Installationshandbuch
BALVI Schnittstellenserver
(Version 3.5.x vom 18.12.2018)

Inhaltsverzeichnis

1 Allgemeine Hinweise

1.1 Schnittstellen und Profile

Der BALVI Schnittstellenserver (BSS) ist ein Zusatz-Server, welcher in eine vorhandene BALVI iP Infrastruktur eingebunden werden kann, um neue automatisierte Webservice-Schnittstellen zur Verfügung zu stellen. Ab Version 3.4.x gibt es zudem die Möglichkeit, „Profile“ zu verwenden. Das Standard-Profil „ip“ verwendet weiterhin zur Authentifizierung und Autorisierung die Benutzerverwaltung von BALVI iP1. Das Profil „Standalone“ verwendet den in Apache Tomcat integrierten UserRealm für die Authentifizierung und Autorisierung und ist nur für Kunden gedacht, die keine BALVI iP Infrastruktur betreiben.

(Nicht vollständige) Liste der unterstützten Schnittstellen:

- Synchronisation mit BALVI mobil XT ab Version 1.9
- Daten-Export an eFi (BVL, Schnittstellen-Pilotbetrieb wurde eingestellt)
- Bidirektionale Datenaustausch-Schnittstelle zum LIMS
- xdomea-konforme DMS-Export-Schnittstelle (Sachsen)
- Betriebsstättenregister-Schnittstelle zum Abgleich von Betriebsstammdaten
    - Erfordert zusätzlich das zBR und den zBR-BSS-Adapter
- TSK-Meldebogen-Schnittstelle (Sachsen)
- Import-Schnittstelle XGewerbeanzeige des BMWi gem. Spez. 1.2/1.3
- RioPP LM – Übernahme der Probenpläne (NRW)
- Synchronisation mit den BALVI iP2-Modulen:
    - iP2 – DM – Düngemittelmodul (NRW)
    - iP2 – LM – Lebensmittelmodul (Pilotbetrieb NRW)
    - iP2 – BTS – Betriebsstättenmodul (Pilotbetrieb NRW)
    - iP2 – Öko – Öko-Kontrollmodul (NRW)
- Kompatibilität mit BALVI mobil2
    - BALVI mobil2 (LM) für iP2 – LM
    - BALVI mobil2 (Öko) für iP2 – Öko-Kontrolle
    - BALVI mobil2 (DM) für iP2 – Düngemittelüberwachung
- Neu: CMIS-Export (BY)
    - Erfordert die Aktivierung der ETL-Engine
- Neu: Export an Archiv-Systeme (BY)

Die meisten Schnittstellen können durch BALVI dynamisch aktualisiert und erweitert werden, ohne dass dabei der Betrieb von BALVI iP oder des BSS beeinflusst wird. In einzelnen Fällen muss der BSS nach der Aktualisierung bzw. Bereitstellung einer neuen Schnittstelle neugestartet werden.

1.2 IP-Patches für BALVI iP1

Da der BSS mit dem Profil „ip“ als Erweiterung für BALVI iP reagiert, gibt es zusätzlich zur Web-Anwendung BALVI iP Patches, die in die BALVI iP-Datenbank eingespielt werden müssen, um die Strukturen für den BSS zu schaffen, neuen Schnittstellen zu installieren/aktualisieren oder zusätzlich erforderliche Plugins zu ergänzen.

Für das Update des BSS auf Version 3.5.0 müssen folgende Grundvoraussetzungen erfüllt sein:

- BALVI iP 1.24.11 (oder höher)
- Java 8 (Oracle Java (Lizenzpflichtig) oder ein zertifiziertes Open JDK 1.8)
- Java 10 oder höher wird noch nicht unterstützt
- Apache Tomcat 8 (8.0.30 oder höher) oder Apache Tomcat 9 (9.0.4 oder höher)

Sollten Sie noch eine ältere Version von BALVI iP einsetzen und aus organisatorischen Gründen nicht auf Version 1.24.13 oder höher aktualisieren können, ist der Einsatz des BSS 3.5.0 nicht möglich.

1.3 Wichtige Hinweise für die Migration auf BSS 3.5

Da die meisten Kunden von Version 3.2.x auf 3.5.x migrieren und somit einige Schritte überspringen, ist es besonders wichtig, dass Sie die Neuerungen von BSS 3.3 zur Kenntnis nehmen, um nach dem Update des BSS auf die neue Oberfläche zugreifen zu können:

- Sie MÜSSEN in BALVI iP1 in der iP2-Rollenverwaltung neue Rollen konfigurieren und den gewünschten Benutzern in der Mandanten- und Benutzerverwaltung zuweisen. Die implizite Berechtigung für den ADMIN** wurde entfernt. Ihnen steht damit nun frei, jedem gewünschten iP1-Benutzer über die Rollenzuweisung Rechte an Funktionen des BSS zu vergeben.
- Die vorhandenen Konfigurationsdatei „[Context].xml“ kann weitergenutzt werden, es sollten jedoch fehlende Einstellungen aus den neuen Releases übernommen und aktiviert werden.
- Der Job für die Schnittstelle XGewerbeanzeige hat einen bekannten Fehler mit dem BSS 3.5.0. Installieren Sie zunächst den Hotfix „B3“[1], bevor Sie den Job XGewerbeanzeige wieder starten. Hinweise zu den bekannten Problemen finden Sie hier: README xGewerbe B3 Hotfix
- Prüfen Sie die vorhandenen und neuen JAR-Dateien in allen Ordern und entfernen Sie die alten Versionen, um Probleme zu vermeiden. Das gilt speziell für die JAR-Dateien in den Ordnern
    - kettleplugins
    - endorsed
    - lib (ojdbc6.jar ist nicht kompatibel mit Oracle 12c R2)

  Wir empfehlen, die alten Dateien komplett zu entfernen und durch die neuen Versionen zu ersetzen, die in dem aktuellen Auslieferungspaket des BSS enthalten sind.

[1] Verfügbar seit 06.12.2018 unter
http://download.balvi.de/public/Schnittstellenserver/xGewerbe_13_fuer_IP_12413_12414_B3_Hotfix.zip

Die Hinweise zur Migration von Versionen als BSS 3.2 auf BSS 3.5 wurden entfernt. Der Auslieferung liegt mit der Datei „conf/Catalina/localhost/bss.xml.example eine vollständige und mit Kommentaren versehene Beispiel-Konfiguration zum Vergleich bei.

1.4 Technische Änderungen in Version 3.5

- Der BSS unterstützt nun den Versand von E-Mails.
  Ergänzen Sie bitte die entsprechenden Einstellungen in der Konfiguration.
- Der in Version 3.3 eingeführte ehCache für das Caching von hohen Mengen an Datenrequests durch BALVI mobil2 (z. B. Voll- bzw. Erstabgleich) wurde auf eine JDBC-Pool-kompatible Datasource umgestellt.
- Der neue Konfigurationsparameter „bssProfil“ ermöglicht nun die Umstellung zwischen „iP“ (Standard) und „Standalone“.
- Der in BSS 3.3 eingeführte ehCache benötigt ab Version 3.5 einen Pool-Datasource namens „jdbc/xmlgenCacheDataSource“, die der Datei „context.xml“ angegeben werden muss, wenn Sie BALVI mobil2 für iP2 verwenden.
  Hinweise finden Sie im Kapitel „Einstellungen für iP2 – Synchronisation“.

Folgende Datenquellen sind ab BSS 3.5 in der Konfiguration erforderlich:

1.) jdbc/Datasource
   ➔ vorhanden, muss ggf. auf Apache Tomcat JDBC Connection Pool umgestellt werden
   ➔ Verweis auf die Daten von BALVI iP1 (Profil „iP“) für BALVI mobil (XT), XGewerbeanzeige, … erforderlich.
   ➔ Im Standalone-Modus muss hier eine „H2-Datenbank“ angegeben werden.
2.) jdbc/KettleIP1DataSource
   ➔ vorhanden seit BSS 3.1, jedoch meist auskommentiert
   ➔ Verweis auf die Daten von BALVI iP1 für ETL-Prozessverarbeitung
3.) jdbc/xmlgenCacheDataSource
   ➔ neu ab BSS 3.5
   ➔ ehCache-Datenquelle für das Caching von Daten für mobil iP2

In Planung ist zudem die Trennung von „BSS-internen Tabellen“ und Zugriff auf Daten, um auch unter Last sicherstellen zu können, dass die Betriebsfähigkeit des BSS nicht eingeschränkt wird.

1.5 Technische Änderungen in Version 3.4

Version 3.4 wurde nie für BALVI iP veröffentlicht, diese Version wurde ausschließlich für einen Kunden bereitgestellt, der kein BALVI iP verwendet.

- Der Standalone-Modus (ohne Verwendung von BALVI iP1) kann nun mit der Datenbank „H2“ genutzt werden. Dieser Modus verwendet den Apache Tomcat-UserRealm zur Authentifizierung.
- Zur Abfrage der Steuer-Tabellen des BSS wurden DB-Werkzeuge ergänzt.
- Der BSS hat nun eine Funktion, um die Job-, Import- und Export-Funktionen ohne BALVI iP zu aktualisieren. Dazu kann eine ZIP-Datei hochgeladen werden.

1.6 Technische Änderungen in Version 3.3

- Neues Rollenkonzept
  Ab Version 3.3 wurde der BSS auf das iP2-Rollenkonzept umgestellt. Daher sind nun implizite Berechtigungen aus BALVI iP wie „der SYSTEM/ADMIN** darf alles“ nicht mehr gegeben. Das Update des BSS auf Version 3.3 setzt daher zusätzlich voraus, dass für die Anwender, die auf die Oberfläche des BSS zugreifen wollen, neue iP2-Rollen angelegt werden.
  Die erforderlichen Objekte zur Pflege der Rollen sind ab BALVI iP 1.24.11 enthalten.
- Bitte beachten Sie beim Update auf Version 3.3 die Einrichtungsschritte, die im Benutzerhandbuch unter „iP2-Rollen für den BSS 3 einrichten“ beschrieben sind.
- Ohne diese Schritte ist der Login im BSS nach dem Prinzip „Security by Design“nur mit eingeschränkten Rechten möglich.

Verzeichnis „endorsed“ erforderlich

Tab. 1: Diverse XML-Parser, XSLT-Transformationen und interne XSD-Validierungen basieren auf der Umsetzung von Apache Xerces und Umsetzung von Apache Xerces und Apache Xalan. Deshalb müssen Sie für den BSS die entsprechenden JAR-Dateien bereitstellen. Da diese entsprechenden JAR-Dateien bereitstellen. Da diese Bibliotheken bereits beim Update der Schnittstelle XGewerbeanzeige installiert wurden, Schnittstelle XGewerbeanzeige installiert wurden, ist in vielen Fällen keine zusätzliche Anpassung erforderlich. Sollten Sie jedoch bislang erforderlich. Sollten Sie jedoch bislang einen BSS ohne die Schnittstelle XGewerbeanzeige betreiben, beachten Sie die erforderlichen betreiben, beachten Sie die erforderlichen Anpassungen in Kapitel Tab. 1: Bibliotheken für Apache Tomcat

- Bibliotheken im Verzeichnis „endorsed“.
- Bislang ist die Einrichtung der XML-Parser über den Weg „endorsed[2]“ erforderlich. Die Funktionalität „endorsed“ ist jedoch mit Java 9 entfernt worden. Wir arbeiten daran, die erforderlichen Bibliotheken zu reorganisieren und das Verzeichnis zu entfernen, um die Kompatibilität mit Java 10 oder höher herzustellen. Geplant ist eine Umstellung zum Release 3.6 des BSS in 2019.

Update der ETL Engine Kettle auf Version 7.1

- Die Open-Source ETL-Engine Pentaho Kettle[3] wird von komplexen Jobs genutzt. Kettle benötigt diverse Dateien, welche Sie lokal bereitstellen müssen. Diese Dateien sind nicht in der WAR-Datei enthalten.
- Die Dateien im Ordner „kettleplugins“ wurden mit dem Update auf Version 7.1 geändert. Um Probleme nach dem Update zu vermeiden, löschen Sie auf Ihrem Server den vorhandenen Ordner „kettleplugins“ und kopieren Sie den gleichnamigen Ordner (aktueller Stand) aus diesem Paket (siehe Ordner „tomcat“) auf den Server.

ehCache zur Optimierung von Zugriffen auf iP2

[2] https://tomcat.apache.org/tomcat-8.0-doc/class-loader-howto.html#XML_Parsers_and_Java

[3] https://sourceforge.net/projects/pentaho/

- Die Interaktion zwischen BALVI mobil2 und den Fachmodulen wird ebenfalls über den BSS verwaltet. Der BSS dient dabei als Fach-Funktions-Proxy und bereitet die Daten für BALVI mobil2 auf.
- Die Interaktion erfolgt über das REST (HTTP)-Protokoll und ist technisch komplexer und daher auch langsamer als der direkte Zugriff auf eine Datenbank.
- Daher wurde die OpenSource-Lösung „ehCache“ als Zwischenspeicher für die einzelnen Requests in den BSS integriert. Dadurch wird die REST-Anfrage-Last auf die iP2-Module deutlich reduziert.

BALVI mobil2 mit LDAP-Authentifizierung

- Die LDAP-Authentifizierung, die schon in iP1 mit BALVI mobil1.x verfügbar war, kann auch für BALVI mobil2.x und die iP2-Module genutzt werden. Dazu müssen Sie jedoch zusätzlich in den Umgebungseinstellungen der iP2 Fachmodule die LDAP-Authentifizierung über REST aktivieren.

Hinweis zur Konfiguration (nur beim Einsatz von BALVI iP2 mit BALVI mobil2):

Ergänzen Sie im BSS die (internen) URLs, die für die Anwendung BALVI mobil2 (LM) erforderlich sind. Weitere Anpassungen sind hier nicht nötig.
 	
`<!-- Beispiel: Lebensmittel-Modul iP2 -->`
`<Environment`
`  name="ip2urls/lmModul"`
`  value="https://lm-test.intern.balvi.de/"`
`  type="java.lang.String" override="true"`
`/>`
 	
Aktivieren Sie im iP2-Fachmodul in der Datei „local.app.properties“ die entsprechende Einstellung:
 	
`#LDAP-Einstellungen`
`# Zugriff auf REST-Schnittstelle über LDAP`
`cuba.rest.ldap.enabled = true`

Menüpunkt „ETL-Tasks“

- Die Menüleiste des BSS wurde dazu um den neuen Menüpunkt „ETL-Tasks“ erweitert. Er wird benötigt für die Jobs zur Synchronisation zwischen BALVI iP1 und BALVI iP2.
- Um den Menüpunkt zu aktivieren, ergänzen Sie in der Konfiguration folgenden Parameter:

Hinweis zur Konfiguration:

Ergänzen Sie im BSS die Einstellung bzw. stellen Sie sicher, dass dem Attribut „value“ dieses Elements „Environment“ der Wert “true“ zugewiesen wird und dass dieses Element nicht auskommentiert ist:
 	
`<!--ETL spezifische Erweiterungen ab BSS 3.1 -->`
`<Environment`
`  name="isEtlTaskViewEnabled"`
`  value="true"`
`  type="java.lang.Boolean" override="true"`
`/>`

Weitere neue Parameter (Optional)

- Die folgenden Parameter sind mit Default-Einstellungen versehen und müssen daher nicht zwingend in Ihre Konfiguration übernommen werden.
- Sie wurden ergänzt, damit im Datenabgleich Performance-Tuning möglich ist:

```
<!-- Timeout-Einstellungen -->
<!-- Wartezeit in Sekunden bei Requests
(von BALVI voreingestellter Wert: 60) -->
<Environment
   name="xmlgen/restRequestTimeoutSeconds"
   value="60"
   type="java.lang.Integer" override="true"
/>

<!-- Anzahl der angefragten Ergebnisdatensätze pro Request
(von BALVI voreingestellter Wert: 20) -->
<Environment
   name="xmlgen/maxEntriesPerRestRequest"
   value="20"
   type="java.lang.Integer" override="true"
/>
```

Download von LOG-Dateien vorkonfigurieren

- Die Ausgabe der Java-Protokolle und deren Anzeige in der Oberfläche wurden überarbeitet.
- Die Dateierweiterungen (Extensions), die beim Export der Protokoll-Dateien in den Download eingepackt werden sollen, können mit dem folgenden Parameter vorbelegt werden:

```
<!-- Neu ab BSS 3.2 auf Wunsch einiger Kunden (OPTIONAL) -->
<!-- Welche Dateierweiterungen sollen als Default
   für LOG-Dateien verwendet werden. (Standard "log txt") -->
<Environment
   name="extensionsOfLogFiles"
   value="log txt out gz"
   type="java.lang.String" override="true"
/>
```

2 Technische Einrichtungshinweise

2.1 Allgemeine Hinweise zum Betrieb

2.1.1 Kompatibilität zu Apache Tomcat und Java

Der BALVI Schnittstellenserver wurde getestet und freigegeben für Apache Tomcat 8 (ab Version 8.0.30 oder neuer) und Apache Tomcat 9. Zudem wurde die aktuellste Version von Java 8 getestet und kann verwendet werden (Java 8u182 und neuer).

Installieren Sie stets die aktuellsten Sicherheitsupdates für Java. Beachten Sie dabei jedoch, dass die Aktualisierung auf eine neuere Major-Version (z. B. Java SE 10) Probleme auslösen kann.

Freigegebene Major-Versionen werden daher explizit von BALVI benannt. Falls Sie beabsichtigen, die Major-Version zu ändern, prüfen Sie bitte die aktuellen Release Notes oder kontaktieren die BALVI Kundenbetreuung.

2.1.2 Oracle Java SE Lizenzierung

Die Verwendung der Java SE von Oracle ist ab Januar 2019 für kommerzielle Nutzung lizenzpflichtig. Die von BALVI gelieferte Software ist jedoch frei von den kommerziellen Teilen der Oracle Java SE und daher generell mit OpenJDK nutzbar. Die gängigen Linux-Distributionen bringen das OpenJDK als installierbares Paket mit. Unsere Testumgebungen mit Red Hat Enterprise Linux 7 oder Ubuntu LTS werden ebenfalls mit dem OpenJDK betrieben.

Bitte haben Sie Verständnis dafür, dass wir keine Funktionstests für jedes OpenJDK-Derivat durchführen können und dass es derzeit offiziell keine Version des OpenJDK für Windows Betriebssysteme gibt.

2.1.3 Spracheinstellung Unicode und Deutsch

Es wurde festgestellt, dass bei einigen Linux-Versionen die Zeichensatzeinstellung beim Starten von Apache Tomcat nicht auf eine Unicode-Umgebung verweist.

Das kann technische Probleme verursachen. Unter Linux sollten die Einstellungen (Umgebungsvariable) entsprechend auf
LANG=de_DE.UTF-8
LC_ALL=de_DE.UTF-8

eingestellt sein. Die Sprachangabe de_DE ist dabei nicht zwingend erforderlich, der Einsatz von UTF-8 als Zeichensatz jedoch schon. Zusätzlich sollte das „file.encoding“ entsprechend gesetzt werden.

2.1.4 Einsatz von TLS/SSL

BALVI hat keine spezifischen Anforderungen an die Konfiguration von SSL bzw. TLS. Der Einsatz von TLS 1.2 gemäß der technischen Richtlinie TR-02102-2 des BSI ist möglich, wenn Sie sich an die Dokumentation von Apache Tomcat halten. Es sind keine Anpassungen des Schnittstellenservers

erforderlich. Um unsichere Protokolle zu deaktivieren, ist die Anpassung der „server.xml“ erforderlich. Rahmenbedingung ist, dass Sie den Server mit dem "SSL-Port" (secure="true") verwenden.

Beispiel-Konfiguration des Connectors in der „server.xml“

```
<!-- Define a SSL HTTP/1.1 Connector on port 8443
     This connector uses the JSSE configuration, when using APR, the
     connector should be using the OpenSSL style configuration
     described in the APR documentation -->
<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
    maxThreads="150" scheme="https" secure="true"
    keystoreFile="/path/to/java_keystore"
    keystorePass="changeit"
    sslProtocol="TLSv1.2"
    sslEnabledProtocols="TLSv1.2"
    ciphers="TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256,
             TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,
             TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384,
             TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,
             TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256,
             TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,
             TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384,
             TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,
             TLS_DHE_DSS_WITH_AES_128_CBC_SHA256,
             TLS_DHE_DSS_WITH_AES_128_GCM_SHA256,
             TLS_DHE_DSS_WITH_AES_256_CBC_SHA256,
             TLS_DHE_DSS_WITH_AES_256_GCM_SHA384,
             TLS_DHE_RSA_WITH_AES_128_CBC_SHA256,
             TLS_DHE_RSA_WITH_AES_128_GCM_SHA256,
             TLS_DHE_RSA_WITH_AES_256_CBC_SHA256,
             TLS_DHE_RSA_WITH_AES_256_GCM_SHA384"
/>
```

Ablauf:

- Stellen Sie sicher, dass Apache Tomcat angehalten wurde, bevor Sie die „server.xml“ editieren
- Nutzen Sie einen Text-Editor, der UTF-8-kompatibel ist
- Öffnen Sie die „server.xml“ mit dem Editor.
- Finden Sie die Connector-Konfiguration für Ihren SSL-Port.
- Ergänzen Sie gemäß des oben angegebenen Beispiels die Angaben "sslProtocol", "sslEnabledProtocols" und "ciphers".

2.1.5 Umgebungsvariablen in Java setzen „-D[variable]=wert“

Viele unterschiedliche Anleitungen für Schnittstellen erwarten, dass ggf. Umgebungsvariablen gesetzt werden sollen. Dafür gibt es in Tomcat mehrere Wege, die von Ihrer eingesetzten Umgebung abhängig sind.

2.1.5.1 Empfohlene Standard-Einstellungen für den BSS

Einstellungen für Datei-Encoding und Spracheinstellungen:

Java arbeitet normalerweise im Modus UTF-8, das gilt speziell für die Verarbeitung von XML-Dateien. Es wurde jedoch festgestellt, dass in einigen Umgebungen die Verarbeitung von XML-Dateien Probleme mit deutschen Umlauten hatte. Ursache ist die interne Verwendung des „Default Charset“, der unter Linux ggf. „POSIX“ ist und damit das file.encoding='ANSI_X3.4-1968' in der VM eingestellt hat. Um das Problem zu umgehen, sollte immer folgendes eingestellt sein:

```
file.encoding=UTF-8
```

Einstellungen, um den Output von Pentaho Kettle[4] zu reduzieren.

```
KETTLE_DISABLE_CONSOLE_LOGGING=Y
KETTLE_HOME=${CATALINA_BASE}/work
```

2.1.5.2 Linux mit dem BALVI-Paket

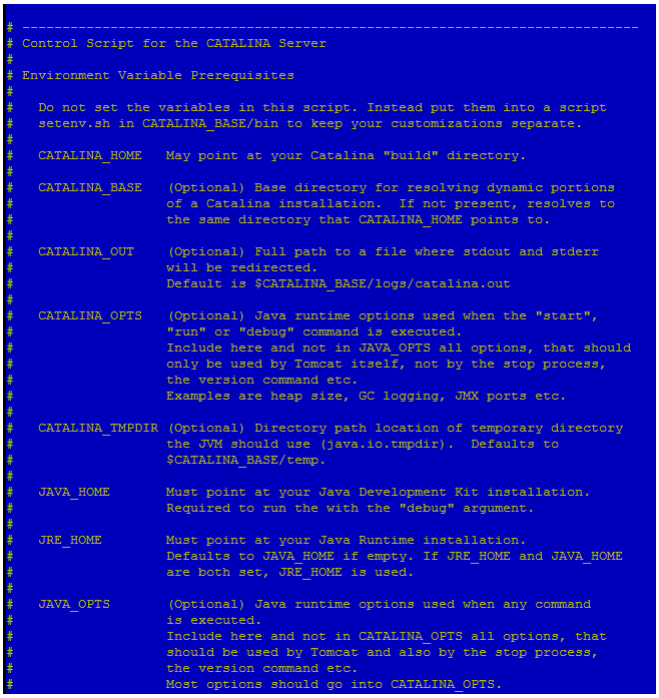
BALVI hat als Wrapper für das Starten der Tomcats im Userkontext die Logik von SuSE-Linux übernommen. Dabei gibt es eine „tomcat8.conf[5]“ im Ordner „conf“ Ihres Tomcat-Base-Verzeichnisses. Diese Datei enthält diverse Beispiel-Einträge mit deutschen Kommentaren.

```
# Endorsed-Dir für CATALINA_BASE vor CATALINA_HOME setzen
export JAVA_ENDORSED_DIRS="${CATALINA_BASE}/endorsed"

#Sicherstellen, dass File-Ecoding korrekt eingestellt wurde
CATALINA_OPTS="${CATALINA_OPTS} -Dfile.encoding=UTF-8"

CATALINA_OPTS="${CATALINA_OPTS} -DKETTLE_DISABLE_CONSOLE_LOGGING=Y"
CATALINA_OPTS="${CATALINA_OPTS} -DKETTLE_HOME=${CATALINA_BASE}/work"
```

Hier sollten die Standard-Umgebungsvariablen von Apache Tomcat eingestellt werden. Welche Umgebungsvariablen durch Apache Tomcat ausgewertet werden und welche Bedeutung diese haben, kann im Start-Skript ${CATALINA_HOME}/bin/catalina.sh nachgelesen werden:

[4] https://github.com/pentaho/pentaho-kettle/blob/6.1.0.1-R/engine/src/kettle-variables.xml

[5] Die Version des Tomcat ist im Namen enthalten, also z.B. tomcat8.conf bzw. tomcat9.conf.



```
# -----------------------------------------------------------------------------
# Control Script for the CATALINA Server
#
# Environment Variable Prerequisites
#
#   Do not set the variables in this script. Instead put them into a script
#   setenv.sh in CATALINA_BASE/bin to keep your customizations separate.
#
#   CATALINA_HOME   May point at your Catalina "build" directory.
#
#   CATALINA_BASE   (Optional) Base directory for resolving dynamic portions
#                   of a Catalina installation.  If not present, resolves to
#                   the same directory that CATALINA_HOME points to.
#
#   CATALINA_OUT    (Optional) Full path to a file where stdout and stderr
#                   will be redirected.
#                   Default is $CATALINA_BASE/logs/catalina.out
#
#   CATALINA_OPTS   (Optional) Java runtime options used when the "start",
#                   "run" or "debug" command is executed.
#                   Include here and not in JAVA_OPTS all options, that should
#                   only be used by Tomcat itself, not by the stop process,
#                   the version command etc.
#                   Examples are heap size, GC logging, JMX ports etc.
#
#   CATALINA_TMPDIR (Optional) Directory path location of temporary directory
#                   the JVM should use (java.io.tmpdir).  Defaults to
#                   $CATALINA_BASE/temp.
#
#   JAVA_HOME       Must point at your Java Development Kit installation.
#                   Required to run the with the "debug" argument.
#
#   JRE_HOME        Must point at your Java Runtime installation.
#                   Defaults to JAVA_HOME if empty. If JRE_HOME and JAVA_HOME
#                   are both set, JRE_HOME is used.
#
#   JAVA_OPTS       (Optional) Java runtime options used when any command
#                   is executed.
#                   Include here and not in CATALINA_OPTS all options, that
#                   should be used by Tomcat and also by the stop process,
#                   the version command etc.
#                   Most options should go into CATALINA_OPTS.
```

Abb. 1: Auszug aus der Datei „catalina.sh“

2.1.5.3 Linux mit „setenv.sh“

Die „setenv.sh“ ist der von Apache Tomcat definierte Standard-Weg, um Umgebungsvariablen zu setzen.

Die Datei „setenv.sh“ finden Sie im Ordner „bin“ Ihres Tomcat-Base-Verzeichnisses. Sollte die Datei nicht vorhanden sein, muss sie angelegt werden. Es sind keine Vorgaben oder Beispiele vorhanden.

Ergänzen Sie neue Variablen dort durch die Erweiterung der „CATALINA_OPTS“. Die Syntax ist jedoch identisch mit der in der Datei „tomcat8.conf“.

```
#Sicherstellen, dass File-Ecoding korrekt eingestellt wurde
CATALINA_OPTS="${CATALINA_OPTS} -Dfile.encoding=UTF-8"
CATALINA_OPTS="${CATALINA_OPTS} -DKETTLE_DISABLE_CONSOLE_LOGGING=Y"
CATALINA_OPTS="${CATALINA_OPTS} -DKETTLE_HOME=${CATALINA_BASE}/work"
```

2.1.5.4 Windows Dienste-Konfiguration

Unter Windows wird Apache Tomcat in der Regel als Dienst registriert. Daher sind die Einstellungen in der Datei „setenv.bat“ nicht relevant, da Apache Tomcat nicht mittels der Datei „catalina.bat“ bzw. „startup.bat“ gestartet wird.

Stattdessen müssen Sie die Einstellungen über das im Ordner ${CATALINA_HOME}/bin befindliche Programm „tomcat8w.exe“ (als Administrator ausführen!) in die Windows-Dienste-Registry eintragen.

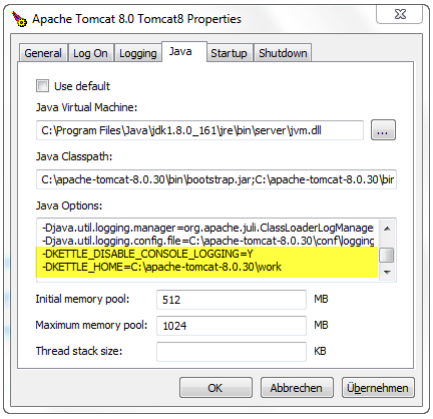


Abb. 2: Einstellungen mithilfe der Datei „tomcatXw.exe“ vornehmen

Achtung:

Die Einstellungen im Textfenster „Java Options:“ sind etwas tückisch.

Jeder Wert muss in einer separaten Zeile angegeben werden. Zudem sind Leer- oder Sonderzeichen am Ende einer Zeile problematisch. Prüfen Sie bitte daher genau, wenn Sie eine neue Zeile ergänzen.

2.1.6 Nutzung von Java-Proxyeinstellungen

Der BALVI Schnittstellenserver verwendet für ausgehende Verbindungen die Java-Net Einstellungen, welche in der Java-Dokumentation[6] beschrieben sind. Falls Sie Jobs einsetzen, welche ausgehenden Zugriff auf Adressen im Internet benötigen, kann es passieren, dass Sie z. B. eine „java.net.UnknownHostException“ im Protokoll finden, dann müssen Sie Ihre Proxy-Umgebung in den Java-Umgebungseinstellungen ergänzen.

Die folgenden Parameter sollten daher in Ihrer Umgebung gesetzt werden, wenn der Zugriff auf das Internet nur über einen Proxy möglich ist:

- -DproxySet=true
- -DproxyHost=10.11.12.13
- -DproxyPort=8888
- -Dhttp.nonProxyHosts=*.domain.org|localhost|10.*

Starten Sie nach Änderung der Einstellungen den Server neu, damit die Einstellungen wirksam werden.

2.1.6.1 Proxy-Einstellung unter Windows (Tomcat-Dienst)

Verwenden Sie das Dienste-Konfigurationsprogramm (z. B. tomcat8w.exe) und ergänzen Sie die fehlenden Einstellungen.

[6] http://docs.oracle.com/javase/8/docs/technotes/guides/net/properties.html

2.1.6.2 Proxy-Einstellung unter Linux

Unter Linux wird in der Konfigurationsdatei (z. B. conf/tomcatX.conf) die Variable „${JAVA_OPTS}“ entsprechend erweitert und die Werte übergeben. Beachten Sie dabei, dass die Variablen von Linux (bzw. der Bash) gelesen werden. Daher muss das „Pipe“-Zeichen „|“ mit „\“ maskiert werden, ansonsten erhalten Sie bei der Verarbeitung der Umgebungsvariable „http.nonProxyHosts“ Fehlermeldungen in der Linux-Konsole, welche dazu führen, dass Java nicht korrekt konfiguriert werden kann.

```
JAVA_OPTS="${JAVA_OPTS} -DproxySet=true -DproxyHost=10.11.12.13 -DproxyPort=8888"
JAVA_OPTS="${JAVA_OPTS} -Dhttp.nonProxyHosts=*.domain.org\|localhost\|10.*"
```

2.1.7 Datenbankverbindung

Wenn Sie zwischen dem Oracle-Server und dem Database Connection Pool (DBCP) eine Firewall verwenden, sollte unbedingt die „validationQuery“ in der DBCP-Konfiguration aktiviert werden, damit die Anwendung schneller freie Connections aus dem Pool erhalten kann. Die Dokumentation zu den Einstellmöglichkeiten finden Sie in der beiliegenden Datei „bss.xml.example“.

Bitte beachten Sie, dass dringend empfohlen wird, die „Factory“ auf "org.apache.tomcat.jdbc.pool.DataSourceFactory" anzupassen, falls sie noch auf "org.apache.commons.dbcp.BasicDataSourceFactory" eingestellt ist.

Die Beispielkonfiguration hat entsprechende Voreinstellungen.

2.1.8 Das Konzept „Context“

Apache Tomcat bezeichnet Web-Anwendungen als Kontext. Um eine Web-Anwendung bzw. einen Kontext innerhalb von Apache Tomcat zu konfigurieren, ist eine Konfigurationsdatei bzw. ein Kontextdeskriptor erforderlich. Ein Kontextdeskriptor ist einfach eine XML-Datei, die eine Tomcat-bezogene Konfiguration für einen Kontext enthält. Es gibt zwei Ordner, in denen Kontextdeskriptoren abgelegt werden können:

Unterhalb vom Tomcat-Konfigurationsordner „conf“

(1) $CATALINA_BASE/conf/[Servicename]/[Hostname]/[Web-Anwendungsname].xml

z. B. $CATALINA_BASE/conf/Catalina/localhost/ROOT.xml

oder $CATALINA_BASE/conf/Catalina/localhost/bss.xml

Innerhalb einer Web-Anwendung (unterhalb von META-INF)

(2) $CATALINA_BASE/webapps/[Web-Anwendungsname]/META-INF/context.xml

z. B. $CATALINA_BASE/webapps/ROOT/META-INF/context.xml

oder $CATALINA_BASE/webapps/bss/META-INF/context.xml

Konfigurationsdateien in (1) haben den Namen „[Web-Anwendungsname].xml“, während Konfigurations-dateien in (2) den Namen „context.xml“ haben. Wenn keine Konfigurationsdatei (kein Kontextdeskriptor) für eine Web-Anwendung (einen Kontext) bereitgestellt wird, konfiguriert Tomcat die Web-Anwendung unter Verwendung von Standardwerten.

Achtung:

Wenn Sie in der Datei „server.xml“ die Einstellung
`<Host name="localhost" appBase="webapps" unpackWARs="true" autoDeploy="true">`
verwenden und in einer WAR-Datei eine „context.xml“ in (2) vorhanden ist, wird diese bei jedem Deployment entpackt und überschreibt die in (1) ggf. manuell angepasste Datei. Daher wird dringend empfohlen, die Einstellung auf
`<Host name="localhost" appBase="webapps" unpackWARs="true" autoDeploy="false">`
abzuändern.

UMBENENNUNG DER GELIEFERTEN WAR-DATEI

Die im Paket von BALVI gelieferte Webanwendung kann jederzeit umbenannt werden (z. B. in ROOT.war), ohne dass dabei spezielle Einstellungen in der Konfiguration erforderlich sind.

Wählen Sie „ROOT“ als Kontextnamen, können Sie die zugehörige Web-Anwendung auf Ihrem Server unter der Default-URL „/“ (Slash), also „http(s)://IhreSubdomain.IhreDomain.TLD:port/“ aufrufen, wählen Sie beispielsweise „bss“ als Kontextnamen, lautet die URL „http://IhreSubdomain.IhreDomain.TLD:port/bss“.

Beachten Sie dabei, dass zu einer Konfigurationsdatei immer eine Datei mit demselben Namen im Ordner „webapps“ existieren muss, z. B. /webapps/bss.war und „/conf/Catalina/localhost/bss.xml“.

Der vergebene „Kontext-Name“ ist dabei auch ausschlaggebend für die Zugriffs-Adresse (den „application-context“), die der Apache Tomcat dabei bereitstellt. Ein WAR-Datei mit dem Namen „bss.war“ wird folgendermaßen aufgerufen: http://IhreSubdomain.IhreDomain.TLD:port/bss

Es wird empfohlen, den Namen der WAR-Datei immer in Kleinbuchstaben anzugeben, da Webserver im Normalfall „case-sensitive“ auf die Angabe der URL reagieren.

Details zu Kontexten, Umbenennungen und Auswirkungen entnehmen Sie bitte der Tomcat-Dokumentation[7].

2.1.9 Server-Schnittstellen und Jobs einspielen

Die Erweiterung der Export- und Importschnittstellen oder Jobs, die über den Schnittstellenserver bereitgestellt werden, ist unabhängig vom Schnittstellenserver selbst. Die Schnittstellen sind daher nicht Bestandteile des Installationspakets für den BALVI Schnittstellenserver, sie werden je Verfahren separat ausgeliefert. Daher ist es für die Bereitstellung neuer Schnittstellen erforderlich, zusätzliche Inhalte zu ergänzen, z. B. über die Updape.exe für BALVI iP1 oder über die neue im BSS integrierte Transportlogik.

Ggf. müssen nach dem Bereitstellen einer neuen Schnittstelle die GRANT-Skripte neu ausgeführt werden. (Vgl. „Wann ist die Ausführung des GRANT-Skripts erforderlich“).

[7] https://tomcat.apache.org/tomcat-8.0-doc/config/context.html#Naming

2.2 Kompatibilitätsliste

Diverse Verfahren setzen eine Mindestversion des BALVI Schnittstellenservers voraus. Die Schnittstellen sind so konzipiert, dass eine Aktualisierung auf eine höhere Version des BSS ohne Anpassung an den vorhandenen Schnittstellen jederzeit möglich ist.

Derzeit bestehen folgende Mindestanforderungen:

1. BALVI mobil XT 1.7.0 oder höher setzt BSS 2.6.5 und BALVI iP 1.24.x voraus.
2. BALVI mobil XT 1.8.3 oder höher setzt BSS 2.8.1 und BALVI iP 1.24.x voraus.
3. BALVI mobil 2 (LM) setzt BSS 3.3.2 und BALVI iP 1.24.11 voraus.
4. Die Schnittstelle eFI setzt BSS 2.5.1 und BALVI iP 1.22c voraus.
5. Die Schnittstelle LIMS setzt BSS 2.6.5 voraus.
6. Die Schnittstelle TSK (SN) setzt BSS 2.6.5 voraus.
7. Die Schnittstelle xdomea (SN) setzt BSS 2.6.5 und Java 7 voraus.
8. Die Schnittstelle XGewerbeanzeige 1.2 (ab Paket 7, Ausbaustufe 3) setzt BSS 3.0.2, BALVI iP 1.24.x, Java 8 mit der „Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy“ voraus.
9. Die Schnittstelle RioPP LM (NRW) setzt BSS 3.0.3 und die ETL-Konfiguration voraus.
10. Die Schnittstellen zu BALVI iP2 (Fachmodule BTS, LM, zDA Stand Jan. 2018) setzen BSS 3.3.2 voraus.
11. Die Schnittstelle XGewerbeanzeige 1.3(B2) setzt BSS 3.2.1, BALVI iP 1.24.11/1.24.12, Java 8 mit der „Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy[8]“ voraus.
    ACHTUNG: die Version „B2“ ist nicht mehr mit dem BSS 3.5.x oder höher kompatibel.
12. Die Schnittstelle XGewerbeanzeige 1.3(B3) setzt BSS 3.2.1, BALVI iP 1.24.13/1.24.14, Java 8 mit der „Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy“ voraus.
    Achtung: Beim Update auf BSS 3.5.x muss das Paket „Schnittstelle XGewerbeanzeige 1.3(B3)“ oder höher eingespielt sein, damit keine Ausführungsfehler auftreten.

[8] Die Einrichtung der „Unlimited Strength Jurisdiction Policy“ ist JDK-versionsabhängig. Entsprechende Hinweise zur Konfiguration sind in den Release Notes oder Rundmails zur Schnittstelle XGewerbeanzeige enthalten.

2.3 Erforderliche Bibliotheken für Apache Tomcat

2.3.1 Dateien im Ordner „lib“

Dateien im Ordner „lib“ werden über den Tomcat-Classloader geliefert. Grundsätzlich sollten die Dateien im Ordner „{CATALINA_BASE}\lib“ liegen, nicht im Ordner „{CATALINA_HOME}\lib“, damit die unterschiedlichen Tomcat-Instanzen ihre eigenen Versionen der JAR-Dateien verwenden.

Es ist möglich, diese JAR-Dateien auch zentral abzulegen, jedoch ist der Classloader sehr empfindlich, wenn Dateien geändert oder gelöscht werden. Wenn Sie sich für eine zentrale Ablage der JAR-Dateien entscheiden, müssen alle Instanzen von Apache Tomcat neu gestartet werden, sobald eine JAR-Datei aktualisiert oder gelöscht wird.

Dateiname	Funktion	Quelle
~~ojdbc6.jar~~	Oracle 11gR2 JDBC Client ist nicht mit Oracle 12c kompatibel. Es wird ab sofort ojdbc7.jar vorausgesetzt.	
ojdbc7.jar	Oracle 12gc1 Clientpaket, Oracle JDBC Client für Java 1.8 ersetzt die bisherige Version „ojdbc6.jar“. Die alte Library muss auf jeden Fall entfernt werden.	https://www.oracle.com/technetwork/database/features/jdbc/jdbc-drivers-12c-download-1958347.html
log4j-1.2.17.jar (oder höher)	Logging-Funktionalität Apache Log4j	http://logging.apache.org/log4j/1.2/download.html
bcprov-jdk15on-1.59.jar (oder höher)	BouncyCastle Security Provider für SSL/AES, … wird ab Version 2.7 im Ordner „lib“ von Apache Tomcat erwartet. Sollte es aus Sicherheitsgründen erwünscht sein, eine aktuellere Version einzusetzen, ist dies i.d.R. möglich. Der Version 1.59 (Jan. 2018) enthält ein Sicherheitsupdate für TLS (**vulnerability described in CVE-2017-13098 has been fixed**) und sollte dringend ausgetauscht werden.	http://mvnrepository.com/artifact/org.bouncycastle/bcprov-jdk15on/1.59
kettledatasource-1.0.0.jar	Von BALVI erstell für die Nutzung der Kettle ETL-Eingine.	BALVI
h2-1.4.197.jar	H2 Database Engine für die xmlGenCache-Funktionen	https://mvnrepository.com/artifact/com.h2database/h2/1.4.197

Tab. 2: Bibliotheken für Apache Tomcat

2.3.2 Bibliotheken im Verzeichnis „endorsed“

Das Verzeichnis „endorsed“[9] besitzt für Java eine spezielle Funktionalität. Nicht jede JAR-Datei kann im Verzeichnis „endorsed“ abgelegt werden. Zudem gilt die Anpassung für die gesamte Java Virtual Machine, also auch für andere Web-Application Archives (WARs), die in derselben Instanz von Apache Tomcat lauffähig sind.

Sie müssen daher entsprechend der Beschreibung von Apache Tomcat im Abschnitt „XML Parsers and Java[10]“ folgendermaßen vorgehen, um den XML-Parser Xerces verwenden zu können:

[9] https://docs.oracle.com/javase/8/docs/technotes/guides/standards/index.html

[10] https://tomcat.apache.org/tomcat-8.0-doc/class-loader-howto.html#XML_Parsers_and_Java

Tomcat erfordert XercesImpl > 2.9.1 und xml-apis = 1.4.01. Die beiden JAR-Dateien müssen im Ordner „endorsed“ abgelegt werden, damit diese korrekt funktionieren.

Dateiname	Funktion	Quelle
serializer-2.7.2.jar	XALAN XML Transformer (erforderliche Bibliothek) https://xalan.apache.org/xalan-j/	https://mvnrepository.com/artifact/xalan/serializer/2.7.2
xalan-2.7.2.jar	XALAN XML Transformer https://xalan.apache.org/xalan-j/	https://mvnrepository.com/artifact/xalan/xalan/2.7.2
xercesImpl-2.11.0.jar	XERCES XML-Parser http://xerces.apache.org/	https://mvnrepository.com/artifact/xerces/xercesImpl/2.11.0
xml-apis-1.4.01.jar	XML-API-Specification 1.4	https://mvnrepository.com/artifact/xml-apis/xml-apis/1.4.01

Tab. 3: Bibliotheken im Verzeichnis „endorsed“

Kopieren Sie den Ordner „endorsed“ in Ihr Tomcat-Verzeichnis. Die aufgeführten Bibliotheken werden durch Apache Tomcat beim Start in die Java-Runtime-Environment integriert.

Wenn Sie Apache Tomcat in der Standard-Konfiguration einsetzen und kein separates {CATALINA_BASE} für die Anwendung verwenden, sind keine weiteren Anpassungen erforderlich, der Ordner „endorsed“ wird beim Serverstart gefunden und automatisch von Apache Tomcat verwendet.

Falls Sie unter Linux die Logik verwenden, um CATALINA_HOME und CATALINA_BASE getrennt zu verwalten, gehen Sie bitte folgendermaßen vor:

- Kopieren Sie den Ordner „endorsed“ in das Tomcat-Verzeichnis im Home-Verzeichnis des Users (NICHT in den Ordner /opt/catalina/…)
  Überschreiben Sie die Variable {JAVA_ENDORSED_DIRS} und setzen Sie den korrekten Pfad. Fügen Sie dazu in der Datei „tomcat[7/8].conf” (bzw. catalinarc) nach der Variable „CATALINA_HOME=“ folgendes in Ihre Konfiguration ein:

```
# Endorsed-Dir für CATALINA_BASE vor CATALINA_HOME setzen
export JAVA_ENDORSED_DIRS="$CATALINA_BASE"/endorsed
#Default in "setclasspath.sh" war: #JAVA_ENDORSED_DIRS="$CATALINA_HOME"/endorsed
```

2.4 Datenbankverbindung konfigurieren

2.4.1 BALVI iP bzw. Profil „ip“

2.4.1.1 Einführung

Der BALVI Schnittstellenserver benötigt keine eigenen Tabellen, wenn er an eine BALVI iP1 Datenbank gekoppelt ist. Die Web-Anwendung kann auch ohne GRANT-Script direkt auf das Schema von BALVI iP zugreifen, indem der Benutzer und das Passwort des BALVI iP-Schemas in der Datei „context.xml“ eingetragen werden. Dies ist aus Sicherheitsgründen entsprechend den BSI-Empfehlungen nicht empfohlen, da der Schemabesitzer von BALVI iP auch DDL-Statements wie CREATE, ALTER und DROP TABLE ausführen kann.

Sollte dies aus Sicherheitsgründen nicht erwünscht sein (empfohlen vom BSI und von BALVI), finden Sie im Paket ein GRANT-Skript, um stark eingeschränkte Zugriffsrechte vom BALVI iP Schema für den Webservice-Schema-Benutzer einzurichten.

Zwischen dem Schema von BALVI iP und jeder Web-Anwendung ist dann eine Trennung möglich, um sicherzustellen, dass der Datenbank-Benutzer, der in der Tomcat-Konfiguration für die Web-Anwendung eingetragen wird, geändert oder gelöscht werden kann, ohne dass die Windows-Anwendung BALVI iP davon betroffen ist. Zudem vergibt das von BALVI gelieferte GRANT-Skript keine Rechte zum Löschen von Tabellenstrukturen oder Tabelleninhalten, so dass der Webservice nur festgelegte Daten ändern oder lesen darf. Wir empfehlen, für jede Web-Anwendung einen eigenen Datenbank-Benutzer zu vergeben, wenn Sie ein BALVI iP mit mehreren Webservices verbinden wollen.

Die Möglichkeiten im Überblick:

1. Sie betreiben den Webservice mit einem eigenständigen Oracle-Benutzer. In diesem Fall muss das GRANT-Skript ausgeführt werden, um den Webservice-Benutzer und den BALVI iP-Benutzer miteinander zu koppeln.
2. In der Tomcat-Konfiguration wird der Oracle-Benutzer eingetragen, welcher der BALVI iP Schemabesitzer ist und der auch in der Datei „balviip_conn.ini“ hinterlegt ist (NICHT empfohlen!). Der Webservice hat 100 % Zugriffsrechte (auch Lösch- oder Änderungsrechte am Schema), es wird kein GRANT-Skript benötigt.

Wenn Sie die Konfiguration 2.) wählen möchten, können Sie die folgenden Unterabschnitte überspringen und in Abschnitt 3 Konfiguration des Schnittstellenservers fortsetzen.

2.4.1.2 Erstinstallation: Datenbank-Benutzer anlegen

Wenn Sie den Webservice-Benutzer vom BALVI iP Schema trennen möchten, muss durch die DB-Administration in der Oracle-Server-Instanz, wo auch BALVI iP sein Schema besitzt, ein Benutzer (ggf. je Web-Anwendung) angelegt werden, welcher folgende Berechtigungen benötigt:

```
CONNECT
ALTER SESSION
CREATE SESSION
CREATE TABLE
CREATE VIEW
CREATE SEQUENCE
CREATE SYNONYM
QUERY REWRITE
```

Der Default-Tablespace ist frei wählbar. Der Benutzer braucht das Recht QUOTA UNLIMITED auf den Tablespace von BALVI iP (Default: BALVI_IP). Ein Beispiel-Create-Skript „oraCreate_WEB_User.sql“ liegt der Auslieferung bei.

2.4.1.3 Rechte zuweisen

Wenn der BALVI iP Schemabesitzer NICHT in der Tomcat-Umgebung eingetragen werden soll, muss ein Oracle DBA das beiliegende GRANT-Skript „**oraGrantIP_to_Web.sql**“ ausführen, um dem jeweiligen

Webservice (in Tomcat verwendeten Oracle-Benutzer) die minimalen Rechte auf das Schema von BALVI iP zu gewähren.

Dieses Skript setzt voraus, dass es von einem Oracle Administrator als „User SYSTEM“ ausgeführt wird. Lesen Sie die im Skript enthaltenen, erläuternden Kommentare und passen Sie die Benutzer (BALVI_IP_SCHEMABESITZER) und (BALVI_WEB_SCHEMABESITZER) an:

```
/* ================================================
      DEKLARATIVER TEIL!! MUSS vom KUNDEN geändert
      bzw. an die korrekten SCHEMA-Namen angepasst
      werden!
      Die USER-Namen müssen in Großbuchstaben
      geschrieben sein, sonst werden sie nicht
      gefunden!
   ================================================ */

:sUserBALVI_IP := 'BALVI_IP_SCHEMABESITZER';
:sUserBALVI_WEB := 'BALVI_WEB_SCHEMABESITZER';
```

Neu ab GRANT-Skript 1.58 ist die Umbenennung der Verfahrenszuordnung.

Stellen Sie sicher, dass die Einstellung für Ihr Bundesland/Verfahren korrekt ist, dort muss eine 1 für mindenstens ein Verfahren gesetzt sein. Diese Einstellung befindet sich direkt unter den Schemabesitzer-Angaben.

```
-- Verfahrensbezogene Konfiguration
-- Setzen Sie für das jeweilige Verfahren eine 1

-- BALVI Schnittstellenserver (BSS)
-- bis V2.66 auch "BALVI Kommunikationsserver (BKS)" genannt
-- generell für die meisten Verfahren gültig.
-- Ab Version 57 entfällt mit der Einführung der
-- Schnittstelle XGewerbeanzeige auch der separate Schalter
-- "WS_BTR", beide Verfahren wurden vereinheitlicht.
:WS_BSS := 1;
```

Führen Sie das Skript nach der Anpassung als Benutzer SYSTEM aus, um die Grants durchzuführen. Dies wurde mit dem von ORACLE gelieferten Programm SQLPLUS (Command-Line), SQLPLUS Worksheet (GUI) und Oracle SQL-Developer getestet.

Das Skript generiert für jedes Objekt ein SYNONYM für den WEB-USER, so dass dieser auf die Objekte ohne Angabe des Besitzers (Owners) des iP-Schemas zugreifen kann. Stellen Sie nach der Ausführung sicher, dass die SYNONYMS vorhanden und VALID sind, dies kann ggf. nicht der Fall sein. Das Skript kann jederzeit erneut ausgeführt werden, um die Grants neu zu erzeugen und bietet ebenfalls die Option, alle Grants und Synonyme wieder zu entfernen. Detaillierte Informationen zur Nutzung des GRANT-Skripts finden Sie in Form von Kommentaren im Skript selbst.

Hinweis:
Die Versionsnummer des GRANT-Skriptes finden Sie als Kommentar direkt im Skript.

```
/* Zugriffsberechtigungen für Webanwendungen auf BALVI iP anlegen
 * ============================================================
 *
 * Version: 1.69
 * Gültig für: BALVI iP 1.24 oder höher
 * Autor: Frank Holler
 *
 * SVN Revisionsinformationen:
 * $Revision: 888 $
 * $Date: 2018-01-03 16:50:36 +0100 (Mi, 03 Jan 2018) $
 *
 * ============================================================
 */
```

2.4.1.4 Wann ist die Ausführung des GRANT-Skripts erforderlich?

Durch das Struktur-Update von BALVI iP (z. B. 1.24.10 auf 1.24.11) werden veraltete Strukturen gelöscht und ggf. neue Views oder Tabellen angelegt. Die Grant-Rechte gehen dabei möglicherweise verloren. Daher ist die Ausführung des GRANT-Skripts immer dann erforderlich, wenn sich die Datenbank-Struktur von BALVI iP verändert.

Zudem kann es fachliche Neuerungen geben, welche im alten GRANT-Skript nicht vorhanden sind. Daher muss das Skript ggf. auch ausgeführt werden, wenn eine neue Schnittstelle im Schnittstellenserver bereitgestellt wurde. Dieses betrifft z. B. auch die Bereitstellung von neuen Versionen von BALVI mobil XT, da auch dort mit einem fachlich erweiterten Client die Möglichkeit besteht, dass neue Schnittstellen bereitgestellt und somit mehr Schreibrechte auf Tabellen benötigt werden.

2.4.1.5 Die richtige Version des GRANT-Skripts verwenden

Wie der Schnittstellenserver sind auch die GRANT-Skripte versioniert. Neuere Versionen des GRANT-Skriptes sind abwärts-kompatibel und können immer verwendet werden, auch wenn in einer Schnittstellenbeschreibung explizit eine ältere Version des GRANT-Skriptes benannt wurde.

2.4.1.6 Kombination von mehreren Verfahren

Wenn Sie zwei oder mehr Schnittstellen im BSS verwenden möchten, müssen Sie auch mehrere Verfahren auf aktiv („1“) zu setzen.

Im Beispiel nehmen wir an, dass Sie „eFi“ und „BALVI mobil XT“ im Schnittstellenserver bereitstellen möchten. Die Beschreibung von „eFi“ fordert, das GRANT-Skript V36 auszuführen, für BALVI mobil XT 1.8 ist jedoch schon V56 nötig.

Wenn Sie beide Verfahren betreiben, verwenden Sie GRANT-Skript (V56 oder höher) und aktivieren dort die Einstellung für das jeweilige Verfahren:

```
/* ================================================
      DEKLARATIVER TEIL!! MUSS vom KUNDEN geändert
      bzw. an die korrekten SCHEMA-Namen angepasst
      werden!
      Die USER-Namen müssen in Großbuchstaben
      geschrieben sein, sonst werden sie nicht
      gefunden!
   ================================================ */
```

```
:sUserBALVI_IP := 'BALVI_IP_SCHEMA';
:sUserBALVI_WEB := 'WEBAPP_SCHEMA';

-- Verfahrensbezogene Konfiguration
-- Setzen Sie für das jeweilige Verfahren eine 1

-- BALVI Schnittstellenserver (BSS)
-- bis V2.66 auch "BALVI Kommunikationsserver (BKS)" genannt
-- generell für die meisten Verfahren gültig.
-- Ab Version 57 entfällt mit der Einführung der
-- Schnittstelle XGewerbeanzeige auch der separate Schalter
-- "WS_BTR", beide Verfahren wurden vereinheitlicht.
:WS_BSS := 1;

-- WebService BTR (Betriebsstättenregister) (NRW, TH)
-- entfernt mit Update V57, setzen Sie stattdessen bei WS_BSS := 1
--:WS_BTR := 0;

-- Probenahme Futtermittel im Browser(BY)
:WS_PROBE_FM := 0;

-- Risikoorientierte Probenplanung Futtermittel - RIOPP FM (BY)
:WS_RIOPP_FM := 0;

-- Verfahren "Export an EFi" (Status Pilotbetrieb).
:WS_EFI := 1;
-- Ab Version 35 des GRANT-Skriptes ist es möglich, den Webuser
-- für das Löschen der Ausgangskörbe explizit zu berechtigen bzw. sogar
-- explizit einen abweichenden User festzulegen. Die Konfiguration in der
-- Job-Oberfläche sieht vor, dass man die Oracle-Datenbank-Zugangsdaten für
-- das Löschen einstellen muss. Hier muss im Normalfall derselbe Oracle
-- Datenbankbenutzer eingetragen werden, wie in der Tomcat-Verbindungs-
-- konfiguration. In diesem Fall sollte, wenn WS_EFI := 1 eingestellt wurde,
-- ebenfalls WS_EFI_DELETE := 1 eingestellt werden.
-- Alternativen:
-- 1.) Datenbankbenutzer ändern
--   Wenn Sie bei :sUserEFI_TRUNC anstatt :sUserBALVI_WEB (Standard) einen
--   anderen Oracle-Benutzernamen angeben, wird das Löschrecht für die EFi
--   Ausgangskörbe datenbankseitig exklusiv dem angegebenen Datenbankbenutzer
--   zugewiesen.
-- 2.) Löschrecht verweigern
--   Setzen Sie die Variablen WS_EFI_DELETE := 0, damit kein Löschrecht
--   zugeweisen wird.
:WS_EFI_DELETE := 0;
-- Beispiel: Ein anderer Benutzer muss in der Datenbank existieren
-- Entfernen Sie die führenden "--" vor der Variable :sUserEFI_TRUNC und
-- setzen Sie den entsprechenden Oracle-Benutzer ein.
--:sUserEFI_TRUNC := 'TEST_EFI_TRUNC';
```

…

2.4.2 Profil „standalone“

Im Profil „standalone“ wird anstatt von BALVI iP1 eine lokale H2-Datenbank verwendet. Diese wird beim ersten Start automatisch generiert und enthält daher keine Jobs oder Importe und auch keine Ausgabe der „Server-Protokolle“, da Protokolle nicht in die Datenbank „H2“ gespeichert werden.

Für den Standalone-Modus gibt es ein separates Installations- und Benutzerhandbuch.

3 Konfiguration des Schnittstellenservers

3.1 Die „Context“-Parameter

Der Knoten „Context“ bzw. die Syntax der XML-Datei wird in der Spezifikation von Tomcat[11] beschrieben. Die von der Servlet-Spezifikation vorgegebene Rump-Struktur sieht folgendermaßen aus:

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE Context>
<Context>
   <!--  Kundenspezifische Einstellungen -->
</Context>
```

Anwendungsspezifische Einstellungen werden zwischen <Context> und </Context> eingefügt. Alle allgemeingültigen Einstellungen in der Datei „context.xml“ werden hier erläutert. Kommentare finden sich auch in der von BALVI ausgelieferten Beispiel-Konfiguration.

Neben den im Folgenden beschriebenen Standard-Einstellungen kann es auch vorkommen, dass bundeslandspezifische Erweiterungen vorhanden sind, welche nicht in dieser Dokumentation aufgeführt werden. Diese funktionieren trotzdem regulär weiter, wenn keine expliziten Informationen zu Kompatibilitätsproblemen in den Release Notes aufgeführt wurden.

3.2 Attribute des Knotens „Context“

Die Dokumentation von Apache Tomcat beschreibt diverse Einstellmöglichkeiten, die im Einzelnen für den BSS nicht relevant sind. Nur die für den Betrieb des BSS sinnvollen Attribute werden hier erläutert.

Attribut	Beschreibung	Empfehlung
sessionCookieName=“JSESSIONID“	Der Default-Cookie-Name ist JSESSIONID. Cookies werden zur Aufrechterhaltung der Sessions im Browser gespeichert und der Webanwendung wieder mitgegeben. Das Setzen des Parameters ist optional, kann jedoch sinnvoll sein, wenn Sie mehrere Tomcat-Instanzen parallel auf einem Host betreiben. Wenn Sie mehrere Tomcats auf demselben Host verwalten und diese über den PORT ansprechen, so kann das dazu führen, dass der Cookie „JESSIONID“ an mehrere Anwendungen weitergegeben wird, da Browser den Port beim Session-Handling ignorieren. Dies führt häufig zu Fehlern, die dann in der Folge die laufende Session invalidieren.	Setzen Sie je Fachanwendung einen eindeutigen CookieName z. B. sessionCookieName=“BSSPRODSESSIONID“, wenn Sie regelmäßig den Fehler haben, dass die Session des BSS beim Wechsel der Anwendungen im Browser geschlossen wird und Sie sich neu einloggen müssen.

Tab. 4: Für den Betrieb des BSS relevante Attribute des Knotens „Context“

[11] https://tomcat.apache.org/tomcat-8.0-doc/config/context.html

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE Context>
<Context sessionCookieName="BSSPRODSESSIONID">
   <!--  Kundenspezifische Einstellungen -->
</Context>
```

3.3 Grundsätzliche Hinweise für Einträge im „Context“

Die Elemente vom Typ

```
<Resource name="" … />
<Environment name="" … />
<Parameter name="" …  />
```

haben alle ein Attribut **name**=“vorgegebener Name von BALVI“, welches nicht geändert werden darf, damit die Anwendung den von Ihnen gesetzten Wert korrekt nutzen kann. Die durch den Kunden zu pflegenden Einstellungen bei den Typen Parameter und Environment müssen bei value=“kundenspezifischer Wert“ eingetragen werden.

Der Typ Resource ist da etwas anders. Er spiegelt die XML-Repräsentation eines Java-Objekts wieder und hat daher unterschiedliche Attribute. Hier empfehlen wir, nur die Werte anzupassen, die im Folgenden explizit erläutert werden, jedoch ansonsten die BALVI-Empfehlungen beizubehalten.

3.4 Jeder Schnittstellenserver benötigt eine „serverID“

Viele Kunden nutzen mehrere Instanzen des BSS in Ihrer Umgebung, da z. B. mehrere Installationen von BALVI iP (Produktion und Test) in der Infrastruktur existieren.

Da die Schnittstellenkonfiguration für den BSS in Teilen in der Datenbank abgelegt wird, kam es in einigen Fällen vor, dass nach dem Klonen der Produktionsdatenbank auf ein Testsystem die im BSS angelegten Jobs sofort nach dem Start in der Testumgebung wieder gestartet sind.

Um dieses Problem zu beseitigen, muss jedem BSS eine eindeutige „serverID“ zugewiesen werden. Diese Server-ID wird in der Konfiguration hinterlegt, um neutral zur Datenbank zu bleiben.

Warnung:

Ist die Environment-Variable „serverID“ nicht vorhanden, wird der Start des BSS sofort abgebrochen. Im Server-Protokoll wird dabei die Fehlermeldung „ERROR *[IdentityServiceImpl] serverID must not be empty*“ ausgegeben. Danach wird der Start des BSS abgebrochen.
 	
`ERROR d.b.c.identity.IdentityServiceImpl - serverID must not be empty`
`ERROR o.s.boot.SpringApplication - Application startup failed`
 	
org.springframework.beans.factory.UnsatisfiedDependencyException: Error creating bean with name 'etlTaskController': Unsatisfied dependency expressed through field 'identityService'; nested exception is org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'currentIdentityService' defined in class path resource [spring/app-context.xml]: Invocation of init method failed; nested exception is java.lang.Exception: serverID must not be empty

Um eine „serverID“ zu vergeben, fügen Sie in der Konfiguration folgende Angabe ein:

```
<!-- ID für den Server
  Diese muss für jeden Schnittstellenserver eindeutig sein. Eine Änderung dieser ID führt
  sofort dazu, dass alle eingerichteten Server-Jobs inaktiv werden und neu mit der Instanz
  gekoppelt werden müssen, um wieder aktiviert werden zu können.
  Als "value" kann ein Text mit einer max. Länge von 32 Zeichen vergeben werden, z. B.
  "ProductionEnvNode1" für die Produktion und "TestEnvNode1" für die Testumgebung.
-->
<Environment
   name="serverID"
   value="ProductionEnvNode1"
   type="java.lang.String"
   override="true"
/>
```

Warnung:
Wenn Sie mehreren BSS-Instanzen eine identische ServerID und eine identische Datenbank-Verbindung zuweisen, werden zum automatischen Start konfigurierte Jobs in allen Instanzen parallel starten. Das kann zu Fehlern führen.

3.4.1 Namensvergabe optional möglich

Auf Wunsch mehrerer Kunden wurde die Option geschaffen, jedem Server zusätzlich einen kundenspezifischen „Titel“ zu geben, um die Server besser unterscheiden zu können.

```
<!-- (Optional) In der Benutzeroberfläche angezeigter Namenszusatz des Servers -->
<Environment
   name="serverName"
   value="Test Mobil Server mit Test 124."
   type="java.lang.String"
   override="true"
/>
```

Der hier angegebene Text wird in der Kopfzeile zwischen den eckigen Klammern ausgegeben:

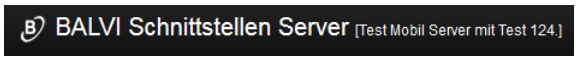


Abb. 3: Anzeige des Servernamens in der Menüleiste

3.5 Datenbankverbindung jdbc/DataSource

Die Konfiguration von Oracle ist nur gültig, wenn Sie das Profil „ip“ verwenden.

Die Datenbankverbindung wird über die von DBCP und dem Oracle-Treiber vorgegebenen Möglichkeiten realisiert. Der Einsatz der org.apache.tomcat.jdbc.pool.DataSourceFactory (automatisch verfügbar ab Apache Tomcat 7 in der mitgelieferten tomcat-dbcp.jar) wird als Beispiel-Konfiguration ausgeliefert:

```
<Resource
 name="jdbc/DataSource"
 auth="Container"
 type="javax.sql.DataSource"
 factory="org.apache.tomcat.jdbc.pool.DataSourceFactory"
 maxActive="10" initialSize="3"
 maxIdle="5"
 maxWait="10000"
 url="jdbc:oracle:thin:@BL2-ORA10-KUNDEN.BALVI.INTERN:1521:ORCL"
 driverClassName="oracle.jdbc.driver.OracleDriver"
 username="balvi_webservice_schema"
 password="balvi_webservice_password"
 description="Oracle Datasource"
 maxOpenPreparedStatements="50"
 removeAbandoned="false"
 removeAbandonedTimeout="100"
 logAbandoned="true"
 jdbcInterceptors="…"
/>
```

Die folgenden Attribute müssen durch den Kunden auf die korrekten Werte gesetzt werden:

1. url = Verbindungs-Einstellung zur Datenbank gem. Oracle JDBC-Spezifikation[12]
2. username = Oracle Schemabesitzer zum Verbinden
3. password = das zugehörige Passwort
4. min/maxActive = die max. zulässige Anzahl an DB-Connections. Die Hinweise beschreiben, wie Sie diese Werte bemessen können.

Hinweise zur Einstellung der "jdbc/DataSource"

Standard-Einstellungen für den Datenbank-Verbindungspool:

Der Verbindungspool muss entsprechend der Anzahl der zu erwartenden gleichzeitigen Benutzer konfiguriert werden. Wenn Sie erwarten, dass 50 Anwender gleichzeitig arbeiten, stellen Sie bitte folgende Werte ein:

```
maxActive=(50 + 10 (für System und Jobs)) = 60
maxIdle= (50 * 10%) = 5
```

1. maxActive - maximale Anzahl der gleichzeitigen aktiven Anfragen an die DB
   Wenn der Wert „maxActive“ erreicht ist, z. B. wenn mehrere Benutzer gleichzeitig einen Export starten möchten, welcher längere Zeit in Anspruch nimmt, dann kann der Server die Anfrage nicht mehr bedienen und der Anwender wird abgewiesen.
   „maxActive“ hat zudem Auswirkungen auf die im Hintergrund ausgeführten Jobs. „maxActive“ wird daher kalkuliert mit der Anzahl der Anwender und möglicher Datenbank-Connections für Hintergrund-Jobs. Jeder aktiv ausgeführte Job benötigt mindestens eine Connection, nur selten wird aus Performance-Gründen mehr als eine Connection während der Job-Ausführung verwendet. Im Normalfall sollten daher 10 Connections für Hintergrundprozesse ausreichend sein.

[12] https://docs.oracle.com/cd/E11882_01/java.112/e16548/urls.htm#JJDBC08200

2. maxIdle - Anzahl der offenen Verbindungen, auch wenn der Server im Leerlauf ist
   Jede offene Verbindung im Leerlauf hält auf dem Datenbankserver eine Session offen. Auch diese erfordert Ressourcen. Wenn die Verbindungen jedoch geschlossen werden, kann die Anfragezeit für eine neue Verbindung steigen, da diese Verbindung erst neu ausgehandelt werden muss. Die Empfehlung für den Wert „maxIdle“ ist ca. 10% der Einstellung von maxActive.

KeepAlive-Einstellungen für den Pool:

1. validationQuery
   Diese wird ausgeführt, um die Verbindung zur DB zu validieren. Es MUSS ein Datensatz zurückgegeben werden, ansonsten nimmt der DBCP an, dass die Abfrage nicht erfolgreich abgesetzt wurde. Achten Sie also bitte darauf, dass die Abfrage nicht zu leitungsintensiv ist und genau einen Datensatz liefert. Oracle bietet dafür die virtuelle Tabelle DUAL an, welche sich für die ValidationQuery sehr gut eignet, da die Abfrage kaum Ressourcen verbraucht.

   ```
   validationQuery="select 1 from dual"
   ```

2. testOnBorrow - Boolscher Wert "true" oder "false"
   Diese Einstellung sorgt dafür, dass der Pool die Verfügbarkeit anhand der validationQuery prüft, bevor die DB-Verbindung an die Anwendung ausgeliehen wird. Empfehlung:

   ```
   testOnBorrow="true"
   ```

3. testWhileIdle - Boolscher Wert "true" oder "false"
   Diese Einstellung ist hilfreich, wenn zwischen Apache Tomcat und der DB eine Firewall mit Stateful Packet Inspection verwendet wird. Nach längerer Inaktivität wird normalerweise durch die Firewall die DB-Verbindung geschlossen. Dieses löst dann beim Ausleihen der Verbindung eine Fehlermeldung im Protokoll aus:

   ```
   "2012-08-16 16:54:18,720 WARN [http-0.0.0.0-8080-8]
   [org.springframework.jdbc.support.SQLErrorCodesFactory] Error while extracting database
   product name - falling back to empty error codes
   org.springframework.jdbc.support.MetaDataAccessException: Error while extracting
   DatabaseMetaData;  nested exception is java.sql.SQLRecoverableException: Getrennte
   Verbindung".
   ```

   Um dieses Problem zu beheben, sollte die Option testWhileIdle="true" eingestellt werden.

   ```
   testWhileIdle="true"
   ```

   Damit diese weiß, wie oft eine Prüfung durchgeführt werden muss, sind jedoch noch zusätzliche Angaben erforderlich, welche für die Einstellung der Zeit zusätzlich gesetzt werden müssen:
   a) minEvictableIdleTimeMillis - Long in Millisekunden
      Min. Zeit, wie lange eine DB-Verbindung im Leerlauf ist, bevor diese zur Prüfung gekennzeichnet wird. Dieses hängt stark von der Einstellung der Firewall ab. Wenn Sie eine sehr restriktive Firewall haben, kann es sein, dass diese die Verbindung nach wenigen

Minuten Leerlauf trennt. Stellen Sie den Wert immer so ein, dass dieser unter der Leerlaufzeit der Firewall liegt, empfohlen "300000" entspricht 300 s = 5 min.

```
minEvictableIdleTimeMillis="300000"
```

b) timeBetweenEvictionRunsMillis - Long in Millisekunden
Der Prüflauf startet in der Default-Konfiguration alle "1800000" Millisekunden = 30 Min. "n" als zu prüfen markierte Verbindungen werden in einem Prüflauf getestet. "n" wird bestimmt durch die Einstellung "numTestsPerEvictionRun".
Empfehlung: Stellen Sie den Wert "900000" ein (entspricht 900 s = 15 min).

```
timeBetweenEvictionRunsMillis="900000"
```

c) numTestsPerEvictionRun - Integer
Wie viele Verbindungen sollen getestet werden, während der Prüflauf ausgeführt wird. Im Normalfall sollte die Anzahl der zu testenden Verbindungen nicht zu hoch eingestellt werden. Der Defaultwert ist 3. Wenn Sie höhere Sicherheit haben wollen, dass keine „toten" Datenbankverbindungen bestehen bleiben, erhöhen Sie den Wert auf max. „(maxActive – maxIdle) /2". Wenn Ihnen 3 ausreichend erscheint, muss der Wert nicht gesetzt sein, dann wird der Defaultwert verwendet.

```
Beispiel: numTestsPerEvictionRun="10"
```

4. Interceptors konfigurieren
Neu ergänzt wurde die Möglichkeit, spezielle „Interceptors" zu konfigurieren, mit denen das Verhalten des Pools gesteuert werden kann. Wichtig hierbei ist, dass die Angabe von Werten auf real existierende Java-Klassen erfolgt, die im Classpath des Tomcat vorhanden sind. Groß-und-Kleinschreibung ist zu beachten. Folgende Default-Einstellung hat BALVI für den BSS festgelegt:

```
jdbcInterceptors="
   org.apache.tomcat.jdbc.pool.interceptor.ConnectionState;
   org.apache.tomcat.jdbc.pool.interceptor.StatementFinalizer;
   org.apache.tomcat.jdbc.pool.interceptor.ResetAbandonedTimer"
```

Details zu den verwendeten Interceptor-Einstellungen finden Sie in der Dokumentation von Apache Tomcat[13].

3.6 Datenbankverbindung jdbc/KettleIP1DataSource

Die Datenquelle "jdbc/DataSource" dient für den Einsatz des BSS mit BALVI mobil XT und zur Selbstverwaltung des BSS. Kettle bzw. ETL-Prozesse führen jedoch massive parallele Abfragen auf das Datenbank-Schema von BALVI iP1 aus. Daher ist es erforderlich, einen anderen Ressourcen-Pool bereit zu stellen, damit die normalen Ressourcen des BSS für BALVI mobil XT nicht durch die neuen, teilweise

[13] http://tomcat.apache.org/tomcat-8.0-doc/jdbc-pool.html#JDBC_interceptors

komplexen Synchronisierungsprozesse des ETL-Prozessors beschränkt werden. Das Aktivieren der Datenquellen ist unkritisch und verursacht noch keine höhere Last.

Der ETL-Prozessor bringt einen internen Pool-Manager mit, daher darf hier kein von Tomcat verwalteter Pool-Manager verwendet werden. Ansonsten kann es bei Oracle-Datenbankverbindungen zu Fehlern vom Typ ORA-10000: TOO MANY OPEN CURSORS kommen.

Die Klasse (type=)"de.balvi.kettle.KettleDataSourceSettings" setzt voraus, dass sich die Bibliothek "kettledatasource-1.0.0.jar" im Ordner /lib befindet.

WICHTIG:

Der POOL wird für jeden parallel ausgeführten ETL-Job aufgebaut. Das ist insbesondere bei den Parametern "initialSize" und "maxActive" zu berücksichtigen, für die relativ kleine Werte gewählt werden können bzw. sollten, um die Datenbank nicht unnötig mit ungenutzten Verbindungen zu belasten.

Als Benutzer (url/username/password) tragen Sie bitte die Zugangsdaten zu BALVI iP1 identisch ein. Wenn kein ETL-Job im BSS vorhanden ist, wird diese Einstellung ignoriert.

```
<Resource
    name="jdbc/KettleIP1DataSource"
    auth="Container"
    type="de.balvi.kettle.KettleDataSourceSettings"
    factory="org.apache.tomcat.jdbc.naming.GenericNamingResourcesFactory"
    maxActive="3" maxIdle="2" minIdle="1" initialSize="1"
    maxWait="10000"
    url="jdbc:oracle:thin:@BHL-ORA12-KUNDEN:1521:ISO"
    driverClassName="oracle.jdbc.driver.OracleDriver"
    username="IP1_SCHEMA_OWNER"
    password="ip1_schema_passwort"
    description="Kettle Commons DBCP-Wrapper"
    validationQuery="select 1 from dual"
    testOnBorrow="true"
    testWhileIdle="true"
    minEvictableIdleTimeMillis="300000"
    numTestsPerEvictionRun="2"
    timeBetweenEvictionRunsMillis="900000"
  />
```

3.7 jdbc/xmlgenCacheDataSource (neu ab BSS 3.5.)

xmlGenerator (xmlgen)-Cache wird ausschließlich für den Einsatz von BALVI mobil 2 benötigt.

Der ab BSS 3.3 vorhandene ehCache zum Caching von xmlgen-Requests durch den mobil-Export kann folgendermaßen konfiguriert werden. Die ehCache-Konfiguration wird derzeit nur für die Synchronisation von BALVI mobil2 mit iP2-Fachmodulen verwendet. Der ehCache speichert dabei übertragenen Daten wie Kataloge zwischen.

BSS 3.3 => Die Konfiguration wurde in einer properitären Datei im ehCache_Format abgelegt.

```
  <!--
    ALTER Pfad auf der Festplatte des Servers zum Speicherort der Cache-Daten.
    Diese Einstellung kann ab BSS 3.5 gefahrlos gelöscht werden, falls Sie dies
```

```
  noch in Ihrer Konfiguration vorfinden.
 -->
<Environment
   name="xmlgen/cachePath"
   value="${catalina.base}/xmlgen-cache"
   type="java.lang.String" override="true"
/>
```

Diese Datei ist jedoch bei abrupten Neustarts des Servers in einigen Fällen korrupt gewesen. Daher hat BALVI die Logik ab BSS 3.5 auf eine reguläre Datenbank-Abfrage-Logik mit Pool-Manager und Transaktionssicherheit umgestellt.

Die Datenquelle wurde jedoch bewusst weiterhin als Dateidatenbank auf Basis des Java-Micro-Datenbanksystems H2 dem Server festgelegt. Die Datenquelle erzeugt und schreibt die Datei "${catalina.base}/xml-cache/cacheDB". Sollten Sie zwei BSS-Instanzen in einem Apache Tomcat betreiben, ändern Sie den Dateinamen ab, damit es keine Kollisionen gibt. Alternativ kann jedoch auch ein Schema in einem zentralen Datenbank-Server angelegt werden. Dies geschieht jedoch NICHT automatisch.

Fordern Sie dazu bitte bei BALVI die Create-Skripte für die CacheDB an.

```
<Resource
   name="jdbc/xmlgenCacheDataSource"
   auth="Container"
   type="javax.sql.DataSource"
   factory="org.apache.tomcat.jdbc.pool.DataSourceFactory"
   initialSize="2" minIdle="1" maxActive="3"
   maxWait="10000" maxIdle="2"
   url="jdbc:h2:${catalina.base}/xmlgen-cache/cacheDB;MODE=Oracle"
   driverClassName="org.h2.Driver"
   username="sa" password=""
   description="H2 Datasource xmlgen Cache"
   minEvictableIdleTimeMillis="300000"
   numTestsPerEvictionRun="2"
   timeBetweenEvictionRunsMillis="900000"
   testWhileIdle="true"
   testOnReturn="true"
   removeAbandoned="true"
   removeAbandonedTimeout="3600"
   logAbandoned="true"
   jdbcInterceptors="org.apache.tomcat.jdbc.pool.interceptor.ConnectionState;
                     org.apache.tomcat.jdbc.pool.interceptor.StatementFinalizer;
                     org.apache.tomcat.jdbc.pool.interceptor.ResetAbandonedTimer"
/>
```

Zusätzlich zur Datenbank-Konfiguration muss eingestellt werden, wie viel RAM der Cache-Logik zur Verfügung steht.

```
<!-- RAM-Konfiguration für ehCache -->
<!-- HEAP und OFF-HEAP werden in der JVM reserviert.
     Erhöhen Sie diesen Wert nur, wenn Sie zudem den MAX RAM (Parameter -Xmx)
     ebenfalls erhöhen.
  -->
```

```
<!-- Max. verfügbarer HEAP-Speicher für Caching in der JVM. Dieser Wert sollte nie mehr als
     2% des Wertes von -XmX betragen. Bei 2 GB RAM sind 20 MB empfohlen. -->
<Environment
   name="xmlgen/cacheMaxHeapMb"
   value="20"
   type="java.lang.Long" override="true"
/>

<!-- Max. verfügbarer NON-HEAP-Speicher in der JVM. Dieser Wert sollte nie mehr als
     10% des Wertes von -XmX betragen. Bei 2 GB RAM sind 100 MB empfohlen. -->
<Environment
   name="xmlgen/cacheMaxOffHeapMb"
   value="100"
   type="java.lang.Long" override="true"
/>
```

3.8 Active-Directory-Authentifizierung

Über die folgenden Parameter wird der Zugriff des Schnittstellen-Servers auf einen Microsoft Active Directory Domain-Controller konfiguriert.

“ldapServerAddress”

Syntax: PROTOKOLL://IP-Adresse oder DNS-Name:PORT/"

```
<Environment
  name="ldapServerAddress"
  value="ldap://bl2-dc02.balvi.intern:389/"
  type="java.lang.String" override="true"
/>
```

Als Protokoll kann auch "ldaps" (LDAP+SSL) verwendet werden, wenn der Domain-Controller entsprechend konfiguriert wurde. Im Java Truststore muss dann das Zertifikat der CA vorhanden sein, die das Zertifikat der Domain-Controller signiert hat. Zudem darf beim Einsatz von "ldaps" keine IP-Adresse vergeben werden, es muss der DNS-Name eingetragen werden, der im Zertifikat hinterlegt wurde. Ansonsten gibt es beim Verbinden Java-Sicherheits-Exceptions.

```
<!-- SSL-Beispiel: -->
<Environment
  name="ldapServerAddress"
  value="ldaps://bl2-dc02.balvi.intern:636/"
  type="java.lang.String" override="true"
/>
```

„ldapBaseDn”

LDAP-Suchpfad für die Domain-Basis. Diese muss der LDAP-Syntax für das Domain-Suffix entsprechen:

Syntax: Gültige LDAP-BaseQuery DC=SERVER,DC=ROOTDN[,OU=…]

```
<Environment
  name="ldapBaseDn"
  value="DC=BALVI,DC=INTERN"
  type="java.lang.String" override="true"
```

```
/>
```

„webLoginUseActiveDirectoryAuthentication"

Die Authentifizierung in der Browser-Oberfläche kann über die Active-Directory-Authentifizierung erfolgen.

```
<Environment
  name="webLoginUseActiveDirectoryAuthentication"
  value="false"
  type="java.lang.Boolean" override="true"
/>
```

3.9 Statement-Protokollierung in BALVI iP

Mit der Einstellung können Sie die Lese- und Änderungsprotokollierung für BALVI iP aktivieren bzw. deaktivieren. Standardmäßig ist diese Funktion deaktiviert ("false" oder auskommentiert). Wir empfehlen allen Kunden, diese Einstellung beizubehalten, solange keine zusätzlichen Datenbank-Jobs eingerichtet wurden, die zeitgesteuert ausgeführt werden und die Protokolleinträge regelmäßig löschen.

Wenn Sie mehr darüber erfahren möchten, kontaktieren Sie bitte die BALVI GmbH.

```
<Environment
  name="enableSqlStatementMonitoring"
  value="true"
  type="java.lang.Boolean" override="true"
/>
```

Beachten Sie, dass diese Option massive INSERT-Statements in der Tabelle VIS_D92 von BALVI iP ausführt. Jedes abgesetzte SQL-Statement wird dabei vollständig protokolliert. Das erzeugt eine sehr hohe INSERT-Last und in der Folge auch erhöhte Redolog-Switches und COMMIT-Transaktionen.

3.10 Einstellungen für BALVI mobil 1.x

3.10.1 Der „FIFO-Puffer" für den Modus „partitionedExport"

Um die Daten aus dem BSS zur Anwendung BALVI mobil XT zu senden, generiert der Server je Benutzer einen XML-Export-Stream. Dieser XML-Export-Stream kann auch mehrere 100 MB beinhalten. Da die Übertragung und die Aufbereitung von großen Datenmengen in der Webservice-Technologie nicht stabil funktionieren, wurde der XML-Export-Daten-Strom in „Partitionen" unterteilt.

Abb. 4: Schematische Darstellung des FIFO-Puffers (1)

Das ermöglicht z. B. dem BALVI mobil XT Client (1.4 oder höher), den aktiven Block abzuholen und dies beim Auftreten eines Fehlers bis zu 10-mal zu wiederholen. Sobald der Block [001] erfolgreich beim Client

angekommen ist und dieser den nächsten Block anfragt, wird der Block [001] verworfen wird. Der Server füllt dann im Hintergrund den frei gewordenen Block wieder, während der Client den Block [002] abholt.

Abb. 5: Schematische Darstellung des FIFO-Puffers (2)

Der FIFO-Puffer wurde so konzipiert, dass dem Rechenzentrum diverse Einstellmöglichkeiten über die Konfiguration zur Verfügung stehen, um die Steuerung zu beeinflussen.

Die Größe und die Anzahl der vorberechneten Blöcke können über die folgenden Einstellungen in der Konfiguration definiert werden. Der Puffercache wird ausschließlich im RAM der JVM aufgebaut, es wird folglich mehr Arbeitsspeicher benötigt. Der Arbeitsspeicher muss bei Erhöhung der gleichzeitig bedienten Sessions auf dem Server erhöht werden. Zudem wurde die Möglichkeit geschaffen, dass die maximale Anzahl gleichzeitiger Exporte durch das Rechenzentrum beschränkt werden kann, um einem OutOfMemory-Fehler vorzubeugen.

Die Default-Einstellung mit 3 Partitionen a 256 KB + dem aktiven Puffer erfordert 1 MB RAM für jeden aktiven Anwender.

Die Einstellung "partitionedExport/maxConcurrentExports" muss entsprechend der Anzahl der zu erwartenden gleichzeitigen Zugriffe konzipiert werden, ansonsten wird der Server einen Verbindungsversuch durch die Clients ablehnen.

Beachten Sie, dass die folgenden Einstellungen „partitionedExport/maxBufferedPartitions“, „partitionedExport/maxConcurrentExports“ und „partitionedExport/partitionSize“ gemeinsam als Rechenbasis für den RAM-Bedarf des Webservers herangezogen werden müssen.

Folgende Faustformel sollte für die Einrichtung verwendet werden:

Min. 256 MB (Serverbasis) + (1,1 * "partitionedExport/maxConcurrentExports" * "partitionedExport/maxBufferedPartitions" * "partitionedExport/partitionSize")

Beispiel 50 gleichzeitige Anwender, Blockgröße 256 KB, 4 Partitionen

256 MB + (1,1 * 50 * 256 * 4) KB = 312 MB für die JVM

Empf. 512 MB (Serverbasis) + (1,2 * "partitionedExport/maxConcurrentExports" * "partitionedExport/maxBufferedPartitions" * "partitionedExport/partitionSize")

Beispiel 50 gleichzeitige Anwender, Blockgröße 256 KB, 4 Partitionen

512 MB + (1,2 * 50 * 256 * 4) KB = 572 MB für die JVM

„partitionedExport/maxConcurrentExports“

Dieser Wert bestimmt die Anzahl der maximalen Export Tasks (Threads), die für partitionierte Exporte bereitgestellt werden. Wenn die Anzahl überschritten wird, wird die entsprechende Exportanfrage abgewiesen und der Client muss zu einem späteren Zeitpunkt erneut anfragen. Stellen Sie diesen Wert entsprechend der zu erwartenden Benutzeranzahl ein, bei 50 Anwendern also 50.

Exporte öffnen auf der Datenbank entsprechend zur Ausführungszeit min. einen Cursor, ggf. für rekursive Unterabfragen kurzfristig sogar mehr. Die hier eingestellte Anzahl muss mit der Anzahl von „maxActive“ korrelieren.

```
<Environment
   name="partitionedExport/maxConcurrentExports"
   value="10" type="java.lang.Integer" override="true"
/>
```

„partitionedExport/maxBufferedPartitions“

Dieser Wert bestimmt die Anzahl der Datenblöcke (Partitionen), die vom Server bei Abruf eines Exportes gepuffert werden. Die FIFO-Puffer-Logik ermöglicht, dass der Server bei einer Datenabfrage die Ergebnisse vorpuffert, wodurch sich für den Client die Wartezeiten verringern. Diese Puffer-Logik unterteilt den Exportdatenstrom (je Export) in mehre Blöcke. Der Client kann nun den letzten Block neu anfordern oder den nächsten Block, wenn der letzte Block erfolgreich verarbeitet wurde. Wie viele nächste Blöcke durch den Server bereit gestellt werden, entscheidet diese Einstellung. Sobald kein nächster Block mehr bereit gestellt werden muss, wird der Datenbank-Cursor geschlossen, eine Erhöhung kann also zu einer Entlastung der aktiven DB-Verbindungen führen, erhöht jedoch den Bedarf an RAM des Servers.

Der von BALVI voreingestellte Wert ist 3.

```
<Environment
   name="partitionedExport/maxBufferedPartitions"
   value="3"
   type="java.lang.Integer" override="true"
/>
```

„partitionedExport/partitionSize“

Dieser Wert bestimmt die Größe der Datenblöcke (Partitionen), die an den Client übermittelt werden, in Bytes. Wenn Sie diesen Wert zu niedrig einstellen, reduziert sich die Effizienz des Caches, da der Client mehr Datenblöcke hintereinander abrufen muss. Wählen Sie einen zu großen Wert, wird die neu zu übertragende Datenmenge im Fall eines Fehlers höher.

Der von BALVI voreingestellte Wert ist 256000 (256 KB).

```
<Environment
   name="partitionedExport/partitionSize"
   value="256000" type="java.lang.Integer" override="true"
/>
```

„exportTokens/maxTokenCount“

Dieser Wert bestimmt die maximale Anzahl der gleichzeitig vergebenen Token, die vom Server gecacht werden. Jede Sitzung, welche einen Export durchführt, erhält einen Session-Token.

Wenn eine Sitzung längere Zeit nicht verwendet wurde, wird dieser Session-Token wieder entfernt. Wenn Sie den Wert zu niedrig einstellen, führt das zu einer generellen Session-Begrenzung, auch für den nicht partitionierten Export-Modus, welcher ab Version 2.1 verfügbar ist.

Ist die Anzahl der Tokens höher als die Datenbankeinstellung „maxActive“, kann es vorkommen, dass zwar eine Session gestartet werden kann, jedoch kein Datenexport zustande kommt.

Der von BALVI voreingestellte Wert ist 2000.

```
<Environment
   name="exportTokens/maxTokenCount"
   value="2000" type="java.lang.Integer" override="true"
/>
```

„exportTokens/tokenExpirationInSeconds“

Dieser Wert bestimmt die Zeit in Sekunden, die vergehen muss, bis ein Token ungültig wird, falls darauf kein Zugriff mehr erfolgt. Nach einem erfolgreichen Export wird der Token automatisch geschlossen, diese Option ist für den Abbruchfall vorgesehen. Der von BALVI voreingestellte Wert ist 180 (180 s = 3 min).

```
<Environment
   name="exportTokens/tokenExpirationInSeconds"
   value="180" type="java.lang.Integer" override="true"
/>
```

„exportTokens/tokenDisposalInSeconds“

Dieser Wert bestimmt die Zeit in Sekunden, die vergehen muss, bis ein Token, das erfolgreich abgearbeitet wurde, aus dem Cache gelöscht wird. Um sicherzustellen, dass der letzte Exportblock noch erfolgreich abgerufen werden kann, bleibt der Token auch nach Fertigmeldung des Clients noch N Sekunden gültig.

Der von BALVI voreingestellte Wert ist 60 (60 s = 1 min).

```
<Environment
   name="exportTokens/tokenDisposalInSeconds"
   value="60" type="java.lang.Integer" override="true"
/>
```

3.11 Job-Verwaltung konfigurieren

Aktivierung der Jobverwaltung

Ab BSS 3.0 wird die Jobs-Engine standardmäßig aktiviert. Sollte die Jobs-Engine nicht benötigt werden, muss „value“ der Wert „false“ zugewiesen werden. Wird „isJobServiceEnabled“ nicht eingefügt, sind die Jobs trotzdem standardmäßig aktiv.

```
<!-- Neue Parameter zur Aktivierung des Job Service. Ersatz für "availableServices" -->
<Environment
   name="isJobServiceEnabled"
   value="true"
   type="java.lang.Boolean" override="true"
/>
```

Nach der Aktivierung ist der Menüpunkt „Jobs“ verfübar.

„passwordKey“ für Jobs

In der Job-Logik wird ein von BALVI definierter "Standard"-Schlüssel für die Ablagen von schützenswerten Daten, z. B. Passwörtern verwendet. Diese Einstellung ermöglicht, den BALVI-Standard zu überschreiben. Wenn Sie diese Einstellung ändern, müssen Sie allerdings alle bislang verschlüsselten Job-Parameter neu verschlüsseln!

```
<Environment
   name="java:/comp/env/jobs/passwordKey"
   value="MeinHochsicheresNeuesPasswort"
   type="java.lang.String" override="true"
/>
```

„Job ID enabled/disabled“ (Deprecated, ersetzt das „Binden“ von Jobs an die Server-ID)

Sondereinstellungen für die Job-Verwaltung. Da alle Jobs in einer Datenbank administriert werden, jedoch ggf. verteilt auf mehreren Tomcat-Servern ausgeführt werden sollen, besteht hier die Option, Jobs für die aktuelle Tomcat-Instanz separat zu deaktivieren.

Die Einstellung {ID} könnte z. B. folgendermaßen aussehen:
java:/comp/env/jobs/C0DB58D8D52C4BBC966D1DC788A4EF8B/enabled

Für weitere Fragen zu dieser Einstellung kontaktieren Sie bitte die BALVI GmbH.

```
<Environment
   name="java:/comp/env/jobs/{ID}/enabled"
   value="false"
   type="java.lang.Boolean" override="true"
/>
```

3.12Einstellungen für iP2 – Synchronisation (ab BSS 3.3)

Der BSS übernimmt auch für BALVI iP2 die Kommunikation zwischen Fachmodul und BALVI mobil Client. Im Gegensatz zu BALVI iP schreibt der BSS dabei nicht mehr direkt in die Datenbank, sondern leitet die Requests per REST an das entsprechende Fachmodul weiter. Daher sind die folgenden Einstellungen für die iP2-Datensynchronisation erforderlich:

Ziel-URLs für die iP2-Module:

```
<!-- Diese Einstellungen müssen gesetzt werden,
     wenn dieser Server für folgende Aufgaben vorgesehen ist:
     1.) Synchronisation mit mobil 2.x
     2.) ETL-Datentransfer iP1 zu iP2
-->
<Environment
   name="oauthServerUrl"
   value="https://dm-test.intern.balvi.de/rest/v2/"
   type="java.lang.String" override="true"
/>

<!-- Modul Düngemittel (derzeit nur für NW LAND/LANUV)-->
<Environment
   name="ip2urls/dmModul"
   value="https://dm-test.intern.balvi.de/"
   type="java.lang.String" override="true"
/>
```

```
<!-- Lebensmittel-Modul iP2 -->
<Environment
   name="ip2urls/lmModul"
   value="https://lm-test.intern.balvi.de/"
   type="java.lang.String" override="true"
/>

<!-- Ziel-URL für ÖKO-Modul -->
<Environment
   name="ip2urls/oekoModul"
   value="https://oeko.intern.balvi.de/"
   type="java.lang.String" override="true"
/>
```

3.13 Weitere optionale Parameter

Es ist für diverse einzelne Schnittstellen ggf. erforderlich, Einstellungen in den Anwendungs-Kontext zu ergänzen. Hinweise zu diesen Parametern sind dann in dem der Schnittstelle beiliegenden Dokument enthalten. Das gilt z. B. für die Schnittstellen XGewerbeanzeige oder Staatsarchiv BY.

3.14Logging mit logback konfigurieren

Protokolle mit Informationen über Programmzustände (Log-Dateien) werden benötigt, um den Ablauf nachvollziehen und verstehen zu können. Für die Protokollierung im BSS verwenden wir das Logging-Framework Logback.

Um das Logging des BSS mit Logback zu aktivieren, sind folgende Schritte nötig:

- ✔ Öffnen Sie die Datei „[context].xml“ (Verzeichnis „conf/Catalina/localhost“).
- ✔ Stellen Sie sicher, dass die Einstellung vorhanden ist:

```
<Environment
  name="logging.config"
  value="${catalina.base}/conf/logback-bss.xml"
  type="java.lang.String" override="true"
/>
```

- ✔ Speichern und schließen Sie die Datei.
- ✔ Kopieren Sie die Datei „logback-bss.xml“ aus dem Auslieferungspaket in den Ordner „conf“
- ✔ Die Standard-Konfiguration muss nicht angepasst werden.

Die Konfiguration der Java Logging-API ist „anwendungsspezifisch“. Sie können die Meldungen auf die Konsole ausgeben oder in externe Speicher wie Text- oder XML-Dateien bzw. in eine Datenbank schreiben.

Es werden Fachfehler und Systemfehler protokolliert. Fachfehler werden vom Root-Logger in die für BALVI relevanten Protokolle geschrieben. Systemfehler werden weitergeleitet an den Server-Logger. Der Root-Logger sammelt alle Logging-Events ein, die nicht explizit durch spezifische „Klassen-Logger“ behandelt werden. Der Root-Logger wird mithilfe der Datei „logback-bss.xml“ konfiguriert, welche auch Meldungen an den Appender "STDERR" weiterleitet bzw. diese auf der Console ausgegeben werden.

Wenn Sie von der Schnittstelle eine Fehlermeldung angezeigt bekommen, können Sie mithilfe der Einträge im Protokoll die Ursache für den Fehler eingrenzen bzw. ermitteln.

In der Benutzeroberfläche lässt sich die Liveausgabe im Protokoll-Fenster aktivieren, sodass die Meldungen zur Laufzeit in der Benutzeroberfläche angezeigt werden. Dort sehen Sie allerdings nur den aktuellen Ausschnitt. Die Standardkonfiguration wurde um BSS-spezifische Features ergänzt. Die Standardkonfiguration ist in der Datei „logback.xml“ festgelegt und unveränderlich.

Sie können die Konfiguration dynamisch zur Laufzeit ändern (Log-Ausgabe). Die Änderung der Parameter in der Benutzeroberfläche des BSS hat allerdings keine Auswirkungen auf die Standardkonfiguration. Die dynamische Änderung hat aber Auswirkung darauf, was in der Log-Datei gespeichert wird.

Zur Protokollierung des Servers Apache Tomcat wird standardmäßig der LogManager „JULI“ („java.util.logging“-Implementation) verwendet.

3.15 Optional: E-Mail-Versand aktivieren (ab BSS 3.5)

Auf Wunsch vieler Kunden wurde die Möglichkeit ergänzt, E-Mail-Nachrichten an unterschiedliche Empfänger zu versenden, wenn ein ausgeführter Job einen Fehler gemeldet hat. Dazu müssen folgende Einstellungen in der Anwendungskonfiguration (context.xml) ergänzt werden. Der BSS verwendet die Java Mail-API. Die möglichen Parameter zur Konfiguration können hier nachgelesen werden:

https://docs.oracle.com/javaee/7/api/javax/mail/package-summary.html oder

https://www.tutorialspoint.com/javamail_api/javamail_api_smtp_servers.htm

BALVI liefert ein vorkonfiguriertes Beispiel für die Übermittlung an einen offenen SMTP-Relay, wenn Sie zusätzlich Authentifizierung oder TLS-Verschlüsselung konfigurieren möchten, müssen zusätzliche Angaben ergänzt werden.

Die Einstellung des Wertes vom Parameter jobMailSettings wird als Liste hinterlegt. Jedes Wertepaar wird dabei mit NAME=WERT definiert, zwischen den Werten muss ein Semikolon angegeben werden.

Ist die Einstellung "jobMailSettings" auskommentiert, so werden keine E-Mails vom BSS versendet.

```
<!-- Sende-Konfiguration - Ausgehender SMTP-Server -->
<Environment
   name="jobMailSettings"
   value="host=smtp-server.domain.de;
          port=25;
          mail.smtp.auth=false;
          mail.smtp.starttls.enable=false;
          mail.debug=false"
   type="java.lang.String" override="true"
/>

<!--
  Sender-Email-Adresse:
  jobMailFromAddress wird als Absender der E-Mail angegeben
 -->
<Environment
   name="jobMailFromAddress"
   value="bss-noresponse-message@balvi.de"
   type="java.lang.String" override="true"
 />
```

```
<!--
  Link-Adresse, die in der Mail angegeben wird.
  Bei Betrieb hinter einem Proxy die externe URL des BSS
  -->
<!--
<Environment
   name="jobMailServerUrl"
   value="https://test-bss.intern.balvi.de"
   type="java.lang.String" override="true"
 />
```

4 Bekannte Probleme und Lösungen

4.1 ORA-00904: „SERVER_ID“ ungültiger Bezeichner

Bei der Job-Bearbeitung tritt folgender Fehler auf: ORA-00904: „SERVER_ID“ ungültiger Bezeichner

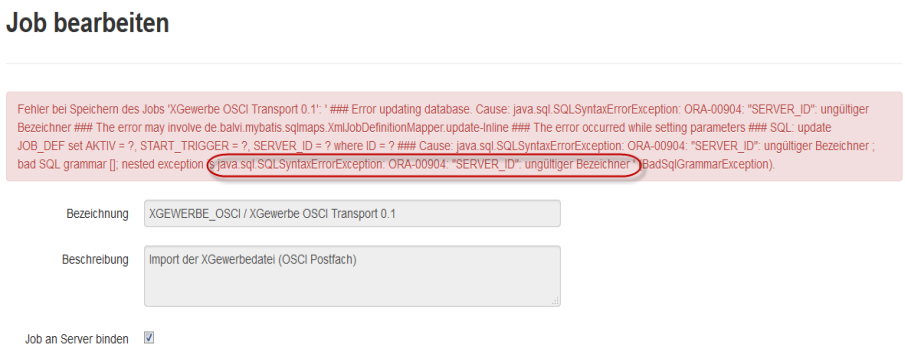


Abb. 6: Fehlermeldung ORA-00904

Behebung:

Führen Sie das beiliegende iP-Patch (UPDATE.exe) aus, bevor Sie die WAR-Datei austauschen.

4.2 Server startet nicht

Sie finden folgende Zeilen im Protokoll (catalina.out oder bss.log):

```
ERROR [localhost-startStop-1] [IdentityServiceImpl] serverID must not be empty
ERROR [localhost-startStop-1] [ContextLoader] Context initialization failed
```

Behebung:

Setzen Sie die Umgebungsvariable “serverID” in der Datei „context.xml“.

4.3 ORA-00942: table or view does not exist

In vielen Fällen wird der Fehler „Tabelle oder View nicht vorhanden“ ausgeworfen (Java-Protokoll im Ordner „logs“ von Apache Tomcat), wenn eine neue Schnittstelle eingerichtet oder ein neues BALVI iP Release bzw. Update eingespielt wurde, ohne anschließend das aktuelle GRANT-Skript auszuführen.

Behebung:

Führen Sie das GRANT-Skript erneut aus.

4.4 ORA-0406x „Bestehender Paketstatus wurde aufgehoben“

```
ORA-04061: Bestehender Status von package body "ABC.XYZ" wurde annulliert
ORA-04065: Ausführung nicht erfolgreich, package body "ABC.XYZ" wurde geändert oder gelöscht
```

Der Fehler tritt häufig auf, nachdem die UPDATE.exe ausgeführt und in der Ausführungsliste ein Package aktualisiert wurde. Oracle merkt sich je User-Session bestimmte Werte im Package, wenn der Anwender es verwendet hat. Durch einen „CREATE OR REPLACE PACKAGE“-Befehl kann dieser Status temporär aufgehoben werden. Normalerweise tritt dieser Fehler nach dem Update genau einmal in jeder aktiven Session auf, wenn der BSS nicht neu gestartet wird.

Behebung:

Starten Sie nach jedem Update den BSS neu, um alle laufenden Sessions zu beenden.

4.5 Jobs sind nicht ausführbar

	Name	Version	Verfahren	Beschreibung	Status
	XGewerbe Datei	1.3	XGEWERBE	Import der XGewerbedatei (Dateiimport)	Deaktiviert
	XGewerbe Datenanalyse	1.3	XGEWERBE_DATENANALYSE	XGewerbe Datenanalyse Job	Aktiviert
	XGewerbe OSCI	1.3	XGEWERBE_OSCI	Import der XGewerbedatei (OSCI Postfach)	Nicht unterstützt

Abb. 7: Jobstatus „Nicht unterstützt“ und „Deaktiviert“

1. Ein Job wird mit dem Status „Nicht unterstützt“ angezeigt, wenn für seine Ausführung eine höhere Version des BSS erforderlich ist.
2. Nach dem Klonen der Datenbank von der Produktions- auf die Testumgebung erhalten alle Jobs automatisch den Status „Deaktiviert“, wenn die „serverID“ nicht identisch ist.

Beide Statusänderungen sind gewollt. Damit soll verhindert werden, dass Jobs nach einem Neustart des Servers oder nach einem Update automatisch starten und dabei Fehler verursachen.

Um Jobs mit dem Status „Nicht unterstützt“ nutzen und konfigurieren zu können, müssen Sie den BALVI Schnittstellenserver aktualisieren.

Deaktivierte Jobs können Sie aktivieren, indem Sie den jeweiligen Job an den aktuellen Server binden und den Haken bei „Aktiv“ setzen.

Abb. 8: Job an Server binden und aktivieren

4.6 XML-Parser Fehler

Da von der Schnittstelle XGewerbeanzeige der Java XML-Parser „Xerces“ verwendet wird, beinhaltet der BSS nun generell das Verzeichnis „endorsed“. Wenn der Ordner „endorsed“ nicht vorhanden ist oder sich bestimmte Dateien nicht in diesem Ordner befinden, kann das auch bei anderen Servlet-Containern (WAR-Dateien) Probleme auslösen.

4.6.1 „Xerces“ im Fehlertext vorhanden

```
java.lang.ClassCastException: org.apache.xerces.parsers.XML11Configuration cannot be cast to
org.apache.xerces.xni.parser.XMLParserConfiguration at
com.sun.xml.ws.transport.http.servlet.WSServletContextListener.contextInitialized
(WSServletContextListener.java:139)

„Unexpected exception parsing XML document; nested exception is java.lang.AbstractMethodError:
org.apache.xerces.dom.NodeImpl.setUserData(Ljava/lang/String;Ljava/lang/Object;Lorg/w3c/dom/UserD
ataHandler;)Ljava/lang/Object;
Caused by: java.lang.AbstractMethodError: org.apache.xerces.dom.NodeImpl.setUserData(…)“
java.lang.ClassNotFoundException: org/apache/xerces/jaxp/DocumentBuilderFactoryImpl
```

Stellen Sie sicher, dass sich die Dateien „xercesImpl-2.11.0.jar“ und „xml-apis-1.4.01.jar“ im Ordner „endorsed“ befinden und dass der Pfad zum Verzeichnis „endorsed_dir“ korrekt ist.

4.6.2 „Xalan“ im Fehlertext vorhanden

```
Caused by: org.springframework.beans.BeanInstantiationException: Could not instantiate bean class
[org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter]: Constructor
threw exception; nested exception is javax.xml.transform.TransformerFactoryConfigurationError:
Provider org.apache.xalan.processor.TransformerFactoryImpl not found
                        ... 5 more
Caused by: javax.xml.transform.TransformerFactoryConfigurationError: Provider
org.apache.xalan.processor.TransformerFactoryImpl not found
                        ... 5 more
```

Stellen Sie sicher, dass sich die Dateien „xalan-2.7.2.jar“ und „serializer-2.72.jar“ im Ordner „endorsed“ befinden und dass der Pfad zum Verzeichnis „endorsed_dir“ korrekt ist.

4.7 Netzwerk-Fehler

Fehler der Klasse „java.net.[]“ (z. B. “java.net.ConnectionException: connection timed out” oder “java.net.UnknownHostException”) sind meistens netzwerkspezifisch. Prüfen Sie, ob der Intermediär ggf. nur über einen Proxy erreicht werden kann. Sollte der Server im lokalen Netzwerk über Routing erreichbar sein, kann ggf. auch die DNS-Namensauflösung fehlen.

4.7.1 UnknownHostException

Im Beispiel ist ein Tippfehler für die UnknownHostException verantwortlich.

```
Caused by: java.net.UnknownHostException: gov.test.osci.ed
                        at
java.net.AbstractPlainSocketImpl.connect(AbstractPlainSocketImpl.java:184)
                        at java.net.SocksSocketImpl.connect(SocksSocketImpl.java:392)
                        at java.net.Socket.connect(Socket.java:589)
                        at java.net.Socket.connect(Socket.java:538)
                        at sun.net.NetworkClient.doConnect(NetworkClient.java:180)
```

Da die öffentlichen Intermediäre über das Internet für jeden zugänglich sein müssen, können Sie die Adresse jederzeit in einem Browser Ihrer Wahl überprüfen. Sie sollten dann auf die Anfrage an den korrekten Soap-Endpunkt eine Meldung erhalten:

„I don't speak GET - Send POST to URL - -“

Wenn diese Adresse jedoch aus dem BSS-Job nicht erreichbar ist, kann das ggf. an einem Proxy-, DNS- oder Routing-Problem des Servers liegen, welcher vom Rechenzentrum behoben werden muss.