

**Schnittstelle für die serielle Meßdatenübermittlung
Anwendungsdienste, Telegramme und Protokolle**
DIN
66348-3

ICS 35.200

Deskriptoren: OSI, Anwendungsschicht, Meßdatenübermittlung,
Schnittstelle, Steuerungsverfahren

Interface for serial measurement data communication –
Part 3: Application services, messages (protocol data units) and protocols

Interfaces et modes de commande pour le transfert de données de
mesure en série – Partie 3: Services d'application, spécification de
messagerie et de protocole

Vorwort

Diese Norm wurde vom Unterausschuß NATG-C.2.9 "Schnittstellen" (früher NLG 2.9) erarbeitet. Es war beabsichtigt, sie zusammen mit DIN 66348-2 in eine Europäische Norm über Bussysteme für meßtechnische Anwendungen beim CENELEC/TC 65 CX einfließen zu lassen, jedoch ist dort ein entsprechendes "Work item" bisher nicht angelaufen, so daß es vorerst bei einer nationalen Norm bleibt.

DIN 66348 "Schnittstellen und Steuerungsverfahren für die serielle Meßdatenübermittlung" besteht aus:

Teil 1: Start-Stop-Übertragung, Punkt-zu-Punkt-Verbindung

Teil 2: Start-Stop-Übertragung, 4-Draht-Meßbus

Teil 3: Anwendungsdienste, Telegramme und Protokolle

Ein auf den Teilen 2 und 3 basierendes Geräteprofil steht als "European Petrol Station Interface (EPSI)" auf dem Arbeitsprogramm des NATank.

Fortsetzung Seite 2 bis 160

Normenausschuß Technische Grundlagen (NATG) – Geometrische Produktspezifikation und -prüfung –
im DIN Deutsches Institut für Normung e.V.

Inhalt

1	Anwendungsbereich	3
2	Normative Verweise	4
3	Begriffe und Abkürzungen	5
3.1	Begriffe	5
3.2	Abkürzungen	8
4	Kommunikationsmodell	8
4.1	Allgemeines	8
4.2	Interne Struktur eines Kommunikationsteilnehmers	8
4.3	Kommunikationsbeziehung	9
4.4	Auftraggeber und Auftragnehmer	10
4.5	Anwendungsdienste	10
4.6	Anwendungsdienstprimitive	11
4.7	Kommunikationsablauf bei der Anwendungsdienstausführung	13
4.8	Struktur der Anwendungsdienst-Tabellen	16
5	Codierregeln	18
5.1	Trennung von Telegrammteilen	18
5.2	Anwendungstelegrammsyntax	20
5.3	Datentypen	23
6	Anwendungsdienste zur Verbindungsverwaltung	31
6.1	Anwendungsdienst Verbindungsaufbau	31
6.2	Anwendungsdienst Verbindungsabbau	35
6.3	Anwendungsdienst Verbindungsabbruch	37
6.4	Anwendungskontext	38
7	Speicherbereich-bezogene Anwendungsdienste	39
7.1	Speicherorganisationsmodell	39
7.2	Ladedienste	44
7.3	Rückladedienste	50
7.4	Anwendungsdienst LöscheSpeicherbereich	57
8	Variable-bezogene Anwendungsdienste	59
8.1	Anwendungsdienst DefiniereVariablenliste	59
8.2	Anwendungsdienst LöscheVariablenliste	62
8.3	Anwendungsdienst VariableLesen	65
8.4	Anwendungsdienst VariableSchreiben	68
8.5	Anwendungsdienst UnangeforderterBericht	71
9	Anwendungsdienste zur Programmablaufsteuerung	75
9.1	Programmausführungsmodell	75
9.2	Anwendungsdienst ProgrammAktivieren	76
9.3	Anwendungsdienst ProgrammDeaktivieren	78
9.4	Anwendungsdienst ProgrammStarten	80
9.5	Anwendungsdienst ProgrammStoppen	82
9.6	Anwendungsdienst ProgrammFortsetzen	83
9.7	Anwendungsdienst ProgrammRücksetzen	85
9.8	Anwendungsdienst ProgrammAbbrechen	86
9.9	Anwendungskontext	87
10	Allgemeine Anwendungsdienste	90
10.1	Anwendungsdienst Ablauffehler	90
10.2	Anwendungsdienst Dienstausführungsabbruch	94
10.3	Anwendungsdienst Identifikation	96
10.4	Anwendungsdienst AktionAuslösen	97
10.5	Anwendungsdienst ÜbermittleNamenliste	99
10.6	Zustandsdienste	103
11	Fehlerbehandlung	108
11.1	Dienstprimitiv-Parameter	108
11.2	Fehlertelegramme	111
12	Schnittstelle zu den transportorientierten Schichten [STS]	113

12.1	Allgemeines	113
12.2	Kommunikationssystem-Modell.....	113
12.3	Verbindungsnummer	114
12.4	Dienst STS-Verbindungsnummer.....	115
12.5	Dienst STS-Anwendungsdaten	118
13	Prüfdienste	120
13.1	Allgemeines	120
13.2	Prüfdienst STS-PrüfparameterSetzen	120
13.3	Prüfdienst STS-PrüfparameterMelden	126
Anhang A (normativ)	Nachladebetrieb.....	127
Anhang B (normativ)	Notation zur Festschreibung von Variablen und Datentypen	129
Anhang C (informativ)	Beispiele zur Anwendungstelegramm-Codierung	135
Anhang D (informativ)	Datenblatt	154

1 Anwendungsbereich

Aufbauend auf das Basisreferenzmodell¹⁾ für die Kommunikation offener Systeme [OSI], beschreiben die Normen der Reihe DIN 66348 eine herstellernerneutrale Kommunikationsschnittstelle für die serielle Informationsübertragung in verteilten Meß-, Prüf-, Zähl- und Anzeigesystemen.

Die wahlweise anzuwendenden Teile 1 und 2 der DIN 66348 legen die transportorientierten Funktionen der Kommunikationsschnittstelle fest. Das heißt, sie definieren die elektrischen Eigenschaften der Schnittstelle und das Steuerungsverfahren für den gesicherten Informationsaustausch. Gegenstand von DIN 66348-1 ist die Beschreibung des Informationsaustauschs auf Punkt-zu-Punkt-Verbindungen. Teil 2 beschreibt den Informationsaustausch auf Mehrpunktverbindungen über einen Bus.

Nach dem OSI-Konzept stellt diese Norm ein sogenanntes Anwendungsdienstelement dar, in dem in Ergänzung zu DIN 66348-1 und DIN 66348-2 die anwendungsorientierten Funktionen der Kommunikationsschnittstelle festgelegt werden. Beschrieben werden in dieser Norm

- 29 Anwendungsdienste;
- der Inhalt, der Aufbau und die Aufgabe der Anwendungsdienstprimitive und Anwendungstelegramme, die den Anwendungsdiensten zugeordnet sind;
- das Protokoll, nach dem der Anwendungstelegrammaustausch erfolgt;
- als sogenannte "Schnittstelle zu den transport-orientierten Schichten" die Schnittstelle zwischen Anwendungsdienstelement und den transportorientierten Funktionen der Kommunikationsschnittstelle nach DIN 66348-1 und 66348-2;
- zwei optionale Dienste zur Vereinfachung der Implementierungsprüfung;
- als Anhänge der sogenannte Nachladebetrieb, eine Notation zur Festschreibung von Variablen und Datentypen, Beispiele zur Telegrammcodierung und die Beschreibung des Datenblatts für Kommunikationsteilnehmer mit Schnittstellen nach dieser Norm.

Die in dieser Norm definierten Anwendungsdienste sind zu ihrer Beschreibung in den fünf Dienstgruppen

- Anwendungsdienste zur Verbindungsverwaltung (siehe Abschnitt 6),
- Speicherbereich-bezogene Anwendungsdienste (siehe Abschnitt 7),
- Variable-bezogene Anwendungsdienste (siehe Abschnitt 8),
- Anwendungsdienste zur Steuerung des Programmablaufs (siehe Abschnitt 9) und
- allgemeine Anwendungsdienste (siehe Abschnitt 10)

zusammengefaßt.

Von den in dieser Norm definierten Anwendungsdiensten müssen als Pflichtdienste zwingend von jedem Kommunikationsteilnehmer in der Rolle als Auftragnehmer die Anwendungsdienste "Identifikation" und "Verbindungsaufbau" und darüber hinaus die Anwendungsdienste "Ablauffehler" und "Verbindungsabbruch" sowohl in der Rolle als Sender wie auch als Empfänger der zugeordneten Meldungstelegramme unterstützt werden. Die übrigen Anwendungsdienste sind optional; ihre Verwendung ergibt sich aus den jeweils über die Kommunikationsschnittstelle zu unterstützenden Geräte- bzw. Systemfunktionen des Kommunikationsteilnehmers. Es obliegt den Herstellern während der Projektierung festzulegen, welche der optionalen Anwendungsdienste für den Zugriff auf die Geräte- bzw. Systemfunktionen benötigt werden.

Es kann je nach Gerätefunktionalität ausreichen, wenn beispielsweise ein Längenmeßgerät von den in

1) ITU-T X.200 | ISO/IEC 7498-1

dieser Norm definierten Anwendungsdiensten nur die vier Pflichtdienste sowie den optionalen Variable-bezogenen Anwendungsdienst "VariableLesen" für den Zugriff auf die Meßdaten unterstützt.

Sofern es in Einzelfällen nicht ausdrücklich angegeben wird, beschreibt diese Norm grundsätzlich die Auftragnehmerrolle bei der Anwendungsdienstausführung. Wegen der umkehrbaren Eindeutigkeit ist damit zwingend auch die Auftraggeberrolle beschrieben.

Das dieser Norm zugrunde liegende Kommunikationsmodell (Abschnitt 4) ist aus dem in ISO/IEC 9506-1 und ISO/IEC 9506-2 beschriebenen Konzept der "Manufacturing Message Specification [MMS]" abgeleitet worden²⁾. In DIN 66348-3 werden die Beziehungen zu den entsprechenden Funktionen und Elementen von MMS angegeben. Das Verstehen von DIN 66348-3 setzt jedoch in keiner Weise die Kenntnis der ISO/IEC 9506 voraus.

Die in dieser Norm festgelegten Anwendungsdienste sind dienstkompatibel mit den jeweils angegebenen MMS-Anwendungsdiensten. Um diese Kompatibilität auch in zukünftigen Ausgaben dieser Norm sicherstellen zu können, ist an verschiedenen Stellen durch den Hinweis "gesperrt für zukünftige Erweiterungen" ein Platzhalter eingefügt worden, ohne daß zum gegenwärtigen Zeitpunkt diese Erweiterung tatsächlich geplant ist.

Die in ISO/IEC 9506 vorgesehenen "Service Modifier" werden in dieser Norm nicht verwendet, ebenso wird ein Variablenzugriff über Adressen nicht unterstützt.

Anstelle der in ISO/IEC 9506 verwendeten Telegrammcodierung nach den Basiscodierregeln zu ASN.1 [siehe ISO 8825] wird in dieser Norm für die Anwendungstelegramme eine 7-Bit-Codierung nach DIN 66 003 verwendet (siehe Abschnitt 5).

Diese Norm ist für die Kommunikation mit Meß-, Prüf-, Zähl- und Anzeigeeinrichtungen mit eingeschränkter Rechnerleistung vorgesehen. Für Einrichtungen, die über eine große lokale Rechnerleistung verfügen, wie beispielsweise für Mehrkoordinatenmeßmaschinen, ist die Anwendung von ISO/IEC 9506-1 und ISO/IEC 9506-2 in Verbindung mit den zugehörigen Companion Standards vorzusehen.

Die Anwendung dieser Norm ist nicht an die gleichzeitige Anwendung von DIN 66348-1 oder DIN 66348-2 gebunden. DIN 66348-3 darf daher grundsätzlich auch in Verbindung mit anderen transportorientierten Kommunikationsprotokollen für codegebundene oder codeunabhängige Datenübertragung, wie beispielsweise mit den TCP/IP-Protokollen oder entsprechenden OSI-Protokollen, angewendet werden.

Die Art und Weise der Nutzung der in dieser Norm beschriebenen Anwendungsdienste für Aufgaben der Qualitätssicherung, der Prozeßüberwachung oder der Betriebsdatenerfassung in der Fertigungstechnik wird in einer in Vorbereitung befindlichen Norm für "meßtechnische Anwendungen und Funktionen" festgelegt.

Neben dem im Rahmen der Qualitätssicherung erfolgenden Einsatz in der Fertigungstechnik darf diese Norm aufgrund ihrer modularen Konzeption auch für die Lösung von Kommunikationsaufgaben in anderen Anwendungsgebieten (z.B. Tankstellen- und Textiltechnik) genutzt werden. Den Geräteherstellern wird daher empfohlen, vor dem Hintergrund der Harmonisierung der Anwendungsdienstelemente in verschiedenen Einsatzgebieten und zur Reduzierung der Schnittstellenvielfalt, diese Norm in allen Bereichen der Meß-, Prüf-, Zähl- und Anzeigetechnik anzuwenden.

2 Normative Verweise

Diese Norm enthält durch datierte oder undatierte Verweisungen Festlegungen aus anderen Publikationen. Diese normativen Verweisungen sind an den jeweiligen Stellen im Text zitiert, und die Publikationen sind nachstehend aufgeführt. Bei datierten Verweisungen gehören spätere Änderungen oder Überarbeitungen dieser Publikation nur zu dieser Norm, falls sie durch Änderung oder Überarbeitung eingearbeitet sind. Bei undatierten Verweisungen gilt die letzte Ausgabe der in Bezug genommenen Publikation.

DIN 44 300 Teil 1-9: 1988	Informationsverarbeitung; Begriffe
DIN 44 302: 1987	Informationsverarbeitung; Datenübertragung, Datenübermittlung, Begriffe
DIN 66003: 1974	Informationsverarbeitung; 7-Bit-Code
DIN 66250: 1987	Informationsverarbeitung; Zahlendarstellung für den Datenaustausch
DIN 66306-1: 1992	Industrielle Automation und Integration - Festlegung der Nachrichtenformate für Fertigungszwecke (siehe ISO 9506-1/EN 29 506-1)
DIN 66306-2: 1992	Industrielle Automation und Integration - Festlegung der Nachrichtenformate für Fertigungszwecke (siehe ISO 9506-2/ EN 29 506-2)

2) In der Bundesrepublik Deutschland genormt als DIN 66306

DIN 66348-1: 1986	Schnittstellen und Steuerungsverfahren für die serielle Meßdatenübermittlung; Start-Stop-Übertragung, Punkt-zu-Punkt-Verbindung
DIN 66348-2: 1989	Schnittstellen und Steuerungsverfahren für die serielle Meßdatenübermittlung; Start-Stop-Übertragung, 4-Draht-Meßbus
ITU-T X.200 ISO/IEC 7498-1:1994 (E)	Information Technology - Open systems interconnection - Basic reference model
EN 29 506-1:	siehe ISO 9506-1
EN 29 506-2:	siehe ISO 9506-2
ISO/TR 8509: 1987 (E)	Information processing systems - Open Systems Interconnection - Service conventions
ISO 6093: 1985 (E)	Information processing - Representation of numerical values in character strings for information interchange
ISO 8649:1988 (E)	Information processing systems - Open Systems Interconnection - Service definition for the Association Control Service Element
ISO 8650:1988 (E)	Information processing systems - Open Systems Interconnection - Protocol specification for the Association Control Service Element
ISO/IEC 8825: 1990 (E)	Information technology - Open Systems Interconnection - Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1)
ISO 9506-1: 1990 (E)	Industrial automation systems - Manufacturing Message Specification - Service definition
ISO 9506-2: 1990 (E)	Industrial automation systems - Manufacturing Message Specification - Protocol specification

3 Begriffe und Abkürzungen

3.1 Begriffe

Für die Anwendung dieser Norm gelten die Begriffe nach DIN 44300, DIN 44302 und die folgenden:

3.1.1 Kommunikationsteilnehmer

Eine Funktionseinheit, das heißt ein materiell abgrenzbares Gebilde wie beispielsweise ein Personal Computer, ein Längenmeßgerät oder die Zapfsäule in einer Tankstelle, die zur Erfüllung einer gemeinsamen Informationsverarbeitungsaufgabe Nachrichten und Daten mit einem oder mit mehreren anderen Kommunikationsteilnehmern austauscht. Ein Kommunikationsteilnehmer gliedert sich in einen oder in mehrere Anwendungsprozesse (siehe 3.1.2) und in das Kommunikationssystem (siehe 3.1.3).

3.1.2 Anwendungsprozeß

Der Teil eines Kommunikationsteilnehmers, der eine abgegrenzte Informationsverarbeitungsaufgabe ausführt und dabei mit Anwendungsprozessen in anderen Kommunikationsteilnehmern zusammenarbeitet.

3.1.3 Kommunikationssystem

Der Teil eines Kommunikationsteilnehmers, der durch Telegrammübertragung den Informationsaustausch mit anderen Kommunikationsteilnehmern ermöglicht.

3.1.4 Anwendungsprozeßschnittstelle [API]

Interne Schnittstelle in einem Kommunikationsteilnehmer, über die der lokale Informationsfluß zwischen Anwendungsprozeß und Kommunikationssystem erfolgt.

3.1.5 Anwendungsdienst

Eine spezifische Informationsverarbeitungsleistung, die ein Anwendungsprozeß nach einer Aufforderung (= bestätigter Anwendungsdienst) oder auf Eigeninitiative (= unbestätigter Anwendungsdienst) für einen anderen Anwendungsprozeß erbringt.

3.1.6 Anwendungsdienstprimitiv (Dienstprimitiv)

Die durch eine implementierungsneutrale Datenstruktur beschriebene Interaktion zwischen Anwendungsprozeß und Kommunikationssystem, die zum internen Informationsaustausch innerhalb eines Kommunikationsteilnehmers führt.

3.1.7 Instanz

Ein nach Aufgabe oder Wirkung abgrenzbares Gebilde in einem Kommunikationsteilnehmer/Anwendungsprozeß, das

- Aufträge erteilt oder erhält,
- erhaltene Aufträge ablehnt oder annimmt und
- angenommene Aufträge ganz oder teilweise selbst ausführt, weitergibt oder bei Unausführbarkeit zurückgibt.

Unter einem Auftrag wird im Rahmen dieser Norm die Ausführung von Anwendungsdiensten verstanden.

3.1.8 Server

Ein Anwendungsprozeß, der dazu ausgelegt ist, für einen oder für mehrere Kommunikationspartner (Client, siehe 3.1.9), neben möglichen anderen, vor allem meß-, prüf-, zähl- oder anzeigetechnische Dienstleistungen zu erbringen. Aufgrund seiner Aufgabe tritt ein Server in der Regel als Auftragnehmer auf. Er kann für bestimmte Anwendungsdienste aber auch als Auftraggeber auftreten.

3.1.9 Client

Ein Anwendungsprozeß, der dazu ausgelegt ist, neben möglichen anderen vor allem die meß-, prüf-, zähl- oder anzeigetechnische Dienstleistungen eines oder mehrerer Server (siehe 3.1.8) zu nutzen. Aufgrund seiner Aufgabe tritt ein Client in der Regel als Auftraggeber auf. Er kann für bestimmte Anwendungsdienste aber auch als Auftragnehmer agieren.

3.1.10 Auftraggeber

Der Anwendungsprozeß, der durch Telegrammübertragung bei einem anderen Anwendungsprozeß die Ausführung eines Anwendungsdienstes anfordert.

3.1.11 Auftragnehmer

Der Anwendungsprozeß, der durch Telegrammübertragung zur Ausführung eines Anwendungsdienstes von einem anderen Anwendungsprozeß aufgefordert wird.

ANMERKUNG: Die Begriffe Client und Server legen die "anwendungstechnische" Rolle von Anwendungsprozessen fest, während die Begriffe Auftragnehmer und Auftraggeber die "kommunikationstechnische" Rolle beschreiben. Ein Anwendungsprozeß in einem Kommunikationsteilnehmer kann entweder als Server oder als Client oder in einer Client/Server-Doppelrolle ausgelegt sein. Die Teilmenge der unterstützten Anwendungsdienste bzw. die kommunikationstechnische Rolle bei der Anwendungsdienstausführung unterscheidet sich im allgemeinen zwischen Server- und Client-Anwendungen und ist daher im Datenblatt eines Kommunikationsteilnehmers (siehe Anhang D) getrennt anzugeben.

3.1.12 Speicher

Eine Funktionseinheit in einem Kommunikationsteilnehmer, die Daten aufnimmt, aufbewahrt und abgibt.

3.1.13 Speicherbereich

Ein Teil eines Speichers, der mit einem Bezeichner (Speicherbereichname) versehen ist.

3.1.14 Speicherzelle

Ein Teil eines Speichers bzw. eines Speicherbereichs, der zur Aufnahme eines Datenobjektes dient.

3.1.15 Lesen

Daten einem Speicher, einem Speicherbereich oder einer Speicherzelle entnehmen.

3.1.16 Schreiben

Daten einem Speicher, einem Speicherbereich oder einer Speicherzelle übergeben.

3.1.17 Variable

Eine mit einem Bezeichner (Variablenname) versehene Speicherzelle, die ein Datenobjekt aufbewahrt. Datenobjekte sind beispielsweise ein Meßwert oder ein Geräteparameter. Auf eine Variable in einem Kommunikationsteilnehmer kann je nach Anwendungsfall unmittelbar von einem Kommunikationspartner lesend und/oder schreibend zugegriffen werden.

3.1.18 Variablenliste

Ein über einen Bezeichner (Variablenlistenname) zugängliches Verzeichnis mit Variablennamen. Der Variablenlistenname wird dazu verwendet, um gleichzeitig lesend oder schreibend auf alle im Verzeichnis namentlich angegebenen Variablen zuzugreifen, ohne in der Lese- bzw. Schreib Anforderung alle Variablennamen einzeln angeben zu müssen.

3.1.19 Datenquelle

Ein Kommunikationsteilnehmer/Anwendungsprozeß, der Daten, wie beispielsweise den Inhalt eines Speicherbereichs oder den Wert einer Variablen, zur Übertragung an einen anderen Kommunikationsteilnehmer bereitstellt.

3.1.20 Datensenke

Ein Kommunikationsteilnehmer/Anwendungsprozeß, an den Daten von einem anderen Kommunikationsteilnehmer (Datenquelle) übertragen werden.

3.1.21 Lademodul

Eine Instanz in einem als Datensenke auftretenden Anwendungsprozeß, die vom Kommunikationspartner übertragene Daten empfängt und diese Daten in einen "lokalen" Speicherbereich schreibt.

3.1.22 Rücklademodul

Eine Instanz in einem als Datenquelle auftretenden Anwendungsprozeß, die aus einem "lokalen" Speicherbereich Daten ausliest und diese Daten an einen Kommunikationspartner überträgt.

3.1.23 Programmausführer

Eine Instanz in einem Anwendungsprozeß, die es einem Kommunikationspartner gestattet, einen "lokalen" Programmablauf im Anwendungsprozeß zu steuern.

3.1.24 Telegramm

Eine nach bestimmten Codierregeln gebildete Zeichenfolge, die zum Informationsaustausch zwischen verschiedenen Kommunikationsteilnehmern über ein Übertragungsmedium gesendet wird.

3.1.25 Anwendungstelegramm

Der Teil eines Telegramms, der sich unmittelbar auf die Ausführung eines Anwendungsdienstes bezieht.

3.1.26 Datenfeld

Der kleinste zusammenhängende Teil eines Telegramms. Beispiele für Datenfelder sind Namen oder Zahlen, die jeweils aus mindestens einem Zeichen gebildet werden.

3.1.27 Informationselement

Eine spezifische Gruppe (ein spezifischer Zusammenschluß) von Parametern.

3.1.28 Anwendungsprozeßname

Lokale Anwendungsprozeßnamen sind vom Hersteller vergebene Namen, die zur Unterscheidung mehrerer Anwendungsprozesse in einem Kommunikationsteilnehmer dienen. Lokale Anwendungsprozeßnamen werden beim Einsatz des Kommunikationsteilnehmers in einem geschlossenen Informationsverbundsystem auf systemweite Anwendungsprozeßnamen abgebildet.

Jeder systemweite Anwendungsprozeßname darf innerhalb eines geschlossenen Informationsverbundsystems nur einmal vorkommen (während lokale Anwendungsprozeßnamen mehrfach, z.B. beim Einsatz mehrerer identischer Geräte in einem System auftreten können).

3.2 Abkürzungen

In dieser Norm werden folgende allgemeine Abkürzungen verwendet:

Anf	Dienstprimitiv vom Typ "Anforderung";
Ant	Dienstprimitiv vom Typ "Antwort";
Anz	Dienstprimitiv vom Typ "Anzeige";
API	Anwendungsprozeßschnittstelle (auch als Anwendungsprogrammschnittstelle bezeichnet);
APM	Anwendungsprotokollmaschine;
Bes	Dienstprimitiv vom Typ "Bestätigung";
MMS	Manufacturing Message Specification (siehe ISO/IEC 9506-1 und ISO/IEC 9506-2);
OSI	Open Systems Interconnection (Kommunikation offener Systeme);
STS	Schnittstelle zu den transportorientierten Schichten;
TPM	Transportprotokollmaschine.

ANMERKUNG: In dieser Norm werden zahlreiche Abkürzungen für Dienstprimitivparameter bzw. für Telegrammteile verwendet. Da diese Abkürzungen nur lokale Bedeutung in einem Abschnitt haben und dort beschrieben sind, werden sie an dieser Stelle nicht angegeben.

4 Kommunikationsmodell

4.1 Allgemeines

Diese Norm geht davon aus, daß mindestens zwei Kommunikationsteilnehmer in einem verteilten System (Informationsverbundsystem) gemeinsam die Informationsverarbeitung für eine bestimmte Anwendungsaufgabe ausführen.

Ziel dieser Norm ist es, die Anwendungstelegamme zu definieren, die zwischen den Kommunikationsteilnehmern eines Informationsverbundsystems ausgetauscht werden müssen, und die Protokolle festzulegen, nach denen der Telegrammaustausch erfolgt.

Zur Beschreibung der in dieser Norm definierten Anwendungstelegamme und der Protokolle definiert diese Norm das implementierungsneutrale Kommunikationsmodell, das in diesem Abschnitt beschrieben ist.

4.2 Interne Struktur eines Kommunikationsteilnehmers

Ein Kommunikationsteilnehmer gliedert sich, wie in Bild 1 am Modell dargestellt, in mindestens einen Anwendungsprozeß und in das Kommunikationssystem.

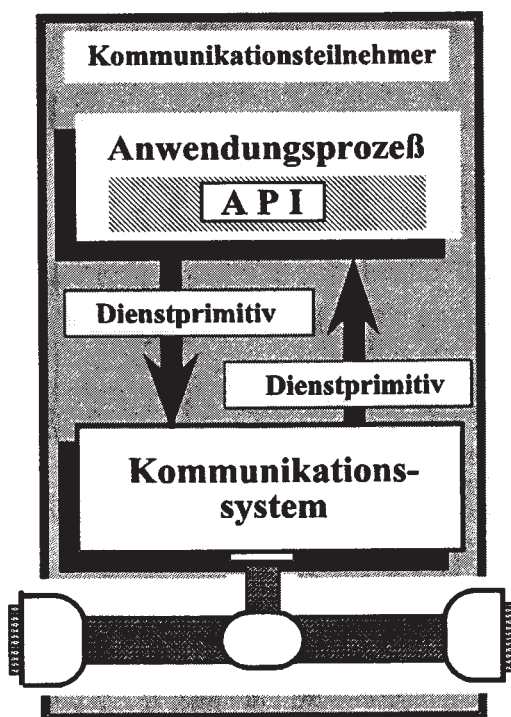


Bild 1: Modell der internen Struktur eines Kommunikationsteilnehmers

Der Anwendungsprozeß ist der Teil des Kommunikationsteilnehmers, der für eine bestimmte Anwen-

dungsaufgabe die Informationsverarbeitung ausführt. Strikt von der Informationsverarbeitung getrennt ist die Informationsübertragung, die Aufgabe des Kommunikationssystems ist.

Zum Informationsaustausch werden über ein physikalisches Übertragungsmedium Telegramme zwischen den Kommunikationssystemen der Kommunikationsteilnehmer gesendet. Die Regeln, nach denen dieser Telegrammaustausch zwischen den Kommunikationspartnern erfolgt, heißen Protokoll.

Da ein Kommunikationsteilnehmer mehrere voneinander unabhängige Anwendungsprozesse besitzen darf, werden die Anwendungsprozesse eines Kommunikationsteilnehmers in dieser Norm durch einen individuellen Namen voneinander unterschieden. Zur Wahrung der Eindeutigkeit darf ein Anwendungsprozeßname je Kommunikationsteilnehmer nur einmal verwendet werden.

Der lokale Informationsaustausch in einem Kommunikationsteilnehmer zwischen Anwendungsprozeß und Kommunikationssystem erfolgt über die Anwendungsprozeßschnittstelle³⁾ [API]. In dieser Norm wird der lokale Informationsaustausch durch sogenannte Anwendungsdienstprimitive beschrieben, deren Funktion in 4.5 erläutert wird.

ANMERKUNG: Wie es ein Kommunikationsteilnehmer intern erreicht, das richtige Telegramm zeitgerecht zu senden und ob er, wie es im Kommunikationsmodell beschrieben ist, intern tatsächlich in einen oder in mehreren voneinander abgegrenzten Anwendungsprozessen und in ein "ausgeprägtes" Kommunikationssystem gegliedert ist, mit dem über eine spezielle Anwendungsprozeßschnittstelle [API] Informationen ausgetauscht werden, ist nach der OSI-Philosophie, die dieser Norm zugrunde liegt, die Angelegenheit des Herstellers. DIN 66348-3 ist, ebenso wie andere OSI-Normen, grundsätzlich implementierungsneutral. Sie definiert mit dem Kommunikationsmodell lediglich das "Referenz"-Modell, das zur Beschreibung der Mechanismen dient, mit dem die genormten Effekte (Telegrammformat, Protokoll, Zeitverhalten) erzielt werden können. Es ist aber in keiner Weise zwingend, dem Kommunikationsmodell bei der technischen Umsetzung zu folgen, solange die anhand des Kommunikationsmodells beschriebenen Eigenschaften (Telegrammformat, Protokoll) eingehalten werden.

4.3 Kommunikationsbeziehung

Unabhängig davon, ob der Anwendungstelegrammtransport zwischen den Kommunikationspartnern über eine Punkt-zu-Punkt-Verbindung⁴⁾ oder über einen Bus⁵⁾ erfolgt, muß zum Informationsaustausch eine spezifische Kommunikationsbeziehung zwischen den beiden am Informationsaustausch beteiligten Anwendungsprozessen aufgebaut werden. Das heißt, die Kommunikationsbeziehung muß initialisiert und nach Abschluß aller vorgesehenen Informationsaustauschvorgänge wieder abgebaut werden. Die Zeit, die zwischen dem Auf- und Abbau einer Kommunikationsbeziehung vergehen darf, ist in keiner Weise technisch begrenzt. Sie wird durch die Informationsverarbeitungsaufgabe bestimmt.

Das in dieser Norm beschriebene Kommunikationsmodell setzt voraus, daß eine spezifische Kommunikationsbeziehung immer nur zwischen zwei Anwendungsprozessen besteht. Zwischen diesen beiden Anwendungsprozessen darf dabei nur eine Kommunikationsbeziehung aufgebaut werden. Ein Anwendungsprozeß kann aber gleichzeitig mehrere, voneinander unabhängige Kommunikationsbeziehungen zu verschiedenen anderen Anwendungsprozessen unterhalten. Jede dieser Kommunikationsbeziehungen muß separat mit den dazu vorgesehenen Anwendungsdiensten aufgebaut werden⁶⁾.

Um zu ermöglichen, daß über die gleiche elektrische Kommunikationsschnittstelle eines Kommunikationsteilnehmers mehrere Kommunikationsbeziehungen existieren können, damit

- mehrere voneinander unabhängige Anwendungsprozesse in einem Kommunikationsteilnehmer erreicht werden können und/oder
- ein Anwendungsprozeß von mehreren assoziierten Anwendungsprozessen in anderen Kommunikationsteilnehmern erreicht werden kann,

werden in dieser Norm nach der Darstellung in Bild 2 "Verbindungsnummern" benutzt.

Für jede in einem verteilten Informationsverarbeitungssystem vorgesehene Kommunikationsbeziehung wird eine individuelle Verbindungsnummer fest zugeteilt, das heißt, sie wird reserviert. Durch diese Verbindungsnummer wird die Kommunikationsbeziehung identifiziert (siehe 12.3).

ANMERKUNG: Wie in Abschnitt 12 detailliert beschrieben wird, wird jedem Anwendungstelegramm, das in einer Kommunikationsbeziehung ausgetauscht wird, die dieser Kommunikationsbeziehung zugeordnete Verbindungsnummer zur Identifikation vorangestellt.

3) OSI-Modell: Application Program Interface [API]

4) siehe z.B. DIN 66348-1

5) siehe z.B. DIN 66348-2

6) detailliert beschrieben in 6.1

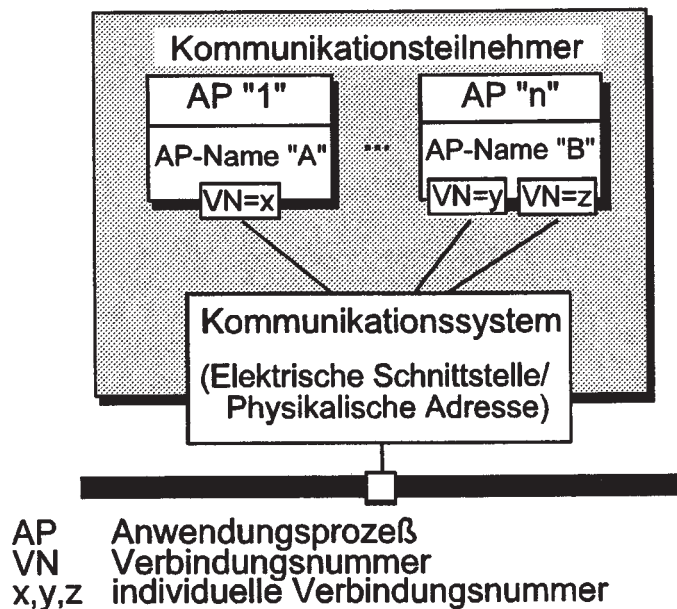


Bild 2: Verbindungsnummern (Beispiel: Busanschlussschaltung des Kommunikationsteilnehmers)

4.4 Auftraggeber und Auftragnehmer

Die beiden miteinander in einer Kommunikationsbeziehung stehenden Anwendungsprozesse werden in Abhängigkeit von der ihnen in der Kommunikationsbeziehung bei der Kommunikation zukommenden Rolle als der Auftraggeber⁷⁾ und als der Auftragnehmer⁸⁾ bezeichnet.

Der Auftraggeber ist der Anwendungsprozeß, der bei einem anderen Anwendungsprozeß, der im Rahmen dieser Norm als Auftragnehmer bezeichnet wird, die Erbringung einer bestimmten Informationsverarbeitungsdienstleistung anfordert⁹⁾.

Je nach Implementierung kann ein Anwendungsprozeß in einer Kommunikationsbeziehung gleichzeitig sowohl Auftraggeber als auch Auftragnehmer sein. Eine Kommunikationsbeziehung besitzt somit, bezogen auf die Möglichkeit zur bidirektionalen Dienstausführungsanforderung, zwei Kommunikationsrichtungen.

ANMERKUNG: Die Rolle eines Anwendungsprozesses als Auftraggeber oder Auftragnehmer ist nicht an eine bestimmte Rolle beim Datentransport (z.B. Master-Slave/Leitstation-Teilnehmerstation) gebunden.

4.5 Anwendungsdienste

4.5.1 Allgemeines

Als Anwendungsdienst wird im Rahmen dieser Norm eine spezifische Informationsverarbeitungsdienstleistung bezeichnet, die ein Anwendungsprozeß

- in der Rolle als Auftragnehmer nach einer Aufforderung durch einen assoziierten Auftraggeber oder
- auf Eigeninitiative beim Eintreten vordefinierter Ereignisse (Auslösebedingungen)

zur Erfüllung einer gemeinsamen Informationsverarbeitungsaufgabe für einen assoziierten Anwendungsprozeß erbringt.

ANMERKUNG: Ein Anwendungsdienst kann beispielsweise zur Ermittlung des Gerätestatus dienen.

In Abhängigkeit davon, ob die Anwendungsdienstausführung auf Anforderung durch einen Auftraggeber oder auf Eigeninitiative erfolgt, unterscheidet diese Norm zwischen den "bestätigten" und den "unbestätigten" Anwendungsdiensten.

7) MMS: Requesting MMS-User

8) MMS: Responding MMS-User

9) siehe 4.5.2

4.5.2 Bestätigte Anwendungsdienste

Ein bestätigter Anwendungsdienst wird grundsätzlich nur nach einer vorher eingetroffenen Anforderung ausgeführt. Die Anforderung erfolgt dadurch, daß vom Kommunikationssystem des Auftraggebers ein "Anforderungstelegramm" an das Kommunikationssystem des Auftragnehmers gesendet wird. Nach erfolgreicher und in bestimmten Fällen auch bei einer nur teilweise erfolgreichen Dienstbringung antwortet das Kommunikationssystem des Auftragnehmers mit einem "Antworttelegramm". Scheitert die Dienstbringung aufgrund eines fundamentalen Fehlers¹⁰⁾, so antwortet das Kommunikationssystem des Auftragnehmers mit einem "Fehlertelegramm".

Unter bestimmten Voraussetzungen kann es geräte- bzw. anwendungstechnisch erforderlich sein, daß sich in einem Auftragnehmer gleichzeitig mehrere Dienstausführungsanforderungen des gleichen Auftraggebers in der Ausführung befinden. Das heißt, bevor eine Dienstausführung mit dem Empfang eines Antwort- oder Fehlertelegramms abgeschlossen worden ist, dürfen, wenn der Auftragnehmer diese gestattet, bereits eine oder mehrere neue Dienstausführungen mit Anforderungstelegrammen durch den Auftraggeber beantragt werden.

Die Gesamtzahl der gleichzeitig "offenen" Dienstausführungsanforderungen wird in dieser Norm sowohl für Anwendungsprozesse in der Rolle als Auftraggeber als auch für solche in der Rolle als Auftragnehmer auf jeweils maximal 63 begrenzt. Es obliegt dem Hersteller für das jeweils zu entwickelnde Gerät bzw. System festzulegen, ob sich in einem Anwendungsprozeß gleichzeitig mehrere Dienstausführungsanforderungen in der Ausführung befinden dürfen oder nicht, und wenn ja, wieviele offene Dienstausführungsanforderungen dies sein dürfen. Die vom Anwender festzulegende Anzahl muß zwischen 1 und 63 liegen¹¹⁾.

Die in einer Kommunikationsbeziehung erlaubte maximale Anzahl offener Dienstausführungsanforderungen wird von den beiden Kommunikationspartnern während des Aufbaus der Kommunikationsbeziehung festgelegt. Dabei wird, sofern die jeweilige Rolle unterstützt wird, nach der Rolle eines Anwendungsprozesses als Auftraggeber und der Rolle als Auftragnehmer unterschieden.

4.5.3 Unbestätigte Anwendungsdienste

Ein unbestätigter Anwendungsdienst wird für einen Kommunikationspartner von einem Anwendungsprozeß unaufgefordert, das heißt spontan auf Eigeninitiative beim Eintreten gewisser Ereignisse erbracht. Die Ereignisse, die zur Ausführung unbestätigter Anwendungsdienste führen, sind bei der Implementierung festzulegen.

Das Ergebnis der Ausführung eines unbestätigten Anwendungsdienstes wird dem Kommunikationspartner durch das Senden eines "Meldungstelegramms" mitgeteilt.

In Verbindung mit den "unbestätigten Anwendungsdiensten" werden die Kommunikationspartner nicht als "Auftragnehmer" und "Auftraggeber" sondern als "Sender" und "Empfänger" des Meldungstelegramms bezeichnet. Dabei ist der Sender der Kommunikationsteilnehmer, der den Anwendungsdienst ausgeführt hat.

4.6 Anwendungsdienstprimitive

Der lokale Informationsaustausch zwischen einem Anwendungsprozeß und dem Kommunikationssystem in einem Kommunikationsteilnehmer wird in dieser Norm durch Anwendungsdienstprimitive beschrieben. Unter einem Anwendungsdienstprimitiv wird in Anlehnung an die allgemeingültige Dienstprimitiv-Definition in ISO/TR 8509 die implementierungsneutrale Beschreibung einer Vorgehensweise verstanden, deren Befolgung zu einer spezifischen lokalen Interaktion zwischen Anwendungsprozeß und Kommunikationssystem führt. Jede Dienstprimitiv-Beschreibung beinhaltet die Beschreibung einer spezifischen Gruppe von Parametern, die der beim Informationsaustausch übergebenen Datenstruktur entspricht.

ANMERKUNG: Durch die implementierungsneutrale Beschreibung der Anwendungsdienstprimitive, das heißt die bewußt unterbleibende Festlegung der Art und Weise der Implementierung des lokalen Informationsaustauschs, ermöglicht es, diese Norm jedem Anwender, den lokalen Informationsaustausch in der von ihm bevorzugten Form zu realisieren. Anwendungsdienstprimitive können beispielsweise als Funktions- oder Prozeduraufrufe realisiert werden. Diese Aufrufe werden in der Anwendungsprozeß-Software bzw. in der Software des Kommunikationssystems verwendet, um mit dem lokalen Partner (Kommunikationssystem oder Anwendungsprozeß) Informationen auszutauschen.

Diese Norm unterscheidet die vier in Bild 3 dargestellten Dienstprimitiv-Grundtypen¹²⁾

¹⁰⁾ Die Beschreibung der fundamentalen Fehler erfolgt in Abschnitt 11

¹¹⁾ MMS: maximale Anzahl offener Dienste < 231

¹²⁾ Wegen der Eindeutigkeit wird im folgenden für den Begriff Anwendungsdienstprimitiv der Begriff Dienstprimitiv verwendet.

- Anforderung-Dienstprimitiv¹³⁾,
- Anzeige-Dienstprimitiv¹⁴⁾,
- Antwort-Dienstprimitiv¹⁵⁾ und
- Bestätigung-Dienstprimitiv¹⁶⁾

voneinander.

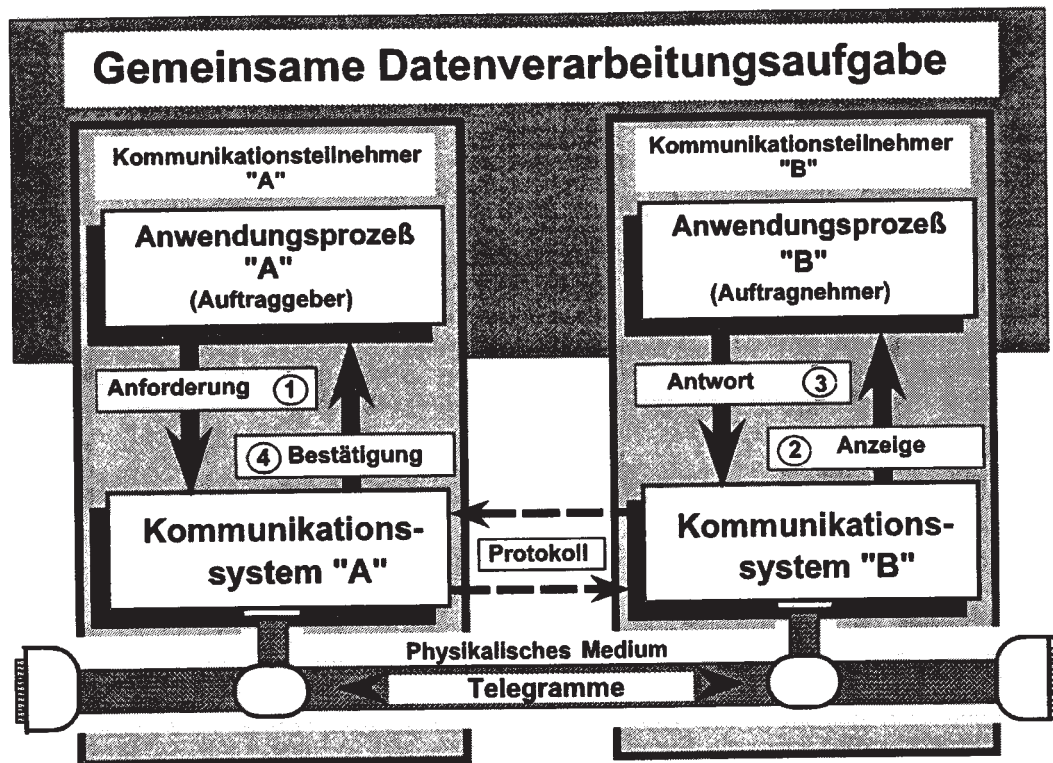


Bild 3: Dienstprimitiv-Grundtypen in einer Kommunikationsbeziehung

Die Benennung Anforderung-Dienstprimitiv steht für eine Vorgehensweise, die ein Anwendungsprozeß befolgt, um entweder

- in der Rolle als Auftraggeber sein lokales Kommunikationssystem zu veranlassen, beim Kommunikationspartner die Ausführung eines bestätigten Anwendungsdienstes zu beantragen oder
- sein lokales Kommunikationssystem zu veranlassen, dem Kommunikationspartner die unangeforderte, d.h. spontane Ausführung eines unbestätigten Anwendungsdienstes mitzuteilen.

Die Benennung Anzeige-Dienstprimitiv steht für eine Vorgehensweise, die ein Kommunikationssystem befolgt, um entweder einem lokalen Anwendungsprozeß

- in der Rolle als Auftragnehmer das Eintreffen einer Anwendungsdienstausführungsanforderung vom Kommunikationspartner,
- das Eintreffen der Mitteilung über die spontane Ausführung eines unbestätigten Anwendungsdienstes durch den Kommunikationspartner oder
- das Eintreten eines vordefinierten Ereignisses im lokalen Kommunikationssystem

anzuzeigen.

Die Benennung Antwort-Dienstprimitiv steht für eine Vorgehensweise, die ein Anwendungsprozeß in der Rolle als Auftragnehmer befolgt, um seinem lokalen Kommunikationssystem den Abschluß und das Ergebnis einer vom Kommunikationspartner angeforderten Anwendungsdienstausführung mitzuteilen.

Die Benennung Bestätigung-Dienstprimitiv steht für eine Vorgehensweise, die ein Kommunikationssystem befolgt, um einem lokalen Anwendungsprozeß

13) OSI: Request

14) OSI: Indication

15) OSI: Response

16) OSI: Confirmation

- in der Rolle als Auftraggeber das Eintreffen einer Antwort als das Ergebnis einer beim Kommunikationspartner erfolgten, zuvor angeforderten Anwendungsdienstausführung oder
 - das Eintreten einer Störung
- mitzuteilen.

4.7 Kommunikationsablauf bei der Anwendungsdienstausführung

Zwischen dem abstrakt durch Dienstprimitive beschriebenen internen Informationsfluß in einem Kommunikationsteilnehmer und dem durch Telegrammübertragung erfolgenden externen Informationsfluß zwischen den Kommunikationspartnern besteht ein eindeutiger Zusammenhang, der in diesem Abschnitt dargestellt wird.

4.7.1 Bestätigte Anwendungsdienste

Das Bild 4 zeigt den systeminternen und externen Kommunikationsablauf bei der Ausführung eines bestätigten Anwendungsdienstes. Im Bild wird der Auftraggeber als System "A" bezeichnet. Er besteht aus dem Anwendungsprozeß "A" und dem Kommunikationssystem "A". Anwendungsprozeß "B" und Kommunikationssystem "B" bilden das System "B", das den Auftragnehmer darstellt.

ANMERKUNG: Unter dem "Aufruf" oder der "Verwendung" eines Dienstprimitivs wird in dieser Norm verstanden, daß im Kommunikationsteilnehmer eine lokale Interaktion erfolgt, die durch das betreffende Dienstprimitiv beschrieben wird.

Mit einem dem jeweiligen Anwendungsdienst zugeordneten Anforderung-Dienstprimitiv beauftragt der Anwendungsprozeß "A" sein lokales Kommunikationssystem "A" mit der Anforderung einer Dienstausführung im Auftragnehmer (System "B"). Die Gruppe der Parameter, die die Beschreibung eines Anforderung-Dienstprimitivs umfaßt, wird in dieser Norm als "Argument" bezeichnet.

Als Reaktion auf die Beauftragung durch den lokalen Anwendungsprozeß überträgt das Kommunikationssystem "A" ein spezifisches, das heißt ein dem entsprechenden Anwendungsdienst zugeordnetes "Anforderungstelegramm" an das Kommunikationssystem "B". Das Kommunikationssystem "B" übermittelt die mit dem Anforderungstelegramm eingetroffene Dienstauführungsanforderung durch ein spezifisches Anzeige-Dienstprimitiv an den lokalen Anwendungsprozeß "B". Die Gruppe der Parameter eines Anzeige-Dienstprimitivs entspricht in der Regel der des zugeordneten Anforderung-Dienstprimitivs und wird deshalb ebenfalls als "Argument" bezeichnet.

Der Auftragnehmer (Anwendungsprozeß) prüft, ob die Anwendungsdienstausführung möglich ist oder nicht und führt diese gegebenenfalls durch. Der Anwendungsprozeß "B" teilt dem lokalen Kommunikationssystem "B" mit dem dem jeweiligen Anwendungsdienst zugeordneten Antwort-Dienstprimitiv das Resultat der angeforderten Dienstaufführung mit. Das Antwort-Dienstprimitiv besitzt bei erfolgreicher bzw. bei teilweise erfolgreicher Dienstaufführung eine Gruppe von Parametern, die in dieser Norm als "Ergebnis(+)" bezeichnet wird. Ist die Dienstaufführung aufgrund eines fundamentalen Fehlers bzw. einer Störung nicht möglich, besitzt das Dienstprimitiv die Parametergruppe "Ergebnis(-)".

Das Kommunikationssystem "B" überträgt seine Reaktion auf die vom Kommunikationspartner angeforderte Dienstaufführung mit einem Anwendungstelegramm an das Kommunikationssystem des Auftraggebers. War die Dienstaufführung erfolgreich bzw. teilweise erfolgreich, das heißt umfaßt das Antwort-Dienstprimitiv die Parametergruppe Ergebnis(+), so wird ein dem entsprechenden Anwendungsdienst zugeordnetes "Antworttelegramm" gesendet. War die Dienstaufführung nicht erfolgreich, das heißt besitzt das Antwort-Dienstprimitiv die Parametergruppe Ergebnis(-), so wird ein "Fehlertelegramm" gesendet.

Den Empfang eines Antworttelegramms oder eines Fehlertelegramms teilt das Kommunikationssystem "A" dem lokalen Anwendungsprozeß "A" mit einem Bestätigung-Dienstprimitiv mit. Wird vom Kommunikationssystem "A" ein Antworttelegramm empfangen, so besitzt das dem Telegramm zugeordnete Bestätigung-Dienstprimitiv in Analogie zum Antwort-Dienstprimitiv die Parametergruppe Ergebnis(+). Wird ein Fehlertelegramm empfangen, so besitzt das Bestätigung-Dienstprimitiv die Parametergruppe Ergebnis(-).

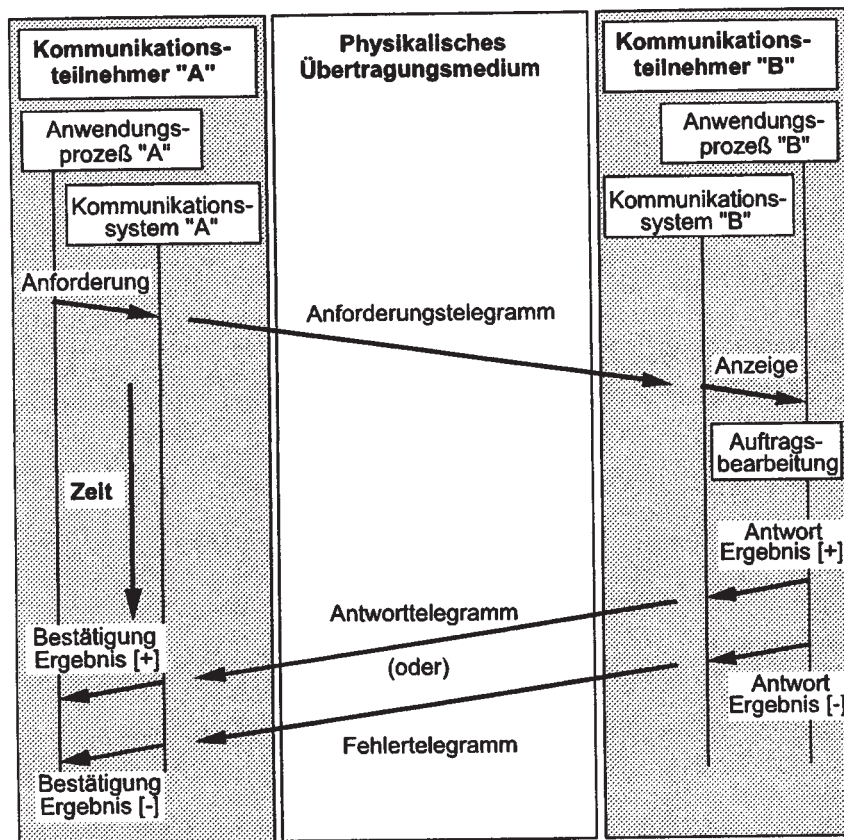


Bild 4: Kommunikationsablauf bei einem bestätigten Anwendungsdienst

Die "Verwendung" der einzelnen, einem Anwendungsdienst bzw. einer Anwendungsdienstgruppe zugeordneten Dienstprimitive wird in dieser Norm durch Zustandsübergangsdiagramme beschrieben. In Bild 5 ist das allgemeine Zustandsübergangsdiagramm für einen bestätigten Anwendungsdienst aus Auftragnehmersicht dargestellt.

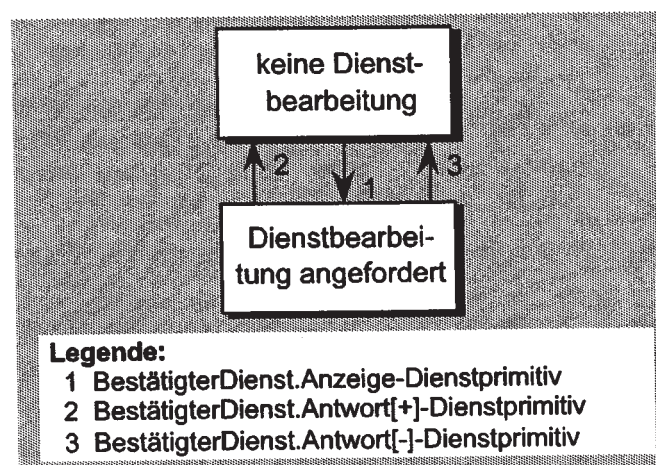


Bild 5: Zustandsübergangsdiagramm eines bestätigten Anwendungsdienstes aus Auftragnehmersicht

4.7.2 Unbestätigte Anwendungsdienste

Das Bild 6 zeigt den systeminternen und externen Kommunikationsablauf bei einem unbestätigten Anwendungsdienst. Bei unbestätigten Anwendungsdiensten wird nicht zwischen Auftragnehmer und Auftraggeber unterschieden.

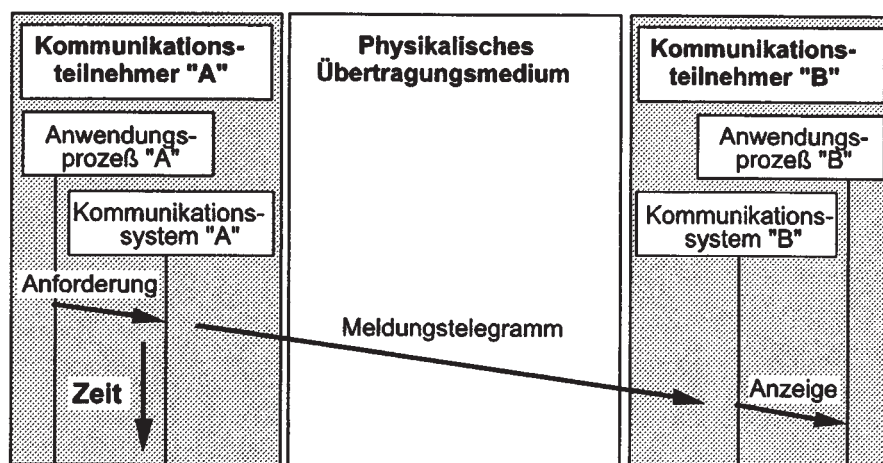


Bild 6: Kommunikationsablauf bei einem unbestätigten Anwendungsdienst

Mit einem dem jeweiligen unbestätigten Anwendungsdienst zugeordneten Anforderung-Dienstprimitiv beauftragt der Anwendungsprozeß "A" sein lokales Kommunikationssystem "A" mit der Übertragung einer Meldung an den Kommunikationspartner (System "B"). Als Reaktion auf diese Beauftragung überträgt das Kommunikationssystem "A" ein dem entsprechenden Anwendungsdienst zugeordnetes "Meldungstelegramm" an das Kommunikationssystem "B". Das Kommunikationssystem "B" übermittelt mit einem Anzeige-Dienstprimitiv das Eintreffen eines Meldungstelegramms und dessen Inhalt an den lokalen Anwendungsprozeß "B".

In Bild 7 ist das Zustandsübergangsdiagramm eines unbestätigten Anwendungsdienstes im Anwendungsprozeß eines Kommunikationsteilnehmers dargestellt.

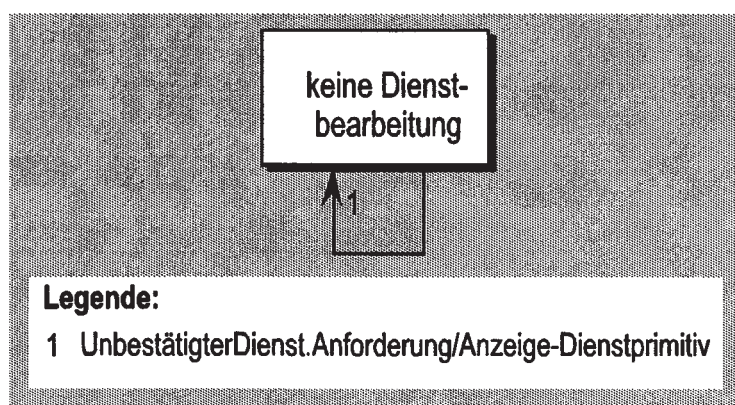


Bild 7: Zustandsübergangsdiagramm eines unbestätigten Anwendungsdienstes

4.7.3 Gestörte Anwendungsdienstauführung

In der Regel wird die Bestätigung einer Dienstauführungsanforderung nach dem vorhergehenden Empfang eines vom Partnersystem gesendeten Antwort- oder Fehlertelegramms erfolgen. Das muß aber nicht immer der Fall sein. Ist beispielsweise, wie in Bild 8 dargestellt, die Telegrammübertragung gestört und kann daher die nach einer Dienstauführungsanforderung erforderliche Telegramm-Sendung vom lokalen Kommunikationssystem nicht durchgeführt werden, so wird auch dieses "lokale" Problem dem Anwendungsprozeß mit einem dem jeweiligen Anwendungsdienst zugeordneten Bestätigung-Dienstprimitiv unter Angabe der Ursache mitgeteilt.

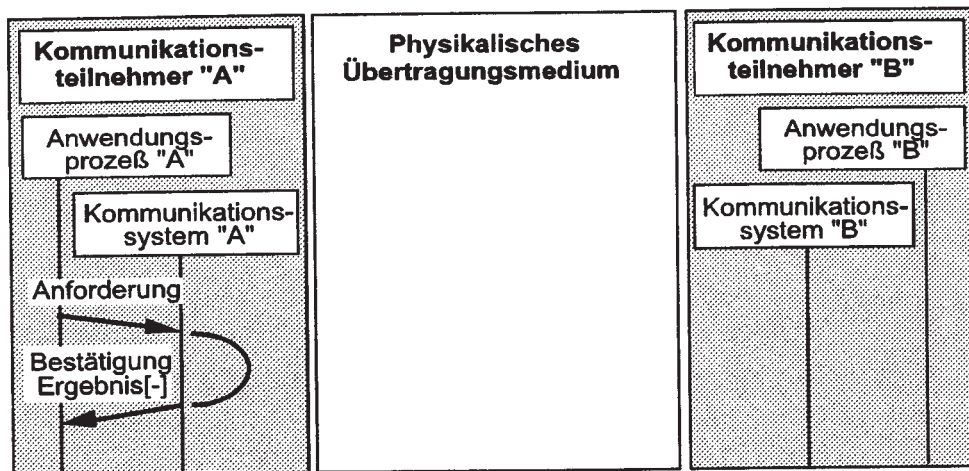


Bild 8: Interner Kommunikationsablauf bei gestörter Telegrammübertragung

4.7.4 Hinweise zur Zustandsübergangsdiagrammdarstellung

In dieser Norm wird für die Darstellung der Zustandsübergangsdiagramme festgelegt:

- Wegen der umkehrbaren Eindeutigkeit werden die Zustandsübergangsdiagramme der bestätigten Anwendungsdienste, wenn nicht besonders vermerkt, grundsätzlich nur aus Auftragnehmersicht dargestellt. Um zur Darstellung aus Auftraggebersicht zu gelangen, ist an den Zustandsübergängen die Bezeichnung 'Anzeige-Dienstprimitiv' bzw. die entsprechende Abkürzung durch 'Anforderung-Dienstprimitiv' zu ersetzen. Analog hierzu ist 'Antwort-Dienstprimitiv' durch 'Bestätigung-Dienstprimitiv' zu ersetzen.
- Zur Vereinfachung der Darstellung werden für die Anwendungsdienstprimitiv-Grundtypen die folgenden Abkürzungen verwendet:

Anf	Anforderung-Dienstprimitiv;
Anz	Anzeige-Dienstprimitiv;
Ant(+)	Antwort-Dienstprimitiv mit der Parametergruppe Ergebnis(+);
Ant(-)	Antwort-Dienstprimitiv mit der Parametergruppe Ergebnis(-);
Bes(+)	Bestätigung-Dienstprimitiv mit der Parametergruppe Ergebnis(+);
Bes(-)	Bestätigung-Dienstprimitiv mit der Parametergruppe Ergebnis(-).

- Bei ungestörtem Kommunikationssystem besteht folgender Zusammenhang zwischen der Dienstprimitiv Darstellung im Zustandsübergangsdiagramm (Auftragnehmersicht) und der Telegrammübertragung:

Anf	Anforderungs-/Meldungstelegramm senden;
Anz	Anforderungs-/Meldungstelegramm empfangen;
Ant(+)	Antworttelegramm senden;
Ant(-)	Fehlertelegramm senden;
Bes(+)	Antworttelegramm empfangen;
Bes(-)	Fehlertelegramm empfangen.

4.8 Struktur der Anwendungsdienst-Tabellen

Diese Norm benutzt Tabellen zur Beschreibung der Dienstprimitiv-Parameter der in den Abschnitten 6 bis 10 beschriebenen Anwendungsdienste. In Bild 9 ist an einem Beispiel der prinzipielle Aufbau einer solchen Anwendungsdiensttabelle dargestellt.

Jede Tabelle kann aus zwei oder drei Spalten bestehen. In der linken Spalte werden die Parameternamen angegeben. Die restlichen Spalten sind für die Anwendungsdienstprimitive vorgesehen. Prinzipiell kann je eine Spalte für das Anforderungs-, das Anzeige-, das Antwort- und das Bestätigung-Dienstprimitiv vorgesehen sein. Da die Parameter der Anforderungs- und Anzeige-Dienstprimitive, bzw. die der Antwort- und Bestätigung-Dienstprimitive im allgemeinen gleich sind, werden diese Dienstprimitive in dieser Norm in der Regel zu jeweils einer gemeinsamen Spalte zusammengefaßt. Bei unbestätigten Anwendungsdiensten fehlen die Spalten für das Antwort- und das Bestätigung-Dienstprimitiv generell, da diese Dienstprimitive bei unbestätigten Anwendungsdiensten nicht existieren.

In der oberen Zeile des rechten Feldes einer Anwendungsdiensttabelle ist jeweils der Anwendungsdienstname angegeben, der immer mit einem Punkt abgeschlossen ist.

ANMERKUNG: Für OSI-konforme Dienstprimitive wird grundsätzlich die Notation "Anwendungsdienstname.Dienstprimitiv-Grundtyp" verwendet.

Die in den Tabellen aufgeführten Parameter werden bei bestätigten Anwendungsdiensten immer zu den drei Parametergruppen Argument, Ergebnis(+) und Ergebnis(-) zusammengefaßt, die jeweils aus einem der drei Parameter Argument, Ergebnis(+) oder Ergebnis(-) und gegebenenfalls zusätzlichen Unterparametern gebildet werden. Die Parametergruppen werden in den Tabellen zur Erhöhung der Lesbarkeit durch eine horizontale Linie voneinander getrennt. Bei unbestätigten Anwendungsdiensten existiert nur die Parametergruppe Argument.

Durch Einrückung des Parameternamens wird in den Tabellen zum Ausdruck gebracht, daß der Parameter ein Unterparameter eines anderen Parameters ist. So sind beispielsweise die Parameter mit den Namen Parameter 7 und Parameter 8 in Bild 9 beide Unterparameter des Parameters mit dem Namen Parameter 6.

Die Einträge, die in eckigen Klammern stehen, sind Kommentare und beziehen sich auf eine Gruppe nachfolgender Parameter.

Parametername	Anwendungsdienstname.			
	Anf	Anz	Ant	Bes
Argument	E	E	-	-
Parameter 1	E	E	-	-
[Kommentarzeile:]	[]	[]	[]	[]
Parameter 2	B	B	-	-
Parameter 3	B	B	-	-
Ergebnis(+)	-	-	A	A
Parameter 4	-	-	E	E
Parameter 5	-	-	O	O
Ergebnis(-)	-	-	A	A
Parameter 6	-	-	E	E
Parameter 7	-	-	A	A
Parameter 8	-	-	A	A
Parameter 9	-	-	E	E

Legende:	A	Auswahl;
	B	Bedingung;
	E	Erforderlich;
	O	Option;
	-	Parameter ist nicht vorgesehen;
	Anf	Anforderung-Dienstprimitiv;
	Anz	Anzeige-Dienstprimitiv;
	Ant	Antwort-Dienstprimitiv;
	Bes	Bestätigung-Dienstprimitiv.

Bild 9: Beispiel einer Anwendungsdienst-Tabelle

ANMERKUNG 1: Die Parameter Argument, Ergebnis(+) und Ergebnis(-) werden in den Anwendungstelegrammen durch die Telegrammtypkennung bzw. für bestimmte Anwendungsdienste zusätzlich durch die Dienstkennung repräsentiert (siehe Abschnitte 6 - 11). Da sich die Werte der Parameter eindeutig aus den jeweiligen Telegrammabbildungen ergeben, werden sie in den Anwendungsdienstbeschreibungen nicht angegeben.

ANMERKUNG 2: Es ist zu beachten, daß verschiedene Parameternamen in den Anwendungsdienst-Tabellen aufgrund ihrer Länge zweizeilig geschrieben werden. Dabei erfolgt in der zweiten Zeile eine Einrückung. Zur Vermeidung der Verwechslung mit einem Unterparameter fehlt in solchen Fällen in der ersten Zeile des Parameters am Schnittpunkt von Parametername und Dienstprimitivname der Eintrag.

Im Schnittpunkt einer Zeile (Parametername) mit einer Spalte (Dienstprimitivname) befindet sich jeweils ein Eintrag mit folgender Bedeutung:

- A Der Parameter ist ein Auswahlparameter, das heißt, er ist einer von mindestens zwei gleichberechtigten wahlweise jeweils unter bestimmten Voraussetzungen benutzbaren Parametern. Im BEISPIEL: Das Antwort- bzw. das Bestätigung-Dienstprimitiv eines Anwendungsdienstes beinhaltet, je nach Ergebnis der Dienstauführung, entweder die Parameter der Parametergruppe Ergebnis(+) oder die Parameter der Parametergruppe Ergebnis(-). Die Auswahlmöglichkeit zwischen beiden Parametergruppen wird in den Anwendungsdiensttabellen durch den Eintrag "A" in der Zeile Ergebnis(+) bzw. Ergebnis(-) angezeigt¹⁷⁾.

¹⁷⁾ Hierbei ist zu beachten, daß Ergebnis(+) und Ergebnis(-) in der Spalte Parameternamen die gleiche Einrückung besitzen.

- B Der Parameter ist nur dann im Dienstprimitiv vorhanden, wenn eine bestimmte Bedingung erfüllt ist. Ist die Bedingung nicht erfüllt, so ist im Gegensatz zum Auswahlparameter (s.o.) weder dieser Parameter noch ein Ersatzparameter vorhanden. Im BEISPIEL: Die Parametergruppe Argument umfaßt neben dem immer vorhandenen Parameter mit dem Namen Parameter 1 in Abhängigkeit von gewissen Bedingungen entweder die beiden zusätzlichen Parameter mit den Namen Parameter 2 und Parameter 3 oder nur einen dieser beiden Parameter oder keinen¹⁸⁾.
- E Die Anwesenheit des Parameters ist zwingend erforderlich.
- O Die Anwesenheit des Parameters ist eine Benutzeroption. BEISPIEL: Beim Parameter mit dem Namen Parameter 5 kann der Implementierer frei entscheiden, ob dieser Parameter im Dienstprimitiv vorgesehen sein soll oder nicht.
- Der Parameter ist in diesem Dienstprimitiv nicht vorgesehen.

Für alle in dieser Norm wiedergegebenen Anwendungsdienst-Tabellen gilt die in Bild 9 dargestellte Legende.

5 Codierregeln

Dieser Abschnitt beinhaltet die allgemeinen Regeln und Vereinbarungen, nach denen die in dieser Norm definierten Anwendungstelegramme aufgebaut und codiert sind.

5.1 Trennung von Telegrammteilen

5.1.1 Allgemeines

Die in dieser Norm definierten Telegramme bestehen in der Regel aus mehreren aneinandergereihten "Datenfeldern", die in logisch zusammengehörende Gruppen zu "Telegrammabschnitten" bzw. zu "Telegrammunterabschnitten" zusammengefaßt sein können.

Datenfelder sind die kleinsten zusammenhängenden Teile eines Anwendungstelegramms. Beispiele für Datenfelder sind Namen oder Zahlen, die jeweils aus mindestens einem Zeichen gebildet werden.

Ein Telegrammunterabschnitt ist ein Telegrammteil, der aus mindestens zwei logisch zusammengehörenden Datenfeldern besteht. Ein Beispiel für einen Telegrammunterabschnitt ist eine Namenliste, die aus mehreren Datenfeldern - den Namen - besteht.

Ein Telegrammabschnitt ist ein Telegrammteil, der aus mindestens zwei logisch zusammengehörenden Elementen besteht. Elemente in Telegrammabschnitten sind Datenfelder und Telegrammunterabschnitte. Ein Beispiel für einen Telegrammabschnitt ist ein Telegrammteil, der eine Namenliste und ein Datenfeld enthält, in dem die Anzahl der in der Namenliste enthaltenen Namen angegeben ist.

Die Länge der Datenfelder, Telegrammabschnitte und Telegrammunterabschnitte in einem Anwendungstelegramm kann fest oder variabel sein. Deshalb sieht diese Norm zwei verschiedene Verfahren zum Erkennen des Endes der verschiedenen Telegrammteile vor. Beide Verfahren dürfen in einem Anwendungstelegramm gleichzeitig verwendet werden.

FESTLEGUNG: In dieser Norm werden die Zeichen nach DIN 66003 zur Unterscheidung von Datenfeldnamen in die Zeichen < und > eingeschlossen. BEISPIEL: <DC2> steht für das Gerätesteuerzeichen 2 (Device Control 2), das in DIN 66003 an der Position Spalte 1/Zeile 2 zu finden ist.

5.1.2 Trennung von Telegrammteilen mit fester Länge

Das Ende von Telegrammteilen mit vorgegebener fester Länge wird in den Anwendungstelegrammen nicht markiert. Der das Telegramm empfangende Kommunikationsteilnehmer bestimmt aus der Kenntnis des Telegrammaufbaus die Position der einzelnen Datenfelder durch Abzählen der Zeichen im Telegramm.

5.1.3 Trennung von Telegrammteilen mit variabler Länge

Das Ende von Telegrammteilen mit variabler Länge wird in den Telegrammen durch Separatoren gekennzeichnet. Als Separatoren werden das Gruppen-Trennzeichen <GS>¹⁹⁾, das Untergruppen-Trennzeichen <RS>²⁰⁾ und das Teilgruppen-Trennzeichen <US>²¹⁾ verwendet. Das Haupt-

18) Die Bedingung, die zur An- bzw. Abwesenheit von Parameter 2 führt, muß nicht an die Bedingung gebunden sein, die zur An- bzw. Abwesenheit von Parameter 3 führt.

19) Group Separator, Platz 1/13 nach DIN 66003:1974-06

20) Record Separator, Platz 1/14 nach DIN 66003:1974-06

21) Unit Separator, Platz 1/15 nach DIN 66003:1974-06

gruppen-Trennzeichen <FS>²²⁾ wird zur Markierung des Telegrammendes verwendet.

Die Separatoren unterscheiden sich in ihrer Wertigkeit voneinander. Aufgrund seiner ausschließlichen Verwendung als Telegrammendezeichen hat das Zeichen <FS> die höchste Wertigkeit. Die zweithöchste Wertigkeit hat das Zeichen <GS>, gefolgt von <RS>. Die niedrigste Wertigkeit ist dem Separatorzeichen <US> zugeordnet. Aus der Wertigkeit der Separatoren leitet sich ihre Verwendungsreihenfolge ab.

Dieser Norm liegt die in Bild 10 dargestellte Verwendungsreihenfolge für Separatorzeichen zur Begrenzung von Datenfeldern mit variabler Länge zugrunde.

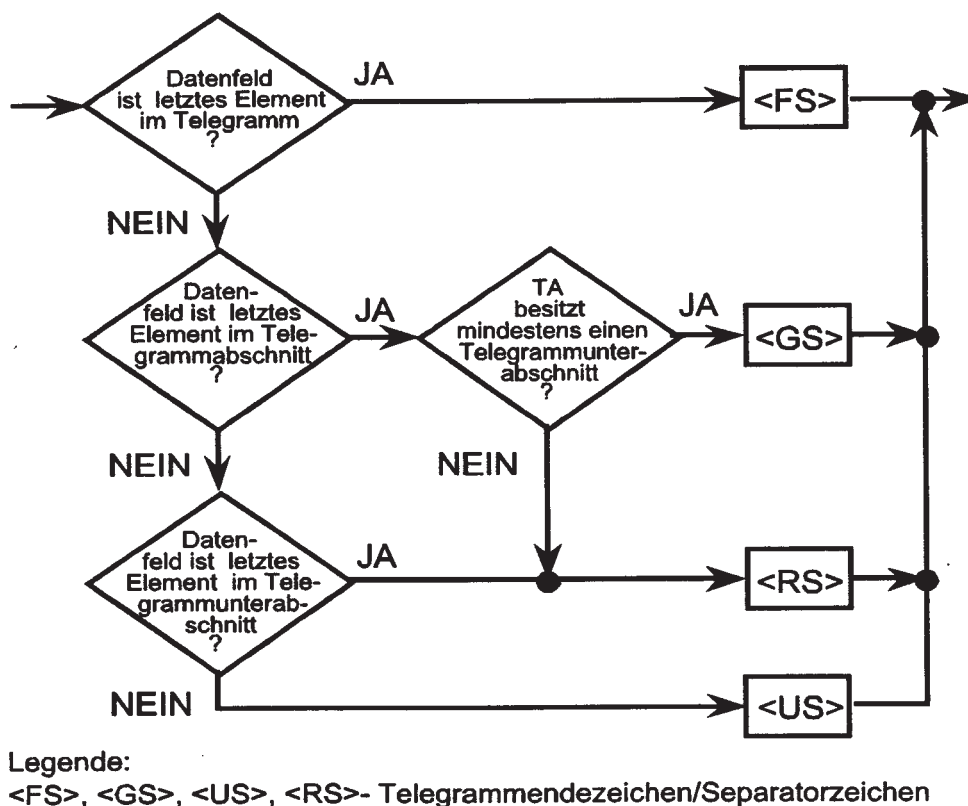


Bild 10: Flußdiagramm zur Separatorzeichen Verwendung

Wie das Bild 10 zeigt, werden Datenfelder nach folgenden Grundregeln durch Separatorzeichen begrenzt:

- Datenfelder werden, wenn sie nicht das letzte Element
 - in einem Anwendungstelegramm,
 - in einem Telegrammabschnitt oder
 - in einem Telegrammunterabschnitt
 sind, mit dem Separatorzeichen <US> abgeschlossen.
- Datenfelder, die nicht das letzte Element im Anwendungstelegramm sind, die aber das letzte Element in einem Telegrammabschnitt oder in einem Telegrammunterabschnitt sind, werden zur Kennzeichnung des Abschnitt- bzw. Unterabschnittendes entweder durch das Separatorzeichen <RS> oder durch das Separatorzeichen <GS> abgeschlossen. Dabei sind folgende Regeln zu beachten:

Das Separatorzeichen <RS> wird dann verwendet,

 - wenn das Datenfeld das letzte Element in einem Telegrammabschnitt ist, der keine Unterabschnitte hat, oder
 - wenn das Datenfeld nicht das letzte Element im Telegrammabschnitt, aber das letzte Element in einem Telegrammunterabschnitt ist.

Das Separatorzeichen <GS> wird dann verwendet, wenn das Datenfeld das letzte Element in einem Telegrammabschnitt mit mindestens einem Unterabschnitt ist.

22) File Separator, Platz 1/12 nach DIN 66003:1974-06

- Ist ein Datenfeld das letzte Element in einem Anwendungstelegramm, so wird es nicht mit einem Separatorzeichen, sondern zur Kennzeichnung des Telegrammendes grundsätzlich mit dem Telegrammendezeichen <FS> abgeschlossen²³⁾.

Bezüglich der Verwendung der Separatoren bei der Übertragung von Variablenwerten sind die Verwendungsregeln in 5.3.12 zu beachten.

Diese Norm sieht vor, daß unter folgenden Bedingungen auf die Übertragung des Inhalts eines Datenfeldes verzichtet werden darf:

- ein im Anwendungstelegramm vorgesehenes Datenfeld, ein Telegrammabschnitt oder -unterabschnitt ist leer, das heißt hat keinen Inhalt²⁴⁾, oder
- das Datenfeld ist optional.

Ist die Übertragung eines Datenfeldes nicht erforderlich, so muß (zur Synchronisation der Telegramminterpretation beim Empfänger) das das Ende des Datenfeldes markierende Separatorzeichen trotzdem im Telegramm enthalten sein.

Braucht ein aus mehreren Datenfeldern bestehender Telegrammteil (Telegrammabschnitt/Telegrammunterabschnitt) nicht übertragen zu werden, so sind die Separatorzeichen, die die einzelnen Datenfelder innerhalb des Telegrammteils begrenzen, nicht zu übertragen. Das den gesamten Telegrammabschnitt, d.h. sein letztes Datenfeld begrenzende Separatorzeichen, muß jedoch zwingend übertragen werden.

BEISPIEL: Wird ein Telegrammunterabschnitt nicht übertragen, so wird im Anwendungstelegramm dennoch das Separatorzeichen <RS> übertragen, das das letzte Datenfeld eines Telegrammunterabschnitts immer begrenzt. Die übrigen Separatorzeichen <US>, die die Datenfelder innerhalb des Telegrammunterabschnitts begrenzen, werden, da niederwertiger als das Separatorzeichen <RS>, nicht übertragen.

Wird auf die Übertragung eines oder mehrerer Datenfelder am Ende eines Anwendungstelegramms aus den vorstehend genannten Gründen verzichtet, so ist eine Auflistung der Separatorzeichen der fehlenden Datenfelder nicht erforderlich. Das Telegrammendezeichen <FS> kennzeichnet alle noch zu erwartenden Telegrammteile als fehlend.

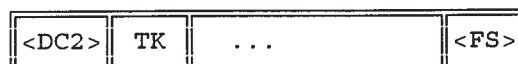
5.2 Anwendungstelegrammsyntax

In diesem Abschnitt werden die Grundstruktur und die wesentlichen Syntaxelemente der in dieser Norm definierten Anwendungstelegramme²⁵⁾ beschrieben. Die detaillierte Beschreibung der einzelnen Anwendungstelegramme erfolgt in den Abschnitten 6 bis 11.

5.2.1 Telegrammgrundstruktur

Die in dieser Norm definierten Anwendungstelegramme bestehen grundsätzlich aus einer Folge von 7-Bit-Zeichen nach DIN 66003:1974-06. In deutschsprachigen Anwendungen ist die Code-Tabelle 2 der DIN 66003 zu bevorzugen. Wird Code-Tabelle 1 benutzt, so ist das Währungszeichen (Spalte 2/Zeile 4) als Dollarzeichen (\$) zu interpretieren.

Wie in Bild 11 dargestellt, beginnen Anwendungstelegramme mit dem Gerätesteuerzeichen <DC2>²⁶⁾ und enden mit dem als Telegrammendezeichen verwendeten Hauptgruppen-Trennzeichen <FS>. Dem Gerätesteuerzeichen <DC2> folgt das ein Zeichen lange Datenfeld Telegrammtypkennung (TK) und je nach Telegrammtyp gegebenenfalls weitere Datenfelder.



Legende: TK Telegrammtypkennung;
... je nach Telegrammtyp weitere Datenfelder.

Bild 11: Anwendungstelegramm-Grundstruktur

5.2.2 Telegrammtypkennung

Die Telegrammtypkennung dient zur Unterscheidung der 15 Anwendungstelegramm-Grundtypen, die in dieser Norm definiert werden. Für das Datenfeld Telegrammtypkennung sind die Zeichen <0> bis <9> und <A> bis <E> zugelassen. Die Zuordnung der Telegrammtypkennung zum Anwendungstelegramm-Grundtyp erfolgt nach Tabelle 1.

²³⁾ siehe 5.2.1

²⁴⁾ Beispielsweise eine Namenliste, die keine Einträge (Namen) besitzt

²⁵⁾ MMS: Protocol Data Units

²⁶⁾ Device Control 2, DIN 66003 Platz 1/2

Tabelle 1: Kennungszeichnung des Anwendungstelegramm-Grundtyps

Anwendungstelegramm-Grundtyp		TK
Auftragsbearbeitung	-Anforderungstelegramm	<0>
Auftragsbearbeitung	-Antworttelegramm	<1>
Auftragsbearbeitung	-Fehlertelegramm	<2>
Ereignis	-Meldungstelegramm	<3>
Ablauffehler	-Meldungstelegramm	<4>
Dienstausführungsabbruch	-Anforderungstelegramm	<5>
Dienstausführungsabbruch	-Antworttelegramm	<6>
Dienstausführungsabbruch	-Fehlertelegramm	<7>
Verbindungsaufbau	-Anforderungstelegramm	<8>
Verbindungsaufbau	-Antworttelegramm	<9>
Verbindungsaufbau	-Fehlertelegramm	<A>
Verbindungsabbau	-Anforderungstelegramm	
Verbindungsabbau	-Antworttelegramm	<C>
Verbindungsabbau	-Fehlertelegramm	<D>
Verbindungsabbruch	-Meldungstelegramm	<E>

ANMERKUNG: Die Bezeichnung Auftragsbearbeitung steht in Tabelle 1 für eine Gruppe von 22 verschiedenen bestätigten Anwendungsdiensten. Die Bezeichnung Ereignis steht für eine Gruppe von 2 unbestätigten Anwendungsdiensten.

5.2.3 Auftragsbearbeitungstelegramme

5.2.3.1 Allgemeines

In diesem Abschnitt werden die Grundstrukturen der Auftragsbearbeitung-Anforderungstelegramme und der Auftragsbearbeitung-Antworttelegramme erläutert.

In den Anforderungs- und Antworttelegrammen der Anwendungsdienste der Gruppe "Auftragsbearbeitung" folgt nach der Telegrammtypkennung, wie in Bild 12 dargestellt, immer das ein Zeichen lange Datenfeld *Auftragsnummer* (Abk. AN) und das zwei Zeichen lange Datenfeld Dienstkennung, repräsentiert durch die Zeichen DK1 und DK2.

<DC2>	TK	AN	DK1	DK2	...	<FS>
-------	----	----	-----	-----	-----	------

Legende:

TK	Telegrammtypkennung {<0> oder <1>};
AN	Auftragsnummer (siehe 5.2.3.2);
DK1	Dienstkennungszeichen 1 (siehe 5.2.3.3);
DK2	Dienstkennungszeichen 2 (siehe 5.2.3.3);
...	je nach Telegrammtyp weitere Datenfelder.

Bild 12: Grundstruktur des Auftragsbearbeitung-Anforderungstelegramms und des Auftragsbearbeitung-Antworttelegramms

5.2.3.2 Auftragsnummer

Das Datenfeld *Auftragsnummer*²⁷⁾ dient zur Unterscheidung von maximal 63 gleichzeitig pro Kommunikationsrichtung einer Kommunikationsbeziehung offenen, d.h. unbeantworteten Dienstauführungsanforderungen²⁸⁾. Hierbei sind als offene Dienstauführungsanforderungen nur die Dienstauführungsanforderungen für die in 5.2.3.3 genannten Dienste zu zählen²⁹⁾.

ANMERKUNG: Die Auftragsnummer ermöglicht es beispielsweise einem Anwendungsprozeß, der zweimal hintereinander verschiedene Variablenwerte angefordert hat, ohne die

²⁷⁾ MMS: Invoke-Id

²⁸⁾ siehe 4.7.1

²⁹⁾ Auch die wiederholte Anforderung des gleichen Anwendungsdienstes (z.B. Status) ist jeweils als eigenständige Dienstauführungsanforderung zu zählen.

Antwort des Kommunikationspartners abzuwarten, beim Eintreffen der beiden Antworten festzustellen, auf welche "Variablen"-Anforderung sich die Antwort bezieht.

Es ist die Aufgabe des Implementierers eines Anwendungsprozesses, aufgrund der zu realisierenden Funktion festzulegen, ob gleichzeitig mehrere "offene" Dienstausführungsanforderungen in einer Kommunikationsbeziehung in der Rolle als Auftragnehmer unbeantwortet, d.h. in der "Schwebe" sein dürfen bzw. aus technischen Gründen sein können oder nicht.

Ist die zulässige Anzahl offener Dienstausführungsanforderungen eines Anwendungsprozesses in der Rolle als Auftragnehmer erreicht, so wird eine weitere Dienstausführungsanforderung bis zur Beantwortung einer der noch offenen Dienstausführungsanforderungen nach 10.1 mit dem Ablauffehler-Meldungstelegramm (Parameter: Telegrammtyp: TT 1; Ablauffehlercode: AFC 6) abgelehnt.

Die Auftragsnummer ist immer größer oder gleich 0 und kleiner 63. Zu ihrer Darstellung im Anwendungstelegramm wird das Zeichen verwendet, dessen dezimale Wertigkeit der Summe aus dem Wert der Auftragsnummer plus der Dezimalzahl 64 entspricht. Damit ergeben sich als zulässige Auftragsnummern die Zeichen <@> bis <DEL> nach DIN 66003:1974-04.

BEISPIEL: Die Auftragsnummer 12 wird im Anwendungstelegramm durch das Zeichen <L> dargestellt, das die dezimale Wertigkeit 76 (= 12 + 64) besitzt.

Ein Anwendungsprozeß darf für eine Dienstausführungsanforderung jede beliebige freie Auftragsnummer aus dem zugelassenen Wertebereich benutzen. Es ist zulässig, daß der Implementierer den Wertebereich der Auftragsnummern entsprechend der maximal zugelassenen Anzahl offener Dienstausführungsanforderungen einschränkt. Ist eine Auftragsnummer in einer Dienstausführungsanforderung benutzt worden, so ist sie für die erneute Verwendung so lange zu sperren, bis die Dienstausführungsanforderung entweder mit einem zugeordneten Antwort- oder Fehlertelegramm oder nach 10.1 mit einem Ablauffehler-Meldungstelegramm (das die betreffende Auftragsnummer enthält), beantwortet wurde oder bis gegebenenfalls eine manuelle Rücksetzung der Auftragsnummernverwaltung im Kommunikationsteilnehmer durch einen Bediener stattgefunden hat.

5.2.3.3 Dienstkennung

Das Datenfeld Dienstkennung wird zur Unterscheidung der verschiedenen Anwendungstelegramme benutzt, die der Anwendungsdienstklasse "Auftragsbearbeitung" zugeordnet sind. Als Dienstkennungszeichen zugelassen sind die Zeichen <0> bis <9> und <A> bis <F>. Die Zuordnung der Dienstkennung zum Anwendungsdienst erfolgt nach Tabelle 2³⁰⁾.

Tabelle 2: Zuordnung der Dienstkennungszeichen zu den Anwendungsdiensten der Gruppe "Auftragsbearbeitung"

Anwendungsdienst	DK1	DK2
Status	<0>	<0>
ÜbermittleNamenliste	<0>	<1>
Identifikation	<0>	<2>
VariableLesen	<0>	<4>
VariableSchreiben	<0>	<5>
DefiniereVariablenliste	<0>	
LöscheVariablenliste	<0>	<D>
BeginneLadesequenz	<1>	<A>
LadeSpeichersegment	<1>	
BeendeLadesequenz	<1>	<C>
BeginneRückladesequenz	<1>	<D>
RückLadeSpeichersegment	<1>	<E>
BeendeRückladesequenz	<1>	<F>
LöscheSpeicherbereich	<2>	<4>
ProgrammAktivieren	<2>	<6>
ProgrammDeaktivieren	<2>	<7>
ProgrammStarten	<2>	<8>
ProgrammStoppen	<2>	<9>
ProgrammFortsetzen	<2>	<A>
ProgrammRücksetzen	<2>	
ProgrammAbbrechen	<2>	<C>
AktionAuslösen	<3>	<4>

30) MMS-Bezug: Die Dienstkennungszeichen eines Anwendungsdienstes ergeben sich aus dem Sedezimalwert des TAGs (s. ISO 8824) des korrespondierenden MMS-Anwendungsdienstes.

5.2.4 Grundstruktur der Ereignis-Meldungstelegramme

In diesem Abschnitt wird die Grundstruktur der Meldungstelegramme der Anwendungsdienste der Gruppe "Ereignis" beschrieben.

In "Ereignis"-Meldungstelegrammen folgt hinter der Telegrammtypkennung <3> nach Bild 13 immer die ein Zeichen lange Dienstkennung, repräsentiert durch das Zeichen DK3.



Legende: DK3 Dienstkennungszeichen 3 (siehe Tabelle 3);
... weitere Datenfelder.

Bild 13: Grundstruktur des Ereignis-Meldungstelegramms

Die Dienstkennung wird zur Unterscheidung der Anwendungsdienste der Gruppe "Ereignis" benutzt. Als Dienstkennungszeichen zugelassen sind die Zeichen <0> bis <9>³¹⁾. Die Zuordnung der Dienstkennung zum Anwendungsdienst erfolgt nach Tabelle 3.

Tabelle 3: Zuordnung des Dienstkennungszeichens zu den Anwendungsdiensten der Gruppe "Ereignis"

Anwendungsdienst	DK3
UnangeforderterBericht	<0>
UnangeforderterStatus	<1>

5.3 Datentypen

5.3.1 Allgemeines

Zur Darstellung der in Verbindung mit den Anwendungsdiensten genutzten Daten läßt diese Norm folgende Datentypen zu:

- *Auftragsnummer*,
- *Sedezimalzahl*;
- *Bitfolge und Oktettfolge*,
- *Boolesche Zahl*,
- *endlicher Dezimalbruch*,
- *vorzeichenlose ganze Zahl*,
- *vorzeichenbehaftete ganze Zahl*,
- *BCD-Zahl (Binär Code für Dezimalziffer-Zahl)*,
- *Textzeichenfolge*,
- *Textzeichenfolgenliste*,
- *Name*,
- *Namenliste und*
- *Variablenwert*.

5.3.2 Auftragsnummer

Die Beschreibung des Datentyps *Auftragsnummer* und die Codierung der Auftragsnummern ist 5.2.3.2 zu entnehmen.

5.3.3 Sedezimalzahl

Der Datentyp *Sedezimalzahl* wird benutzt, um vorzeichenlose (siehe 5.3.7) und vorzeichenbehaftete ganze Zahlen (siehe 5.3.8) im Sechzehnersystem (Stellenwertsystem zur Basis 16) darzustellen. Für die Darstellung der Ziffern einer Sedezimalzahl in Telegrammen sind die 16 Zeichen <0>, <1>, <2>, <3>, <4>, <5>, <6>, <7>, <8>, <9>, <A>, , <C>, <D>, <E> und <F> zugelassen. Sedezimalzahlen werden aus einer Folge von Sedezimalzeichen gebildet. Die Übertragung von Sedezimalzahlen erfolgt beginnend mit der Ziffer mit der höchsten Wertigkeit.

5.3.4 Bitfolge und Oktettfolge

Als *Bitfolge*³²⁾ werden gruppenweise zusammengefaßte binäre Zustände (FALSCH/WAHR, JA/NEIN, AN/AUS) bezeichnet. Zu ihrer eindeutigen Identifikation werden die einzelnen Bits der Bitfolgen indi-

³¹⁾ Der Wertebereich berücksichtigt eine mögliche zukünftige Anwendungsdienstelement-Erweiterung

³²⁾ MMS: Bitstring

ziert. Die Indizierung der Bits einer Bitfolge erfolgt in dieser Norm von links nach rechts beginnend mit der Nummer 0. Das heißt, das erste Bit einer Bitfolge ist das Bit b_0 , das zweite Bit ist das Bit b_1 und so weiter. Das letzte Bit wird als Endebit bezeichnet.

Die höchste Länge einer Bitfolge ist anwendungsdienstabhängig. Sie wird dort, wo es erforderlich ist, in der jeweiligen Anwendungsdienstbeschreibung angegeben.

Die Bits einer Bitfolge werden zur Darstellung im Anwendungstelegramm beginnend mit dem Bit b_0 in Gruppen zu je vier Bits eingeteilt. Gemäß dem Dualsystem wird die Gesamtwertigkeit der Bitgruppe berechnet. Dabei hat innerhalb der Bitgruppe das Bit mit dem kleinsten Index den Wert 2^3 und das Bit mit dem größten Index den Wert 2^0 . Im Anwendungstelegramm werden die Bitgruppen durch das ihrem Sedezimalwert entsprechende Sedezimalzeichen codiert.

Eine Bitfolge mit einer nicht durch vier restfrei teilbaren Bitanzahl wird zur Übertragung vom lokalen Anwendungsprozeß nach dem Endebit mit einer entsprechenden Anzahl von Nullbits ergänzt. Da Bitfolgen grundsätzlich eine fest definierte Länge besitzen, ist der Empfänger in der Lage, die aufgefüllten Nullbits zu erkennen und sie aus der empfangenen Bitfolge wieder zu entfernen.

BEISPIEL: Soll beispielsweise die aus 15 Bits bestehende Bitfolge `101010011000110` übertragen werden, so muß sie, wie in Bild 14 dargestellt, an ihrem rechten Rand zur Übertragung mit einem zusätzlichen Nullbit aufgefüllt werden, da die Anzahl ihrer Bits nicht restfrei durch 4 teilbar ist. Dann sind beginnend vom linken Rand die Bits in Vierergruppen zusammengefaßt und die sedezimale Wertigkeit der Vierergruppen berechnet. Diese Wertigkeit ergibt sich im vorliegenden Beispiel zu A, 9, 8, C.

Bit-Nr.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Wert	1	0	1	0	1	0	0	1	1	0	0	0	1	1	0	0
	= A				= 9				= 8				= C			

Bild 14: Codierungsbeispiel

Die in Bild 14 dargestellte Bitfolge `101010011000110` wird, wie in Bild 15 gezeigt, als Zeichenfolge <A><9><8><C> übertragen, wobei die Übertragung mit dem Zeichen <A> beginnt.

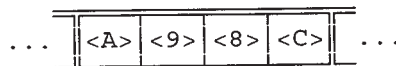


Bild 15: Codierung der Bitfolge `101010011000110`

Ein Oktett besteht aus 8 Bits und wird wie eine aus 8 Elementen bestehende Bitfolge behandelt. Ein Oktett wird, da es aus zwei Gruppen zu je vier Bit besteht, durch eine zweistellige Sedezimalzahl dargestellt.

Eine *Oktettfolge*³³⁾ ist eine spezielle Bitfolge, die aus einer restfrei durch 8 teilbaren Anzahl von Bits besteht. Das erste Oktett einer Oktettfolge steht links im Anwendungstelegramm und wird zuerst übertragen. Bei der Übertragung von Oktettfolgen mit den Anwendungsdiensten LadeSpeichersegment (siehe 7.2.3) und RückladeSpeichersegment (siehe 7.3.3) ist auch die Codierung der Oktettfolge in 6-Bit-Einheiten (Hextett) möglich. Jede 6-Bit-Einheit wird mit einem führenden Eins-Bit zu einem 7-Bit-Zeichen ergänzt. Ist die zu übertragende Bitfolge nicht restfrei durch 6 teilbar, so ist sie mit einer dazu erforderlichen Anzahl von Nullbits zu erweitern.

5.3.5 Boolesche Zahl

*Boolesche Zahlen*³⁴⁾ repräsentieren jeweils einen einzelnen binären Zustand (FALSCH/WAHR). Datenfelder vom Typ *Boolesche Zahl* werden im Anwendungstelegramm durch jeweils ein Zeichen dargestellt. Das zur Darstellung des Wertes einer Booleschen Zahl zu verwendende Zeichen ist Tabelle 4 zu entnehmen.

Tabelle 4: Codierungstabelle für die Zustände einer Booleschen Zahl

Boolesche Zahl	Zeichen
Bedingung/Aussage ist FALSCH	<0>
Bedingung/Aussage ist WAHR	<1>

33) MMS: Octetstring

34) MMS: Boolean

5.3.6 Endlicher Dezimalbruch

Der Datentyp *endlicher Dezimalbruch*³⁵⁾ wird verwendet, um vorzeichenbehaftete endliche Dezimalbrüche (rationale Zahlen) in einem Anwendungstelegramm darzustellen.

Für die Darstellung eines endlichen Dezimalbruchs sind zugelassen

- die Zeichen <0>, <1>, <2>, <3>, <4>, <5>, <6>, <7>, <8> und <9> als Ziffern,
- die Zeichen <+> und <-> als positives und negatives Vorzeichen,
- das Zeichen <.> als Dezimalpunkt (Radixtrenner) und
- das Zeichen <E> als Exponentenmarkierungszeichen.

Diese Norm verwendet zur Darstellung endlicher Dezimalbrüche die Darstellungsformen F1, F2 und F3 sowie die Normalform nach DIN 66250 (siehe ISO 6093), die sich aus dem in Bild 16 dargestellten Syntaxdiagramm ergeben.

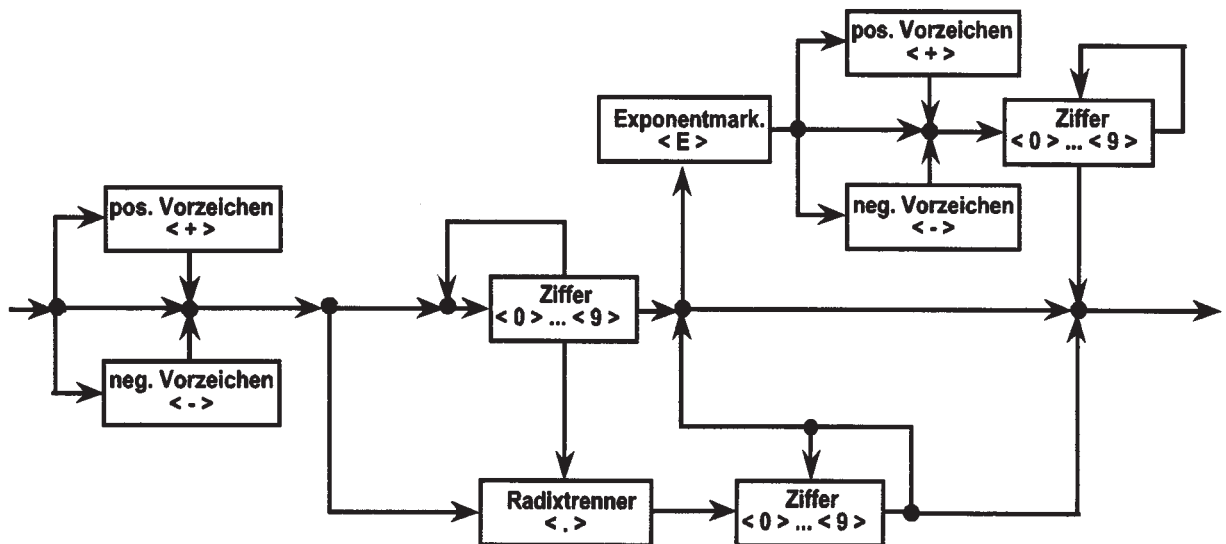


Bild 16: Syntaxdiagramm für endliche Dezimalbrüche

BEISPIEL: Erlaubte Darstellungen für Dezimalbrüche sind zum Beispiel -2300, 11, +4711, -17.46, 00.19, .23, 19.11, +378.253, -12.345E+12, 17E4, .12E03, 234E-3 und +12.2342E-04.

Führende Nullen in einem endlichen Dezimalbruch brauchen im Anwendungstelegramm nicht dargestellt zu werden. Das Vorzeichen oder die Ziffer mit der höchsten Wertigkeit (bei Zahlen ohne Vorzeichen) ist das erste Zeichen links im Datenfeld und wird zuerst übertragen.

ANMERKUNG: Die Anzahl der zur Darstellung eines endlichen Dezimalbruchs in einem Anwendungstelegramm verwendeten Zeichen wird in dieser Norm nicht begrenzt. Es wird jedoch darauf hingewiesen, daß aufgrund technischer Gegebenheiten die Anzahl der von einem Kommunikationsteilnehmer send- oder auswertbaren Vorkomma-, Nachkomma- und Exponentstellen begrenzt ist und bei den einzelnen Kommunikationsteilnehmern unterschiedlich sein kann.

Einem endlichen Dezimalbruch folgt wegen seiner variablen Länge immer ein Separatorzeichen, das sein Ende markiert. Der Typ des Separatorzeichens hängt von der Position des Datenfeldes im Anwendungstelegramm ab. Die Verwendungsregeln für den Gebrauch von Separatorzeichen in 5.1.3 und 5.3.12 sind zu beachten.

Das Bild 17 zeigt als Beispiel die Darstellung des endlichen Dezimalbruchs mit dem Wert -12.34E+12.



Legende: SZ Separatorzeichen (siehe 5.1.3 und 5.3.12)

Bild 17: Codierung des endlichen Dezimalbruchs -12.34E+12

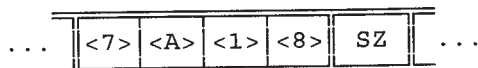
35) MMS: Real

5.3.7 Vorzeichenlose ganze Zahl

Der Datentyp *vorzeichenlose ganze Zahl*³⁶⁾ dient zur Repräsentation von natürlichen Zahlen³⁷⁾. Die Menge der natürlichen Zahlen umfaßt die Elemente {0; 1; 2; 3; 4; ...}³⁸⁾.

Vorzeichenlose ganze Zahlen werden im Anwendungstelegramm als Sedezimalzahlen dargestellt. Die Länge der Zahlen ergibt sich aus dem Zusammenhang des Anwendungstelegramms. Führende Nullen brauchen im Anwendungstelegramm nicht dargestellt zu werden.

Das Bild 18 zeigt als Beispiel die Darstellung der vorzeichenlosen ganzen Zahl mit dem dezimalen Wert 31256. Der dezimale Wert 31256 entspricht der Sedezimalzahl 7A18.



Legende: SZ Separatorzeichen (siehe 5.1.3 und 5.3.12)

Bild 18: Codierung der vorzeichenlosen ganzen Zahl 31256 (sedez. 7A18)

Datenfelder vom Typ *vorzeichenlose ganze Zahl*, die aufgrund des vorgesehenen Wertebereichs {0; ...; 15} in einem Anwendungstelegramm immer durch ein Zeichen dargestellt werden können, werden nicht durch ein Separatorzeichen begrenzt. Zur Kennzeichnung solcher Datenfelder wird in den Parameterbeschreibungen (Abschnitt 6 - 11) die Notation *vorzeichenlose ganze Zahl* (<16) benutzt.

5.3.8 Vorzeichenbehaftete ganze Zahl

5.3.8.1 Allgemeines

Der Datentyp *vorzeichenbehaftete ganze Zahl*³⁹⁾ repräsentiert die Menge der Zahlen { ... -2; -1; 0; 1; 2; ... }⁴⁰⁾.

Vorzeichenbehaftete ganze Zahlen werden im Anwendungstelegramm als Sedezimalzahlen dargestellt. Dabei ist bei der Codierung zu unterscheiden, ob die zu übertragende Zahl positiv oder negativ ist.

5.3.8.2 Vorzeichenbehaftete negative ganze Zahl

Vorzeichenbehaftete negative m-stellige ganze Zahlen werden durch ihr (16^m)-Komplement dargestellt. Führende Zeichen <F> einer negativen ganzen Zahl brauchen nicht dargestellt zu werden, wenn der verbleibende Rest mit einer der Ziffern 8 bis F beginnt.

Das Bild 19 zeigt als Beispiel die Darstellung der vorzeichenbehafteten negativen ganzen Zahl mit dem dezimalen Wert -31256. Der dezimale Wert 31256 entspricht dem Wert der Sedezimalzahl 7A18. Das (16^m)-Komplement zu dieser Sedezimalzahl ist die Sedezimalzahl 85E8.



Legende: SZ Separatorzeichen (siehe 5.1.3 und 5.3.12)

Bild 19: Codierung der vorzeichenbehafteten negativen ganzen Zahl -31256 (sedez. 85E8)

5.3.8.3 Vorzeichenbehaftete positive ganze Zahl

Vorzeichenbehaftete positive ganze Zahlen werden wie vorzeichenlose ganze Zahlen (siehe 5.3.7) codiert und übertragen. Ist die führende Ziffer aus dem Bereich 8 bis F, so muß der Zahl eine führende 0 vorangestellt werden.

Das Bild 20 zeigt als Beispiel die Darstellung der vorzeichenbehafteten positiven ganzen Zahl mit dem dezimalen Wert 45347. Der dezimale Wert 45347 entspricht dem Wert der Sedezimalzahl B123. Wird die ganze Zahl als Zeichenfolge <1> <2> <3> im Anwendungstelegramm dargestellt, so interpretiert der Telegrammempfänger das Datum fälschlich als die negative ganze Zahl -20189 (siehe 5.3.8.2). Daher muß zur ordnungsgemäßen Interpretation als vorzeichenbehaftete positive ganze Zahl

36) MMS: Unsigned

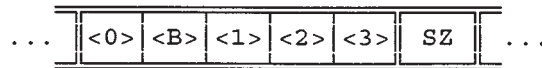
37) Siehe 5.3.9 "BCD-Zahlen"

38) Es wird nach MMS empfohlen, bei Anwendung dieser Norm als größte darstellbare vorzeichenlose ganze Zahl die Dezimalzahl 4294967295 (= 2³²-1) zu verwenden.

39) MMS: Integer

40) Es wird nach MMS empfohlen, bei Anwendung dieser Norm als größte zulässige darstellbare vorzeichenbehaftete ganze Zahl die Dezimalzahl 2147483647 (= 2³¹-1) und als kleinste die Dezimalzahl -2147483648 (= -2³¹) zu verwenden.

das Datum mit dem Zeichen <0> beginnen.



Legende: SZ Separatorzeichen (siehe 5.1.3 und 5.3.12)

Bild 20: Codierung der vorzeichenbehafteten positiven ganzen Zahl +45347 (sedez. 0B123)

5.3.9 BCD-Zahl

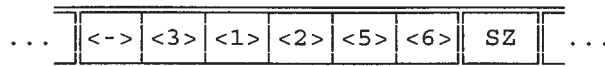
Der Datentyp *BCD-Zahl* (Binär Code für Dezimalziffer-Zahl) dient zur Repräsentation von ganzen Zahlen, d.h. von Zahlen aus der Menge {...; -2; -1; 0; 1; 2; ...}⁴¹⁾.

Für die Darstellung von BCD-Zahlen sind zugelassen

- die Zeichen <0>, <1>, <2>, <3>, <4>, <5>, <6>, <7>, <8> und <9> als Ziffern,
- die Zeichen <+> und <-> als Vorzeichen.

Negativen BCD-Zahlen muß das Vorzeichen <-> vorangestellt werden. Positiven Zahlen kann das Vorzeichen <+> vorangestellt werden. Eine vorzeichenlose BCD-Zahl wird als positive Zahl interpretiert.

Das Bild 21 zeigt als Beispiel die Darstellung der Zahl -31256 als BCD-Zahl.



Legende: SZ Separatorzeichen (siehe 5.1.3 und 5.3.12)

Bild 21: Codierung der Zahl -31256 als BCD-Zahl

5.3.10 Textzeichenfolge und Name

Ein Datenfeld vom Typ *Textzeichenfolge*⁴²⁾ besteht aus einer Folge von Zeichen. Als Textzeichen zugelassen sind:

- die alphanumerischen Zeichen und die Schriftzeichen nach DIN 66003:1974-06,
- die Formatsteuerzeichen <BS>, <HT>, <LF>, <VT>, <FF> und <CR> sowie
- die Zeichen <BEL>, <ESC>, <SI>, <SO> und <SYN>.

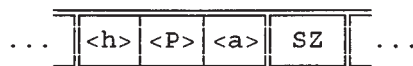
Ein *Name*⁴³⁾ ist eine spezielle Textzeichenfolge mit einem eingeschränkten Zeichensatz und einer auf maximal 32 Zeichen beschränkten Länge. Ein Name darf nicht mit einer Ziffer beginnen. Die in Namen erlaubten Zeichen sind

- die Großbuchstaben <A> bis <Z>,
- die Kleinbuchstaben <a> bis <z>,
- die Ziffern <0> bis <9>,
- das Dollarzeichen <\$> sowie
- die Unterstreichung <_>.

Textzeichenfolgen und Namen folgt wegen ihrer variablen Länge immer ein Separatorzeichen, das ihr Ende markiert. Der Typ des Separatorzeichens hängt von der Position des Datenfeldes im Anwendungstelegramm ab. Die Verwendungsregeln für den Gebrauch von Separatorzeichen in 5.1.3 sind zu beachten.

Das erste Zeichen einer Textzeichenfolge oder eines Namens ist das erste Zeichen links im Datenfeld und wird zuerst übertragen.

Das Bild 22 zeigt als Beispiel die Darstellung der Textzeichenfolge "hPa" in einem Anwendungstelegramm.



Legende: SZ Separatorzeichen (siehe 5.1.3 und 5.3.12)

Bild 22: Codierung der Textzeichenfolge "hPa"

41) Es wird in dieser Norm keine Empfehlung für die größte darstellbare BCD-Zahl ausgesprochen.

42) MMS: Visible String und ergänzende Sonderzeichen

43) MMS: Identifier

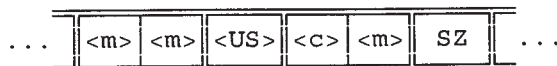
5.3.11 Textzeichenfolgenliste und Namenliste

Eine *Textzeichenfolgenliste* enthält eine oder mehrere Textzeichenfolgen. Eine *Namenliste* ist eine spezielle Textzeichenfolgenliste, die einen oder mehrere Namen enthält.

Textzeichenfolgen oder Namen in einer Textzeichenfolgenliste bzw. einer Namenliste, die nicht das letzte Element der Liste sind, folgt wegen ihrer variablen Länge immer das Separatorzeichen <US>, das ihr Ende markiert. Der Typ des Separatorzeichens des letzten Elements einer Liste hängt von der Position der Liste im Anwendungstelegramm und von der Verschachtelungstiefe ab. Die Verwendungsregeln für den Gebrauch von Separatorzeichen in Abs. 5.1.3 sind zu beachten.

Ist die Reihenfolge der Textzeichenfolgen oder Namen in der Textzeichenfolgenliste bzw. Namenliste signifikant, so hat die am linken Rand des Datenfeldes stehende Textzeichenfolge bzw. der am linken Rand des Datenfeldes stehende Name die höchste Wertigkeit.

Das Bild 23 zeigt als Beispiel die Codierung der aus den beiden Textzeichenfolgen "mm" und "cm" bestehende Textzeichenfolgenliste in einem Anwendungstelegramm.



Legende: SZ Separatorzeichen (siehe 5.1.3 und 5.3.12)

Bild 23: Codierung der Textzeichenfolgenliste ("mm", "cm")

5.3.12 Variablenwert

In Zusammenhang mit den Variable-bezogenen Anwendungsdiensten, die in Abschnitt 8 beschrieben werden, wird der Datentyp *Variablenwert* benutzt, mit dem in den Dienstprimitiven und Anwendungstelegrammen der zu übermittelnde "Inhalt" von Variablen dargestellt wird. Da der Datentyp *Variablenwert* jeweils von der inneren Struktur der Variable abhängt, auf die er sich bezieht, ist er anwendungsabhängig und kann sich bei allen Variablen, die keine gleiche Struktur besitzen, unterscheiden.

Existiert in zwei Kommunikationspartnern, die mit den Variable-bezogenen Anwendungsdiensten Informationen austauschen, jeweils eine Variable mit dem gleichen Namen aber mit unterschiedlicher Struktur, so kann es bei Nutzung der Variable-bezogenen Anwendungsdienste zu Datenverlusten kommen, die durch Überprüfung der "gelieferten Daten" verhindert werden müssen.

Um die Ermittlung der Variablenstruktur aus dem Anwendungstelegramm zu ermöglichen, sind Telegrammabschnitte vom Datentyp *Variablenwert* rekursiv aufgebaut. Das heißt, der einer bestimmten Variable zuzuordnende Datentyp *Variablenwert* wird nach fest definierten Regeln umkehrbar eindeutig aus den Grunddatentypen

- *BCD-Zahl* nach 5.3.9,
- *Bitfolge* nach 5.3.4,
- *Boolesche Zahl* nach 5.3.5,
- *endlicher Dezimalbruch* nach 5.3.6,
- *Oktettfolge* nach 5.3.4,
- *vorzeichenbehaftete ganze Zahl* nach 5.3.8,
- *vorzeichenlose ganze Zahl* nach 5.3.7 und
- *Textzeichenfolge* nach 5.3.10

abgeleitet. Datenfelder in einem Telegrammabschnitt vom Typ *Variablenwert*, die einem der vorstehend aufgeführten Grunddatentypen zuzuordnen sind, werden in diesem Abschnitt als Grunddatenelemente bzw. dort, wo der Hinweis auf den Grunddatentyp nicht erforderlich ist, als Datenelement bezeichnet.

Mehrere logisch zusammengehörende Grunddatenelemente können zu sogenannten Datenelementfolgen zusammengefaßt werden. Mehrere so gebildete Datenelementfolgen können ihrerseits gemeinsam mit Grunddatenelementen oder ohne diese die Datenelemente einer übergeordneten Datenelementfolge sein. Eine Datenelementfolge kann somit maximal zwei untergeordnete Schachtelungsebenen besitzen, von denen jeweils die unterste ausschließlich aus Grunddatenelementen gebildet wird.

In Abhängigkeit davon, ob die Datenelemente in einer Datenelementfolge alle den gleichen Typ besitzen oder ob sich der Typ von mindestens zwei Elementen unterscheidet, wird die Folge als "homogene" oder als "heterogene" Datenelementfolge bezeichnet.

Die Struktur der drei möglichen Grundvarianten eines Telegrammabschnitts vom Typ *Variablenwert* ist in Bild 24 dargestellt. Wie das Bild zeigt, besteht der Telegrammabschnitt neben einer verbindlich vorgeschriebenen einleitenden Datentypkennung

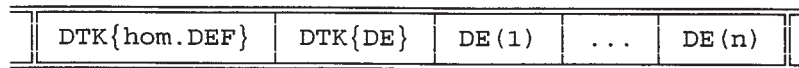
- aus einem folgenden Grunddatenelement oder
- bei homogenen Datenelementfolgen aus einer weiteren Datentypkennung, die den Typ der folgenden gleichartigen Datenelemente spezifiziert oder

- bei heterogenen Datenelementfolgen aus mehreren Datenelementen, denen jeweils eine Datentypkennung vorsteht, die den Typ des folgenden Datenelements spezifiziert.

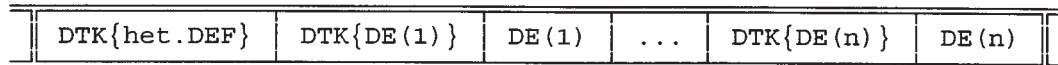
Telegrammabschnitt bestehend aus einem Grunddatenelement:



Telegrammabschnitt bestehend aus homogener Datenelementfolge:



Telegrammabschnitt bestehend aus heterogener Datenelementfolge:



Legende:	DE(1)	1. Datenelement
	DE(n)	n-tes Datenelement
	DEF	Datenelementfolge
	DTK	Datentypkennung
	DTK{ }	Datentypkennung für
	GDE	Grunddatenelement

Bild 24: Grundstruktur eines Telegrammabschnitts vom Datentyp *Variablenwert*

Die in dieser Norm vorgesehene Datentypkennung ist vom Typ *vorzeichenlose ganze Zahl* (<16) und kann folgende Werte annehmen:

- 1 homogene Datenelementfolge,
- 2 heterogene Datenelementfolge,
- 3 Boolesche Zahl,
- 4 Bitfolge,
- 5 vorzeichenbehaftete ganze Zahl,
- 6 vorzeichenlose ganze Zahl,
- 7 gesperrt für zukünftige Erweiterungen dieser Norm,
- 8 endlicher Dezimalbruch,
- 9 Oktettfolge,
- 10 Textzeichenfolge,
- 11 gesperrt für zukünftige Erweiterungen dieser Norm,
- 12 gesperrt für zukünftige Erweiterungen dieser Norm und
- 13 BCD-Zahlen.

BEISPIEL 1: Eine Variable, die einen Zählerstand repräsentiert, der als "vorzeichenlose ganze Zahl" realisiert ist, wird bei der Übertragung in einem Anwendungstelegramm durch das Grunddatenelement "vorzeichenlose ganze Zahl" dargestellt. Eine Variable, die einen Längenmeßwert repräsentiert, der aus dem Zahlenwert (endlicher Dezimalbruch) und der Einheit (Textzeichenfolge) besteht, kann bei der Übertragung als "heterogene Datenelementfolge" dargestellt werden, die aus dem Grunddatenelement "endlicher Dezimalbruch" und dem Grunddatenelement "Textzeichenfolge" besteht.

Das Flußdiagramm zur Codierung eines Telegrammabschnitts vom Typ *Variablenwert* ist in Bild 25 dargestellt. Es ist zu beachten, daß die Codierung der Grunddatenelemente nach 5.3.4 bis 5.3.10 erfolgt. Außerdem ist zu beachten, daß die in Bild 25 mit "*" markierten Separatorzeichen dann, wenn der Telegrammabschnitt am Ende des Anwendungstelegramms steht, durch das Zeichen <FS> zu ersetzen ist, das das Telegrammende markiert.

Wie das Bild 25 zeigt, sieht diese Norm folgende Varianten für den Aufbau des Telegrammabschnitts vor:

- Der Wert der Variable besteht aus einem einzigen Grunddatenelement (dargestellt im Zweig A).
- Der Wert der Variable setzt sich aus mehreren Grunddatenelementen gleichen Typs zusammen (dargestellt im Zweig B1).
- Der Wert der Variable setzt sich aus mehreren homogenen Datenelementfolgen zusammen (dargestellt im Zweig B2).
- Der Wert der Variable setzt sich aus mehreren heterogenen Datenelementfolgen zusammen (dargestellt im Zweig B3).
- Der Wert der Variable setzt sich aus mehreren Datenelementen zusammen. Bei mindestens zwei dieser Datenelemente unterscheidet sich der Typ. Das Datenelement ist ein Grunddatenelement (dargestellt in Zweig C1). Das Datenelement ist eine homogene Datenelementfolge (dargestellt in Zweig C2). Das Datenelement ist eine heterogene Datenelementfolge (dargestellt in Zweig C3).



Bild 25: Flußdiagramm zur Codierung eines Telegrammabschnitts vom Typ Variablenwert

ANMERKUNG: Von den in Bild 25 dargestellten Varianten muß ein Kommunikationsteilnehmer nur die unterstützen, die für die Codierung/Decodierung der Werte seiner Variablen nötig sind. Kommunikationsteilnehmer mit Variablen, deren Werte beispielsweise jeweils auf einem Grunddatentyp basieren, müssen somit nur die Variablenwertcodierung/-decodierung gemäß Zweig A vorsehen.

BEISPIEL 2: In Bild 26 ist die Codierung des Telegrammabschnitts Wert dargestellt, der aus einem einzigen Grunddatenelement (vorzeichenbehaftete ganze Zahl: "10") besteht (codiert nach Zweig A). Eine vorzeichenbehaftete ganze Zahl besitzt die Datentypkennung DTK = 5, die im Telegramm durch das Zeichen <5> repräsentiert wird. Der Wert der ganzen Zahl "10" wird nach 5.3.8 durch die Zeichen <0><A> codiert.

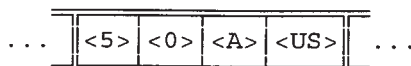


Bild 26: Codierung eines Variablenwertes bestehend aus einem Grunddatenelement

BEISPIEL 3: In Bild 27 ist die Codierung des Telegrammabschnitts "Wert" dargestellt, der sich aus zwei Grunddatenelementen unterschiedlichen Typs zusammensetzt, d.h. der eine heterogene Datenelementfolge darstellt (codiert nach Zweig C1). Das erste Grunddatenelement ist vom Typ *vorzeichenbehaftete ganze Zahl* (Wert: "100", codiert als <6><4>). Das zweite Grunddatenelement ist vom Typ *Textzeichenfolge* (Inhalt: "mm", codiert als <m><m>). Da sich der Telegrammabschnitt aus zwei Datenelementen unterschiedlichen Typs zusammensetzt, beginnt er mit dem Zeichen <2>, das die Datentypkennung (DTK = 2) für eine heterogene Datenelementfolge repräsentiert. Dann folgt mit dem Zeichen <5> die Datentypkennung für eine vorzeichenbehaftete ganze Zahl. Die vorzeichenbehaftete ganze Zahl wird durch das Separatorzeichen <US> begrenzt, dem in Form des Zeichens <A> die Datentypkennung für ein Grunddatenelement vom Typ *Textzeichenfolge* folgt. Da die Textzeichenfolge letztes Datenelement im Telegrammabschnitt ist, dieser Telegrammabschnitt aber nicht am Ende des Telegramms steht, wird sie durch das Separatorzeichen <RS> abgeschlossen.

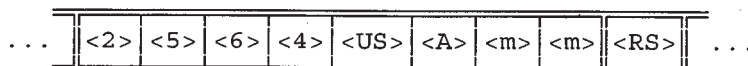


Bild 27: Codierung einer aus zwei ungleichartigen Grunddatenelementen bestehenden heterogenen Datenelementfolge

ANMERKUNG: Weitere Beispiele sind in C.3.6 dargestellt.

6 Anwendungsdienste zur Verbindungsverwaltung

Die drei Anwendungsdienste Verbindungsaufbau, Verbindungsabbau und Verbindungsabbruch bilden die Dienstgruppe Verbindungsverwaltung. Das Zusammenwirken der drei Anwendungsdienste der Dienstgruppe während des Bestehens einer Kommunikationsbeziehung, ist in 6.4 dargestellt.

6.1 Anwendungsdienst Verbindungsaufbau

6.1.1 Allgemeines

Der bestätigte Anwendungsdienst Verbindungsaufbau⁴⁴⁾ wird von einem Anwendungsprozeß benutzt, um auf der OSI-Anwendungsebene eine Kommunikationsbeziehung zu einem Kommunikationspartner aufzubauen. Unter Aufbau der Kommunikationsbeziehung ist einerseits die wechselseitige Vereinbarung bestimmter Parameter und andererseits die Freigabe des Informationsaustauschs zu verstehen. Das Vorhandensein einer Kommunikationsbeziehung ist zwingende Voraussetzung dafür, daß die übrigen in dieser Norm definierten bestätigten und unbestätigten Anwendungsdienste benutzt werden können.

Der Anwendungsprozeß, der den Verbindungsaufbau veranlaßt, wird in dieser Norm als der "rufende" und der vorgesehene Kommunikationspartner als der "gerufene" Anwendungsprozeß bezeichnet. Die Bezeichnung "rufend" und "gerufen" bezieht sich dabei ausschließlich auf den Anwendungsdienst Verbindungsaufbau.

Das Aufbauen einer durch eine reservierte Verbindungsnummer festgelegten Kommunikationsbeziehung (siehe 4.3) darf prinzipiell gleichberechtigt von jedem der beiden Kommunikationspartner eingeleitet werden, sofern er den Anwendungsdienst in der Rolle als Auftraggeber unterstützt.

⁴⁴⁾ Der Anwendungsdienst Verbindungsaufbau umfaßt verschiedene Parameter des MMS-Dienstes Initiate und des zugeordneten ACSE-Dienstes A-Associate nach ISO 8649 und ISO 8650.

Ein Kommunikationsteilnehmer, der diese Norm unterstützt, muß den Anwendungsdienst Verbindungsaufbau zwingend in der Rolle als gerufener Kommunikationspartner unterstützen (Pflichtdienst). Ob ein Kommunikationsteilnehmer darüber hinaus auch den Anwendungsdienst Verbindungsaufbau als rufender Kommunikationspartner unterstützt, ist vom Implementierer festzulegen.

ANMERKUNG: Ein Anwendungsprozeß, der im Rahmen seiner Informationsverarbeitungsaufgaben die Dienste eines oder mehrerer Kommunikationspartner nutzt, sollte zweckmäßigerweise über die Möglichkeit verfügen, als rufender Kommunikationsteilnehmer die Verbindungen zu diesen Kommunikationspartnern aufzubauen.

6.1.2 Dienstprimitiv-Parameter

Den Dienstprimitiven des Anwendungsdienstes Verbindungsaufbau sind die in Tabelle 5 angegebenen Parameter zugeordnet.

Tabelle 5: Dienstprimitiv-Parameter des Anwendungsdienstes Verbindungsaufbau

Parametername	Verbindungsaufbau.			
	Anf	Anz	Ant /	Bes
Argument	E	E	-	
Systemweiter-Name-des-gerufenen-Anwendungsprozesses	E	E	-	
Lokaler-Name-des-rufenden-Anwendungsprozesses	E	-	-	
Systemweiter-Name-des-rufenden-Anwendungsprozesses	-	E	-	
Vorgeschlagene-Anzahl-offene-Dienste-rufend	E	E	-	
Vorgeschlagene-Anzahl-offene-Dienste-gerufen	E	E	-	
Vorgeschlagene-Version	O	O	-	
Ergebnis(+)	-	-	A	
Festgelegte-Anzahl-offene-Dienste-rufend	-	-	E	
Festgelegte-Anzahl-offene-Dienste-gerufen	-	-	E	
Festgelegte-Version	-	-	E	
Ergebnis(-)	-	-	A	
Fehlerklasse	-	-	E	
Fehlercode	-	-	E	
Zusätzlicher-Code	-	-	O	
Zusätzliche-Beschreibung	-	-	O	

6.1.3 Argument

Die Parameter der Parametergruppe Argument haben folgende Bedeutung:

Systemweiter-Name-des-gerufenen-Anwendungsprozesses (AG): Dieser Parameter vom Typ *Name* legt den Anwendungsprozeß mit seinem systemweit gültigen Namen fest, zu dem die Kommunikationsbeziehung aufgebaut werden soll.

Lokaler-Name-des-rufenden-Anwendungsprozesses (LAR): Dieser Parameter vom Typ *Name* legt den Anwendungsprozeß mit seinem lokal gültigen Namen fest, der den Verbindungsaufbau wünscht.

Systemweiter-Name-des-rufenden-Anwendungsprozesses (AR): Dieser Parameter vom Typ *Name* legt den Anwendungsprozeß mit seinem systemweit gültigen Namen fest, der den Verbindungsaufbau wünscht.

Die Länge der Namen von Anwendungsprozessen werden in dieser Norm auf maximal 16 Zeichen begrenzt.

Vorgeschlagene-Anzahl-offene-Dienste-rufend (ODRV): Mit diesem Parameter vom Typ *Auftragsnummer* übermittelt der rufende Anwendungsprozeß die von ihm in der Kommunikationsbeziehung als Auftraggeber unterstützbare maximal Anzahl gleichzeitig offener Dienstausführungsanforderungen für die in Tabelle 2 angegebenen Anwendungsdienste. Er schlägt damit dem gerufenen Anwendungsprozeß die maximale Anzahl gleichzeitig offener bestätigter Anwendungsdienste vor, die dieser als Auftragnehmer unterstützen soll. Die vorgeschlagene Anzahl muß entsprechend der technischen Möglich-

keiten zwischen 1⁴⁵⁾ und 63 liegen.

Vorgeschlagene-Anzahl-offene-Dienste-gerufen (ODGV): Mit diesem Parameter vom Typ *Auftragsnummer* übermittelt der **rufende Anwendungsprozeß** die von ihm in der Kommunikationsbeziehung als **Auftragnehmer** unterstützbare maximale Anzahl gleichzeitig offener Dienstauführungsanforderungen für die in Tabelle 2 angegebenen Anwendungsdienste. Er schlägt damit dem gerufenen Anwendungsprozeß die maximale Anzahl gleichzeitig offener bestätigter Anwendungsdienste vor, die dieser als Auftraggeber unterstützen soll. Die vorgeschlagene Anzahl muß entsprechend der technischen Möglichkeiten zwischen 1 und 63 liegen. Wegen der zwingend vorgeschriebenen Unterstützung des Pflichtdienstes Identifikation (siehe 10.3), der zu den in Tabelle 2 angegebenen Anwendungsdiensten gehört, muß ein gerufener Anwendungsprozeß in der Rolle als Auftragnehmer mindestens einen offenen bestätigten Anwendungsdienst unterstützen. Daher gilt: ODGV > 0.

Vorgeschlagene-Version (VV): Mit diesem optionalen Parameter vom Typ *Textzeichenfolge* teilt der rufende Anwendungsprozeß die von ihm unterstützte Protokollversion mit, die er dem gerufenen Anwendungsprozeß als gemeinsame Protokollversion für die Kommunikationsbeziehung vorschlägt. Das in dieser Norm beschriebene Protokoll wird durch die Textzeichenfolge "66348-3/V100" identifiziert.

6.1.4 Anforderungstelegramm

Die Struktur des Verbindungsaufbau-Anforderungstelegramms ist in Bild 28 dargestellt. Die Beschreibung der Datenfelder und deren Typen sind Abs. 6.1.3 zu entnehmen⁴⁶⁾.



Legende:

AG	Systemweiter-Name-des-gerufenen-Anwendungsprozesses;
AR	Systemweiter-Name-des-rufenden-Anwendungsprozesses;
ODRV	Vorgeschlagene-Anzahl-offene-Dienste-rufend;
ODGV	Vorgeschlagene-Anzahl-offene-Dienste-gerufen;
VV	Vorgeschlagene-Version.

Bild 28: Struktur des Verbindungsaufbau-Anforderungstelegramms

Die Datenfelder ODRV und ODGV werden im Anwendungstelegramm durch je ein Zeichen dargestellt. Die Datenfelder AG, AR und VV haben eine variable Länge.

Aufgabe der Protokollmaschine der Anwendungsschicht ist es, die vom Verbindungsaufbau-Anforderungstelegramm benötigten Parameter "systemweiter Name des gerufenen Anwendungsprozesses (AG)" und "systemweiter Name des rufenden Anwendungsprozesses (AR)" bereitzustellen. Dies betrifft auch die jedem Anwendungstelegramm voranzustellende Verbindungsnummer. Ist der Protokollmaschine der Anwendungsschicht im Kommunikationsteilnehmer mit dem rufenden Anwendungsprozeß der dem lokalen Anwendungsprozeßnamen im Informationsverbundsystem zugeordnete, systemweit gültige Anwendungsprozeßname nicht bekannt, so muß dieser mit geeigneten Diensten der Schnittstelle zu den transportorientierten Schichten (STS) erfragt werden (vergl. Dienst STS-Verbindungsnummer, Abschnitt 12.4).

ANMERKUNG 1: Die Codierung der einzelnen Datenfelder der Anwendungstelegramme wird in dieser Norm grundsätzlich nicht dargestellt. Diese Codierung wird jeweils durch die Angabe des Datentyps in der Parameterbeschreibung festgelegt und ergibt sich aus den Beschreibungen in 5.3.

ANMERKUNG 2: Zu jedem in dieser Norm definierten Anwendungstelegramm ist in Anhang C die Codierung eines Beispiel-Telegramms dargestellt.

6.1.5 Ergebnis(+)

Die Parameter der Parametergruppe Ergebnis(+) haben folgende Bedeutung:

Festgelegte-Anzahl-offene-Dienste-rufend (ODRF): Mit diesem Parameter vom Typ *Auftragsnummer* übermittelt der gerufene Anwendungsprozeß die von ihm für die Kommunikationsbeziehung festgelegte maximale Anzahl offener bestätigter Anwendungsdienste nach Tabelle 2, die er in der Rolle als Auftrag-

⁴⁵⁾ Wegen der zwingend vorgeschriebenen Unterstützung des Anwendungsdienstes Identifikation (siehe 10.3), der zu den in Tabelle 2 angegebenen Anwendungsdiensten gehört, muß ein Anwendungsprozeß in seiner Rolle als Auftraggeber wie in der Rolle als Auftragnehmer mindestens einen offenen bestätigten Anwendungsdienst unterstützen. Daher gilt ODRV und ODGV > 0.

⁴⁶⁾ In der Regel korrespondieren die Dienstprimitiv-Parameter mit gleichbedeutenden Datenfeldern in den Anwendungstelegrammen. Da beide die gleiche Bedeutung besitzen, wird in allen Anwendungsdienstbeschreibungen grundsätzlich auf die Erläuterung der Datenfelder in den Anwendungstelegrammen verzichtet und jeweils auf die entsprechende Beschreibung der Dienstprimitiv-Parameter verwiesen.

nehmer unterstützt. Diese Anzahl ist kleiner oder gleich der Anzahl, die vom rufenden Anwendungsprozeß im Parameter/Datenfeld ODRV vorgeschlagen wurde.

Festgelegte-Anzahl-offene-Dienste-gerufen (ODGF): Mit diesem Parameter vom Typ *Auftragsnummer* übermittelt der gerufene Anwendungsprozeß die von ihm für die Kommunikationsbeziehung festgelegte maximale Anzahl offener bestätigter Anwendungsdienste nach Tabelle 2, die er in der Rolle als Auftraggeber unterstützt. Diese Anzahl ist kleiner oder gleich der Anzahl, die vom rufenden Anwendungsprozeß im Parameter/Datenfeld ODRV vorgeschlagen wurde.

Festgelegte-Version (VF): Mit diesem Parameter vom Typ *Textzeichenfolge* teilt der gerufene Anwendungsprozeß die von ihm unterstützte Protokollversion mit, die damit zu der in der Kommunikationsbeziehung festgelegten Protokollversion wird.

6.1.6 Antworttelegramm

Die Struktur des Verbindungsaufbau-Antworttelegramms ist in Bild 29 dargestellt. Die Beschreibung der Datenfelder und deren Typen sind 6.1.5 zu entnehmen.



Legende: ODRF Festgelegte-Anzahl-offene-Dienste-rufend;
ODGF Festgelegte-Anzahl-offene-Dienste-gerufen.
VF Festgelegte-Version

Bild 29: Struktur des Verbindungsaufbau-Antworttelegramms

Die Datenfelder ODRF und ODGF werden im Anwendungstelegramm durch je ein Zeichen dargestellt. Das Datenfeld VF hat eine variable Länge.

6.1.7 Ergebnis(-)

Die Beschreibung der Parameter der Parametergruppe Ergebnis(-) ist 11.1 zu entnehmen.

6.1.8 Fehlertelegramm

Das Verbindungsaufbau-Fehlertelegramm entspricht dem Fehlertelegramm Typ II (siehe 11.2).

6.1.9 Dienstprozedur

Ein (rufender) Anwendungsprozeß, der als Auftraggeber eine bestimmte Kommunikationsbeziehung mit dem Anwendungsdienst Verbindungsaufbau aktivieren möchte, muß zwingend hierzu

- die dieser Verbindung zugeordnete Verbindungsnummer,
- den systemweit gültigen Anwendungsprozeßnamen des Kommunikationspartners und
- seinen eigenen (lokalen) Namen

kennen. Die Protokollmaschine der Anwendungsschicht des rufenden Kommunikationsteilnehmers muß darüberhinaus den systemweit gültigen Namen des rufenden Anwendungsprozesses kennen.

Ist der Protokollmaschine der Anwendungsschicht im rufenden Kommunikationsteilnehmer der systemweit gültige Name des rufenden Anwendungsprozesses nicht bekannt, so muß sie sich diesen mit geeigneten Diensten der Schnittstelle zu den transportorientierten Schichten (STS) erfragen. Dies betrifft auch die jedem Anwendungstelegramm voranzustellende Verbindungsnummer (siehe 12.4).

Ein Kommunikationsteilnehmer, der von einem Kommunikationspartner zum Aufbau einer Kommunikationsbeziehung aufgefordert ist, muß für den bzw. die in ihm existierenden Anwendungsprozesse die Zuordnung lokaler zu systemweiten Anwendungsprozeßnamen, die ihm im Informationsverbundsystem zugeteilt wurden, kennen.

Ist der Protokollmaschine der Anwendungsschicht im gerufenen Kommunikationsteilnehmer diese Zuordnung nicht bekannt, so muß sie sich diese mit geeigneten Diensten der Schnittstelle zu den transportorientierten Schichten (STS) erfragen (siehe 12.4). Es wird empfohlen, dies in der Initialisierungsphase eines Informationsverbundsystems durchzuführen.

Darüber hinaus bleibt es dem Implementierer überlassen, ob er

- Schutzmechanismen vorsieht, die es nur Anwendungsprozessen mit "berechtigten" Namen ermöglichen, den Verbindungsaufbau erfolgreich anzufordern und
- bestimmte Zugriffsrechte auf Objekte (z.B. Variable, siehe Abschnitt 8) vom Namen des rufenden Anwendungsprozesses abhängig macht.

Der Kommunikationsteilnehmer, der zum Verbindungsaufbau aufgefordert wird, prüft, ob in ihm der gerufene Anwendungsprozeß existiert und ob eventuell vorhandene Sicherungsmechanismen eine Kommunikationsbeziehung mit dem rufenden Anwendungsprozeß erlauben.

Ist der gerufene Kommunikationsteilnehmer/Anwendungsprozeß gewillt, den eingetroffenen Verbindungsaufbauwunsch zu akzeptieren, so speichert er den Namen des rufenden Anwendungsprozesses und die Verbindungsnummer, mit der die Verbindungsaufbau-Anforderung eingetroffen ist. Diese Verbindungsnummer wird zur Identifikation aller in der Kommunikationsbeziehung zu sendenden bzw. empfangenen Telegramme⁴⁷⁾ benutzt. Dann ruft er das Verbindungsaufbau.Antwort[Ergebnis(+)]-Dienstprimitiv mit den Werten auf, die von ihm unterstützt werden können.

Kann der gerufene Anwendungsprozeß den Verbindungsaufbauwunsch nicht akzeptieren, so benutzt er das Verbindungsaufbau.Antwort[Ergebnis(-)]-Dienstprimitiv.

Wird der Verbindungsaufbau zu einem Anwendungsprozeß gewünscht, der im gerufenen Kommunikationsteilnehmer namentlich unbekannt ist, so antwortet das Kommunikationssystem des gerufenen Kommunikationsteilnehmers mit einem Verbindungsaufbau-Fehlertelegamm (siehe 11.2). In diesem Fehlertelegamm sind die Datenfelder Fehlerklasse: FK = 1 (Anwendungsreferenz) und Fehlercode: FC = 3 (Anwendungsprozeßname unbekannt) anzugeben.

Zum Aufbau einer Kommunikationsbeziehung auf der OSI-Anwendungsebene darf nur der Anwendungsdienst Verbindungsaufbau benutzt werden.

Der Anwendungsdienst Verbindungsaufbau darf von einem Anwendungsprozeß in der Rolle als Auftraggeber nur benutzt werden, wenn sich bei ihm die mit der betreffenden Verbindungsnummer identifizierte Kommunikationsbeziehung im Zustand KEINE-VERBINDUNG befindet (s. Zustandsübergangsdiagramm in 6.4).

Befindet sich in einem Kommunikationsteilnehmer eine Kommunikationsbeziehung im Zustand VERBINDUNG-AUFGEBAUT, und der Kommunikationsteilnehmer empfängt eine Verbindungsaufbau-Anforderung für diese Kommunikationsbeziehung, identifiziert

- über eine identische Verbindungsnummer oder
- identische systemweite Namen für den rufenden und den gerufenen Anwendungsprozeß,

so wird dieser Verbindungsaufbauwunsch nicht akzeptiert. Für die bestehende Verbindung wird in beiden Fällen der Anwendungsdienst Verbindungsabbruch aufgerufen. Die vor dem Eintreffen der Anforderung bereits bestehende Kommunikationsbeziehung wird damit nach 6.4 sofort abgebrochen.

Im zweiten Fall wird auf die mit einer anderen Verbindungsnummer empfangene Verbindungsaufbau-Anforderung zusätzlich mit einem Verbindungsaufbau-Fehlertelegamm geantwortet. In diesem Telegramm sind die Datenfelder Fehlerklasse: FK = 1 (Anwendungsreferenz) und Fehlercode: FC = 14 (falscher Zustand) anzugeben.

Dabei gelten folgende Randbedingungen: Die systemweiten Namen in dem geschlossenen Informationsverbundsystem wurden nicht verändert und alle beteiligten Geräte haben ihre hierarchische (Adress) Position im Netz beibehalten.

Wird von einem Anwendungsprozeß beim Verbindungsaufbau eine Kollision erkannt, das heißt, wurde quasi gleichzeitig von beiden zukünftigen Kommunikationspartnern ein Verbindungsaufbau zum jeweils anderen Anwendungsprozeß gewünscht, so ist von beiden Anwendungsprozessen der Verbindungsaufbauwunsch (des Kommunikationspartners) mit einem Verbindungsaufbau.Antwort[Ergebnis(-)]-Dienstprimitiv abzulehnen. Ein Verbindungsaufbau zwischen den beiden Kommunikationspartnern erfolgt nicht und muß neu angefordert werden. Im Verbindungsaufbau.Antwort[Ergebnis(-)]-Dienstprimitiv sind als Parameter die Fehlerklasse: FK = 8 (Verbindungsaufbau) und der Fehlercode: FC = 15 (Aufbaukollision) anzugeben.

6.2 Anwendungsdienst Verbindungsabbau

6.2.1 Allgemeines

Der bestätigte Anwendungsdienst Verbindungsabbau⁴⁸⁾ wird von einem Anwendungsprozeß benutzt, um eine Kommunikationsbeziehung mit einem Kommunikationspartner geordnet, das heißt ohne Datenverlust zu beenden.

Das Abbauen der Kommunikationsbeziehung darf von jedem der beiden Kommunikationspartner gleichberechtigt eingeleitet werden, sofern sie den Anwendungsdienst in der Rolle als Auftraggeber unterstützen.

6.2.2 Dienstprimitiv-Parameter

Den Dienstprimitiven des Anwendungsdienstes Verbindungsabbau sind die in Tabelle 6 angegebenen Parameter zugeordnet.

⁴⁷⁾ Die Verbindungsnummer, eingeleitet mit dem Gerätesteuersymbol DC4, wird jedem Anwendungstelegamm unabhängig vom verwendeten Transportsystem vorangestellt.

⁴⁸⁾ MMS: Conclude

Tabelle 6: Dienstprimitiv-Parameter des Anwendungsdienstes Verbindungsabbau

Parametername	Verbindungsabbau.			
	Anf	/	Anz	Ant / Bes
Argument	E			-
Ergebnis (+)	-			A
Ergebnis (-)	-			A
Fehlerklasse	-			E
Fehlercode	-			E
Zusätzlicher-Code	-			O
Zusätzliche-Beschreibung	-			O

6.2.3 Argument

Die Parametergruppe Argument ist leer.

6.2.4 Anforderungstelegramm

Die Struktur des Verbindungsabbau-Anforderungstelegramms ist in Bild 30 dargestellt.

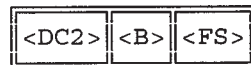


Bild 30: Struktur des Verbindungsabbau-Anforderungstelegramms

6.2.5 Ergebnis (+)

Die Parametergruppe Ergebnis(+) ist leer.

6.2.6 Antworttelegramm

Die Struktur des Verbindungsabbau-Antworttelegramms ist in Bild 31 dargestellt.

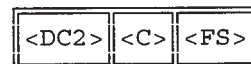


Bild 31: Struktur des Verbindungsabbau-Antworttelegramms

6.2.7 Ergebnis(-)

Die Beschreibung der Parameter der Parametergruppe Ergebnis(-) ist 11.1 zu entnehmen.

6.2.8 Fehlertelegramm

Das Verbindungsabbau-Fehlertelegramm entspricht dem Fehlertelegramm Typ II (siehe 11.2).

6.2.9 Dienstprozedur

Der den Verbindungsabbau anfordernde Anwendungsprozeß beginnt den Abbau mit dem Aufruf des Verbindungsabbau.Anforderung-Dienstprimativs. Nach Aufruf dieses Dienstprimativs darf er, mit Ausnahme des Verbindungsabbruch.Anforderung-Dienstprimativs in der abzubauenen Kommunikationsbeziehung, kein anderes Anforderung-Dienstprimativ mehr aufrufen, bevor er nicht mit einem Verbindungsabbau.Bestätigung-Dienstprimativ vom lokalen Kommunikationssystem über das Ergebnis des angeforderten Verbindungsabbaus informiert worden ist. Der den Verbindungsabbau anfordernde Anwendungsprozeß muß in der Rolle als Auftragnehmer die Beantwortung zwischenzeitlich eintreffender Dienstausführungsanforderungen vom assoziierten Anwendungsprozeß fortsetzen, um diese Dienstausführungsanforderungen abzuschließen.

Bei Verwendung von Anwendungsdiensten aus der Dienstgruppe Speicherbereich-bezogene Anwendungsdienste (siehe Abschnitt 7) ist zu beachten, daß ein Anwendungsprozeß einen Verbindungsabbau nicht anfordern darf, wenn in ihm ein vom assoziierten Anwendungsprozeß über die abzubauenende Kommunikationsbeziehung gesteuertes Lademodul (siehe 7.2) aktiv ist, das sich in einem der Zustände SPEICHERBEREICH-WIRD-GELADEN, SPEICHERBEREICH-VOLLSTÄNDIG-GELADEN, SPEICHERBEREICH-UNVOLLSTÄNDIG-GELADEN, L1, L2, L3 oder L9 befindet.

Ein zum Verbindungsabbau aufgeforderter Anwendungsprozeß muß ein Verbindungsabbau-Antwort-

Dienstprimitiv aufrufen, in dem er durch die Besetzung der Parametergruppe Ergebnis spezifiziert, ob er den Verbindungsabbau akzeptiert [Ergebnis(+)], oder ob er den Verbindungsabbau nicht akzeptiert [Ergebnis(-)], bevor er mit Ausnahme eines Verbindungsabbruch.Anforderung-Dienstprimitivs ein anderes Dienstprimitiv in der Kommunikationsbeziehung aufrufen darf, deren Abbau angefordert wurde.

Ein zum Verbindungsabbau aufgeforderter Anwendungsprozeß darf den Verbindungsabbau unter folgenden Bedingungen nicht akzeptieren:

- Wenn in ihm ein vom assoziierten Anwendungsprozeß über die abzubauenende Kommunikationsbeziehung gesteuertes Rücklademodul aktiv ist (siehe 7.3).
- Wenn in ihm ein vom assoziierten Anwendungsprozeß über die abzubauenende Kommunikationsbeziehung gesteuertes Lademodul aktiv ist, das sich in einem der Zustände SPEICHERBEREICH-WIRD-GELADEN, SPEICHERBEREICH-VOLLSTÄNDIG-GELADEN, SPEICHERBEREICH-UNVOLLSTÄNDIG-GELADEN, L1, L2, L3 oder L9 befindet bzw. als Zwischenzustand kurzfristig eingenommen wurde (siehe 7.2).
- Solange er in der Rolle als Auftraggeber über die abzubauenende Kommunikationsbeziehung noch auf die Beantwortung von Dienstausführungsanforderungen wartet, die von ihm veranlaßt worden sind.

Außerdem darf ein Verbindungsabbau von einem Anwendungsprozeß nicht durchgeführt werden, wenn er vom lokalen Kommunikationssystem mit einem Verbindungsabbau.Bestätigung-Dienstprimitiv über das Ergebnis eines Verbindungsabbaus informiert wird, der nicht von ihm mit einem Verbindungsabbau.Anforderung-Dienstprimitiv angefordert wurde.

ANMERKUNG: In diesem Fall nimmt der Anwendungsprozeß die erhaltene Information über den Verbindungsabbau nicht zur Kenntnis. Auch wird kein Ablauffehler-Meldungstelegramm oder Verbindungsabbau-Fehlertelegramm an den Kommunikationspartner gesendet.

Wird beim Verbindungsabbau eine Kollision erkannt, so ist in gleicher Weise zu verfahren, wie bei einer Verbindungsaufbaukollision (siehe 6.1.9). Ein Verbindungsabbau erfolgt nicht und muß neu angefordert werden. Im Verbindungsabbau.Antwort[Ergebnis(-)]-Dienstprimitiv sind als Parameter die Fehlerklasse: FK = 9 (Verbindungsabbau) und der Fehlercode: FC = 15 (Abbaukollision) anzugeben.

6.3 Anwendungsdienst Verbindungsabbruch

6.3.1 Allgemeines

Der unbestätigte Anwendungsdienst Verbindungsabbruch⁴⁹⁾ wird benutzt, um eine Kommunikationsbeziehung ungeordnet, das heißt ohne Absprache mit dem Kommunikationspartner abzubrechen. Ein solcher Abbruch kann beispielsweise bei der Ausführung eines Verbindungsaufbaus oder -abbaus durch eine Zeitüberwachung in den Zuständen VERBINDUNGS-AUFBAU-ANGEFORDERT oder VERBINDUNGS-ABBAU-ANGEFORDERT (siehe 6.4) oder in einem Zwischenzustand der anderen bestätigten Anwendungsdienste ausgelöst werden. Auf den eventuell erfolgenden Verlust von Daten wird dabei keine Rücksicht genommen.

Jeder Kommunikationsteilnehmer muß den Anwendungsdienst Verbindungsabbruch zwingend in der Rolle als Sender und als Empfänger unterstützen (Pflichtdienst).

6.3.2 Dienstprimitiv-Parameter

Den Dienstprimitiven des Anwendungsdienstes Verbindungsabbruch ist der in Tabelle 7 angegebene Parameter zugeordnet.

Tabelle 7: Dienstprimitiv-Parameter des Anwendungsdienstes Verbindungsabbruch

Parametername	Verbindungsabbruch.	
	Anf	Anz
Argument	E	E
Lokal-Erzeugt	-	E

6.3.3 Argument

Der Parameter der Parametergruppe Argument hat folgende Bedeutung:

⁴⁹⁾ MMS: Abort

Lokal-Erzeugt (LE): Dieser Parameter vom Typ *Boolesche Zahl* zeigt an, ob der Verbindungsabbruch vom Kommunikationspartner veranlaßt wurde oder lokal im eigenen Kommunikationssystem erzeugt worden ist. Der Parameter hat den Wert WAHR, wenn das lokale Kommunikationssystem den Verbindungsabbruch erzeugt hat. Wurde ein Verbindungsabbruch-Meldungstelegramm vom Kommunikationspartner empfangen, so hat der Parameter den Wert FALSCH.

6.3.4 Meldungstelegramm

Die Struktur des Verbindungsabbruch-Meldungstelegramms ist in Bild 32 dargestellt.



Bild 32: Struktur des Verbindungsabbruch-Meldungstelegramms

ANMERKUNG: Der Dienstprimitivparameter Lokal-Erzeugt wird nicht als Datenfeld im Anwendungstelegramm übertragen. Wird ein Verbindungsabbruch-Meldungstelegramm empfangen, so wird dadurch automatisch der Wert des Dienstprimitivparameters im empfangenden Kommunikationssystem zu Lokal-Erzeugt = FALSCH gesetzt.

6.3.5 Dienstprozedur

Der Verbindungsabbruch kann beim Erkennen von funktionalen Störungen zu jedem beliebigen Zeitpunkt entweder vom Kommunikationssystem oder vom Anwendungsprozeß eines Kommunikationsteilnehmers ausgelöst werden.

Wird der Verbindungsabbruch von einem Anwendungsprozeß nach dem Erkennen einer fatalen anwendungstechnischen Störung ausgelöst, so informiert der Anwendungsprozeß mit einem Verbindungsabbruch.Anforderung-Dienstprimitiv sein lokales Kommunikationssystem über den Verbindungsabbruch. Dieses sendet ein Verbindungsabbruch-Meldungstelegramm an das Kommunikationssystem des Kommunikationspartners. Der Anwendungsprozeß im Kommunikationspartner wird nach dem Empfang des Meldungstelegramms durch sein lokales Kommunikationssystem mit einem Verbindungsabbruch.Anzeige-Dienstprimitiv (Parameter: Lokal-Erzeugt = FALSCH) über den erfolgten Verbindungsabbruch informiert. Die Kommunikationsbeziehung ist damit bei beiden Kommunikationspartnern abgebrochen.

Wird der Verbindungsabbruch vom Kommunikationssystem nach dem Erkennen einer fatalen Störung ausgelöst (z.B. dann, wenn ein Verbindungsaufbau mit der Verbindungsnummer für eine Kommunikationsbeziehung angefordert wird, die bereits besteht), so muß der lokale Anwendungsprozeß mit einem Verbindungsabbruch.Anzeige-Dienstprimitiv (Parameter: Lokal-Erzeugt = WAHR) über den Verbindungsabbruch informiert werden.

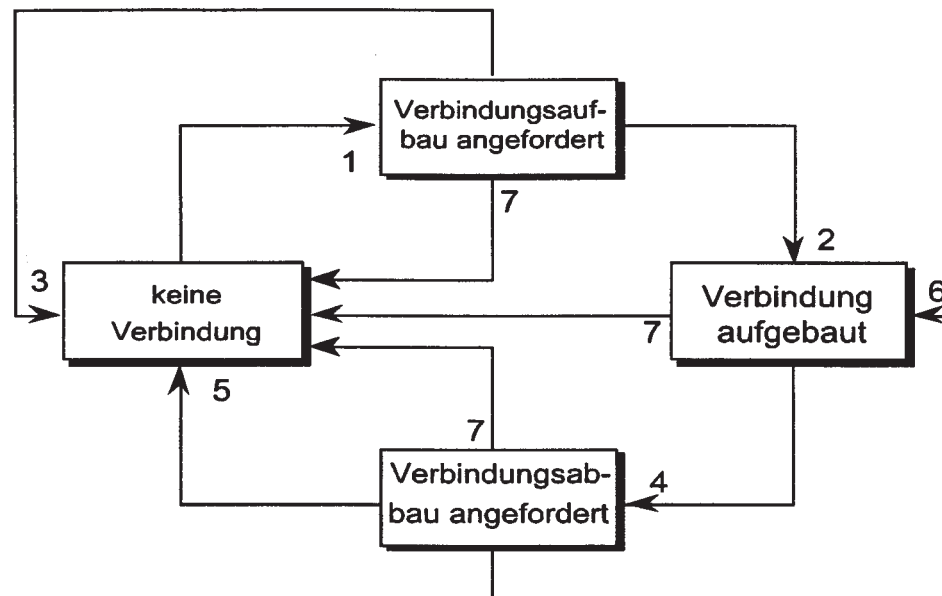
ANMERKUNG: Diese Verfahrensweise weicht von der in MMS beschriebenen Verfahrensweise bei der Anforderung zum Aufbau einer bereits bestehenden Verbindung ab, da es bei der Befolgung der MMS-Dienstprozedur geschehen kann, daß für den einen Kommunikationsteilnehmer die Verbindung besteht, während sie für den anderen nicht mehr besteht.

Besitzt ein Kommunikationsteilnehmer mehrere Anwendungsprozesse, so kann es je nach Ursache der vom Kommunikationssystem erkannten Störung erforderlich sein, alle Kommunikationsbeziehungen des Kommunikationsteilnehmers abubrechen und die Anwendungsprozesse hierüber zu informieren. Wenn dies trotz der Kommunikationsstörung möglich ist, ist der bzw. sind gegebenenfalls die assoziierten Kommunikationspartner jeweils mit einem Verbindungsabbruch-Meldungstelegramm zu informieren.

6.4 Anwendungskontext

Das Bild 33 zeigt das Zustandsübergangsdiagramm einer Kommunikationsbeziehung. Es ist zu beachten, daß zur Ermöglichung einer rollenneutralen Darstellung, die Zustandsübergänge im Bild durch Anwendungstelegramm-Namen gekennzeichnet sind.

ANMERKUNG: In Bild 33 ist zur Wahrung der Übersichtlichkeit die in Verbindung mit dem Anwendungsdienst Verbindungsabbau prinzipiell zulässige Benutzung des Anwendungsdienstes Dienstausführungsabbruch (siehe 10.2) nicht dargestellt.



Legende:

- 1 Verbindungsabbau-Anforderungstelegramm
- 2 Verbindungsabbruch-Antworttelegramm
- 3 Verbindungsabbau-Fehlertelegramm
- 4 Verbindungsabbau-Anforderungstelegramm
- 5 Verbindungsaufbau-Antworttelegramm
- 6 Verbindungsaufbau-Fehlertelegramm
- 7 Verbindungsaufbau-Meldungstelegramm

Bild 33: Zustandsübergangsdiagramm einer Kommunikationsbeziehung

7 Speicherbereich-bezogene Anwendungsdienste

7.1 Speicherorganisationsmodell

Diese Norm sieht vor, daß ein Kommunikationsteilnehmer einen Speicher besitzen kann, auf den ein oder gegebenenfalls mehrere Kommunikationspartner lesend und/oder schreibend zugreifen dürfen. Der Speicher kann im Rahmen der technischen Möglichkeiten eines Kommunikationsteilnehmers in Speicherbereiche⁵⁰⁾ beliebiger Größe unterteilt sein (siehe Bild 34).

Speicherbereiche besitzen einen individuellen Namen, der innerhalb eines Kommunikationsteilnehmers nur einmal verwendet werden darf. Es werden in dieser Norm "dynamische" und "statische" Speicherbereiche voneinander unterschieden:

- Dynamische Speicherbereiche "entstehen" dadurch, daß einem bestimmten Teil des Speichers auf Anforderung eines Kommunikationspartners ein individueller Name vorübergehend zugeordnet wird. Die Bindung dieses Namens an den Speicherbereich kann vom Kommunikationspartner wieder aufgehoben werden. Dynamische Speicherbereiche werden daher als "löschbare" Speicherbereiche bezeichnet. Durch das Aufheben der Namensbindung an den Speicherbereich geht für den Kommunikationspartner die Möglichkeit des Zugriffs auf die im Speicherbereich enthaltenen Daten unwiderruflich verloren.
- Statische Speicherbereiche werden vom Geräte- bzw. Systemhersteller während der Implementierung fest eingerichtet und sind bezüglich ihrer Attribute (Größe und Name) durch einen Kommunikationspartner nicht veränderbar. Daher werden statische Speicherbereiche im folgenden auch als "nicht löschbare" Speicherbereiche bezeichnet. Es ist zu beachten, daß sich hier "nicht löschar" ausschließlich auf den Speicherbereichnamen bezieht und nicht auf die Daten im Speicherbereich, die vom Kommunikationspartner mit den dazu vorgesehenen Anwendungsdiensten verändert werden können.

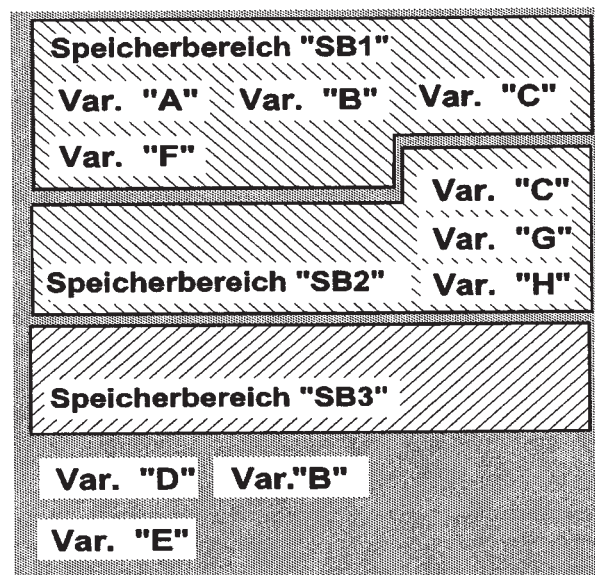
Ein Speicher bzw. ein Speicherbereich ist grundsätzlich in Speicherzellen unterteilt. Speicherzellen die-

⁵⁰⁾ MMS: Domain

nen zur Aufnahme von Datenobjekten. Speicherzellen, auf die gezielt mittels eines Bezeichners (Namens)⁵¹⁾ zugegriffen werden soll und die Datenobjekte wie beispielsweise Meßwerte oder Geräteparameter aufnehmen, werden in dieser Norm als Variable bezeichnet. Es werden zwei Klassen von Variablen unterschieden:

- Variablen, wie in Bild 34 beispielsweise die Variablen mit den Namen "A", "B" und "C", die in einem Speicherbereich abgelegt sind, werden als Variablen mit "Speicherbereich-bezogen"⁵²⁾ gültigem Namen bzw. als "Speicherbereich-bezogene Variable" bezeichnet. Für den Zugriff auf Speicherbereich-bezogene Variable ist grundsätzlich der Speicherbereichname (z.B. "SB1") und der Variablenname (z.B. "A") anzugeben. Der Name einer Speicherbereich-bezogenen Variablen darf nur einmal innerhalb eines Speicherbereichs, im Kommunikationsteilnehmer aber mehrfach verwendet werden.
- Variablen, wie in Bild 34 die Variablen "D" und "E", die in dem Teil des Speichers abgelegt sind, der keinen namentlich identifizierbaren Speicherbereich darstellt, werden als Variablen mit "geräteweit"⁵³⁾ gültigem Namen bzw. als "geräteweite Variable" bezeichnet. Für den Zugriff auf geräteweite Variable ist nur der Variablenname anzugeben. Daher kann der Name einer geräteweiten Variablen nur einmal pro Kommunikationsteilnehmer für die Identifikation einer geräteweiten Variable verwendet werden. Der Name darf aber, wie in Bild 34 bei der Variable mit dem Name "B", darüber hinaus zusätzlich im Kommunikationsteilnehmer ein- oder mehrfach zur Identifikation von Speicherbereich-bezogenen Variablen verwendet werden.

Die in dieser Norm vorgesehenen Variablen sind "statisch", das heißt ihre Attribute (Name, Typ und bei Speicherbereich-bezogenen Variablen auch der Name des Speicherbereichs, in dem die Variable existiert) werden vom Geräte- bzw. Systemhersteller während der Implementierung festgelegt und sind vom Kommunikationspartner nicht veränderbar. Variablen können daher nicht in dynamischen Speicherbereichen existieren, da deren Speicherbereichname "löschar" und damit nicht unveränderbar fest ist.



Legende:

Var. Variable (Speicherzelle)



statischer Speicherbereich



dynamischer Speicherbereich

Bild 34: Beispiel einer Speicherorganisation

ANMERKUNG: Die Existenz statischer Variablen beginnt üblicherweise mit dem "Einschalten" und endet mit dem "Ausschalten" des Kommunikationsteilnehmers. Als statische Variablen werden in dieser Norm aber auch solche Variablen bezeichnet, deren Attri-

51) Diese Norm unterstützt den in ISO/IEC 9506-1 beschriebenen Variablenzugriff über Adressen nicht.

52) MMS: Domain-specific

53) MMS: VMD-specific

bute bei der Implementierung zwar vordefiniert werden, deren "temporäre Existenz" jedoch zur Laufzeit durch lokale Aktionen im Kommunikationsteilnehmer beim Eintreten bestimmter vordefinierter Ereignisse begrenzt wird. Unter solchen Ereignissen sind beispielsweise die Aktivierung oder Deaktivierung eines Programmausführers (siehe Abschnitt 9) zu verstehen. Das heißt, an die Existenz eines Programmausführers kann automatisch die Existenz einer zugeordneten statischen Variable gebunden werden, wenn der Implementierer das vorsieht.

Der Anwendungsprozeß bzw. Kommunikationsteilnehmer, der Daten, wie beispielsweise den Inhalt eines Speicherbereichs bzw. den Wert einer Variable, zur Übertragung an einen anderen Kommunikationsteilnehmer bereitstellt, wird als "Datenquelle" bezeichnet. Als "Datensenke" wird der Anwendungsprozeß/Kommunikationsteilnehmer bezeichnet, an den die Daten von der Datenquelle übertragen werden.

Der Zugriff auf den Inhalt eines Speicherbereiches erfolgt segmentweise. Dazu definiert diese Norm die in diesem Abschnitt beschriebene Gruppe der "Speicherbereich-bezogenen Anwendungsdienste".

Die Speicherbereich-bezogenen Anwendungsdienste bieten je eine Anwendungsdienstgruppe zum segmentweisen Laden von Daten in einen dynamischen Speicherbereich (siehe 7.2) und zum segmentweisen Rückladen von Daten (siehe 7.3) aus einen Speicherbereich. Darüber hinaus definiert diese Norm den Anwendungsdienst LöscheSpeicherbereich, der die Zuordnung des Speicherbereichnamens an einen dynamischen Speicherbereich aufhebt (siehe 7.4).

Die in Abschnitt 8 beschriebenen "Variable-bezogenen Anwendungsdienste" werden für den Zugriff auf eine oder mehrere Variablen in einem Speicher oder Speicherbereich benutzt. Zur Vereinfachung des gleichzeitigen Zugriffs auf mehrere Variablen sieht diese Norm die Möglichkeit vor, "Variablenlisten" zu nutzen.

Eine Variablenliste ist ein über einen Bezeichner (Listenname) zugängliches Verzeichnis mit Variablennamen. Der Listenname wird dazu verwendet, um gleichzeitig lesend und/oder schreibend auf alle im Verzeichnis namentlich angegebenen Variablen zuzugreifen, ohne in der Lese-/Schreibanforderung alle Variablennamen einzeln angeben zu müssen. Es werden in dieser Norm "dynamische" und "statische" Variablenlisten unterschieden:

- Dynamische Variablenlisten werden auf Anforderung eines Kommunikationspartners angelegt. Dabei kann dieser Kommunikationspartner im Rahmen der technischen Möglichkeiten die Attribute der Variablenliste (Listenname, Anzahl und Namen der im Verzeichnis angegebenen Variablen) festlegen. Die Bindung des Listennamens an die Variablenliste kann vom Kommunikationspartner wieder aufgehoben werden. Dynamische Variablenlisten werden daher als "löschbare Variablenlisten" bezeichnet. Durch das Aufheben der Namensbindung an die Variablenliste geht die Möglichkeit zum Verwenden des Listennamens für Variablenzugriffsoperationen unwiderruflich verloren.
- Statische Variablenlisten werden vom Geräte- bzw. Systemhersteller während der Implementierung fest eingerichtet und sind bezüglich ihrer Attribute durch einen Kommunikationspartner nicht veränderbar. Daher werden statische Variablenlisten im folgenden auch als "nicht löschbare Variablenlisten" bezeichnet.

Ebenso wie Variablen können Variablenlisten einen "Speicherbereich-bezogen" oder einen "geräteweit" gültigen Namen besitzen. Dabei ist zu beachten, daß dynamische Variablenlisten in statischen Speicherbereichen existieren dürfen. Dynamische Variablenlisten dürfen jedoch nicht in dynamischen Speicherbereichen existieren.

7.2 Ladedienste

7.2.1 Allgemeines

Die Dienstgruppe "Ladedienste" umfaßt die bestätigten Anwendungsdienste

- BeginneLadesequenz;
- LadeSpeichersegment;
- BeendeLadesequenz.

Mit den Ladediensten können von einem Anwendungsprozeß (Datenquelle) in einen anzugebenden, dynamischen Speicherbereich eines Kommunikationspartners (Datensenke) segmentweise Daten geladen werden.

In dem Anwendungsprozeß, der als Datensenke auftritt, ist zur Steuerung und Überwachung des Ladevorgangs eine Instanz erforderlich, die im Rahmen dieser Norm als "Lademodul" bezeichnet wird. In einem Kommunikationsteilnehmer/Anwendungsprozeß können gleichzeitig mehrere Lademodule existieren, die über Bezeichner (die Namen des jeweils zu ladenden Speicherbereichs) voneinander unterschieden werden. Das heißt, es können, wenn es die technischen Gegebenheiten in einem Kommunikationsteilnehmer zulassen, quasi parallel zueinander verschiedene Ladevorgänge durchgeführt werden.

Das Zusammenwirken der drei Ladedienste ist anhand der graphischen Darstellung des Ladeprotokolls und eines Zustandsübergangsdiagramms in 7.2.5 gezeigt.

7.2.2 Anwendungsdienst BeginneLadesequenz

7.2.2.1 Allgemeines

Der bestätigte Anwendungsdienst BeginneLadesequenz⁵⁴⁾ wird von dem als Datenquelle auftretenden Anwendungsprozeß benutzt, um einen als Datensenke auftretenden Kommunikationspartner zu veranlassen, die Datenübertragung in einen namentlich bezeichneten, dynamischen Speicherbereich vorzubereiten und nach Abschluß der vorbereitenden Maßnahmen segmentweise die Übertragung der Daten anzufordern (Ziehprinzip).

Soll in einer Datensenke ein bereits existierender, mit einem bestimmten Namen versehener Speicherbereich unter Verwendung der Ladedienste mit neuen Daten geladen werden bzw. soll mit den Ladediensten unter einem bereits existierenden Namen ein neuer Speicherbereich angelegt werden, so ist von der Datenquelle mit dem Anwendungsdienst LöscheSpeicherbereich zuerst der Speicherbereichname zu löschen, bevor die Neuladung erfolgen kann. Das "Überschreiben" eines Speicherbereichs in einer Datensenke durch die Datenquelle mit dem Ladedienst kann somit nur bei dynamischen (löschbaren) Speicherbereichen erfolgen. Statische (aber auch dynamische) Speicherbereiche in einer Datensenke können auf Eigeninitiative der Datensenke mit den Rückladediensten "überschrieben" werden, ohne den Speicherbereichnamen vorher löschen zu müssen (siehe 7.3).

7.2.2.2 Dienstprimitiv-Parameter

Den Dienstprimitiven des Anwendungsdienstes BeginneLadesequenz sind die in Tabelle 8 angegebenen Parameter zugeordnet.

Tabelle 8: Dienstprimitiv-Parameter des Anwendungsdienstes BeginneLadesequenz

Parametername	BeginneLadesequenz.			
	Anf	/	Anz	Ant / Bes
Argument	E			-
Speicherbereichname	E			-
[<u>Liste-von-Fähigkeiten</u> :]	[E]			[-]
Code-der-Längenangabe	E			-
Benötigter-Speicherplatz	E			-
[zusätzliche Fähigkeiten]	[O]			[-]
Zugriff-teilbar	E			-
Ergebnis (+)	-			A
Ergebnis (-)	-			A
Fehlerklasse	-			E
Fehlercode	-			E
Zusätzlicher-Code	-			O
Zusätzliche-Beschreibung	-			O

7.2.2.3 Argument

Die Parameter der Parametergruppe Argument haben folgende Bedeutung:

Speicherbereichname (SN): Dieser Parameter vom Typ *Name* bezeichnet den Speicherbereich in der Datensenke, der "angelegt" werden soll bzw. in den die vorgesehenen Daten zu laden sind.

Liste-von-Fähigkeiten (LVF): Die Liste-von-Fähigkeiten repräsentiert eine Gruppe von Parametern, die die Anforderungen beschreiben, die vom Speicher der Datensenke zur erfolgreichen Einrichtung des Speicherbereichs bzw. die vom Lademodul zum erfolgreichen Laden erfüllt werden müssen.

Die Liste-von-Fähigkeiten muß als erstes Element die Angabe der zum Laden der Daten erforderlichen Speicherbereichgröße der Datensenke besitzen. Die Angabe der erforderlichen Speicherbereichgröße erfolgt durch die beiden Parameter:

Code-der-Längenangabe (CLA): Dieser Parameter vom Typ *vorzeichenlose ganze Zahl* (<16) legt die Art der Längenangabe im Parameter Benötigter-Speicherplatz fest. Zulässige Werte sind:

⁵⁴⁾ MMS: InitiateDownloadSequence

- 0 Längenangabe erfolgt in Byte;
- 1 Längenangabe erfolgt in Vielfachen von 256 Byte;
- 2 anwendungsspezifische Längenangabe;
- 3 gesperrt für zukünftige Erweiterungen dieser Norm.

Benötigter-Speicherplatz (BSP): Dieser Parameter vom Typ *vorzeichenlose ganze Zahl* legt die Länge des Speicherbereichs bzw. eine anwendungsspezifische Längenangabe fest.

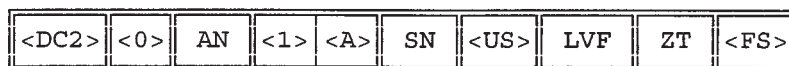
Die Liste-von-Fähigkeiten kann neben der erforderlichen Speicherbereichgröße optional weitere Anforderungen (**zusätzliche Fähigkeiten**) an den Speicher der Datensenke bzw. das Lademodul enthalten. Für Kommunikationsteilnehmer, die als Datensenke für die Ladedienste konzipiert sind, müssen die optional unterstützten "zusätzlichen Fähigkeiten" im Geräte- bzw. Systemhandbuch angegeben werden.

Zugriff-teilbar (ZT): Besitzt dieser Parameter vom Typ *Boolesche Zahl* den Wert WAHR, so bedeutet das, daß auf die Daten im Speicherbereich gleichzeitig durch mehrere Programmausführer (siehe Abschnitt 9) zugegriffen werden darf. Besitzt der Parameter den Wert FALSCH, so darf jeweils nur ein Programmausführer auf die Daten des Speicherbereichs zugreifen.

ANMERKUNG: Der Parameter ZT hat nur dann eine Bedeutung, wenn vom Kommunikationsteilnehmer die in Abschnitt 9 beschriebene ferngesteuerte Programmausführung unterstützt wird und der dazu erforderliche Programmausführer die zur Programmausführung benötigten Daten lokal aus dem betreffenden Speicherbereich ausliest.

7.2.2.4 Anforderungstelegramm

Die Struktur des BeginneLadesequenz-Anforderungstelegramms ist in Bild 35 dargestellt. Die Beschreibung der Datenfelder und deren Typen sind 7.2.2.3 zu entnehmen.



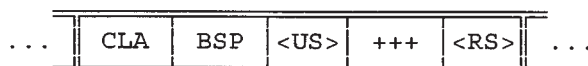
Legende:

AN	Auftragsnummer (siehe 5.2.3.2);
SN	Speicherbereichname;
LVF	Liste-von-Fähigkeiten;
ZT	Zugriff-teilbar.

Bild 35: Struktur des BeginneLadesequenz-Anforderungstelegramms

Das Datenfeld ZT wird im Anwendungstelegramm durch ein Zeichen dargestellt. Die Datenfelder SN und LVF haben eine variable Länge. Wie Bild 36 zeigt, ist die Struktur des Telegrammunterabschnitts LVF davon abhängig, ob er "zusätzliche Fähigkeiten" enthält oder nicht.

Telegrammunterabschnitt LVF enthält "zusätzliche Fähigkeiten":



Telegrammunterabschnitt LVF enthält keine "zusätzlichen Fähigkeiten":



Legende:

CLA	Code-der-Längenangabe;
BSP	Benötigter-Speicherplatz;
+++	zusätzliche Fähigkeiten (optional).

Bild 36: Codierung des Telegrammunterabschnitts LVF

Das Datenfeld CLA wird im Telegrammunterabschnitt LVF durch ein Zeichen dargestellt. Das Datenfeld BSP hat eine variable Länge und wird, wenn die Liste-von-Fähigkeiten "zusätzliche Fähigkeiten" spezifiziert, durch das Separatorzeichen <US> begrenzt. Anderenfalls wird sein Ende und damit das Ende des Telegrammunterabschnitts LVF durch das Separatorzeichen <RS> angezeigt. Enthält die Liste-von-Fähigkeiten weitere Elemente, so werden diese jeweils durch das Separatorzeichen <US> voneinander getrennt.

7.2.2.5 Ergebnis(+)

Die Parametergruppe Ergebnis(+) ist leer.

7.2.2.6 Antworttelegramm

Die Struktur des BeginneLadesequenz-Antworttelegramms ist in Bild 37 dargestellt.



Legende: AN Auftragsnummer (siehe 5.2.3.2)

Bild 37: Struktur des BeginneLadesequenz-Antworttelegramms

7.2.2.7 Ergebnis(-)

Die Beschreibung der Parameter der Parametergruppe Ergebnis(-) ist 11.1 zu entnehmen.

7.2.2.8 Fehlertelegramm

Das BeginneLadesequenz-Fehlertelegramm entspricht dem Fehlertelegramm Typ I (siehe 11.2).

7.2.2.9 Dienstprozedur

Die Datensenke, die zum Beginnen der Ladesequenz aufgefordert wurde, muß als Ladesequenz-Startbedingung durch Prüfung sicherstellen,

- daß im Kommunikationsteilnehmer kein Speicherbereich mit dem angegebenen Speicherbereichnamen existiert und
- daß die in der Liste von Fähigkeiten spezifizierten Anforderungen erfüllbar sind.

Sind die vorstehenden Ladesequenz-Startbedingungen erfüllt, so muß die Datensenke (der Anwendungsprozeß)

- einen Speicherbereich unter dem im BeginneLadesequenz.Anzeige-Dienstprimitiv angegebenen Speicherbereichnamen reservieren (anlegen),
- alle notwendigen Maßnahmen einleiten, um den folgenden Ladevorgang durchzuführen, d.h. ein Lademodul aktivieren und
- ein BeginneLadesequenz-Antwort[Ergebnis(+)]-Dienstprimitiv aufrufen.

Sind die Ladesequenz-Startbedingungen nicht erfüllt, so ist ein BeginneLadesequenz.Antwort[Ergebnis(-)]-Dienstprimitiv aufzurufen. In diesem Dienstprimitiv sind als Parameter die Fehlerklasse: 2 (Definition), Fehlercode 5 anzugeben, wenn der spezifizierte Speicherbereich bereits existiert, oder die Fehlerklasse 3 (Ressource), Fehlercode: 4 oder 5, wenn die im Parameter Liste-von-Fähigkeiten spezifizierten Anforderungen nicht erfüllbar sind.

7.2.3 Anwendungsdienst LadeSpeichersegment

Der bestätigte Anwendungsdienst LadeSpeichersegment⁵⁵⁾ wird von dem als Datensenke auftretenden Anwendungsprozeß benutzt, um bei der Datenquelle die Übertragung eines Datensegments anzufordern.

ANMERKUNG: Die Datenquelle ist der Auftraggeber des Anwendungsdienstes BeginneLadesequenz. Beim Anwendungsdienst LadeSpeichersegment tritt die Datenquelle jedoch als Auftragnehmer auf. Das heißt, es erfolgt innerhalb der Ladesequenz ein Rollentausch (Bild 42).

7.2.3.1 Dienstprimitiv-Parameter

Den Dienstprimitiven des Anwendungsdienstes LadeSpeichersegment sind die in Tabelle 9 angegebenen Parameter zugeordnet.

⁵⁵⁾ MMS: DownloadSegment

Tabelle 9: Dienstprimitiv-Parameter des Anwendungsdienstes LadeSpeichersegment

Parametername	LadeSpeichersegment.		
	Anf	/ Anz	Ant / Bes
Argument	E		-
Speicherbereichname	E		-
Ergebnis (+)	-		A
Datentyp	-		E
Nutzdaten	-		E
Mehr-folgt	-		E
Ergebnis (-)	-		A
Fehlerklasse	-		E
Fehlercode	-		E
Zusätzlicher-Code	-		O
Zusätzliche-Beschreibung	-		O

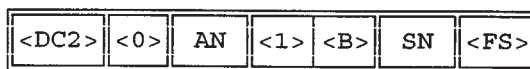
7.2.3.2 Argument

Der Parameter der Parametergruppe Argument hat folgende Bedeutung:

Speicherbereichname (SN): Dieser Parameter vom Typ *Name* bezeichnet den Speicherbereich in der Datensinke, in den die Daten zu laden sind.

7.2.3.3 Anforderungstelegramm

Die Struktur des LadeSpeichersegment-Anforderungstelegramms ist in Bild 38 dargestellt. Die Beschreibung des Datenfeldes SN und dessen Typ sind 7.2.3.2 zu entnehmen.



Legende: AN Auftragsnummer (siehe 5.2.3.2);
SN Speicherbereichname.

Bild 38: Struktur des LadeSpeichersegment-Anforderungstelegramms

Das Datenfeld SN hat eine variable Länge.

7.2.3.4 Ergebnis(+)

Die Parameter der Parametergruppe Ergebnis(+) haben folgende Bedeutung:

Datentyp (DT): Dieser Parameter vom Typ *vorzeichenlose ganze Zahl* (<16) gibt an, ob die zu ladenden Nutzdaten als Oktettfolge oder als Textzeichenfolge übertragen werden. In dieser Norm werden die Nutzdatentypen

- 0 Oktettfolge (4-Bit-codiert);
- 1 Textzeichenfolge;
- 2 Oktettfolge (6-Bit-codiert);

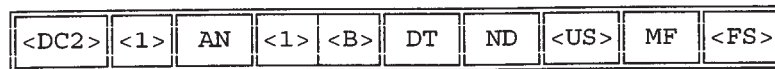
unterschieden.

Nutzdaten (ND): Dieser Parameter repräsentiert die zu ladenden Daten.

Mehr-folgt (MF): Besitzt dieser Parameter vom Typ *Boolesche Zahl* den Wert WAHR, so bedeutet das, daß mindestens ein weiteres LadeSpeichersegment-Anforderungstelegramm zum vollständigen Laden des Speicherbereichs erforderlich ist. Besitzt der Parameter den Wert FALSCH, so ist mit diesem LadeSpeichersegment-Anforderungstelegramm der zugehörige Speicherbereich vollständig übertragen.

7.2.3.5 Antworttelegramm

Die Struktur des LadeSpeichersegment-Antworttelegramms ist in Bild 39 dargestellt. Die Beschreibung der Datenfelder und deren Typen sind 7.2.3.4 zu entnehmen.



Legende: AN Auftragsnummer (siehe 5.2.3.2);
DT Datentyp;
ND Nutzdaten;
MF Mehr-folgt.

Bild 39: Struktur des LadeSpeichersegment-Antworttelegramms

Die Datenfelder DT und MF werden im Anwendungstelegramm durch je ein Zeichen dargestellt. Das Datenfeld ND hat eine variable Länge.

7.2.3.6 Ergebnis(-)

Die Beschreibung der Parameter der Parametergruppe Ergebnis(-) ist 11.1 zu entnehmen.

7.2.3.7 Fehlertelegramm

Das LadeSpeichersegment-Fehlertelegramm entspricht dem Fehlertelegramm Typ I (siehe 11.2).

7.2.3.8 Dienstprozedur

Akzeptiert der Anwendungsprozeß der Datensenke die Anforderung zum Speicherbereich-Laden, so ruft er das BeginneLadesequenz.Antwort[Ergebnis(+)]-Dienstprimitiv auf und bekundet damit dem Kommunikationspartner seine Bereitschaft zum Beginn der Ladesequenz. Dann wechselt er seine Rolle. Vom Auftragnehmer wird er zum Auftraggeber. Durch das Aufrufen des LadeSpeichersegment.Anforderung-Dienstprimitivs fordert er bei der Datenquelle den Inhalt (die Daten) des ersten Segmentes des zu ladenden Speicherbereichs an. Das Anfordern von Daten wird so lange fortgesetzt, wie LadeSpeichersegment.Bestätigung-Dienstprimitive mit dem Parameter Mehr-folgt = WAHR vom lokalen Kommunikationssystem empfangen werden. Wird ein LadeSpeichersegment.Bestätigung[Ergebnis(+)]-Dienstprimitiv mit dem Parameter Mehr-folgt = FALSCH empfangen, wird die Beendigung der Ladesequenz mit dem BeendeLadesequenz.Anforderung-Dienstprimitiv veranlaßt (s. Abs. 7.2.4).

Wird in der Phase der Ladesequenz, in der die Nutzdatenübertragung stattfindet, die Verbindung zwischen den beiden Kommunikationspartnern als unterbrochen erkannt, so ist der Inhalt des zu ladenden Speicherbereichs von beiden Kommunikationspartnern als nicht übertragen zu betrachten. Darüber hinaus ist in der Datensenke die Bindung des Speicherbereichsnamens an den Speicherbereich aufzuheben. Die Ladesequenz ist somit gescheitert und muß nach dem Wiederherstellen der Kommunikationsbeziehung komplett neu begonnen werden.

Wird in der Phase der Ladesequenz, in der die Nutzdatenübertragung stattfindet, mit dem Anwendungsdienst Dienstausführungsabbruch (siehe 10.2) das Abbrechen einer LadeSpeichersegment-Anforderung angefordert, so ist diese Dienstausführungsabbruch-Anforderung zurückzuweisen.

7.2.4 Anwendungsdienst BeendeLadesequenz

Der bestätigte Anwendungsdienst BeendeLadesequenz⁵⁶⁾ wird von der Datensenke benutzt, um bei der Datenquelle die Beendigung des Ladevorgangs anzufordern.

ANMERKUNG: Die Rolle der beiden an der Ladesequenz beteiligten Kommunikationspartner als Auftraggeber und Auftragnehmer ist beim Anwendungsdienst BeendeLadesequenz gegenüber dem Anwendungsdienst BeginneLadesequenz getauscht.

7.2.4.1 Dienstprimitiv-Parameter

Den Dienstprimitiven des Anwendungsdienstes BeendeLadesequenz sind die in Tabelle 10 angegebenen Parameter zugeordnet.

⁵⁶⁾ MMS: TerminateDownloadSequence

Tabelle 10: Dienstprimitiv-Parameter des Anwendungsdienstes BeendeLadesequenz

Parametername	BeendeLadesequenz.			
	Anf	/	Anz	Ant / Bes
Argument	E			-
Speicherbereichname	E			-
Fehlerklasse	B			-
Fehlercode	B			-
Zusätzlicher-Code	B, O			-
Zusätzliche-Beschreibung	B, O			-
Ergebnis (+)	-			A
Ergebnis (-)	-			A
Fehlerklasse	-			E
Fehlercode	-			E
Zusätzlicher-Code	-			O
Zusätzliche-Beschreibung	-			O

7.2.4.2 Argument

Die Parameter der Parametergruppe Argument haben folgende Bedeutung:

Speicherbereichname (SN): Dieser Parameter vom Typ *Name* bezeichnet den Speicherbereich, für den der Ladevorgang beendet werden soll.

Die folgenden Parameter sind nur dann Bestandteil des BeendeLadesequenz-Anforderungs- bzw. Anzeige-Dienstprimativs, wenn der geladene Speicherbereich aufgrund einer oder mehrerer Fehlerbedingungen vom Lademodul wieder "gelöscht" worden ist.

Fehlerklasse (FK): Dieser Parameter vom Typ *vorzeichenlose ganze Zahl* (<16) legt die Klasse des aufgetretenen Fehlers fest. Die zulässigen Werte des Parameters sind Abs. 11.1 zu entnehmen.

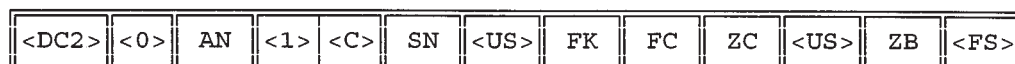
Fehlercode (FC): Dieser Parameter vom Typ *vorzeichenlose ganze Zahl* (<16) legt den Fehlertyp fest. Die zulässigen Werte des Parameters sind Abs. 11.1 zu entnehmen.

Zusätzlicher-Code (ZC): Dieser optionale Parameter vom Typ *vorzeichenlose ganze Zahl* repräsentiert eine implementierungsspezifische Fehlernummer, die ergänzende Informationen zum aufgetretenen Fehler liefert und im Geräte- bzw. Systemhandbuch offenzulegen ist.

Zusätzliche-Beschreibung (ZB): Dieser optionale Parameter vom Typ *Textzeichenfolge* repräsentiert eine implementierungsspezifische, vom Geräte- bzw. Systembediener auf einer geeigneten Anzeige unmittelbar lesbare, d.h. unverschlüsselte Fehlerkurzbeschreibung.

7.2.4.3 Anforderungstelegramm

Die Struktur des BeendeLadesequenz-Anforderungstelegramms ist in Bild 40 dargestellt. Die Beschreibung der Datenfelder und deren Typen sind 7.2.4.2 zu entnehmen.



Legende:

AN	Auftragsnummer (siehe 5.2.3.2);
SN	Speicherbereichname;
FK	Fehlerklasse (Bedingung);
FC	Fehlercode (Bedingung);
ZC	Zusätzlicher-Code (Bedingung, optional);
ZB	Zusätzliche-Beschreibung (Bedingung, optional).

Bild 40: Struktur des BeendeLadesequenz-Anforderungstelegramms

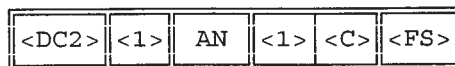
Die Datenfelder FK und FC werden im Anwendungstelegramm durch je ein Zeichen dargestellt. Die Datenfelder SN, ZC und ZB haben eine variable Länge.

7.2.4.4 Ergebnis(+)

Die Parametergruppe Ergebnis(+) ist leer.

7.2.4.5 Antworttelegramm

Die Struktur des BeendeLadesequenz-Antworttelegramms ist in Bild 41 dargestellt.



Legende: AN Auftragsnummer (siehe 5.2.3.2)

Bild 41: Struktur des BeendeLadesequenz-Antworttelegramms

7.2.4.6 Ergebnis(-)

Die Beschreibung der Parameter der Parametergruppe Ergebnis(-) ist 11.1 zu entnehmen.

7.2.4.7 Fehlertelegramm

Das BeendeLadesequenz-Fehlertelegramm entspricht dem Fehlertelegramm Typ I (siehe 11.2).

7.2.4.8 Dienstprozedur

Die Datenquelle fordert die Beendigung der Ladesequenz mit dem Anwendungsdienst BeendeLadesequenz an, wenn sie mit dem LadeSpeichersegment.Bestätigung[Ergebnis(+)]-Dienstprimitiv (Parameter Mehr-folgt = FALSCH) darüber informiert worden ist, daß die Übertragung aller Daten erfolgreich abgeschlossen ist.

Die Beendigung der Ladesequenz wird auch dann von der Datenquelle angefordert, wenn sie mit einem LadeSpeichersegment.Bestätigung[Ergebnis(-)]-Dienstprimitiv über ein nicht behebbares Problem in der Datenquelle beim zur Verfügungstellen der zu ladenden Daten informiert worden ist.

Wird eine Datenquelle zum Beenden einer Ladesequenz aufgefordert, obwohl noch nicht alle Daten in den vorgesehenen Speicherbereich der Datenquelle geladen wurden und auch keine Ladestörung vorliegt, so ist die BeendeLadesequenz.Anforderung mit einem Antwort[Ergebnis(-)]-Dienstprimitiv abzulehnen.

Die Anzeige eines Antwort[Ergebnis(-)]-Dienstprimitivs bedeutet, daß die gesamte Ladesequenz gescheitert ist. Die Bindung des Speicherbereichsnamens an den Speicherbereich ist aufzuheben. Der Inhalt dieses Speicherbereichs ist damit gelöscht.

7.2.5 Ladesequenz

Den Ablauf des Telegrammaustauschs bei der Speicherbereichladung, das heißt, das Protokoll der Ladedienste ist in Bild 42 dargestellt.

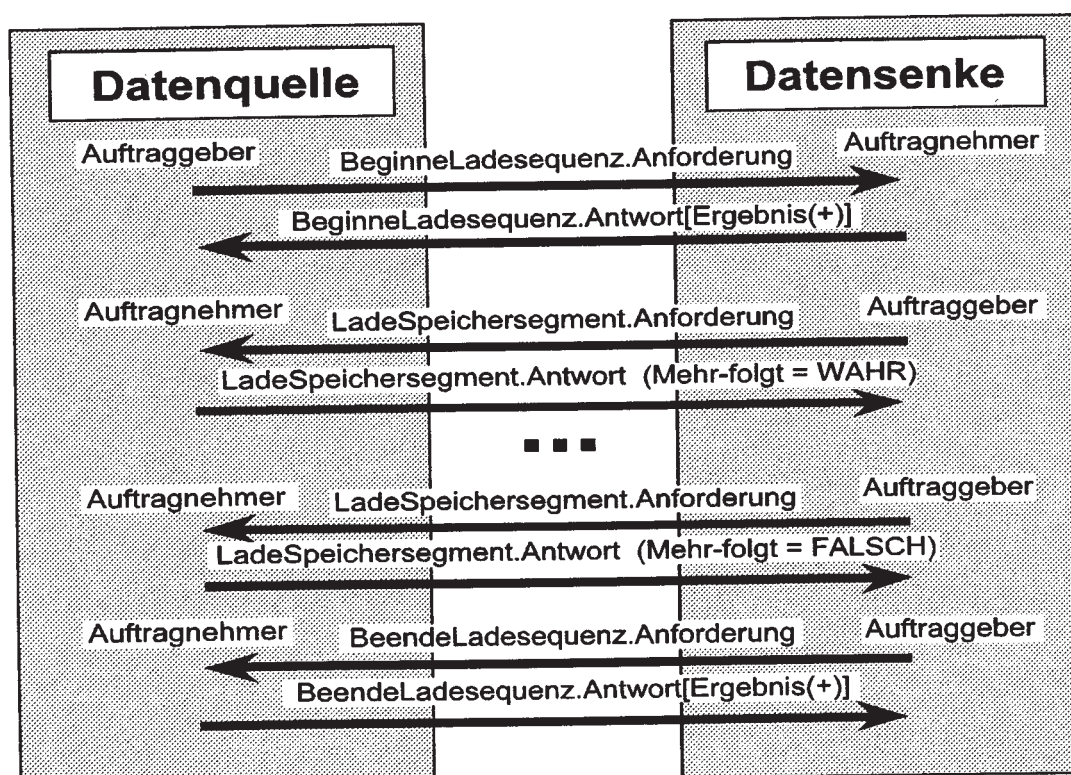


Bild 42: Ladeprotokoll