

eCRF, Trial Control System und SAS-BI unter einem Dach: Studienüberwachung und Workflow-Optimierung durch real- time Reporte mit SAS-BI, integriert in eine Web-Application, die eCRF und Trial Control System miteinander verbindet

Uta Mehdorn
KOEHLER eClinical
Seeburger Str. 8
13581 Berlin
uta.mehdorn@koehler-eclinical.com

Zusammenfassung

Aus der Web-Application aufrufbare, durch Filter steuerbare SAS Stored Processes ermöglichen jetzt unseren Kunden, jederzeit auf sie zugeschnittene Berichte, Grafiken und Listen datenaktuell im gewünschten Format selbst zu generieren.

Dieser Beitrag berichtet über Herausforderungen und Lösungen bei der Vernetzung der von uns entwickelten Web-Application VITESSA mit SAS-BI:

- Konfiguration in der SAS-Management-Console und Zusammenhänge in den Metadaten der SAS-BI-Umgebung.
- Abbildung getrennter Umgebungen (Development/Test/Produktiv) in der SAS-BI-Welt analog zu den etablierten Umgebungen der VITESSA.
- Sicherstellung des Rollenkonzepts und der damit verbundenen Rechte, Umsetzung von Filtern und Auswahlfeldern.

Schlüsselwörter: SAS-BI-Metadaten, SAS-Management-Console (MMC), SAS Stored Process (STP), libname META, Access to ORACLE, real-time web-based reporting tool, clinical trial workflow management

1 Hintergrund

Die KOEHLER eClinical GmbH ist eine Contract Research Organisation (CRO) in Freiburg i. Br., Berlin und Konstanz mit fast 30 Jahren Erfahrung in Data Management, Statistik und EDC-Lösungen im Rahmen von klinischen Studien. Bereits 1998 entwickelten wir den ersten eCRF zur Erfassung von Patientendaten. Seit 2005 werden zusätzlich zu den klinischen Daten auch Studiensteuerungsdaten mit einem Trial Control System (TCS) erfasst. Damit erweiterte sich der Nutzerkreis. Die Web-Application VITESSA wurde in Hinsicht auf ergonomische, aufgabenbezogene Nutzerführung und dynamischen, nutzerfreundlichen Aufbau kontinuierlich weiterentwickelt. Eine der letzten Weiterentwicklungen betraf das Reporting. Die bisher genutzten SAS offline-Berichte genügten nicht mehr den Anforderungen. Projektleitung und Studienmanagement benötigen real-time Reporte zur Studienüberwachung und proaktiven Studiensteuerung.

Das sind z.B.

- Allgemeine Überblick-Berichte, Zusammenfassungen, Grafiken
- Plan – Ist – Vergleiche, Milestone - Tracking
- Ursachensuche bei schlechter Performance (Rekrutierung, Dokumentation,...)
- Detailliertere Berichte
- Berichte zur Datenqualität
- Berichte/Listen zur Unterstützung/Erleichterung des täglichen Arbeitsablaufs

basierend sowohl auf den Studiensteuerungsdaten als auch auf den klinischen Patientendaten. Wir entschieden uns, unsere Web-Application VITESSA mit SAS-BI zu vernetzen, um bei Änderungen oder Ergänzungen der Reporte flexibel und unabhängig vom BUILD-Prozess der VITESSA zu sein. Eine Änderung eines SAS-Programms ist schneller umgesetzt, als ein erneuter BUILD der VITESSA.

2 Anforderungen für die Vernetzung mit SAS-BI

2.1 Anforderungen aus Sicht des Nutzers

Zu Beginn einer Studie, oder bei Bedarf auch während der Laufzeit, werden auf die Bedürfnisse des Nutzers zugeschnittene Reporte definiert. Diese Reporte sollen in der Web-Application in einem Selektionsfeld angeboten werden. Der Nutzer soll Parameter, wie Zeiträume oder ausgewählte Patientengruppen, selbst durch weitere Eingabe-/Selektionsfelder festlegen können. Das Ausgabeformat (pdf, html, csv, txt, ...) soll frei wählbar sein. Sind die Parameter festgelegt, wird der Report per Button generiert. Jeder Nutzer unterliegt Restriktionen hinsichtlich der Sichtbarkeit von Daten. So darf ein Arzt nur die Daten „seiner“ Patienten sehen, nicht die einer anderen Praxis. Selbstverständlich müssen die Reporte diese Restriktionen erfüllen.

Bereits erstellte Reporte sollen für den Nutzer einsehbar bleiben. Zum Aufräumen brauchen Nutzer allerdings die Möglichkeit zum Löschen unnötiger Reporte.

2.2 Anforderungen und Ziele aus Sicht der Entwickler

Für folgende Anforderungen mussten Lösungen entwickelt werden:

1. SAS – BI mit 3 Umgebungen wie bei VITESSA: DEV, TEST und PROD
2. Nativer, auf Lesen beschränkter Zugriff auf die jeweilige VITESSA-Oracle-Datenbank (DEV, TEST oder PROD)
3. Transaktionssicherheit
4. Umsetzung der Nutzer-Restriktionen
5. Umsetzung der Nutzer-Parameter: Eingaben, Selektionen, Output-Formate
6. Log oder SAS-code niemals sichtbar für den Nutzer
7. Abfangen von SAS-BI Server-Fehlermeldungen; z.B. für einen deutschen Nutzer Ersetzung durch „Keine Daten bei dieser Selektion und daher kein Bericht.“
8. Durchgängige Verwendung von Unicode (UTF-8)

9. Konfiguration der Reporte über Oberfläche der VITESSA, jederzeit durch Administrator änderbar, unabhängig vom BUILD-Prozess der Web-Application
10. Versionskontrolle der SAS-Programme mit vorhandenem System

Zu Punkt 4 der Liste:

SAS-BI bietet eine elaborierte user- und Rechte-Verwaltung. Da die Nutzerverwaltung ausschließlich in der VITESSA erfolgt, kann der BI-Server leider nicht auf existierende user-Konten zurückgreifen, um die notwendigen Restriktionen sicher zu stellen.

3 Realisierung

3.1 Erweiterung der SAS-Architektur

Zusätzlich zum SAS-Analytics-Pro-Server wurden ein SAS-BI-Web-Server (Middle-Tier) und ein SAS-BI-Application-Server (Server-Tier) installiert, die mit der VITESSA zu vernetzen waren (Abbildung 1).

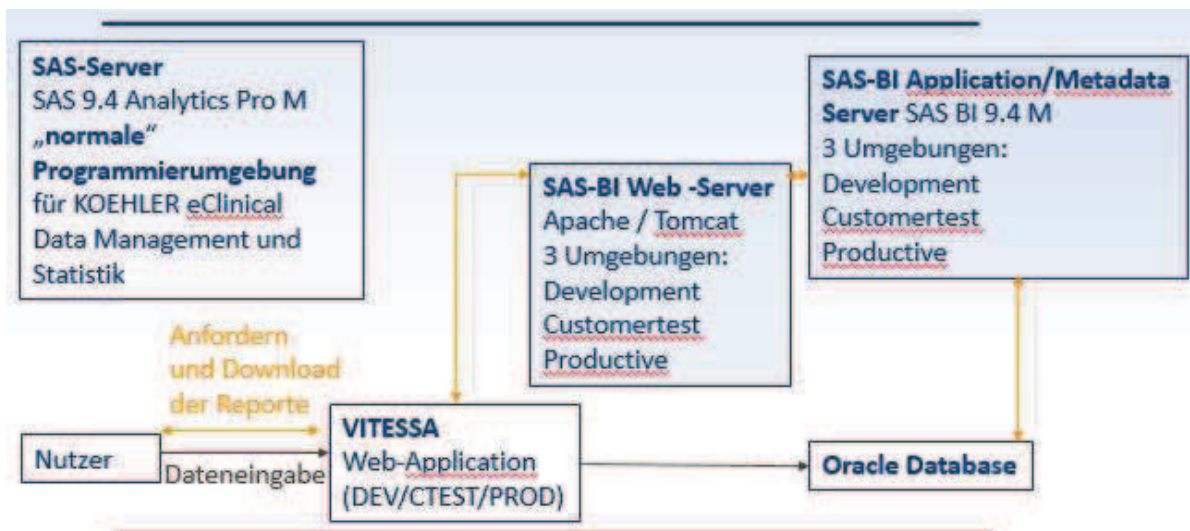


Abbildung 1: Erweiterung der SAS-Architektur

3.2 Installation von 3 Umgebungen

Bei der Umsetzung der ersten Anforderung

1. SAS – BI mit 3 Umgebungen wie bei VITESSA: DEV, TEST und PROD

wurden auf beiden Servern 3 Umgebungen durch identische Installationen aller SAS-Services mit 3 unterschiedlichen Ports realisiert (Abbildung 2):

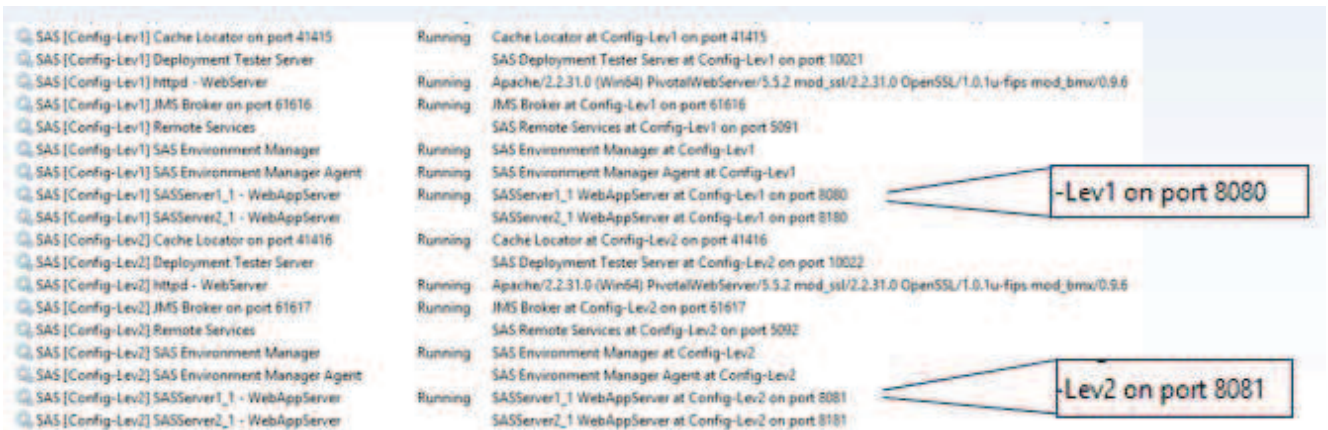


Abbildung 2: SAS Services auf dem Middle-Tier (Ausschnitt)

Die korrekte Vernetzung der 3 VITESSA – Umgebungen mit den 3 SAS-BI-Umgebungen erfolgt durch Konfigurationseinstellungen in den SAS-BI-Metadaten.

3.3 Datenzugriff und Zusammenhänge in den Metadaten

Die in den Anforderungen genannten Punkte

2. Nativer, auf Lesen beschränkter Zugriff auf die jeweilige VITESSA-Oracle-Datenbank (DEV, TEST oder PROD)
3. Transaktionssicherheit

sind als ein Zusammenspiel von URL, SAS Stored Process (STP) und den Steuerinformationen in den SAS-BI-Metadaten umgesetzt. Abbildung 3 zeigt die VITESSA-Oberfläche zur Generierung eines Reports.



Abbildung 3: Ansicht der Web-Application-Oberfläche zur Generierung der Reports

URL: Wird in der Web-Application der Report per Button gestartet (http-get-Aufruf), so wird eine URL generiert, die den STP startet und Parameter als Makrovariablenbelegungen an den STP übergibt. Zwei Parameter beinhalten Projektbezeichnung (&project) und Umgebung (&runtime):

```
http://srvsasweb01.xx.xx:81/SASStoredProcess/do?_program=/pathname/stpname/&..
...&project=P208&runtime=CUSTOMERTEST&...
```

SAS Stored Process(STP): Der Quellcode des STP beginnt mit %STPBEGIN, endet mit %STPEND und ist ansonsten im Grunde ein „normales“ SAS-Programm, welches die empfangenen Makrovariablen verarbeiten muss. Alle durch die URL übergebenen Makrovariablen werden mit einem %global Statement zu Beginn des STP Quellcodes als globale Makrovariablen deklariert.

Metadaten: Die SAS-BI-Metadaten werden über die SAS-Management-Console (MMC) konfiguriert. Zu jeder Umgebung gibt es einen separaten Metadatenserver bzw. Port. Bei Konfigurationen in der MMC ist darauf zu achten, dass die korrekte Umgebung verwendet wird (Abbildung 4):



Abbildung 4: Anzeige des SAS-BI-Application-Servers und des Ports in der MMC

Unter dem MMC-Reiter „Plug-In“ findet man u.a. die Konfigurationsmöglichkeiten für die Steuerung der Umgebung, der Datenbank-Verbindungen, der Datenbank-Zugriffsrechte, der STPs, sowie für die Meta-Informationen der Oracle-Datenbank-Tabellen. Der MMC-Reiter „Ordner“ enthält die virtuelle Ablagestruktur aller Projekt-/Umgebungs-spezifischen Meta-Informationen. Ein passendes Unterverzeichnis sollte als „Speicherort“ zu Beginn der Konfiguration angelegt werden

3.3.1 Steuerkonzept des Zugriffs auf die Oracle-Daten

Der zentrale Ausgangspunkt (innerhalb der SAS-Welt) für die Steuerung des Datenzugriffs ist das META-libname-Statement im Quellcode des STP. Das META-libname-Statement verlinkt den STP mit den zugehörigen SAS-BI-Metadaten in BIBLIOTHEKEN. Wie im Folgenden gezeigt wird, werden Konfigurationen im Plug-In BIBLIOTHEKEN mit Konfigurationen in anderen MMC Plug-Ins *und* mit Einträgen in einer externen Konfigurationsdatei für Oracle-Datenbank-Verbindungen verknüpft (Abbildung 5). Diese Konfigurationsdatei wurde bei der Installation des Oracle-Treibers auf dem Server-Tier hinterlegt. Sie ist Grundlage für die Vernetzung der 3 VITESSA-Umgebungen mit den 3 SAS-BI-Umgebungen. Das META-libname-Statement mit den Makrovariablenbelegungen für &project und &runtime steuert so letztlich den Zugriff auf die Oracle-Datenbank.

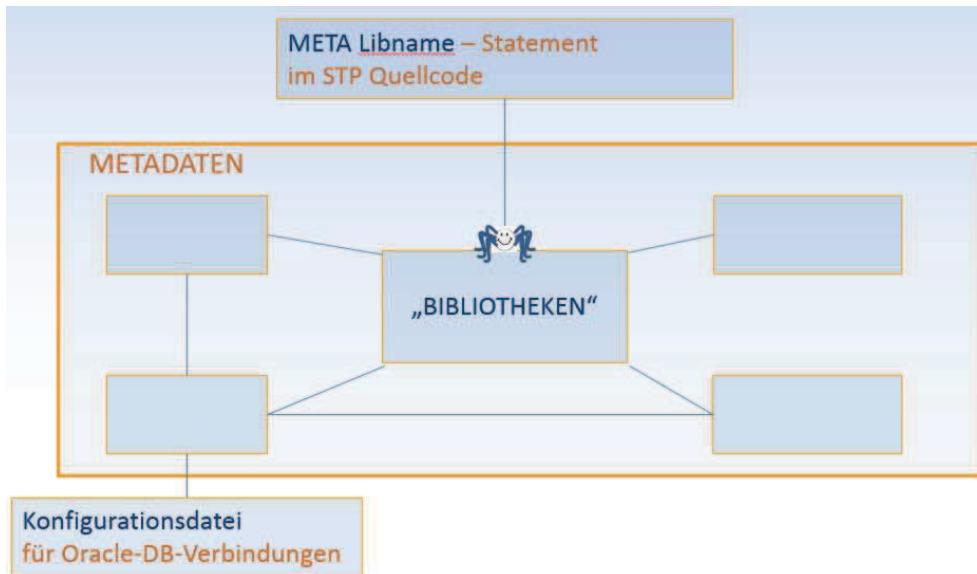


Abbildung 5: Steuerkonzept

3.3.2 Vom STP zu den Metadaten

Durch die VITESSA werden über Makrovariablenbelegungen Umgebung und Projektbezeichnung an den STP übergeben. Aus diesen Makrovariablenbelegungen wird der Name der META libname library zusammengesetzt (z.B. P208_ORA_DEV für die Development-Umgebung des Projektes P208, siehe Abbildung 6).

```

%STPBEGIN;
***** global macro variables;
%global ... project runtime ...;
%macro startup;
    %let timestamp = %sysfunc( date(), date9.) %sysfunc( time(),time5.);
    * set environment conditions;
    libname master META library="&project._ORA_&runtime" ;
    option compress=yes; |
%mend startup;
%startup;

%STPEND;
    
```

```

&projekt=P208
&runtime=DEV
    
```

Abbildung 6: Zentraler Ausgangspunkt: das META-libname-Statement im STP Quellcode

Die Verknüpfung des STP mit den Metadaten erfolgt durch eine BIBLIOTHEK mit identischem virtuellen Namen „P208_ORA_DEV“ (MMC Plug-In BIBLIOTHEKEN – Reiter „Allgemein“, „Name:“ - siehe Abbildung 7).

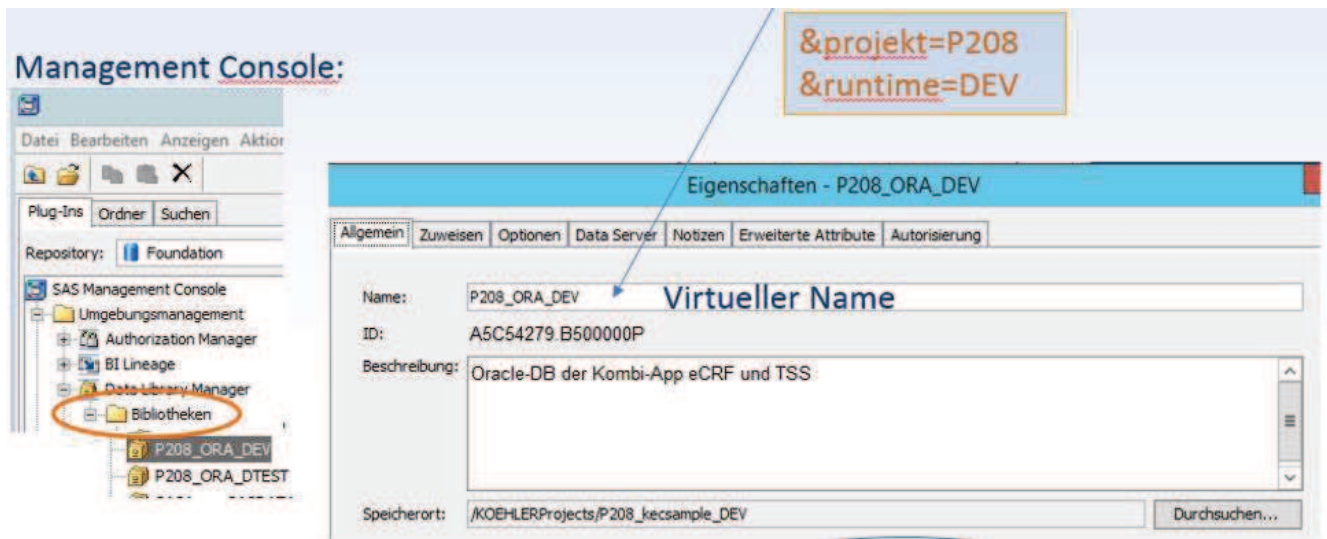


Abbildung 7: Ausschnitte der MMC
Zentraler Anker: der Name der BIBLIOTHEK in den Metadaten

3.3.3 BIBLIOTHEKEN – Die Spinne im Zentrum des Netzes

Im MMC Plug-In BIBLIOTHEKEN sind alle Verknüpfungen zu Konfigurationen in anderen MMC Plug-Ins gebündelt.

Der Reiter „Allgemein“ verweist mit „Speicherort:“ auf das virtuelle Verzeichnis unter dem MMC-Reiter „Ordner“, in dem alle projektspezifischen Meta-Informationen für diese Umgebung gespeichert werden.

Unter dem Reiter „Optionen“ muss in der BIBLIOTHEK eine maximal achtstellige in allen SAS-BI-Metadaten eindeutige Zeichenfolge „libref“ eingetragen werden. Diese Zeichenfolge wird vom BI-Server dazu verwendet, mit der jeweiligen umgebungsspezifischen Projekt-Datenbank eine Verbindung aufzubauen.

Die Konfiguration des Verbindungstyps als „GLOBAL“ unter dem Reiter „Optionen“, „erweiterte Optionen“, „Verbindung“ gewährleistet *Transaktionssicherheit*.

Der Reiter „Data Server“ – „Verbindungsinformationen“ (Abbildung 8) bündelt alle weiteren Verknüpfungen und kann mit einer Spinne verglichen werden, die in der Netzmitte sitzt, wo alle Fäden zusammenlaufen:



Abbildung 8: MMC Plug-In BIBLIOTHEKEN – Verbindungsinformationen:
4 „Stränge“: Verweise auf andere Plug-Ins der MMC

Zu besserer Unterscheidung wird im Folgenden konsequent für tatsächliche (menschliche) Benutzer der Begriff *Nutzer* und für technischen Benutzer der Begriff *user* verwendet.

Erster „Strang“ des Netzes: Datenbankserver

Der virtuelle Name des *Datenbankservers* „VDEV_Oracle“ verknüpft die BIBLIOTHEK mit der Konfiguration im MMC Plug-In SERVER MANAGER. Dort wird der Typ der zu verbindenden Datenbank konfiguriert, in diesem Fall ein ORACLE Server.

Zweiter „Strang“ des Netzes: Datenbankverbindung

Der virtuelle Name der *Verbindung* zur Oracle-Datenbank „Connection VDEV_Oracle“ verknüpft die BIBLIOTHEK mit der Konfiguration im MMC Plug-In SERVER MANAGER → Verbindungen und verweist damit auf die Verbindungsinformationen zum Datenbankserver.

Der dort unter dem Reiter „Optionen“ angegebene Pfad triggert mittels der externen Konfigurationsdatei für Oracle-Datenbank-Verbindungen „tnsnames.ora“ die Verbindung zur Datenbank des Projekts in der jeweiligen Umgebung, also DEV, TEST oder PROD. Abbildung 9 zeigt die Konfiguration für die Verbindung zur Oracle-Datenbank in der Development-Umgebung: „Pfad: VDEV“.

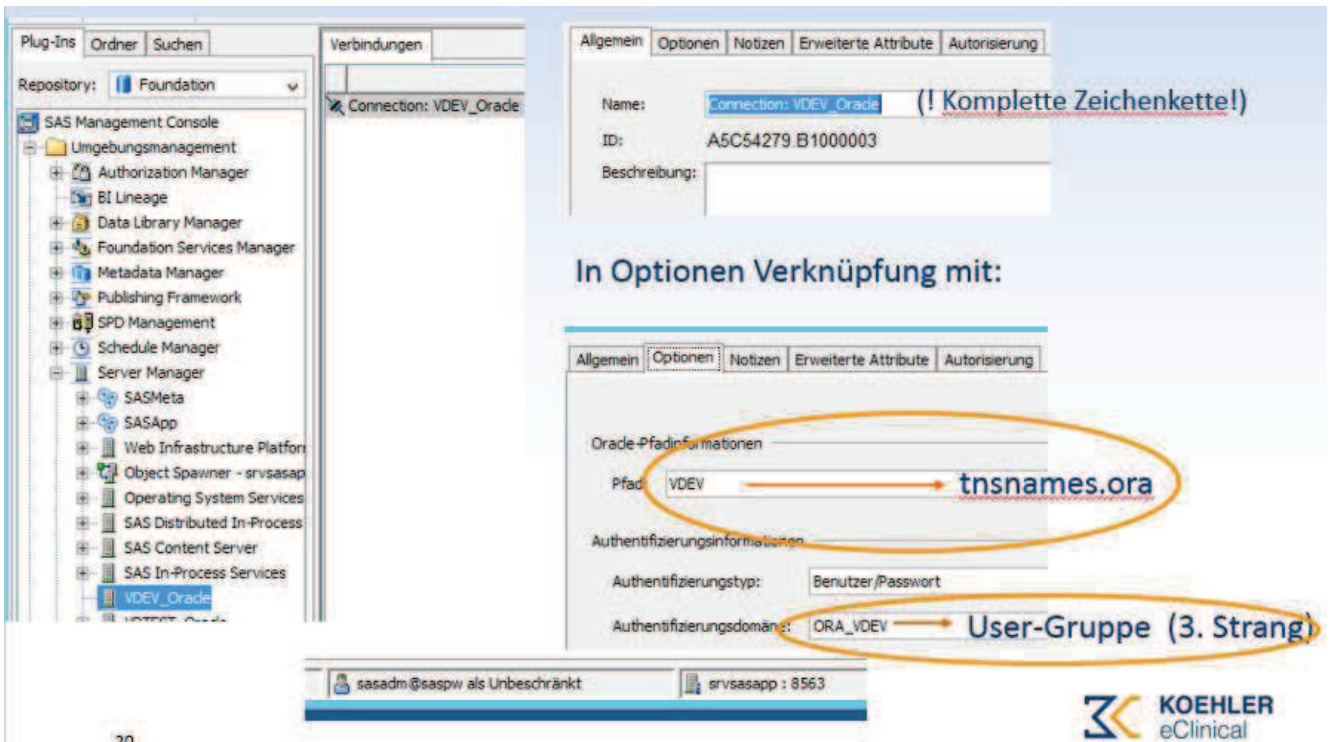


Abbildung 9: MMC Plugin SERVER MANAGER – Verbindungen:
 Referenz der Einträge in der Konfigurationsdatei tnsnames.ora
 Referenz der Authentifizierungskonfiguration (user-Gruppe)

Abbildung 10 zeigt die Einträge in der Konfigurationsdatei tnsnames.ora. Diese Datei triggert den Zugriff auf die Oracle-Datenbanken der VITESSA:

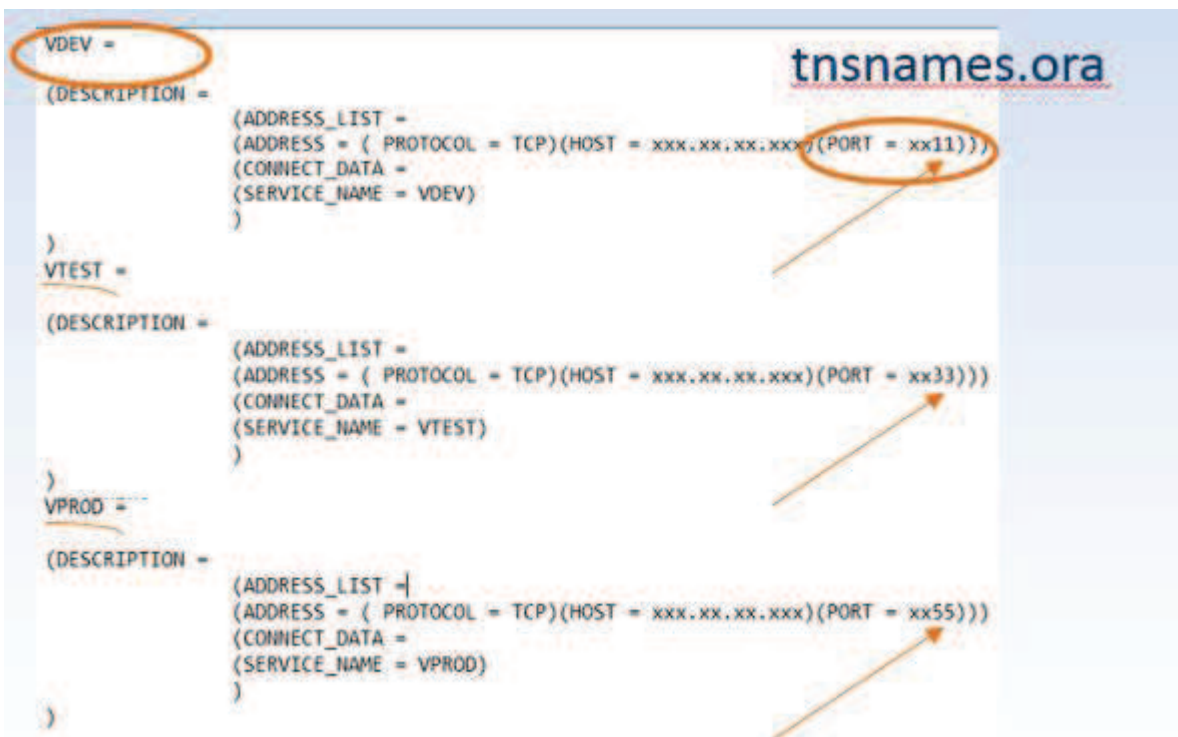


Abbildung 10: VITESSA Konfigurationsdatei tnsnames.ora
 und Zusammenhang zu den Einträgen in den SAS-BI-Metadaten

Dritter „Strang“ des Netzes: User Gruppe und Anmeldekonto

Im dritten Strang der Abbildung 8 verweist „Standardanmeldung: Konto (user-Gruppe)“ auf die Konfiguration der user-Gruppe und deren Zugangsrechte zur Datenbank im MMC Plug-In USER MANAGER. Wie man am Eintrag der Authentifizierungsdomäne in Abbildung 9 sieht, sind die Meta-Informationen zu den Zugangsrechten auch mit den Meta-Informationen zu den Datenbankverbindungen (quer-)verknüpft.

Da die Rechteverwaltung der Nutzer ausschließlich in der VITESSA erfolgt, erfolgt der Zugriff auf STPs incl. SAS-BI-Metadaten mittels eines einzigen users. Als technischer SAS-user mit dem Recht, die Metadaten zu lesen und den STP auf dem Server-Tier auszuführen, verwenden wir sasdemo. Sasdemo ist einziges Mitglied der umgebungsspezifischen user-Gruppe, im Beispiel der Abbildung 11 der user-Gruppe ORA_VDEV.

Der Zugang zur VITESSA Oracle-Datenbank eines Projektes in der jeweiligen Umgebung ist wiederum durch user der Oracle-Datenbank geregelt. Diese user werden in den SAS-BI-Metadaten als Konten der user-Gruppe angelegt. Für ein und dieselbe user-Gruppe, deren einziges Mitglied der technische SAS-user sasdemo ist, wird pro Projekt eine user/password-Kombination für die VITESSA Oracle-Datenbank hinterlegt. Der technische Oracle-Datenbank-user &project_readonly hat ausschließlich *Leserechte* auf der Oracle-Datenbank.

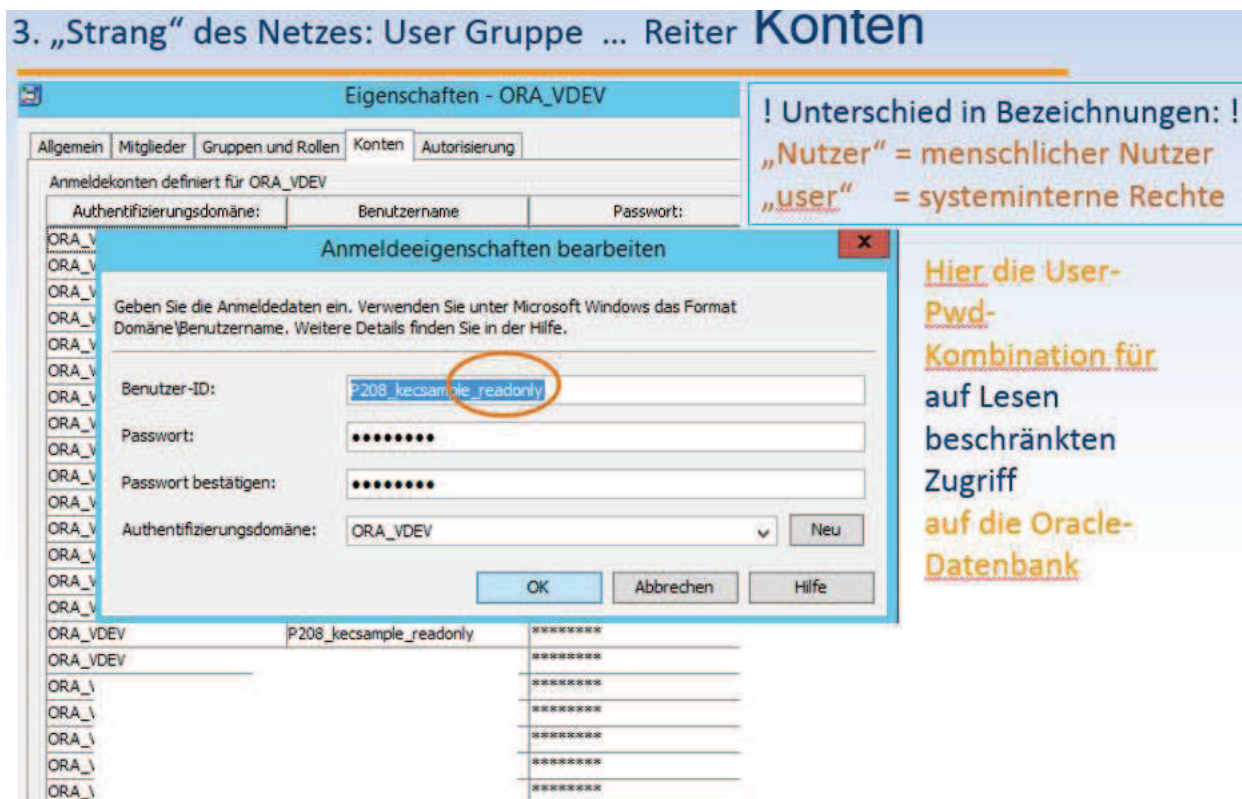


Abbildung 11: Ausschnitte der MMC: Konten in der User-Gruppe

Vierter „Strang“ des Netzes: Datenbankschema

Der vierte Strang ist der Eintrag des Datenbankschemas. Das Hinterlegen des Datenbankschemas ist Voraussetzung für die Erstellung der SAS-BI-Metadaten der Oracle-Datenbank-Tabellen. Im Plug-In BIBLIOTHEKEN werden über die Aktion „Tabellen registrieren“ die Metadaten der Datenbank-Tabellen erzeugt (Abbildung 12).

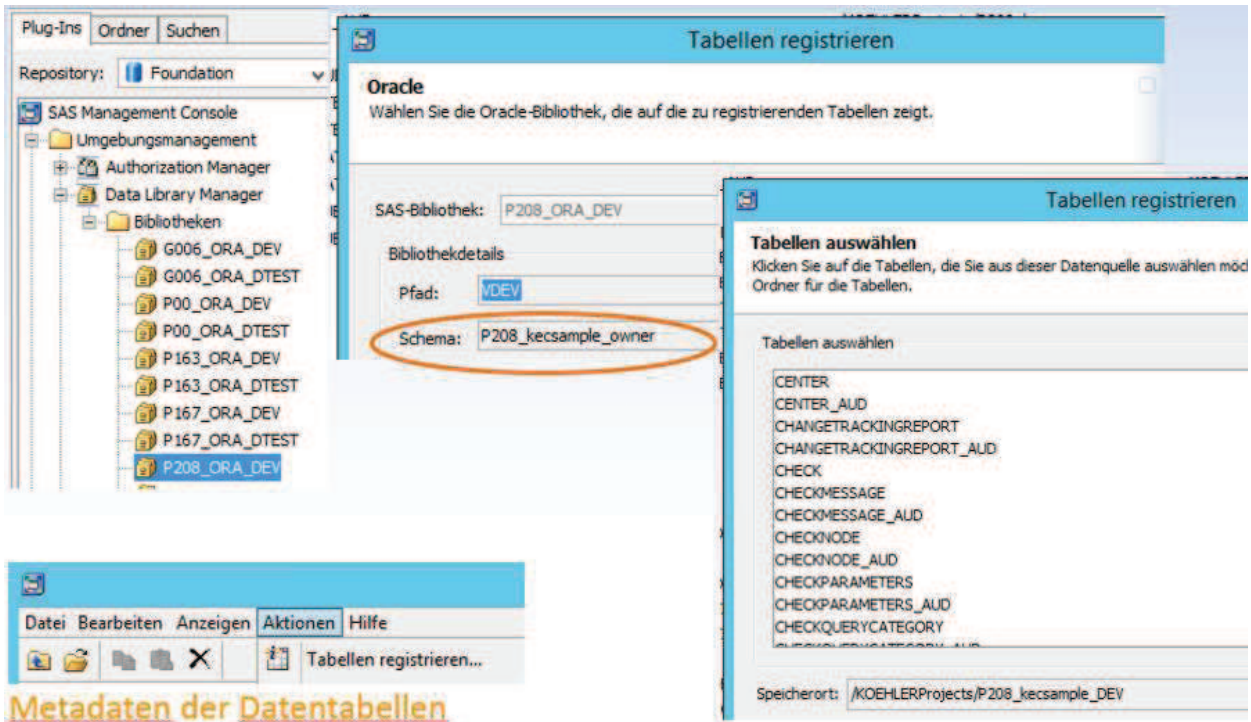


Abbildung 12: Ausschnitte der MMC: Registrieren der Metadaten der Oracle-Datenbank-Tabellen

3.4 Report Steuerung

3.4.1 Restriktionen, Selektionen und weitere Eingaben

Die in den Anforderungen genannten Punkte

4. Umsetzung der Nutzer-Restriktionen
5. Umsetzung der Nutzer-Parameter: Eingaben, Selektionen, Output-Formate

werden in gleicher Weise durch ein Zusammenspiel von URL, STP und SAS-BI-Metadaten realisiert. Die durch die VITESSA erzeugte URL beinhaltet auch diese Parameter als Makrovariablenbelegungen. Die Makrovariable &idcenterforuser verweist auf bestimmte IDs, die mit *genau den* Daten verknüpft sind, die *genau der* Nutzer sehen darf (Restriktionen). Weitere Makrovariablen beinhalten die Parameter der Selektionen (ein bestimmtes Zentrum, siehe Abbildung 13) oder anderer Eingaben des Nutzers, wie z.B. bestimmte Patienten, ein Startdatum, ein Enddatum. Auf die Output-Formate wird später eingegangen.

Selektions- und Restriktions-Parameter, die als Makrovariablen an SAS gegeben werden:

[http://srvsasweb01.xx.xx:81/SASStoredProcess/do?_program=/pathname/stpname&...
&idcenterforuser=23456&...](http://srvsasweb01.xx.xx:81/SASStoredProcess/do?_program=/pathname/stpname&...&idcenterforuser=23456&...)

[http://srvsasweb01.xx.xx:81/SASStoredProcess/do?_program=/pathname/stpname&...
&idscenter=72514,72358&idspatient=72894,75624&...](http://srvsasweb01.xx.xx:81/SASStoredProcess/do?_program=/pathname/stpname&...&idscenter=72514,72358&idspatient=72894,75624&...)

[http://srvsasweb01.xx.xx:81/SASStoredProcess/do?_program=/pathname/stpname&...
&startdate=06NOV2014&enddate=05SEP2016&...](http://srvsasweb01.xx.xx:81/SASStoredProcess/do?_program=/pathname/stpname&...&startdate=06NOV2014&enddate=05SEP2016&...)



Abbildung 13: Ausschnitt der Web-Application-Oberfläche zur Generierung der Reports sowie Beispiel-URLs mit Restriktions- und Selektions/Eingabe - Parametern

Die Nutzer-Restriktionen werden nur im STP-Quellcode umgesetzt. Daraus resultieren hohe Anforderungen an die Validierungsprozedur. Da die Parameter, die die VITESSA als Makrovariablen in der URL übergibt, standardisiert sind, können sie mit Hilfe eines Standard-Makros verarbeitet werden.

Restriktions-Makrovariablenbelegungen und für das Filter-Statement geeignete Selektions- Makrovariablenbelegungen werden in diesem Standard-Makro zu einem filter-statement zusammengefasst, wie in einfacher Form hier dargestellt:

```
%macro create_filter;
%let filter= ;
%if &idcenterforuser ne %then %do;
  %let filter=IDCENTER in (&idcenterforuser);
  %if &cohort ne %then %do; *cohort;
    %let filter=&filter. and COHORT in ("&cohort.");
  %if &idscenter ne %then %do; *center;
    %let filter=&filter. and IDCENTER in
      (&idscenter);
  %if &idspatient ne %then %do; *patient;
  %let filter=&filter. and IDPATIENT in (&idspatient);
%end;
...
%mend;
%create_filter;
```

Dieser Filter *muss* dann beim Zugriff auf die Oracle-Daten angewendet werden.

```
data selected;
set master.ECRFdata;
if &filter;
run;
```

Nicht-Restriktions-Makrovariablenbelegungen müssen an geeigneter Stelle im SAS-code berücksichtigt werden:

```
data in_time_frame;
set selected;
if input(&startdate.,date9.)<=AESTART and in-
put(&startdate.,date9.)<= AEEND <=input(&enddate.,date9.) then ...;
run;
```

3.4.2 Output-Formate

Die im 5. Punkt den Anforderungen genannte Steuerung der Output-Formate kann bei einem SASStoredProcess/do? – Aufruf leicht realisiert werden.

Die Steuerung erfolgt durch die Makrovariablenbelegung &_odsdest in der URL durch Selektion des entsprechenden Document Types (siehe Abbildung 14) in der VITESSA:

The screenshot shows the 'SAS Reporting' web application interface. The main heading is 'SAS Reporting'. Below it, there is a section titled 'Run new Report'. This section contains three input fields: 'Report' with the value '01.2 Reasons For Gap Between Planned And Enrolled I', 'Document Type' with the value 'PDF Document', and 'Filename' with the value 'SasReport_11'. The 'Document Type' field is highlighted with an orange circle. At the bottom left of the form, there is a blue button labeled 'Run SAS-Report'.

Abbildung 14: Ausschnitt der Web-Application-Oberfläche zur Generierung der Reports mit Auswahlfeld für das Output-Format

Die den STP aufrufende URL lautet dann:

```
http://srvsasweb01.xx.xx:81/SASStoredProcess/do?_program=/pathname/stpname&..
&_odsdest=pdf&....
```

Im STP Quellcode ist kein weiterer code nötig. Auch in den Metadaten müssen keine diesbezüglichen Konfigurationen vorgenommen werden.

Allerdings hat dieses einfache Verfahren den Nachteil, dass Grafiken ausschließlich im pdf-Format angeboten werden können.

3.5 Stabilität

Um die Stabilität des Tools zu gewährleisten, müssen u.a. folgende Punkte realisiert werden:

6. Log oder SAS-code niemals sichtbar für den Nutzer
7. Abfangen von SAS-BI Server-Fehlermeldungen; z.B. für einen deutschen Nutzer Ersetzung durch „Keine Daten bei dieser Selektion und daher kein Bericht.“

Der Aufruf der Reporte durch SASStoredProcess/do? hat dahingehend einige Tücken:

Im Fall eines ERRORS *nach* Erzeugung eines outputs wird das log für den Nutzer sichtbar. Um das zu vermeiden, darf in den SAS-Programmen für real-time-Reporte nach Erzeugung des Outputs kein weiterer code enthalten sein. Eine sorgfältigste Validierung der SAS-Programme ist selbstverständlich.

Vorhersehbare Fehlermeldungen werden durch Standard-Makros abgefangen, wie z.B. durch dieses: Sollte ein Nutzer eine Selektion treffen, für die keine Daten in der Datenbank sind, erzeugt ein Makro einen Dummy-output mit einem entsprechenden Hinweis: “No data retrieved”

```
%check_empty(ds=reportdataset,empty_info=No data retrieved);
```

Bei internationalen Studien kann dieses Makro basierend auf den login-Informationen des Nutzers (als Makrovariable in der URL übergeben) eine Ausgabe in der entsprechenden Sprache generieren.

3.6 Restliche Anforderungen

Im Folgenden wird kurz auf die restlichen Anforderungen aus der Sicht der Entwickler eingegangen:

8. Durchgängige Verwendung von Unicode (UTF-8)

Die durchgängige Verwendung von Unicode (UTF-8) verhindert nicht korrekt dargestellte Zeichen in den Reporten, besonders bei internationalen Studien. Um das zu gewährleisten, muss bei der Installation aller beteiligten Systeme konsequent auf diese Konfigurationseinstellung geachtet werden.

9. Konfiguration der Reporte über Oberfläche der VITESSA, jederzeit durch Administrator änderbar, unabhängig vom BUILD-Prozess der Web-Application

Für die Web-Application VITESSA wurde eine Konfigurationsoberfläche entwickelt, die von einem Nutzer mit admin-Rechten zur Konfiguration der real-time Reporte verwendet wird. Dieser Nutzer kann zu jedem beliebigen Zeitpunkt Reporte hinzufügen, entfernen oder die Konfiguration der Reporte ändern (Abbildung 15).

Reportdefinition Identifier	07
relative URL	/contacts
mandatory Parameter	
optional Parameter	CENTERS
Assigned Documentsection	1. Clinical Operations Reports
Report Title (z.B. en=report title;de=Name des Berichts)	en=07. Site Contacts
Report Description (z.B. en=report description;de=Beschreibung des Berichts)	en=Site Contacts
Hints for report (z.B. en=report hints;de=Hinweise zum Bericht)	
Possible output formats (pdf;rtf;html;csv;xml, blank=all)	html;csv
View/Open Rights for Report (z.B. SHOW1;SHOW2)	
Request Rights for Report (z.B. REQUEST1;REQUEST2)	
List of choosable StudyEvent-OIDs (z.B. DEMOVISIT;VISIT1, blank=all)	
List of choosable Form-OIDs (z.B. DEMOVISIT;VISIT1, blank=all)	

Abbildung 15: Ausschnitt der Web-Application-Oberfläche zur Konfiguration eines Reports

10. Versionskontrolle der SAS-Programme mit vorhandenem System

Die SAS-Programme der STPs werden wie alle anderen unserer SAS-Programme in einem Versions-Kontroll-System verwaltet. Um zu gewährleisten, dass Kunden bei Änderungswünschen bis zur Wirksamkeit des Updates den Report in der gegenwärtigen Form generieren können, kann das jeweilige SAS-Programm in den unterschiedlichen Umgebungsverzeichnissen auf dem Server-Tier mit unterschiedlichen Subversion-Nummern ausgecheckt werden. Hat das SAS-Programm die Entwicklungs-Phase von DEV

bis PROD durchlaufen, so hat dieses Programm in jedem Verzeichnis die gleiche Subversion-Nummer.

Zum Schluss noch eine Bemerkung zum letzten Punkt der Anforderung aus Nutzer-Sicht:

Bereits erstellte Reporte sollen für den Nutzer einsehbar bleiben. Zum Aufräumen brauchen Nutzer allerdings die Möglichkeit zum Löschen unnötiger Reporte.

Die vom BI-Server erzeugten Report-files werden von der VITESSA „entgegengenommen“, abgespeichert und auf der Report-Seite für den Download bereitgestellt. Der Nutzer mit entsprechenden Rechten kann per Button Reporte gezielt löschen.

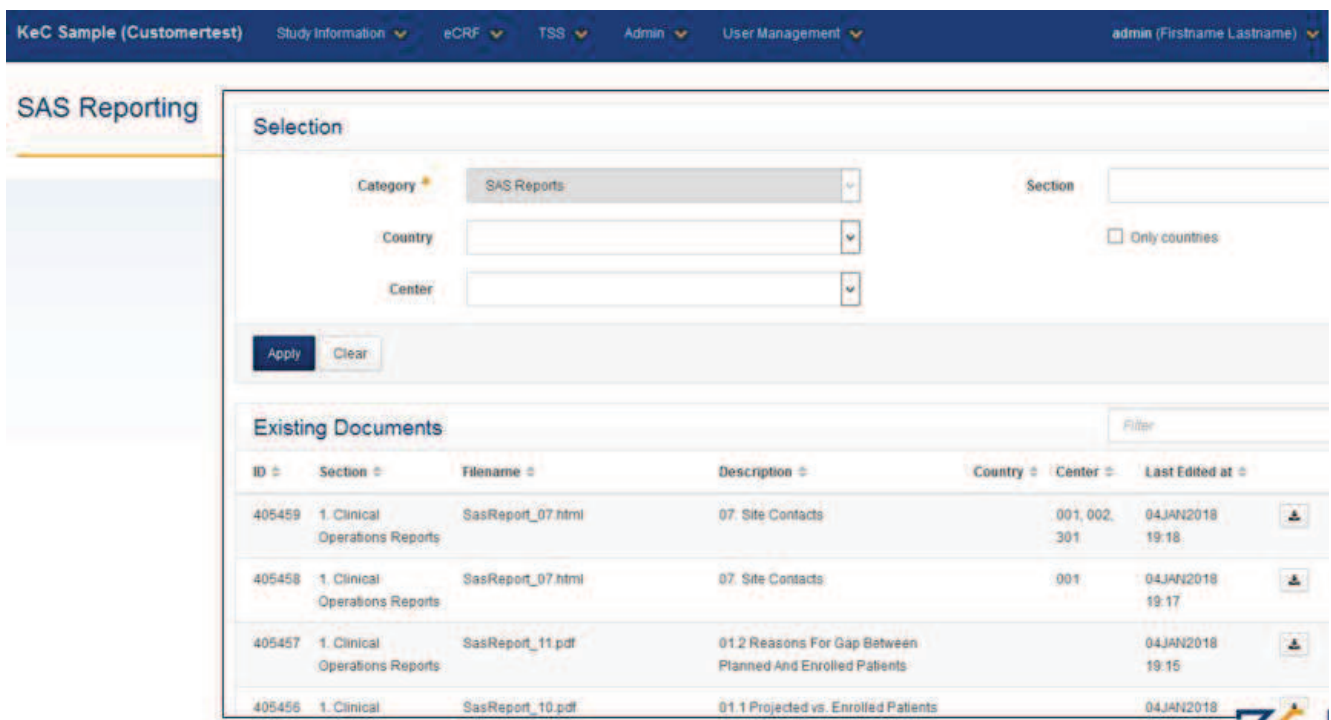


Abbildung 16: Ausschnitt der Web-Application-Oberfläche zur Generierung der Reports sowie die Liste der zum Download verfügbaren Reports

4 Weiterentwicklung

Das real-time web-based reporting tool der VITESSA wird aktuell noch weiterentwickelt. Der Fokus liegt auf der Erweiterung der output-Formate über die REST-Schnittstelle, um auch direkt Grafik-formate (gif, jpg) und Excel für unsere Kunden bereitzustellen.