

Entwicklung einer standardisierten Steuerungssoftware
für eine Streckenbeeinflussungsanlage am Beispiel der A 8
zwischen AD Leonberg und AS Wendlingen (SSW-SBA-A8)

Betriebshandbuch
Anwendungshandbuch
Diagnosehandbuch

Segment 7 (Ste), SWE 7.5 Steuerungsalgorithmen SBA

Algorithmus Nachlauf

Version	2.0
Stand	14.12.2015
Produktzustand	Akzeptiert
Datei	BetrInf_STE_Nachlauf_FREI_V2.0_D2015-12-14.doc

Projektmanager	Herr Dr. Bettermann
Projektleiter	Herr Hannes
Projektträger	Regierungspräsidium Tübingen Landesstelle für Straßentechnik Heilbronner Straße 300 - 302 70469 Stuttgart
Ansprechpartner	Herr Hannes

0.3 Inhaltsverzeichnis

0 Allgemeines	2
0.1 Verteiler	2
0.2 Änderungsübersicht.....	2
0.3 Inhaltsverzeichnis	3
0.4 Abkürzungsverzeichnis.....	5
0.5 Definitionen.....	5
0.6 Referenzierte Dokumente.....	5
0.7 Abbildungsverzeichnis.....	5
0.8 Tabellenverzeichnis	5
1 Zweck des Dokuments	6
2 Betriebshandbuch	7
2.1 Installation der Software	7
2.1.1 Erstinstallation der Software	7
2.1.2 Aktualisieren der Software	7
2.1.3 Deinstallation der Software	7
2.2 Konfiguration und Aufnahme des Betriebs	7
2.2.1 Voraussetzungen für den Betrieb.....	7
2.2.2 Konfiguration	8
2.2.2.1 Startparameter.....	8
2.2.2.2 Parametrierung der SWE.....	9
2.2.3 Aufnahme des Betriebs	9
2.2.3.1 Manueller Start.....	9
2.2.4 Wiederaufnahme des Betriebs nach einem Störfall.....	9
2.3 Überwachen des Betriebs	9
2.4 Vermeiden von Fehlern	9
2.5 Erkennen von Fehlern	10
2.6 Beheben von Fehlern	10
2.7 Unterbrechung oder Beendigung des Betriebs	10
2.7.1 Voraussetzungen	10
2.7.2 Unterbrechung des Betriebs	10
2.7.3 Beenden des Betriebs	10
3 Anwendungshandbuch	11
4 Diagnosehandbuch	12
4.1 Benötigte Werkzeuge	12
4.2 Diagnosemöglichkeiten	12
4.2.1 Prozessliste des Betriebssystem	12
4.2.2 GTM.....	12

4.2.3	Analyse der Log-Files.....	12
4.2.3.1	Allgemeine Meldungen	12
4.2.3.2	Fehlermeldungen	12
5	Anhang	14
5.1	Verzeichnisstruktur des Algorithmus	14
5.2	Startscript für Windowssysteme (exemplarisch)	15
5.3	einstellungen.bat für Windowssysteme (exemplarisch)	15
5.4	Startscript für Linux-Systeme (exemplarisch).....	17
5.5	einstellungen.sh für Linux-Systeme (exemplarisch).....	18

0.4 Abkürzungsverzeichnis

Siehe [AbkBLAK].

0.5 Definitionen

Keine

0.6 Referenzierte Dokumente

BetrInf_Gesamtsystem BetrInf_Gesamt_SSW_SBA_A8.pdf

AbkBLAK SE-02.0001-Abk-4.0 [Abkürzungsverzeichnis (global)].pdf

0.7 Abbildungsverzeichnis

Keine

0.8 Tabellenverzeichnis

Tabelle 1-1: Typographie.....	6
Tabelle 1-2: Konventionen.....	6
Tabelle 2-1: SWE-spezifische Algorithmus-Einstellungen	8
Tabelle 4-1: Allgemeine Meldungen	12
Tabelle 4-2: Fehlermeldungen	13
Tabelle 3: Verzeichnisstruktur des Algorithmus	14

1 Zweck des Dokuments

In diesem Dokument sind die drei Bestandteile der Betriebsinformation zu finden.

- Betriebshandbuch
- Anwendungshandbuch
- Diagnosehandbuch

Die drei Dokumente wurden aus Gründen der Übersichtlichkeit in einem Dokument zusammengefasst.

Hinweise zu Typographie:

<i>kursiv</i>	Datei-, Ordner- und Benutzernamen werden kursiv dargestellt
Maschinenschrift	Befehle und Texte, die Sie eingeben müssen, werden in Maschinenschrift dargestellt
Maschinenschrift fett	Teile von Befehlen und Texten, die ggf. angepasst werden müssen, sind in Maschinenschrift und fett dargestellt

Tabelle 1-1: Typographie

Konventionen

~	Die Tilde steht für das Home-Verzeichnis des sba-Benutzers
\$SBA_HOME	Steht symbolisch für das Verzeichnis, in dem die SBA Software installiert wurde.

Tabelle 1-2: Konventionen

2 Betriebshandbuch

2.1 Installation der Software

Die SWE 7.5 ist Teil des Gesamtpakets SSW SBA A8. Zum Betrieb der Software sind notwendig:

1. Kernsystem ab Version 3.4.10
2. Bibliothek Algorithmusrahmen: die jar-Datei `de.bsvrz.ste.stealgsba.algorithmusrahmen.jar` muss sich im Klassenpfad befinden

Der Algorithmus wird als ZIP-Archiv geliefert (Beispiel: `SWE_de.bsvrz.ste.stealgsba.algorithmen.nachlauf_FREI_V1.0_D20150325_145828.zip`) und sollte in das Verzeichnis „`de.bsvrz.ste.stealgsba.algorithmen.nachlauf`“ parallel zu den anderen SWE entpackt werden.

2.1.1 Erstinstallation der Software

siehe [BetrInf_Gesamtsystem].

2.1.2 Aktualisieren der Software

siehe [BetrInf_Gesamtsystem].

2.1.3 Deinstallation der Software

siehe [BetrInf_Gesamtsystem].

2.2 Konfiguration und Aufnahme des Betriebs

2.2.1 Voraussetzungen für den Betrieb

Grundsätzliche Voraussetzung ist, dass folgende Kernsoftware-Komponenten gestartet sind:

1. Datenverteiler
2. Konfiguration
3. Parametrierung

Weiterhin sind zum Betrieb notwendig:

4. Gestartete Betriebsmeldungsverwaltung (optional)
5. installierter Algorithmusrahmen (**`de.bsvrz.ste.stealgsba.algorithmusrahmen`**)

Um den Algorithmus in Betrieb zu setzen, müssen der Parametrierung, für die Konfigurationsbereiche in denen die Ursacheneinheiten „Nachlauf“ enthalten sind, folgende Attributgruppen als parametrierend bekannt gemacht werden:

1. EinzelMassnahme
2. UrsachenEinheit
3. UrsachenEinheitNachlauf

2.2.2 Konfiguration

2.2.2.1 Startparameter

Vor der ersten Inbetriebnahme müssen die Startparameter des Algorithmus konfiguriert werden.

Der Algorithmus Nachlauf wird über Start-Stop gestartet.

Hier werden auch die Algorithmus spezifischen Einstellungen vorgenommen. Zentrale Einstellungen werden in der Datei *einstellungen* festgelegt.

Einstellungen, die die Java Virtual Maschine betreffen, sollten direkt nach der Variable \$JAVA_ARGS bzw. %JAVA_ARGS% in das Startscript eingetragen werden.

Nr.	Parameter	Beschreibung
1	<JVM Parameter>	Nach \$ JAVA_ARGS bzw. %JAVA_ARGS% können weiter Parameter für die Java Virtual Maschine angegeben werden.
2	-benutzer=<Benutzer>	Unter diesem Benutzer wird der Nachlauf beim Datenverteiler angemeldet. \$BENUTZER bzw. %BENUTZER% verwenden, wenn die Daten aus der Datei <i>einstellungen</i> verwendet werden soll.
3	-authentifizierung=<passwd-Datei>	Verweis auf eine Datei, die für einen oder mehrere anzumeldende Benutzer das Passwort enthält. \$AUTHENTIFIZIERUNG bzw. %AUTHENTIFIZIERUNG% verwenden, wenn die Daten aus der Datei <i>einstellungen</i> verwendet werden sollen.
4	-datenverteiler=<host>:<port>	Adresse des Datenvertailers \$HOST:\$PORT bzw. %HOST:PORT% verwenden, wenn die Daten aus der Datei <i>einstellungen</i> verwendet werden sollen.
2	-Anlage=	Das Argument –Anlage muss ein Objekt vom Typ <code>typ.sswSbaStreckenBeeinflussungsAnlage</code> , über die Pid, bezeichnen
3	-AspektVersand=	Pid Aspekt, unter dem die Massnahmenanforderung versendet werden soll. z.B. <code>asp.sswSbaNachlauf</code>
5	-ZyklusLaenge=	(optional) Länge des Steuerungszykluses in Millisekunden. Defaultwert = 60000.
6	-ZyklusOffset=	(optional) Offset für den Steuerungszyklus in Millisekunden. Defaultwert = 0.
7	-MaxZyklusDauer=	(optional) Maximale Dauer eines Steuerungszyklus in Millisekunden. Bei Überschreitung der eingestellten Zeit, wird eine Betriebsmeldung abgesetzt. Defaultwert = 30000.
8	-Spontan=<BOOLEAN>	(optional) Spontane Abarbeitung der Befehle durch den Steuerungskern. Defaultwert = false.

Tabelle 2-1: SWE-spezifische Algorithmus-Einstellungen

2.2.2.2 Parametrierung der SWE

Das Betriebsverhalten des Algorithmus wird durch die folgenden Parameter beeinflusst.

Parameter	Beschreibung
EinzelMassnahme	Definition der Massnahmenanforderung
UrsachenEinheit	Globale Parameter der Ursacheneinheit
UrsachenEinheitNachlauf	Spezielle Parameter für den Algorithmus

2.2.3 Aufnahme des Betriebs

Der Algorithmus wird über Start-Stop gestartet.

2.2.3.1 Manueller Start

Der Algorithmus kann ebenfalls über Start-Stop manuell gestartet werden.

2.2.4 Wiederaufnahme des Betriebs nach einem Störfall

Wenn der Algorithmus nicht mehr funktionsfähig ist, kann der Prozess beendet werden, sofern er sich nicht selbst beendet hat. der Algorithmus kann wie in [BetrInf_Gesamtsystem] beschrieben manuell gestartet werden.

2.3 Überwachen des Betriebs

Ob der Algorithmus noch läuft, lässt sich mittels des LINUX-Tools `ps` bzw. über den Windows Task Manager überprüfen.

Der zu suchende Prozessname lautet:

`de.bsvrz.ste.stealgsba.algorithmen.nachlauf.NachlaufAlg`

2.4 Vermeiden von Fehlern

1. Die im Kapitel „Voraussetzungen für den Betrieb“ genannten Komponenten des Kernsystems müssen gestartet sein.
2. Die SWE Parametrierung muss für die im Kapitel „Voraussetzungen für den Betrieb“ als parametrierend genannten Attributgruppen zuständig sein.

2.5 Erkennen von Fehlern

Die Fehler des Algorithmus werden in den Logfiles mitprotokolliert (siehe Diagnosehandbuch).

Wenn nicht alle Voraussetzungen für den Start der Applikation erfüllt sind, wird die Applikation ihren Start mit einer Fehlermeldung abbrechen. Eine typische Fehlersituation dafür ist z. B.:

Ein der Applikation per PID übergebenes Konfigurationsobjekt wird nicht gefunden

Eine typische, leicht zu behebbende Ursache für dieses Beispiel wäre ein Schreibfehler bei der PID.

Bei einem schwerwiegenden Fehler wird sich die SWE beenden und taucht dem gemäß nicht mehr in der Prozessliste des Betriebssystems auf.

2.6 Beheben von Fehlern

siehe Diagnosehandbuch

2.7 Unterbrechung oder Beendigung des Betriebs

2.7.1 Voraussetzungen

Der Betrieb kann jederzeit beendet werden. Alle anstehenden und in Bearbeitung befindlichen Aufträge werden abgebrochen.

2.7.2 Unterbrechung des Betriebs

Eine Unterbrechung des Betriebs ist nur durch Beendigung des Betriebs möglich.

2.7.3 Beenden des Betriebs

Zum Beenden des Algorithmus verbindet man sich per Telnet mit der Bedienfunktion (BF) der SWE Start/Stop und beendet den Prozess über die entsprechende BF, oder man beendet den Prozess mittels `kill`.

Zum Beenden des Algorithmus sind folgende Linux Befehle in einem Konsolenfenster einzugeben:

```
ps -ef | grep de.bsvrz.ste.stealgsba.algorithmen.nachlauf.NachlaufAlg
```

Es erscheint eine Liste mit dem gesuchten Prozess. In der Liste wird am Anfang die ID des Prozesses aufgelistet.

```
kill <ID>
```

3 Anwendungshandbuch

Der Algorithmus ist ein reiner Serverprozess, der keine direkten Anwendungsfunktionen besitzt.

4 Diagnosehandbuch

4.1 Benötigte Werkzeuge

Zur Diagnose des Algorithmus werden Betriebssystemmittel und der GTM benötigt.

4.2 Diagnosemöglichkeiten

Es gibt folgende Möglichkeiten um Informationen über den Algorithmus zu erhalten:

1. Prozessliste des Betriebssystems
2. GTM
3. Analyse der Logfiles Analyse der Log-Files

4.2.1 Prozessliste des Betriebssystems

Über die Prozesslisten des Betriebssystems (Linus: ps -ef, Windows: Taskmanager) kann geprüft werden ob der Algorithmus noch aktiv ist.

4.2.2 GTM

Mit Hilfe des GTM kann geprüft werden, ob der Algorithmus (UrsachenEinheiten) noch als Quelle für die Attributgruppe `atg.sswSbaEinzelssituationVorstufe, asp.zustand` angemeldet ist. Ebenso kann überprüft werden, ob sich diese Attributgruppe bei Änderungen der Zustände, noch aktualisiert.

4.2.3 Analyse der Log-Files

In den Logfiles speichert der Algorithmus alle wichtigen Vorkommnisse, die während des Betriebs auftreten. Fehlersituationen können u. U. durch Analyse der Logfiles identifiziert werden.

Da die Einträge im Logfile von der Einstellung des Log-Levels abhängen, kann es vorkommen, dass nicht alle relevanten Meldungen gespeichert werden. Es ist im regulären Betrieb auf Grund der erheblichen Performance-Einbußen nicht möglich, alle Meldungen in den Logfiles zu speichern.

4.2.3.1 Allgemeine Meldungen

Nr.	Meldung	Beschreibung
1.	Es existiert bereits eine Sendeanmeldung für diese Datenidentifikation <Objekt> <Datenidentifikation>	Es wird versucht, eine Datenidentifikation mehrfach als Quelle anzumelden
2.		

Tabelle 4-1: Allgemeine Meldungen

4.2.3.2 Fehlermeldungen

Nr.	Meldung	Beschreibung
1.	Daten nicht zum Empfangen angemeldet <stacktrace>	Fehler bei der Anmeldung einer Datenidentifikation als Empfänger am Datenverteiler.
2.	Daten nicht zum Empfang (Senke) angemeldet <stacktrace>	Fehler bei der Anmeldung einer Datenidentifikation als Senke am Datenverteiler.

Nr.	Meldung	Beschreibung
3.	Daten konnten nicht abgemeldet werden <stacktrace>	Eine angemeldete Datenidentifikation konnte nicht ordnungsgemäß am Datenverteiler abgemeldet werden.

Tabelle 4-2: Fehlermeldungen

5 Anhang

5.1 Verzeichnisstruktur des Algorithmus

Name	Inhalt bzw. Bedeutung
BetrInf_STE_Nachlauf_FREI_V1.0_D2015-03-25.doc	Betriebsinformationen der SWE (diese Datei) als Word Dokument
BetrInf_STE_Nachlauf_FREI_V1.0_D2015-03-25.pdf	Betriebsinformationen der SWE (diese Datei) als Pdf Dokument
de.bsvrz.ste.stealgsba.algorithmen.nachlauf.jar	Jar File des Algorithmus
de.bsvrz.ste.stealgsba.algorithmen.nachlauf-runtime.jar	Runtime-Jar File des Algorithmus
de.bsvrz.ste.stealgsba.algorithmen.nachlauf-doc-api.zip	Dokumentation des Algorithmus (API)
de.bsvrz.ste.stealgsba.algorithmen.nachlauf-doc-design.zip	Dokumentation des Algorithmus (Design)
de.bsvrz.ste.stealgsba.algorithmen.nachlauf-GPL-lizenz.txt	Lizenz
de.bsvrz.ste.stealgsba.algorithmen.nachlauf-Build-Report.txt	Build-Report
de.bsvrz.ste.stealgsba.algorithmen.nachlauf-src.zip	Quelltexte des Algorithmus (Java Dateien)
test	(JUnit Test)

Tabelle 3: Verzeichnisstruktur des Algorithmus

5.2 Startscript für Windowssysteme (exemplarisch)

```
rem @echo off
call einstellungen.bat

title StartStopp

rem Um einzelne Programme in eigenen Console-Fenstern zu starten, kann man
rem einfach das "/b" hinter dem jeweiligen "start" Befehl entfernen

rem StartStopp im Hintergrund starten:
start /b %java% ^
    de.bsvrz.ste.stealgsba.algorithmen.nachlauf.NachlaufAlg ^
    %dav1% ^
    -debugLevelStdErrText=CONFIG ^
    -debugLevelFileText=CONFIG ^
    -Anlage=bw.uz.A8.sba_stgt.streckenBeeinflussungsAnlage.SBA ^
    -AspektVersand=asp.sswSbaNachlauf ^

rem Fenster nicht wieder schließen, damit eventuelle Fehler noch lesbar sind.
pause
```

5.3 einstellungen.bat für Windowssysteme (exemplarisch)

```
@echo off
rem Umlaute richtig darstellen
chcp 1252
rem In den Einstellungen des Konsolenfensters muss für die korrekte Darstellung von
rem Umlaute ausserdem ein anderer Zeichensatz eingestellt werden (z.B. Lucida Console)
echo Bitte zur korrekten Darstellung von Umlauten (öäüßÖÄÜ) den Zeichensatz Lucida Console im
Konsolfenster einstellen
rem #####
rem Globale Einstellungen

rem Mit JAVA_HOME wird das Verzeichnis der lokalen Java-Installation angegeben.
rem Wenn java sich im Suchpfad befindet oder JAVA_HOME systemglobal eingestellt
rem ist, dann muß JAVA_HOME hier nicht spezifiziert werden. JAVA_HOME kann auch zum
rem einfachen umschalten zwischen verschiedenen Java-Umgebungen benutzt werden.
rem set JAVA_HOME=D:\Programme\Java...
set JAVA_HOME=C:\Programme\Java\jre1.6.0_03

rem Mit 'benutzer' wird der Name eines konfigurierten Benutzers spezifiziert unter dem sich
rem Applikationen beim Datenverteiler authentifizieren.
set benutzer=Tester

rem Mit 'dav1Host' wird die IP-Adresse oder der Domainname des ersten Datenverteilers
rem spezifiziert. Der eingestellte Wert wird von Applikationen benutzt, um die Verbindung
rem zum Datenverteiler herzustellen. Wenn der Datenverteiler auf dem lokalen Rechner
rem läuft, dann kann hier auch 'localhost' oder '127.0.0.1' angegeben werden.
set dav1Host=localhost

rem Mit 'dav1DavPort' wird der TCP-Port des ersten Datenverteilers für Verbindungen mit
```

```

rem anderen Datenverteilern spezifiziert. Der eingestellte Wert wird vom ersten Datenvertei-
ler
rem für den passiven Verbindungsaufbau (Server-Socket) benutzt.
set dav1DavPort=8082

rem Mit 'dav1AppPort' wird der TCP-Port des ersten Datenverteilers für Verbindungen mit
rem Applikationen spezifiziert. Der eingestellte Wert wird vom ersten Datenverteiler
rem für den passiven Verbindungsaufbau (Server-Socket) benutzt. Außerdem wird der Wert von
rem Applikationen benutzt, die sich aktiv mit dem ersten Datenverteiler verbinden sollen.
set dav1AppPort=8083

rem 'passwordDatei' spezifiziert eine lokale Datei in dem Applikationen nach dem Passwort
rem des Benutzers für die Authentifizierung beim Datenverteiler suchen.
set passwordDatei=password

rem Die Variable 'cp' spezifiziert den Classpath für die Java Virtual Machine unter der
rem nach dem übersetzten Java-Code gesucht wird.
set cp=^
../distributionspakete/de.bsvrz.dav.daf/de.bsvrz.dav.daf.jar;^
../distributionspakete/de.bsvrz.sys.funclib.application/de.bsvrz.sys.funclib.application.jar;^
../distributionspakete/de.bsvrz.sys.funclib.asyncReceiver/de.bsvrz.sys.funclib.asyncReceiver.j
ar;^
../distributionspakete/de.bsvrz.sys.funclib.commandLineArgs/de.bsvrz.sys.funclib.commandLineAr
gs.jar;^
../distributionspakete/de.bsvrz.sys.funclib.debug/de.bsvrz.sys.funclib.debug.jar;^
../distributionspakete/de.bsvrz.sys.funclib.communicationStreams/de.bsvrz.sys.funclib.communic
ationStreams.jar;^
../distributionspakete/de.bsvrz.sys.funclib.operatingMessage/de.bsvrz.sys.funclib.operatingMes
sage.jar;^
../distributionspakete/de.bsvrz.sys.funclib.crypt/de.bsvrz.sys.funclib.crypt.jar;^
../distributionspakete/de.bsvrz.sys.funclib.concurrent/de.bsvrz.sys.funclib.concurrent.jar;^
../distributionspakete/de.bsvrz.sys.funclib.dataIdentificationSettings/de.bsvrz.sys.funclib.da
taIdentificationSettings.jar;^
../distributionspakete/de.bsvrz.sys.funclib.dataSerializer/de.bsvrz.sys.funclib.dataSerializer
.jar;^
../distributionspakete/de.bsvrz.sys.funclib.timeout/de.bsvrz.sys.funclib.timeout.jar;^
../distributionspakete/de.bsvrz.sys.funclib.filelock/de.bsvrz.sys.funclib.filelock.jar;^
../distributionspakete/de.kappich.pat.configBrowser/de.kappich.pat.configBrowser.jar;^
../distributionspakete/de.bsvrz.pat.sysprot/de.bsvrz.pat.sysprot.jar;^
../distributionspakete/de.bsvrz.pat.sysbed/de.bsvrz.pat.sysbed.jar;^
../distributionspakete/de.kappich.puk.param/de.kappich.puk.param.jar;^
../distributionspakete/de.bsvrz.pat.datgen/de.bsvrz.pat.datgen.jar;^
../distributionspakete/de.bsvrz.pat.onlprot/de.bsvrz.pat.onlprot.jar;^
../distributionspakete/de.bsvrz.sys.funclib.xmlSupport/de.bsvrz.sys.funclib.xmlSupport.jar;^
../distributionspakete/de.bsvrz.sys.funclib.operatingMessage/de.bsvrz.sys.funclib.operatingMes
sage.jar;^
../distributionspakete/
de.bsvrz.ste.stealgsba.algorithmusrahmen/de.bsvrz.ste.stealgsba.algorithmusrahmen.jar;^
../distributionspakete/
de.bsvrz.ste.stealgsba.algorithmen.nachlauf/de.bsvrz.ste.stealgsba.algorithmen.nachlauf.jar
rem Die Variable 'jvmArgs' enthält die Standard-Aufrufargumente der Java Virtual Machine
set jvmArgs=-showversion -Dfile.encoding=ISO-8859-1 -Xms32m

rem #####

```

```

rem Die folgenden Variablen sollten nicht angepasst werden, da sie von den oben definierten
rem Variablen abgeleitet sind.

rem Die Variable 'authentifizierung' enthält die Aufrufargumente, die zur Authentifizierung
rem von Applikationen beim Datenverteiler verwendet werden.
set authentifizierung=-benutzer=%benutzer% -authentifizierung=%passwortdatei%

rem Das debug-Verzeichnis soll ein Verzeichnis höher angelegt werden
set debugDefaults=-debugFilePath=..

rem Die Variable 'dav1' enthält Standard-Argumente für Applikationen, die sich mit dem
rem ersten Datenverteiler verbinden sollen.
set dav1=-datenverteiler=%dav1Host%:%dav1AppPort% %authentifizierung% %debugDefaults%

rem Die Variable 'dav1OhneAuthentifizierung' enthält Standard-Argumente für Applikationen,
rem die sich mit dem
rem ersten Datenverteiler verbinden sollen, ohne Benutzer und Passwortdatei vorzugeben.
set dav1OhneAuthentifizierung=-datenverteiler=%dav1Host%:%dav1AppPort% %debugDefaults%

rem Die Variable 'davleinstellungen' enthält Einstellungen für ersten Datenverteiler selbst.
set davleinstellungen=-davAppPort=%dav1AppPort% -davDavPort=%dav1DavPort% %debugDefaults%

rem Die Variable 'java' enthält den Programmnamen und die Standard-Aufrufargumente
rem der Java Virtual Machine.
if "%JAVA_HOME%" == "" ( set java=java) else set java=%JAVA_HOME%\bin\java
set java=%java% -cp %cp% %jvmArgs%

if "%JAVA_HOME%" == "" ( set javac=javac) else set javac=%JAVA_HOME%\bin\javac

rem echo cp[%cp%] authentifizierung[%authentifizierung%] dav1[%dav1%] java[%java%]

rem Erzeugen von Standard-Verzeichnissen, falls diese noch nicht existieren
if not exist ..\logs mkdir ..\logs

```

5.4 Startscript für Linux-Systeme (exemplarisch)

```

#!/bin/bash
. einstellungen.sh

# StartStopp starten:

$java \
  de.bsvrz.ste.stealgsba.algorithmen.nachlauf.NachlaufAlg \
  ${dav1} \
  -debugLevelStdErrText=CONFIG \
  -debugLevelFileText=CONFIG \
  -Anlage=bw.uz.A8.sba_stgt.streckenBeeinflussungsAnlage.SBA \
  -AspektVersand=asp.sswSbaNachlauf \

# Auf das Ende von allen im Hintergrund gestarteten Prozessen warten

```

wait

5.5 einstellungen.sh für Linux-Systeme (exemplarisch)

```
#####
# Globale Einstellungen

# Mit JAVA_HOME wird das Verzeichnis der lokalen Java-Installation angegeben.
# Wenn java sich im Suchpfad befindet oder JAVA_HOME systemglobal eingestellt
# ist, dann muß JAVA_HOME hier nicht spezifiziert werden. JAVA_HOME kann auch zum
# einfachen umschalten zwischen verschiedenen Java-Umgebungen benutzt werden.
# JAVA_HOME=/usr/lib/java

# Mit 'benutzer' wird der Name eines konfigurierten Benutzers spezifiziert unter dem sich
# Applikationen beim Datenverteiler authentifizieren.
export benutzer=Tester

# Mit 'dav1Host' wird die IP-Adresse oder der Domainname des ersten Datenverteilers
# spezifiziert. Der eingestellte Wert wird von Applikationen benutzt, um die Verbindung
# zum Datenverteiler herzustellen. Wenn der Datenverteiler auf dem lokalen Rechner
# läuft, dann kann hier auch 'localhost' oder '127.0.0.1' angegeben werden.
export dav1Host=localhost

# Mit 'dav1DavPort' wird der TCP-Port des ersten Datenverteilers für Verbindungen mit
# anderen Datenverteilern spezifiziert. Der eingestellte Wert wird vom ersten Datenverteiler
# für den passiven Verbindungsaufbau (Server-Socket) benutzt.
export dav1DavPort=8082

# Mit 'dav1AppPort' wird der TCP-Port des ersten Datenverteilers für Verbindungen mit
# Applikationen spezifiziert. Der eingestellte Wert wird vom ersten Datenverteiler
# für den passiven Verbindungsaufbau (Server-Socket) benutzt. Außerdem wird der Wert von
# Applikationen benutzt, die sich aktiv mit dem ersten Datenverteiler verbinden sollen.
export dav1AppPort=8083

# 'passwordDatei' spezifiziert eine lokale Datei in dem Applikationen nach dem Passwort
# des Benutzers für die Authentifizierung beim Datenverteiler suchen.
export passwordDatei=passwd

rem Die Variable 'cp' spezifiziert den Classpath für die Java Virtual Machine unter der
rem nach dem übersetzten Java-Code gesucht wird.
set cp=^
../distributionspakete/de.bsvrz.dav.daf/de.bsvrz.dav.daf.jar:\
../distributionspakete/de.bsvrz.sys.funclib.application/de.bsvrz.sys.funclib.application.jar:\
../distributionspakete/de.bsvrz.sys.funclib.asyncReceiver/de.bsvrz.sys.funclib.asyncReceiver.jar:\
../distributionspakete/de.bsvrz.sys.funclib.commandLineArgs/de.bsvrz.sys.funclib.commandLineArgs.jar:\
../distributionspakete/de.bsvrz.sys.funclib.debug/de.bsvrz.sys.funclib.debug.jar:\
../distributionspakete/de.bsvrz.sys.funclib.communicationStreams/de.bsvrz.sys.funclib.communicationStreams.jar:\
../distributionspakete/de.bsvrz.sys.funclib.operatingMessage/de.bsvrz.sys.funclib.operatingMessage.jar:\
../distributionspakete/de.bsvrz.sys.funclib.crypt/de.bsvrz.sys.funclib.crypt.jar:\
```

```

../distributionspakete/de.bsvrz.sys.funclib.concurrent/de.bsvrz.sys.funclib.concurrent.jar:\
../distributionspakete/de.bsvrz.sys.funclib.dataIdentificationSettings/de.bsvrz.sys.funclib.dataIdentificationSettings.jar:\
../distributionspakete/de.bsvrz.sys.funclib.dataSerializer/de.bsvrz.sys.funclib.dataSerializer.jar:\
../distributionspakete/de.bsvrz.sys.funclib.timeout/de.bsvrz.sys.funclib.timeout.jar:\
../distributionspakete/de.bsvrz.sys.funclib.filelock/de.bsvrz.sys.funclib.filelock.jar:\
../distributionspakete/de.kappich.pat.configBrowser/de.kappich.pat.configBrowser.jar:\
../distributionspakete/de.bsvrz.pat.sysprot/de.bsvrz.pat.sysprot.jar:\
../distributionspakete/de.bsvrz.pat.sysbed/de.bsvrz.pat.sysbed.jar:\
../distributionspakete/de.kappich.puk.param/de.kappich.puk.param.jar:\
../distributionspakete/de.bsvrz.pat.datgen/de.bsvrz.pat.datgen.jar:\
../distributionspakete/de.bsvrz.pat.onlprot/de.bsvrz.pat.onlprot.jar:\
../distributionspakete/de.bsvrz.sys.funclib.xmlSupport/de.bsvrz.sys.funclib.xmlSupport.jar:\
../distributionspakete/de.bsvrz.sys.funclib.operatingMessage/de.bsvrz.sys.funclib.operatingMessage.jar:\
../distributionspakete/
de.bsvrz.ste.stealgsba.algorithmusrahmen/de.bsvrz.ste.stealgsba.algorithmusrahmen.jar:\
../distributionspakete/
de.bsvrz.ste.stealgsba.algorithmen.nachlauf/de.bsvrz.ste.stealgsba.algorithmen.nachlauf.jar
# Die Variable 'jvmArgs' enthält die Standard-Aufrufargumente der Java Virtual Machine
export jvmArgs="-showversion -Dfile.encoding=ISO-8859-1 -Xms32m"

# #####
# Die folgenden Variablen sollten nicht angepasst werden, da sie von den oben definierten
# Variablen abgeleitet sind.

# Die Variable 'authentifizierung' enthält die Aufrufargumente, die zur Authentifizierung
# von Applikationen beim Datenverteiler verwendet werden.
export authentifizierung="-benutzer=${benutzer} -authentifizierung=${passwortDatei}"

# Das debug-Verzeichnis soll ein Verzeichnis höher angelegt werden
export debugDefaults="-debugFilePath=.."

# Die Variable 'dav1' enthält Standard-Argumente für Applikationen, die sich mit dem
# ersten Datenverteiler verbinden sollen.
export dav1="-datenverteiler=${dav1Host}:${dav1AppPort} ${authentifizierung} ${debugDefaults}"

# Die Variable 'dav1OhneAuthentifizierung' enthält Standard-Argumente für Applikationen, die
# sich mit dem
# ersten Datenverteiler verbinden sollen, ohne Benutzer und Passwortdatei vorzugeben.
export dav1OhneAuthentifizierung="-datenverteiler=${dav1Host}:${dav1AppPort} ${debugDefaults}"

# Die Variable 'davleinstellungen' enthält Einstellungen für ersten Datenverteiler selbst.
export davleinstellungen="-davAppPort=${dav1AppPort} -davDavPort=${dav1DavPort} ${debug-
Defaults}"

# Die Variable 'java' enthält den Programmnamen und die Standard-Aufrufargumente
# der Java Virtual Machine.
if test "${JAVA_HOME}" == "" ;then java=java; else java=${JAVA_HOME}/bin/java; fi
java="$java $jvmArgs"

if test "${JAVA_HOME}" == "" ;then javac=javac; else set javac=${JAVA_HOME}/bin/javac; fi

```

```
export JAVA_HOME
export java
export javac
# echo cp[${cp}] authentifizierung[${authentifizierung}] dav1[${dav1}] java[${java}]

# Erzeugen von Standard-Verzeichnissen, falls diese noch nicht existieren
mkdir -p ../logs
```