



Datenverteiler-Applikationsfunktionen

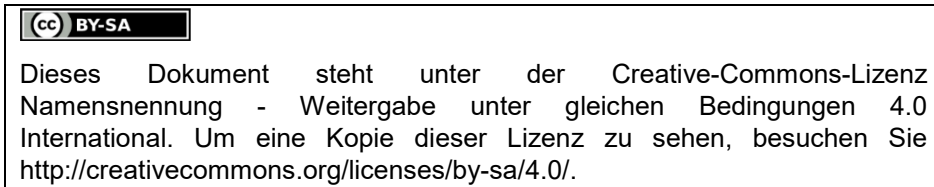
Betriebshandbuch **Anwendungshandbuch** **Diagnosehandbuch**

Version	5.0
Stand	19.04.2017
Produktzustand	akzeptiert
Datei	BetrInf_DaV-DAF_FREI_V5.0_D2017-04-19.docx

Projektkoordinator	NERZ e.V.
Projektleiter	NERZ e.V.
Projektträger	NERZ e.V. www.nerz-ev.de
Ansprechpartner	FTB des NERZ e.V.

0 Allgemeines

0.1 Lizenzen



0.1.1 Ursprüngliche Dokumente und Autoren

Dieses Dokument basiert auf den Inhalten folgender Dokumente (und ggf. Vorgängerversionen):

- BetrInf_SWE1.2_DAF_BLAKE-VRZ_V1.1.doc

Beteiligte Autoren an den Vorgängerdokumenten:

- Gerhard Kappich, Kappich Systemberatung
- Christoph Westermann, Kappich Systemberatung

0.2 Verteiler

Organisationseinheit	Name	Anzahl Kopien	Vermerk
NERZ e.V.		1	

Tabelle 0-1: Dokumentenverteiler

0.3 Änderungsübersicht

Version	Datum	Kapitel	Bemerkungen	Bearbeiter
1.0	19.02.2009		Erstellung des 1. Entwurfs	G. Kappich
1.1	18.11.2009		Korrekturen u.a. entsprechend Anmerkungen von Herrn Frik vom 02.04.2009	C. Westermann
1.1	18.11.2009	2	Text umformuliert, so dass erkennbar ist, dass die beschriebenen Methoden bereits zur Verfügung stehen (war teilweise noch als zu realisierende Anforderungen beschrieben).	
1.1	18.11.2009	2.2.5.1 2.2.7	Methode zur Auswahl von Auswahl von bestimmten Attributen aus einer Attributgruppe gestrichen, da diese Anforderung nicht mehr besteht.	
1.1	18.11.2009	2.4.3 2.4.4	Aktualisierung der SW und Deinstallieren der SW konkretisiert. Es sollte nur das Verzeichnis gelöscht werden, das den Inhalt des Distributionspaketes enthält. Kapitel "Komplettes Entfernen der Installation" gestrichen.	
1.1	18.11.2009	2.4 2.6	Unterbrechung oder Beendigung des Betriebs ergänzt, da bei der Aktualisierung der Datenverteiler-Applikationsfunktionen der Betrieb beendet werden muss.	
1.1	18.11.2009	2.5	Fehlertyp ergänzt	

Version	Datum	Kapitel	Bemerkungen	Bearbeiter
1.9	15.08.2016	alle	Überarbeitung und Ergänzung der BetrInf auf Basis der Anforderungen aus der laufenden Zertifizierung. Grundlage dieses Dokuments waren die aktuellsten veröffentlichten Betriebsinformationen zu dieser SWE, Dokument „BetrInf_SWE9.1_LosB_VRZ3_5.1.doc“	H. C. Kniß (HCK), FTB NERZ
2.0	20.08.2016		Überführung in den Zustand akzeptiert	H. C. Kniß (HCK), FTB NERZ
3.0	31.08.2016		Redaktionelle Überarbeitung, Dokumentenname an Standardvorgabe angepasst.	H. C. Kniß (HCK), FTB NERZ
4.0	19.03.2017	0.7 2.1.3 2.2.3.1.2.1 2.2.2.1.1.1	Korrekturen entsprechend Prüfbericht zur Zertifizierung.	H. C. Kniß (HCK), FTB NERZ
5.0	19.04.2017	0.7 2.2.3.1.1	Korrekturen entsprechend Anmerkungen zur Nachzertifizierung. Folgende Aufrufparameter ergänzt: -debugGui -debugSetLoggerAndLevel	H. C. Kniß (HCK), FTB NERZ

Tabelle 0-2: Änderungsübersicht

0.4 Inhaltsverzeichnis

0	Allgemeines	2
0.1	Lizenzen	2
0.1.1	Ursprüngliche Dokumente und Autoren	2
0.2	Verteiler	2
0.3	Änderungsübersicht.....	2
0.4	Inhaltsverzeichnis	3
0.5	Abkürzungsverzeichnis.....	6
0.6	Definitionen.....	6
0.7	Referenzierte Dokumente / URLs.....	6
0.8	Abbildungsverzeichnis	7
0.9	Tabellenverzeichnis	7
1	Überblick	8
1.1	Inhalt des Dokuments	8
1.2	Kurzbeschreibung der Funktionalität	8
1.3	Aufbau der Datenverteiler-Verbindung	9
1.3.1	Initialisierung der Kommunikationsverbindung.....	9
1.3.2	Initialisierung der Datenverteiler-Verbindung	9
1.3.3	Terminierung der Datenverteiler-Verbindung.....	10
1.3.4	Konfigurationsabfragen	10
1.3.5	Parametrierung des Datenverteilers	11

1.3.5.1	Eingangsdaten der Applikation	11
1.3.5.2	Ausgangsdaten der Applikation	12
1.3.6	Entgegennahme von angemeldeten Daten.....	12
1.3.7	Lesen von Daten	12
1.3.8	Versenden von Daten.....	13
1.3.9	Abmelden von Daten.....	13
1.3.10	Datenrepräsentation	13
1.3.11	Konfigurationsänderungen	13
1.3.12	Verwaltung der Datenverteilerverbindung.....	14
1.3.12.1	Telegrammpuffer und Flusskontrolle	14
1.3.12.2	Priorisierung von Telegrammen	14
1.3.12.3	Kommunikationsüberwachung	14
1.3.12.4	Telegrammlaufzeitermittlung	15
1.3.12.5	Abruf von internen Zustandsinformationen des Datenverteilers	15
2	Betriebshandbuch	15
2.1	Installation der Software	15
2.1.1	Voraussetzungen	15
2.1.1.1	Java	15
2.1.1.2	SWE	16
2.1.2	Laufzeitumgebung.....	16
2.1.3	Benötigte zusätzliche Softwarekomponenten	16
2.1.4	Erstinstallation der Software.....	17
2.1.4.1	Weitere Installationsschritte	17
2.1.5	Kontrolle der Installation.....	17
2.1.6	Deinstallation der Software	17
2.1.7	Aktualisierung der Software	17
2.2	Einrichtung von Software und Datenverteiler	18
2.2.1	Konfiguration	18
2.2.1.1	Konfigurationsbereiche	18
2.2.1.2	Spezifische Konfigurationen	18
2.2.1.3	Parametrierung der Parametrierung	18
2.2.2	Ein- und Ausgangsdaten.....	18
2.2.2.1	Eingangsdaten	18
2.2.2.2	Ausgangsdaten	19
2.2.3	Aufnahme des Betriebs.....	20
2.2.3.1	Startparameter	20
2.2.3.2	Starten der SWE	29
2.3	Überwachung des Betriebs	29
2.3.1	Überwachung des Prozesses	29
2.3.2	Prüfung Störungsfreiheit.....	29
2.3.2.1	Prüfung durch Log-Files	29
2.4	Unterbrechung oder Beendigung des Betriebs	29
2.4.1	Voraussetzungen	29
2.4.2	Unterbrechung des Betriebs	29
2.4.3	Beenden des Betriebs.....	29
3	Anwenderhandbuch	30
4	Diagnosehandbuch.....	30
4.1	Benötigte Werkzeuge	30
4.2	Diagnosemöglichkeiten	30
4.2.1	Betriebsmeldungen	30
4.2.2	Allgemeine Meldungen (Startmeldungen).....	30
4.2.3	Allgemeine Meldungen (Debug-Level WARNING)	30
4.2.4	Allgemeine Meldungen (Debug-Level ERROR).....	32
4.2.5	Überprüfung der Parametrierung	37

NERZ e.V.	BetrInf Datenverteiler-Applikationsfunktionen	Seite: 5 von 37 Version: 5.0 Stand: 19.04.2017
-----------	--	--

	4.2.6 Überprüfung des Prozesszustandes	37
5	Anhang	37
5.1	Verzeichnisstruktur	37

NERZ e.V.	BetrInf Datenverteiler-Applikationsfunktionen	Seite: 6 von 37 Version: 5.0 Stand: 19.04.2017
-----------	--	--

0.5 Abkürzungsverzeichnis

Siehe [AbkBSVRZ].

Darüber hinaus werden folgende Abkürzungen verwendet:

BASt	Bundesanstalt für Straßenwesen
BSVRZ	Basis System VRZ
ERZ	Einheitliche Rechnerzentralensoftware
NERZ	Nutzer der ERZ, siehe auch www.nerz-ev.de
AOE	Autarke Organisationseinheit (der lokale Konfigurationsverantwortliche eines Datenverteilersystems)
KV	Konfigurationsverantwortlicher eines Teils einer Konfiguration
KB	Konfigurationsbereich einer Konfiguration

0.6 Definitionen

Siehe [GlossarBSVRZ].

Darüber hinaus werden folgende Definitionen verwendet:

--- ---

0.7 Referenzierte Dokumente / URLs

Die folgende Tabelle listet die im Dokument verwendeten Referenzen auf. Zum aktuellen Zeitpunkt sind die folgenden Archiv-URLs vorhanden:

- NERZ-Archiv: <http://www.nerz-ev.de/> → Dokumente und Software

[VMOD97]	Der Bundesminister des Inneren, Entwicklungsstandard für IT-Systeme des Bundes Vorgehensmodell, Juni 1997, KBSt, Koordinations- und Beratungsstelle der Bundesregierung für Informationstechnik in der Bundesverwaltung.
[AbkBSVRZ]	Abkürzungsverzeichnis BSVRZ Gesamt NERZ-Archiv: Abk_BSVRZ-Gesamt_FREI_V4.0_D2006-08-15.doc
[GlossarBSVRZ]	Glossar BSVRZ Gesamt NERZ-Archiv: SE-02.0002-Glos-0.4__Glossar__global__.pdf
[JDK]	Übersicht über JavaSE inklusive dem Java SE Development Kit http://www.oracle.com/technetwork/java/javase/overview/index.html
[JRE]	Übersicht über JavaSE inklusive des Java SE Runtime Environments http://www.oracle.com/technetwork/java/javase/overview/index.html
[BetrInf_DaV-DaV]	Betriebsinformationen zum Datenverteiler NERZ-Archiv: BetrInf_DaV-DaV_FREI_V5.0_D2017-04-19.docx/.pdf
[BetrInf_PuK-Konfig]	Betriebsinformationen zur Konfiguration NERZ-Archiv: BetrInf_PuK-Konfig_FREI_V5.0_D2017-04-19.docx/.pdf
[BetrInf_PuK-Param]	Betriebsinformationen zur Parametrierung NERZ-Archiv: BetrInf_PuK-Param_FREI_V9.0_D2017-04-19.docx/.pdf
[BetrInf_Sys-StartStopp]	Betriebsinformationen zu StartStopp NERZ-Archiv: BetrInf_Sys-StartStopp_FREI_V5.0_D2012-09-04.doc

[NERZ]	Homepage des Vereins der "Nutzer der einheitlichen Rechnerzentralensoftware für Verkehrsrechnerzentralen - NERZ e.V." http://nerz-ev.de/
[NERZSoftware]	Softwarearchiv des NERZ e.V. http://nerz-ev.de/produkte/software-dokumente
[DatKatHTML]	Datenkatalog der ERZ-Software http://www.nerz-ev.de/datkat/start.html
[Tutorial_KernSoftware]	Tutorial zur ERZ-Software NERZ-Archiv: Tutorial_BSVRZ-Tutorial-Kernsoftware_FREI_V0.5_D2006-08-09.pdf

0.8 Abbildungsverzeichnis

0.9 Tabellenverzeichnis

Tabelle 0-1: Dokumentenverteiler	2
Tabelle 0-2: Änderungsübersicht.....	3
Tabelle 1-1: Typographie.....	8
Tabelle 1-2: Konventionen.....	8
Tabelle 4-1: Analysewerkzeuge	30
Tabelle 4-2: Warnungen.....	32
Tabelle 4-3: Fehlermeldungen.....	37

1 Überblick

1.1 Inhalt 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.

Folgende Typographie wird verwendet:

<i>Maschinenschrift kursiv</i>	Namen von Dateien, Ordnern und Benutzern
Maschinenschrift	Befehle und Texte die in der Kommandozeile oder einem graphischem Dialog eingegeben werden
Maschinenschrift im Fettdruck	Teil eines Befehls oder Eingabetextes, der individuell angepasst werden muss

Tabelle 1-1: Typographie

Folgende Konventionen werden festgelegt:

<i>\$PROJEKT_HOME</i>	Das Verzeichnis in dem das Projekt installiert ist
-----------------------	--

Tabelle 1-2: Konventionen

1.2 Kurzbeschreibung der Funktionalität

Die Clients des Datenverteilers sind die Applikationen. Da alle Applikationen ein identisches systemtechnisches Verhalten bezüglich des Datenverteilers (Server) besitzen **müssen**, wird der Clientanteil des Segments Datenverteiler in Form einer Softwarebibliothek zur Verfügung gestellt. Dieser Clientanteil des Datenverteilers stellt innerhalb der Applikation das Bindeglied zwischen dem Datenverteiler (Server) und der eigentlichen Applikation dar. Bei der Applikationsentwicklung ist diese Bibliothek einzubinden und die dadurch zur Verfügung gestellte Schnittstelle zum Datenverteiler zu verwenden. Die nachfolgend beschriebene Funktionalität dieser **Datenverteiler-Applikationsfunktionen** ist innerhalb dieser SW-Einheit realisiert:

- Aufbau einer Verbindung zum lokalen Datenverteiler (SW-Einheit „Datenverteiler“).
- Verhandlung der im weiteren Verlauf zu verwendenden Protokollversion
- Authentifizierung der Applikation bzw. des Anwenders gegenüber dem Datenverteiler
- Verhandlung von Verbindungsparametern auf der Verbindung zum Datenverteiler.
- Abfrage von Konfigurationsinformationen
- Übernahme der zu versendenden Daten über das API, Erstellung der festgelegten Telegramme und Versenden dieser Telegramme an die SW-Einheit „Datenverteiler“.
- Entgegennahme von anliegenden Telegrammen (Empfang) und Bereitstellung der Informationen über die API-Schnittstelle.
- Verwaltung und Bereitstellung von historischen Daten
- Automatische Abwicklung der Flusskontrolle zwischen Client und Server.
- Sende- und Empfangspufferverwaltung
- Automatisches Versenden von Keep-Alive Telegrammen an den Server (Datenverteiler)
- Priorisierung von Telegrammen

NERZ e.V.	BetrInf Datenverteiler-Applikationsfunktionen	Seite: 9 von 37 Version: 5.0 Stand: 19.04.2017
-----------	--	--

- Realisation der Darstellungsschicht (OSI 6) zur Konvertierung zwischen maschineninternen Datendarstellungen und einer maschinenunabhängigen Darstellung der Daten zum telegrammbasierten Datenaustausch zwischen den beteiligten Softwareeinheiten auf evtl. unterschiedlichen Rechnerplattformen.
- Pufferung von Telegrammen.
- Abbau einer Verbindung zum Datenverteiler.

1.3 Aufbau der Datenverteiler-Verbindung

Für die Kommunikation mit dem Datenverteiler wird eine logische Datenverteiler-Verbindung eingesetzt, die seitens der Applikation durch ein Objekt dargestellt wird. Das Verbindungsobjekt stellt einerseits eine objektorientierte Programmierschnittstelle für Applikationen zur Verfügung und implementiert andererseits die telegrammbasierten Kommunikationsabläufe mit einem Datenverteiler.

Die zwischen Datenverteiler-Applikationsfunktion und Datenverteiler ausgetauschten Telegramme werden über eine Kommunikationsverbindung übertragen. Als Kommunikationsverbindung wird eine TCP-Verbindung verwendet. Die Integration von anderen Kommunikationsmechanismen muss möglich sein. Daraus ergibt sich, dass die logische Datenverteiler-Verbindung unabhängig von der einzusetzenden Kommunikationsverbindung ist, d.h. eine Umstellung auf einen anderen Kommunikationsmechanismus erfordert keinerlei Änderung in der Implementierung der logischen Datenverteiler-Verbindung.

Es können beliebige Kommunikationsmechanismen eingesetzt werden, die eine gesicherte Übertragung von Daten ermöglichen. Insbesondere müssen Datenübertragungsfehler, verlorengangene Daten und doppelt übertragene Daten erkannt und korrigiert werden. Weiterhin muss sichergestellt sein, dass die Reihenfolge der versendeten Daten beim Empfang erhalten bleibt. Diese Anforderungen werden durch die Datenverteiler-Applikationsfunktionen erfüllt.

Im Allgemeinen verwaltet eine Applikation nur jeweils eine logische Verbindung zu einem Datenverteiler. Für Test- und Überwachungsprogramme ist es allerdings erforderlich, dass mehrere Datenverteiler-Verbindungen aufgebaut werden können. Diese werden dann vollkommen unabhängig voneinander von den Datenverteiler-Applikationsfunktionen verwaltet.

1.3.1 Initialisierung der Kommunikationsverbindung

Die TCP-Kommunikationsverbindung wird unter Angabe von Rechnername und TCP-Portnummer aufgebaut. Bei anderen Kommunikationsmechanismen wird das jeweilige Verbindungsobjekt unter Angabe der dabei relevanten Adressierungsinformationen erzeugt.

1.3.2 Initialisierung der Datenverteiler-Verbindung

Vor dem Aufbau einer logischen Datenverteilerverbindung müssen folgende Objekte durch die Applikation bereitgestellt werden.

- Kommunikationsverbindung
- Authentifizierungsinformationen
- Parameter der logischen Verbindung

Um das zwischen Datenverteiler-Applikationsfunktion und Datenverteiler eingesetzte logische Protokoll änderbar zu halten und um in Falle einer Änderung einen reibungslosen und zeitlich entkoppelten Übergang von einer Protokollversion zur nächsten zu ermöglichen, wird bei der Initialisierung der logischen Verbindung als erstes die Version des zu verwendenden Protokolls verhandelt. Dazu senden die Datenverteiler-Applikationsfunktionen eine priorisierte Liste von unterstützten Versionsnummern aus denen der Datenverteiler im Normalfall die Version der höchsten Priorität auswählt, die auch vom Datenverteiler unterstützt wird. Diese wird dann zu den Datenverteiler-Applikationsfunktionen gesendet.

NERZ e.V.	BetrInf Datenverteiler-Applikationsfunktionen	Seite: 10 von 37 Version: 5.0 Stand: 19.04.2017
-----------	--	---

Nach der Versionsverhandlung erfolgt die Authentifizierung der Applikation bzw. des Anwenders gegenüber dem Datenverteiler. Die Authentifizierung geschieht über einen Namen mit einem zugeordneten Passwort, das beiden Seiten bekannt ist. Zu beachten ist, dass ein Passwort nicht im Klartext übertragen werden darf, sondern nur verschlüsselt. Weiterhin muss sich das verschlüsselte Passwort bei jedem Authentifizierungsvorgang ändern. Zur Verschlüsselung ist der HMAC-MD5 Algorithmus einzusetzen. Andere Verfahren müssen mit minimalem Aufwand integrierbar sein. Diese Anforderungen werden von den Datenverteiler-Applikationsfunktionen erfüllt.

Der Datenverteiler sendet zur Authentifizierung seine eigene eindeutige Identifizierung zusammen mit einem Zufallstext der sich jedes Mal ändert. Die Datenverteiler-Applikationsfunktionen senden den Namen des eingesetzten Authentifizierungsverfahrens und die Benutzerkennung zusammen mit dem über den Zufallstext verschlüsselten Passwort an den Datenverteiler zurück. Bei erfolgreicher Authentifizierung sendet der Datenverteiler eine eindeutige Applikationsidentifikation, die für die Lebenszeit der Datenverteiler-Verbindung gültig ist, zurück.

Im Anschluss daran erfolgt eine Verhandlung von Verbindungsparametern zwischen Datenverteiler-Applikationsfunktionen und Datenverteiler (wie z.B. Timeout-Zeiten). Dabei werden Standardparameter oder von der Applikation vorgegebene Parameter als Vorschläge an den Datenverteiler übertragen. Dieser trifft eine Festlegung der Parameter und überträgt das Ergebnis zurück an die Applikation.

1.3.3 Terminierung der Datenverteiler-Verbindung

Bei Terminierung einer logischen Datenverteiler-Verbindung seitens der Applikation wird ein Terminierungstelegramm an den Datenverteiler gesendet und die Kommunikationsverbindung anschließend geschlossen. Sämtliche in der Datenverteiler-Verbindung gespeicherten Informationen und noch nicht verarbeitete gepufferte Datentelegramme werden gelöscht.

1.3.4 Konfigurationsabfragen

Über eine aufgebaute logische Datenverteiler-Verbindung können sämtliche Konfigurationsinformationen abgefragt werden. Dazu stellen die Datenverteiler-Applikationsfunktionen eine objektorientierte Funktionsschnittstelle zur Verfügung.

Zu den Konfigurationsabfragen gehören:

- Abfrage der im System konfigurierten Objekte (Konfigurationsobjekte) und deren Zusammenhänge
- Abfrage von definierten Attributgruppen, den zugehörigen Attributen und ihren Metainformationen (z.B. Typ, Wertebereich, Einheiten)
- Alle Konfigurationsobjekte werden in einer Applikation durch Stellvertreterobjekte dargestellt. Über diese Stellvertreterobjekte ist der Zugriff auf die einem Objekt zugeordneten Informationen (Daten) möglich.
- Um mehrfache Abfragen der gleichen Konfigurationsdaten zu vermeiden, werden diese von den Datenverteiler-Applikationsfunktionen lokal zwischengespeichert und durch entsprechende interne Anmeldungen auf Änderungstelegramme immer auf dem aktuellen Stand gehalten.
- Für Applikation, die sehr viele Konfigurationsdaten benötigen und keine schnelle Verbindung zur zugeordneten Datenhaltung haben (wie z.B. abgesetzte Bedienstationen), besteht die Möglichkeit, die in den Datenverteiler-Applikationsfunktionen intern gespeicherten Konfigurationsdaten beim Beenden der Applikation persistent zu speichern und beim Neustart wieder einzulesen. Die Applikation stellt jeweils einen Stream (z.B. eine geöffnete Datei) zur Verfügung auf den die Daten geschrieben bzw. von dem die Daten gelesen werden. Die Datenverteiler-Applikationsfunktionen überprüfen beim Einlesen der gespeicherten Daten deren Aktualität und übernehmen nur die Daten in den lokalen Zwischenspeicher, die sich seit der Speicherung nicht verändert haben.

NERZ e.V.	BetrInf Datenverteiler-Applikationsfunktionen	Seite: 11 von 37 Version: 5.0 Stand: 19.04.2017
-----------	--	---

1.3.5 Parametrierung des Datenverteilers

Jede Applikation muss, nachdem sie eine Verbindung zum lokalen Datenverteiler aufgebaut hat, „ihre“ Daten anmelden. Dies bedeutet, dass sie sowohl

- alle Daten, die sie vom Datenverteiler in der Rolle eines Empfängers oder als Senke erhalten will (Eingangsdaten der Applikation), als auch
- alle Daten, die sie in der Rolle eines Senders oder als Quelle potentiell an den Datenverteiler sendet¹ (Ausgangsdaten der Applikation)

vollständig mitteilt.

Die Datenverteiler-Applikationsfunktionen übertragen die Anmeldungen mit den jeweils relevanten Parametern in **System-Parametertelegammen** an den Datenverteiler. Dieser wird damit **parametriert**.

1.3.5.1 Eingangsdaten der Applikation

Bei der Parametrierung der Eingangsdaten spezifiziert eine Applikation, ob sie bestimmte Daten als Empfänger oder als Senke empfangen möchte.

Neben der Identifikation der gewünschten Daten über Konfigurationsobjekt, Attributgruppe, Aspekt und Simulationsvariante kann bei der Anmeldung auf Eingangsdaten optional ein Zeitbereich angegeben werden.

Wenn kein Zeitbereich spezifiziert ist, dann bedeutet dies für die Anmeldung, dass der Applikation die aktuellen Daten der spezifizierten Attributgruppe des angegebenen Objekts zur Verfügung gestellt werden sollen und bis zur expliziten Abmeldung bei Änderungen aktualisiert werden sollen.

Über die Angabe eines Zeitbereichs können die Fälle abgedeckt werden, in denen eine Applikation neben den aktuellen Daten auch Daten aus der unmittelbaren Vergangenheit benötigt, z.B. für die Darstellungen von Online-Histogrammen. Für die Verwaltung dieser Anmeldungen müssen Online-Anmeldungen beim Datenverteiler und Archivabfragen bei der Datenhaltung durchgeführt werden. Die jeweiligen Ergebnisse werden so zusammengestellt, dass die Datensätze in ihrer natürlichen Reihenfolge weitergeleitet werden (kein Datensatz doppelt, keine Lücke und zeitlich monoton steigend).

Auch unabhängig von diesem Fall kann die Applikation über die Angabe eines Vorhaltezeitraums steuern, wie lange die empfangenen Daten in den Datenverteiler-Applikationsfunktionen zwischengespeichert werden.

Weiter kann die Applikation die Datenverteilung so steuern, dass sie jeden gesendeten Datensatz oder nur die Datensätze empfängt, die sich geändert haben. Außerdem ist einstellbar, ob auch als "nachgeliefert" gekennzeichnete Daten empfangen werden sollen.

¹ Auch potentiell zu versendende Daten, zu denen kein Zustand ermittelbar ist, werden angemeldet.

NERZ e.V.	BetrInf Datenverteiler-Applikationsfunktionen	Seite: 12 von 37 Version: 5.0 Stand: 19.04.2017
-----------	--	---

Mit der Anmeldung wird von den Datenverteiler-Applikationsfunktionen ein von der Applikation zur Verfügung gestelltes Benachrichtigungsobjekt verwaltet, über das der Empfang von neuen Daten signalisiert wird.

Mehrfache Anmeldungen der Applikation auf die gleichen Daten (z.B. gleiche Daten in unterschiedlichen Fenstern der Bedienstationen) werden von den Datenverteiler-Applikationsfunktionen aus Sicht der Applikation wie mehrere unabhängige Anmeldungen verwaltet. Die Daten werden aber von den Datenverteiler-Applikationsfunktionen nur einmal beim Datenverteiler angemeldet und die Daten werden bei Aktualisierungen auch nur einmal zu den Datenverteiler-Applikationsfunktionen übertragen. Diese signalisieren der Applikation dann allerdings nicht nur einmal den Empfang neuer Daten, sondern mehrfach (für jede Anmeldung einmal).

1.3.5.2 Ausgangsdaten der Applikation

Bei der Parametrierung der Ausgangsdaten spezifiziert eine Applikation, ob sie die Daten als Sender oder als Quelle bereitstellen möchte.

Die Ausgangsdaten der Applikation werden über die Konfigurationsobjekte und die jeweils relevanten Attributgruppen, Aspekte und Simulationsvarianten identifiziert. Die initialen Daten werden (im Falle einer Quelle) bei dieser Parametrierung von der Applikation zur Verfügung gestellt. Das sind entweder die aktuellen Daten oder – falls diese noch nicht vorliegen – eine entsprechende Kennung.

1.3.6 Entgegennahme von angemeldeten Daten

Wenn vom Datenverteiler Daten bereitgestellt werden, auf die sich eine Applikation zum Empfang angemeldet hat, dann werden die Daten von den Datenverteiler-Applikationsfunktionen beim zugehörigen Konfigurationsobjekt gespeichert. Außerdem wird die Aktualisierungs-Methode des bei der Anmeldung von der Applikation spezifizierten Benachrichtigungsobjekts aufgerufen. Der Aktualisierungsmethode wird die Identifikation der Daten (Konfigurationsobjekt sowie Attributgruppe, Aspekt und optional Simulationsvariante) und der Zeitbereich der empfangenen Daten übergeben. Mit Hilfe dieser Informationen kann die Applikation dann auf die empfangenen Daten zugreifen.

1.3.7 Lesen von Daten

Zum Lesen von Daten wird eine entsprechende API-Funktion der Datenverteiler-Applikationsfunktionen aufgerufen. Dabei übergibt die Applikation die Identifikation der Daten (Konfigurationsobjekt sowie Attributgruppe, Aspekt und optional Simulationsvariante) und einen optionalen Zeitbereich.

Wenn kein Zeitbereich spezifiziert ist, dann wird das aktuelle Datum zur Verfügung gestellt.

Die Datenverteiler-Applikationsfunktionen ermitteln den aktuellen Stand der gewünschten Daten folgendermaßen: Wenn die Daten noch nicht angemeldet sind, wird eine implizite² Anmeldung (als Empfänger) an den Datenverteiler übertragen. Wenn der aktuelle Stand der gewünschten Daten noch nicht in den Datenverteiler-Applikationsfunktionen vorliegt wird darauf gewartet.

Über die Angabe eines Zeitbereichs kann auch auf historische Daten zugegriffen werden. Falls die Daten nicht im Vorhaltezeitraum liegen wird eine Archivanfrage bei dem Archivsystem durchgeführt. Die Ergebnisse der Archivanfrage werden mit den Daten im Vorhaltezeitraum so zusammengestellt, dass die Datensätze in ihrer natürlichen Reihenfolge weitergeleitet werden (kein Datensatz doppelt, keine Lücke und zeitlich monoton steigend).

Weiter kann die Applikation spezifizieren, ob jeder Datensatz oder nur geänderte Datensätze zurückgegeben werden sollen. Außerdem ist einstellbar, ob auch als "nachgeliefert" gekennzeichnete Daten gelesen werden sollen.

² Implizite Anmeldungen beim Lesen von Daten führen dazu, dass bei weiteren Lesevorgängen keine Wartezeiten entstehen. Wenn keine weiteren Lesevorgänge durchgeführt werden, müssen die impliziten Anmeldungen automatisch zurückgenommen werden .

NERZ e.V.	BetrInf Datenverteiler-Applikationsfunktionen	Seite: 13 von 37 Version: 5.0 Stand: 19.04.2017
-----------	--	---

Abschließend werden die Daten von der maschinenunabhängigen Darstellung in das jeweils benötigte Format konvertiert (siehe 1.3.10 „Datenrepräsentation“) und der Applikation zur Verfügung gestellt.

1.3.8 Versenden von Daten

Zum Versenden von Daten ruft eine Applikation eine entsprechende Methode der Datenverteiler-Applikationsfunktionen auf. Dabei wird neben den zu versendenden Daten auch die Identifikation der Daten (Konfigurationsobjekt sowie Attributgruppe, Aspekt und optional Simulationsvariante) übergeben.

Nachgelieferte Daten, die einen älteren Stand als die aktuellen Daten haben, werden beim Versenden der Daten markiert.

Die Datenverteiler-Applikationsfunktionen konvertieren die zu übertragenden Nutzdaten in eine maschinenunabhängige Darstellung (siehe 1.3.10 „Datenrepräsentation“) und übertragen sie mit der Identifikation der Daten und weiteren Parametern in einem Zustandstelegramm an den Datenverteiler.

1.3.9 Abmelden von Daten

Werden bestimmte Daten auf die sich die Applikation angemeldet hat nicht mehr benötigt oder nicht mehr zur Verfügung gestellt, dann kann sie die Daten wieder abmelden. Wenn innerhalb der Applikation keine weitere Anmeldung auf die Daten besteht, dann werden die Daten von den Datenverteiler-Applikationsfunktionen beim Datenverteiler abgemeldet und die im Konfigurationsobjekt zwischengespeicherten Daten wieder freigegeben.

Eine von den Datenverteiler-Applikationsfunktionen eventuell durchgeführte implizite Anmeldung auf Daten wird automatisch wieder zurückgenommen, wenn die Applikation für eine gewisse Zeit die Daten nicht mehr benötigt.

1.3.10 Datenrepräsentation

Die Darstellungsschicht (OSI 6) in den Datenverteiler-Applikationsfunktionen dient zur Konvertierung zwischen maschineninternen Datendarstellungen und einer maschinenunabhängigen Darstellung der Daten zum telegrammbasierten Datenaustausch mit dem Datenverteiler.

Maschinenintern wird eine Darstellung der Daten in Speicherbereichen genutzt. Hierbei sind sowohl Wortlängen als auch Bitreihenfolgen von der eingesetzten Maschine (und dem genutzten Compiler) abhängig. Aus diesem Grund muss für die Kommunikation zwischen verschiedenen Maschinen die Darstellung der verschiedenen Datentypen vereinbart werden. Die Festlegung geschieht nicht für jede Attributgruppe speziell, sondern leitet sich jeweils aus allgemeinen Festlegungen für die verwendeten Grunddatentypen, Listen, Arrays und Sequenzen ab.

Vor dem Versenden von Daten werden diese durch die Datenverteiler-Applikationsfunktionen in die maschinenunabhängige Darstellung konvertiert und nach dem Empfang von Daten werden diese in die jeweilige maschineninterne Darstellung konvertiert.

1.3.11 Konfigurationsänderungen

Die Konfiguration von bestimmten Objekten kann Online geändert werden. Beispiele für diese dynamischen Objekte sind Benutzergruppen, Sonderprogramme, Staus etc. Diese Objekte können zur Laufzeit des Systems erzeugt, gelöscht und umbenannt werden. Dazu stellen die Datenverteiler-Applikationsfunktionen entsprechende API-Funktionen zur Verfügung.

Damit Applikationen, die von diesen Änderungen der Konfiguration abhängig sind, über den jeweils aktuellen Zustand der Konfiguration informiert sind, besteht die Möglichkeit, sich auf die oben genannten Modifikationen anzumelden. Nach Aufruf der entsprechenden API-Funktion wird die Applikation von den Datenverteiler-Applikationsfunktionen bei Konfigurationsänderungen benachrichtigt.

NERZ e.V.	BetrInf Datenverteiler-Applikationsfunktionen	Seite: 14 von 37 Version: 5.0 Stand: 19.04.2017
-----------	--	---

1.3.12 Verwaltung der Datenverteilerverbindung

1.3.12.1 Telegrammpuffer und Flusskontrolle

Um einen weitgehend asynchronen Datenaustausch zwischen Applikation und Datenverteiler und damit eine zeitliche Entkopplung der Kommunikationspartner zu erreichen, werden zu sendende und empfangene Telegramme in einem Pufferspeicher zwischengespeichert, bis sie weiterverarbeitet werden können.

Zu beachten ist allerdings, dass die Pufferspeicher in Ihrer Größe beschränkt sind. Wenn eine Applikation Daten schneller versendet, als die empfangende Applikation diese Daten verarbeiten kann, dann werden sich die Sende- und Empfangspuffer füllen. Bei vollen Pufferspeichern wird mit einer Flusskontrolle gewährleistet, dass keine Daten verloren gehen. Bei vollem Empfangspuffer werden keine neuen Daten mehr entgegengenommen, durch die Flusskontrolle der eingesetzten Kommunikationsverbindung wird dann der Sendepuffer des sendenden Prozesses gefüllt. Wenn dieser auch voll ist, dann blockiert die Sendefunktion.

Wenn der Sendepuffer eine parametrierbare Zeit einen parametrierbaren Füllstand überschreitet, dann wird zyklisch eine Durchsatzprüfung durchgeführt, um eventuell vorhandene Probleme der Kommunikation bzw. des Empfängers zu erkennen. Ist der ermittelte Durchsatz über parametrierbare Zeitspanne unterhalb einer parametrierbaren Grenze, wird die Verbindung terminiert und eine entsprechende Meldung generiert.

1.3.12.2 Priorisierung von Telegrammen

Die Datenverteiler-Applikationsfunktionen unterstützen die priorisierte Verarbeitung von Telegrammen. Die Telegramme werden dazu entsprechend ihrer Priorität in den Sendepuffer bzw. Empfangspuffer eingetragen. Dabei werden Telegramme mit höherer Priorität an den Anfang gestellt, wobei Telegramme mit gleicher Priorität ihre Sende- bzw. Empfangsreihenfolge beibehalten. Innerhalb einer Prioritätsstufe ist die natürliche Reihenfolge zu garantieren (FIFO-Prinzip), ansonsten werden die Telegramme entsprechend ihre Prioritätenklasse abgearbeitet.

Die Priorisierung erfolgt nicht durch die Applikation, sondern wird über Telegrammklassen³ als konfigurierendes Datum des Datenvertailers global festgelegt. Die Festlegung **ist systemweit eindeutig!**

1.3.12.3 Kommunikationsüberwachung

Die verwendete Kommunikationsverbindung stellt eine gesicherte Übertragung von Daten zur Verfügung. Eine Unterbrechung der Kommunikationsstrecke ist bei einigen Kommunikationsmechanismen nicht unterscheidbar von dem Fall das aktuell keine Daten übertragen werden.

Aus diesem Grund wird ein Keep-Alive Mechanismus eingesetzt. Dabei werden sendeseitig Keep-Alive-Telegramme versendet, wenn eine parametrierbare Zeit (`timeoutSendeKeepAlive`) keine Nutzdaten übertragen wurden. Der Empfänger terminiert die Verbindung, wenn er eine parametrierbare Zeitspanne (`timeoutEmpfangeKeepAlive`) keine Telegramme auf der Verbindung empfangen hat.

Die Parameter `timeoutSendeKeepAlive` und `timeoutEmpfangeKeepAlive` werden in der Initialisierungsphase zwischen Datenverteiler-Applikationsfunktionen und Datenverteiler verhandelt, damit beide Kommunikationspartner zueinander passende Parameter für den Keep-Alive Mechanismus benutzen.

³ Online-Daten, nachgelieferte Daten, aus dem Archiv abgefragte Daten und Systemtelegramme sind Beispiele für Telegramme unterschiedlicher Telegrammklassen.

NERZ e.V.	BetrInf Datenverteiler-Applikationsfunktionen	Seite: 15 von 37 Version: 5.0 Stand: 19.04.2017
-----------	--	---

1.3.12.4 Telegrammlaufzeitermittlung

Neben diesen Verwaltungsfunktionen gibt es eine Funktion zur Ermittlung von Laufzeiten, die von der Applikation bei bestehender Verbindung zum Datenverteiler aufgerufen werden kann. Zur Messung der Telegrammlaufzeiten wird ein spezielles Telegramm übertragen, das von der Gegenseite quittiert wird. Durch Messung der Zeit zwischen dem Absenden des Telegramms und dem Empfang der Antwort kann die Telegrammlaufzeit ermittelt werden. Diese Information kann von einer Applikation z.B. für Überwachungs- oder Debug-Zwecke benutzt werden.

1.3.12.5 Abruf von internen Zustandsinformationen des Datenverteilers

Die vom Datenverteiler bereitgestellten internen Zustandsinformationen können von einer Applikation z.B. für Überwachungs- oder Debug-Zwecke abgefragt werden.

2 Betriebshandbuch

Im Rahmen der Installation, der Einrichtung und des Betriebs der SWE sind folgende Schritte durchzuführen:

- Installation der Software
- Einrichtung von Software und Datenverteiler
 - Konfiguration und Parametrierung
 - Aufnahme des Betriebs
- Überwachung des Betriebs
- Unterbrechung und Beendigung des Betriebs

Diese Schritte werden in den folgenden Abschnitten genau beschrieben.

2.1 Installation der Software

2.1.1 Voraussetzungen

2.1.1.1 Java

Es muss eine Java Runtime Umgebung (JRE) ab Version 8.0⁴ oder höher installiert und über den Suchpfad auffindbar sein [jre].

Das Java Runtime Umgebung (JRE) ist für den Betrieb ausreichend, jedoch bietet das Java Development Kit (JDK) zusätzlich nützliche Tools für die Diagnose [jdk].

Ein korrektes Funktionieren von Java lässt sich auf der Kommandozeile mit folgendem Befehl überprüfen:

```
java -version
```

Erfolgt die Ausgabe der installierten Javaversion für den Befehl `java -version` ist der Pfad korrekt eingerichtet.

Erfolgt eine Meldung, dass der Befehl nicht gefunden wurde, muss die Pfadvariable angepasst werden.

Unter Linux-Systemen (unter anderem Linux, Mac OS X) kann dies mit folgendem Kommando erfolgen:

```
export PATH=$PATH:/pfad_zu_java/bin
```

⁴ Der aktuelle Stand dieser BetrInf geht davon aus, dass die Kernsoftware V3.8.x oder höher eingesetzt wird. Diese erfordert ein JRE ab Version 8.

NERZ e.V.	BetrInf Datenverteiler-Applikationsfunktionen	Seite: 16 von 37 Version: 5.0 Stand: 19.04.2017
-----------	--	---

Unter Windows muss der Pfad im Dialog *Systemsteuerung/System/Erweitert/Umwgebungsvariablen* angepasst werden. Der Wert der Variablen `PFAD` muss um den Text `;/pfad_zu_java/bin` ergänzt werden.

Im Folgenden wird davon ausgegangen, dass die Projektumgebung (Kernsoftware und weitere SWE) im Ordner `$PROJEKT_HOME` installiert ist.

Zusätzlich benötigte Komponenten: siehe Kapitel 2.1.3.

2.1.1.2 SWE

Die SWE ist Bestandteil des sogenannten „Kernsystems“ und wird i.d.R. als Bestandteil dieses Kernsystems bereitgestellt.

Die Kernsoftware wird als ZIP-Archiv ausgeliefert, dessen Dateiname dem Muster

- `SWE_KernSoftware_FREI_V3.8.0_D2016-05-31_Alle-SWE.zip`

entspricht (Status, Version und Datum entsprechen dem aktuellen Stand).

Wird die Kernsoftwareauslieferung als Basis der Installation verwendet, so ist das obige Paket mit einem Entpackprogramm zu entpacken (siehe dazu auch Kapitel 2.1.4ff). Im entpackten Ordner befindet sich dann die SWE als ZIP-Archiv mit folgendem Namen:

- `de.bsvrz.dav.daf.zip`

Die SWE wird teilweise auch einzeln als ZIP-Archiv ausgeliefert, dessen Dateiname dem Muster

- `SWE_de.bsvrz.dav.daf_FREI_V3.8.0_D2016-05-31.zip`

entspricht (Status, Version und Datum entsprechen dem aktuellen Stand).

Die jeweils aktuellste Version steht auf dem Server des Vereins der „Nutzer der einheitlichen Rechnerzentralensoftware für Verkehrsrechnerzentralen - NERZ e.V.“ [NERZ] als Distributionspaket unter [NERZSoftware] zur Verfügung und kann dort herunter geladen werden.

Es werden Windows- und Linux-basierte Systeme unterstützt.

2.1.2 Laufzeitumgebung

- Die SWE benötigt einen laufenden Datenverteiler⁵, mit dem sich die SWE verbindet. Der Datenverteiler, bzw. die Verbindungsparameter zu diesem, werden über die Startparameter der SWE festgelegt (siehe Kapitel 2.2.3.1).
- Die SWE benötigt eine Java SE Runtime Environment [jre] (siehe Kapitel 2.1.1.1).

2.1.3 Benötigte zusätzliche Softwarekomponenten

Folgende Pakete der ERZ-Software⁶ müssen in der aktuellen Version⁷ installiert⁸ sein:

⁵ Benötigt wird minimal eine gestartete Kernsystemumgebung, bestehend aus Datenverteiler, Konfiguration, Parametrierung (sowie der Betriebsmeldungsverwaltung). Anmerkung: Die Betriebsmeldungsverwaltung ist zum Betrieb nicht zwingend notwendig. Da aber fast alle Applikationen Betriebsmeldungen erzeugen, ist die Nutzung als Standardanwendung sinnvoll.

⁶ Die entsprechenden Pakete sind analog zur beschriebenen SWE auf dem NERZ-Server verfügbar (siehe Kapitel 2.1.1.2). Abhängigkeiten dieser Pakete zu eventuell weiteren Paketen sind hier nicht aufgeführt. Diese sind bei den entsprechenden Paketen dokumentiert und müssen ggf. ebenfalls installiert sein.

⁷ Sind keine speziellen Angaben zur zu verwendenden Version vorhanden, ist jeweils die aktuellste auf dem NERZ-Server verfügbare Version zu verwenden.

NERZ e.V.	BetrInf Datenverteiler-Applikationsfunktionen	Seite: 17 von 37 Version: 5.0 Stand: 19.04.2017
-----------	--	---

- *de.bsvrz.sys.funclib.commandLineArgs*
- *de.bsvrz.sys.funclib.communicationStreams*
- *de.bsvrz.sys.funclib.concurrent*
- *de.bsvrz.sys.funclib.crypt*
- *de.bsvrz.sys.funclib.dataSerializer*
- *de.bsvrz.sys.funclib.debug*
- *de.bsvrz.sys.funclib.filelock*
- *de.bsvrz.sys.funclib.hexdump*
- *de.bsvrz.sys.funclib.timeout*

2.1.4 Erstinstallation der Software

Der Inhalt des ZIP-Archivs der SWE muss in das Verzeichnis *\$PROJEKT_HOME/lib*⁹ kopiert werden.

Unter Linux-Systemen wird das ZIP-Archiv mit

```
unzip SWE_de.bsvrz.dav.daf_FREI_V3.8.0_D2016-05-31.zip
```

entpackt und mit

```
cp -r de.bsvrz.dav.daf $PROJEKT_HOME/lib
```

in den Ordner mit den Bibliotheken des Projekts kopiert.

Unter Windows kann ab Windows XP der Windows-Explorer sowohl für das Entpacken, als auch für das Kopieren verwendet werden. Für ältere Windows-Systeme muss ein zusätzliches Tool zum Entpacken des ZIP-Archivs verwendet werden (z. B. das kostenlose 7-Zip <http://7-zip.org>).

2.1.4.1 Weitere Installationsschritte

keine

2.1.5 Kontrolle der Installation

Nach erfolgreicher Installation wurde dem Ordner *\$PROJEKT_HOME/lib* ein Unterordner

- *de.bsvrz.dav.daf*

hinzugefügt und der Unterordner bzw. sein Inhalt entspricht der Struktur im Anhang 5.1.

2.1.6 Deinstallation der Software

Für die Deinstallation muss die SWE gestoppt werden.

Zur Deinstallation der Software werden die Dateien und Verzeichnisse, die in Kapitel 2.1.4 installiert bzw. kopiert wurden, gelöscht.

Gleiches gilt für Dateien und Verzeichnisse, die bei der Installation angelegt bzw. kopiert wurden.

2.1.7 Aktualisierung der Software

Für die Aktualisierung muss die SWE gestoppt werden.

Die Aktualisierung entspricht der Deinstallation und anschließender Erstinstallation der SWE.

⁸ Die Pakete werden genauso wie die hier beschriebenen SWE als ZIP-Archive auf dem NERZ-Server bereitgestellt und sind analog wie in Kapitel 2.1.4 zu installieren.

⁹ Welches Verzeichnis verwendet wird, hängt von der jeweiligen Implementierung des Projektordners ab (ist also anwenderspezifisch). Im Folgenden wird bei den Beispielkommandos immer der Ordner *\$PROJEKT_HOME/lib* verwendet.

2.2 Einrichtung von Software und Datenverteiler

2.2.1 Konfiguration

2.2.1.1 Konfigurationsbereiche

Im Folgenden sind die von der SWE benötigten Konfigurationsbereiche¹⁰ aufgeführt. Diese müssen im Datenverteiler, mit dem sich die SWE verbindet, vorhanden sein¹¹:

- Für die Nutzung der Datenverteiler-Applikationsfunktionen werden keine zusätzlichen Konfigurationsbereiche benötigt.

2.2.1.2 Spezifische Konfigurationen

Für die Nutzung der Datenverteiler-Applikationsfunktionen werden keine weiteren spezifischen Konfigurationsbereiche benötigt.

2.2.1.3 Parametrierung der Parametrierung

Die Datenverteiler-Applikationsfunktionen benötigt keine zusätzliche Parametrierung.

2.2.2 Ein- und Ausgangsdaten

Die Datenverteiler-Applikationsfunktionen, welche die Schnittstelle zwischen den Applikationen und dem Datenverteiler darstellt, verarbeitet als Ein- und Ausgangsdaten grundsätzlich alle möglichen Attributgruppen-Aspektkombinationen, die im Datenkatalog definiert sind (siehe [DatKatHTML]).

Nachfolgend sind deshalb nur die Ein- und Ausgangsdaten aufgeführt, die bei jeder SWE, welche die Datenverteiler-Applikationsfunktionen verwendet, automatisch beim Datenverteiler angemeldet werden. Dies sind die Attributgruppen, die für die systeminterne Kommunikation (Konfigurationsanfragen etc.) zwischen Datenverteiler-Applikationsfunktionen auf der einen und Datenverteiler und Konfiguration auf der anderen Seite benötigt werden.

2.2.2.1 Eingangsdaten

2.2.2.1.1 Datenverteilerschnittstelle

2.2.2.1.1.1 Senke

Folgende Daten werden von allen SWE, welche die Datenverteiler-Applikationsfunktionen verwenden, als **Senke** vom Datenverteiler empfangen:

PID Objekttyp	PID Attributgruppe	PID Aspekt
PID KB-PID KV		
typ.applikation	atg.konfigurationsAnfrageSchnittstelleLesend atg.konfigurationsAnfrageSchnittstelleSchreibend atg.konfigurationsAntwort atg.konfigurationsBenutzerverwaltungsAnfrageSchnittstelle atg.konfigurationsBereichsverwaltungsAnfrageSchnittstelle	asp.antwort

¹⁰ Aufgeführt sind hier nur die Konfigurationsbereiche, die durch den Betrieb dieser SWE zusätzlich zu den Konfigurationsbereichen benötigt werden, welche durch die minimal notwendige Kernsystemumgebung, bestehend aus Datenverteiler, Konfiguration, Parametrierung (sowie der Betriebsmeldungsverwaltung) bereits verfügbar ist. Anmerkung: Die Betriebsmeldungsverwaltung ist zum Betrieb nicht zwingend notwendig. Da aber fast alle Applikationen Betriebsmeldungen erzeugen, ist die Nutzung als Standardanwendung sinnvoll.

¹¹ Die Integration von Konfigurationsbereichen in die Konfiguration eines Datenverteilersystems ist in den [BetrInf_PuK-Konfig] beschrieben.

atg.konfigurationsSchreibAntwort
kb.systemModellGlobal - kv.kappich

Erläuterung

Attributgruppen zur Kommunikation mit der Konfiguration (Details zu Inhalten siehe [DatKatHTML]).

Angaben zu den Objekten

Das Applikationsobjekt der SWE, welches die DAF verwendet

2.2.2.1.1.2 Empfänger

Folgende Daten werden von allen SWE, welche die Datenverteiler-Applikationsfunktionen verwenden, als **Empfänger** vom Datenverteiler empfangen:

- keine

2.2.2.2 Ausgangsdaten

2.2.2.2.1 Datenverteilerschnittstelle

2.2.2.2.1.1 Quelle

Folgende Daten werden von allen SWE, welche die Datenverteiler-Applikationsfunktionen verwenden, als **Quelle** an den Datenverteiler publiziert:

PID Objekttyp	PID Attributgruppe	PID Aspekt
PID KB-PID KV		
typ.applikation	atg.applikationsFertigmeldung	asp.standard
kb.systemModellGlobal - kv.kappich		

Erläuterung

Fertigmeldung der Applikation (wird insbesondere von StartStopp ausgewertet)

Angaben zu den Objekten

Das Applikationsobjekt der SWE, welches die DAF verwendet

2.2.2.2.1.2 Sender

Folgende Daten werden von allen SWE, welche die Datenverteiler-Applikationsfunktionen verwenden, als **Sender** an den Datenverteiler publiziert:

PID Objekttyp	PID Attributgruppe	PID Aspekt
PID KB-PID KV		
typ.konfigurationsVerantwortlicher		asp.anfrage
	atg.konfigurationsAnfrage	
	atg.konfigurationsAnfrageSchnittstelleLesend	
	atg.konfigurationsAnfrageSchnittstelleSchreibend	
	atg.konfigurationsBenutzerverwaltungsAnfrageSchnittstelle	
	atg.konfigurationsBereichsverwaltungsAnfrageSchnittstelle	
	atg.konfigurationsSchreibAnfrage	

kb.systemModellGlobal - kv.kappich
kb.metaModellGlobal - kv.kappich

Erläuterung

NERZ e.V.	BetrInf Datenverteiler-Applikationsfunktionen	Seite: 20 von 37 Version: 5.0 Stand: 19.04.2017
-----------	--	---

Attributgruppen zur Kommunikation mit der Konfiguration (Details zu Inhalten siehe [DatKatHTML]).

Angaben zu den Objekten

Konkretes Objekt vom `typ.konfigurationsVerantwortlicher`. I.d.R. ist dies der lokale Konfigurationsverantwortliche des Systems (also die AOE).

2.2.3 Aufnahme des Betriebs

2.2.3.1 Startparameter

2.2.3.1.1 Debug-Ausgaben

Die SWE verwendet (wie im prinzip alle anderen SWE auch) die Funktionsbibliothek zur Ausgabe von Debug-Meldungen und unterstützt damit die folgenden Aufrufparameter.

--debugFilePath = Ordner

Angabe des Verzeichnisses, in dem die Debugausgaben abgelegt werden

Anzugeben ist ein Verzeichnis. Innerhalb dieses Verzeichnisses wird automatisch ein Unterordner „debug“ angelegt (sofern noch nicht vorhanden), in den dann die eigentlichen Debugdateien geschrieben werden

Default-Wert: .

Optional: ja

Beispiel:

```
-debugFilePath=..\..
```

-debugLevelStdErrText = Level

Festlegung des Ausgabelevels für Debugausgaben auf dem Standard-Error-Kanal

Über diese Aufrufparameter wird festgelegt, bis zu welchem Ausgabelevel (bis zu welchem Informationsniveau sollen Meldungen ausgegeben werden) die Debugausgaben auf dem Standard-Error-Kanal (i.d.R. ist dies das Konsolenfenster, aus dem die SWE gestartet wurde) ausgegeben werden sollen.

Mögliche Werte für den Level sind:

- **ERROR:** ist der höchste Level. Verwendung ausschließlich für „echte“ Fehler.
- **WARNING:** Verwendung nur für Warnungen, die vom Programm zwar noch abgefangen werden können aber unbedingt behoben werden müssen.
- **CONFIG:** Verwendung für Konfigurationsinformationen (z.B. angemeldete Objekte etc.).
- **INFO:** Verwendung für Infoausgaben (z.B. Status des Programms, verwendete Startparameter etc.).
- **FINE:** Verwendung für programmnahe Ausgaben zur Verfolgung des Programmablaufs
- **FINER:** wie FINE aber mehr Details
- **FINEST:** wie FINER aber mit allen Details
- **ALL:** schaltet die Ausgabe aller Level ein
- **OFF:** schaltet die Ausgabe aller Level aus

NERZ e.V.	BetrInf Datenverteiler-Applikationsfunktionen	Seite: 21 von 37 Version: 5.0 Stand: 19.04.2017
-----------	--	---

Default-Wert: WARNING

Optional: ja

Beispiel:

`-debugLevelStdErrText=INFO`

Es werden alle Debug-Meldungen bis zum Level INFO (einschließlich, also ERROR, WARNING, CONFIG und INFO) ausgegeben.

-debugLevelFileText = Level

Festlegung des Ausgabelevels für Debugausgaben als Text-Datei

Wie bei `-debugLevelStdErrText`, allerdings erfolgt die Ausgabe in eine Textdatei in den über den Parameter `-debugFilePath` festgelegten Debug-Ordner mit dem Namen

- `NameDerApplikation-NrInkarnation-lfdNr.log.txt`
- **Beispiel:**
 - `BetriebsMeldungApp-0-0.log.txt`

NameDerApplikation:

Name der SWE

NrInkarnation:

Wird nur eine Inkarnation gestartet, ist diese Nr. i.d.R. gleich 0, bei mehreren Inkarnationen werden diese durchnummeriert.

lfdNr:

Die Ausgabedateien werden bei Erreichen einer bestimmten Größe (siehe Aufrufparameter `-debugFileSize`) abgeschlossen und es wird jeweils eine weitere Datei angelegt und durchnummeriert. Die Anzahl der Dateien, die wie ein Ringbuffer reihum geschrieben werden, wird durch den Aufrufparameter `-debugFileSize` festgelegt. Die Datei mit der `lfdNr = 0` ist dabei immer die aktuellste (gerade in Bearbeitung befindliche Datei), die Datei mit der `lfdNr = 1` ist die zeitlich davorliegende Ausgabedatei usw.

Default-Wert: OFF

Optional: ja

Beispiel:

`-debugLevelFileText=CONFIG`

-debugLevelFileXML = Level

Festlegung des Ausgabelevels für Debugausgaben als XML-Datei

Wie bei `-debugLevelFileText`, allerdings erfolgt die Ausgabe im XML-Format in eine Datei mit dem Namen

- `NameDerApplikation-NrInkarnation-lfdNr.log.xml`

NERZ e.V.	BetrInf Datenverteiler-Applikationsfunktionen	Seite: 22 von 37 Version: 5.0 Stand: 19.04.2017
-----------	--	---

Default-Wert: OFF

Optional: ja

Beispiel:

`-debugLevelFileXML=CONFIG`

Hinweis: Die aktuelle (Dummy-)Implementierung erzeugt lediglich eine Textdatei (kein XML) mit obigem Namen mit der Endung `.xml`

`-debugLevelFileExcel` = *Level*

Festlegung des Ausgabelevels für Debugausgaben als CSV-Datei

Wie bei `-debugLevelFileText`, allerdings erfolgt die Ausgabe im CSV-Format in eine Datei mit dem Namen

- `NameDerApplikation-NrInkarnation-lfdNr.log.csv`

Default-Wert: OFF

Optional: ja

Beispiel:

`-debugLevelFileExcel=CONFIG`

`-debugLevelFileHTML` = *Level*

Festlegung des Ausgabelevels für Debugausgaben als HTML-Datei

Wie bei `-debugLevelFileText`, allerdings erfolgt die Ausgabe im HTML-Format in eine Datei mit dem Namen

- `NameDerApplikation-NrInkarnation-lfdNr.log.html`

Default-Wert: OFF

Optional: ja

Beispiel:

`-debugLevelFileHTML=CONFIG`

Hinweis: Die aktuelle (Dummy-)Implementierung erzeugt lediglich eine Textdatei (kein HTML) mit obigem Namen mit der Endung `.html`

`-debugSetLoggerAndLevel` = *Name:Level*

Erzeugt einen zusätzlichen Logger mit dem Namen `Name` und setzt diesen auf den Debuglevel `Level`

Default-Wert: es wird kein zusätzlicher Logger erstellt

Optional: ja

Beispiel:

`-debugSetLoggerAndLevel=MeinLogger:CONFIG`

NERZ e.V.	BetrInf Datenverteiler-Applikationsfunktionen	Seite: 23 von 37 Version: 5.0 Stand: 19.04.2017
-----------	--	---

-debugFileSize = Zahl

Ungefähre maximale Größe einer Ausgabedatei.

Default-Wert: 10000000

Optional: ja

Beispiel:

`-debugFileSize=30000000`

-debugFileCount = Zahl

Anzahl der Ausgabedateien, die wie ein Ringpuffer beschrieben werden.

Default-Wert: 5

Optional: ja

Beispiel:

`- debugFileCount=10`

-debugStdErrTextStacktraces = Boolescher Wert

Gibt wahlweise Stacktraces auf dem Kanal debugLevelStdErrText aus.

Ist mit einer Debug-Log-Meldung eine Exception assoziiert, wird diese auf dem debugLevelStdErr-Kanal ausgegeben, sofern diese Option eingeschaltet ist.

Wert:

`ja|wahr|an|true|yes|on` → Ausgabe ist eingeschaltet

`nein|falsch|aus|false|no|off` → Ausgabe ist ausgeschaltet

`auto` → Ausgabe wird eingeschaltet auch wenn Debug-Level auf OFF steht.

Default-Wert: auto

Optional: ja

Beispiel:

`-debugStdErrTextStacktraces=aus`

-debugGui = Boolescher Wert

Öffnet eine einfache Grafikoberfläche zu Einstellung der Debuglevel zur Laufzeit

Wert:

`ja|wahr|an|true|yes|on` → GUI wird gestartet

`nein|falsch|aus|false|no|off` → Ausgabe ist ausgeschaltet

Default-Wert: nein

Optional: ja

Beispiel:

NERZ e.V.	BetrInf Datenverteiler-Applikationsfunktionen	Seite: 24 von 37 Version: 5.0 Stand: 19.04.2017
-----------	--	---

-debugGui=ja

2.2.3.1.2 Datenverteiler-Applikationsfunktionen (DAF)

Alle SWE, welche die Datenverteiler-Applikationsfunktionen zur Kommunikation mit dem Datenverteiler verwenden, verfügen zusätzlich zu den bei der jeweiligen SWE dokumentierten Aufrufparametern, noch über folgende Aufrufparameter für die Datenverteiler-Applikationsfunktionen:

2.2.3.1.2.1 Allgemeine Aufrufparameter für die Datenverteiler-Applikationsfunktionen

-datenverteiler = *IP-Adresse:Port*

IP-Adresse und Port, mit dem die Applikation (welche die Datenverteiler-Applikationsfunktionen nutzt) sich mit dem Datenverteiler verbindet.

Die Angabe besteht aus den durch einen Doppelpunkt getrennten Angaben zur IP-Adresse und zum zu verwendenden Port.

Default-Wert: localhost:8083

Optional: ja

Beispiel:

-datenverteiler=198.168.7.5:8092

-benutzer = *Text*

Name (Benutzerkennung) des Benutzers, mit dem die Applikation gestartet wird

Benutzerkennung des Benutzers, mit dem die Applikation gestartet wird, welche die Datenverteiler-Applikationsfunktionen nutzt.

Default-Wert: ---

Optional: nein

Beispiel:

-benutzer=nerzdav

-authentifizierung = *Dateiname*

Authentifizierungsinformationen zu den Benutzern

Pfadangabe zu der Datei, in der das zugehörige Passwort abgelegt ist (i.d.R. die Datei „passwd“). Die angegebene Datei ist eine einfache Textdatei, in der zeilenweise die Paare

Benutzer1=PasswortFuerBenutzer1

Benutzer2=PasswortFuerBenutzer2

...

abgelegt sind.

Default-Wert: ---

Optional: nein

Beispiel:

-authentifizierung=passwd

NERZ e.V.	BetrInf Datenverteiler-Applikationsfunktionen	Seite: 25 von 37 Version: 5.0 Stand: 19.04.2017
-----------	--	---

2.2.3.1.2.2 Spezielle Aufrufparameter für die Datenverteiler-Applikationsfunktionen

Nachfolgend sind sehr spezielle Aufrufparameter angegeben, die im Regelfall nicht verwendet werden (müssen).

-authentifizierungsVerfahren = Text

Angabe des zu verwendenden Verschlüsselungsverfahrens

Klassenname des Verschlüsselungsverfahrens, mit dem der vom Datenverteiler gesendete Zufallstext über das Passwort chiffriert wird (optional, wenn keine Angabe erfolgt, wird das zur Zeit einzig unterstützte HMAC-MD5-Verfahren zur Verschlüsselung genutzt).

Default-Wert: de.bsvrz.dav.daf.communication.hmacmd5.AuthenticationHmacMD5

Optional: ja

Beispiel:

```
-authentifizierungsVerfahren=
de.bsvrz.dav.daf.communication.hmacmd5.AuthenticationHmacMD5
```

-konfigurationsBereich = PIDKonfiguration

Angaben zu dem zu verwendenden Konfigurationsverantwortlichen, sofern nicht die als Standard verwendete lokale Konfiguration verwendet werden soll.

Soll nicht die lokale Konfiguration verwendet werden, kann alternativ der zu startende Datenverteiler mit einer entfernten Konfiguration an einem gekoppelten Datenverteiler verbunden werden. *PIDKonfiguration*: PID der zu koppelnden Konfiguration am entfernten Datenverteiler (der Konfigurationsverantwortliche (AOE) für dieses gekoppelte System)

Default-Wert: ---

Optional: ja

Beispiel:

```
-konfigurationsBereich=kv.aoe.inovat.nerz
```

-inkarnationsname = Text

Inkarnationsname der Applikation

Dieser Wert wird standardmäßig von der SWE StartStopp (siehe [BetrInf_Sys-StartStopp]) vergeben (und sollte ansonsten nicht verwendet werden).

Default-Wert: „“

Optional: ja

Beispiel:

```
-inkarnationsname=InkarnationXyzApplikation
```

-zweiteVerbindung = [ja|nein]

Verwendung einer zweiten Datenverteilerverbindung für Konfigurationsanfragen

NERZ e.V.	BetrInf Datenverteiler-Applikationsfunktionen	Seite: 26 von 37 Version: 5.0 Stand: 19.04.2017
-----------	--	---

Mit diesem Aufrufparameter für Applikationen wird für die Durchführung von Konfigurationsanfragen eine zweite Datenverteilerverbindung aufgebaut. Hiermit kann in Hochlast-Situationen die Gefahr von Blockierungen vermieden werden.

Default-Wert: nein

Optional: ja

Beispiel:

-zweiteVerbindung=ja

-zweiteVerbindungPufferAnteil = Wert

Aufteilung der Sende- und Empfangspuffer der zweiten Verbindung

Über den Parameter lässt sich der Anteil an den Sende- und Empfangs- Puffern vorgeben, der der zweiten Verbindung zugewiesen wird. Der Wert muss zwischen 0,0 und 1,0 liegen und beträgt standardmäßig 0,01 (1%).

Default-Wert: 0,01

Optional: ja

Beispiel:

-zweiteVerbindungPufferAnteil=0,01

2.2.3.1.2.3 Aufrufparameter zur Kommunikationsüberwachung

-timeoutSendeKeepAlive = Zahl

Timeout Versand Keep-Alive-Telegramm

Zeit in Sekunden, nach der spätestens ein Keep-Alive-Telegramm versendet wird, wenn keine sonstigen Telegramme während dieser Zeit versendet wurden.

Der Timeout-Wert muss ≥ 1 Sekunde sein.

Default-Wert: 10 (s)

Optional: ja

Beispiel:

-timeoutSendeKeepAlive=5

-timeoutEmpfangeKeepAlive = Zahl

Timeout Empfang Keep-Alive-Telegramm

Timeoutzeit in Sekunden, in der spätestens ein Telegramm empfangen werden muss. Wird diese Zeit ohne Empfang von Telegrammen überschritten, wird die Verbindung zum Kommunikationspartner terminiert.

Der Timeout-Wert muss ≥ 1 Sekunde sein.

Default-Wert: 90 (s)

Optional: ja

Beispiel:

NERZ e.V.	BetrInf Datenverteiler-Applikationsfunktionen	Seite: 27 von 37 Version: 5.0 Stand: 19.04.2017
-----------	--	---

-timeoutEmpfangeKeepAlive=120

-durchsatzPruefung=PufferFüllgrad:PrüfIntervall:MindestDurchsatz

Einstellungen für die Durchsatzprüfung

Einstellungen für die Durchsatzprüfung, wobei drei durch Doppelpunkt getrennte Angaben gemacht werden müssen:

PufferFüllGrad: Gibt die Belegung des Sendepuffers in Prozent (Default 75 %) an, ab der zyklisch die Durchsatzprüfung erfolgen soll.

PrüfIntervall: Gibt das Messintervall in Sekunden (Default 60 Sekunden) an, nachdem der Durchsatz jeweils ermittelt werden soll.

MindestDurchsatz: Legt den Schwellwert in Bytes pro Sekunde (Default 3000 Bytes pro Sekunde) fest, mit dem der ermittelte Durchsatz verglichen wird. Ist der ermittelte Durchsatz kleiner als der MindestDurchsatz wird die Verbindung zum Datenverteiler terminiert.

Default-Wert: 75:60:3000

Optional: ja

Beispiel:

-durchsatzPruefung=75:60:3000

-puffer= SendepufferGröße:EmpfangspufferGröße:AuslieferungspufferGröße

Festlegung der Telegrammpuffer für Applikationen

Der Parameter enthält drei mit Doppelpunkt getrennte Zahlen, die die Sendepuffergröße, die Empfangspuffergröße und die Auslieferungspuffergröße für Telegramme in Bytes spezifizieren.

Die Puffergröße muss mindestens 100000 Byte betragen.

Bedeutung der einzelnen Werte:

SendepufferGröße: Die Größe des Sendepuffers in Bytes, der bei der Kommunikation mit dem Datenverteiler eingesetzt wird.

EmpfangspufferGröße: Die Größe des Empfangspuffers in Bytes, der bei der Kommunikation mit dem Datenverteiler eingesetzt wird.

AuslieferungspufferGröße: Die Größe des Auslieferungspuffers in Bytes, der zur Zwischenspeicherung von Datensätzen genutzt wird, die an einen Empfänger der Applikation versendet werden soll.

Default-Wert: 1000000:1000000:2500000

Optional: ja

Beispiel:

-puffer=50000000:50000000:2500000

2.2.3.1.2.4 Aufrufparameter zur Spezifikation der zu behandelnden Daten

-lokaleSpeicherungKonfiguration = Dateiordner[:Applikationsname]

Speicherungsart der lokalen Konfigurationsinformationen

Datei: BetrInf_DaV-DAF_FREI_V5.0_D2017-04-19.docx

NERZ e.V.	BetrInf Datenverteiler-Applikationsfunktionen	Seite: 28 von 37 Version: 5.0 Stand: 19.04.2017
-----------	--	---

Über diesen Parameter kann festgelegt werden, ob Konfigurationsinformationen persistent zwischengespeichert werden sollen. Wenn an dieser Stelle ein Verzeichnis angegeben wird, werden die Dateien mit den zwischengespeicherten Konfigurationsinformationen an dieser Stelle abgelegt.

Fehlt dieser Parameter erfolgt keine Zwischenspeicherung

Zusätzlich KANN, durch Doppelpunkt getrennt, der Name der Applikation, der im Namen der lokalen Cache-Datei für Konfigurationsdaten verwendet werden soll, angegeben werden. Dies dient lediglich zur ggf. notwendigen Unterscheidung der Cachedateien.

Dieser Aufrufparameter macht vor allem dann Sinn, wenn eine Applikation über eine langsame Verbindung mit dem Datenverteiler (und damit der Konfiguration) verbunden ist. So kann z. B. eine abgesetzte Bedienung mittels dieses Parameters erreichen, dass die Konfigurationsdaten nur einmal beim ersten Start auf den Bedienrechner übertragen werden und so bei weiteren Starts die Übertragung der Daten entfällt.

Bei Änderungen der Konfiguration werden die Daten beim nächsten Verbindungsaufbau wiederum einmal neu übertragen.

Default-Wert: *keine lokale Übertragung der Daten*

Optional: ja

Beispiel:

`-lokaleSpeicherungKonfiguration=C:\TmpKonfig\12`

-aspekt = ATG:ASP:NeuerASP

Aspektumleitung

Die Anmeldung der durch die Applikation benötigten und gelieferten Daten erfolgt über die Angabe der entsprechenden Attributgruppe mit dem dabei betrachteten Aspekt. Diese Angaben sind in den Applikationen als Default vorgegeben. Über diesen optionalen Aufrufparameter kann der Aspekt jeder zu verwendenden Attributgruppe geändert werden. Damit besteht die Möglichkeit, den Datenfluss beim Start einer Applikation zu modifizieren und damit beispielsweise einen weiteren Prozess in eine Bearbeitungskette einzufügen.

Beispiel

Eine Applikation erwartet als Default das Eingangsdatum die Attributgruppe "atg.verkehrswerte" mit dem Aspekt "asp.logischPLgeprueft". Diese Attributgruppe wird z.B. durch die Applikation PL-Prüfung FG 1 geschrieben. Über den Aufrufparameter

`-aspekt=atg.verkehrswerte:asp.logischPLgeprueft:asp.formalPLgeprueft`

wird eine Änderung des Datenflusses bewirkt. Die Applikation erhält nun als Eingangsdaten die Attributgruppe "atg.verkehrswerte" mit dem Aspekt "asp.formalPLgeprueft", die von einer anderen Applikation (PL-Prüfung formal) geschrieben wird.

Default-Wert: *keine Aspektumleitung*

Optional: ja

¹² Wenn im zweiten Zeichen ein Doppelpunkt steht, dann wird dieser als zum Pfad zugehörig betrachtet damit die Angabe mit absoluten Pfadnamen unter Windows funktioniert (C:\xyz...).

NERZ e.V.	BetrInf Datenverteiler-Applikationsfunktionen	Seite: 29 von 37 Version: 5.0 Stand: 19.04.2017
-----------	--	---

Beispiel:

`-aspekt=atg.verkehrswerte:asp.logischPLgeprueft:asp.formalPLgeprueft`

-simVariante = Zahl

Simulationsvariante vorgeben

Simulationsvariante für Telegramme, mit der die Applikation arbeitet

Wenn die Applikation im Normalbetrieb (keine Simulation) arbeiten soll, muss keine Angabe der Simulationsvariante erfolgen. Der Wert wird implizit auf 0 gesetzt. Durch Angabe eines Wertes im Bereich zwischen 1 und 999 kann die Applikation für eine bestimmte Simulationsvariante gestartet werden.

Default-Wert: 0

Optional: ja

Beispiel:

`-simVariante=27`

2.2.3.2 Starten der SWE

Entfällt, da die SWE eine reine Softwarebibliothek darstellt. Gestartet werden muss immer die Applikation (SWE), welche die Datenverteiler-Applikationsfunktionen verwendet (siehe dazu das entsprechende Kapitel in den Betriebsinformationen der jeweiligen SWE).

2.3 Überwachung des Betriebs

2.3.1 Überwachung des Prozesses

Entfällt, da die SWE eine reine Softwarebibliothek darstellt. Überwacht werden muss immer die Applikation (SWE), welche die Datenverteiler-Applikationsfunktionen verwendet (siehe dazu das entsprechende Kapitel in den Betriebsinformationen der jeweiligen SWE).

2.3.2 Prüfung Störungsfreiheit

2.3.2.1 Prüfung durch Log-Files

Meldungen der Datenverteiler-Applikationsfunktionen, die beim Ausführen der diese Bibliothek verwendenden SWE ausgegeben werden, lassen sich anhand des Logfiles überprüfen. Siehe dazu die Kapitel 4.2ff.

2.4 Unterbrechung oder Beendigung des Betriebs

2.4.1 Voraussetzungen

Da die SWE eine reine Softwarebibliothek darstellt, hängen die Voraussetzungen für die Unterbrechung oder Beendigung des Betriebs von der Applikation ab, welche die Bibliothek verwendet (siehe dazu das entsprechende Kapitel in den Betriebsinformationen der jeweiligen SWE).

2.4.2 Unterbrechung des Betriebs

Entfällt, da die SWE eine reine Softwarebibliothek darstellt. Unterbrochen werden muss immer die Applikation (SWE), welche die Datenverteiler-Applikationsfunktionen verwendet.

2.4.3 Beenden des Betriebs

Entfällt, da die SWE eine reine Softwarebibliothek darstellt. Beendet werden muss immer die Applikation (SWE), welche die Datenverteiler-Applikationsfunktionen verwendet.

3 Anwenderhandbuch

Die SWE ist eine intern genutzte Bibliothek. Es besteht keine direkte Verbindung zum Anwender.

Um Applikationen zu schreiben, die diese Bibliothek verwenden, muss das JAR-File `de.bsvrz.dav.daf-runtime.jar` im Classpath der Applikation aufgenommen werden.

Eine ausführliche Einführung in das ERZ-System mit Beispielen zur Erstellung von Applikationen mit den Datenverteiler-Applikationsfunktionen enthält das Tutorial zur Kernsoftware [Tutorial_KernSoftware].

4 Diagnosehandbuch

4.1 Benötigte Werkzeuge

In Tabelle 4-1 sind die benötigten Werkzeuge für die Diagnose aufgeführt.

Nr.	Werkzeug/Tool	Verwendung
1	Generischer Testmonitor [GTM]	Überprüfung der Parametrierung
2	Texteditor/Textviewer	Analyse der Log-Dateien
3	Prozess-Status-Monitor (Windows: z.B. „jps“; Linux: z.B. „ps“)	Überwachung des Betriebs der Applikation, die diese SWE verwendet

Tabelle 4-1: Analysewerkzeuge.

4.2 Diagnosemöglichkeiten

4.2.1 Betriebsmeldungen

Die Datenverteiler-Applikationsfunktionen versenden keine Betriebsmeldungen.

4.2.2 Allgemeine Meldungen (Startmeldungen)

Da die SWE eine reine Softwarebibliothek darstellt, hängen die beim Start der ausgegebenen Meldungen von der Applikation ab, welche die Bibliothek verwendet (siehe dazu das entsprechende Kapitel in den Betriebsinformationen der jeweiligen SWE).

4.2.3 Allgemeine Meldungen (Debug-Level WARNING)

Log-Einträge mit dem Level WARNING behindern zwar den Betrieb, verhindern ihn jedoch nicht. Es stehen jedoch nicht alle Funktionen der SWE zur Verfügung. Die Ursache einer Warnung sollte behoben werden, damit die SWE voll funktionstüchtig ist.

Angegeben sind die Warnungen, die bei Verwendung der Datenverteiler-Applikationsfunktionen zusätzlich zu den Warnungen der SWE ausgegeben werden, die diese Bibliothek verwendet.

Folgende Warnungen werden bei Bedarf generiert (*kursive* Einträge sind variabel):

Meldung	Beschreibung/Handlungsanweisung
Aktualisierung der MutableCollection <i>dynamischeMenge</i> kann nicht verarbeitet	Bei der Aktualisierung einer dynamischen Menge oder dynamischen Typs konnte der Typ der

Meldung	Beschreibung/Handlungsanweisung
werden, weil das Objekt einen unbekanntem Typ hat	Zusammenstellung nicht ermittelt werden.
Aktualisierung einer dynamischen Kollektion kann nicht weitergeleitet werden	Aktualisierung einer dynamischen Kollektion kann nicht weitergeleitet werden. Entwickler benachrichtigen
Anmeldung mit Historie gibt nur den Onlinestrom zurück: <i>exception</i>	Archivsystem nicht erreichbar
Antwort auf eine Konfigurationsanfrage konnte nicht interpretiert werden: <i>exception</i>	Problem bei der Kommunikation mit der Konfiguration (siehe <i>exception</i>)
Ausnahme bei der Verarbeitung eines empfangenen Telegramms <i>exception</i>	Bei der Verarbeitung eines Telegramms ist eine unerwartete Ausnahme (siehe <i>exception</i>) aufgetreten.
ClientDavConnection.removeConnectionListener: Zu entfernender Listener wurde nicht gefunden"	Es wurde versucht einen nicht angemeldeten ConnectionListener abzumelden
Der konfigurierende Datensatz für das Objekt <i>objekt</i> und der Attributgruppenverwendung <i>verwendung</i> konnte nicht deserialisiert werden: <i>exception</i>	Der konfigurierende Datensatz für ein Objekt und einer Attributgruppenverwendung konnte nicht deserialisiert werden: <i>exception</i>
Die Applikation: <i>applikationsname</i> will eine Archivfrage beim Archivsystem: <i>archivname</i> stellen, bekommt aber keine Rückmeldung vom Archivsystem. Die Archivfrage wird verworfen.	Archivsystem nicht erreichbar
Die Applikation: <i>applikationsname</i> will eine Archivfrage beim Archivsystem: <i>archivname</i> stellen, bekommt aber keine Rückmeldung vom Archivsystem. Die Archivfrage wird verworfen.	Archivsystem nicht erreichbar
Die Konfiguration ist nicht mehr für Anfragen bereit	Konfiguration hat sich beendet
Die Verbindung zum Archivsystem ist verloren gegangen: <i>archivname</i>	Archivsystem nicht mehr erreichbar
Die verwendete Konfiguration unterstützt nicht die Fertigmeldung für Start/Stop.	Älterer Stand des Datenmodells ohne Fertigmeldung für Start/Stop wird verwendet.
Eine optionale Referenz auf das Objekt mit der Pid <i>objectPid</i> konnte nicht aufgelöst werden"	Konfigurationsbereich des referenzierten Objekts fehlt oder Objekt ungültig
Element der Menge <i>name</i> mit ID <i>objektid</i> nicht gefunden (wird ignoriert)"	Nicht konsistente Konfiguration!?
Es gibt keine Verbindung zum Datenverteiler, die abgebaut werden könnte.	Beim Versuch eine nicht aufgebaute Datenverteilerverbindung zu schließen
Fehler bei der Archivfrage, es wird nur der Onlinestrom durchgereicht, da es aufgrund des Fehlers keine Archivdaten gibt: <i>exception</i>	Archivsystem nicht erreichbar
Fehler bei der asynchronen Benachrichtigung bzgl. Änderungen von dynamischen Mengen und dynamischen Typen: <i>exception</i>	Bei der Benachrichtigung der Applikation über Änderungen einer dynamischen Menge kam es zu einem Ausnahmezustand
Fehler beim Laden der lokalen Konfigurationsdatei aus dem Verzeichnis <i>dateipfad</i> , <i>exception</i>	Beim Laden der lokal zwischengespeicherten Konfigurationsdateien ist ein Fehler aufgetreten (siehe <i>exception</i>)
Fehler beim Lesen eines Zwischengespeicherten Datensatzes <i>exception</i>	Beim Lesen der lokal zwischengespeicherte Konfigurationsdatei konnte ein Datensatz nicht deserialisiert werden
Fehler beim Speichern der lokalen Konfigurationsdatei in das Verzeichnis <i>verzeichnis</i> , <i>exception</i>	Beim Speichern der lokal zwischengespeicherten Konfigurationsdateien ist ein Fehler aufgetreten (siehe <i>exception</i>)

Meldung	Beschreibung/Handlungsanweisung
Fehler beim Verarbeiten der connectionClosed-Meldung <i>exception</i>	Applikationsseitige Ausnahme bei der Verarbeitung der Verbindungsabbruch-Benachrichtigung an die Applikation
Fehler beim Verarbeiten der connectionClosed-Meldung: <i>exception</i>	Applikationsseitige Ausnahme bei der Verarbeitung der Verbindungsabbruch-Benachrichtigung an die Applikation
Initialer Datensatz bei Anmeldung als Quelle konnte nicht versandt werden und Anmeldung konnte nicht rückgängig gemacht werden: <i>exception</i>	Initialer Datensatz bei Anmeldung als Quelle konnte nicht versandt werden und Anmeldung konnte nicht rückgängig gemacht werden.
Initialisierung des RemoteRequestManager fehlgeschlagen: <i>exception</i>	Initialisierung des RemoteRequestManager fehlgeschlagen (siehe <i>exception</i>)
KommunikationszustandAktualisierung für systemObjekt kann nicht verarbeitet werden	Bei der Änderung des Kommunikationszustandes einer dynamischen Menge oder dynamischen Objekts konnte der Objekttyp nicht ermittelt werden.
Lesender Zugriff auf lokale Konfigurationsdatei nicht erlaubt <i>file</i>	Zugriffsrechte auf Konfigurationsdatei sollten überprüft werden.
Lokale Konfigurationsdatei kann nicht gespeichert werden, weil die dazu erforderlichen Konfigurationsbereichsinformationen nicht ermittelt werden konnten	Lokale Konfigurationsdatei kann nicht gespeichert werden, weil die dazu erforderlichen Konfigurations-bereichsinformationen nicht ermittelt werden konnten
Schreibender Zugriff auf vorhandene lokale Konfigurationsdatei nicht erlaubt <i>konfigurationsdatei</i>	Zugriffsrechte auf Konfigurationsdatei sollten überprüft werden.
Telegramm mit unbekanntem Typ <i>type</i> empfangen: <i>bytes</i>	Es wurde eine undefinierte Telegrammtypkennung (=1.Byte) eingelesen. Die Verbindung wird terminiert.
Thread wurde unterbrochen <i>exception</i>	Thread wurde unterbrochen (siehe <i>exception</i>)
Unbekannter IO-Fehler, es werden nur der Onlinestrom angezeigt: <i>exception</i>	Fehler bei der Kommunikation mit dem Archivsystem
Unerwarteter Fehler bei der Verarbeitung von asynchronen Konfigurationsantworten: <i>exception</i>	Unerwarteter Fehler bei der Verarbeitung von asynchronen Konfigurationsantworten (siehe <i>exception</i>)
Verbindung zum Datenverteiler sollte terminiert werden, obwohl sie noch nicht aufgebaut worden war	Beim Versuch eine nicht aufgebaute Datenverteilerverbindung zu schließen
Vorbereitung zum Laden der lokalen Konfigurationsdatei fehlgeschlagen <i>exception</i>	Beim Laden der lokal zwischengespeicherten Konfigurationsdateien ist ein Fehler aufgetreten (siehe <i>exception</i>)

Tabelle 4-2: Warnungen

4.2.4 Allgemeine Meldungen (Debug-Level ERROR)

Log-Einträge mit dem Level ERROR verhindern den korrekten Betrieb. Die Funktionen der SWE stehen nicht zur Verfügung. Die Ursache eines Fehlers muss umgehend behoben werden, damit die SWE funktionstüchtig ist.

Angegeben sind die Fehlermeldungen, die bei Verwendung der Datenverteiler-Applikationsfunktionen zusätzlich zu den Fehlermeldungen der SWE ausgegeben werden, die diese Bibliothek verwendet.

Folgende Fehlermeldungen werden bei Bedarf generiert (*kursive* Einträge sind variabel):

NERZ e.V.	BetrInf Datenverteiler-Applikationsfunktionen	Seite: 33 von 37 Version: 5.0 Stand: 19.04.2017
-----------	--	---

Meldung	Beschreibung/Handlungsanweisung
Alle Bereiche anfordern: <i>exception</i>	Problem bei der Kommunikation mit der Konfiguration (siehe <i>exception</i>)
Anfrage kann nicht serialisiert werden: <i>exception</i>	Beim Serialisieren eines zu speichernden konfigurierenden Datensatzes ist ein Fehler aufgetreten. Entwickler kontaktieren.
Angegebene Attributgruppe ist null, Aspekt: <i>aspekt</i> , Objekte: <i>objekte</i>	Es wurde beim Anmelden/Abmelden oder beim Lesen von Datensätzen eine ungültige Attributgruppe angegeben
Angegebener Aspekt ist null, Attributgruppe: <i>atg</i> , Objekte: <i>objekte</i>	Es wurde beim Anmelden/Abmelden oder beim Lesen von Datensätzen ein ungültiger Aspekt angegeben
Angegebenes Objekt-Array ist null, Attributgruppe: <i>atg</i> , Aspekt: <i>aspekt</i>	Es wurde beim Anmelden/Abmelden oder beim Lesen von Datensätzen ein ungültiges Objektarray angegeben
Anmeldung für Konfigurationsanfragen fehlgeschlagen: <i>exception</i>	Anmeldung für Konfigurationsanfragen fehlgeschlagen (siehe <i>exception</i>)
Aspekt <i>aspekt</i> darf nicht in Kombination mit Attributgruppe <i>atg</i> verwendet werden, Objekte: <i>objekte</i>	Es wurde beim Anmelden/Abmelden oder beim Lesen von Datensätzen eine ungültige Kombination von Attributgruppe und Aspekt angegeben
Attributgruppe <i>atg</i> darf nicht in Kombination mit Objekten vom Typ <i>typ</i> verwendet werden, Aspekt: <i>aspekt</i> , Objekt: <i>objekt</i>	Es wurde beim Anmelden/Abmelden oder beim Lesen von Datensätzen eine ungültige Kombination von Attributgruppe und Objekt angegeben
Betroffene Bereiche <i>configurationAreas</i> : <i>exception</i>	Problem bei der Kommunikation mit der Konfiguration (siehe <i>exception</i>)
Betroffene Bereiche <i>konfigurationsbereiche</i> : <i>exception</i>	Problem bei der Kommunikation mit der Konfiguration (siehe <i>exception</i>)
Die Verbindung wird terminiert, weil keine Telegramme mehr empfangen werden.	Die Verbindung wird terminiert, weil keine Telegramme mehr empfangen werden.
Die Verbindung wird terminiert: Sendedurchsatz war in den letzten t ms zu gering: durchsatz Byte/s	Die Verbindung wird terminiert: Sendedurchsatz war in den letzten t ms zu gering: durchsatz Byte/s
Die Verbindung zum Datenverteiler wird terminiert. Ursache: Es wurde x mal in Folge für t Sekunden keine KeepAlive- oder sonstige Telegramme empfangen	Die Verbindung zum Datenverteiler wird terminiert. Ursache: Es wurde x mal in Folge für t Sekunden keine KeepAlive- oder sonstige Telegramme empfangen
Die Verbindung zum Datenverteiler wird terminiert. Ursache: Kommunikationsfehler beim Lesen eines Telegramms: <i>exception</i>	Die Verbindung zum Datenverteiler wird terminiert. Ursache: Kommunikationsfehler beim Lesen eines Telegramms: <i>exception</i>
Die Verbindung zum Datenverteiler wird terminiert. Ursache: Sendedurchsatz war in den letzten t ms zu gering: <i>durchsatz</i> Byte/s	Die Verbindung zum Datenverteiler wird terminiert. Ursache: Sendedurchsatz war in den letzten t ms zu gering: <i>durchsatz</i> Byte/s
Die Verbindung zum Datenverteiler wird terminiert. Ursache: TCP-Verbindung wurde von der Gegenseite geschlossen: <i>exception</i>	Die Verbindung zum Datenverteiler wird terminiert. Ursache: TCP-Verbindung wurde von der Gegenseite geschlossen: <i>exception</i>
Die Verbindung zum Datenverteiler wird terminiert. Ursache: Verbindung soll erneut aufgebaut werden.	Die Verbindung zum Datenverteiler wird terminiert. Ursache: Verbindung soll erneut aufgebaut werden.
Die Verbindung zum Datenverteiler wird terminiert. Ursache: Verbindung wird applikationsseitig wegen eines Fehlers	Die Verbindung zum Datenverteiler wird terminiert. Ursache: Verbindung wird applikationsseitig wegen eines Fehlers

Meldung	Beschreibung/Handlungsanweisung
geschlossen.	geschlossen.
Die Verbindung zum Datenverteiler wird terminiert. Ursache: Verbindung wird auf Wunsch der Applikation geschlossen	Die Verbindung zum Datenverteiler wird terminiert. Ursache: Verbindung wird auf Wunsch der Applikation geschlossen
Die Verbindung zum Datenverteiler wird terminiert. Ursache: Verbindung wird wegen eines Kommunikationsfehlers beim Senden terminiert	Die Verbindung zum Datenverteiler wird terminiert. Ursache: Verbindung wird wegen eines Kommunikationsfehlers beim Senden terminiert
Die Verbindung zum Datenverteiler wird terminiert. Ursache: Verbindung wurde vom Datenverteiler geschlossen	Die Verbindung zum Datenverteiler wird terminiert. Ursache: Verbindung wurde vom Datenverteiler geschlossen
Die Verbindung zum Datenverteiler wird terminiert. Ursache: Verbindung wurde vom Datenverteiler terminiert. Ursache:	Die Verbindung zum Datenverteiler wird terminiert. Ursache: Verbindung wurde vom Datenverteiler terminiert. Ursache:
Ein Fehler beim serialisieren/deserialisieren: <i>exception</i>	Ein vom Archiv übertragener Datensatz konnte nicht deserialisiert werden (siehe <i>exception</i>)
Eine Quittung konnte nicht verschickt werden. Somit stoppt der Datenfluss vom StreamMultiplexer zum StreamDemultiplexer.	Archivsystem nicht mehr erreichbar
Empfänger Datensatz hat ungültigen Datensatzindex, Objekt: <i>name</i> , Attributgruppenverwendung: <i>verwendung</i> , letzter Index: <i>aindexT1#aindexT2#aindexT3</i> , aktueller Index: <i>indexT1#indexT2#indexT3</i>	Datensatzindex ist kleiner als der des vorhergehenden Datensatzes Entwickler kontaktieren.
Empfänger Datensatz konnte nicht verarbeitet werden, weil die Initialisierung noch nicht abgeschlossen wurde. Anmeldeinfo:objectID: <i>objektID</i> , attributeGroupCode: <i>attributeGroupCode</i> , aspectCode: <i>aspectCode</i> , simulationVariante: <i>simulationVariante</i> , atgu: <i>usageIdentification</i>	Empfänger Datensatz konnte nicht verarbeitet werden, weil die Initialisierung noch nicht abgeschlossen wurde (siehe Detailinfo).
Erzeugen eines neuen Konfigurationsberichts fehrgeschlagen, Name <i>name</i> Pid <i>pid</i> AuthorityPid <i>authorityPid</i> : <i>exception</i>	Problem bei der Kommunikation mit der Konfiguration (siehe <i>exception</i>)
Es wurde ein Systemobjekt auf ungültig gesetzt, zu dessen Id ein anderes Objekt im Cache gewesen ist	Es wurde ein Systemobjekt auf ungültig gesetzt, zu dessen Id ein anderes Objekt im Cache gewesen ist
Exportverzeichnis <i>verzeichnis</i> Betroffene Bereiche <i>konfigurationsbereiche</i> : <i>exception</i>	Problem bei der Kommunikation mit der Konfiguration (siehe <i>exception</i>)
Fehler bei der Verarbeitung der Sendesteuerung <i>exception</i>	Fehler bei der applikationsseitigen Verarbeitung der Sendesteuerung.
Fehler bei einer Konfigurationsanfrage nach der aktiven Version eines Konfigurationsbereichs, Bereich: <i>konfigurationsbereich</i> : <i>exception</i>	Problem bei der Kommunikation mit der Konfiguration (siehe <i>exception</i>)
Fehler bei einer Konfigurationsanfrage nach Objekten, die in bestimmten Bereichen gespeichert sind und bestimmte Typen besitzen, Bereiche <i>konfigurationsbereich</i> Typen: <i>systemobjekttyp</i> Zeitspezifikation <i>zeitspezifikation</i> : <i>exception</i>	Problem bei der Kommunikation mit der Konfiguration (siehe <i>exception</i>)
Fehler bei einer Konfigurationsanfrage nach Objekten, die in einem bestimmten Zeitbereich aktiv waren, angefragte Pid: <i>pid</i> : <i>exception</i>	Problem bei der Kommunikation mit der Konfiguration (siehe <i>exception</i>)

Meldung	Beschreibung/Handlungsanweisung
Fehler bei einer Konfigurationsanfrage, betroffenes Objekt und Menge: <i>pid menge: exception</i>	Problem bei der Kommunikation mit der Konfiguration (siehe <i>exception</i>)
Fehler beim Abfragen der Elemente einer Menge in einer bestimmten Version: <i>exception</i>	Problem bei der Kommunikation mit der Konfiguration (siehe <i>exception</i>)
Fehler beim Abfragen der Elemente einer Menge: <i>exception</i>	Problem bei der Kommunikation mit der Konfiguration (siehe <i>exception</i>)
Fehler beim aktiven Verbindungsaufbau zum Rechner <i>adresse</i> auf TCP-Port <i>portnummer</i> <i>exception</i>	TCP-Verbindung zum Datenverteiler kann nicht hergestellt werden.
Fehler beim Deserialisieren der Konfigurationsantwort beim Abfragen von noch nicht aktivierten neuen Objekten: <i>exception</i>	Problem bei der Kommunikation mit der Konfiguration (siehe <i>exception</i>)
Fehler beim Deserialisieren der Konfigurationsantwort beim Duplizieren eines Konfigurationsobjektes: <i>exception</i>	Problem bei der Kommunikation mit der Konfiguration (siehe <i>exception</i>)
Fehler beim Deserialisieren der Konfigurationsantwort beim Erzeugen eines neuen dynamischen Objektes: <i>exception</i>	Problem bei der Kommunikation mit der Konfiguration (siehe <i>exception</i>)
Fehler beim Deserialisieren der Konfigurationsantwort beim Erzeugen eines neuen Konfigurationsobjektes: <i>exception</i>	Problem bei der Kommunikation mit der Konfiguration (siehe <i>exception</i>)
Fehler beim Senden der Konfigurationsanfrage zum Abfragen von noch nicht aktivierten neuen Objekten: <i>exception</i>	Problem bei der Kommunikation mit der Konfiguration (siehe <i>exception</i>)
Fehler beim Senden der Konfigurationsanfrage zum Ändern von Einmalpasswörter: <i>exception</i>	Problem bei der Kommunikation mit der Konfiguration (siehe <i>exception</i>)
Fehler beim Senden der Konfigurationsanfrage zum Duplizieren eines Konfigurationsobjektes: <i>exception</i>	Problem bei der Kommunikation mit der Konfiguration (siehe <i>exception</i>)
Fehler beim Senden der Konfigurationsanfrage zum Erzeugen eines neuen dynamischen Objektes: <i>exception</i>	Problem bei der Kommunikation mit der Konfiguration (siehe <i>exception</i>)
Fehler beim Senden der Konfigurationsanfrage zum Erzeugen eines neuen Konfigurationsobjektes: <i>exception</i>	Problem bei der Kommunikation mit der Konfiguration (siehe <i>exception</i>)
Fehler beim Senden der Konfigurationsanfrage zum Ändern des Benutzerpasswortes: <i>exception</i>	Problem bei der Kommunikation mit der Konfiguration (siehe <i>exception</i>)
Fehler beim Senden der Konfigurationsanfrage zum Ändern von Benutzerrechten: <i>exception</i>	Problem bei der Kommunikation mit der Konfiguration (siehe <i>exception</i>)
Fehler beim Senden der Konfigurationsanfrage zum Anlegen eines neuen Benutzers: <i>exception</i>	Problem bei der Kommunikation mit der Konfiguration (siehe <i>exception</i>)
Fehler beim Serialisieren der Konfigurationsanfrage zum Ändern des Benutzerpasswortes: <i>exception</i>	Fehler beim Serialisieren der Konfigurationsanfrage zum Ändern des Benutzerpasswortes: (<i>siehe exception</i>)
Fehler beim Serialisieren der Konfigurationsanfrage zum Ändern von Benutzerrechten: <i>exception</i>	Fehler beim Serialisieren der Konfigurationsanfrage zum Ändern von Benutzerrechten: (<i>siehe exception</i>)
Fehler beim Serialisieren der Konfigurationsanfrage zum Ändern von Einmalpasswörter: <i>exception</i>	Fehler beim Serialisieren der Konfigurationsanfrage zum Ändern von Einmalpasswörter: (<i>siehe exception</i>)

Meldung	Beschreibung/Handlungsanweisung
Fehler beim Serialisieren der Konfigurationsanfrage zum Anlegen eines neuen Benutzers: <i>exception</i>	Fehler beim Serialisieren der Konfigurationsanfrage zum Anlegen eines neuen Benutzers: (<i>siehe exception</i>)
Fehler beim Verbindungsaufbau zum Rechner <i>name</i> auf TCP-Port <i>portnummer</i> : Ungültiges Argument	TCP-Verbindung zum Datenverteiler kann nicht hergestellt werden, weil die angegebene Portnummer ungültig ist
Fehler beim Verbindungsaufbau: Angegebener Rechner ist nicht erreichbar: <i>adresse</i>	TCP-Verbindung zum Datenverteiler kann nicht hergestellt werden, weil das angegebene Ziel nicht über IP erreicht werden kann.
Fehler beim Verbindungsaufbau: Unbekannter Rechnername: <i>name</i>	Beim Aufbau der TCP-Verbindung zum Datenverteiler wurde ein Rechnername mit unbekannter IP-Adresse angegeben.
Fehler beim Verbindungsaufbau: Verbindung zum Rechner <i>name</i> auf TCP-Port <i>portnummer</i> nicht möglich	TCP-Verbindung zum Datenverteiler kann nicht hergestellt werden, weil DaV/Rechner nicht läuft oder evtl. Kabel unterbrochen etc.
Fehler führt zum Beenden der Verbindung zum Datenverteiler: <i>exception</i>	Problem bei der Kommunikation mit der Konfiguration (<i>siehe exception</i>)
Hinzufügen von Objekten zu einer Menge fehlgeschlagen: <i>exception</i>	Problem bei der Kommunikation mit der Konfiguration (<i>siehe exception</i>)
Importverzeichnis <i>verzeichnis</i> Betroffene Bereiche <i>konfigurationsbereiche</i> : <i>exception</i>	Problem bei der Kommunikation mit der Konfiguration (<i>siehe exception</i>)
Kommunikationsproblem bei Abmeldung auf Änderungen des Kommunikationsstatus für <i>objekt</i> : <i>exception</i>	Kommunikationsproblem bei Abmeldung auf Änderungen des Kommunikationsstatus (<i>siehe exception</i>). Eventuell liegt Systemmodel nicht in richtiger Version vor
Kommunikationsproblem bei Anmeldung auf Änderungen der dynamischen Zusammenstellung <i>dynamischeMenge</i> : <i>exception</i>	Problem bei der Kommunikation mit der Konfiguration (<i>siehe exception</i>)
Kommunikationsproblem bei Anmeldung auf Änderungen des Kommunikationsstatus für <i>objekt</i> : <i>exception</i>	Kommunikationsproblem bei Anmeldung auf Änderungen des Kommunikationsstatus (<i>siehe exception</i>). Eventuell liegt Systemmodel nicht in richtiger Version vor
Kommunikationsproblem mit dem Datenverteiler, Attributgruppenverwendung: <i>verwendung</i> : <i>exception</i>	Problem bei der Kommunikation mit der Konfiguration (<i>siehe exception</i>)
Konfiguration hat das Erzeugen eines neuen dynamischen Objekts verweigert: <i>exception</i>	Konfiguration hat das Erzeugen eines neuen dynamischen Objekts verweigert: (<i>siehe exception</i>)
Konfiguration hat das Erzeugen eines neuen Objekts verweigert: <i>exception</i>	Konfiguration hat das Erzeugen eines neuen Objekts verweigert: (<i>siehe exception</i>)
Konfigurationsanfrage konnte nicht versendet werden: <i>exception</i>	Das Senden einer Konfigurationsanfrage ist fehlgeschlagen (<i>siehe exception</i>)
Löschen von Objekten aus einer Menge fehlgeschlagen: <i>exception</i>	Problem bei der Kommunikation mit der Konfiguration (<i>siehe exception</i>)
null-Objekt, Attributgruppe: <i>atg</i> , Aspekt: <i>aspekt</i>	Es wurde beim Anmelden/Abmelden oder beim Lesen von Datensätzen ein ungültiges Objekt angegeben
Telegramm zum Aktualisieren von Objekten konnten nicht verarbeitet werden	Bei der Verarbeitung eines Aktualisierungstelegramms von Objekten ist eine unerwartete Ausnahme (<i>siehe exception</i>) aufgetreten.
unerwartete Ausnahme im KeepAliveThread <i>exception</i>	unerwartete Ausnahme im KeepAliveThread <i>exception</i>

Meldung	Beschreibung/Handlungsanweisung
Unerwarteter Fehler beim Schreiben von konfigurierenden Datensätzen: <i>exception</i>	Unerwarteter Fehler beim Schreiben von konfigurierenden Datensätzen: (siehe <i>exception</i>)
Ungültige Antwort auf Anmeldung für Kommunikationsänderungen, communicationState <i>kommunikationsstatus</i>	Ungültige Antwort auf Anmeldung für Kommunikationsänderungen, communicationState (siehe <i>kommunikationsstatus</i>)
UpdateThread im RemoteRequestManager wurde unterbrochen <i>responseAspect responseAtg</i> Nachrichtentyp: <i>dynamischeMengeAktualisierung: exception</i>	Thread zur Benachrichtigung von Konfigurationsänderungen wurde unterbrochen (siehe <i>exception</i>)
Von einem Objekt konnte kein Duplikat erstellt werden, da das nötige Telegramm nicht an den Datenverteiler verschickt werden konnte oder die Rückantwort konnte nicht entschlüsselt werden. Betroffenes Objekt: <i>objektPid: exception</i>	Kommunikationsproblem beim Duplizieren eines Objekts (siehe <i>exception</i>). Eventuell liegt Systemmodel nicht in richtiger Version vor

Tabelle 4-3: Fehlermeldungen

4.2.5 Überprüfung der Parametrierung

Entfällt, da SWE keine Parametrierung benötigt.

4.2.6 Überprüfung des Prozesszustandes

Siehe hierzu Kapitel 2.3 „Überwachung des Betriebs“

5 Anhang

5.1 Verzeichnisstruktur

```

de.bsvrz.dav.daf
├── build.xml
├── de.bsvrz.dav.daf-doc-api.zip
├── de.bsvrz.dav.daf-doc-design.zip
├── de.bsvrz.dav.daf-info.xml
├── de.bsvrz.dav.daf-lizenz.txt
├── de.bsvrz.dav.daf-runtime.jar
├── de.bsvrz.dav.daf-src.zip
├── de.bsvrz.dav.daf.jar
└── lib

```