Erweiterterte Einstellungen -> Umgebungsvariablen -> PATH

	dware Erweitert Computerschutz Remote	
mgebungsvaria	blen	٤
Benutzervaria	ible bearbeiten	×
		_
Name der Var	iablen: PATH	
Wert der Vari	ablan: Itools/groov/groov/-2.4.10/bin:9	
Wert der van	abient (tools groovy groovy-2.4. to pin, 7	
	OK Abb	rechen
	OK Abb	rechen
Systemyariabler	OK Abb	rechen
Systemvariable	OK Abb	rechen
Systemvariabler Variable	OK Abb	rechen
Systemvariabler Variable ComSpec	OK Abb	rechen
Systemvariabler Variable ComSpec FP_NO_HOST NIIMBER_OF	OK Abb	rechen
Systemvariabler Variable ComSpec FP_NO_HOST NUMBER_OF_ OS	OK Abb	rechen
Systemvariabler Variable ComSpec FP_NO_HOST NUMBER_OF_ OS	Vert C:\Windows\system32\cmd.exe C NO P 1 Windows_NT	rechen
Systemvariabler Variable ComSpec FP_NO_HOST NUMBER_OF_ OS	OK Abb Wert C:\Windows\system32\cmd.exe _C NO P 1 Windows_NT Neu Bearbeiten Lös	chen

• unter Linux z.B.:

PATH=\$PATH:~/tools/groovy/groovy-2.4.10/bin export PATH

- Wechseln Sie in der Shell (DOS oder Linux) in das Verzeichnis, wo die ETL-Jobs liegen (hier also /superx/superx/WEB-INF/conf/edustore/db/module/myjobs/etl)
- Legen Sie dort einen Unterordner an mit dem Namen einer Kopie des ETL-Verzeichnisses , hier also z.B. "jobmonitor_git".
- Dann führen Sie das Script aus

Unter Linux:

groovy ~/git/myjobs/scripts/groovy/copy_kettlejob.groovy --strip-connections jobmonitor/jobmonitor.kjb jobmonitor_git

Unter Windows:

groovy Z:\\git\\myjobs\\scripts\\groovy\\copy_kettlejob.groovy --strip-connections jobmonitor\\jobmonitor.kjb jobmonitor_git

Damit werden alle Connection Elemente in dem Job und in den darin aufgerufenen Transformationen entfernt:

C:\Users\superx>groovy Z:\git\myjobs\scripts\groovy\copy_kettlejob.groovystri p-connections jobmonitor/jobmonitor.kjb jobmonitor_git WARNING: An illegal reflective access operation has occurred WARNING: Illegal reflective access by org.codehaus.groovy.reflection.CachedClass (file:/Z:/tools/groovy/groovy-2.4.10/bin//lib/groovy-2.4.10.jar> to method ja va.lang.Object.finalize() WARNING: Please consider reporting this to the maintainers of org.codehaus.groov y.reflection.CachedClass WARNING: Useillegal-access=warn to enable warnings of further illegal reflect ive access operations WARNING: All illegal access operations will be denied in a future release Config file: Copying jobmonitor\jobmonitor.kjb to jobmonitor_git strip params: false, strip c onnections : true \$(Internal.Job.Filename.Directory}/set_wails_to_send.ktr \$(Internal.Job.Filename.Directory/collect logs.ktr
C:\Users\superx}_

Wenn das geklappt hat, können Sie die bereinigten Dateien in den "richtigen" Ordner "jobmonitor" kopieren und versionieren.

Jobmanagement

Im folgenden zeigen wir wie Sie Jobs "on demand" direkt aus HISinOne oder SuperX im Browser vom Nutzer starten lassen.



Achten Sie darauf, dass Ihre lokale genutzte Version zu der in HISinOne-/SuperX integrierten Version paßt. Sonst könnte es ggf. Probleme geben.

Software	Version	Kettle Version
HISinOne-BI	ab 2021.06	Kettle 8.3
SuperX	ab Kernmodul 4.9	Kettle 6.0

Eigenes BI-Modul

Ein Kettle Job ohne Eingabeparameter kann in einem eigenen "Mini"-Modul deklariert und über das HISinOne-Jobmanagement ausgeführt werden.

- 1. Erstellen Sie ein minimales Modul mit einer Modul-XML-Datei
 - Siehe z.B. https://superx-rocks.de/git/Memtext/Kettle-Schulung
- 2. In der Modul-XML-Datei im Bereich ETL können Sie Ladejobs deklarieren
- Das Modul können Sie auf eine SuperX- oder BI-Installation synchronisieren oder die zip-Datei dort unter \$SUPERX_DIR bzw. webapps/superx entpacken
- 4. Danach ist es in der BI-Komponentenverwaltung sichtbar:

Gebäude, Räume, Flächen Komponente	mbs	1.2	25.05.2021	@ ○ ↔ → ←
Studierende, Prüfungen Komponente	hisinone	1.3	11.08.2022	<u>∎</u>
Amtliche Statistik Komponente		<u>1.0</u>	11.08.2022	副の一
Studienverlauf Komponente		0.9	11.08.2022	@ ┍ →
Studiengänge Komponente		0.4	16.08.2021	<u>∎</u>
Promovierende Komponente	hisinone	1.0	10.11.2022	〕 ♀ ¢ ≯ ↓ ↓
+ Bewerbung, Zulassung Romponente	hisinone	0.6	25.05.2021	∄ ↔ + ↔ + ↔
Management Komponente		1.8b	07.01.2022	₫ ៚ →
Grunddaten und Kennzahlen Komponente	eduetl	2.0b	25.11.2022	〕 ♀ ¢ ≯ ↓ ↓
Forschung Komponente	hisinone	1.2	25.05.2021	〕 ↔ ↔ + ↔
Leistungsmonitoring Komponente	hisinone	0.4b	16.08.2021	〕 ♀ ↩ → ←
Qualitätssicherung Komponente		0.5	19.11.2022	<u>∎</u> ભ
Wuppertaler Ladejobs Komponente				

Mit Klick auf das "Installations-"Icon wird es installiert



und danach ist es auch in der Konnektorenübersicht sichtbar, inkl. Unter-Ladejobs:

Wuppertaler Ladejobs Komponente	0.1b	01.01.1900	0 m	1	
Johmonitor Konsektor				1	+
Qualitätssicherung Xomponente	0.5	19.11.2022	0	1	

Die Jobs können dann über das BI Jobmanagement ausgeführt werden.

Auch ein zeitgesteuertes Ausführen ist möglich , per Kommandozeile.

Über die Browser-Oberfläche ist ebenfalls ein zeitgesteuerter Update möglich. Gehen Sie dazu in die Admin-Rolle, und wählen das Menü Administration -> Konfigurationsassistenten:

🛢 🏟 \star HISin One 📜 🗨 Menü durc	Û	٠	₽
Sie sind hier: Startseite) Administration) Konfigurationsassistenten			
Konngurationsassistenten			
Kernsystem verwalten			
PSV verwalten			
Business Intelligence verwalten			
Konnguration Business Intelligence			
Informationon			=
mornauonen			
Konfigurationsassistenten für die Erstkonfiguration der Produktbereiche			

Ajuf dem Reiter "Business Intelligence" können Sie den jew. Job markieren und so in die nächtliche Laderoutine aufnehmen:



Weitere Details siehe die Beschreibung des Installationsassistenten bei HIS.

Ladejob-Maske erstellen und ausführen

Tomcat Konfiguration

Vorab: Sie müssen die Umgebungsvariable CATALINA_OPTS erweitern:

CATALINA_OPTS="... -DMODULE_PFAD=/var/lib/tomcat9/webapps/superx/WEB-INF/conf/edustore/db/module ..."

Unter Ubuntu Linux liegt das in der Datei

/etc/default/tomcat9

dort die Variable

JAVA_OPTS="... -DMODULE_PFAD=/var/lib/tomcat9/webapps/superx/WEB-INF/conf/edustore/db/module ..."

Danach muss man Tomcat neu starten.

Job registrieren

Um den Job im System bekannt zu machen wird ein Eintrag in der Tabelle sx_jobs angelegt. Sie erreichen die Bearbeitung über das Menü Administration -> Tabelle suchen.



Datensatz 1 - 2 von insgesamt 2 Sätzen.

Im Listenformular können Sie neue Jobs anlegen, oder vorhandene ändern.

		4	4	4	
	58	personal_astat	Amtl. Lieferung Personal einlesen	kenn/etl/personal_astat /pbv_load.kjb	Kennzahlen
BODONS LA DE X DE CONSTRUCTOR ESCALAR DE CONSTRUCTOR ESCALAR DE CONSTRUCTOR DE CONSTRUCTOR ESCALAR DE CONSTRUCTOR	838	sva_pbv_astat	Amtl. Lieferung Personal einlesen (Komponente Personal, Stellen)	sva/eti/personal_astat /pbv_load.kjb	Personal
Weiterverarbeitung: Generisches Standardlayout	59	stud_astat	Amtl. Lieferung Studierende einlesen	kenn/etl/stud_absolv_astat /stud_load.kjb	Kennzahlen
Stichwort: job ; User: superx Stand: 30.07.2014	56	zul_bew_d	Bewerber Mathenote einlesen	zul/etl/mathenote /zul_bew_d_load.kjb	Bewerbung, Zulassung
Name Tabelle Beschriftung Sachgebiet Bearbeiten sx_jobs_edt sx_jobs Ladejobs verwalten Administration Image: Comparison of the state of	19	import_cw	Curricularanteile einlesen	gang/etl/import_cw /import_cw.kjb	Studiengänge
sx_jobs_list sx_jobs Ladejobs verwalten Administration	62	doc_upload	Dokument hochladen	kenn/etl/doc_upload /doc_upload.kjb	Kennzahlen
Datensatz 1 - 2 von insgesamt 2 Sätzen.	06	import_ects_soll	ECTS Soll einlesen	Im/etl/import_ects_soll /import_ects_soll.kjb	Leistungsmonitoring
	54	hsfinanz_stat_kam	Hochschulfinanzstat. (kam.) einlesen	kenn/etl/hs_finanz /hs_finanz_kam_load.kjb	Kennzahlen
	-55	hsfinanz_stat_kfm	Hochschulfinanzstat. (kaufm.) einlesen	kenn/etl/hs_finanz /hs_finanz_kaufm_load.kjb	Kennzahlen
	64	jobmonitor	Jobmonitor	myjobs/etl/jobmonitor /jobmonitor_logausgabe.kjb	
	56	kennz2017_manuell	Kennzahlen-Katalog BaWue 2017 (manuell) einlesen (XLSX)	kenn/etl /kennz2017_manuell /kennz2017_manuell.kjb	Kennzahlen
	57	nhs_manuell	Kennzahlen NHS (manuell) einlesen (XLSX)	kenn/etl/nhs_manuell /nhs_manuell.kjb	Kennzahlen
Immose	19	import_lehr_fb_rsz	Lehreinheiten, FB und RSZ einlesen	sos/etl/import_lehr_fb_rsz /import_lehr_fb_rsz.kib	Studierende

Hier unser Beispiel:

-		
Ladejobs verwa	alten. 🔁	
	巴 @ 40	0
tid	464	
Unique Name	jobmonitor	
Bezeichnung	Jobmonitor	
Pfad zur Datei	myjobs/etl/jobmonitor/jobmonitor_logausgabe.kjb	
Sachgebiet	×	
Optional: Hochschulnummer		
Optional: Kenn- Profil		
Modus unterstützt?	1	
optionale Parameter		_
optionaler Prüf- SQL	select count(*) from xdummy	
	1	

Das Ergebnis des Prüfprotokolls erscheint nach Ausführen des Jobs im Ladeprotokoll. Hier können Sie z.B. die Anzahl der Datensätze in einer Zieltabelle zählen.

Ladejob-Masken erstellen

:

Die Masken, welche Ladejobs ausführen benötigen zwingend das Feld **dokettlejob**. Anhand dieses Feldes wird dem System mitgeteilt, dass ein Kettle-Job auszuführen ist. Das Feld darf versteckt werden.

}6	B089	s l ᢓ ⊡ @ ∠	2 🖉 🖻 🖓 🕻		Ì		
sind hie	r: Abfragen > Administrati	ion 🗡 Masken verwalten ≯ F	Felder > Feld suchen/Bericht erstellen	Datensätze/Feld	suchen		
'eiterv	erarbeitung: Gen	erisches Standard	layout 🗸 🗸				
Fei	a sucher	1					
eld der	Maske: 8000 - Jo	obmonitor ausführe	en ; User: superx Stand: 30	0.07.2014			
=eld Nr	Name	Nummer (Sortierung)	Art	Obligatorisch	Masken ID	Masken Name	Bearbeiten
8.000	Datei	40	19-CSV-Upload	0	8.000	Jobmonitor ausführen	
8.001	Job	20	1-Nummer+Text, nur mit Dialog	1	8.000	Jobmonitor ausführen	
8.002	Jahr	10	1-Nummer+Text, nur mit Dialog	1	8.000	Jobmonitor ausführen	
8.003	dokettlejob	100	0-nur Text, direkte Eingabe	0	8.000	Jobmonitor ausführen	ø
8.004	Semester	3	1-Nummer+Text, nur mit Dialog	0	8.000	Jobmonitor ausführen	A
8.005	Modus	50	1-Nummer+Text, nur mit Dialog	1	8.000	Jobmonitor ausführen	ø
8.006	Organisationseinheit	1	12-Sicht	0	8.000	Jobmonitor ausführen	ø
8.007	maxoffset	100	0-nur Text, direkte Eingabe	0	8.000	Jobmonitor ausführen	

Um die auswählbaren Kettle-Jobs zu definieren wird auf die Tabelle sx_jobs zugegriffen.

				N	/lemtext l	Jniversity		*	Maske Felderinfo verwalten - Moz	💦 📗	ł
							Iocalhost:8080/superx/edit/kern/felderinfo_edit.jsp?tid=8001				
Sie sind hier. Abfragen > Administration > Masken verwalten > Felder > Feld suchen/Bericht erstellen > Datensätze/Feld suchen							Felderinfo verwalten. 🕄				
Weiterverarbeitung: Generisches Standardlayout										••	
Feld suchen							Tid	8.001	_		
Feld der Mesler 2000 Jahmeniter susführen i Lleer super: Stand 20.07.2014							Name	Job			
rea dei maske. 0000 - 300monitor dusiunren , Osei. Superx Stand. 30.07.2014							X	0			
Feld Nr	Name	Nummer (Sortierung)	Art	Obligatorisch	Masken ID	Masken Name	Bearbeiten	Y	0		
8.000	Datei	40	19-CSV-Upload	0	8.000	Jobmonitor ausführen	S	Buttonbreite Feldbreite	150		
8.001	Job	20	1-Nummer+Text, nur mit Dialog	1	8.000	Jobmonitor ausführen	ø	Zeilenanzani Typ Laenge	1 integer v 200		
8.002	Jahr	10	1-Nummer+Text, nur mit Dialog	1	8.000	Jobmonitor ausführen	B	Obligatorisch Art	1-Ja v 1-Nummer+Text, nur mit Dialog	~	
8.003	dokettlejob	100	0-nur Text, direkte Eingabe	0	8.000	Jobmonitor ausführen		Relation	< <sql>> select tid,caption from sx_jobs where sachgebiete_id in (1400) order by 2</sql>		
8.004	Semester	3	1-Nummer+Text, nur mit Dialog	0	8.000	Jobmonitor ausführen		Attribut			
8.005	Modus	50	1-Nummer+Text, nur mit Dialog	1	8.000	Jobmonitor ausführen	A	Defaultwert			
8.006	Organisationseinheit	1	12-Sicht	0	8.000	Jobmonitor ausführen	A				
8.007	maxoffset	100	0-nur Text, direkte Eingabe	0	8.000	Jobmonitor ausführen	A				
Datensatz 1 - 8 von insgesamt 8 Sätzen.											

Die Maske kann nach den vorhandenen SuperX-BI-Techniken mit speziellen Benutzer- und Gruppenrechten versehen werden. Auch neue Maskenfelder sind möglich, diese werden automatisch als Parameter an den Job übergeben (sofern

:

der Kettle-Job den Parameter kennt).

Ladejob-Masken nutzen

Der Ladejob bietet eine einfache Browser-Oberfläche:

Sie sind hier: Abfragen > Administration > Jobmonitor a	e sind hier: Administration > Joberonitor ausführen - Bericht erstellen							
Organisationseinheit	nichts gewählt							
* Job	Jobmonitor 🗸							
	Abschicken Zurücksetzen							

In der Maske wählen Sie den Ladejob, dies ist ein Pflichtfeld. Mit dem Abschicken wird der Job gestartet, und der oben definierte Test-SQL im Sinne einer "vorher-nachher"-Messung abgesetzt.

Memtext University	×
Collection	
Weiterverarbeitung: Generisches Standardlayout	
Jobmonitor ausführen	
Semester: WiSe 2022/2023 ; Jahr: 2021 ; Job: Jobmonitor ; Modus: Alles löschen und hinzufügen ; User: superx Stand: 17.07.2022	
Import Kommentar	
Prüfselektion vor Durchführung von Ladejob Johmonitor	
1	
Prüfselektion nach Durchführung von Ladejob Jobmonitor	
Prince 1. Company College	
Datensatz 1 - 5 von insgesamt 5 Satzen.	
Erläuterung	

Kitchen

Kitchen dient der Ausführung von Kettle Jobs via Kommandozeile. Wichtig ist die Umgebung, in der Kitchen gestartet wird.

Home-Verzeichnis

Zunächst ist es wichtig, das Home-Verzeichnis von Kettle beim Aufruf von Kitchen zu kennen. Standardmäßig liegt das Verzeichnis in HOME/.kettle, also unter Linux (3 Varianten):

\$HOME/.kettle ~/.kettle /home/Benutzername/.kettle

und unter Windows (2 Varianten):

C:\users\Benutzername C:\Documents and Settings\{Benutzername}\.kettle

Im Home-Verzeichnis von Kettle werden Konfigurationsdateien gesucht. Sie können das Home-Verzeichnis auch variieren, indem Sie die System-Umgebungsvariable KETTLE_HOME setzen.

Umgebungsvariablen

Im Home-Verzeichnis wird nach zwei Dateien gesucht: kettle.properties: Allgemeine Umgebungsvariablen shared.xml: Datenbankverbindungen

Kettle.properties

Sie können in der Textdatei beliebige Konfigurationen setzen, die Sie nicht in den Jobs / Transformationen oder in der Datenbank speichern wollen, z.B. Zugangsdaten für Email-Server.

Beispielinhalt:

MAILUSER = schulung@superx-projekt.de MAILPW = anfang12

Sie können Passworte sogar verschlüsseln, indem Sie das Kommandozeilen-Tool "encr" nutzen. Unter Linux:

encr.sh -kettle anfang12

Unter Windows:

encr.bat -kettle anfang12



Wenn Ihr Passwort Leerzeichen oder Sonderzeichen enthält, ist es sicherer das Passwort mit einfachem Hochkomma (Linux) oder doppelten Anführungsstrichen (Windows) zu umschließen.

Es wird ein verschlüsseltes Passwort ausgegeben:

Encrypted 2be98afc86aa7f2e4aa17a871d095fe88

Dieses Passwort können Sie dann in die kettle.properties eintragen:

MAILUSER = schulung@superx-projekt.de MAILPW =Encrypted 2be98afc86aa7f2e4aa17a871d095fe88

Shared.xml

Speziell für Datenbankverbindungen gibt es eine eigene Datei **shared.xml**, diese liegt im Homeverzeichnis der Betriebssystem-Benutzerkennung, die den Job startet, im Unterverzeichnis .kettle . Die Datei wird von Spoon angelegt, wenn man eine Connection "teilt" (engl. "share"). Hier ein Beispiel:



Es werden zwei Datenbankverbindungen konfiguriert, die dann in Kettle Transformationen im Element "connection" genutzt werden können.



Bei der Entwicklung von Jobs und Transformationen werden die Connection-Elemente auch in den jew. Quelltext geschrieben und gespeichert. Und bei der Ausführung haben diese Elemente eine höherer Priorität als die shared.xml, so dass man diese Elemente vor dem Installieren im Zielsystem entfernen muss.

Shared.xml in SuperX bzw. HISinOne-BI

Die Datenbankverbindungen im jew. Job-Quellcode, bzw. in der Transformation, bzw. in der Kettle Kurs Teil 2 stehen in SuperX bzw. HISinOne-BI in "Konkurrenz" zur "databases.xml" (HISinOne-BI) bzw. zur "db.properties" (SuperX). Hier eine Erläuterung des Zusammenhangs:

- 1. in erster bzw. höchster Priorität wird die Connection im jew. Job-Quellcode, bzw. in der Transformation genutzt
- 2. in zweiter Priorität die Datei shared.xml im Homeverzeichnis der Benutzerkennung.
- 3. in dritter Priorität, also wenn es weder 1. noch 2. gibt, und nur beim Lauf eines Jobs in der SuperX-Webanwendung und nur bei der dbconnection "eduetl" wird die "databases.xml" (HISinOne-BI) bzw. zur "db.properties" (SuperX) genutzt.



Beim Betrieb in SuperX bzw. HISinOne-BI unter Debian/Ubuntu/Suse Linux liegt die shared.xml standardmäßig im Verzeichnis /var/lib/tomcat*/.kettle , und gehört dem User tomcat. Achten Sie auf Lese- und Schreibrechte.

Ausführung, Logging und Fehlermeldung

KETTLE_ENV einrichten

Für das im folgenden Abschnitt erläuterte Skript wird eine KETTLE_ENV benötigt, hier ein Beispiel für Ubuntu Linux.



Für Kettle-Version 7.0.0.0-25 wird Java 8 benötigt.

#!/bin/bash KETTLE_PFAD=~/kettle/7.0.0.0-25 export KETTLE_PFAD PENTAHO_JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64 export PENTAHO_JAVA_HOME PENTAHO_JAVA=java export PENTAHO_JAVA PATH=\$PENTAHO_JAVA PATH=\$PENTAHO_JAVA_HOME/bin:\$PATH export PATH PENTAHO_DI_JAVA_OPTIONS="-Xms1024m -Xmx2048m -XX:MaxPermSize=256m ' export PENTAHO_DI_JAVA_OPTIONS

Ausführungsskript inkl. Mailversand

Folgendes Skript führt Kitchen aus, schreibt die Ausgabe in eine Logdatei und sendet bei fehlerhafter Ausführung eine Mail mit der Logdatei im Anhang. Das Beispiel gilt für Ubuntu Linux mit dem Mailclient "s-nail".

#!/bin/bash
. ./KETTLE_ENV
JOB_PFAD="\$MYJOBS_PFAD/etl/jobmonitor"
JOB_FILE="jobmonitor_logausgabe"

\$KETTLE_PFAD/kitchen.sh /file=\$JOB_PFAD/\$JOB_FILE.kjb /norep >\$JOB_PFAD/\$JOB_FILE.log 2>&1

if "0" -ne "\$?"

then # Mailversand echo "Mail wird versandt." echo "Logfile im Anhang." | s-nail -s "Kettle-Job \$JOB_FILE.kjb ist fehlgeschlagen!" -a "\$JOB_PFAD/\$JOB_FILE.log" test@mailserver.de

else echo "Job lief ohne Fehler." fi

Zunächst wird KETTLE_PFAD gesetzt, also der Ort, wo Kettle installiert ist. Dann werden der Jobpfad sowie -name übergeben.

Zum Ausführen des Kettle-Jobs durch Kitchen wird im Kettle-Verzeichnis unter Linux kitchen.sh ausgeführt. Über die Option -file werden der Dateipfad und -name übergeben. Die Ausgabe wird in eine Logdatei geschrieben.

\$KETTLE_PFAD/kitchen.sh /file=\$JOB_PFAD/\$JOB_FILE.kjb /norep >\$JOB_PFAD/\$JOB_FILE.log 2>&1

Abschließend erfolt eine Auswertung des Returncodes. Falls dieser ungleich 0 ist, also einen Fehler jeglicher Art kennzeichnet, wird eine Mail inklusive Logdatei versandt.

if "0" -ne "\$?" then # Mailversand echo "Mail wird versandt." echo "Logfile im Anhang." | s-nail -s "Kettle-Job \$JOB_FILE.kjb ist fehlgeschlagen!" -a "\$JOB_PFAD/\$JOB_FILE.log" test@mailserver.de else echo "Job lief ohne Fehler." fi

Achtung:

Kitchen versteht als Parameter-Präfix sowohl "-" als auch "/", also z.B. "-file:..." und "/file:...". Man könnte auch "=" statt ":" als Trenner nehmen. Aber in einer Publikation von Matt Casters et al. (2010, S.323) werden "/" und ":" empfohlen, die machen unter Windows/DOS weniger Probleme.

Parameter übergeben

Neben der Übergabe eines Jobfiles lässt kitchen.sh weitere Optionen zu (s. Kitchen Optionen). Sehr nützlich ist die Übergabe von Parametern für den ausgeführten Job.

./kitchen.sh /file=test.kjb -param:Semester=20222

In diesem Beispiel wird dem Jobparameter Semester der Wert 20222 übergeben. Bei Parametern mit Leerzeichen müssen Sie Anführungszeichen drum herum setzen.

Sie können auch spezielle JVM-Parameter übergeben, java-typisch mit -D vorangestellt.

./kitchen.sh /file:test.kjb -param:Semester=20222 -Dlanguage=de