

1 Einleitung

Das Datenkommunikationssystem DCM (data communication methods) im BS2000 ist Teil des Datenkommunikationssystems. Dieses umfaßt Hardware für den Aufbau von Netzen und den Anschluß von Verarbeitungsrechnern, PCs und den Datenstationen sowie Software für das Konfigurieren, Verwalten und Steuern der Datenkommunikation über diese Netze. DCM stellt die Dienste dieser Netze als Funktionen im BS2000 zur Verfügung, und zwar in einer Form, die unabhängig von der Art und der Konfiguration des Netzes ist. Dabei ist DCAM (data communication access method) mit allen weiteren Zugriffsmethoden und Zugriffsmöglichkeiten in das Zugriffssystem für Datenkommunikation DCM des BS2000 eingebettet. Das dynamische Subsystem DCAM ist die Zugriffsmethode für die Kommunikation zwischen Programmen bzw. zwischen Programm und Datenstation. DCM setzt sich zusammen aus den Bausteinen:

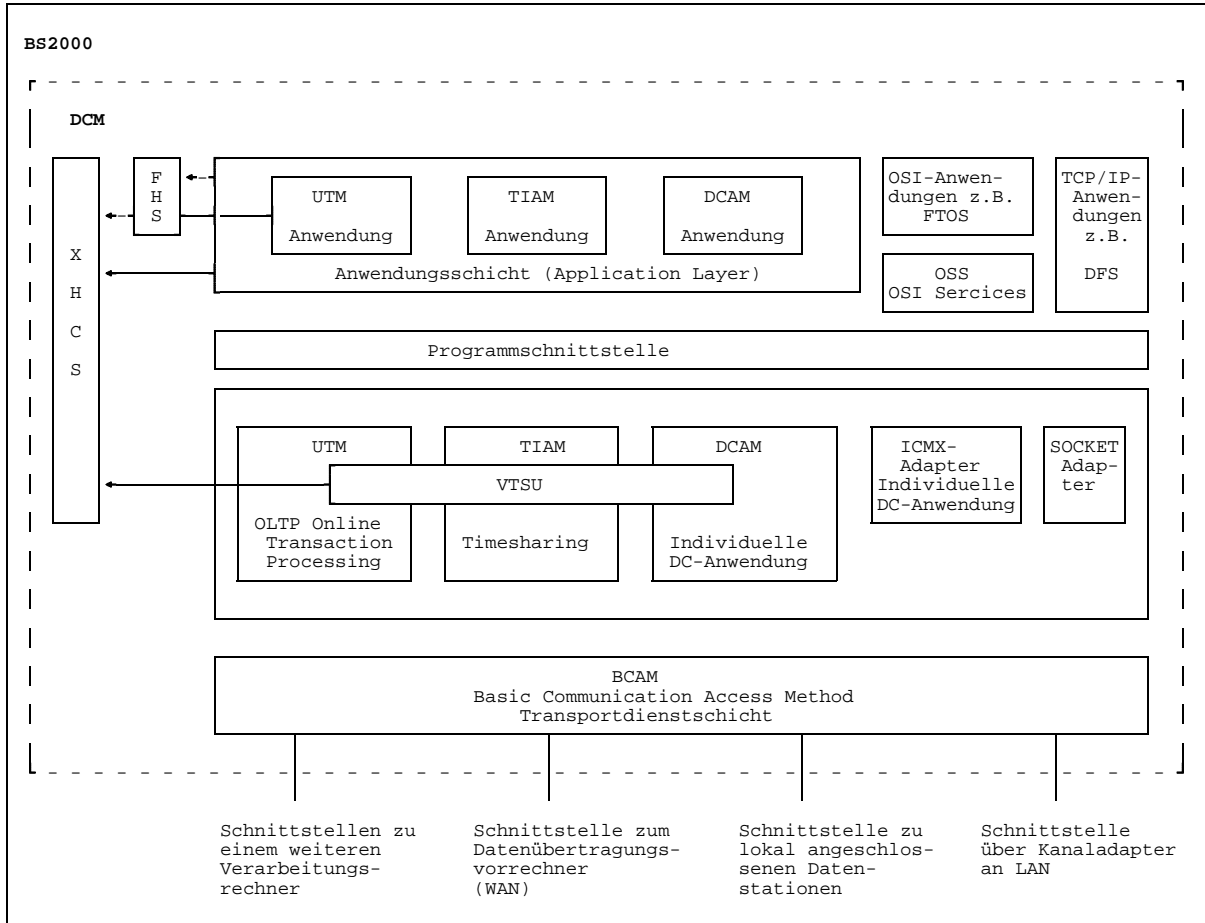
- **DCAM (data communication access method)** für die Realisierung der DCAM-Schnittstelle in Assembler und COBOL.
- **VTSU (virtual terminal support)** für die Realisierung der logischen Datenstationen.
- **TIAM (terminal interactive access method)** für die Realisierung der RTIO-Assembler-Schnittstelle (remote terminal input output).
- **UTM (universeller transaktions-monitor)** für die Realisierung der UTM-Schnittstelle.

Diese Bausteine haben ihrerseits eine definierte Schnittstelle zu **BCAM (basic communication access method)**. In diesem Basis-Baustein sind die, allen Zugriffsmethoden gemeinsamen Aufgaben wie Transportsteuerung, Zwischenspeicherung, usw. realisiert.

DCM setzt auf entkoppelte Schnittstellen des BS2000 auf und bietet selbst entkoppelte Schnittstellen. DCM bietet:

- **Zugriffsmethode DCAM** zur Kommunikation zwischen Programmen bzw. Programm und Datenstation.
- Unterstützung bei der Programmierung der Datenstationen durch den Einsatz **logischer Datenstationen**;
- Allgemeine C-Programmschnittstelle **SOCKETS** für den Zugang zu TCP/IP-Netzen.
- Allgemeine C-Schnittstelle **ICMX** die die OSI-Transportfunktionalität anbietet.
- Administration an wahlfreien Bedienplätzen, auch wenn gewünscht, in einem DCAM-Programm unter Verwendung des Bausteins für **Mehrkonsolbetrieb (UCON)**;
- Leistungsfähige Unterstützung der bekannten Schnittstellen für **Teilnehmerbetrieb (RTIO)**.

Dem Anwender von DCAM erschließen sich die vielfältigen Möglichkeiten des Datenkommunikationssystems auf freizügigen Datenverkehr mit allen Kommunikationspartnern (Anwendungen und Datenstationen).



Aufbau von DCM

XHCS kann erst ab BS2000 V10 eingesetzt werden

Diese Beschreibung der DCAM (**D**ata **C**ommunication **A**ccess **M**ethod)-Programmschnittstellen wendet sich an verschiedene Adressaten:

- Organisatoren und Einsatzplaner, die sich eine Übersicht über Umfang und Leistungsfähigkeit der Schnittstelle verschaffen wollen.
- Programmierer (Assembler, COBOL), die hier grundlegendes Wissen erwerben können, um im jeweiligen Programmierhandbuch weitere Details zu verstehen und anzuwenden.
- Systemverwalter und Netzadministratoren, die keine speziellen Kenntnisse der Schnittstelle benötigen, jedoch allgemeines Wissen über DCAM erwerben möchten.

Die Adressaten sollten mit dem BS2000 vertraut sein. Soweit sie DCAM anwenden wollen, ist auch die Kenntnis von Assembler oder COBOL sowie des OSI-Referenzmodells vorausgesetzt.

1.1 Wegweiser durch das Handbuch

Die Beschreibung der Kommunikationszugriffsmethode DCAM besteht aus folgenden drei Handbüchern:

- Allgemeine Beschreibung der DCAM-Programmschnittstelle
- Beschreibung der DCAM-COBOL-Schnittstelle
- Beschreibung der DCAM-Assembler-Schnittstelle

Die allgemeine Beschreibung der DCAM-Programmschnittstelle ist als Grundlage für den DCAM-Programmierer gedacht; sie ist aber auch für denjenigen geeignet, der sich nur einen Überblick über den Umfang und die Leistungsfähigkeit der DCAM-Schnittstelle verschaffen will. Der DCAM-Programmierer findet anschließend die Details für die Programmierung, je nach verwendeter Programmiersprache, in der Beschreibung der DCAM-COBOL- bzw. der DCAM-Assembler-Schnittstelle.

Die DCAM-Handbücher enthalten die Beschreibungen sowohl für DCAM(ISO)-Transport-Service-Anwendungen, wie auch der DCAM(NEA)-Transport-Service-Anwendungen. Dort wo es Unterschiede zwischen den beiden Anwendungen gibt, ist dies gekennzeichnet. Textteile, Abschnitte oder ganze Kapitel, die nur für DCAM(NEA)-Transport-Service-Anwendungen gelten, sind am Anfang des Textes links mit



gekennzeichnet.

Die vorliegende Beschreibung ist wie folgt gegliedert:

- Im Kapitel '**Einführung in die DCAM-Schnittstelle**' werden dem Anwender allgemeine Informationen über die DCAM-Schnittstelle gegeben. Ebenso werden grundlegende Begriffe und Überlegungen zur Planung von Programmen oder Programmsystemen erläutert.
- Im Kapitel '**Funktionen von DCAM**' werden die Funktionen aller DCAM-Aufrufe und Meldungen beschrieben.
- Im Kapitel '**Unterstützung logischer Datenstationen**' wird ein kurzer Überblick über Format-Datenstationen, logische Zeilendatenstationen und Edit-Optionen gegeben.
- Im Kapitel '**DCAM-Programme**' wird ein Überblick über die Codierung von DCAM-Programmen in Assembler und COBOL, ferner über die Ausführung solcher Programme gegeben.
- Im **Anhang** wird ein Überblick über die DCAM-Aufrufe, die Grenzwerte und dem Verbindungsaufbau von der Datenstation gegeben.

Die Beschreibung der Funktionen aller DCAM-Aufrufe und Meldungen ist identisch aufgebaut wie die entsprechenden Abschnitte in den Benutzerhandbüchern für Assembler und COBOL. Das erleichtert die parallele Benutzung der Handbücher.

Am Ende dieses Handbuchs befinden sich ein Fachwortverzeichnis, ein Literaturverzeichnis und ein Stichwortverzeichnis.

Thematisch zugeordnet sind diesem Benutzerhandbuch Informationsschriften über Rechnetze und Datenfernverarbeitung mit BS2000. Für Generierung und Administration, zur Programmierung der Kommunikationsrechner und der Datenstationen und zur Unterstützung logischer Datenstationen stehen eigene Handbücher zur Verfügung.

1.2 Änderungen gegenüber Vorgängerausgabe

Unterstützung logischer Datenstationen

Die Beschreibung des Kapitels 'Unterstützung logischer Datenstationen' wurde gekürzt. Eine genaue Beschreibung der VTSU-Schnittstelle mit dem VTSU-Control-Block, den logischen Steuerzeichen und den Statusinformationen finden Sie im Handbuch 'VTSU'.

Readme-Datei

Funktionelle Änderungen und Nachträge der aktuellen Produktversion zu diesem Handbuch entnehmen Sie bitte ggf. der produktspezifischen Readme-Datei. Sie finden die Readme-Datei auf Ihrem BS2000-Rechner unter dem Dateinamen `SYSDOC.produkt.version.READ-ME.D`. Die Benutzerkennung, unter der sich die Readme-Datei befindet, erfragen Sie bitte bei Ihrem zuständigen Systemverwalter. Die Readme-Datei können Sie mit dem Kommando `/SHOW-FILE` oder mit einem Editor ansehen oder auf einem Standarddrucker mit folgendem Kommando ausdrucken:

```
PRINT-FILE FILE-NAME=dateiname,LAYOUT-CONTROL=PARAMETERS(CONTROL-CHARACTERS=EBCDIC)
```

2 Einführung in die DCAM-Schnittstelle

Der Aufbau der DCAM-Programme ist bestimmt durch die Verwendung der DCAM-Schnittstelle und den Zugriff zum Datenkommunikationssystem.

Dieses Kapitel beschreibt:

- die Zugriffsmethode DCAM
- die Verkehrsbeziehungen im Datenkommunikationssystem mit DCAM
- Begriffe, die für das Verständnis wichtig sind
- Merkmale, die die **Leistungsfähigkeit** der DCAM-Schnittstelle bestimmen
- die **Grundstruktur** eines DCAM-Programms
- die **Planung** eines DCAM-Programms

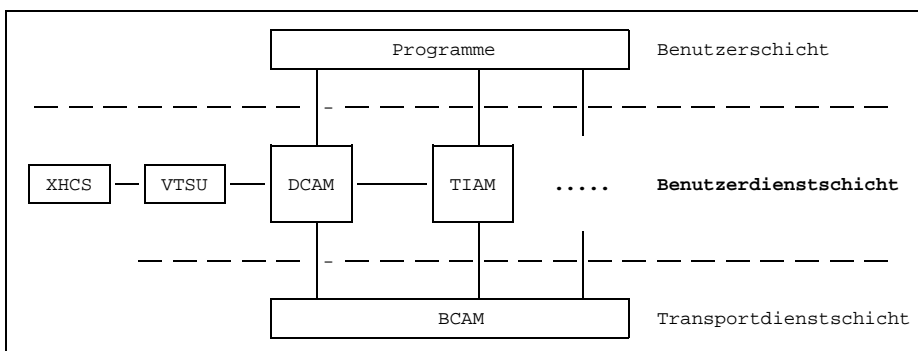
2.1 Die Kommunikationszugriffsmethode DCAM

DCAM bietet zwei unterschiedliche Funktionalitäten:

- DCAM(NEA)-Transport-Service-Funktionen und
- DCAM(ISO)-Transport-Service-Funktionen

DCAM(NEA)-Transport-Services

In der Netz-Architektur (NEA) ist DCAM der Benutzerdienstschicht zugeordnet. Diese ermöglicht Programmierern und Datenstationsbedienern den Zugang zum Datenkommunikationssystem.



Benutzerdienstschicht im BS2000 (NEA)

DCAM(NEA)-Transport-Services sind unter anderem:

- Mitteilung der Partnereigenschaften
- Verständigung über die Regeln des Nachrichtenaustausches
- Austausch von Anwenderdaten
- Darstellungsfunktionen für die Kommunikation mit Datenstationen

Um die Funktionen für jeweils zwei Partner zu erbringen, tauschen die betroffenen Instanzen Informationen nach vereinbarten Protokollen aus. Diese Protokolle bilden den Rahmen für DCAM(NEA)-Transport-Services-Anwendungen.

DCAM(ISO)-Transport-Services

Mit den DCAM(ISO)-Transport-Services haben Sie die Möglichkeit, Datenkommunikation auf der Basis der durch ISO genormten Transportdienste zu betreiben. Die DCAM-Schnittstelle wurde für diese Transportdienste angepaßt.

Überblick über das OSI-Referenzmodell

Das OSI-Referenzmodell stellt einen Rahmen dar für die Einordnung von Diensten, Funktionen und Schnittstellen. Es bildet damit die Basis für die Festlegung herstellerunabhängiger Kommunikationsprotokolle, die den Verbund "offener Systeme" ermöglichen.

In nachfolgendem Bild ist dieses Modell mit seinen 7 Schichten und deren Funktionen kurz dargestellt.

Näheres dazu finden Sie in der Druckschrift 'Wege zur offenen Kommunikation'.

Schicht	Bezeichnung der Schicht	Funktionen
7	Anwendungsschicht (Application Layer)	Steuerung der Durchführung von Kommunikationsaufgaben für eine Anwendung
6	Darstellungsschicht (Presentation Layer)	Festlegungen zur Darstellung und Bedeutung der ausgetauschten Daten
5	Kommunikationssteuerungsschicht (Session Layer)	Steuerung des Ablaufs der Kommunikation auf Transportverbindungen
4	Transportschicht (Transport Layer)	Steuerung der Verbindungen und des Datentransports zwischen den Teilnehmern in den Endsystemen
3	Vermittlungsschicht (Network Layer)	Steuerung der Verbindungen auf dem gesamten Übertragungsweg, d.h. zwischen den Endsystemen.
2	Sicherungsschicht (Link Layer)	Fehlergesicherte Übertragung auf den einzelnen Teilstrecken des gesamten Übertragungsweges
1	Bit-Übertragungsschicht (Physikal Layer)	Steuerung des physischen Übertragungsmediums

Das OSI-Referenzmodell

Die Schichten und ihre Funktionen werden häufig in zwei Gruppen unterteilt: In den Schichten 1 bis 4 liegen die Transportdienste, in den Schichten 5 bis 7 die Anwendungsdienste.

Im Rahmen der TRANDATA-Architektur für die Kommunikation gemäß ISO-Standard bietet DCAM(ISO) einen reinen Transportdienst. Demzufolge hat DCAM(ISO) einige Funktionen von DCAM(NEA) nicht. Insbesondere unterstützt DCAM(ISO) kein Protokoll oberhalb des Transportdienstes, also keine Nachrichtenaufbereitung für die Kommunikation mit Datenstationen.

Beachten Sie, daß der DCAM(ISO)-Transport-Service durch unterschiedliche Kommunikationsprotokolle erbracht werden kann, z.B. durch TCP/IP in Kombination mit einem Konvergenzprotokoll.

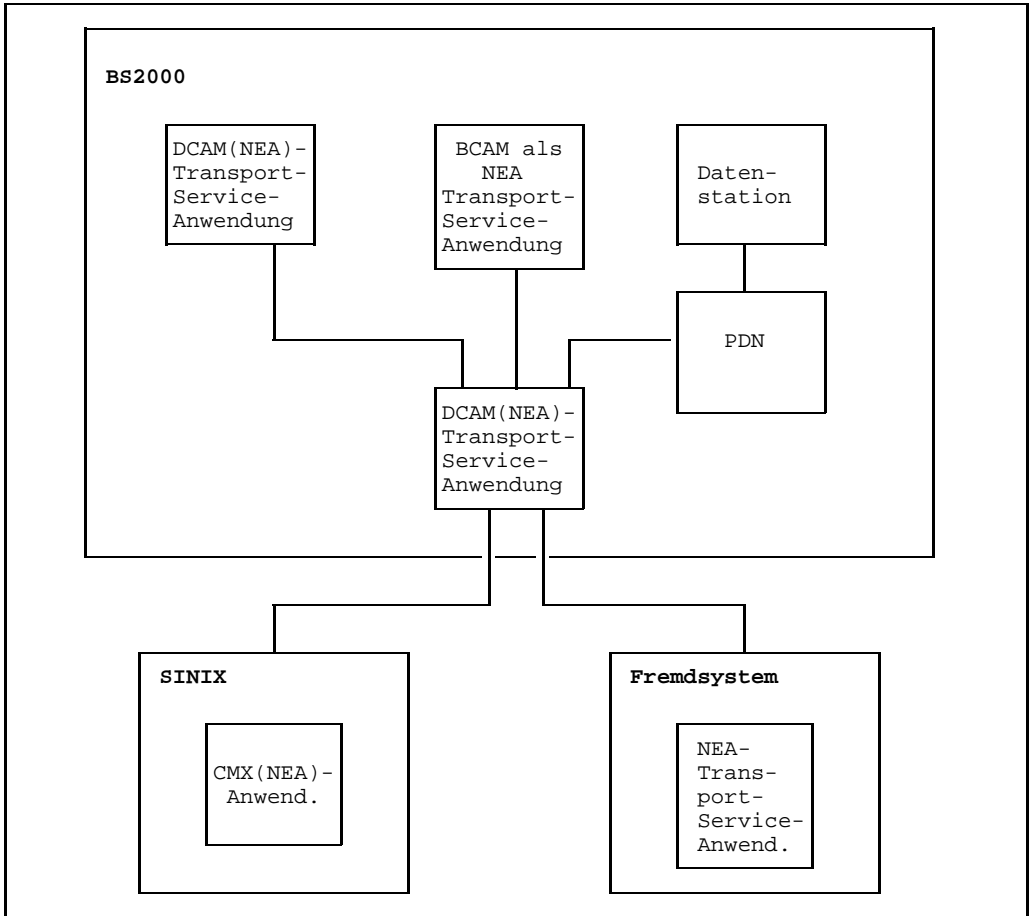
2.2 Verkehrsbeziehungen im Datenkommunikationssystem mit D

DCAM ermöglicht Kommunikation von einer oder mehreren Tasks (Programmen) im Verarbeitungsrechner mit beliebigen Anwendungen bzw. Programmen. Mit DCAM(NEA) ist zusätzlich die Kommunikation mit einer oder mehreren Datenstationen möglich. Mit DCAM(ISO) ist zusätzlich die Kommunikation mit CMX-Anwendungen möglich. Voraussetzung dazu sind entsprechende /BCMAP-Kommandos.

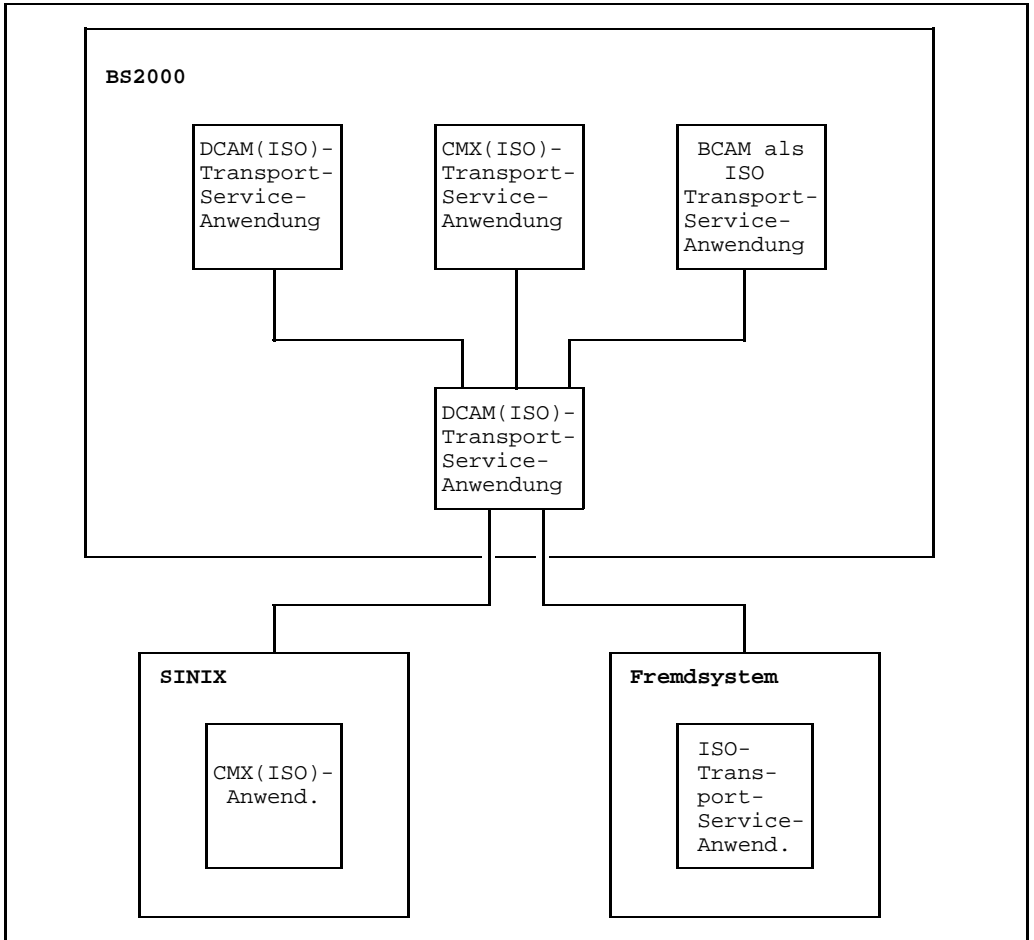
Damit unterstützt DCAM Verbindungen zu

- anderen DCAM-Anwendungen im selben oder in einem anderen BS2000-Rechner
- CMX-Anwendungen in SINIX-Rechnern
- CMX-Anwendungen in BS2000-Rechnern
- Anwendungen in Rechnern anderer Hersteller, sofern der entsprechende Transport-Service angeboten wird
- Datenstationen

Diese Kopplungen sind über LAN (local area network), WAN (wide area network) bzw. lokal möglich.



Kopplungen mit DCAM(NEA)-Transport-Service-Anwendungen



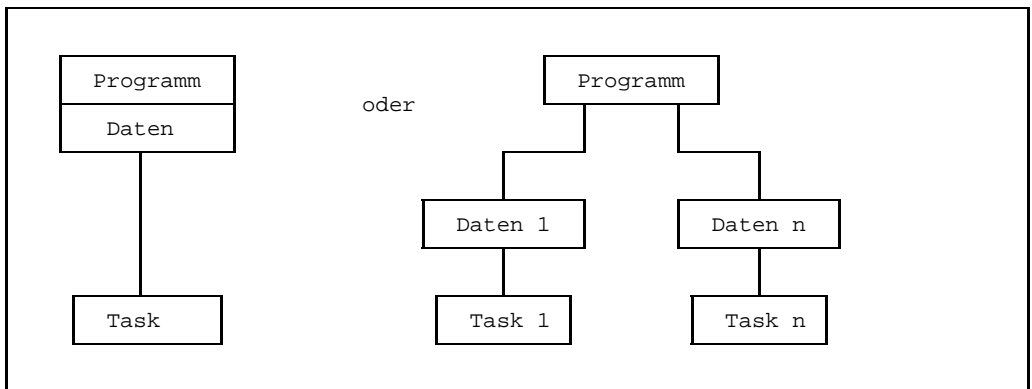
Kopplungen mit DCAM(ISO)-Transport-Service-Anwendungen

2.3 Grundlegende Begriffe

2.3.1 Programm, Daten, Task

Im BS2000 wird in der Regel der Programm- und Datenspeicher (Datenspeicher= Datenbereich im Programm), sowie die vom Programm gesteuerte Task als Einheit verwaltet. In den folgenden Ausführungen wird darum der Übersichtlichkeit halber auf die Unterscheidung dieser drei Komponenten verzichtet. So ist diese Einheit immer angesprochen, wenn von Programm oder Task die Rede ist. Für das Senden und Empfangen von Nachrichten ist der taskspezifische Datenspeicher von Bedeutung, da von ihm übertragen oder in ihn übertragen wird. Darum wird auch an einigen Stellen eigens vom Datenspeicher gesprochen.

Die spezifischen Eigenschaften eines DCAM-Programms, sein Aufbau und seine Organisation, sind in einem eigenen Abschnitt auf Seite 27ff beschrieben. Programme, die mehrere Tasks steuern, werden in auf den Seiten 115ff und 125ff beschrieben. Die Programmausführung, die DCAM-Task, ist auf Seite 129ff beschrieben.



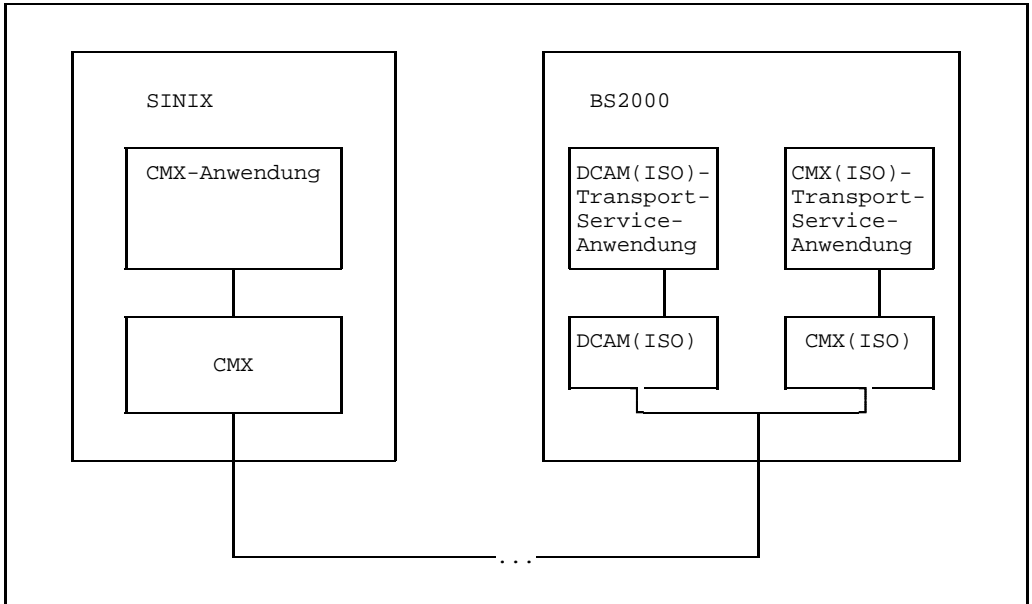
Die Task als Verwaltungseinheit im BS2000

2.3.2 Kommunikationspartner

Kommunikationspartner im Datenkommunikationssystem sind Anwendungen und, für DCAM(NEA)-Transport-Service-Anwendungen auch Datenstationen.

Anwendungen können sein:

- DCAM-Anwendungen
- UTM-Anwendungen
- PDN-Anwendungen
- CMX-Anwendungen
- Anwendungen von ISO-Transportdiensten in beliebigen Rechnern, z.B. CMX-Anwendungen in SINIX-Rechnern



Kommunikation zwischen einer CMX- und einer DCAM(ISO)-Transport-Service-Anwendung

Eine **DCAM-Anwendung** existiert ausschließlich in einem, vom BS2000 gesteuerten Verarbeitungsrechner. Sie wird in mindestens einem DCAM-Programm definiert und durch die Kommunikationszugriffsmethode erzeugt. Als adressierbare Einheit für eingehende und zu sendende Nachrichten, bzw. Meldungen wird sie von der Kommunikationszugriffsmethode verwaltet und auf Anforderung des Programms wieder aufgelöst. Sie ist die Adresse, unter der Tasks (Programme) oder Gruppen von Tasks dem Datenkommunikationssystem gegenüber bekannt sind. Sie werden damit zu Kommunikationspartnern.

Die Erzeugung des Kommunikationspartners DCAM-Anwendung ist eine Funktion der DCAM-Schnittstelle.

Eine DCAM-Anwendung kann **von einer Task eröffnet** werden, wenn sie einfach benutzbar definiert ist oder **von mehreren Tasks** (mehrfach benutzbar). Die erste eröffnende Task ist die Primärtask. Sekundärtasks können sich anschließen, wenn dies vorgesehen ist. In einem Programm können mehrere DCAM-Anwendungen eröffnet werden. Diese können isoliert, jede für sich betrachtet werden, darum wird diese Möglichkeit in der Beschreibung, die hier vorliegt, nicht weiter behandelt. Mit der Erzeugung der DCAM-Anwendung wird ihr Name festgelegt, siehe Beispiel im folgenden Bild.

Dieser **Name** kann vom Anwender beim YOPEN angegeben werden. Wird der Name vom Anwender nicht angegeben, wird er von DCAM erzeugt, oder es kann ein vordefinierter und dem Kommunikationssystem bekannter Name verwendet werden. Der Name der DCAM-Anwendung muß in dem Verarbeitungsrechner, in dem die DCAM-Anwendung erzeugt wird, eindeutig sein.

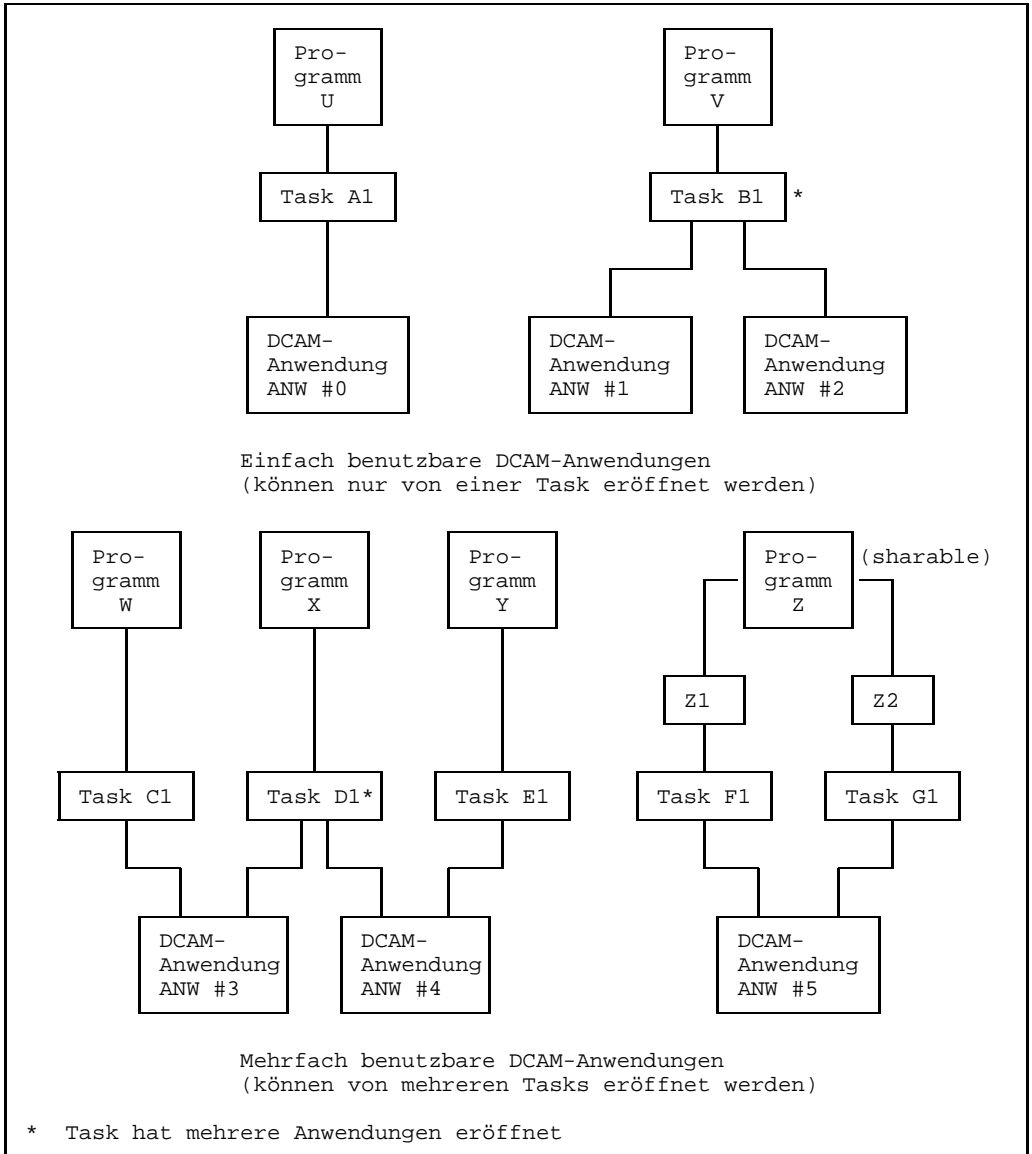


nur für DCAM(NEA)-Transport-Service-Anwendungen:

Auf die Existenz des Kommunikationspartners Datenstation hat der Programmierer keinen Einfluß. Dieser Kommunikationspartner wird entweder bei der Generierung des Kommunikationssystems oder durch Administrationsbefehle des Netzadministrators erzeugt und aufgelöst.

Eine **Datenstation** ist gekennzeichnet durch ihre technische Realisierung einerseits und die sie bedienende Person andererseits. Das Zusammenwirken von Gerät und Anwender des Geräts, ergeben zu einem jeweilig bestimmten Zeitpunkt den Kommunikationspartner Datenstation. Dieser kann nur mit den Mitteln des Datenkommunikationssystems Verbindung zu anderen Partnern unterhalten und über diese Verbindungen Daten senden und empfangen.

Im weiteren Sinn werden alle Teilnehmer des Datenkommunikationssystems Kommunikationspartner genannt, die Verbindungen zu anderen Partnern aufbauen können. Im engeren Sinne werden mit diesem Namen die beiden Partner einer bestehenden Verbindung bezeichnet.



Beispiele für DCAM-Anwendungen

2.3.3 Adressierung

Ein Partner wird mit zwei Namen adressiert: dem Namen des Prozessorknotens, in dem sich der Partner befindet und dem Namen des Kommunikationspartners selbst.

Der Name des Prozessorknotens und, für DCAM(NEA)-Transport-Service-Anwendungen der Name der Datenstation werden bei der Generierung des Datenkommunikationssystems festgelegt. Die Namen der DCAM-Anwendungen werden bei deren Eröffnung im Programm festgelegt.

Alle verwendeten Namen können maximal 8 Zeichen lang sein. Das erste Zeichen muß alphabetisch sein oder gleich \$, @, #. Das 2. bis 8. Zeichen kann zusätzlich aus den Ziffern 0...9 bestehen (Assembler-Namenskonvention). Der Name einer Anwendung des Anwenders darf nur dann mit dem Dollarzeichen '\$' beginnen, wenn die Anwendung unter TSOS abläuft.

Hinweis

Der Name des Kommunikationspartners wird bei ISO-Transport-Service-Verbindungen nicht auf Konventionen überprüft.

Bei DCAM(ISO)-Transport-Service-Anwendungen und heterogenen Netzen ist es u.U. notwendig, auch anders codierte bzw. längere Partnernamen anzugeben. Solche Namen erhalten über das Konsolkommando /BCMAP einen Aliasnamen, der im Programm angegeben wird und den Konventionen entspricht (siehe Handbuch 'Netzmanagement im BS2000').

2.3.4 Verbindungen

Bevor eine Kommunikation zwischen zwei Kommunikationspartnern durchgeführt werden kann, muß zwischen den beiden Partnern eine Verbindung aufgebaut werden. Ein Partner beginnt mit einer **Aufforderung zum Verbindungsaufbau (acquisition)**, der andere kann mit einer **Annahme** reagieren (**acceptance**) und bewirkt so den Verbindungsaufbau im Datenkommunikationssystem.

Auf einer aufgebauten Verbindung können Daten gesendet und empfangen werden. Bei DCAM(ISO)-Transport-Service-Anwendungen können zu einem Zeitpunkt zwischen zwei Anwendungen mehrere Verbindungen bestehen (Parallelverbindungen). Parallelverbindungen dürfen derzeit nur seriell aufgebaut werden.

2.4 Charakteristische Merkmale von DCAM

2.4.1 Verteilung ankommender Nachrichten

Die Verteilung von Nachrichten, die für eine DCAM-Anwendung eintreffen, kann über absenderspezifische oder empfängerglobale Warteschlangen erfolgen. Sie ist sowohl für einfach wie auch für mehrfach benutzbare DCAM-Anwendungen möglich.

Bei DCAM(NEA)-Transport-Service-Anwendungen kann die Verteilung auch über verteilcodespezifische Warteschlangen erfolgen, allerdings nur für mehrfach benutzbare DCAM-Anwendungen.

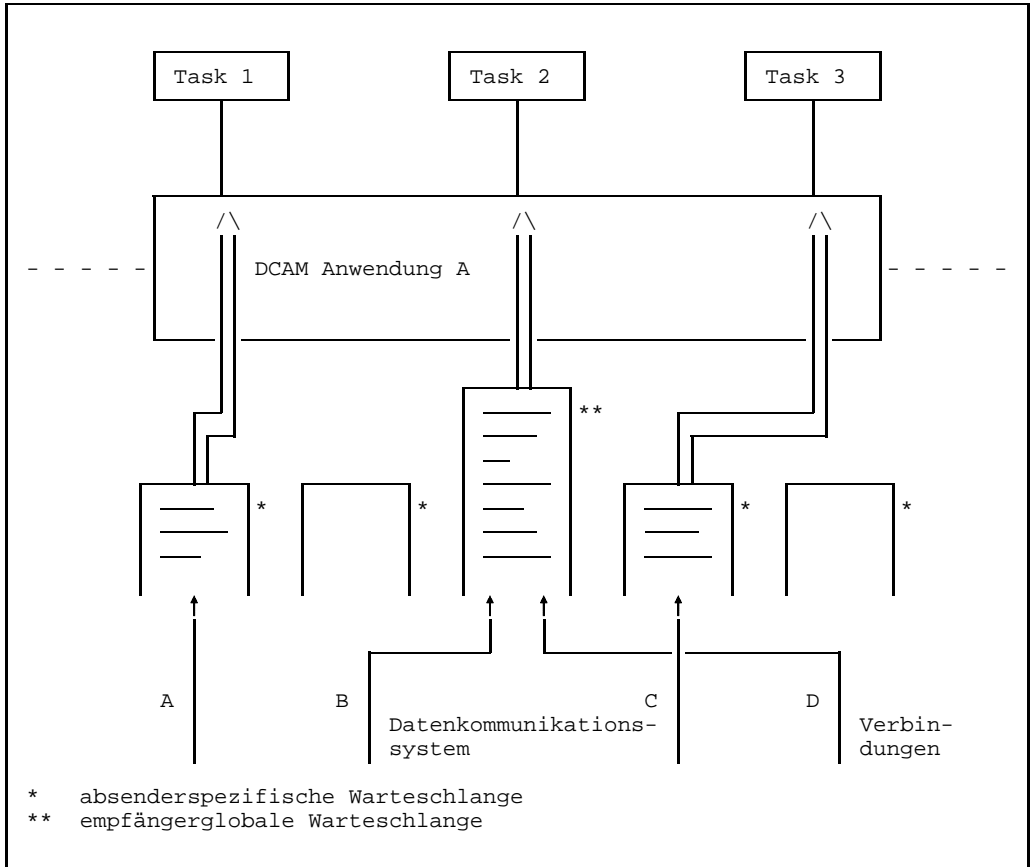
2.4.1.1 Absenderspezifische und empfängerglobale Warteschlange

Für jede aufgebaute Verbindung wird eine Warteschlange für angekommene Nachrichten angelegt, so daß absenderspezifisch auf die Nachrichten zugegriffen werden kann. Der Zugriff in der Reihenfolge des Eintreffens (unabhängig vom Absender der Nachricht) wird durch eine weitere, empfängerglobale Warteschlange realisiert. Eine Nachricht ist jeweils in eine der beiden Warteschlangen eingetragen. Den Zugriff auf die eine oder andere Warteschlange legt der Anwender vorher im Programm fest. Dies ist möglich beim Aufbau einer Verbindung und kann mit jedem Senden und Empfangen einer Nachricht für die folgende Verarbeitung neu festgelegt werden.

Auf diese Weise können über einen bestimmbaren Zeitraum nur Nachrichten über die absenderspezifische Warteschlange abgeholt werden. Bei mehrfach benutzbaren Anwendungen wird damit die Verknüpfung einer Task mit einer Verbindung hergestellt. Bei Umschaltung auf die empfängerglobale Warteschlange wird diese Verknüpfung wieder aufgelöst.

Einen Überblick gibt das folgende Bild.

Task 1 hat beim Aufbau der Verbindung, beim letztem Senden oder Empfangen festgelegt, daß sie Nachrichten von Verbindung A empfangen möchte. Sie greift nun mit Empfangsaufrufen auf diese Warteschlange zu. Das gleiche gilt für Task 3 und Verbindung C. Die Nachrichten der Verbindung B und D werden von Task 2 empfangen, können aber auch von den anderen Tasks empfangen werden.



Beispiel für Zugriff auf Warteschlangen

2.4.1.2 Verteilcodespezifische Warteschlange



Dieser Abschnitt gilt nur für DCAM(NEA)-Transport-Service-Anwendungen.

Es wird angenommen, daß die Taskgruppe von Programmen mit verschiedenen Fähigkeiten gesteuert wird, d.h., jedes der Programme kann nur bestimmte Nachrichten verarbeiten. Die Zuordnung Nachricht zu Programm (Task) erfolgt durch einen Code, der sich in der Nachricht selbst befindet. Der Anwender kann den Code wählen und legt beim Aufbau für die Verbindung bestimmte Codes fest (genaue Beschreibung der Möglichkeiten siehe Seite 69ff). Für jede Codegruppe wird eine eigene Warteschlange angelegt, die Zuordnung von Warteschlangen und Programm wird von der Primärtask gesteuert (hierzu stehen eigene Aufrufe zur Verfügung, siehe Seite 89ff). Bei der Eröffnung der Anwendung wird ein Verteilungsname definiert, der den Verteilcodes zugeordnet wird, um sie mit einer Task zu verknüpfen.

Der Name kann für mehrere Tasks gleich sein, wenn z.B. ein Programm mehrere Tasks steuert. DCAM gibt dann die Nachrichten nach dem 'fifo'-Prinzip (first in - first out: der erste Eintrag in der Warteschlange wird auch zuerst bearbeitet) an die verschiedenen Tasks weiter.

Gibt der Kommunikationspartner den Code an, der beim Verbindungsaufbau oder später definiert wurde, erreicht er die Task, deren Verteilungsname von der Primärtask diesem Code zugeordnet wurde.

Die Primärtask steuert demnach

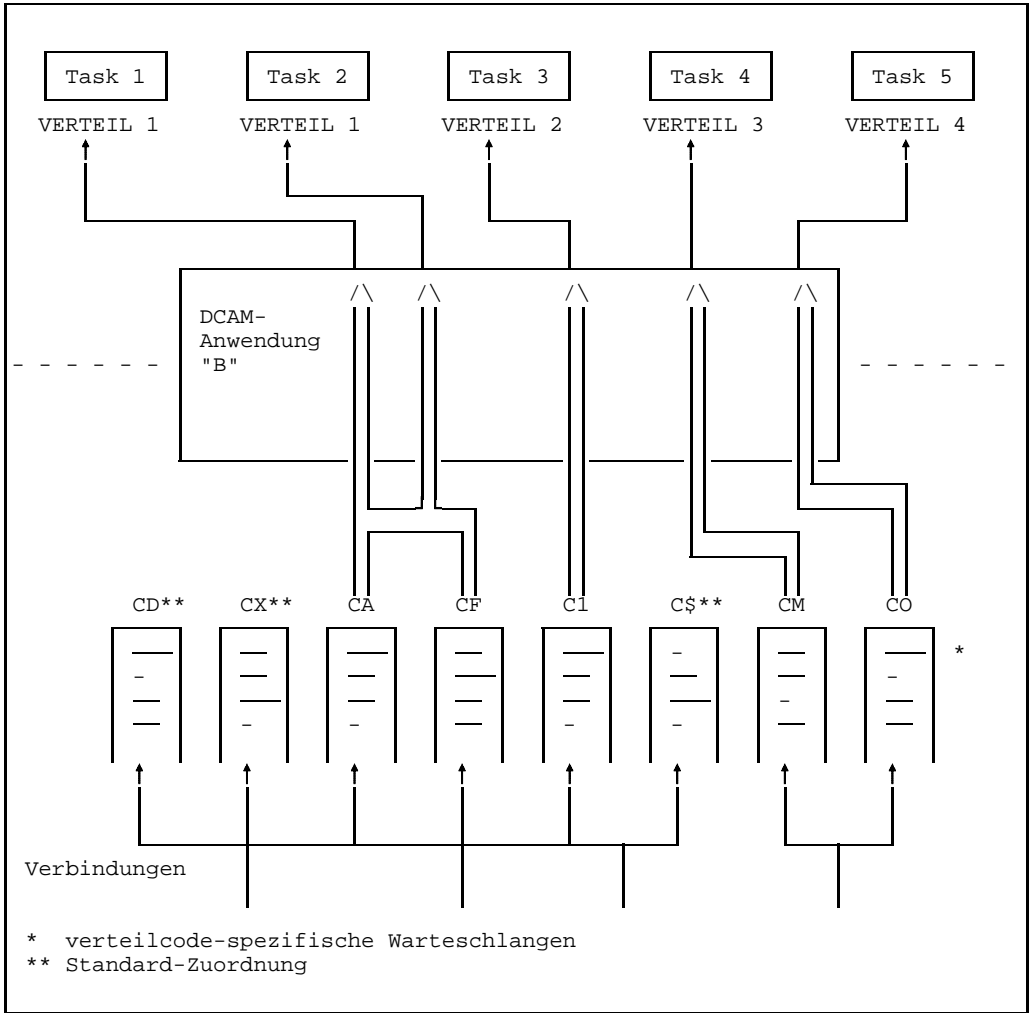
- die Zuordnung der Verteilcodes zu einer Verbindung.
Wenn sie eine Verbindung aufbaut, legt sie die verwendbaren Codes fest und kann jederzeit neue Codes mit einem eigenen Aufruf für die bestehende Verbindung definieren.
- die Zuordnung der Verteilcodes zu einer Task über den Verteilungsnamen.

Nachrichten, die nicht zugestellt werden können, weil ihr Code nicht zugeordnet oder ungültig ist, werden der Primärtask zugestellt.

Das folgende Bild zeigt an einem Beispiel Warteschlangen von Verteilcodegruppen, die nur einen Code enthalten.

Da Nachrichten mit gleichem Code von unterschiedlichen Partnern eintreffen, kann auch absenderspezifisch auf diese Nachrichten zugegriffen werden.

Der Verteilungsname 'VERTEIL1' ist sowohl der Task 1 (Primärtask), wie auch der Task 2 zugeordnet. Hier tritt das erwähnte 'fifo'-Prinzip ein: Nachrichten mit den Codes 'CA' und 'CF' werden in der Reihenfolge ihres Eintreffens jeweils der Task übergeben, die als nächste einen Empfangsaufruf abgibt. Die Nachrichten mit den Codes 'CD', 'CX' und 'C\$' werden Task 1 übergeben, da sie keiner anderen Task zugeordnet sind.



Beispiel für Zugriff auf Verteilcodespezifische Warteschlangen

2.4.1.3 Impliziter Verteilcode



Dieser Abschnitt gilt nur für DCAM(NEA)-Transport-Service-Anwendungen.

Soll eine Folge von Nachrichten den gleichen Verteilcode enthalten, genügt es, ihn nur in der ersten Nachricht anzugeben. Die Primärtask legt zu diesem Zweck ein **Codeanzeiger-Zeichen** fest, das anzeigt, daß in einer Nachricht oder einer Folge von Nachrichten ein Verteilcode enthalten ist. Der Verteilcode folgt unmittelbar auf dieses Zeichen. Er darf in diesem Fall höchstens 7 Zeichen lang sein.

Treffen Nachrichten ohne Verteilcode ein, werden sie entsprechend dem zuletzt gesendeten Verteilcode zugestellt. DCAM nimmt den zuletzt vorhandenen Verteilcode für die folgenden Nachrichten implizit an. Dies gilt solange, bis dieser Partner eine Nachricht schickt, in der er explizit einen Verteilcode angibt.

Wenn die Primärtask kein Codeanzeiger-Zeichen vereinbart hat, muß in jeder Nachricht ein Verteilcode stehen. Andernfalls werden sie der Primärtask zugestellt.

Das Gleiche gilt auch, wenn in einer Folge von Nachrichten nicht mindestens eine Nachricht mit einem Verteilcode beginnt.

2.4.2 Aufrufe und Meldungen

Die von DCAM angebotenen Funktionen bestehen aus Aufrufen, mit denen der BS2000-Anwender die Ausführung bestimmter Aktionen veranlassen kann und asynchronen Meldungen (siehe Seite 111ff), mit denen DCAM den Anwender über bestimmte Ereignisse im Kommunikationssystem informiert.

Die Aufrufe an DCAM beginnen alle mit dem Buchstaben Ypsilon. Sie werden entweder als **Makroaufrufe (Assembler)** oder als **'CALL'-Aufrufe (COBOL)** abgesetzt. Die Aufrufe werden nach ihrer Ausführung beendet oder, wenn eine festgesetzte Bearbeitungszeit abgelaufen ist (**synchroner Ausführung**). Außerdem besteht die Möglichkeit, die Steuerung unmittelbar nachdem der Aufruf abgesetzt wurde, zurückzuerhalten (**asynchrone Ausführung** des Befehls an DCAM). Den Abschluß des Befehls gibt eine **asynchrone Meldung** bekannt, die gezielt im Programm abgefragt werden kann. Ferner kann mit dem Eintreffen der Meldung eine eigens dafür definierte Routine angestoßen werden (Contingency Routine siehe Seite 105ff). Die asynchrone Verarbeitung dient insbesondere dazu, Wartezeiten zu nutzen.

2.4.3 Zugriffsschutz

Innerhalb des Verarbeitungsrechners kann eine DCAM-Anwendung gegen unberechtigten **Anschluß einer Task** geschützt werden. Soll die Anwendung einfach benutzbar sein, so ist kein Anschluß einer Sekundärtask möglich. Ist die Anwendung mehrfach benutzbar, kann mit einem Kennwort der unberechtigte Anschluß einer Sekundärtask verhindert werden.

Vordefinierte Anwendungen können schon bei der Systemgenerierung in der Netzdatei durch ein (Benutzungs-)Kennwort gegen unberechtigtes Eröffnen geschützt werden. Dieses RDF-Kennwort muß von der Primär- und Sekundärtask mitgegeben werden.

Innerhalb des Datenkommunikationssystems kann der unberechtigte **Aufbau einer Verbindung** verhindert werden. Einmal, indem DCAM-Anwendungen dauernd oder eine bestimmte Zeit keine Aufforderungen zum Aufbau einer Verbindung bearbeiten oder annehmen. Zum zweiten, indem bei der Aufforderung zum Aufbau einer Verbindung ein Kennwort angegeben werden muß. Außerdem kann der Anwender aufgrund einer Verbindungsnachricht und/oder der Adresse des auffordernden Partners entscheiden, ob er die Aufforderung annimmt oder zurückweist.

Bei den genannten Verfahren ist gewährleistet, daß irrtümlicher oder auch provozierter Fremdanschuß kontrolliert werden kann. Kennworte können zu diesem Zweck auch dynamisch verändert werden (siehe Seite 93ff).

2.4.4 Expreßnachrichten



Dieser Abschnitt gilt nur für DCAM(NEA)-Transport-Service-Anwendungen.

Ein anderer Aspekt der Datensicherheit, die Lösung von Konfliktfällen, ist auch berücksichtigt. Im Datenkommunikationssystem sorgt eine verbindungsbezogene Kapazitätsverteilung für die reibungslose Datenübermittlung. Die vorhandenen Zwischenspeicherbereiche werden genutzt, indem jede Verbindung nur einen bestimmten Teil belegen darf. Die Zwischenspeicher können so nicht vollständig durch die Daten einer Verbindung belegt werden. Damit aber bei Rückstaus, die eine Verbindung betreffen, diese noch bedienbar bleibt, können kurze Expreßnachrichten **fester Länge mit hoher Priorität** dem Empfänger zugestellt werden. Solche Nachrichten 'überholen' sozusagen die anderen. Wenn der Empfänger dies will, bekommt er die Expreßnachricht als asynchrone Meldung sofort übergeben (siehe Seite 111ff), ansonsten wird sie möglichst weit vorn in die Warteschlange eingetragen. Expreßnachrichten können nicht übermittelt werden, wenn die entsprechende Verbindung mit einer logischen Datenstation zusammenarbeitet.

Einschränkung:

COBOL-Anwender können Expreßnachrichten nicht als asynchrone Meldung empfangen.

2.4.5 Datenfluß-Steuerung

Das System kann eine überlastete Verbindung entlasten, indem es die Nachrichtenübermittlung vorübergehend unterbricht. Davon können Normalnachrichten und, bei DCAM(NEA)-Transport-Service-Anwendungen, auch Expreßnachrichten betroffen sein. Sobald die Verbindung wieder bedienbar ist, kann dem Anwender durch ein GO-Signal angezeigt werden, daß der Stau für normale bzw. für Expreßnachrichten behoben ist und daß erneut mit dem Senden von Nachrichten begonnen werden kann.

2.5 Programmaufbau

2.5.1 Aufgaben eines DCAM-Programms

DCAM-Programme realisieren Datenkommunikation im BS2000. Im Gegensatz zum Teilnehmerbetrieb ist die Aufgabenstellung und Realisierung abgeschlossen, wenn die Datenkommunikation gestartet wird. Ferner werden hier meist Aufgaben gelöst, die nur mit spezifischen Mitteln des BS2000 lösbar sind (siehe nachstehende Tabelle).

Aufgaben	Unterstützt im BS2000 durch:
Dialoge in einzelnen Schritten oder in einer Folge von Anfragen und Antworten	<ul style="list-style-type: none"> - Warteschlangen für Zugriff auf bestimmte oder beliebige Partner - Begleitinformation der Verbindung zur Übergabe mit der Nachricht (z.B. mit der Adresse eines Speicherbereichs); bei COBOL nicht anwendbar.
Datenübernahme von Programmen in eigenen oder in anderen Verarbeitungsrechnern des Datenkommunikationssystems	<ul style="list-style-type: none"> - Synchrone oder asynchrone Bearbeitung von DCAM-Aufrufen - Ereignisgesteuerte Verarbeitung mit EVENTING und CONTINGENCIES
Weitergabe von Daten an reine Datenverarbeitungsprogramme im eigenen Verarbeitungsrechner und Koordinierung der Zusammenarbeit	<ul style="list-style-type: none"> - Benutzung lokaler oder gemeinsamer Speicherbereiche mit COMMON MEMORY und Serialisierung - Koordinierung durch EVENTING, Inter-Prozess-Kommunikation oder Benutzer-Schalter (USER SWITCHES)
Sicherung gegen unbefugte Zugriffe auf Programme oder Programmsysteme, bzw. auf Dateien	<ul style="list-style-type: none"> - Kennwort-Schutz der DCAM-Anwendung - Kennwort-Schutz der Verbindung - Kennworte zum Schutz von Dateien gegen unberechtigtes Lesen, Schreiben und Ausführen
Die folgenden Punkte gelten nur für DCAM(NEA)-Transport-Service-Anwendungen.	
Dialog zwischen einzelnen wenigen Datenstationen und einem Programm	<ul style="list-style-type: none"> - Synchrone Bearbeitung von DCAM-Aufrufen - Verwaltung von Primärtasks (einfach verwendbare DCAM-Anwendungen)

Aufgaben	Unterstützt im BS2000 durch:
Bearbeitung von Anfragen einer großen Zahl von Datenstationen in einem Programmsystem	<ul style="list-style-type: none"> - Asynchrone Bearbeitung von DCAM-Aufrufen mit EVENTING und CONTINGENCIES - Verwaltung von Primär- und Sekundärtasks (mehrfach benutzbare DCAM-Anwendung) - Unterstützung der 'reentrant'-Programmierung
Aufbau von Nachrichten für zeilenweise Ausgabe oder Formatdarstellung auf Bildschirm	<ul style="list-style-type: none"> - Unterstützung durch Umsetzung von logischen Datenstationen in physikalische Eigenschaften einzelner Datenstationstypen (VTSU)
Ausnutzung spezifischer Datenstationseigenschaften	<ul style="list-style-type: none"> - Möglichkeit der 'physikalischen' Programmierung von Datenstationen
Bearbeitung von Anfragen in einer möglichst kurzen Zeit, um dem Datenstationsbenutzer günstige Antwortzeiten zu bieten	<ul style="list-style-type: none"> - Steuerung der Tasks im TIMESHARING - Schnelle Dateizugriffe mit USER PAM - Reentrant-Steuerung von Programm mit SHARED CODE verbunden mit günstiger Speicherauslastung und geringer Seitenwechselrate
Bearbeitung unterschiedlicher Anforderungen einer Datenstation in einem Programmsystem	<ul style="list-style-type: none"> - Nachrichtenverteilung über Verteilcode-spezifische Warteschlangen

Dem Programmierer stehen zur Verfügung:

Die Dienstleistungen der Schnittstellen von DCAM, TIAM und RBAM, des Organisationsprogramms und Datenverwaltungssystems (DVS).

Es ist zweckmäßig, DCAM-Programme zu unterteilen. Bei der SHARED CODE-Programmierung ist eine Gliederung in Befehls-Teil, Konstantenteil und dynamischen Arbeitsbereich gefordert. Dies ist auch zu empfehlen für die Dokumentation und spätere Wartung.

2.5.2 Grundstruktur eines DCAM-Programms

Ein DCAM-Programm setzt sich wie folgt zusammen:

- DCAM-Anwendung eröffnen
- Verbindung aufbauen
- Datenübermitteln
- Verbindung abbauen bzw. DCAM-Anwendung schließen

DCAM-Anwendung eröffnen

Damit ein Programm für andere Kommunikationspartner adressierbar wird, muß es mindestens eine DCAM-Anwendung eröffnen. Der Name, der dabei festgelegt wird, bildet zusammen mit den Namen des Prozessors (Prozessorknotens) die netzweit eindeutige Adresse dieser Anwendung. Es werden ferner Eigenschaften der DCAM-Anwendung bestimmt und es werden Vorgaben für Kennworte gemacht. Die Anweisung, die diesen Schritt ausführt, lautet YOPEN.

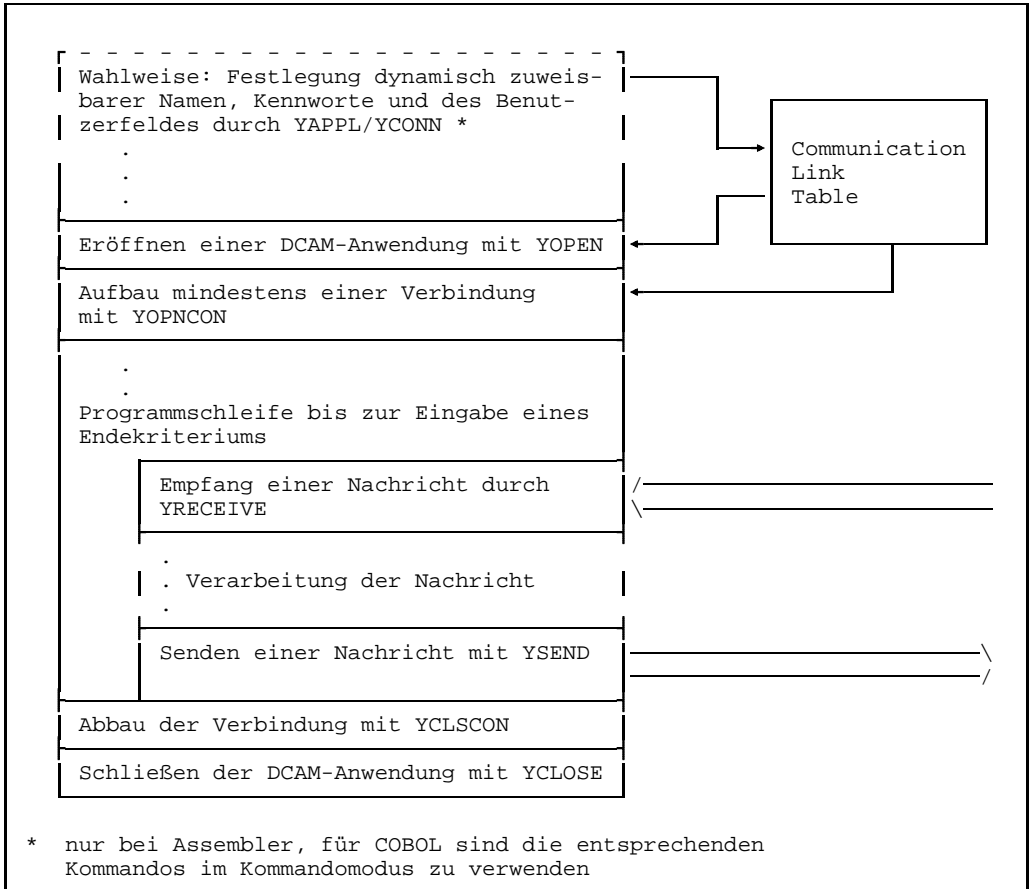
Verbindung aufbauen

Damit Datenübermittlung zwischen zwei Kommunikationspartnern erfolgen kann, muß vorher ein Partner die Aufforderung zum Aufbau einer Verbindung an den anderen richten und dieser muß sie annehmen. Damit wird die Verbindung aufgebaut. Zwischenspeicherung und Verteilung der Nachrichten erfolgt aufgrund der Angaben, die dabei festgelegt werden. Ferner ist anzugeben, welche Nachrichtenaufbereitung gewünscht wird, bzw. ob der Benutzer sie selbst übernimmt. Des weiteren können die Kommunikationspartner gegenseitig Verbindungsnachrichten austauschen. Die Anweisung für diesen Schritt lautet YOPNCON ACCEPT (Annahme Verbindungswunsch) oder YOPNCON ACQUIRE (Ausgabe Verbindungswunsch).

Nach diesen beiden vorbereitenden Schritten kann die Nachricht mit YRECEIVE **empfangen** bzw. mit YSEND **gesendet** werden.

Ist die Datenübermittlung beendet, wird die Verbindung abgebaut; entweder explizit vom Programm mit YCLSCON oder implizit mit der Schließung der DCAM-Anwendung durch die Anweisung YCLOSE, bzw. durch die Beendigung des Programms. Der Abbruch einer Verbindung, z.B. durch Anforderung des Kommunikationspartners, Ausfall einer Leitung oder eines Rechners, kann dem Programm gemeldet werden.

Für DCAM(NEA)-Transport-Service-Anwendungen kann eine Verbindung auch indirekt an der Datenstation durch die Eingabe eines vereinbarten Endekriteriums getrennt werden, das muß im Programm zur Ausführung des YCLSCON führen.



Prinzipieller Ablauf eines DCAM-Programms

Einfluß auf die Struktur des Programms kann auch die **dynamische Namen-Zuweisung** haben. Die dynamisch zugewiesenen Namen, Kennworte und das Benutzerfeld müssen vor Eröffnen der DCAM-Anwendung bzw. vor Aufbau der Verbindung in die Communication Link Table (CLT) eingetragen werden. In einem eigenen Programm oder in einer Vorlaufroutine sollten die Aufrufe YAPPL bzw. YCONN stehen, oder es muß dafür gesorgt werden, daß entsprechend die Kommandos /SET-DCAM-APPLICATION-LINK und /SET-DCAM-CONNECTION-LINK gegeben werden.

Einschränkung:

In COBOL-Programmen sind keine YAPPL- bzw. YCONN-Aufrufe möglich.

Nach jedem Aufruf an DCAM ist zu prüfen, ob der Aufruf richtig ausgeführt wurde. Dafür wird eine **Rückmeldung** (Feedback-Feld) zur Verfügung gestellt. Das Ergebnis der Prüfung kann sein, daß ein Aufruf wiederholt wird, in anderer Weise auszuführen ist, ggf. andere Maßnahmen zu treffen sind, oder aber auch der Programmablauf beendet werden muß. Die Fehlerbehandlungsroutine kann unmittelbar im Anschluß an den Aufruf codiert werden. Da meist dieselben oder ähnliche Maßnahmen zu treffen sind, ist es sinnvoll eine zentrale Fehlerbehandlungsroutine zu erstellen, die immer wieder aufgerufen werden kann.

2.5.3 Steuerung von Primär- und Sekundärtasks

Eine **Primärtask**, die als erste eine DCAM-Anwendung eröffnet, kann alle prinzipiellen DCAM-Funktionen enthalten. In einer Taskgruppe ist sie automatisch **die steuernde Task** und enthält die Verbindungsfunktion, Festlegungen von Kennworten, bei DCAM(NEA)-Transport-Service-Anwendungen die Verteilcodesteuerung, also alles, was die Taskgruppe als ganzes betrifft. Daß die Primärtask auch zeitlich die ersteröffnende ist, kann bei YOPEN mit VERIFY überprüft werden. Sie kann, wie auch die Sekundärtasks, Nachrichten senden und empfangen. Beim Schließen der DCAM-Anwendung in der Primärtask wird sie auch für die Sekundärtasks geschlossen. Der Aufbau des Programms, das die Primärtask steuert, ist von verschiedenen Faktoren abhängig, die in der nachfolgenden Tabelle zusammengestellt sind.

Fragen zum Aufbau eines Programms, welches eine Primärtask steuert	Maßnahmen bei bejahender Antwort der Frage
<ul style="list-style-type: none"> - Ist die zu bedienende Konfiguration klein? - Ist die mittlere Zeit, die verstreicht zwischen der Ankunft zweier Nachrichten groß (geringe Auslastung)? - Ist die Aufgabe in in ihrem Umfang so begrenzt, daß keine Aufteilung in unterschiedliche Bearbeitungsprogramme notwendig ist? 	<p>Die Primärtask führt die Bearbeitung alleine durch, es sind also keine Sekundärtasks notwendig und erlaubt (einfach verwendbare DCAM-Anwendung).</p>
<ul style="list-style-type: none"> - Ist die mittlere Zeit zwischen der Ankunft zweier Nachrichten klein? - Ist die Konfiguration groß? - Wird eine kurze Bearbeitungszeit gefordert? 	<p>Die Primär- und Sekundärtasks werden von einem SHARED CODE Programm gesteuert. Da die Primärtask spezifische Funktionen ausführen muß, ist dies im Programmablauf vorzusehen. Oder das Programm wird modular aufgebaut und es werden in den Tasks nur die spezifischen Moduln benutzt und geladen.</p>

Nur für DCAM(NEA)-Transport-Service-Anwendungen:

<ul style="list-style-type: none"> - Müssen die Anfragen, die über Verbindungen eintreffen unterschiedlicher Bearbeitung übergeben werden? - Wechseln die Verteilcodes in den Nachrichten? - Sollen aus Sicherheitsgründen Kennworte den Anschluß einer Task an eine DCAM-Anwendung schützen? - Sollen alle Daten überprüft werden? - Sollen weitere Steuerfunktionen realisiert werden? 	<p>Die Primärtask steuert die Verteilcode-Zuordnung zu den Sekundärtasks und damit die Nachrichtenverteilung. Sie legt ein Kennwort zum Anschluß von Sekundärtask fest. Sie überprüft die Nachrichten, die innerhalb der Gruppe nicht zustellbar sind. Im Rahmen von Common Memory, Eventing usw. stehen der Primärtask weitere Mittel zur Steuerung der Taskgruppe zur Verfügung.</p>
---	--

Eine **Sekundärtask**, die als zweite oder weitere Task eine DCAM-Anwendung eröffnet, hat auf die Eigenschaften der DCAM-Anwendung keinen Einfluß, ggf. muß sie ein Kennwort angeben, um sich an die Gruppe anzuschließen. Mit VERIFY kann sie überprüfen, ob sie zeitlich nach der Primärtask die Anwendung eröffnet. Die Verbindungen kann sie weder aufbauen noch abbauen. Innerhalb der Gruppe hat sie keine Steuerfunktion. Ihre Aufgabe ist das **Senden und Empfangen** von Nachrichten und deren Verarbeitung. Beim Aufbau des Programms ist daher nur von Bedeutung, in welcher Form YSEND-, YRECEIVE- oder YSENDREC-Aufrufe abgesetzt und bearbeitet werden. Zu beachten ist, daß ggf. Steuereingriffe der Primärtask berücksichtigt werden müssen, z.B. Verbindungssteuerung. Wird in der Sekundärtask die DCAM-Anwendung geschlossen, hat dies keinen Einfluß auf die anderen Tasks der Gruppe.

2.5.4 Nachrichten und lokale Dateneinheiten - More Data-Funktion

Die Gesamtmenge von Daten, die eine DCAM(ISO)-Transport-Service-Anwendung als logische Einheit über eine Verbindung zu einem Verbindungspartner schicken will, wird als **Nachricht** bezeichnet (auch TSDU = transport service data unit). Die Länge einer solchen Nachricht kann grundsätzlich beliebig sein.

Durch konkrete Voraussetzungen im Datenkommunikationsnetz (festgelegt in der Generierung des Datenkommunikationssystems) sowie durch die Realisierung des lokalen Transportsystems (Speicherverwaltung, Puffergrößen, usw.) kann mit einem lokalen Schnittstellenaufruf (YSEND, YRECEIVE) immer nur eine begrenzte Datenmenge auf einmal von der Anwendung an das Transportsystem übergeben werden. Dieser, bei einem Schnittstellenaufruf übergebene Datenpuffer, wird als **Dateneinheit** bezeichnet (auch TIDU = transport interface data unit).

DCAM bietet nun die Möglichkeit, für DCAM(ISO)-Transport-Service-Anwendungen große Nachrichten als logische Einheiten in mehreren Schnittstellenaufrufen, d.h., in mehreren Dateneinheiten zu übergeben. Diese Eigenschaft wird als **More Data-Funktion** bezeichnet.

Eine DCAM(ISO)-Transport-Service-Anwendung kann beim Aufbau einer Verbindung angeben, ob auf dieser Verbindung die More Data-Funktion genutzt werden soll oder nicht.

Hinweis

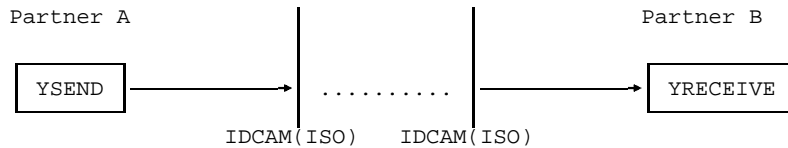
- Die More Data-Funktion wird nur lokal an DCAM des eigenen Rechners eingestellt. Sie wird dem Verbindungspartner weder mitgeteilt noch mit ihm ausgehandelt.
- Die More Data-Funktion bestimmt die Dateneinheiten, wie sie an der eigenen lokalen DCAM-Schnittstelle übergeben werden. Sie sagt nichts aus über die Aufteilung der "physikalischen" Datenblöcke auf der Übertragungsstrecke zum fernen Transportsystem.
- Die Verwendung der More Data-Funktion in einer Anwendung sagt nichts darüber aus, in welchen Dateneinheiten der Verbindungspartner die (logischen) Nachrichten sendet bzw. empfängt.

An einigen Beispielen soll die mögliche Wirkung der More Data-Funktion erläutert werden.

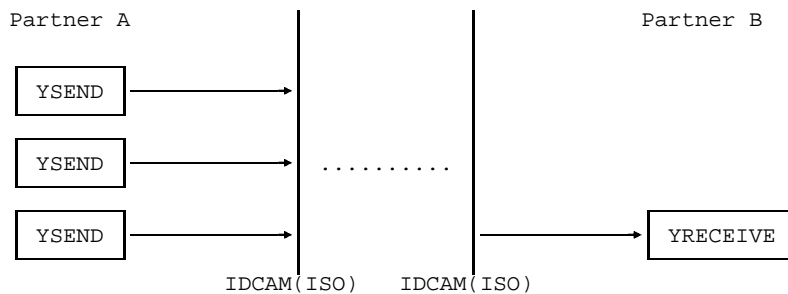
Es wird angenommen, daß Partner A und Partner B jeweils DCAM(ISO)-Transport-Service-Anwendungen sind. Je Beispiel wird genau eine (logische) Nachricht betrachtet, die von A nach B gesendet wird.

Beispiel 1

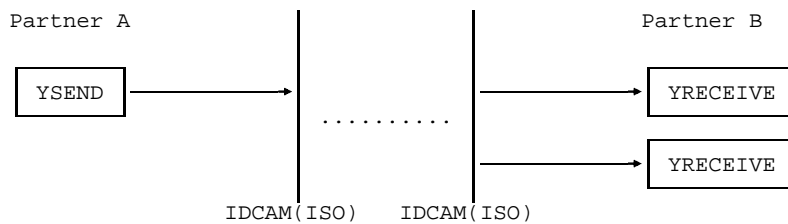
- Partner A ohne More Data-Funktion
- Partner B ohne More Data-Funktion

**Beispiel 2**

- Partner A mit More Data-Funktion
- Partner B ohne More Data-Funktion

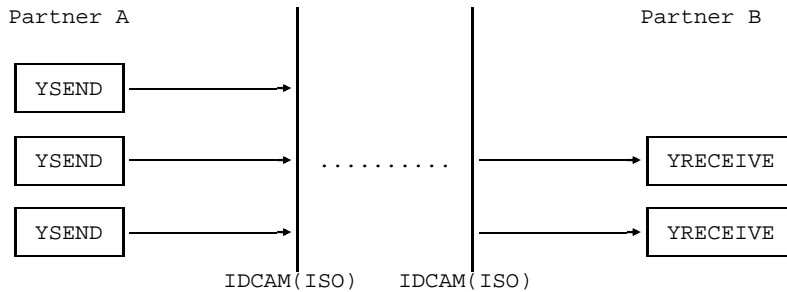
**Beispiel 3**

- Partner A ohne More Data-Funktion
- Partner B mit More Data-Funktion



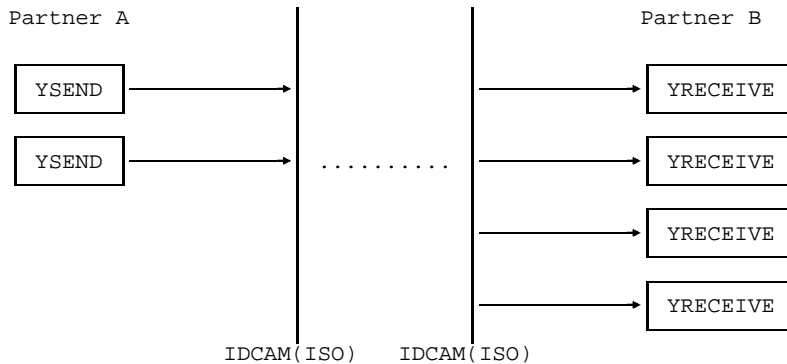
Beispiel 4

- Partner A mit More Data-Funktion
- Partner B mit More Data-Funktion



Beispiel 5

- Partner A mit More Data-Funktion
- Partner B mit More Data-Funktion



Die Beispiele 4 und 5 zeigen deutlich, daß die Aufteilung der Dateneinheiten beim Empfänger nicht unbedingt identisch ist mit der Aufteilung beim Absender.

Hinweis

In den DCAM-Handbüchern wird nur dann explizit zwischen Nachricht und Dateneinheit unterschieden, wo es für das Verständnis notwendig erscheint.

Wo eine Unterscheidung nicht wichtig ist oder die Bedeutung aus dem Kontext der Beschreibung deutlich hervorgeht, wird auf eine Formulierung "Nachricht bzw. Dateneinheit" verzichtet und im Allgemeinen nur von "Nachricht" gesprochen.

2.5.5 Zugriff zu Datenstationen



Dieser Abschnitt gilt nur für DCAM(NEA)-Transport-Service-Anwendungen.

Für die Planung von DCAM-Programmen ist von Bedeutung, wie die Zugriffe auf Datenstationen ausgeführt werden sollen.

Dem DCAM-Programmierer stehen zwei Methoden zur Auswahl:

- 'physikalische' Programmierung oder
- Benutzung der logischen Datenstationen.

Die '**physikalische**' **Programmierung** erfordert die Bedienung einer Datenstation mit den Steuerzeichen, die sie versteht (siehe die Benutzerhandbücher der einzelnen Datenstationen). Damit kann er zwar sehr viele Möglichkeiten nutzen, hat aber die Mühe der Bedienung mit Steuerzeichen. Die logischen Datenstationen nehmen ihm diese Arbeit ab. Alle wesentlichen Funktionen sind über diese standardisierten Datenstationen erreichbar (nicht bei lokal angeschlossenen Datenstationen).

Die **logischen Datenstationen** sind im Datenkommunikationssystem Software-Bausteine des Benutzer-Services. Bereits beim Aufbau einer Verbindung stimmen sich die Kommunikationspartner darüber ab, welche logische Datenstation benutzt werden soll. Dazu ist beispielsweise zu klären, ob ein Hardcopy-Gerät an einer Datenstation angeschlossen ist oder ob ein Ausweisleser benutzt werden soll.

Der Programmierer hat zwei Arten von logischer Datenstation zur Auswahl: Die Zeilen-Datenstation und die Format-Datenstation, siehe Kapitel 'Unterstützung logischer Datenstationen' Seite 95 und Handbuch 'VTSU'.

2.6 Realisierung des Verarbeitungsverbundes

Das Kommunikationssystem ermöglicht die Kommunikation zwischen

- Anwendungen und Anwendungen
- und zusätzlich für DCAM(NEA)-Transport-Service-Anwendungen zwischen Datenstationen und Anwendungen

Dabei können sich sowohl Datenstationen wie auch Anwendungen in einem LAN oder WAN-Netz befinden.

Abgesehen von der Verbindung mit Datenstationen ergeben sich daraus für eine DCAM-Anwendung z.B folgende Möglichkeiten: Der Partner ist

- eine DCAM-Anwendung
- eine UTM-Anwendung
- eine CMX-Anwendung
- eine Systemanwendung (z.B. \$DIALOG, \$CONSOLE)

2.6.1 DCAM-Anwendung als Partner

Der Kommunikationspartner DCAM-Anwendung wird im Abschnitt 'Funktionen von DCAM ab Seite 41 beschrieben. Dieser Abschnitt beschreibt die Existenz einer DCAM-Anwendung (Seite 41ff), den Aufbau einer Verbindung zwischen DCAM-Anwendungen (Seite 51ff) und die Datenübermittlung zwischen DCAM-Anwendungen (Seite 78ff).

2.6.2 UTM-Anwendung als Partner

Die Kommunikation zwischen einer DCAM-Anwendung und einer UTM-Anwendung ist immer dann möglich, wenn die DCAM-Anwendung für die UTM-Anwendung als Kommunikationspartner generiert ist und eine Verbindung zwischen beiden Anwendungen aufgebaut ist. Es spielt keine Rolle, ob sich die Anwendungen im gleichen oder in verschiedenen Verarbeitungsrechnern befinden.

Der Verbindungsaufbau zwischen einer DCAM-Anwendung und einer UTM-Anwendung ist auf 3 Arten möglich:

- Die UTM-Anwendung wurde so generiert, daß bei ihrem Start die Verbindung aufgebaut werden soll. Sie ist aufgebaut, wenn die DCAM-Anwendung zu diesem Zeitpunkt existiert und den Verbindungswunsch explizit akzeptiert.
- Die UTM-Anwendung wurde so generiert, daß Verbindungswünsche der DCAM Anwendung angenommen werden. Die Verbindung ist aufgebaut, wenn beide Anwendungen existieren und die DCAM-Anwendung einen expliziten Verbindungswunsch ausgegeben hat.
- Mit einem UTM-Administrationskommando kann veranlaßt werden, daß ein Verbindungswunsch an eine DCAM-Anwendung ausgegeben wird.

Bei der Datenübermittlung ist zu beachten, daß die UTM-Anwendung eine Struktur besitzt. Die UTM-Anwendung besteht aus Teilprogrammen, die über einen ihnen zugeordneten Transaktionscode adressiert werden können. Beim Beginn eines Vorgangs müssen die Daten am Anfang diesen Transaktionscode enthalten.

3 Funktionen von DCAM

Die Funktionen, die DCAM bietet, lassen sich in 4 Gruppen einteilen:

- Existenzfunktion
- Verbindungsfunktion
- Datenübermittlungsfunktion
- Namen-Zuweisungsfunktion

Im folgenden werden diese Funktionen beschrieben. Die weitere Aufteilung entspricht der Aufteilung des Abschnittes 'Anwendung der Funktionen von DCAM' im Assembler-Benutzerhandbuch (siehe DCAM-Makroaufrufe) und dem COBOL-Benutzerhandbuch (siehe DCAM-COBOL-Aufrufe). Dort sind die entsprechenden Aufrufe mit den erforderlichen Parametern dargestellt.

3.1 Existenzfunktion

Die Existenzfunktion der DCAM-Schnittstelle führt aus:

- **DCAM-Anwendung eröffnen bzw. erzeugen (YOPEN).**
Dies ist die Grundfunktion von DCAM, die je nach Art der DCAM-Anwendung unterschiedlich ausgeführt wird.
- **Zustand einer DCAM-Anwendung abfragen (YINQUIRE).**
Mit dieser Funktion können Sie den Zustand einer DCAM-Anwendung abfragen.
- **DCAM-Anwendung schließen (YCLOSE).**
Zwar wird die DCAM-Anwendung bei Programmende automatisch geschlossen, mit dieser Funktion kann sie aber auch jederzeit im Programmablauf geschlossen werden.



und zusätzlich für DCAM(NEA)-Transport-Service-Anwendungen:

Zustand einer DCAM-Anwendung verändern (YSETLOG).

Mit dieser Funktion kann der Zustand der DCAM-Anwendung - die Bereitschaft, Aufforderungen zum Verbindungsaufbau zu bearbeiten dynamisch verändert werden.

3.1.1 DCAM-Anwendung eröffnen

Eine DCAM-Anwendung wird erzeugt, wenn das erste Programm sie eröffnet. Dabei wird festgelegt, ob diese Anwendung einfach oder mehrfach benutzbar sein soll. Eine einfach benutzbare Anwendung kann nur in einem Programm eröffnet werden und wird auch durch dieses wieder geschlossen. Eine mehrfach benutzbare Anwendung kann auch von weiteren Programmen (Sekundärtasks) eröffnet werden, die sich allerdings nur anschließen, auf die Festlegungen des ersteröffnenden/erzeugenden Programms (Primärtask) haben sie keinen Einfluß. Für DCAM(NEA)-Transport-Service-Anwendungen ist weiterhin anzugeben, ob die Nachrichtenverteilung über die verteilcodespezifischen Warteschlangen erfolgen soll. Daraus geben sich unterschiedliche Varianten des Aufrufs YOPEN.

Bei allen DCAM(ISO)-Transport-Service-Anwendungen muß im Anwendungssteuerblock das ISO-Attribut gesetzt werden. Dieses Attribut gilt dann grundsätzlich für alle Verbindungen, die diese Anwendung unterhält. Verbindungen gemäß NEA-Dienst sind für eine DCAM(ISO)-Transport-Service-Anwendung nicht möglich.

3.1.1.1 Einfach benutzbare DCAM-Anwendung

Eine DCAM-Anwendung, die einfach benutzbar sein soll (NSHARE), kann nur von einer Task eröffnet werden. Jeder weitere Versuch, diese Anwendung zu eröffnen, wird zurückgewiesen. Im Rahmen dieser Einschränkung sind weitere Festlegungen sinnvoll zu treffen, abhängig von verschiedenen Voraussetzungen, die in der folgenden Tabelle zusammengefaßt sind.

Voraussetzung	Festlegung
Der Name der DCAM-Anwendung soll nicht von der Kommunikationszugriffsmethode generiert werden.	Name der DCAM-Anwendung; bei COBOL muß der Name der DCAM-Anwendung immer vom Anwender angegeben werden
Die Anwendung ist durch ein Kennwort in der RDF (APPPW=) geschützt.	Kennwort USEPW
Es sollen asynchrone Meldungen angenommen werden.	Verweis auf das Kennzeichen einer Contingency-Routine, das bei Angabe eines ENACO-Aufrufs erzeugt wurde. Wird die Meldung gegeben, wird diese Routine zum Ablauf gebracht (für COBOL nicht notwendig).
Es sollen bestimmte Funktionen, die ab DCAM-Version 8.0 verfügbar sind, benutzt werden.	Angabe der DCAM-Versionsnummer (DCAMVER).
Die folgenden Punkte gelten nur für DCAM(NEA)-Transport-Service-Anwendungen:	
Es muß geklärt sein, ob die DCAM-Anwendung nur Aufforderungen zum Verbindungsaufbau ausgibt. Voraussetzung dafür ist, daß alle möglichen Partner entweder Datenstationen sind oder DCAM-Anwendungen, die Aufforderungen bearbeiten.	Attribut NLOGON: Aufforderungen zum Verbindungsaufbau werden nicht bearbeitet.
Im Programm ist nicht bekannt, ob und wann ein Partner sich anschließen will, bzw. der Partner selbst ist nicht bekannt.	Attribut LOGON: Aufforderungen zum Verbindungsaufbau werden bearbeitet.
Unbefugte Aufforderungen zum Aufbau einer Verbindung sollen automatisch zurückgewiesen werden.	Kennwort (LOGPASS), das dann von den Kommunikationspartnern angegeben werden muß.

3.1.1.2 Erstmaliges Eröffnen einer mehrfach benutzbaren DCAM-Anwendung

Eine mehrfach benutzbare DCAM-Anwendung kann in mehreren Tasks eröffnet werden. Die erste Task, die die DCAM-Anwendung eröffnet, ist die Primärtask. Die Nachrichtenverteilung erfolgt über die absenderspezifischen bzw. empfängerglobalen Warteschlangen. Der Name der DCAM-Anwendung **muß** im Programm festgelegt werden. Weitere Festlegungen sind in der folgenden Tabelle zusammengefaßt.

Voraussetzung	Festlegung
Die Task soll in jedem Fall die erst-eröffnende sein, weil sie Funktionen einer Primärtask erfüllen soll. Ist sie es nicht, soll der YOPEN zurückgewiesen werden.	Überprüfung mit VERIFY = PRIMARY.
Es ist sichergestellt, daß die Task die ersteröffnende ist, oder alle sich anschließenden Programme haben gleiche Verarbeitungslogik.	Keine Überprüfung, also VERIFY = NO.
Unbefugtes Anschließen von Sekundär-tasks kann nicht ausgeschlossen werden, daher soll es verhindert werden.	Kennwort USEPASS, das dann von den Sekundärtasks beim nachfolgenden Eröffnen angegeben werden muß. <i>Hinweis</i> Ist die DCAM-Anwendung bereits durch ein Kennwort (APPW=) in der Netzdatei RDF geschützt, muß das USEPASS-Kennwort ggf. mit dem APPW-Kennwort übereinstimmen. Der Parameter USEPW wird dann auch bei der Primärtask ausgewertet; er muß das RDF-Kennwort enthalten.
Es sollen asynchrone Meldungen angenommen werden.	Verweis auf das Kennzeichen einer Contingency-Routine, das bei Abgabe eines ENACO-Aufrufs erzeugt wurde. Wird die Meldung gegeben, wird diese Routine zum Ablauf gebracht (für COBOL nicht notwendig).

Voraussetzung	Festlegung
<p>Die folgenden Punkte gelten nur für DCAM(NEA)-Transport-Service-Anwendungen:</p> <p>Es muß geklärt sein, ob die DCAM-Anwendung nur Aufforderungen zum Verbindungsaufbau ausgibt. Voraussetzung dafür ist, daß alle möglichen Verbindungsaufbau ausgibt. Voraussetzung dafür ist, daß alle möglichen Partner entweder Datenstationen sind oder DCAM-Anwendungen, die Aufforderungen bearbeiten.</p>	<p>Attribut NLOGON: Aufforderungen zum Verbindungsaufbau werden nicht bearbeitet.</p> <p>werden nicht bearbeitet.</p>
<p>Im Programm ist nicht bekannt, ob und wann ein Partner sich anschließen will, bzw. der Partner selbst ist nicht bekannt.</p>	<p>Attribut LOGON: Aufforderungen zum Verbindungsaufbau werden bearbeitet.</p>
<p>Unbefugte Aufforderungen zum Aufbau einer Verbindung sollen automatisch zurückgewiesen werden.</p>	<p>Kennwort (LOGPASS), das dann von den Kommunikationspartnern angegeben werden muß.</p> <p><i>Hinweis</i> LOGPASS und USEPASS können in einer bestehenden DCAM-Anwendung nicht geändert werden.</p>
<p>Transportquittungen für diese Anwendung werden generell in der Primärtask bearbeitet.</p>	<p>Eigenschaft PRIMTASK für die Zustellung von Transportquittungen.</p>
<p>Transportquittungen werden von der Task bearbeitet, die unter der entsprechenden Laufnummer (SEQNO) die Nachricht abgesendet hat und dabei eine Quittung angefordert hat (OPTCD=TACK).</p>	<p>Eigenschaft REQTASK für die Zustellung von Transportquittungen.</p>
<p>Diese Anwendung verarbeitet grundsätzlich keine Transportquittungen, weil anwenderspezifische Sicherungsverfahren eingesetzt werden sollen.</p>	<p>Eigenschaft NOTACK, keine Zustellung von Transportquittungen, weder positiven noch negativen.</p>
<p>Es sollen bestimmte Funktionen einer DCAM-Version (ab 8.0) benutzt werden.</p>	<p>Angabe der DCAM-Versionsnummer (DCAMVER).</p>

3.1.1.3 Erstmaliges Eröffnen - Verwendung von Verteilcodes



Dieser Abschnitt gilt nur für DCAM(NEA)-Transport-Service-Anwendungen.

Die Primärtask, die eine DCAM-Anwendung eröffnet, kann wahlweise festlegen, daß anstelle des Standardverfahrens mit der Nachrichtenverteilung anhand von Verteilcodes gearbeitet wird (SHARE,DISCO). Dies ist insbesondere dann günstig, wenn die beteiligten Programme unterschiedliche Aufgaben erfüllen, aber von einer gemeinsamen Anzahl von Partnern angesteuert werden. Zu diesem Zweck müssen von allen beteiligten Tasks Verteilungsnamen definiert werden, die eine Zuordnung von Verteilcode(s) und Task erlauben (DISNAME). Für weitere Festlegungen gilt ebenfalls die vorhergehende Tabelle.

3.1.1.4 Nachfolgendes Eröffnen

Eine Task, die als nachfolgende eine DCAM-Anwendung eröffnet, muß sich in den Festlegungen nach der Primärtask richten. Sie muß den dort festgelegten Namen der DCAM-Anwendung verwenden und ebenfalls ATTR=SHARE setzen (mehrfach verwendbare DCAM-Anwendung). Im Standardfall wird sie mit Nachrichtenverteilung über die absenderspezifische oder empfängerglobale Warteschlange arbeiten. Weitere Festlegungen sind in der nachfolgenden Tabelle zusammengestellt.

Voraussetzung	Festlegungen
Die Primärtask hat zum Schutz gegen unbefugten Anschluß ein Kennwort festgelegt oder die Anwendung ist durch ein RDF-Kennwort geschützt.	Kennwort (USEPW) zum Anschluß an die DCAM-Anwendung, wie in der Primärtask oder in der RDF angegeben.
Die Task muß in jedem Fall die nachfolgend eröffnende sein, denn in der Programmlogik ist dies so vorgesehen, ggf. soll der YOPEN zurückgewiesen werden.	Überprüfung mit VERIFY = SECONDARY.

Voraussetzung	Festlegungen
Es ist sichergestellt, daß die Task die nachfolgend eröffnende ist, bzw. die Programmlogik prüft diese Voraussetzung.	Keine Überprüfung, also VERIFY = NO.
Es sollen asynchrone Meldungen angenommen werden. <i>Einschränkung:</i> An der COBOL-Schnittstelle gibt es keine asynchronen Meldungen, die an die Sekundärtask gehen.	Verweis auf das Kennzeichen einer Contingency-Routine, das bei Abgabe eines ENACO-Aufrufs erzeugt wurde. Wird die Meldung gegeben, wird diese Routine zum Ablauf gebracht.
Nur für DCAM(NEA)-Transport-Service-Anwendungen:	
Die Primärtask hat beim Eröffnen der Anwendung eine DCAM-Versionsnummer angegeben.	Angabe der gleichen DCAM-Versionsnummer wie bei der Primärtask (DCAMVER).

3.1.1.5 Nachfolgendes Eröffnen - Verwendung von Verteilcodes



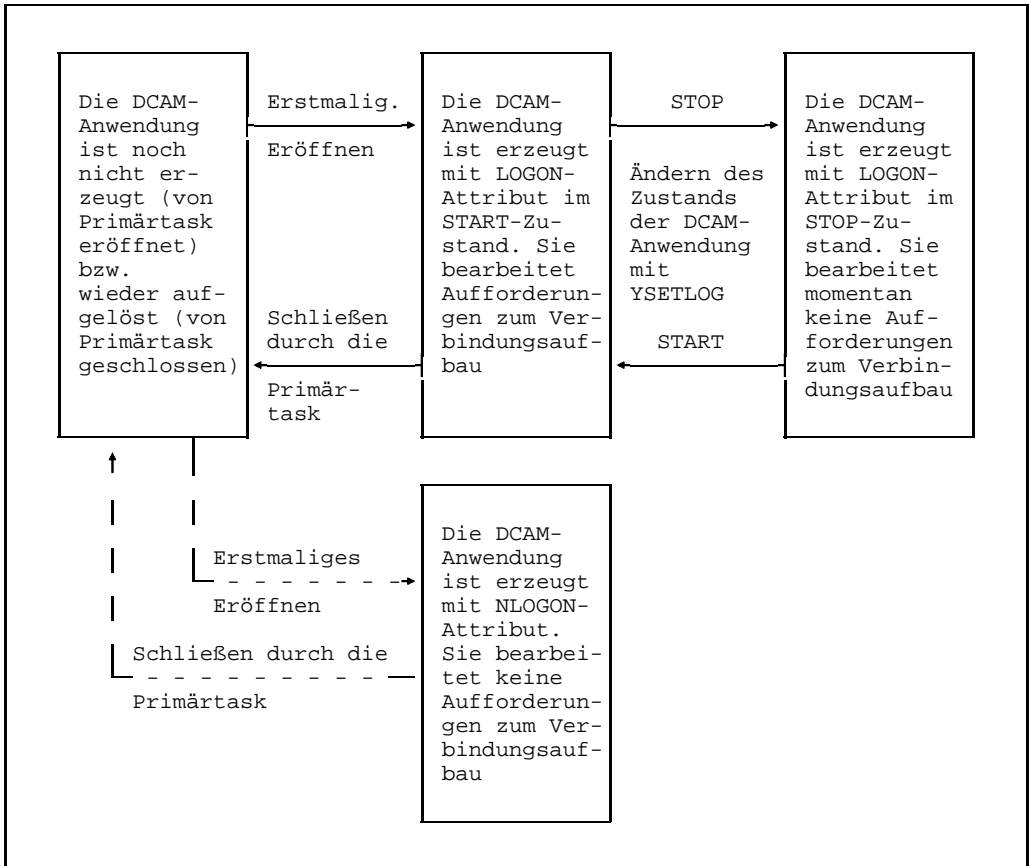
Dieser Abschnitt gilt nur für DCAM(NEA)-Transport-Service-Anwendungen.

Bei einem nachfolgenden Eröffnen der DCAM-Anwendung muß auch die Nachrichtenverteilung nach der ersteröffnenden Task (Primärtask) ausgerichtet werden. Sie ist also stets für die Taskgruppe einheitlich. Bei dieser Variante wird mit Verteilcodes gearbeitet. In der Primärtask werden die Attribute SHARE und DISCO angegeben, in der Sekundärtask muß der gleiche Name der Anwendung und das Attribut SHARE angegeben werden. Zusätzlich muß ein Name (DISNAME) für die Verteilcode-Zuordnung definiert werden. Alle weiteren Festlegungen sind der vorhergehenden Tabelle zu entnehmen.

3.1.2 Zustand einer DCAM-Anwendung verändern



Dieser Abschnitt gilt nur für DCAM(NEA)-Transport-Service-Anwendungen.



Zustände der DCAM-Anwendung

DCAM-Anwendungen, die Aufforderungen zum Verbindungsaufbau bearbeiten (Attribut LOGON), können zwei Zustände annehmen, den START- oder den STOP-Zustand. Im START-Zustand werden die Aufforderungen der DCAM-Anwendung zugestellt oder, wenn mehrere Anforderungen eintreffen, in eine Warteschlange eingereiht.

Im STOP-Zustand hat die Primärtask mit dem Aufruf YSETLOG mitgeteilt, daß sie Aufforderungen zum Verbindungsaufbau nicht mehr bearbeitet. Gründe dafür können sein, daß die maximale Anzahl zulässiger Verbindungen erreicht ist oder, daß die Primärtask für eine bestimmte Zeit selbst Aufforderungen ausgeben will. Die Primärtask kann allerdings jederzeit wieder den START-Zustand herstellen, d.h. Aufforderungen zum Verbindungsaufbau werden nicht mehr zurückgewiesen.

DCAM-Anwendungen, die keine Aufforderungen zum Verbindungsaufbau bearbeiten (Attribut NLOGON), können nicht mehr verändert werden.

3.1.3 Zustand einer DCAM-Anwendung abfragen

Mit dem Aufruf YINQUIRE der Funktion 'APPSTAT' hat der Anwender die Möglichkeit, sowohl den Zustand der DCAM-Anwendung, die er selbst eröffnet hat, abzufragen, wie auch den Zustand einer DCAM-Anwendung, die im selben Verarbeitungsrechner eröffnet wurde und deren Name er kennt.

3.1.4 DCAM-Anwendung schließen

Die DCAM-Anwendung kann auf zwei Arten geschlossen werden:

- implizit durch Beenden des Programms, in dem die DCAM-Anwendung eröffnet wurde (normales oder fehlerbedingtes Ende), bzw. Abbruch der Task
- explizit durch den Aufruf YCLOSE, bzw. durch Kommandos /SHUTDOWN, /BCEND oder /BCAPPL des Operators.

Das explizite Schließen durch YCLOSE ist dann erforderlich, wenn z.B. in einem Programm nacheinander mit mehreren DCAM-Anwendungen gearbeitet wird, oder, wenn während des Ablaufs Festlegungen geändert werden. Nach dem Schließen kann die Anwendung mit neuen Festlegungen (Namen, Eigenschaften, usw.) durch die Primär-task wieder eröffnet werden.

Die **Sekundärtask** kann jederzeit die Anwendung schließen, ohne weitere Folgen für die übrigen Tasks der Gruppe.

Schließt die **Primärtask** die Anwendung, bewirkt das, daß

- die DCAM-Anwendung aufgelöst wird.
- Sekundärtasks unterrichtet werden durch Anstoßen der COMEND-Contingency-Routine (siehe auch Seite 111ff) oder durch eine entsprechende Rückmeldung beim nächsten Aufruf oder bei einem noch nicht beendeten Befehl.
- alle bestehenden Verbindungen abgebaut werden.

Dies bedeutet auch, daß bereits angekommene, aber noch nicht übergebene Daten nicht mehr verfügbar sind. Bereits wartende Aufforderungen zum Verbindungsaufbau werden gelöscht.

3.2 Verbindungsfunktion

Voraussetzung für die Datenübermittlung ist der Aufbau einer Verbindung zwischen den Kommunikationspartnern, nachdem eine DCAM-Anwendung eröffnet wurde.

Die Verbindungsfunktion führt aus:

- Verbindung aufbauen (YOPNCON)
- Einträge über Partner und Verbindungen abfragen (YINQUIRE)
- Aufforderung zum Verbindungsaufbau zurückweisen (YREJLOG)
- Aufforderung zurücknehmen (YCLSCON)
- Verbindung abbauen (YCLSCON)
- Eigenschaften einer Verbindung ändern (YCHANGE)

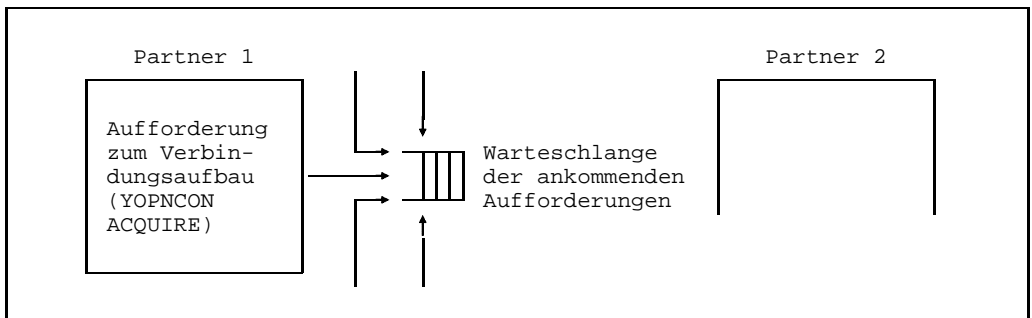
3.2.1 Verbindung aufbauen YOPNCON

Um eine Verbindung aufzubauen sind stets **zwei Schritte** notwendig:

- Ein Partner richtet an einen anderen eine Aufforderung.
- Der andere Partner nimmt die Aufforderung an.

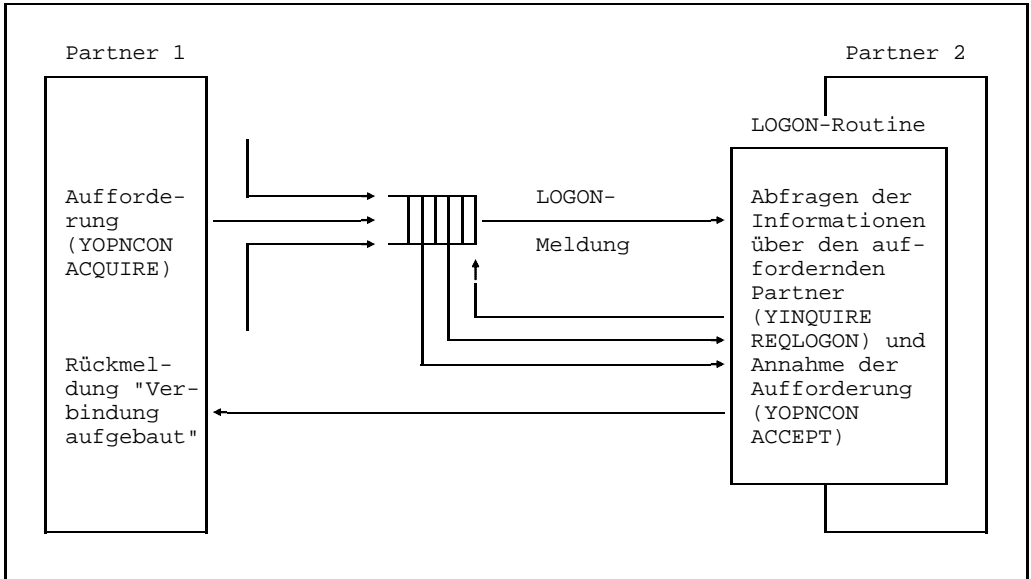
Die Koordinierung dieser Schritte erfolgt durch das System:

- Die **Aufforderungen** werden dem Partner zugestellt oder, wenn noch andere Aufforderungen vorliegen, in eine Warteschlange eingetragen.



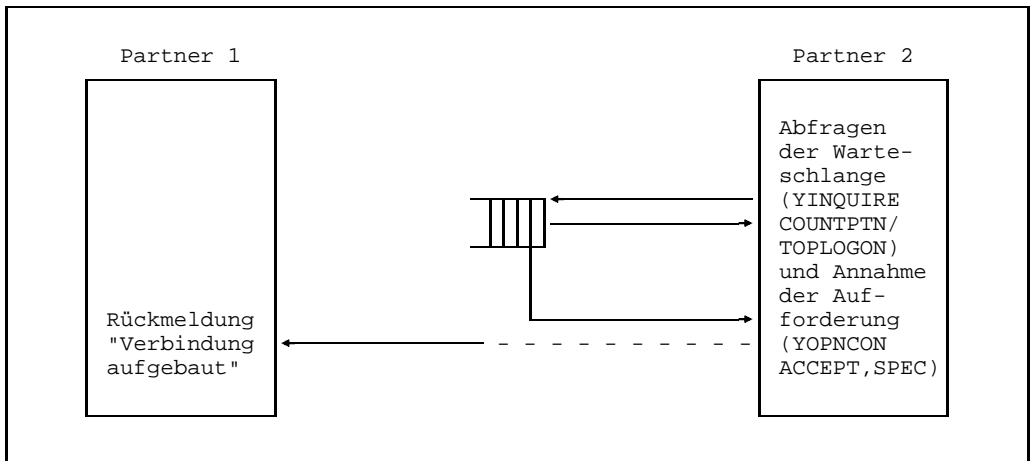
Aufforderung zum Aufbau einer Verbindung

- Es besteht die Möglichkeit, das Eintreffen einer Aufforderung durch eine **asynchrone Meldung** anzeigen zu lassen (LOGON-MELDUNG) (siehe Seite 111ff).
- Nach der Meldung kann in einer **Contingency-Routine** abgefragt werden, welcher Partner die Aufforderung gegeben hat. Daraufhin kann sie angenommen oder zurückgewiesen werden.



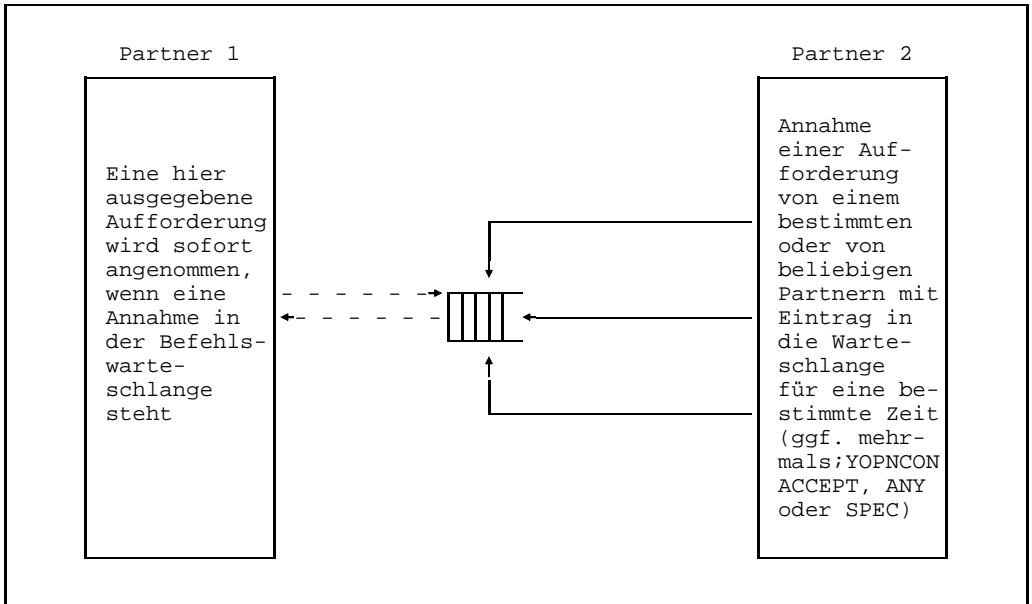
LOGON-Meldung; Abfrage und Annahme der Aufforderung

- Der Partner kann auch die Warteschlange **abfragen, ohne eine Meldung** erhalten zu haben (siehe auch Seite 67), um festzustellen, ob Einträge vorliegen. Abhängig vom Ergebnis wird er Annahmearufe für diese Aufforderung ausgeben (YOPNCON ACCEPT SPEC).
- Er kann auch die Aufforderung zurückweisen, bzw. durch Nichtannahme dafür sorgen, daß nach einer im System definierten Zeit (Kommando /BCTIMES, siehe Handbuch 'Netzmanagement im BS2000') die Aufforderung gelöscht wird.



Abfragen Partnerinformation und Annahme der Aufforderung

- Er hat ferner die Möglichkeit, ohne abzufragen, sozusagen **auf Verdacht**, Annahmeforderungen abzugeben. Trifft in einer bestimmten, angegebenen Zeit die Aufforderung ein, wird sie angenommen. Mehrere solche Annahmen 'auf Verdacht' werden, falls die Aufforderungen noch nicht da sind, in eine Warteschlange eingetragen. Sie können für einen bestimmten (SPEC) oder beliebigen Partner (ANY) ausgegeben werden.

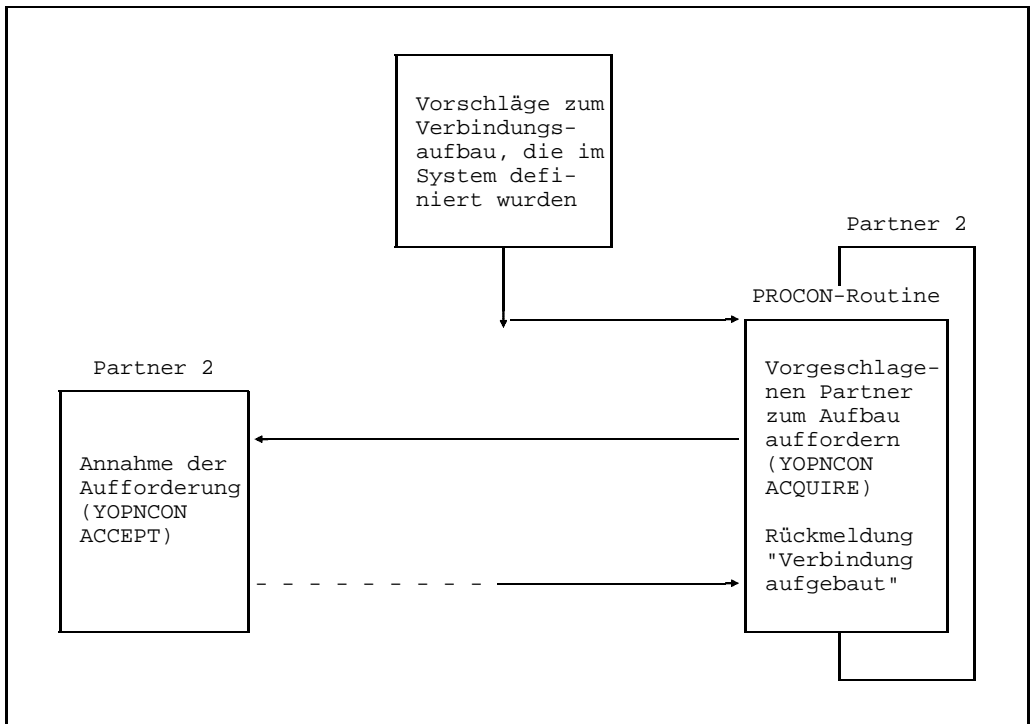


Annahme der Aufforderung

- Eine weitere Möglichkeit ist die Zuordnung von Partnernamen zu einer DCAM-Anwendung bereits **beim Generieren und Starten des Kommunikations-Zugriffssystem** (siehe Makro XSTAT im Handbuch 'Generierung eines Datenkommunikationssystems'). Wird die DCAM-Anwendung eröffnet, bzw. ein /BCIN-Kommando vom Administrator gegeben, das den Prozessorknoten des vorgeschlagenen Partners betrifft, werden ihr diese Partner zum Verbindungsaufbau vorgeschlagen. Dies geschieht durch die PROCON-Meldung. Aufgrund der Meldung muß eine Aufforderung ausgegeben werden (YOPNCON ACQUIRE). Erst wenn diese vom vorgeschlagenen Partner angenommen wurde, ist die Verbindung aufgebaut.

Einschränkung:

An COBOL-Programme werden keine PROCON-Meldungen gegeben.



PROCON-Meldung, Aufforderung an vorgeschlagene Partner und Annahme

- Parallelverbindungen können nur "seriell" aufgebaut werden, d.h. eine weitere Parallelverbindung kann erst dann aufgebaut werden, wenn der Aufbau der vorhergehenden Parallelverbindung abgeschlossen ist (nur DCAM(ISO)-Transport-Service-Anwendungen).



Der weitere Teil dieses Abschnitts gilt nur für DCAM(NEA)-Transport-Service-Anwendungen.

- Beim Generieren und Starten der Kommunikationszugriffsmethode kann man nicht nur Verbindungsvorschläge, sondern auch **Verbindungen** selbst selbst vordefinieren (Makro XKON).

Diese Verbindungen sind bereits aufgebaut, wenn das Kommunikations-Zugriffssystem gestartet wird. Datenübermittlung ist aber erst dann möglich, wenn die Primärtask einen Aufruf abgibt, mit dem sie einen Verbindungswunsch äußert (YOPNCON ACQUIRE).

Ist der Partner eine Datenstation, genügt es, wenn diese eingeschaltet und betriebsbereit ist.

Jeder Kommunikationspartner, der einen Aufruf zum Aufbau einer Verbindung abgibt, beschreibt die von ihm gewünschten Eigenschaften der Verbindung. Bei der Aufforderung zum Verbindungsaufbau werden die Eigenschaften, die den Partner betreffen, diesem mitgeteilt. Dies sind:

- Art der verwendeten Nachrichtenaufbereitung (EDIT)
- Festlegung, wer mit der Datenübermittlung beginnt (PROC)

Das folgende Bild zeigt, wie diese Informationen abgefragt und festgelegt werden:

Partner 1 fordert Partner 2 zum Verbindungsaufbau auf (YOPNCON ACQUIRE). Er macht Partner 2 Vorschläge für EDIT und PROC. Partner 2 hat zwei Möglichkeiten auf den Aufruf von Partner 1 zu reagieren:

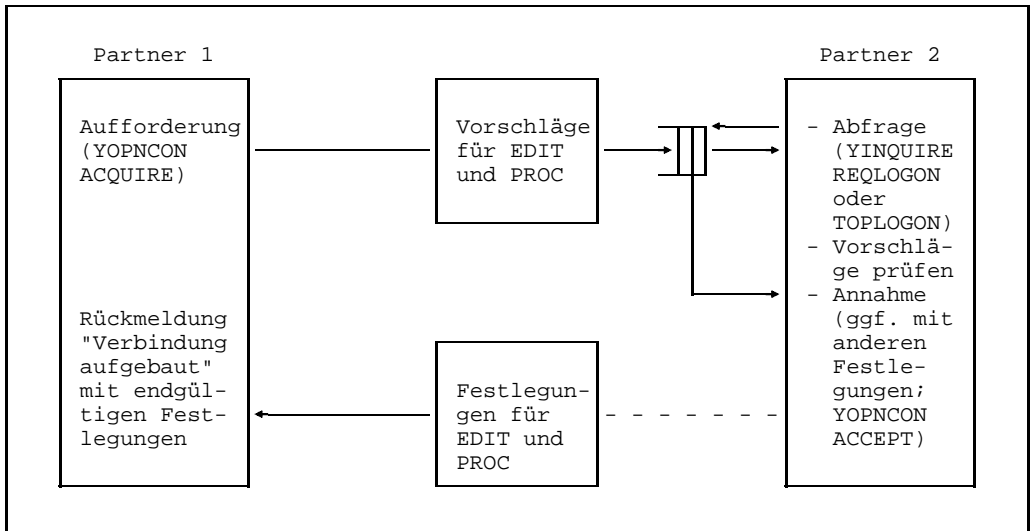
- Er fragt die Vorschläge von Partner 1 ab (YINQUIRE) und prüft sie, um sie dann anzunehmen oder abzulehnen, d.h., andere Werte für EDIT und PROC an Partner 1 zurückzuschicken (YOPNCON ACCEPT).
- Er schickt gleich seine eigenen Werte für EDIT und PROC an Partner 1, ohne dessen Vorschläge zu betrachten.

Für Partner 1 sind in jedem Fall die Werte bindend, die Partner 2 geschickt hat. Die von Partner 1 vorgeschlagenen Werte für EDIT und PROC brauchen also nicht mit den aktuellen Werten übereinzustimmen. Daher muß sich der Anwender nach Aufbau der Verbindung vergewissern, welche Werte aktuell sind.

Sowohl bei einer Aufforderung zum Verbindungsaufbau (YOPNCON ACQUIRE) wie auch bei der Annahme der Verbindung (YOPNCON ACCEPT) können beide Partner einander Verbindungsnachrichten zukommen lassen.

Hinweis

Wenn eine Verbindung vordefiniert wurde, sind ihre Eigenschaften bei der Systemgenerierung festgelegt worden.



Abstimmung über Eigenschaften

3.2.1.1 Beschreibung der aufzubauenden Verbindung

Die Verbindung, die aufgebaut werden soll, muß im DCAM-Programm beschrieben werden. Folgende Angaben sind dazu notwendig:

– **Name und Prozessorname (=Adresse) des Partners.**

Mit dieser Angabe wird ein Partner adressiert, wenn eine Aufforderung ausgegeben wird oder von einem bestimmten Partner (SPEC) eine Aufforderung angenommen wird. Mit Prozessorname=' ' wird der eigene Rechner als Partner adressiert.

Wurde eine Annahme für einen beliebigen Partner (ANY) ausgegeben, wird nach Abschluß des Aufrufs von DCAM der Partner- und Prozessorname zurückgegeben.

– **Begleitinformation**

Hier legt der Anwender eine beliebige Zeichenfolge fest, die er z.B. bei jedem Empfang von Nachrichten über diese Verbindung mitgeliefert bekommt. Es kann die Adresse eines Datenbereichs sein, der dieser Verbindung zugeordnet werden soll oder die Adresse einer Routine, die speziell für diese Verbindung aufgerufen werden soll. Beim Zugriff auf die empfängerglobale Warteschlange mit YRECEIVE ANY ist dieses Verfahren besonders hilfreich. Der Inhalt der Begleitinformation ist frei wählbar. Sie darf nicht länger als 4 byte sein.

Einschränkung:

In COBOL kann die Begleitinformation nicht definiert werden.

– **Datenflußkontrolle**

Beim YSEND kann es vorkommen, daß die betroffene Verbindung überlastet ist, z.B. weil der Partner, welcher sich auch im selben System befinden kann, die Nachrichten zu langsam abholt. In diesem Fall wird der YSEND nicht ausgeführt sondern mit dem RC "Wait for GO" (X'10040C00') beendet. Ein weiterer YSEND auf dieser Verbindung kann erst nach Eintreffen des GO-Signals bei DCAM erfolgreich sein. YSENDS vor Eintreffen des GO-Signals werden mit dem RC "Shortage" (X'10040800') beendet.

Wurde die Verbindung mit PROC=SIGNAL aufgebaut, wird der Anwender über das Eintreffen des GO-Signals informiert (siehe SOLSIG, BS2000 Makroaufrufe an den Ablaufteil). In COBOL durch das Eintreten der Ereignisses "GOSIGNAL" nach YWAIT. Wird das GO-SIGNAL empfangen, ist die Verbindung wieder frei und es können erneut Daten gesendet werden. Zu beachten ist, daß beim Aufbau einer Verbindung mit PROC=SIGNAL beim YSEND die Adresse einer gültigen EID angegeben werden muß.

Wurde die Verbindung nicht mit PROC=SIGNAL aufgebaut, kann das Ende der Überlastung nur durch einen erfolgreichen YSEND festgestellt werden.

Das GO-Signal bietet keine Garantie, daß der nächste YSEND-Aufruf erfolgreich ist.

– **More Data-Funktion**

Mit dem Operanden MDATA=Y/N kann die Anwendung festlegen, ob sie für die Verbindung die More Data-Funktion zur Übergabe von Dateneinheiten nutzen will (siehe Seite 34ff).

– **Länge zu sendender Nachrichten bzw. Dateneinheiten**

Mit dem MAXLN-Operanden gibt die Anwendung an, welche maximale Nachrichtenlänge (bei MDATA=N) bzw. Länge der Dateneinheiten (bei MDATA=Y) auf dieser Verbindung gesendet werden soll.

Die tatsächlich verfügbare Länge (evtl. kleiner als die von der Anwendung im MAXLN-Operanden gewünschte Länge) gibt DCAM als Rückinformation zum Verbindungsaufbau (siehe Seite 64 und 67).

Die Länge dient zur Optimierung der vom System bereitgestellten Puffer und wird nicht an den Kommunikationspartner weitergegeben oder mit ihm ausgehandelt.

– **Länge der erwarteten Empfangsnachrichten (nur bei MDATA=N)**

Arbeitet die Anwendung auf der Verbindung ohne die MDATA-Funktion (MDATA=N), kann sie dem Kommunikationssystem im Operanden RLTH mitteilen, welche maximale Länge von Empfangsnachrichten erwartet wird. Die Größe dient der Speicherplatz-Optimierung und Performance-Verbesserung des Systems. Sie hat rein lokale Bedeutung und wird dem Kommunikationspartner weder mitgeteilt noch mit ihm ausgehandelt.

Die Angabe von RLTH bietet daher auch keine Gewähr, daß nicht doch längere Nachrichten eintreffen und durch die Anwendung zu verarbeiten sind.

Hinweis

Bei MDATA=Y gibt DCAM als Rückinformation zum Verbindungsaufbau die maximal mögliche Länge zu empfangender Dateneinheiten. Die Anwendung hat hier keinen Einfluß auf die Festlegung dieser Größe. Das Kommunikationssystem stellt sicher, daß keine längeren Dateneinheiten übergeben werden.

– Behandlung langer Nachrichten

Da bei MDATA=N (keine More Data-Funktion) selbst durch die Angabe des RLTH-Operanden nicht verhindert werden kann, daß längere Empfangsnachrichten eintreffen, bietet DCAM je nach Problemstellung zwei Alternativen an, siehe dazu nachstehende Tabelle.

Diese Angabe kann später bei der Ausführung eines Empfangsaufrufs (siehe Seite 84ff) wieder geändert werden.

Voraussetzung	Festlegungen
<ul style="list-style-type: none"> - Es werden für diese Verbindung nur Nachrichten bis zu einer maximalen Länge erwartet. Diese Länge sollte im RLTH-Operanden angegeben sein. - Eine Nachricht, die länger ist als erwartet, kann nur fehlerhaft sein. Darum soll die Überlänge weggeworfen werden. 	<p>Es wird für PROC der Wert TRUNC angegeben.</p> <p>Nachrichten, die länger sind als erwartet, werden nur in der erwarteten Länge übergeben. Die Überlänge wird zwar angezeigt, aber weggeworfen.</p>
<ul style="list-style-type: none"> - Es ist nicht vorhersehbar, in welcher Länge die Nachrichten über diese Verbindung eintreffen. Es wird darum die häufigst vermutete Länge angegeben (dementsprechend sollte auch RLTH gesetzt sein) und es wird der Eingabebereich dafür reserviert. - Die mögliche Überlänge soll nicht weggeworfen werden, sondern soll mit einem weiteren Empfangsaufruf abgeholt werden. 	<p>Es wird für PROC der Wert KEEP angegeben. Die Überlänge einer Nachricht wird beim Empfang des 1. Teils angezeigt. Sie wird für einen weiteren Empfangsaufruf aufgehoben.</p>

Hinweis

Da bei der More Data-Funktion (MDATA=Y) DCAM die maximal mögliche Länge zu empfangender Dateneinheiten anzeigt, sollte grundsätzlich ein Empfangsbereich in dieser Länge reserviert werden. Ein Truncation-Fall kann dann nicht eintreten und PROC=KEEP ist nicht relevant.

Wird trotzdem der Eingabebereich kleiner als die gemeldete Länge zu empfangender Dateneinheiten gewählt, so gilt obige Tabelle analog für jede einzelne Dateneinheit der zu empfangenden Nachricht.

– Routenauswahl

Es können bis zu 8 Routen (COBOL) bzw. 16 Routen (Assembler) zu spezifischen Partnern angegeben werden. In der Reihenfolge ihres Auftretens versucht DCAM über die angegebenen Routen eine Verbindung aufzubauen. Auf welcher Route die Verbindung zustande kommt, kann das Anwendungsprogramm bei Angabe mehrerer Routen nicht feststellen.



Der weitere Teil dieses Abschnitts gilt nur für DCAM(NEA)-Transport-Service-Anwendungen.

– Terminalstatus

Bei statusfähigen Datensichtstationen (z.B. 9763) besteht bei Verwendung logischer Datenstationen die Möglichkeit, beim Verbindungsaufbau den aktuellen Terminalstatus abzufragen. Dadurch wird die Information vervollständigt, die VTSU (siehe Handbuch 'VTSU') über die Datenstation besitzt z.B. über geladene Zeichensätze.

– Code der Nachrichten

Da in einem Datenkommunikationssystem auf den Teilstrecken der Übertragung mit unterschiedlichen Codes gearbeitet werden kann, wird hier dem Anwender überlassen, sich nach seiner Problemstellung zu entscheiden; welche Möglichkeiten er hat zeigt die folgende Tabelle.

Voraussetzung	Festlegungen
<ul style="list-style-type: none"> - Der Anwender arbeitet mit logischen Datenstationen zusammen. - Er möchte die Nachrichten immer im Code des eigenen Verarbeitungsrechners bekommen (in der Regel EBCDIC). - Er sendet Nachrichten immer im Code des eigenen Verarbeitungsrechners. 	<p>Es wird für PROC der Wert SYSCODE eingestellt. Die Nachrichten werden, falls erforderlich, im Datenkommunikationssystem übersetzt.</p>
<ul style="list-style-type: none"> - Der Anwender arbeitet nicht mit logischen Datenstationen zusammen. - Er möchte Nachrichten in dem Code erhalten, den der Kommunikationspartner benutzt. - Er sendet Nachrichten in dem Code, den der Partner verarbeiten kann. 	<p>Es wird für PROC der Wert BINARY angegeben. Die Nachrichten werden in jedem beliebigen Code (transparent) übermittelt.</p>

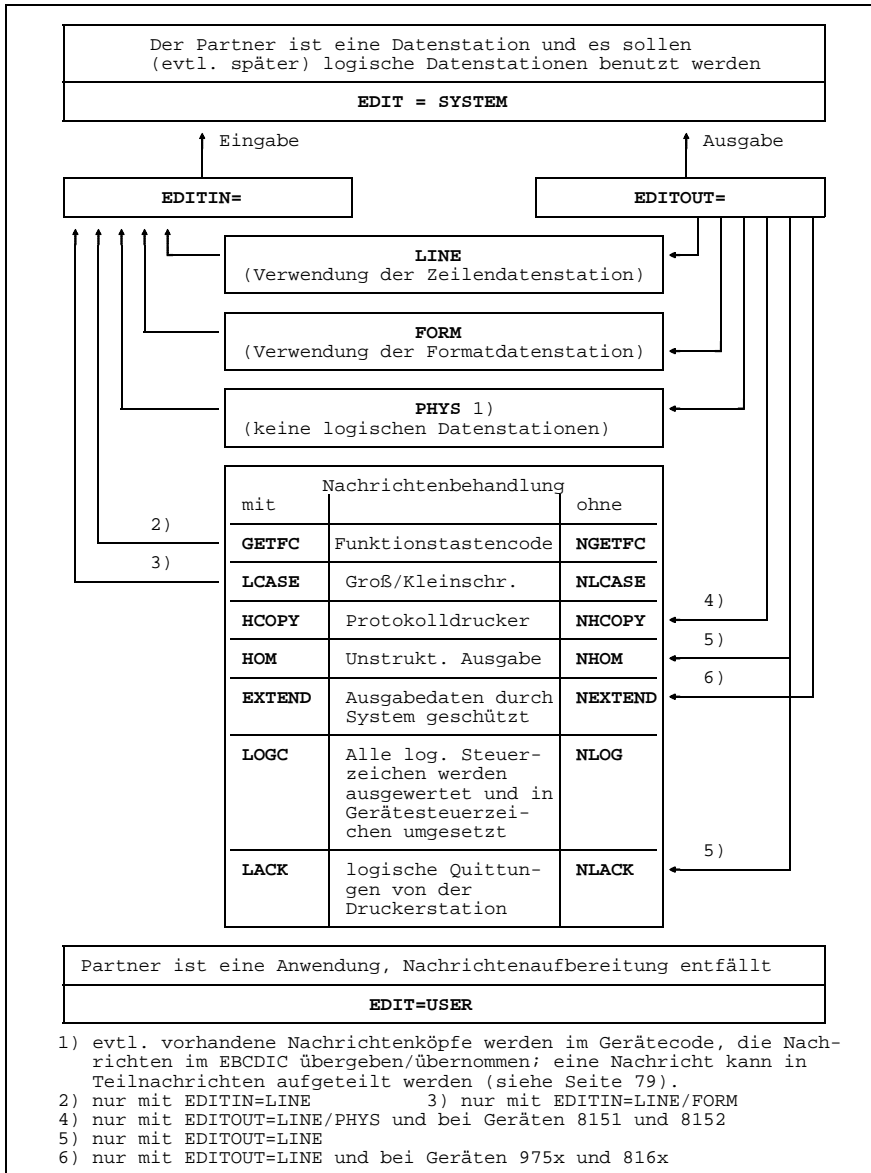
- Das **Kennwort zum Verbindungsaufbau** wird angegeben, wenn eine Aufforderung zum Verbindungsaufbau gestellt werden soll. Es muß so angegeben werden, wie es der Partner (in diesem Fall eine DCAM-Anwendung) beim Eröffnen definiert hat.
- Wenn mit **Nachrichtenverteilung anhand von Verteilcodes** gearbeitet wird, muß der Verweis auf die weitere Beschreibung dieser Verteilcodes hier angegeben werden. Dies wird auf Seite 69ff ausführlich beschrieben.
- Die **maximale Länge der Daten** (MAXLN) die auf dieser Verbindung gesendet werden sollen. Sie ist ein Wert, der zur Optimierung der vom System bereitgestellten Puffer dient; sie wird nicht an den Kommunikationspartner weitergegeben.
- **Eigenschaften von Nachrichten, die mit dem Partner abgestimmt werden.** Über einige Eigenschaften müssen sich, wie bereits im vorigen Abschnitt erwähnt, die beiden Partner verständigen. Diese werden ebenfalls hier beschrieben:

Der **Beginn der Datenübermittlung** wird festgelegt. Soll die DCAM-Anwendung mit der Übermittlung beginnen, wird APPSTART festgesetzt. Ansonsten gilt keine Festlegung (ANYSTART).

Die Art der verwendeten **Nachrichtenaufbereitung** wird festgelegt. Es besteht die Möglichkeit, die Nachrichtenaufbereitung mit den EDIT-Operationen festzulegen (siehe folgendes Bild und Seite 95ff).

Hinweis

Bei Verbindungen mit EDIT=SYSTEM wird im Fall ATTR=DISCO der Verteilcode stets im LINE-Modus aufbereitet.



Möglichkeiten der Nachrichtenaufbereitung

3.2.1.2 Auffordern zum Aufbau

Zum Verbindungsaufbau aufgefordert werden kann **jeder beliebige Partner** im Datenkommunikationssystem.



nur für DCAM(NEA)-Transport-Service-Anwendungen:

Ist der Partner eine Datenstation, wird der zugehörige Prozessor (z.B ein Netzknotenrechner) die Aufforderung beantworten. Im anderen Fall wird die Kommunikationszugriffsmethode im BS2000 reagieren bzw. der Partner (in diesem Fall eine DCAM-Anwendung) selbst. Für den Prozessorknoten, an dem eine **Datenstation** angeschlossen ist, gibt es zur Zeit zwei Möglichkeiten:

- Erhält er eine Aufforderung für eine Datenstation, die nicht erreichbar ist (ausgeschaltet bei Standverbindung, nicht frei bei Wählverbindung u.ä.), wird er sie zurückweisen.
- Ist die Datenstation erreichbar, wird vorausgesetzt, daß auch eine Bedienungsperson vorhanden ist, und die Verbindung wird aufgebaut (der Prozessorknoten nimmt die Aufforderung an).

Die **Kommunikationszugriffsmethode** im BS2000 unterscheidet nach unterschiedlichen Merkmalen. Die Übersicht auf dem folgenden Bild zeigt, daß hier der Anwender von DCAM mehrere Möglichkeiten der Steuerung hat, des Weiteren im Programm selbst, wenn er die Aufforderung bearbeitet. Für eine Prüfung oder auch nur zur einmaligen Übermittlung kann eine Verbindungsnachricht mit der Aufforderung übergeben werden. Diese Nachricht kann einen beliebigen Inhalt haben und eine Maximallänge von 32 byte bei DCAM(ISO) bzw. 80 byte bei DCAM(NEA). Sie ist die einzige Nachricht, die ohne eine schon bestehende Verbindung übermittelt wird.

Ein Partner, der eine **Aufforderung abgeben** will, macht folgende Angaben:

- die Adresse des Partners (Partner und Prozessorname), an den die Aufforderung gerichtet wird.
- wahlweise eine Verbindungsnachricht, die er mit der Aufforderung übermitteln will.



und zusätzlich für DCAM(NEA)-Transport-Service-Anwendungen:

- Beschreibung der Eigenschaften der Verbindung.
- Kennwort zum Verbindungsaufbau, falls gefordert.
- die Nachrichtenverteilung (wenn ohne Verteilcodes gearbeitet wird). Er bestimmt damit, ob er die folgenden Nachrichten dieser Verbindung empfänger-global oder absenderspezifisch empfangen möchte. Diese Angabe kann bei der Datenübermittlung wieder geändert werden.

Wurde die Aufforderung angenommen, erhält er als **Rückinformation**:

- die Begleitinformation, die er festgelegt hat.
- die maximale Länge von Nachrichten/Dateneinheiten, die auf dieser Verbindung in Senderichtung übermittelt werden können (MAXLN-Operand).
- die maximale Länge zu erwartender Dateneinheiten, falls die More Data-Funktion verwendet wird.



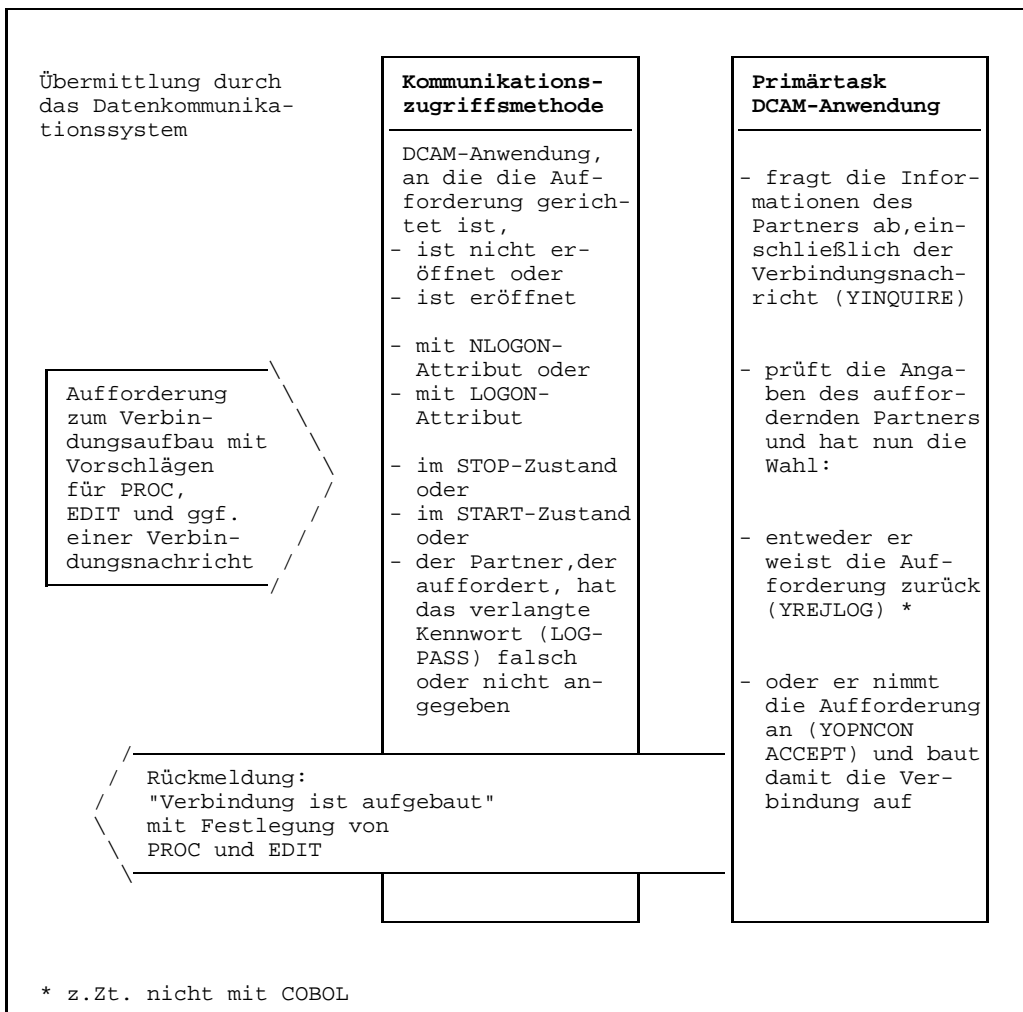
und zusätzlich für DCAM(NEA)-Transport-Service-Anwendungen:

- die endgültige Festlegung für PROC und EDIT.
- Informationen über den Partner (Partnercharakteristika).

Die Möglichkeit, den in diesem Aufruf enthaltenen Befehl asynchron ausführen lassen zu können, wird bei der Beschreibung der Sprach-Eigenheiten erläutert.

Einschränkung für DCAM(NEA)-Transport-Service-Anwendungen

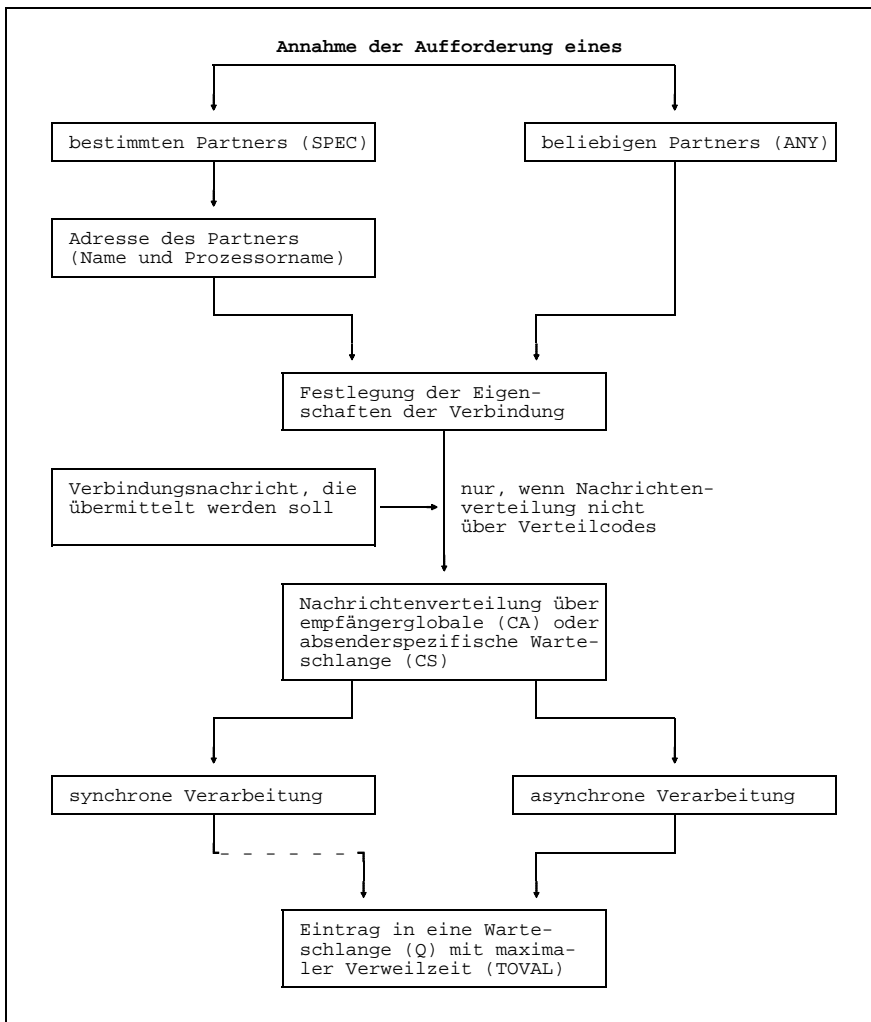
Verbindungsnachrichten werden nur DCAM-Anwendungen zugestellt. Datenstationen, die an Kommunikationsrechner angeschlossen sind, können keine Verbindungsnachricht erhalten.



Bearbeitung einer Aufforderung für DCAM(NEA)-Transport-Service-Anwendungen

3.2.1.3 Annahme einer Aufforderung

Die grundsätzliche Beschreibung der Annahme ist bereits in den vorigen Abschnitten enthalten. Zusammenfassend wird hier aufgezählt, welche Angaben gemacht werden können, wenn eine Annahme ausgegeben wird. Die einzelnen Angaben in ihrer gegenseitigen Abhängigkeit zeigt das folgende Bild.



Angaben zur Annahme einer Aufforderung

Nachdem der Aufruf ausgeführt wurde, erhält die Task:

- die maximale Länge von Nachrichten/Dateneinheiten, die auf dieser Verbindung übermittelt werden können (siehe auch MAXLN-Operand).
- Die maximale Länge zu erwartender Dateneinheiten, falls die More Data-Funktion verwendet wird.



und zusätzlich für DCAM(NEA)-Transport-Service-Anwendungen:
Informationen über den Partner mitgeteilt.

Ferner, wenn die Aufforderung eines beliebigen Partners angenommen wurde:

- den Namen des Partners und
- den Prozessornamen des Partners
- die Begleitinformation

3.2.1.4 Aufbau einer Verbindung - Verwendung von Verteilcodes



Dieser Abschnitt gilt nur für DCAM(NEA)-Transport-Service-Anwendungen.

Wenn die Nachrichtenverteilung anhand von Verteilcodes erfolgen soll (definiert beim Eröffnen der Anwendung), sind bereits beim Aufbau einer Verbindung wesentliche Festlegungen zu treffen. In der Beschreibung einer Verbindung ist in diesem Fall ein Verweis auf den Teil enthalten, der die Verteilcodes beschreibt. Die Beschreibung der Verteilcodes erfolgt in 2 Stufen, um in der Zuordnung der Verbindung zu den Verteilcodes möglichst viel Spielraum zu bieten:

1. Stufe

Beschreibung der Codeposition in der Nachricht und der Länge des/der verwendeten Codes. Der Code muß in den ersten 256 byte der Nachricht enthalten sein und kann maximal 8 byte lang sein.

Ferner: Verweis auf die Beschreibung der 2. Stufe, die maximal 8-mal (COBOL) oder 16-mal (Assembler) vorhanden sein kann.

2. Stufe

Beschreibung der Codes, wie sie von dieser Verbindung benutzt werden. Es können maximal 8 Codes beschrieben werden.

Diese Beschreibungen können mehrfach benutzt werden, d.h. - es können mehrere Verbindungen die gleiche Beschreibung der 1. Stufe verwenden und es können mehrere Beschreibungen der 1. Stufe eine Beschreibung der 2. Stufe verwenden.

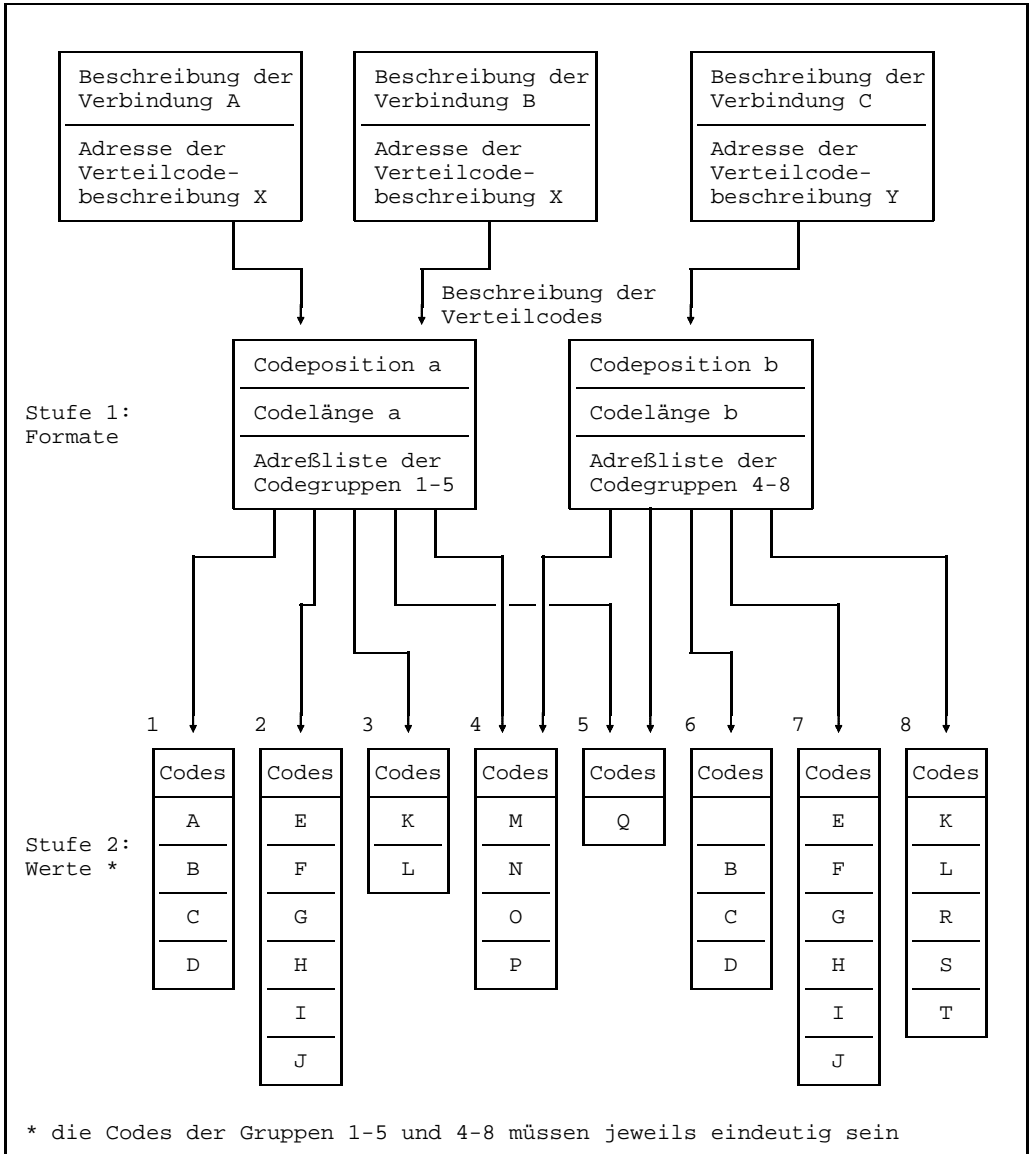
Zu beachten ist folgendes:

- Die Codes, die durch diesselbe Beschreibung der Stufe 1 adressiert werden, müssen eindeutig sein und
- müssen alle die gleiche Länge haben.
- Adressieren mehrere Beschreibungen der Stufe 1 eine Beschreibung der Stufe 2, müssen die Längenangaben in den Beschreibungen der Stufe 1 übereinstimmen.

Durch dieses Verfahren der Verteilcodebeschreibung beim Aufbau einer Verbindung ergeben sich folgende Möglichkeiten:

- Codelänge und Codeposition in der Nachricht sowie Codeanzeiger werden beim Verbindungsaufbau festgelegt und sind für eine bestehende Verbindung nicht veränderbar.
- Von einem Partner können maximal 64 (COBOL) bzw. 128 (Assembler) unterschiedliche Verteilcodes in Gruppen zu je 8 angegeben werden. Er erreicht damit einerseits eine Task über maximal 8 verschiedene Codes, andererseits eine unterschiedliche Anzahl von Tasks.

Unterschiedlich ist die Anzahl deshalb, weil die Zuordnung von Codes und Task über den Verteilungsnamen während des Ablaufs erfolgt. Diese Zuordnung wird von der Primärtask durch die Aufrufe YPERMIT und YFORBID gesteuert (siehe Seite 89ff).



Beispiel für Beschreibung der Formate und Werte von Verteilcodes

3.2.1.5 Anschließen an eine vordefinierte Verbindung



Dieser Abschnitt gilt nur für DCAM(NEA)-Transport-Service-Anwendungen.

Eine vordefinierte Verbindung wird beim **Generieren der Kommunikationszugriffsmethode** festgelegt. Sie ist aufgebaut, wenn die Kommunikationszugriffsmethode gestartet wird. Bevor aber Datenübermittlung möglich ist, müssen sich die beiden Partner erst der vordefinierten Verbindung anschließen.

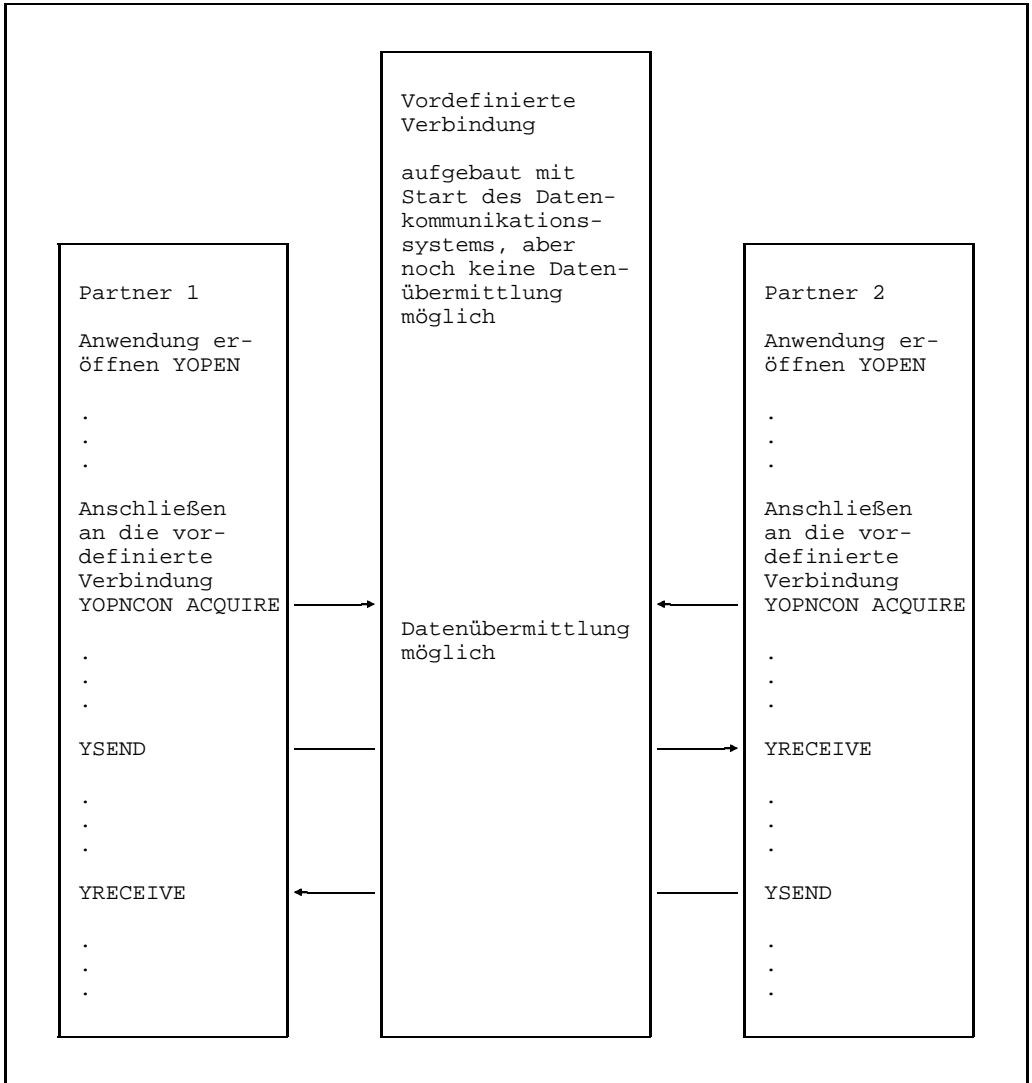
Dazu müssen bei Anwendungen die Primärtasks einen Verbindungswunsch abgeben (YOPNCON ACQUIRE). Bei einer Datenstation genügt es, wenn diese eingeschaltet und betriebsbereit ist.

Da die Eigenschaften einer vordefinierten Verbindung beim Generieren der Kommunikationszugriffsmethode festgelegt werden und die Verbindung bei dessen Start automatisch aufgebaut wird, gibt es einige **Besonderheiten**:

- Senden von Daten, bevor der Partner sich der Verbindung angeschlossen hat, führt zu negativen Transportquittungen.
- Es wird keine LOSCON-ROUTINE angestoßen, wenn ein Partner sich von der Verbindung zurückziehen will (die vordefinierte Verbindung bleibt nämlich weiter aufgebaut).
- Ein YOPNCON ACCEPT SPEC auf einen vordefinierten Partner wird nie erfolgreich beendet. Ist OPTCD=Q angegeben, wird der Aufruf in eine Warteschlange eingetragen und durch TIMEOUT beendet.

Folgende Angaben können bei einem YOPNCON ACQUIRE bei vordefinierten Verbindungen nicht gemacht werden:

- Verbindungsnachricht
- LOGON-Paßwort
- Nachrichtenaufbereitung durch das System oder den Anwender (EDIT=SYSTEM/USER)
- Start der Datenübermittlung (PROC=APPSTART/ANYSTART)



Anschließen an eine vordefinierte Verbindung

3.2.2 Einträge über Partner und Verbindungen abfragen

Wie schon beim Auffordern zum Verbindungsaufbau und der Annahme einer Aufforderung deutlich wurde, ist es notwendig, Einträge über Partner und Verbindungen abzufragen. Dazu stehen verschiedene Varianten des Aufrufs YINQUIRE zur Verfügung. Für Sekundärtasks sind nur die Varianten CIDXLATE, NAMXLATE und zusätzlich für DCAM(NEA)-Transport-Service-Anwendungen die Abfrage der Statusinformationen möglich.

Einschränkung:

Mit COBOL stehen nicht alle Varianten des YINQUIRE zur Verfügung.

Die folgende Tabelle zeigt die Möglichkeiten des YINQUIRE in Abhängigkeit von der verwendeten Programmiersprache.

Abfrage	Variante des YINQUIRE	Programmiersprache
<ul style="list-style-type: none"> - Welcher Partner fordert zur Verbindung auf? - Welche Verbindungsnachricht wird übermittelt und in welcher Länge? <p>Nur DCAM(NEA)-Transport-Service-Anwendung: Wie lauten die vorgeschlagenen Eigenschaften der Verbindung?</p>	Abfrage nach einer asynchronen LOGON-Meldung (REQLOGON)	Assembler
<ul style="list-style-type: none"> - Welcher Partner ist der nächste in der Warteschlange, der zur Verbindung auffordert? - Welche Verbindungsnachricht wird übermittelt und in welcher Länge? <p>Nur DCAM(NEA)-Transport-Service-Anwendung: Wie lauten die vorgeschlagenen Eigenschaften der Verbindung?</p>	Abfrage vor Annahme einer Aufforderung (TOPLOGON)	Assembler/ COBOL

Abfrage	Variante des YINQUIRE	Programmiersprache
Nur DCAM(NEA)-Transport-Service-Anwendung: Welche charakteristischen Merkmale hat der Partner?	Abfrage der Statusinformationen (PTNCHAR,MONCHARS,PEROTERM,BTERMINF)	Assembler/ COBOL
- Von wievielen Partnern liegt eine Aufforderung vor? - Mit wievielen Partnern wurde eine Verbindung aufgebaut?	Abfrage der Partneranzahl (COUNTPTN)	Assembler/ COBOL
- Wie lautet das Kennzeichen der Verbindung, von der Partner- und Prozessorname bekannt ist?	Abfrage des Kennzeichens CID (NAMXLATE)	Assembler
- Wie lauten Partner- und Prozessorname von einer Verbindung, deren Kennzeichen bekannt ist?	Abfrage des Partner- und Prozessornamens (CIDXLATE)	Assembler

3.2.3 Aufforderung zum Verbindungsaufbau zurückweisen

Mit dem Aufruf YREJLOG kann eine Aufforderung zum Verbindungsaufbau zurückgewiesen werden. Dabei genügt es, die mit YINQUIRE ermittelte Adresse des auffordernden Partners (Partner- und Prozessorname) anzugeben.

3.2.4 Eigenschaften einer Verbindung ändern



Dieser Abschnitt gilt nur für DCAM(NEA)-Transport-Service-Anwendungen.

Nachdem beim Aufbau einer Verbindung Ihre Eigenschaften festgelegt wurden, hat der Anwender später noch die Möglichkeit Eigenschaften zu ändern.

Geändert werden können Eigenschaften, die nur den Ändernden selbst betreffen, also nicht solche, die mit dem Partner abgestimmt werden müssen.

Bei Abgabe des Aufrufs YCHANGE müssen alle, im folgenden aufgezählten Werte eingestellt werden. Dabei ist es gleich, ob es geänderte (neue) Werte sind oder die bereits früher (alte) eingestellten.

Es können geändert werden:

- Begleitinformation
- Behandlung zu langer Nachrichten
- Code der Nachrichten
- Werte für den VTSUCB (Nachrichtenbehandlung), siehe Handbuch 'VTSU'
- ggf. Adresse der Verteilcode-Beschreibung, zur Änderung der in Gruppen beschriebenen Verteilcodes (Lage und Länge in der Nachricht sowie Codeanzeiger können nicht geändert werden)

Die Erläuterung dieser Angaben wurde im Abschnitt 'Beschreibung der aufzubauenden Verbindung' gegeben (siehe Seite 58).

3.2.5 Aufforderung zum Verbindungsaufbau zurücknehmen

Soll eine Aufforderung zum Verbindungsaufbau, die an einen Partner gerichtet wurde, wieder zurückgenommen werden, kann dies mit dem Aufruf YCLSCON erfolgen.

Die Aufforderung zum Verbindungsaufbau wird zurückgenommen, wenn der Aufbau noch nicht erfolgt ist. Im anderen Fall wird die Verbindung abgebaut.

3.2.6 Verbindung abbauen

Der explizite Abbau einer Verbindung wird ebenfalls mit dem Aufruf YCLSCON durchgeführt.

Es besteht keine Verpflichtung dazu, da alle Verbindungen implizit abgebaut werden, sobald eine DCAM-Anwendung geschlossen wird. Noch nicht abgeholte Nachrichten gehen bei Verbindungsabbau verloren. Ebenso können unmittelbar vor YCLSCON abgeschickte Nachrichten verloren gehen, die dem Empfänger noch nicht zugeschickt wurden.



Für DCAM(NEA)-Transport-Service-Anwendungen wird empfohlen die letzte Nachricht vor dem Verbindungsabbau mit Transportquittung abzusichern.

Der explizite Abbau kann erforderlich sein:

- für die zeitliche Begrenzung einer Verbindung
- zur Regelung des Datendurchsatzes (Verbindungen können abgebaut werden, um anderen Verbindungen günstigere Durchsatzraten zu ermöglichen), usw.

3.3 Datenübermittlungsfunktion

Nachdem eine DCAM-Anwendung eröffnet und eine Verbindung aufgebaut wurde, sind die Voraussetzungen zur Datenübermittlung gegeben.

Die Datenübermittlungsfunktion führt aus:

- **Senden einer Nachricht/Dateneinheit (YSEND)**

Die einem Kommunikationspartner zu übermittelnde Nachricht/Dateneinheit wird in den Datenspeicher der Kommunikationszugriffsmethode übertragen.

- **Empfangen einer Nachricht/Dateneinheit (YRECEIVE)**

Die Nachricht/Dateneinheit eines Kommunikationspartners (irgendeines, eines bestimmten oder, bei DCAM(NEA)-Transport-Service-Anwendungen mit einem bestimmten Verteilcode) wird in den Datenspeicher des Anwendungsprogramms eingetragen.

- **Senden und Empfangen kombiniert (YSENDREC)**

Mit diesem Aufruf wird eine Nachricht/Dateneinheit in den Datenspeicher der Kommunikationszugriffsmethode eingetragen und es wird eine Nachricht/ Dateneinheit des gleichen Kommunikationspartners in den eigenen Datenspeicher übernommen.

- **Empfangsaufrufe zurücknehmen und CS/CA-Zustand einer Verbindung ändern (YRESET)**

Mit diesem Aufruf kann eine Task alle ihre anstehenden YRECEIVE-Aufrufe einer bestimmten Anwendung oder einer bestimmten Verbindung zurücknehmen und den CS/CA-Zustand ändern.



Für DCAM(NEA)-Transport-Service-Anwendungen stehen zur Steuerung der Nachrichtenverteilung anhand von Verteilcodes zusätzlich zur Verfügung:

- **Verteilungsnamen zur Verteilcodegruppen zuordnen (YPERMIT)**

Die Primärtask ordnet dem Verteilungsnamen einer oder mehrerer Sekundär-tasks bestimmte Verteilcodes zu.

- **Zuordnung auflösen (YFORBID)**

Eine getroffene Zuordnung wird von der Primärtask wieder rückgängig gemacht.

3.3.1 Senden einer Nachricht/Dateneinheit

Jede Task einer DCAM-Anwendung kann eine Nachricht/Dateneinheit senden (YSEND) und löst damit folgende Aktionen aus:

- Die Daten, die als Nachricht/Dateneinheit für einen Partner bestimmt sind, werden in den Datenspeicher der Kommunikationszugriffsmethode übertragen.
- Die Zugriffsmethode erhält den Auftrag, die Daten dem Partner zu übermitteln.
- Der Aufruf wird abgeschlossen, und die Zugriffsmethode beginnt mit der Übermittlung.

Mit der zu übermittelnden Nachricht/Dateneinheit erhält die Zugriffsmethode vom Anwender eine Reihe von Angaben. Diese werden ergänzt durch Angaben, die beim Aufbau der Verbindung festgelegt wurden (siehe Seite 58).

Die folgende Tabelle zeigt, welche Zusatzangaben unter bestimmten Voraussetzungen notwendig sind.

Voraussetzung	Festlegungen
Nach erfolgtem Verbindungsaufbau wurde die maximale Länge der auf dieser Verbindung zu sendenden Nachricht/Dateneinheit bekanntgegeben.	Im Rahmen der Maximallänge wird die Länge der zu sendenden Nachricht/Dateneinheit angegeben.
Die auf dieser Verbindung ankommenden Nachrichten/Dateneinheiten sollen über die absenderspezifische Warteschlange empfangen werden.	Es wird die absenderspezifische Warteschlange eingestellt (CS-Zustand der Verbindung).
Die auf dieser Verbindung ankommenden Nachrichten/Dateneinheiten sollen über die empfängerglobale Warteschlange empfangen werden.	Es wird die empfängerglobale Warteschlange eingestellt (CA-Zustand der Verbindung).
Bei der Eröffnung der Verbindung wurde mit PROC=SIGAL das GO-Signal für den Fall der Überlastung der Verbindung angefordert.	Bei jedem YSEND-Aufruf ist ein gültiges Ereigniskennzeichen (EID) anzugeben, über welches ggf. ein GO-Signal zugestellt werden kann.
Auf der Verbindung wird mit der More Data-Funktion gearbeitet (MDATA=Y).	Es ist anzugeben, ob die Dateneinheit die letzte innerhalb der logischen Nachricht ist (OPTCD=GROUP) oder nicht (OPTCD=ELEMENT).

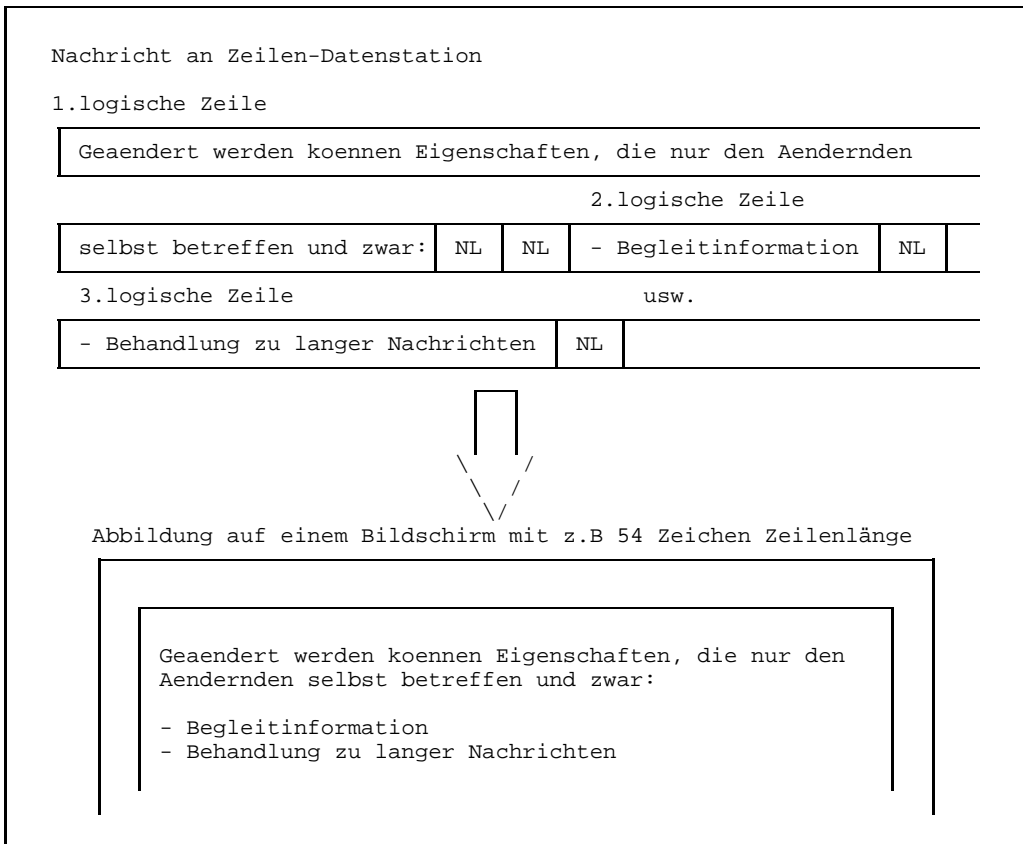
Voraussetzung	Festlegungen
<p>Die folgenden Punkte gelten nur für DCAM(NEA)-Transport-Service-Anwendungen:</p> <ul style="list-style-type: none"> - Bei der Eröffnung der Anwendung wurde der Quittungsempfang ermöglicht (für die Primärtask = PRIMTASK oder die auffordernde Task = REQTASK). - Es ist die Bearbeitung von Transportquittungen vorgesehen. - Dem YSEND-Aufruf soll unmittelbar ein YLSCON- oder YCLOSE-Aufruf folgen, sodaß die Verbindung u.U. abgebaut wird, bevor die Daten übermittelt sind. Hier sollte in jedem Fall die Transportquittung abgewartet werden. 	<p>Es wird</p> <ul style="list-style-type: none"> - eine Laufnummer angegeben, damit bei Eintreffen der Transportquittung diese der richtigen Nachricht zugeordnet werden kann. (höchster Wert = 2047). - es wird eine positive Quittung angefordert (negative Quittungen werden automatisch erzeugt, wenn die Nachricht nicht zugestellt werden kann).
<p>Auf dieser Verbindung wurde keine Nachrichtenaufbereitung durch das System vorgesehen. Es wurde eingestellt:</p> <ul style="list-style-type: none"> - EDIT=USER: Der Anwender stellt die Nachricht so bereit, wie sie der Kommunikationspartner verarbeiten kann. - EDIT=SYSTEM und EDITOUT=PHYS: Das System übernimmt die Blockung bei Geräten mit kleinem Eingabepuffer. Alles übrige stellt der Anwender so bereit, wie es die Datenstation verarbeiten kann. 	<p>Die Nachricht kann klassifiziert werden als:</p> <ul style="list-style-type: none"> - ELEMENT - SUBGROUP - GROUP <p>Diese Kennzeichnung der Nachricht wird durch das Netz zum Empfänger übertragen. Er erhält sie mit der Nachricht. Ist er eine Datenstation, wird ein ELEMENT als Teilnachricht, SUBGROUP und GROUP als Nachricht mit Abschluß der Übertragung übermittelt.</p>
<p>Auf dieser Verbindung wurde Nachrichtenaufbereitung durch das System vorgesehen.</p>	<p>Für die Benutzung der logischen Datenstationen gelten besondere Regeln (siehe Handbuch 'VTSU').</p>
<p>Es ist eine besondere Situation eingetreten (z.B. Rückstau auf der Verbindung), die durch entsprechende Rückmeldung bei einem abgewiesenen YSEND festgestellt wurde. Es soll darum der Partner zur sofortigen Ausgabe von Empfangsaufrufen aufgefordert werden.</p>	<p>Die Nachricht wird als Expresnachricht deklariert, was allerdings beinhaltet, daß ihre Länge höchstens 8 byte betragen darf.</p>



Der weitere Teil dieses Abschnitts gilt nur für DCAM(NEA)-Transport-Service-Anwendungen.

Je nach Einstellung der **Nachrichtenaufbereitung** ist für den Aufbau der Nachrichten unterschiedlich vorzugehen.

Bei Verwendung der **Zeilen-Datenstation** (siehe Handbuch 'VTSU') kann der Anwender die Nachrichten in logische Zeilen unterteilen. Jede logische Zeile wird mit dem Zeichen 'Neue Zeile', sedezimal 15, abgeschlossen. Der darauf folgende Text wird an der Datenstation automatisch auf den Zeilenanfang der nächsten Zeile gesetzt. Ist eine logische Zeile länger als die Zeilenlänge der Datenstation, wird die logische Zeile automatisch weiter unterteilt. Der Beginn einer Nachricht wird immer auf den Zeilenanfang einer neuen Zeile gesetzt. Im folgenden Bild wird dies an einem Beispiel veranschaulicht.

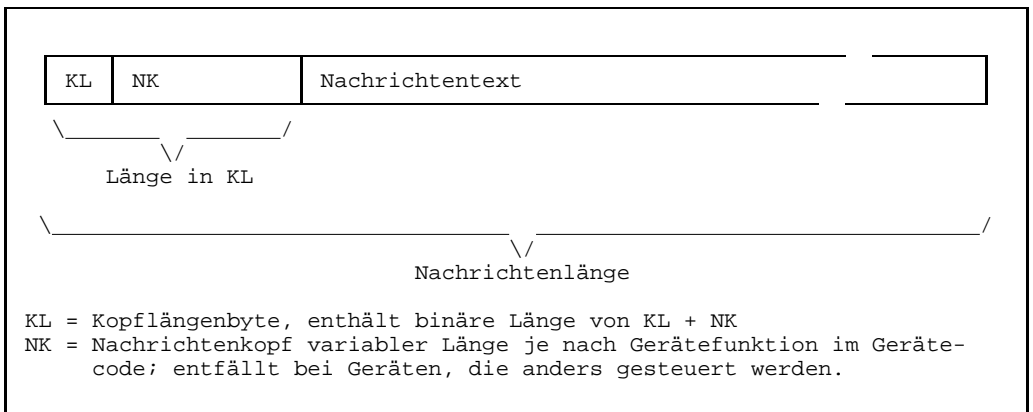


Nachricht an die Zeilen-Datenstation

Wird die **Format-Datenstation** eingesetzt, übergibt der Anwender die Nachricht so, wie er sie von der Formatierungsroutine bekommt, jedoch ohne das vorgesetzte Satzlängenfeld. Die Satzlänge muß er aus dem Satzlängenfeld in das entsprechende DCAM-Feld übertragen.

Wurden für die Verbindung die Werte **EDIT=SYSTEM** und **EDITOUT=PHYS** eingestellt, muß jeder Nachricht ein sog. Kopflängenbyte vorangestellt werden. Die dort enthaltene Binärzahl gibt an, wie lang der darauf folgende Nachrichtenkopf einschließlich dieses Bytes ist. Der Nachrichtenkopf wird der Datenstation, im Unterschied zur eigentlichen Nachricht, ohne Codeübersetzung übermittelt. Der Anwender kann sich bei der Erstellung und auch späteren Auswertung des Nachrichtenkopfes somit an die Gerätebeschreibung halten. Bei Geräten, die keinen Nachrichtenkopf verarbeiten, muß dennoch das Kopflängenbyte vor die Nachricht gesetzt werden, in diesem Fall mit der Länge 1.

Im folgenden Bild wird im Überblick gezeigt, wie die Nachrichten aufzubauen sind.



Aufbau der Nachricht bei EDITOUT=PHYS

Die Nachricht kann in diesem Fall und wenn EDIT=USER eingestellt wurde, in Teilnachrichten unterteilt werden. Jede Teilnachricht muß entsprechend markiert werden.

Gruppe von Nachrichten						
Untergruppe 1 von Nachrichten				Untergruppe 2 von Nachrichten		
Nachricht 1	Nachricht 2	Nachricht 3	Nachricht 4	Nachricht 5	Nachricht 6	Nachricht 7
ELEMENT	ELEMENT	ELEMENT	SUBGROUP	ELEMENT	ELEMENT	GROUP

Beispiel für Strukturierung von Nachrichten

Abhängig vom Gerätetyp und -zustand ist mit einem unterschiedlichen Verhalten der Geräte zu rechnen.

Hinweis

Nach der Übertragung von Teilnachrichten werden mehrfach genutzte physikalische Leitungen (Konzentrator) jeweils wieder freigegeben.

3.3.2 Nachricht/Dateneinheit empfangen

Empfangen einer Nachricht/Dateneinheit bedeutet, daß die Nachricht/ Dateneinheit eines Partners in den Datenspeicher des eigenen Programms übertragen wird, nachdem sie aus der Warteschlange ausgetragen wurde.

Dieser Vorgang ist mit Beendigung des Aufrufs YRECEIVE abgeschlossen, wenn dieser synchron ausgeführt wurde. Wurde er asynchron ausgeführt, ist mit Abschluß des Aufrufs lediglich der Befehl an das Kommunikationssystem gegeben, die die Nachricht/Dateneinheit, wenn sie eintrifft, zu übertragen. Der Anwender erhält auf ein vereinbartes Ereigniskennzeichen ein Signal, wenn die Nachricht/Dateneinheit übertragen ist. Er kann dieses Signal abfragen und danach die Nachricht/Dateneinheit verarbeiten (siehe auch Seite 105). Bei dieser Methode kann evtl. unvermeidliche Wartezeit auf eine Nachricht/Dateneinheit für eine andere Verarbeitung genutzt werden.

Welche Nachrichten/Dateneinheiten eine Task empfangen kann, hängt von der eingestellten Warteschlange ab:

- Ist die empfängerglobale Warteschlange eingestellt (CA-Zustand der Verbindung), wird die Nachricht/Dateneinheit der Task übergeben, die zuerst auf die empfängerglobale Warteschlange (ANY) zugreift.
- Ist die absenderspezifische Warteschlange eingestellt (CS-Zustand der Verbindung), wird die Nachricht/Dateneinheit der Task übergeben, die die absenderspezifische Warteschlange eingestellt hat (den CS-Zustand verursacht hat).



Welche Nachricht eine Task empfangen kann, hängt für DCAM(NEA)-Transport-Service-Anwendungen auch davon ab, ob es sich um eine Normal- oder Expressnachricht handelt. Die folgende Tabelle zeigt Nachrichtenverteilung und Empfang in Abhängigkeit der Nachrichtenart.

Art der Nachrichtenverteilung		Typ der Nachricht	
		Normalnachricht	Expressnachricht
Anhand von Verteilcodes (DISCO)		Nach ihrem Verteilcode wird sie der Task übergeben, die zum Empfang berechtigt ist 1)	Sie wird der Primärtask übergeben
ohne Verteilcodes (NDISCO)	über empfänger-globale Warteschlange (CA-Zustand der Verbindung)	Sie wird der Task übergeben, die zuerst auf die empfängerglobale Warteschlange (ANY) zugreift	Sie wird der Primärtask übergeben
	über absenderspezifische Warteschlange (CS-Zustand der Verbindung)	Sie wird der Task übergeben, die die absenderspezifische Warteschlange eingestellt hat (den CS-Zustand verursacht hat)	

1) Ist der Verteilcode noch nicht vorgesehen oder keiner Sekundärtask zugeordnet, wird die Nachricht der Primärtask zugestellt.

Bei Abgabe des YRECEIVE-Aufrufs macht der Anwender zusätzlich zur Beschreibung der Verbindung, die bereits mit YOPNCON festgelegt wurde (siehe Seite 58) weitere Angaben:

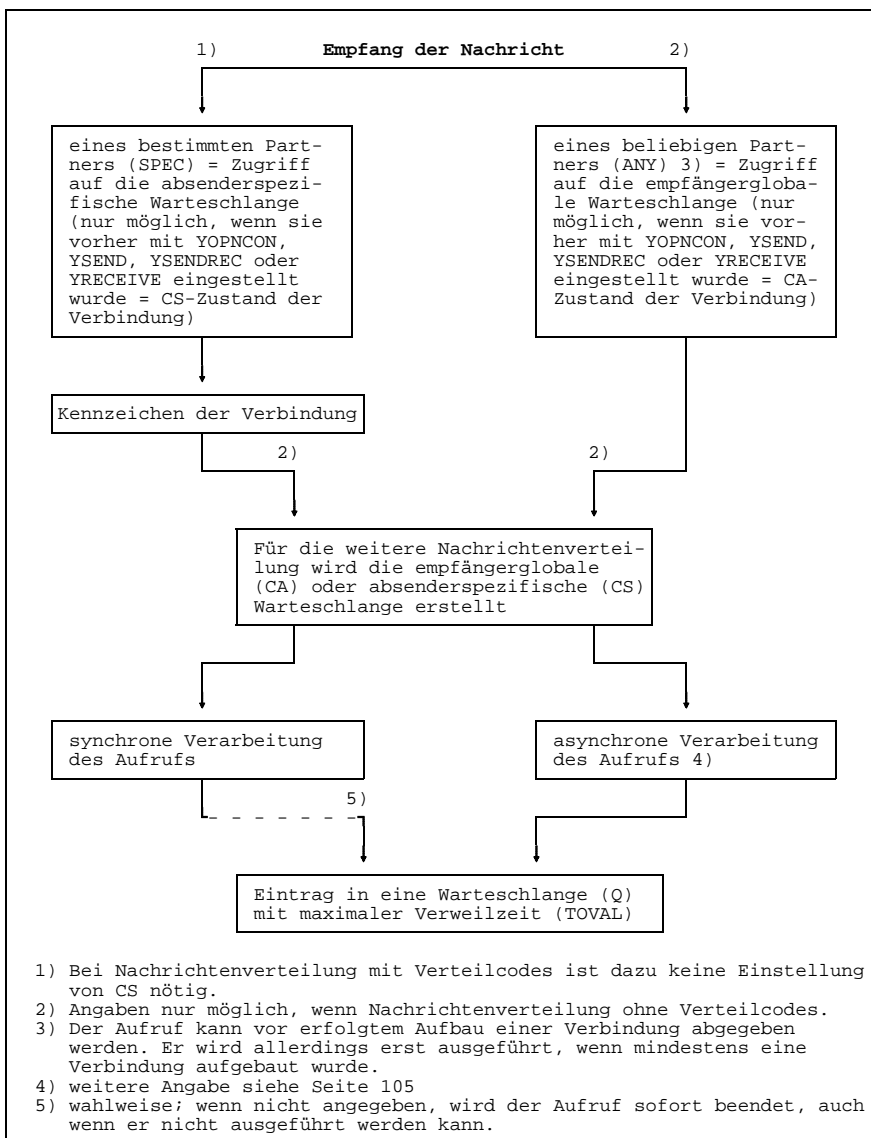
– zur **Nachricht/Dateneinheit**:

Er legt die Adresse des Feldes fest, in das die Nachricht/Dateneinheit eingetragen werden soll mit der Länge, die durch dieses Feld vorgegeben ist.

Er kann die Behandlung zu langer Nachrichten, wie sie beim Verbindungsaufbau festgelegt wurde, dieser Nachricht angepaßt ändern. Er kann also für diesen Aufruf festlegen, ob eine Überlänge für weitere Abholung aufgehoben oder ob sie wegwerfen werden soll.

– Zur Ausführung des Aufrufs:

Hier sind voneinander abhängige Angaben zu machen, wie sie im folgenden Bild dargestellt sind.



Angaben zum Empfang einer Nachricht/Dateneinheit

Nach erfolgreichem Nachrichtenempfang im angegebenen Bereich werden von DCAM weitere Informationen zurückgegeben:

- **Absenderangabe**,
wenn von einem beliebigen Partner empfangen wurde (ANY).
- **Begleitinformation** des Anwenders,
wie er sie beim Aufbau der Verbindung festgelegt hat (Empfang von einem beliebigen Partner: ANY).
- **Länge** der Nachricht/Dateneinheit
oder wenn die Nachricht/Dateneinheit länger als der zur Verfügung gestellte Speicherbereich ist, die Länge des zu langen, noch nicht übermittelten Teils (KEEP). Soll auf den zu langen Teil mit einem weiteren Empfangsaufwurf zugegriffen werden, muß vorher die absenderspezifische Warteschlange eingestellt worden sein.

Bei Verwendung der More Data-Funktion (MDATA=Y) wird im Rückmeldefeld angezeigt, ob die Dateneinheit die letzte Einheit einer logischen Nachricht ist.



Der weitere Teil dieses Abschnitts gilt nur für DCAM(NEA)-Transport-Service-Anwendungen.

- **Laufnummer** der Nachricht,
wie vom Partner festgelegt oder eine im System erzeugte (bei Datenstationen als Absender).
- **Struktur** der Nachricht
(Angaben im Rückmeldefeld) und zwar, ob es ein Element, das letzte Element einer Untergruppe oder einer Gruppe ist.
- **Typ** der Nachricht
(Angaben im Rückmeldefeld) und zwar, ob eine Normal- oder Expressnachricht übermittelt wurde.

Die Möglichkeit, **Transportquittungen** zu empfangen, ist abhängig von Festlegungen bei der Eröffnung der Anwendung und Angaben beim Senden der Nachricht, um deren Quittung es geht, siehe dazu nachstehende Tabelle.

	Festlegung bei YSEND oder YSENDREC	
Festlegung bei YOPEN der Anwendung in der Primärtask	Eine positive Quittung wird angefordert (TACK)	Es wird keine positive Quittung angefordert (NTACK)
Die anfordernde Task soll Quittungen erhalten (REQTASK)	Auslieferung von positiven und negativen Quittungen an die anfordernde Task	Auslieferung nur von negativen Quittungen an die Primärtask
Die Primärtask soll alle Quittungen erhalten (PRIMTASK)	Auslieferung von positiven und negativen Quittungen an die Primärtask	Auslieferung von negativen Quittungen an die Primärtask
Es sollen keine Quittungen übergeben werden (NOTACK)	Es werden weder positive noch negative Quittungen übergeben	

Soweit nicht die Übergabe mit asynchroner Meldung vereinbart wurde (siehe Seite 111), werden Transportquittungen mit einem YRECEIVE-Aufruf empfangen. Alle Angaben über die Art der Quittung usw. werden im Rückmeldefeld des Aufrufs übergeben. Ferner steht im Feld 'TACKNO' die Laufnummer der Nachricht, wie sie beim Senden festgelegt wurde. Damit ist eine Identifikation möglich.

Zu beachten ist, daß der Empfang einer Transportquittung mit YRECEIVE keine Auswirkung auf die Einstellung der Warteschlange hat (CS/CA-Zustand), auch wenn dies im Aufruf angegeben war.

Einschränkung:

Auf YSEND kann bei EDIT=SYSTEM eine zweite, negative Transportquittung eintreffen. Sie geht bei SHARE-Anwendungen unabhängig von REQTASK an die Primärtask.

3.3.3 Senden und Empfangen kombiniert

Wenn der kombinierte Aufruf verwendet werden soll, gilt das, was bereits zu den einzelnen Aufrufen gesagt wurde. Da die Kombination nur den Zugriff auf die absenderspezifische Warteschlange zuläßt, muß für diese Verbindung die absenderspezifische Warteschlange eingestellt worden sein (=CS-Zustand der Verbindung); beim YOPNCON oder bei einem früheren YSEND, YSENDREC oder YRECEIVE.

Einschränkung:

Bei COBOL gibt es YSENDREC nicht.

3.3.4 Empfangsaufrufe zurücknehmen und CS/CA-Zustand ändern

Mit dem Aufruf YRESET können asynchrone YRECEIVE-Aufrufe zurückgenommen werden.

Ein **YRESET(ANY)-Aufruf** nimmt alle YRECEIVE-Aufrufe beliebiger Partner (ANY) zurück.

Ein **YRESET(SPEC)-Aufruf** nimmt alle YRECEIVE-Aufrufe eines bestimmten Partners (SPEC) zurück.

Ein YRESET(SPEC)-Aufruf setzt gleichzeitig den CS/CA-Zustand neu. Dies gilt nicht für DCAM(NEA)-Transport-Service-Anwendungen, die mit verteilcodespezifischen Warteschlangen arbeiten (ATTR=SHARE,DISCO).

3.3.5 Verteilcode-Zuordnung steuern

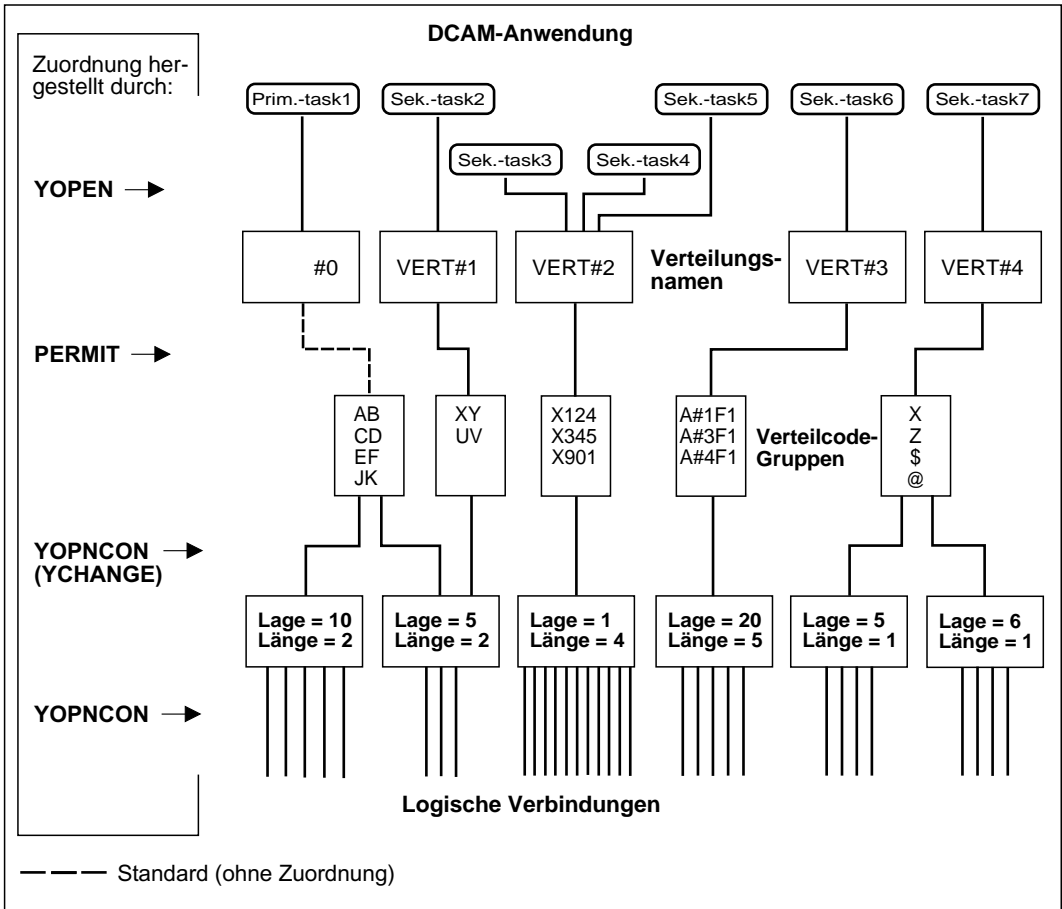


Dieser Abschnitt gilt nur für DCAM(NEA)-Transport-Service-Anwendungen.

Die Primärtask steuert die Verteilcode-Zuordnung, indem sie einer Verteilcode-Gruppe einen Verteilungsnamen zuordnet bzw. die Zuordnung wieder auflöst. Sie definiert die Verteilcode-Gruppe für eine oder mehrere Verbindungen, wenn sie die Verbindungen aufbaut (siehe Seite 69).

Eine oder mehrere Tasks geben den Verteilcodenamen beim Eröffnen der Anwendung an (siehe Seite 46 und 47).

Die Primärtask ordnet den Verteilcodes einer oder mehrerer Verbindungen Verteilungsnamen einer oder mehrerer Tasks zu. Löst die Primärtask eine solche Zuordnung auf, verhindert sie die Datenübermittlung von den betroffenen Verbindungen zu den betroffenen Tasks.



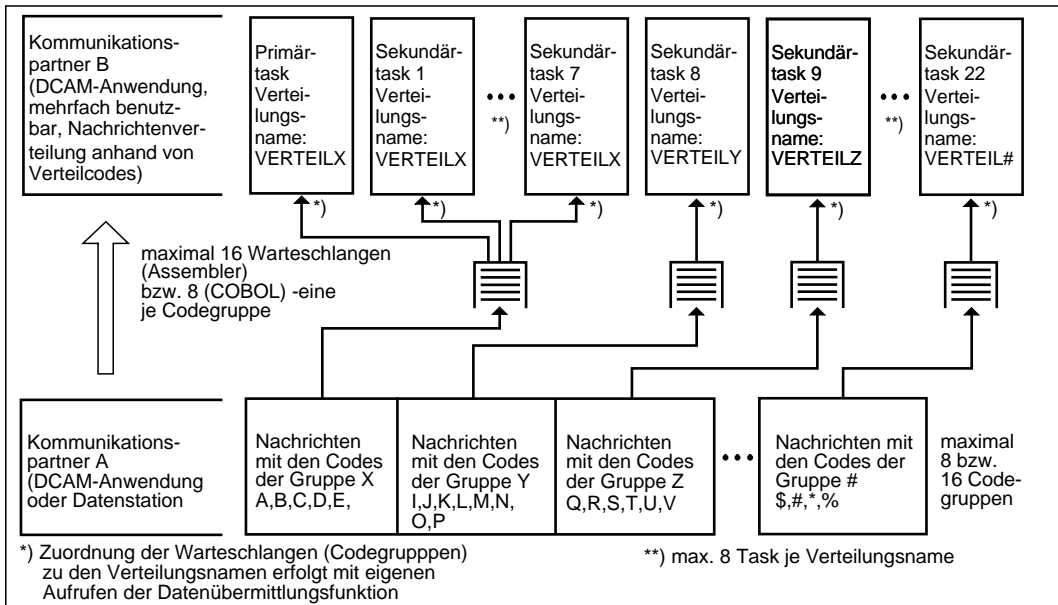
Gesamtübersicht der Nachrichtenverteilung anhand von Verteilcodes (Beispiel)

Verteilungsname einer Verteilcode-Gruppe zuordnen

Mit dem Aufruf YPERMIT wird einer Verteilcode-Gruppe ein Verteilungsname zugeordnet. Damit können Tasks, die diesen Namen angegeben haben, Nachrichten mit den Codes der zugeordneten Gruppe erhalten.

Für die Zuordnung gilt:

- Eine Gruppe von Verteilcodes (Beschreibung der Stufe 2) kann immer **nur einem Verteilungsnamen** zugeordnet werden.
- Bis zu **8 Tasks** können **den gleichen Verteilungsnamen** benutzen. Dies ist dann notwendig, wenn ein Programm (SHARED CODE, siehe Seite 115 und 125) mehrere Tasks steuert. Bei gleichem Verteilungsnamen werden die Tasks nach dem 'fifo'-Prinzip bedient (fifo=first in first out; der Warteschlangeneintrag, der zuerst erfolgte, wird auch zuerst bearbeitet).
- Im Rahmen der getroffenen Zuordnung kann eine Task auch **absenderspezifisch** auf die Nachrichten zugreifen (YRECEIVE SPEC). Dazu ist nicht notwendig, diese Warteschlange vorher einzustellen (es gibt keinen CS/CA-Zustand der Verbindung, wenn mit Verteilcodes gearbeitet wird).



Beispiel für Adressierung einer Taskgruppe über Verteilcodes

Nachrichten mit Verteilcodes, die keinem Verteilungsnamen zugeordnet wurden oder die nicht interpretierbar sind, werden der Primärtask zugeleitet.

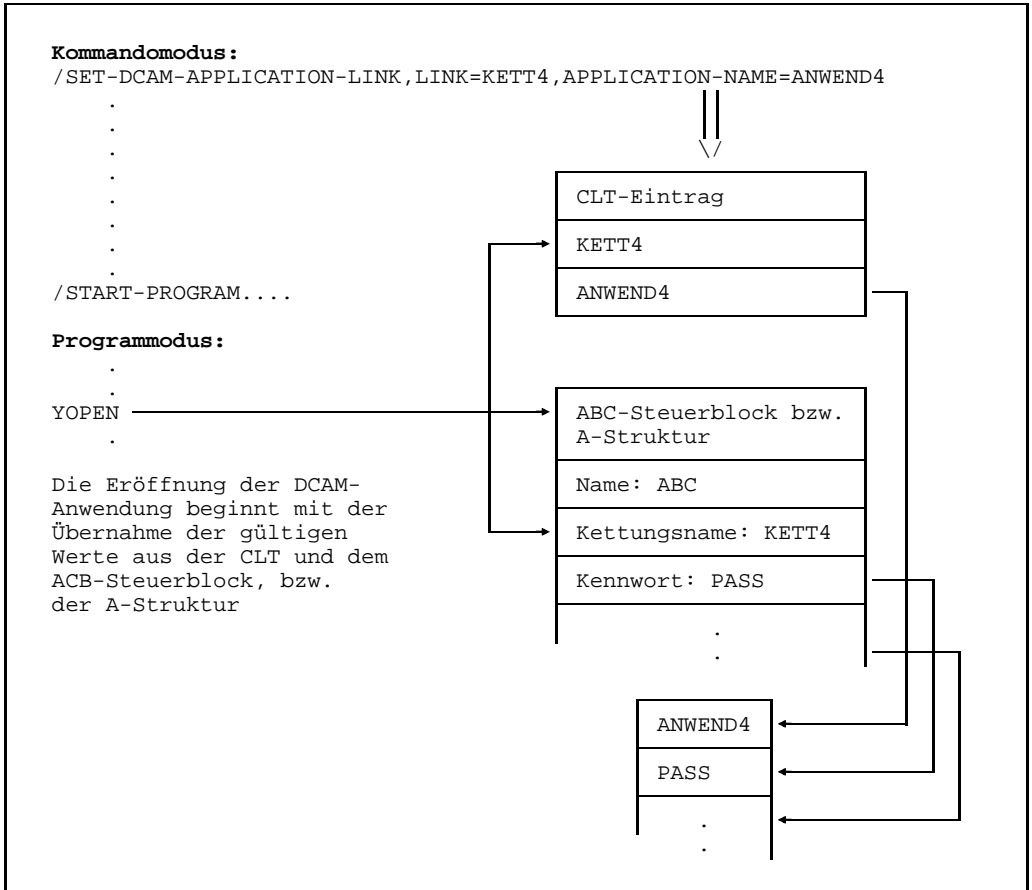
Alte Zuordnungen können durch neue ersetzt werden, wenn ein neuer YPERMIT-Aufruf gegeben wird.

Zuordnung auflösen

Soll die Zuordnung für einen Verteilungsnamen nicht neu getroffen, sondern nur rückgängig gemacht (aufgelöst) werden, steht dafür der YFORBID-Aufruf zur Verfügung. Bereits eingetroffene Daten werden danach der Primärtask übergeben.

3.4 Namen-Zuweisungsfunktion

Mit dieser Funktion können Parameterwerte für die DCAM-Anwendung oder die Verbindungen erst **zum Ablaufzeitpunkt** angegeben werden. Dies wird erreicht durch Verknüpfung gültiger Werte in einer taskspezifischen Tabelle (CLT=communication link table) mit den Angaben im Programm (zum Vergleich: die Aktualisierung der Einträge im FCB (file control block) eines Programms erfolgt durch die TFT (task file table), deren Einträge durch den FILE-Makro, bzw. das /SET-FILE-LINK-Kommando erzeugt wurden).



Beispiel für Zuweisung des Namens einer DCAM-Anwendung

Die Tabelle wird aufgebaut:

- im Programmmodus (nur Assembler) durch die Makros YAPPL für Angaben zur Anwendung und YCONN für Angaben zur Verbindung.
- im Kommandomodus (Assembler/COBOL) durch die Kommandos

`/SET-DCAM-APPLICATION-LINK` bzw. `/REMOVE-DCAM-APPLICATION-LINK` für Angaben zur Anwendung und

`/SET-DCAM-CONNECTION-LINK` bzw. `/REMOVE-DCAM-CONNECTION-LINK` für Angaben zur Verbindung (siehe Handbuch 'Benutzer-Kommandos (SDF-Format)').

Die Verknüpfung wird hergestellt durch die Angabe eines **Kettungsnamens** sowohl im Programm als auch in der CLT. Folgende Werte können in dieser Weise aktualisiert werden:

- **für eine DCAM-Anwendung:**

- der Name der DCAM-Anwendung
- das Kennwort für den Anschluß einer Sekundärtask an eine Anwendung, vorgegeben in der Primärtask
- das Kennwort für den Anschluß einer Sekundärtask an eine Anwendung, angegeben in der Sekundärtask



und zusätzlich für eine DCAM(NEA)-Transport-Service-Anwendung:

- der Verteilungsname
- das Kennwort zum Aufbau einer Verbindung, vorgegeben in der Primärtask

- **für die Verbindung:**

- der Name des Partners
- der Name des Prozessorknotens des Partners
- die Begleitinformation



und zusätzlich für DCAM(NEA)-Transport-Service-Anwendungen das Kennwort zum Aufbau einer Verbindung, angegeben von der Task, die die Aufforderung dazu abgibt.

Bei der Ausführung von YOPEN, bzw. YOPNCON werden die Angaben in Steuerblöcke oder Datenstrukturen von DCAM übernommen. Sie werden, soweit Einträge in der CLT vorliegen, um diese Werte aktualisiert. Wurden keine Einträge gemacht, bleiben sie unverändert. Danach ist eine dynamische Namen-Zuweisung für diese Anwendung oder Verbindung erst möglich, wenn sie wieder geschlossen wurde. Sollen Einträge in der CLT gelöscht werden, genügt es, die Makros YAPPL bzw. YCONN oder die entsprechenden Kommandos nur mit dem Kettungsnamen anzugeben.

4 Unterstützung logischer Datenstationen



Dieses Kapitel gilt nur für DCAM(NEA)-Transport-Service-Anwendungen

Eine logische Datenstation ist die Modellvorstellung einer Datenstation mit bestimmten Standard-Eigenschaften. Im Gegensatz zu physikalischen Datenstationen erlaubt die logische Datenstation eine Programmierung, die von den physikalischen Eigenschaften der angeschlossenen Datenstation unabhängig ist. D.h., für den Anwender ist es unwichtig, auf welche physikalische Datenstation er Nachrichten ausgeben will, für ihn gibt es nur eine logische Datenstation.

Es gibt zwei Arten logischer Datenstationen:

- die Format-Datenstation
- die logische Zeilendatenstation

Format-Datenstation

Bei der Format-Datenstation wird ein Format (=Formular oder Maske) an die Datenstation ausgegeben. Der Anwender füllt nur die vorgegebenen Felder aus. Die Formatdatenstation bietet dem Programm strukturierte Formulare mit einer Feldstruktur. Die Format-Datenstation wird neben dem über VTSU einstellbaren eingeschränkten Formatbetrieb auch durch das Softwareprodukt FHS (Format Handling System) angeboten.

Ausführliche Informationen zu FHS finden Sie im Handbuch 'FHS'.

Logische Zeilendatenstation

Die logische Zeilendatenstation wird durch das Software-Produkt VTSU (Virtual Terminal Support) im Betriebssystem BS2000 realisiert. Bei einer logischen Zeilendatenstation wird eine Ausgabe in mehrere Zeilen aufgeteilt. VTSU bietet Ihnen für Ihre Anwendungen sowohl Assembler-Makros wie auch COBOL-Datenstrukturen an. Über diese Makros und Datenstrukturen können Sie:

- Empfangsfelder oder symbolische Feldnamen für Statusinformationen erzeugen. Diese Statusinformationen können Sie dann über den Aufruf YINQUIRE abfragen.
- logische Steuerzeichen in Gerätesteuerzeichen umsetzen. Dafür stehen der Makroaufruf VTCSET bzw. das Copy-Element TIAMCTRC zur Verfügung.
- VTSU-Parameter einstellen. Mit diesen VTSU-Parametern wird die Art der logischen Datenstation und die Behandlung der Nachricht beim Senden und Empfangen eingestellt. Die Parameter können Sie über den VTSU-Control-Block (VTSUCB) bei den Aufrufen YSEND, YRECEIVE und YSENDREC angeben.

Ausführliche Informationen zu VTSU finden Sie im Handbuch 'VTSU'.

Die Statusinformationen über die Partnercharakteristika für COBOL-Anwendungen sind im Handbuch 'DCAM COBOL-Aufrufe' beschrieben.

Edit-Optionen

Die Edit-Optionen, bei Verbindungen mit EDIT=SYSTEM, werden durch die Angaben im VTSUCB ersetzt. Mit den Edit-Optionen können Sie wie im VTSUCB die Art der logischen Datenstation und die Behandlung der Nachricht beim Senden und Empfangen einstellen. Jedoch ist der Funktionsumfang des VTSUCB größer, da neue Edit-Optionen nur noch über den VTSUCB zur Verfügung gestellt werden.

5 DCAM-Programme

Dieses Kapitel beschreibt die sprachspezifischen Eigenheiten der DCAM-Schnittstelle. DCAM-Programme können in Assembler oder COBOL erstellt werden. Die genaue Anleitung zur Codierung der DCAM-Programme finden Sie in den Handbüchern 'DCAM Makroaufrufe' für Assembler und 'DCAM COBOL-Aufrufe' für COBOL.

5.1 Assembler-Programme

Die Assembler-Programmschnittstelle bietet den vollen Umfang aller DCAM-Funktionen, wie sie bereits detailliert beschrieben wurden.

In diesem Abschnitt soll ein Überblick über die spezifischen Merkmale der Schnittstelle gegeben werden.

5.1.1 Makroaufrufe und Steuerblöcke

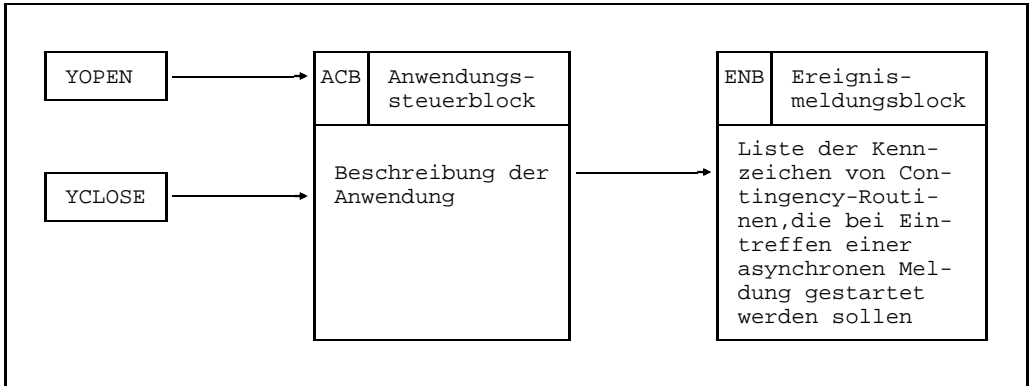
Alle DCAM-Funktionen werden durch Makroaufrufe ausgelöst. Die Aktionsmakros beziehen sich auf Steuerblöcke, in denen alle Parameter hinterlegt werden. Es werden zwei Gruppen von Aufrufen unterschieden, je nach Typ des Steuerblocks, auf den sie sich beziehen:

- Makroaufrufe, die sich auf einen **Anwendungssteuerblock (ACB application control block)** beziehen.
- Makroaufrufe, die sich auf einen **Anforderungsparameterblock (RPB request parameter block)** beziehen.

Der **ACB** enthält alle Informationen über die DCAM-Anwendung. Auf ihn beziehen sich die Makroaufrufe:

- **YOPEN** (DCAM-Anwendung eröffnen)
- **YCLOSE** (DCAM-Anwendung schließen)


Sollen asynchrone DCAM-Meldungen verarbeitet werden, wird zusätzlich ein **Ereignis-Meldungsblock (ENB event notification block)** benötigt. In ihm werden Verweise auf die Kennzeichen für die Contingency-Routinen hinterlegt (siehe auch Seite 111).



ACB-bezogene Makroaufrufe


Der **RPB** enthält die aktuellen Parameterwerte der Aktionsaufrufe. Auf ihn beziehen sich die Makroaufrufe:

- **YINQUIRE** Einträge über DCAM-Anwendungen, Partner und Verbindungen abfragen
- **YOPNCON** Verbindung aufbauen
- **YCLSCON** Verbindung abbauen
- **YREJLOG** Aufforderung zum Verbindungsaufbau zurückweisen
- **YCHANGE** Eigenschaften einer Verbindung ändern
- **YSEND** Nachricht senden
- **YRECEIVE** Nachricht empfangen
- **YSENDREC** Nachricht senden und neue Nachricht empfangen
- **YRESET** Empfangsaufrufe zurücknehmen und CS/CA-Zustand ändern

 und zusätzlich für DCAM(NEA)-Transport-Service-Anwendungen:

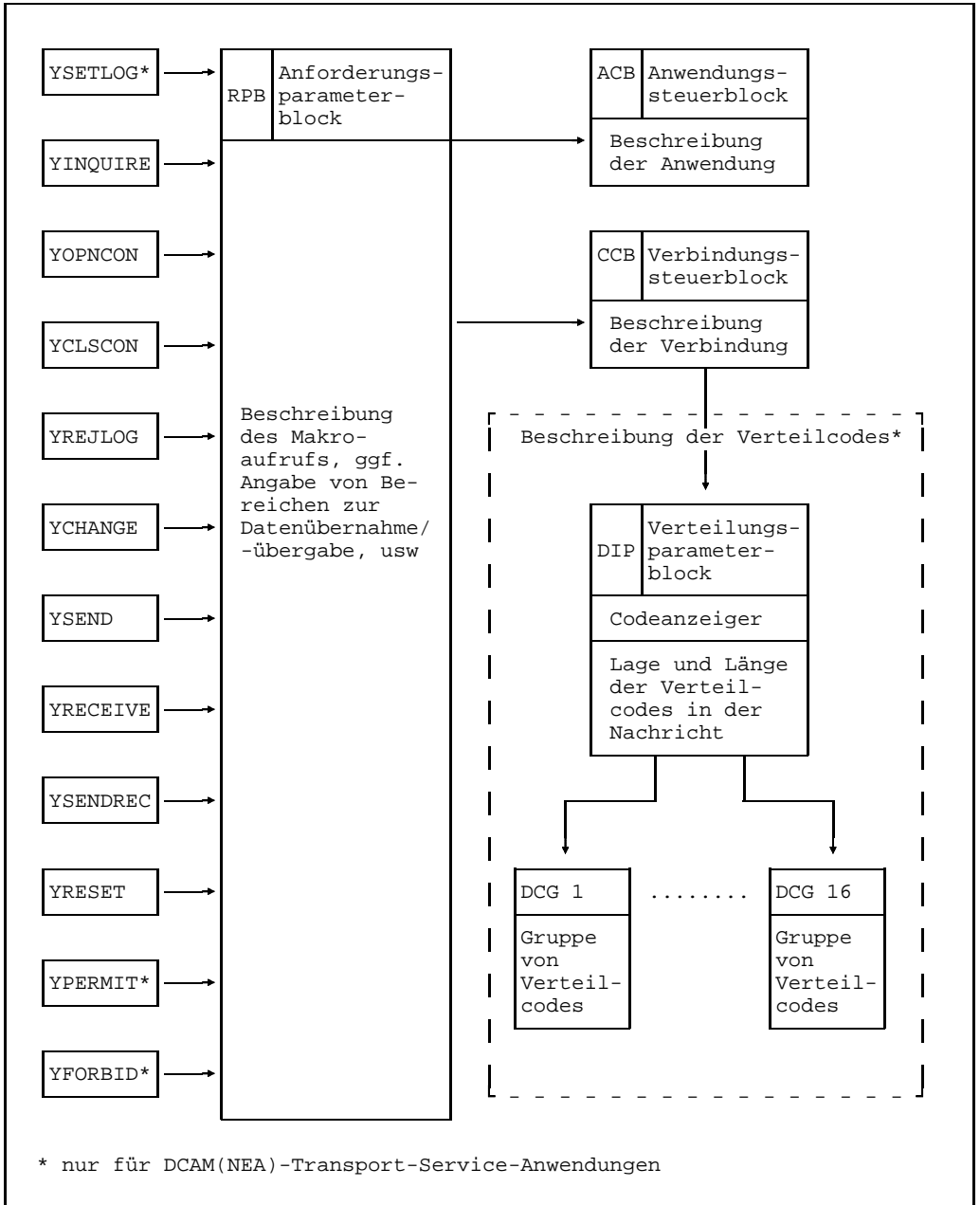
- **YSETLOG** Zustand einer DCAM-Anwendung ändern
- **YPERMIT** Empfang mit Verteilcode ermöglichen
- **YFORBID** Empfang mit Verteilcode verbieten

Zusätzlich zum RPB wird für verschiedene Aufrufe der **Verbindungssteuerblock (CCB connection control block)** benötigt. Im CCB stehen die Operanden für eine Verbindung.

 Für DCAM(NEA)-Transport-Service-Anwendungen sind Lage und Länge der Verteilcodes, der Codeanzeiger und die Verweise auf zugeordnete Verteilcodegruppen im **Verteilungsparameter-Block (DIP distribution parameter block)** und die Verteilcodes selbst im **Verteilcode-Gruppenblock (DCG distribution code group block)** beschrieben.

Hinweis

DCAM-Anwender, die über eine X.25-Schnittstelle an ein Paketvermittlungsnetz (z.B. Datex-P) angeschlossen sind, müssen bestimmte Einschränkungen bei den Aufrufen, die sich auf Verbindungen oder auf die Datenübermittlung beziehen, berücksichtigen. Sie sind beschrieben im Handbuch 'DCAM Makroaufrufe' Abschnitt 'Auswirkungen der CCITT-Empfehlung X.25 auf die DCAM-Benutzerschnittstelle'.



RPB-bezogene Makroaufrufe

Der Programmierer braucht den internen Aufbau der Steuerblöcke nicht zu kennen, denn es stehen ihm Makroaufrufe zur Verfügung, die Steuerblöcke erzeugen und behandeln.

Makroaufrufe zur **statischen Steuerblockerzeugung**:

- **YACB** ACB erzeugen
- **YENB** ENB erzeugen
- **YRPB** RPB erzeugen
- **YCCB** CCB erzeugen



und zusätzlich für DCAM(NEA)-Transport-Service-Anwendungen:

- **YDIP** DIP erzeugen
- **YDCG** DCG erzeugen

Mit diesen Makroaufrufen werden Steuerblöcke bei der Assemblierung erzeugt.

Die Steuerblöcke können auch direkt durch den Anwender erzeugt und behandelt werden und zwar mit den Makroaufrufen YDDACB, YDDCCB, YDDENB und YDDRPB;



und zusätzlich für DCAM(NEA)-Transport-Service-Anwendungen mit YDDDCG und YDDDIP.

Diese Makroaufrufe beschreiben das Layout (DSECT und CSECT) der Steuerblöcke.

Zur **dynamischen Erzeugung** eines Steuerblocks während des Programmablaufs im Speicherbereich des Anwenders (Klasse-6-Speicher) oder wahlweise in einem Bereich außerhalb des Anwendungsprogramms (Klasse-5-Speicher) steht der Makroaufruf **YGENCB** (Steuerblock erzeugen) zur Verfügung.

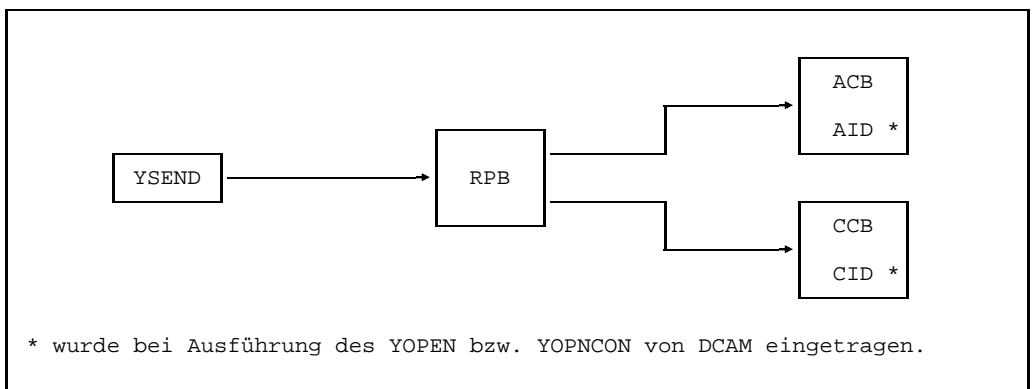
Für die **Behandlung der Steuerblöcke** gibt es die Aufrufe

- **YMODCB** Inhalt von Steuerblockfeldern ändern
- **YSHOWCB** Inhalte bezeichneter Steuerblockfelder in einen, im Programm reservierten Bereich, übertragen
- **YTESTCB** Steuerblockfeldinhalte mit vorgegebenen Werten vergleichen

Die Steuerblöcke können mit Hilfe dieser Makroaufrufe nur nach erfolgreichem Laden des DCAM-Subsystems erzeugt und behandelt werden. Ebenso darf sich das DCAM-Subsystem nicht im HOLD/DELETE-Status befinden. Wenn ein Task bereits vor dem HOLD/DELETE-Status ein DCAM-Kommando oder einen DCAM-Aufruf erfolgreich abgesetzt hat, kann er trotz /HOLD-Subsystem oder /DELETE-Subsystem bis Taskende mit DCAM arbeiten (gilt auch für %).

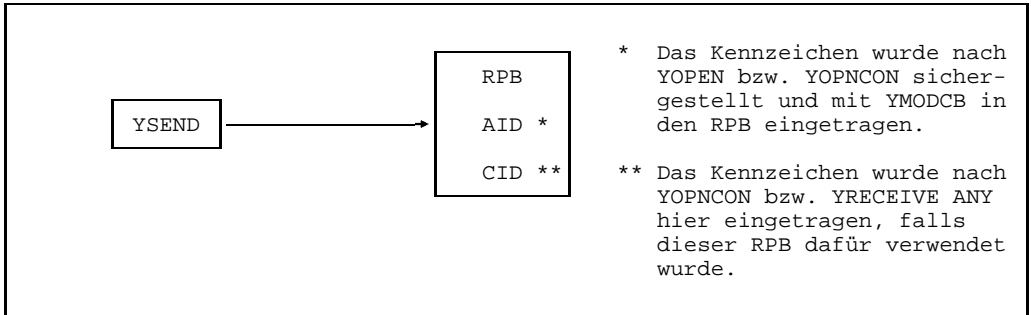
Der RPB kann auch mit jedem Makroaufruf, der ihn adressiert, geändert werden (implizite RPB-Modifikation). Es bleibt dem Anwender überlassen, ob er mehrere RPB-Steuerblöcke anlegt oder jeweils denselben adressiert, wobei er die Feldinhalte aktualisiert.

Mit dem Aufruf YOPEN werden die im ACB beschriebenen Eigenschaften der Anwendung an DCAM übergeben. Nach Ausführung des YOPEN gibt DCAM dafür im ACB ein **Kennzeichen** zurück, den **AID (application identifier)**. Alle Aufrufe, die sich auf diese Anwendung beziehen, können nun die Adresse des Steuerblocks ACB verwenden, in dem das gültige Kennzeichen steht.



Beispiel für Adressierung von Steuerblöcken ohne Verwendung der Kennzeichen

Es ist aber auch möglich, das Kennzeichen AID sicherzustellen und in den für weitere Aufrufe verwendeten RPB einzusetzen.



Beispiel für Adressierung eines RPB, in den gültige Kennzeichen eingetragen wurden

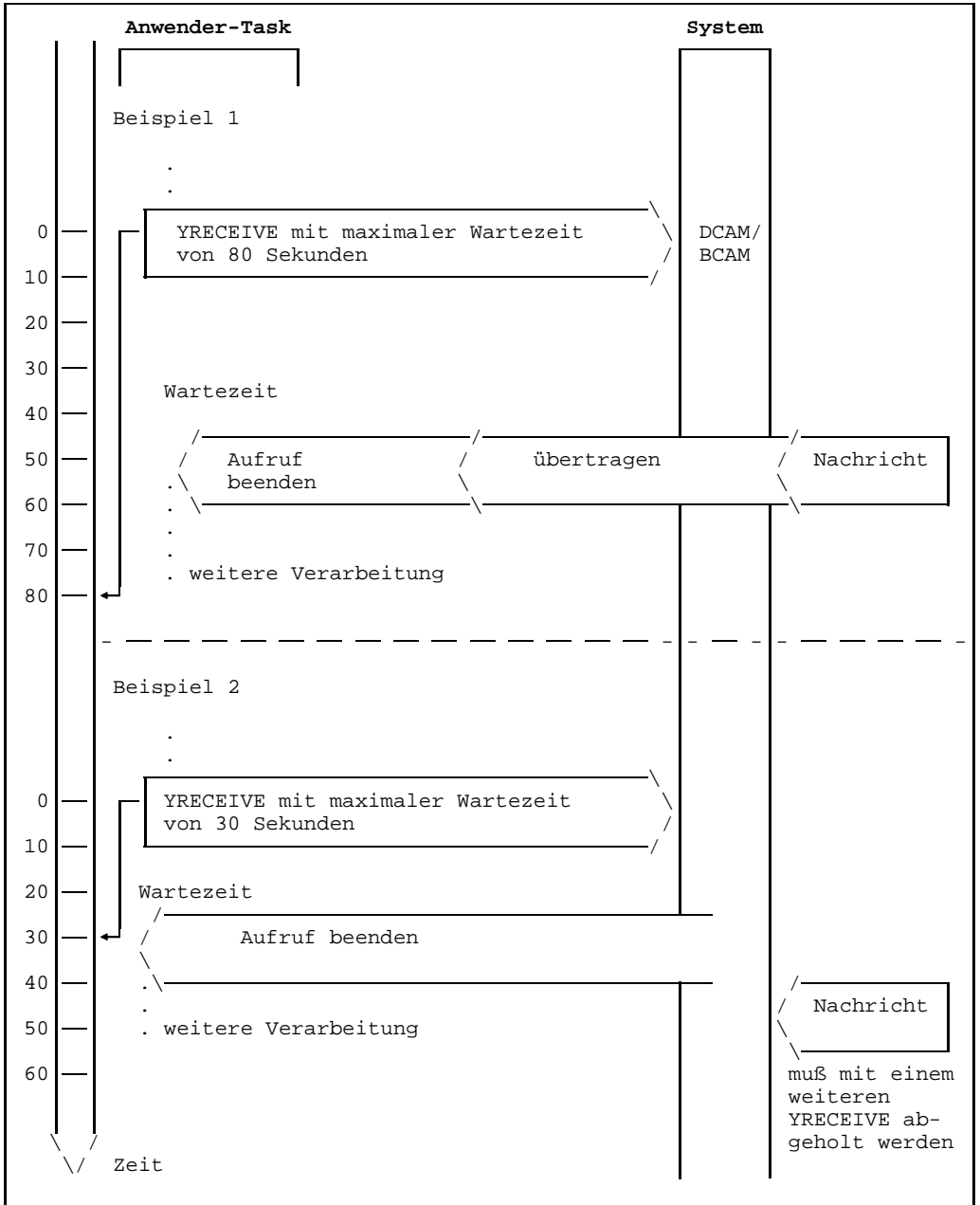
In diesem Fall kann der Speicherbereich für den ACB anderweitig genutzt werden. Das gleiche gilt für die Beschreibung der Verbindung im CCB, und für DCAM(NEA)-Transport-Service-Anwendungen auch im DIP und DCG. Auch nach einem YOPNCON kann z.B. das Kennzeichen **CID (connection identifier)**, das im CCB und RPB zurückgegeben wird, benutzt werden. YSEND und YRECEIVE bzw. YSENDREC benötigen nur noch Steuerblöcke RPB, in denen entsprechende Kennzeichen AID und CID eingetragen wurden. Wenn der Eintrag nicht schon durch DCAM erfolgte, kann er jederzeit mit YMODCB erfolgen, vorausgesetzt, das Kennzeichen wurde vorher mit YSHOWCB sichergestellt.

Die Verwendung der Kennzeichen hat den Vorteil, daß DCAM alle für den Aufruf nötigen Informationen im RPB findet. Damit wird die Verarbeitung beschleunigt.

5.1.2 Synchrone Ausführung der Befehle an DCAM

Bei synchroner Ausführung der Befehle an DCAM erhält das Programm von der Taskverwaltung die Steuerung erst dann wieder zurück, wenn dieser Aufruf von DCAM bearbeitet worden ist. Kann DCAM den Aufruf sofort bearbeiten, entstehen keine Wartezeiten. Bei einigen Funktionen kann es allerdings sein, daß Wartezeiten entstehen, da auf eine Reaktion des Kommunikationspartners gewartet werden muß. Beim Aufbau einer Verbindung (YONCON) muß der Kommunikationspartner seinerseits die Verbindung aufgebaut haben, und beim Empfang von Nachrichten (YRECEIVE) muß der Partner diese vorher gesendet haben. Sie müssen im Datenspeicher des Kommunikationssystems zur Verfügung stehen und in einer Warteschlange eingetragen sein.

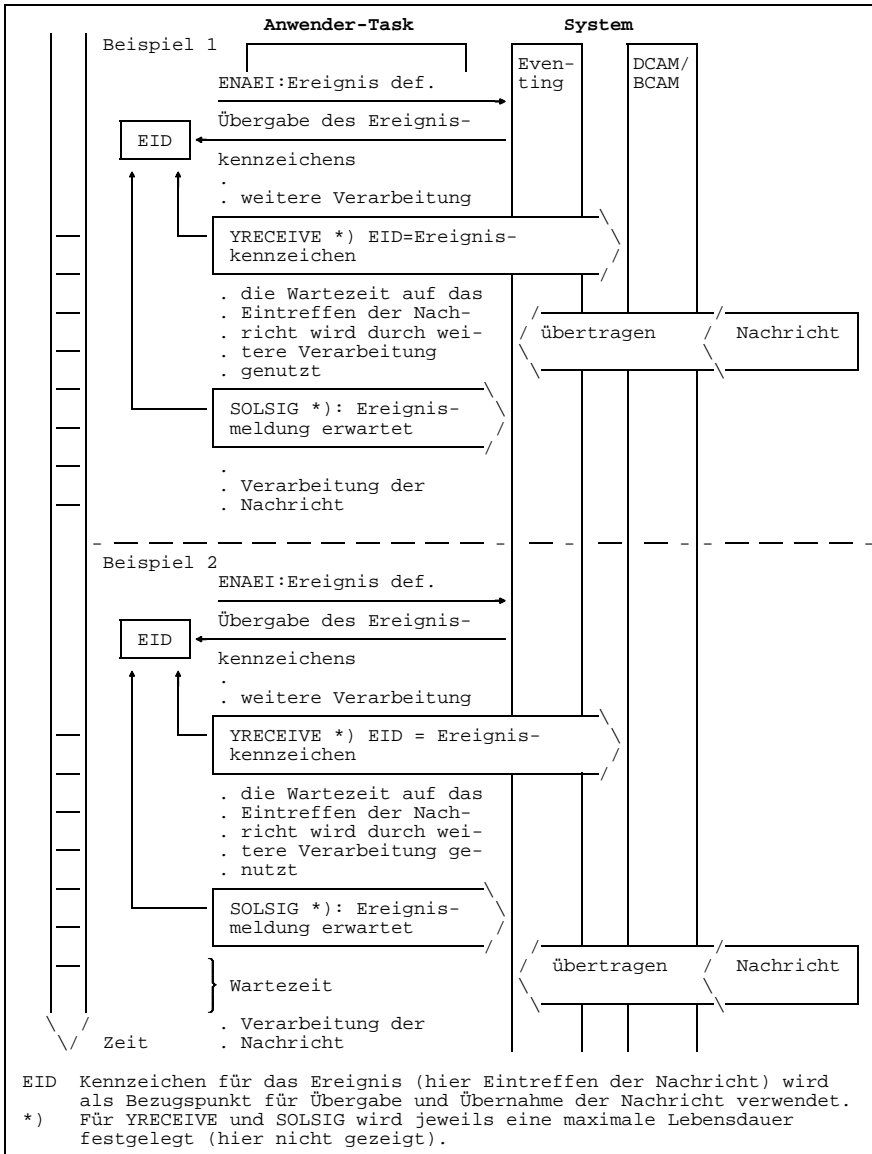
Der Programmierer kann entscheiden, ob er eine bestimmte Zeit wartet bis der Aufruf ausgeführt werden kann. Er legt die maximale Wartezeit für jeden Aufruf fest. Er kann aber auch festlegen, daß ein Aufruf sofort beendet wird, auch wenn der Befehl nicht ausgeführt werden kann. Im Programm muß der Aufruf dann zu einem anderen Zeitpunkt ggf. wiederholt werden.



Beispiele für synchrone Ausführung des YRECEIVE

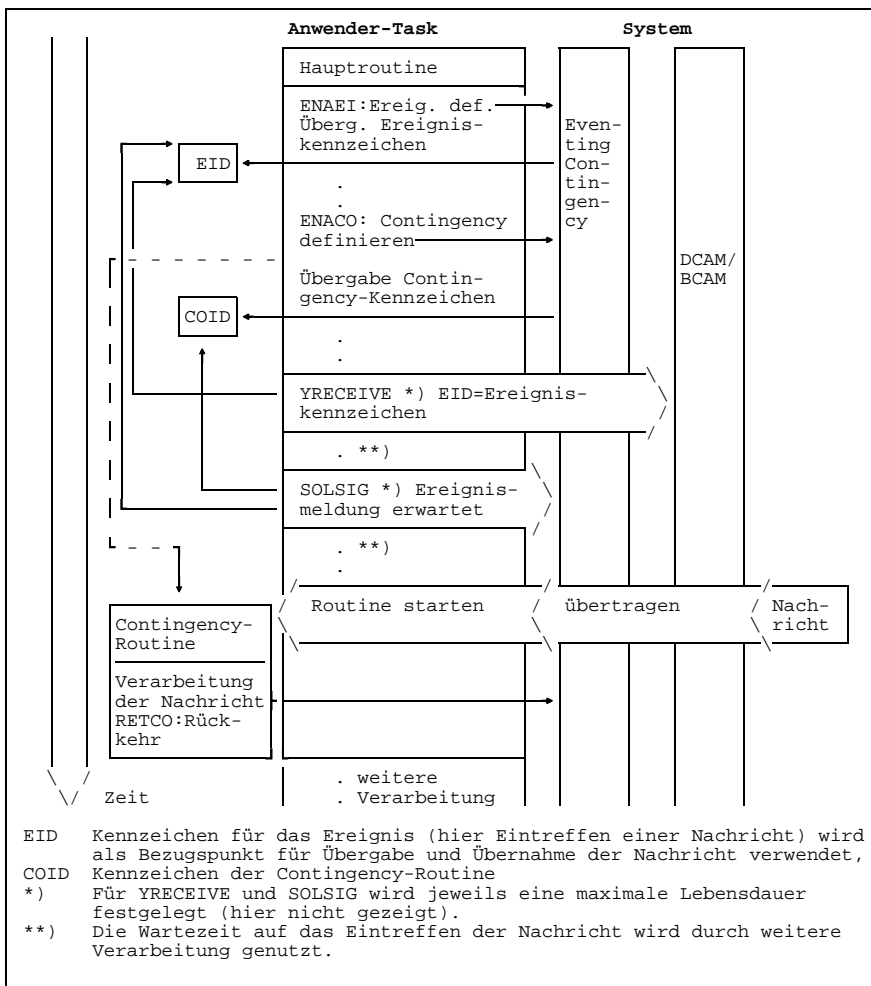
5.1.3 Asynchrone Ausführung der Befehle an DCAM

Bei der asynchronen Ausführung der Befehle erhält das Programm die Steuerung sofort zurück, wenn DCAM den Befehl angenommen hat. Damit kann die Wartezeit bis zur Ausführung des Befehls für anderweitige Verarbeitung genutzt werden. Für die asynchrone Ausführung der Befehle definiert der Anwender ein Ereignis (Makroaufruf ENAEI) und gibt dessen Kennzeichen EID bei dem Aufruf YOPNCON, YRECEIVE oder YSENDREC an. Ein Ereignis ist z.B. die Nachrichtenübergabe. Der Anwender kann dieses Ereignis an der Stelle im Programm abfragen (Makroaufruf SOLSIG), an der er z.B. die Nachricht verarbeiten möchte. Dort kann das Programm ggf. eine bestimmte Zeit warten, falls die Nachricht noch nicht eingetroffen ist.



Beispiele für asynchrone Ausführung des YRECEIVE

Der Anwender kann aber auch mit dem Ereignis eine Contingency-Routine verbinden. Dazu muß eine Contingency-Routine definiert sein und ihr Kennzeichen COID muß zur Verfügung stehen (Makroaufruf ENACO). In diesem Fall ist es zweckmäßig, das Ereignis abzufragen, sobald der Aufruf an DCAM abgeschlossen ist. Trifft das Ereignis ein, wird automatisch die Routine gestartet.



Beispiel mit Contingency-Routine

Zu beachten ist, daß für die einzelnen Aufrufe eine maximale Lebensdauer zu definieren ist, die aufeinander abgestimmt sein muß. Die Lebensdauer des SOLSIG sollte größer sein als die des Befehls an DCAM. Sonst muß der SOLSIG ggf. wiederholt gegeben werden.

Es sollte verhindert werden, daß im Programm bereits Maßnahmen getroffen werden, die von einem Abschluß der asynchronen Verarbeitung ausgehen, ohne daß diese jedoch erfolgt ist (ein Abbau der Verbindung z.B. bei asynchroner Verarbeitung eines YRECEIVE).

Einschränkung:

Maximal können 8 Befehle eines Typs gleichzeitig asynchron bearbeitet werden (siehe Seite 138).

Um bei asynchronen Befehlen auch nach Eintreffen des Ereignisses noch auf Werte zurückgreifen zu können, die bei Absetzen des Aufrufs gültig waren, oder um die Adresse des verwendeten Steuerblocks zu übergeben, kann der Benutzer eine Ereignisinformation festlegen.

Dies geschieht beim Absetzen des YOPNCON, YRECEIVE oder YSENDREC. Sie wird wieder übergeben, wenn das Ereignis eingetroffen ist, entweder in einem definierten Feld (ohne Contingency) oder in einem Register (in der Contingency).

5.1.4 Rückmeldung

Wurde ein Befehl an DCAM ausgeführt, so erhält der Anwender nach Abschluß die Rückmeldung. Sie besteht aus einem 4 byte langen Code der in einem Feld des Steuerblocks ACB bzw. RPB hinterlegt wird und zwar im FDBK-Feld. Zusätzlich wird er in Register 15 übergeben.

FDBK/Register 15			
Byte1 (höchstwertiges)	Byte2	Byte3	Byte4
FDB1: Zusammenfassung der Rückmeldung (summary)	FDB2: Grund einer Zurückweisung (error code)	FDB3: Anzeigen	FDB4: Datenanzeigen

Die Rückmeldung gliedert sich in mehrere Bytes. Die Auswertung des Bytes 1 (FDB 1) erlaubt es schon dem Anwender, grob zu beurteilen, was ggf. getan werden muß. Das Byte 2 (FDB 2) kann er wahlweise auswerten, oder sich mit der Protokollierung begnügen, die eine spätere Auswertung erlaubt. Die jeweilige Problemstellung wird letztlich den Ausschlag geben, wie hier verfahren wird.

Die Bytes 3-4 (FDB 3-4) enthalten Zusatzangaben, wie z.B. letzte Dateneinheit einer Nachricht eingetroffen.

Die Rückmeldungen stehen immer in den FDBK-Feldern der entsprechenden Steuerblöcke.

Bei einem **synchronen Befehl** enthält die Rückmeldung Angaben über die Ausführung, bzw. Nichtannahme.

Bei einem **asynchronen Befehl** sind nach Beendigung des Aufrufs zunächst in Register 15 Angaben über Annahme oder Nichtannahme des Befehls verfügbar. Register 15 kann noch keine Werte enthalten, die die Ausführung des Befehls betreffen. Sie sind erst nach Befehlsausführung im RPB-Steuerblock zu finden. Das FDBK-Feld des RPB-Steuerblocks kann bereits direkt nach Absetzen des asynchronen Befehls Werte enthalten, die die Ausführung des Befehls betreffen, z.B. ist schon eine Nachricht in der Eingabewarteschlange, wenn der asynchrone YRECEIVE abgesetzt wird, oder es enthält wie Register 15 Angaben über Annahme oder Nichtannahme.

Der Anwender kann den RPB wieder auffinden, indem er beim asynchronen Aufruf die RPB-Adresse im Feld 'EIDREF' übergibt. DCAM überschreibt das erste Byte von EIDREF mit X'0C'. Daher sollte im 31-bit-Modus für Adressen EIDREF2 verwendet werden. Das Feld RPOSTAD muß dann 8 Byte lang sein und beim SOLSIG muß "RPOSTL=2" angegeben werden. Nach einem SOLSIG erhält der Anwender die RPB-Adresse über das Feld 'RPOSTAD', bzw. Register 3 der Contingency zurück. So kann er auf den verwendeten RPB und mit YSHOWCB auf die Rückmeldung im FDBK-Feld zurückgreifen. Da der asynchrone Befehl bereits nach dem DCAM-Aufruf ausgeführt sein kann, sollte zur Überprüfung der Annahme bzw. Nichtannahme nur Register 15 ausgewertet werden.

In der Rückmeldung des Aufrufs SOLSIG (Register 15) bzw. im Register 2 der Contingency erhält der Anwender keine Angaben über die Befehlsausführung durch DCAM, sondern nur über den SOLSIG-Aufruf; d.h., nur Angaben darüber, ob das erwartete Ereignis eingetreten ist oder ob ein Fehler oder Zeitüberschreitung auftrat (vgl. SOLSIG im Handbuch 'BS2000 Makroaufrufe an den Ablaufteil').

Bei den Aufrufen YOPNCON und YRECEIVE werden, wenn die Rückmeldung keinen Fehler anzeigt (d.h. FDB1 = X'00'), noch weitere Informationen übergeben (siehe dazu entsprechende Abschnitte der Funktionsbeschreibung).

5.1.5 Asynchrone DCAM-Meldungen

Für eine Reihe von Ereignissen im Datenkommunikationssystem, die unabhängig vom Verarbeitungsablauf des Programms auftreten können, die ferner die Verarbeitung unter Umständen entscheidend beeinflussen, können von DCAM asynchrone Meldungen erzeugt und der Task übergeben werden (siehe nachfolgende Tabelle).

Meldung	wird von DCAM gegeben	ist bezogen auf
LOGON	nur an die Primärtask	die Verbindungsfunktion
PROCON		
SECOND		Existenzfunktion
LOSCON		Verbindungsfunktion
COMEND	an Primär- oder Sekundärtask	Existenzfunktion
EXPR *		Datenübermittlungsfunktion
TACK *		

* nur für DCAM(NEA)-Transport-Service-Anwendungen

Ereignisse sind:

LOGON

Eine Aufforderung zum Aufbau einer Verbindung ist eingetroffen.

LOSCON

Eine Verbindung wurde vom Kommunikationspartner, dem Operator oder durch Fehler abgebaut.

Der Abbau einer Verbindung steht bevor; Warnung (siehe /BCON- und /BCTIMES-Kommando im Handbuch 'Netzmanagement im BS2000').

Hinweis

Wenn das LOSCON-Ereignis eintritt und keine Warnung erfolgt, ist die Verbindung schon abgebaut; gibt der Anwender jetzt den Makroaufruf YCLSCON ein, wird er mit einem Returncode $\neq 0$ abgelehnt.

PROCON

Bereits bei der Generierung der Datenkommunikationszugriffsmethode (siehe XSTAT-Aufruf im Handbuch 'Generierung eines Datenkommunikationssystems') festgelegte Partner werden beim YOPEN bzw. einem /BCIN-Kommando des Operators zum Verbindungsaufbau vorgeschlagen.

SECOND

Eine Sekundärtask wurde eröffnet oder geschlossen.



für DCAM(NEA)-Transport-Service-Anwendungen gilt:

Der Verteilcode einer Nachricht ist einem Verteilungsnamen zugeordnet, der von keiner Sekundärtask eröffnet ist. Um diese Nachricht selbst zu empfangen, muß die Primärtask erst YFORBID für den Verteilungsnamen geben.

COMEND

Die Kommunikationszugriffsmethode wurde beendet.

Die Beendigung steht bevor; Warnung (siehe /BCTIMES-Kommando im Handbuch 'Netzmanagement im BS2000').

Die DCAM-Anwendung wurde geschlossen; Meldung an Sekundärtask.

Das Schließen der DCAM-Anwendung steht bevor; Warnung an Sekundärtask (siehe /BCTIMES-Kommando im Handbuch 'Netzmanagement im BS2000').



EXPR Eine Expreßnachricht ist eingetroffen.

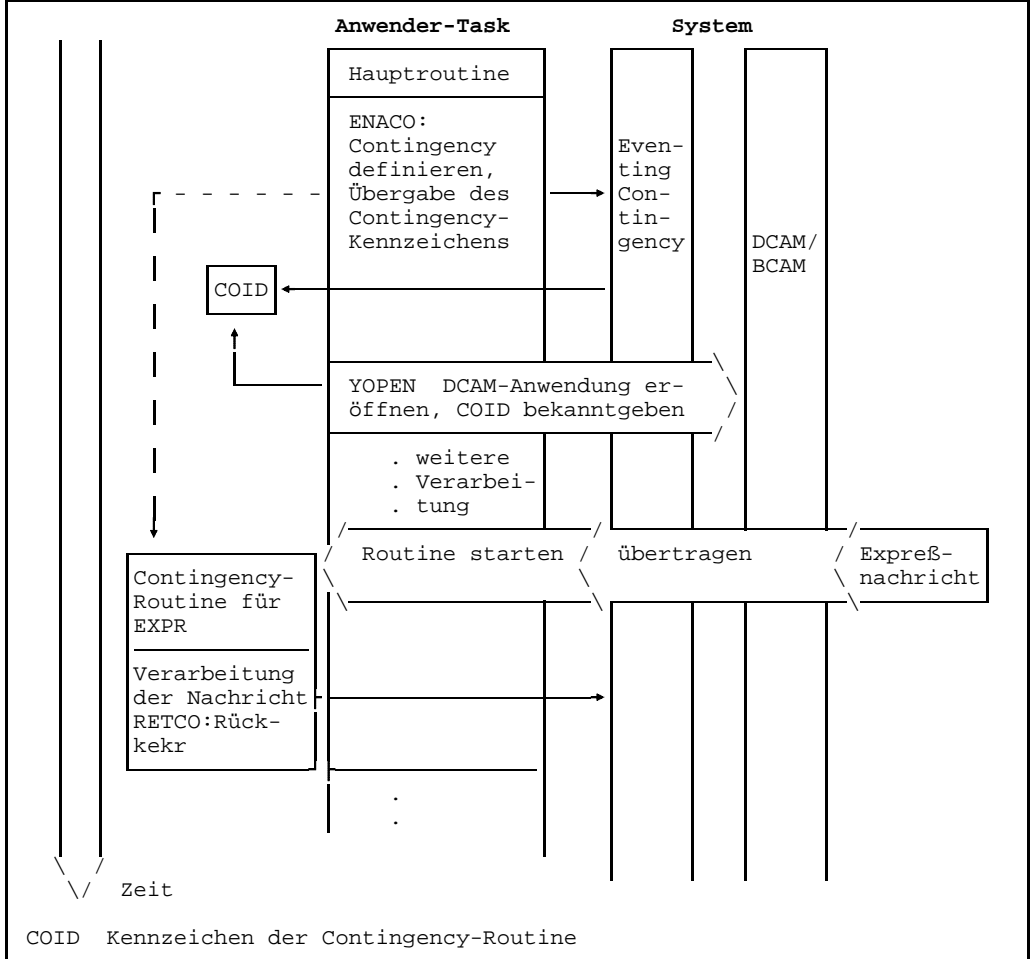


TACK Eine Transportquittung ist eingetroffen.

Die DCAM-Anwendung kann von dem sie betreffenden Ereignis unterrichtet werden, wenn dies bei ihrer Erzeugung/Eröffnung festgelegt wird. Da die Ereignismeldung zum Ablauf einer Contingency führt, muß für eine Meldung, die in Empfang genommen werden soll, eine **Contingency definiert** werden. Dazu ist der **Makro ENACO** zu verwenden (siehe Handbuch 'DCAM-Makroaufrufe'). Das zurückgegebene Kennzeichen (COID) ist über den Ereignissteuerblock ENB (siehe auch Seite 51) an DCAM zu übergeben. Beim YOPEN ist dann für die Lebensdauer einer Anwendung festgelegt, welche Meldungen angenommen werden und welche Contingency jeweils angestoßen werden soll.

Primärtasks erhalten alle Ereignisse gemeldet, Sekundärtasks nur LOSCON, COMEND, und für DCAM(NEA)-Transport-Service-Anwendungen EXPR und TACK. Wird eine Contingency angestoßen, enthalten die Register 1-8 alle Informationen, die für die Verarbeitung eines Ereignisses notwendig sind. Alle weiteren Register enthalten keinen definierten Inhalt. Somit bleibt auch die Versorgung der Basisregister Aufgabe des Anwenders. Der Zugriff auf Registerinhalte der unterbrochenen Routine oder der Hauptroutine kann mit dem Makroaufruf CONTXT erfolgen. Die Prioritäten der Contingency-Routinen werden bei ihrer Definition (ENACO) festgelegt und können mit LEVCO verändert werden.

Auf die LOGON-Meldung wird von DCAM eine Reaktion erwartet. Die Reaktion muß nicht in der Contingency-Routine selbst erfolgen, sondern nur innerhalb eines bestimmten Zeitraums, der beim Start des Kommunikationssystems (/BCTIMES-Kommando) festgelegt wurde. Erfolgt innerhalb diese Zeitraums keine Reaktion, so wird dies als Zurückweisung der Meldung gewertet.



Beispiel für das Empfangen einer Expresßnachricht über eine asynchrone DCAM-Meldung (EXPR), nur für NEA-Transport-Service

Die Entscheidung, bestimmte **Meldungen nicht entgegenzunehmen**, wenn eine DCAM-Anwendung erzeugt bzw. eröffnet wird, hat folgende Konsequenzen:

- **LOGON** ist nicht definiert: Aufforderungen zum Aufbau einer Verbindung seitens der Kommunikationspartner können nur durch einen, im Programmablauf, auf Verdacht abgesetzten, YOPNCON-Aufruf bearbeitet werden.
- **LOSCON** ist nicht definiert: Der Verbindungsverlust ist frühestens der Rückmeldung eines Aufrufs zu entnehmen, der sich auf diese Verbindung bezieht.
- **SECOND** ist nicht definiert: Die Primärtask muß auf anderem Weg, z.B. Eventing, erfahren, daß eine Sekundärtask die Anwendung eröffnet hat und welchen Verteilungsnamen sie hat.



nur für DCAM(NEA)-Transport-Service-Anwendungen:

Die Primärtask erfährt nichts über Nachrichten, die einem Verteilungsnamen ohne angeschlossene Sekundärtask zugeordnet werden. Diese Nachrichten werden von keiner Task empfangen und nach Ablauf einer System-Überwachungszeit gelöscht.

- **PROCON** ist nicht definiert: Vordefinierte Vorschläge zum Aufbau von Verbindungen werden nicht an die DCAM-Anwendung gemeldet.
- **COMEND** ist nicht definiert: Frühestens beim nächsten Aufruf an DCAM wird durch entsprechende Rückmeldung bemerkt, daß die DCAM-Anwendung nicht mehr besteht oder die Kommunikationszugriffsmethode beendet wurde.



EXPR ist nicht definiert: Expreßnachrichten müssen in der Reihenfolge der eintreffenden Nachrichten mit YRECEIVE abgeholt werden.



TACK ist nicht definiert: Transportquittungen müssen in der Reihenfolge der eintreffenden Nachrichten mit YRECEIVE abgeholt werden.

5.1.6 'Reentrant'-fähige Assembler-Programme

Im BS2000 kann ein Programm mehrere Tasks steuern. Dieses Programm ist für diese Anzahl von Tasks nur einmal geladen. Das bedeutet:

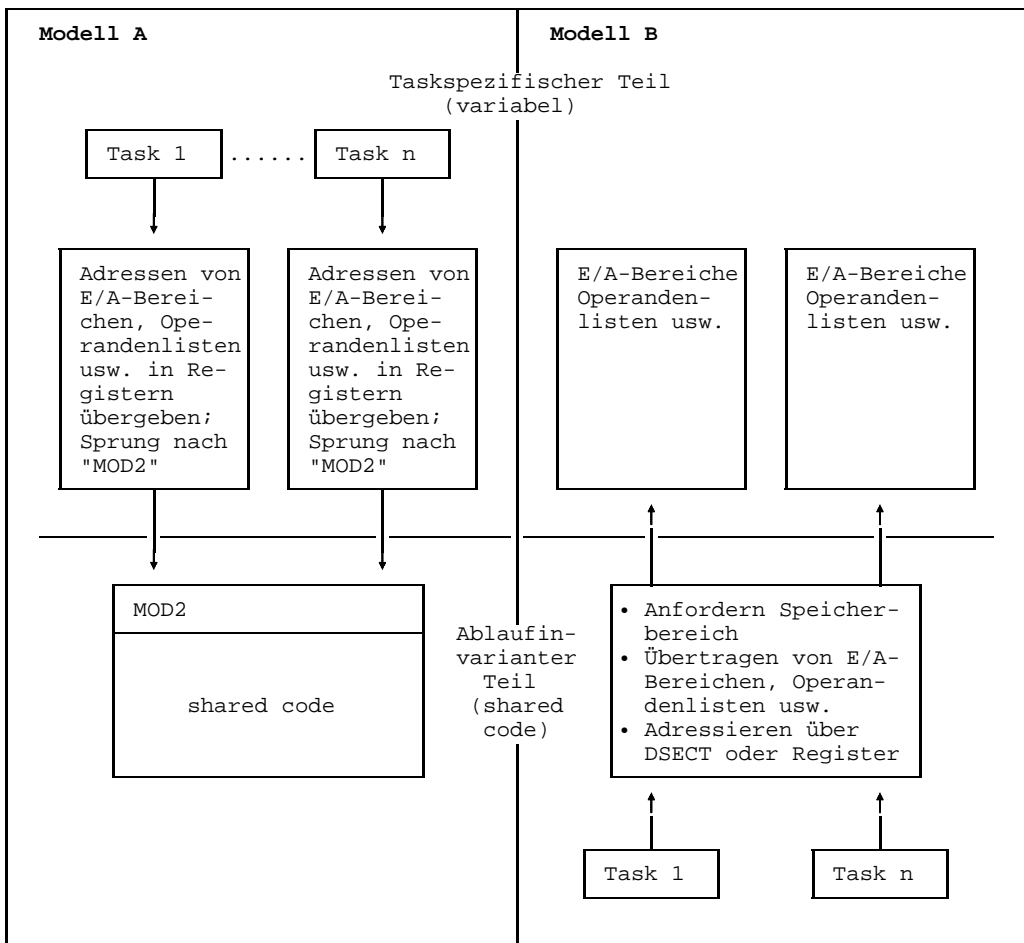
- **Ladezeiten werden verkürzt**, denn das Programm wird geladen, wenn es von der ersten Task aufgerufen wird. Beim Aufruf durch weitere Tasks wird es nicht mehr geladen, sondern auch von ihnen, eben mehrfach, genutzt.
- **Antwortzeiten bei Dialogen werden verkürzt**, denn die DCAM-Anwendung als Kommunikationspartner hat die 'Last' der ankommenden Anfragen auf mehrere Tasks verteilt. Die Tasksteuerung wird im 'time sharing' genutzt, obwohl nur ein Programm dahinter steht (siehe Seite 27).
- Die **Anlagenausnutzung wird verbessert**, denn die Seitenwechselrate (paging) ist geringer, und das Programm wird für mehrere Tasks nur einmal im Haupt- und Seitenspeicher verwaltet.

Bedingung für diese Verarbeitung ist, daß ein solcher Programmmodul, der als '**shared code**' im System verwaltet werden soll, **ablaufinvariant** und damit 'reentrant'-fähig ist. Da aber meist auch taskspezifische Ein-/Ausgabe-Bereiche, usw. benötigt werden, bedeutet dies, daß ein Programm in einen ablaufinvarianten (read only) und einen ablaufvarianten (variablen) Teil zu unterteilen ist. Von DCAM wird dieses Verfahren unterstützt, indem

- **Steuerblöcke** während des Ablaufs in einem von DCAM verwalteten taskspezifischen Bereich oder in einem Anwenderbereich erzeugt werden können.
- bei Aufrufen zur Erzeugung und Behandlung der Steuerblöcke Parameterliste und Befehlsteil des Makros (SVC) durch Verwendung des **MF-Parameters** getrennt erzeugt werden können.
- weitgehend die **Register-Schreibweise** verwendet werden kann.

Die nachfolgende Tabelle zeigt die Möglichkeiten der 'Reentrant'-Programmierung und die erforderliche Aufteilung des Programms.

Programmteil	ablaufinvariant (shared code)	variabel (taskspezifisch)
Befehle	<ul style="list-style-type: none"> - invarianter Befehlscode, ggf. unter Verwendung des EX-Befehls - Adressierung des variablen Teils über Register (MF=E) - ggf. Bereitstellung von Speicher für den variablen Teil (Modell B im folgenden Bild) 	<ul style="list-style-type: none"> - Befehlscode incl. Versorgung von Registern bei Sprung zum ablaufinvarianten Modul (Modell A im folgenden Bild)
Daten und Bereiche	<ul style="list-style-type: none"> - Zahlenkonstanten - Textkonstanten - Adreßdefinition (DSECT) - Operandenlisten (MF=L) 	<ul style="list-style-type: none"> - Ein-/Ausgabebereiche - Rechenfelder - Sicherstellungsbereiche (Register) - Operandenlisten (MF=L)



Modelle für "shared code"-Programmierung

5.2 COBOL-Programme

Für die Verwendung in COBOL-Programmen gibt es eine Reihe von Zugriffsmoduln auf DCAM-Funktionen, die synchrone und asynchrone Verarbeitung erlauben. COBOL-Programme können Primärtasks steuern, sind aber auch zur Steuerung von Sekundärtasks, gerade bei komplexen Problemen geeignet (siehe auch Seite 27).

5.2.1 COBOL-CALLS und Datenstrukturen

Alle Aufrufe an DCAM werden in Form von Unterprogrammssprüngen (CALL...) formuliert, die zugehörigen Datenstrukturen mit den Parametern und Datenfeldern sind in der WORKING STORAGE SECTION zu deklarieren.

Folgende Datenstrukturen sind in ihrem Aufbau vorgegeben:

– **Anwendungsstruktur (A-Struktur):**

Die Anwendungsstruktur enthält die Beschreibung der DCAM-Anwendung. Sie muß mindestens einmal im Programm vorhanden sein.

– **Verbindungsstruktur (V-Struktur):**

Die Verbindungsstruktur enthält die Beschreibung der Verbindung. Sie kann für jede Verbindung gesondert oder für mehrere Verbindungen einmal vorhanden sein.

– **Befehlsstruktur (B-Struktur):**

Die Befehlsstruktur enthält die Beschreibung eines Befehls. Sie kann für jeden Befehl gesondert oder einmal für mehrere Befehle vorhanden sein.

– **Wartestruktur (W-Struktur):**

Die Wartestruktur enthält die Beschreibung von Operanden, die darauf warten, daß asynchrone CALL-Aufrufe beendet sind.



– **VTSU-Control-Block (VTSUCB-Struktur)**

Die VTSUCB-Struktur enthält die VTSU-Parameter (siehe Handbuch 'VTSU') für die Ein- und Ausgabe.



– **Verteilungsstruktur (VTLG-Struktur):**


Die Verteilungsstruktur enthält die Beschreibung der Nachrichtenverteilung über Verteilcodes. Sie ist nur notwendig, wenn der Nachrichtenempfang über Verteilcodes gesteuert werden soll.

 – **Datenstruktur (FHS-Struktur):**

Sie ist nur dann notwendig, wenn die Formatierung der Daten mit dem integrierten FHS-Anschluß erfolgen soll. (Dazu ist das Softwareprodukt FHS ab Version 3.0 Voraussetzung). Die FHS-Moduln müssen in der Anwender-TASKLIB zur Verfügung stehen.

Für die genannten Datenstrukturen stehen folgende COPY-Elemente zur Verfügung:

YDDCUAPL	für die A-Struktur
YDDCUCON	für die V-Struktur
YDDCUCOM	für die B-Struktur
YDDCUWAI	für die W-Struktur

 und zusätzlich für DCAM(NEA)-Transport-Service-Anwendungen:

VTSUCBC	für die VTSUCB-Struktur
YDDCUDIS	für die VTLG-Struktur
FHSMAINP	für die FHS-Parameter

Die folgende Tabelle zeigt, welche Datenstrukturen in den Aufrufen der einzelnen Unterprogramme benützt werden. Bei einigen Aufrufen ist ein zusätzlicher Bereich zur Datenübernahme/-übergabe erforderlich.

Einzelne Aufrufe benötigen noch weitere Bereiche, die hier nicht gezeigt werden (siehe dazu Handbuch 'DCAM-COBOL-Aufrufe').

Funktion	Unterprogramm	Datenstrukturen					VTSUCB	Bereich	FHS-Parameter 1)
		A	V	B	W	VTLG 1)			
Existenzfunktion, Verbindungsfunktion	YOPEN	x							
	YCLOSE	x							
	YOPNCON 2)	x	x	x		[x]			
	YCLSCON 2)	x	x						
	YCHANGE 2)	x	x			[x]			
	YREJLOG YSETLOG 1,2)	x	x		x		x		
Datenübermittlungsfunktion	YSEND	x	x	x			[x]	x	[x]
	YRECEIVE	x	x	x			[x]	x	[x]
	YRSET	x	x	x					
	YPERMIT 1,2)	x				[x]		x	
	YFORBID 1,2)	x						x	
Wartefunktion	YWAIT	x	x	x	x				

1) nur für DCAM(NEA)-Transport-Service-Anwendungen

2) nur in der Primärtask aufrufbar

Die Unterprogramme erfüllen im einzelnen folgende Aufgaben:

YOPEN	DCAM-Anwendung eröffnen
YCLOSE	DCAM-Anwendung schließen
YOPNCON	Verbindung aufbauen
YCLSCON	Verbindung abbauen
YCHANGE	Eigenschaften einer Verbindung ändern
YSEND	Nachricht senden
YRECEIVE	Nachricht empfangen
YREJLOG	Aufforderung zum Verbindungsaufbau zurückweisen
YRESET	Anstehenden YRECEIVE-Aufrufe zurücknehmen
YWAIT	Auf DCAM-Ereignis warten



und zusätzlich für DCAM(NEA)-Transport-Service-Anwendungen:

YSETLOG	Zustand einer DCAM-Anwendung ändern
YPERMIT	Empfang über Verteilcode ermöglichen
YFORBID	Empfang über Verteilcode verbieten

Für die **Statusabfrage** der Existenz- oder Verbindungsfunktion gilt eine besondere Vereinbarung. Sie wird mit **YINQUIRE** aufgerufen. Folgende Abfragen können gemacht werden:

- **Welcher Partner** wünscht als nächster eine Verbindung aufzubauen, ggf. mit Empfang der Verbindungsnachricht im Feld 'LGMSG' (TOP)?
- **Wieviele Partner** wünschen eine Verbindung bzw. mit wievielen besteht eine Verbindung (CNT)?
- **Welchen Zustand** hat eine bestimmte DCAM-Anwendung (APP)?



und zusätzlich für DCAM(NEA)-Transport-Service-Anwendungen:

- **Welche charakteristischen Eigenschaften** hat ein Partner (PTN)?
- **Grundinformationen der Datenstation** (BTI).
- **Beschreibung der Datensichtstation und der Zeichensätze** (MCS).
- **Beschreibung der Peripherie** (POT).

Es wird die A-Struktur und die B-Struktur verwendet und es muß ein Bereich angegeben werden. Dieser Bereich ist in Abhängigkeit von der Funktion des Aufrufs unterschiedlich strukturiert. Die Funktion wird in einem eigenen Feld bestimmt.

Funktion	Unterprogramm	Funktion der YINQUIRE	Datenstrukturen		Bereiche 1)	LGMSG
			A	V		
Statusabfrage der Existenz- und Verbindungsfunktion	YINQUIRE	TOP	X	X	X	[X]
		CNT	X	X	X	
		APP	X	X	X	
		PTN 2)	X	X	X	
		BTI 2)	X	X	X	
		MCS 2)	X	X	X	
		POT 2)	X	X	X	

1) Bereich in Abhängigkeit der Funktion

2) nur für DCAM(NEA)-Transport-Service-Anwendungen

Hinweis

Die Unterprogramme können auch mit einem auf die ersten 4 Buchstaben reduzierten Namen aufgerufen werden, z.B., 'YOPNCON' mit 'YOPN'.

Zur einfacheren Handhabung der Programmierung sollten die Datenstrukturen einmal in einer **Quellcode-Bibliothek (PLAM)** definiert sein. PLAM-Bibliotheken können mit dem Dienstprogramm LMS erstellt und verwaltet werden. Mit der COPY-Anweisung können die gespeicherten Datenstrukturen jeweils ins Programm übernommen und ggf. modifiziert werden.

5.2.2 CALL-Aufrufe ausführen

5.2.2.1 Synchroner Ausführung

Bei der synchronen Verarbeitung wird der nächste Befehl nach einem Unterprogramm-Sprung erst ausgeführt, wenn der DCAM-Aufruf bearbeitet wurde. In Aufrufen, bei denen Wartezeiten entstehen können, kann eine maximale Wartezeit festgelegt werden. Wartezeiten können entstehen z.B bei YOPNCON das Warten auf die Aufforderung zum Verbindungsaufbau bzw. die Annahme des Verbindungswunsches seitens des Partners, bei YRECEIVE die Ankunft der Nachricht aus dem Datenkommunikationssystem. Nach dieser Wartezeit wird der Aufruf beendet. Soll nicht gewartet werden, wird der Aufruf sozusagen auf Verdacht abgesetzt und muß ggf. wiederholt werden.

5.2.2.2 Asynchrone Ausführung

Der Aufbau einer Verbindung und der Empfang von Nachrichten kann zu Wartezeiten führen. Um den Datendurchsatz zu erhöhen, insbesondere, wenn sehr viele Partner bedient werden sollen, können diese Wartezeiten für anderweitige Verarbeitung genutzt werden. Mit dem YWAIT-Aufruf kann der Anwender auf irgendein Ereignis warten. Nach dem Eintreffen des Ereignisses kann er seine Verarbeitung fortsetzen.

Mögliche Ereignisse:

- OPENED der YOPNCON-Auftrag ist beendet
- LETTER der YRECEIVE-Auftrag ist beendet
- GOSIGNAL der Speicherplatz-Engpass ist aufgehoben
- LOSCON die Verbindung wurde durch das System oder durch den Partner abgebaut
- NOEVENT obwohl gewartet wurde, ist kein DCAM-Ereignis eingetroffen

5.2.3 Rückmeldung

Nach Rückkehr aus einer Unterroutine, d.h. nach Ausführung eines DCAM-Aufrufs werden in der jeweils benutzten Datenstruktur (Anwendungsstruktur oder Befehlsstruktur) Rückmeldungen hinterlegt.

Die Rückmeldung ist in 3 Felder aufgeteilt:


- **Rückgabecode**
- **Fehlercode**
- **Indikator**

Der Rückgabecode gibt eine Zusammenfassung der Information wieder, die in Fehlercode und Indikator aufgeschlüsselt ist. Er wird benötigt, um beispielsweise in eine Fehleroutine zu verzweigen.

Der Indikator enthält Zusatzinformationen nach Ausführung des YRECEIVE.

Ferner werden **nach YRECEIVE** folgende Eintragungen in der Befehlsstruktur gemacht:

- die **tatsächliche Länge** der empfangenen Nachricht, auch wenn der Eingabebereich kleiner war.

 und zusätzlich für DCAM(NEA)-Transport-Service-Anwendungen:

- die **laufende Nummer der empfangenen Nachricht**,
- die **laufende Nummer der quittierten Nachricht** wenn eine Transportquittung empfangen wurde.

Bei weiteren Aufrufen (YOPEN, YOPNCON, YWAIT) werden - wenn die Rückmeldung keinen Fehler anzeigt - noch weitere Informationen übergeben (siehe dazu Abschnitt 'Anwendung der Funktionen von DCAM' in der Funktionsbeschreibung).

5.2.4 'Reentrant' -fähige COBOL-Programme

Bedingung für die 'Reentrant'-Fähigkeit eines Codes ist, daß er ablaufinvariant geschrieben wurde (vgl. Seite 115). Nur dann kann er auch als "SHARED CODE" im System verwaltet und benutzt werden.

Die COBOL-Übersetzer erzeugen bei Verwendung der Segmentierung ein Wurzel-Segment (root), das ablaufvariant ist und mehrere Überlagerungssegmente (overlays), die mit wenigen Einschränkungen ablaufinvariant sind. Das Wurzel-Segment enthält die V-Konstanten zum Einbinden sowohl der Segmente, wie auch des COBOL-Laufzeit-Systems (ITC...) und der Moduln, die per "CALL" aufgerufen werden. Dieses Segment erhält den Namen, der in der "PROGRAM-ID." angegeben wurde.

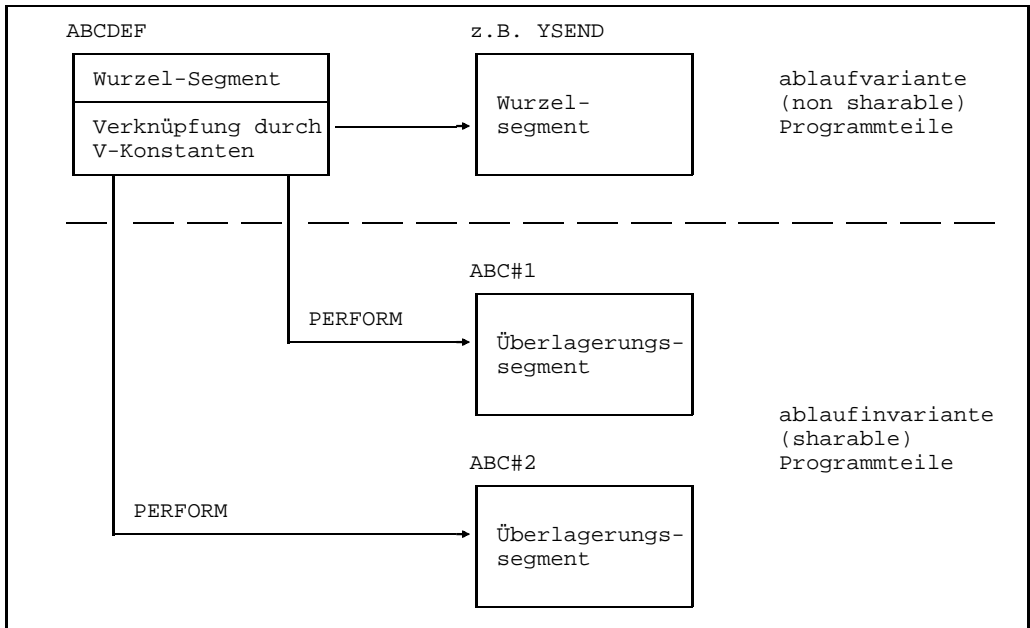
Hinweis

Wird der COBOL85-Compiler verwendet, ist auch das COBOL-Laufzeitsystem ablaufinvariant.

Die Überlagerungssegmente werden vom Wurzelsegment mit "PERFORM" aufgerufen. Ihre Namen setzen sich zusammen aus den ersten 3 Stellen des in der "PROGRAM-ID." angegebenen Namens, einem Nummernzeichen (#) und der Segment-Nummer (z.B. ABC#50).

In den Überlagerungssegmenten dürfen keine DCAM-COBOL-Moduln mit "CALL" aufgerufen werden, da diese ablaufvariant sind.

Bei Verwendung des COBOL85-Übersetzers wird in den Überlagerungssegmenten ablaufinvarianter Code erzeugt.



Segmentierung eines COBOL-Programms

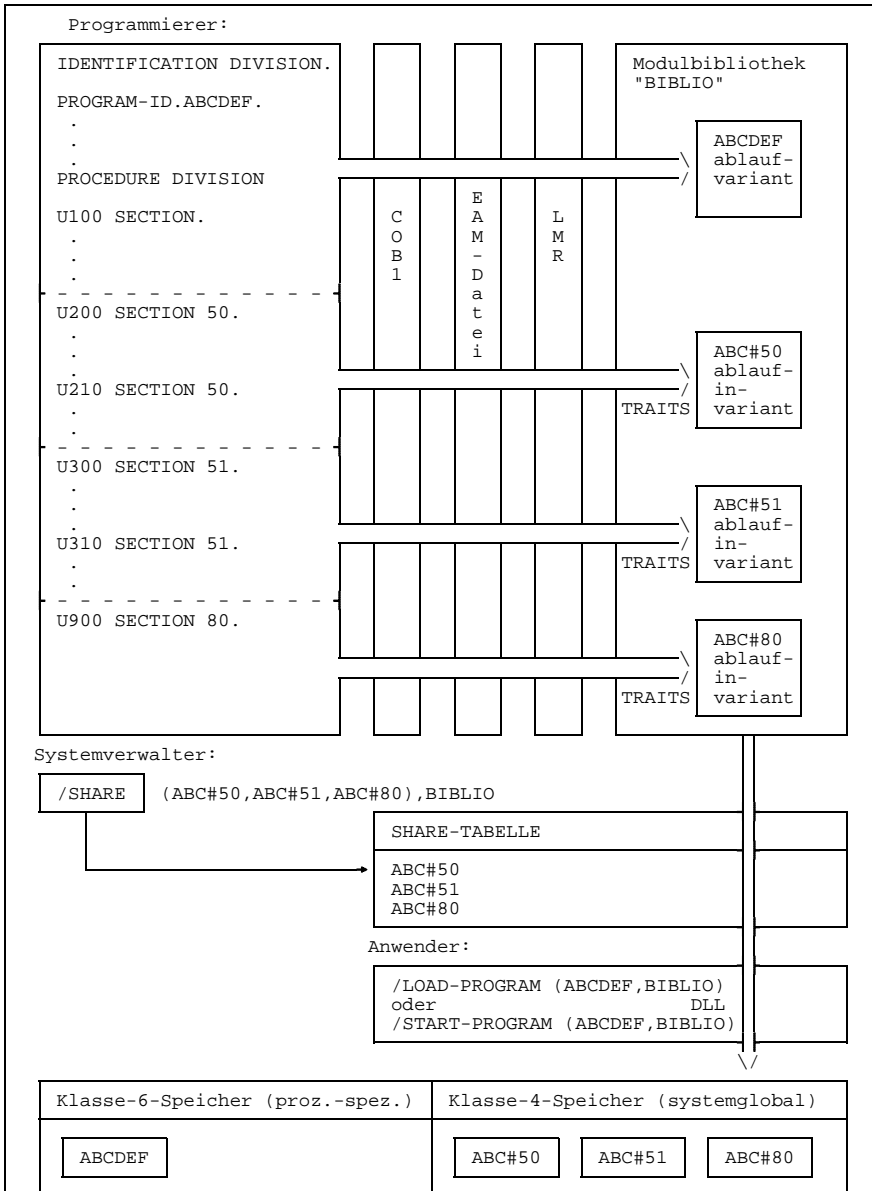
Damit nun die Überlagerungssegmente auf ihre Ablaufinvarianz (Read-only-Eigenschaft) getestet werden können, muß beim Eintrag in eine Modul-Bibliothek mit dem Dienstprogramm LMR die TRAITS-Anweisung verwendet werden. Dies ist deshalb notwendig, weil die COBOL-Übersetzer derzeit diese Eigenschaft nicht explizit prüfen.

Der ANSICOB-Übersetzer erzeugt am Anfang eines Moduls eine Bindesteuerkarte (OVERLAY...), um das Überlagerungssegment zu kennzeichnen. Beim LMR-Lauf werden diese wieder entfernt mit einer entsprechenden Meldung. Für die weitere Bearbeitung mit dem dynamischen Bindelader (DLL) ist das notwendig und daher in Ordnung. Nur der dynamische Bindelader (DLL) kann "SHARED CODE" laden und muß in diesem Fall verwendet werden. Er wird mit

```
/START-PROGRAM (modulname, bibliotheksname) oder
/LOAD-PROGRAM (modulname, bibliotheksname)
```

aufgerufen.

SHARED-Module werden allerdings nur dann in den Klasse 4-Speicher des Systems geladen, wo sie für alle Prozesse zugänglich sind, wenn sie zu Beginn der BS2000-Sitzung vom Systemverwalter in die entsprechende Modultabelle des Systems aufgenommen wurden (siehe Systemüberwachung/ SHARE-Kommando).



Erzeugen von SHARED CODE

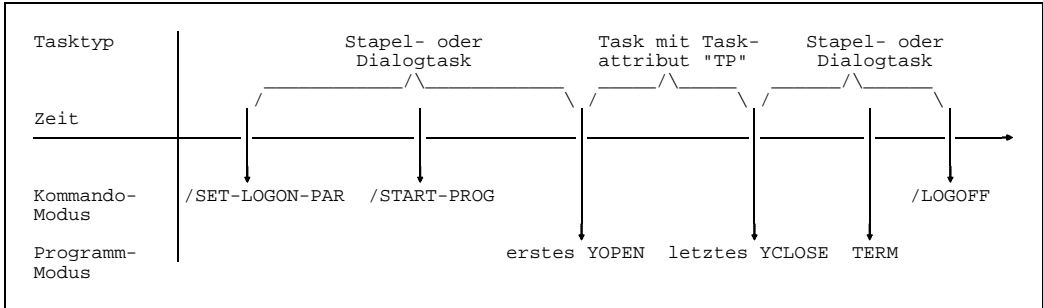
5.2.5 Verwendung weiterer Systemschnittstellen

Im Rahmen der COBOL-Anwendung im BS2000 gibt es für ein DCAM-COBOL-Programm keine Einschränkungen.

Besonders hingewiesen werden soll auf die Möglichkeit bei den Zugriffsmethoden USER PAM und ISAM des Datenverwaltungssystems den SHARED UPDATE-Modus zu verwenden. Er stellt für den Zugriff einer Taskgruppe auf eine Datei die erforderlichen Schutzmechanismen zur Verfügung.

5.3 DCAM-Programm ausführen: DCAM-Task

Eine DCAM-Task ist zunächst eine beliebige Task des BS2000. Sobald innerhalb dieser Task ein Programm eine DCAM-Anwendung eröffnet, wird das Taskattribut auf "TP" (transaction processing) gesetzt, falls im JOIN-Eintrag erlaubt.



Task-Typen

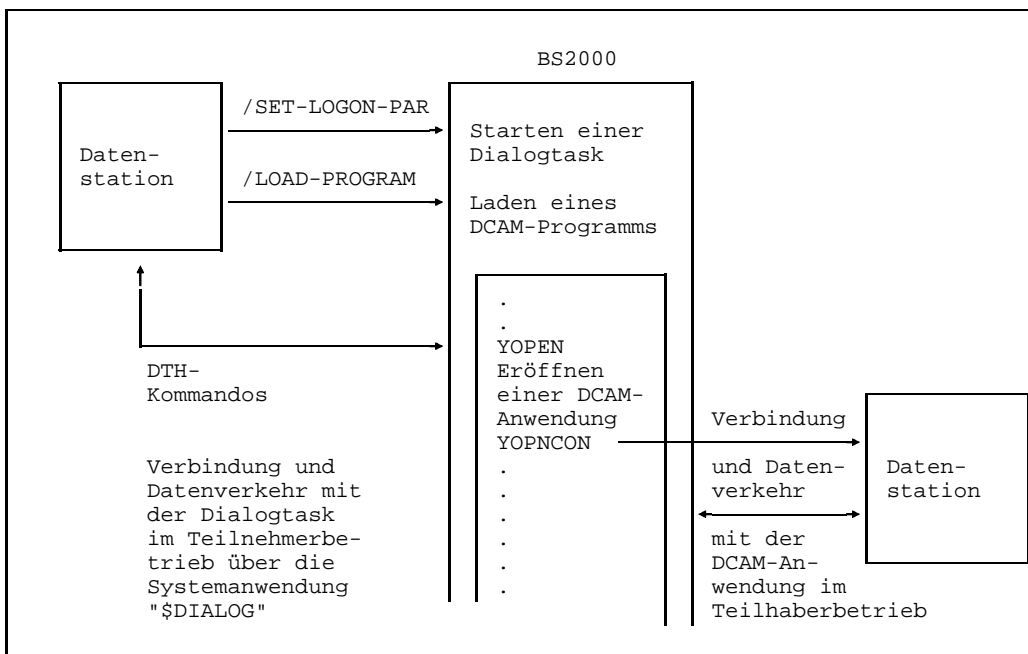
5.3.1 DCAM-Task starten

Bevor eine Task zur DCAM-Task werden kann, wird sie als Stapel- oder Dialogtask mit dem Kommando /SET-LOGON-PARAMETERS gestartet. Dieses Kommando kann gegeben werden

- für **Stapeltasks** von anderen Tasks aus durch sogenannte **ENTER-Dateien**, in denen als erstes ein /SET-LOGON-PARAMETERS-Kommando vorhanden ist. Stapeltasks werden vom System gestartet, wenn die Betriebsmittel bereit stehen und die vom Systemverwalter gesetzte Begrenzung dies erlaubt.
- für **Dialog-Tasks** von einer beliebigen **dialogfähigen Datenstation** aus, z.B. einer Datensichtstation 9750. In diesem Fall wird der Task sofort gestartet.

Hinweis

Bereits mit dem Prädialog wird die Verbindung zwischen einer Datenstation und einer Anwendung aufgebaut. Bei Teilnehmer-Tasks ist dies die Verbindung zwischen der Datenstation und der Anwendung '\$ DIALOG'. Nachdem eine solche Task zur DCAM-Task wurde, besteht nach wie vor die Verbindung zur Datenstation und somit ist es nicht möglich, von dieser Datenstation aus zu dem die Task steuernden DCAM-Programm eine Verbindung aufzubauen. Von dieser Datenstation aus kann aber das Programm mit DTH getestet und überwacht werden.



Teilnehmer- und Teilhaberbetrieb in einem Prozeß

5.3.2 DCAM-Task beenden

Eine DCAM-Task erhält ihr ursprüngliches Taskattribut, sobald das sie steuernde Programm die letzte DCAM-Anwendung schließt, bzw. das Programm beendet wird. Danach wird sie wie vorher als Stapel- oder Dialogtask geführt und mit dem /LOGOFF-Kommando beendet. Dies kann gegeben werden:

- als letztes Kommando in einer ENTER-Datei
- als Eingabe von der Dialogstation
- im Programm-Modus durch den LOGOFF-Makro oder indirekt durch TERMJ-Makro, wenn auf /LOGOFF gesprungen wird

Ferner können Tasks beendet werden durch

- das /CANCEL-JOB-Kommando des Operators oder
- das /CANCEL-JOB-Kommando in der Dialogtask für Tasks gleicher Benutzerkennung, aber anderer TSN
- das /SHUTDOWN-Kommando des Operators
- durch 'ABNORMAL TASK TERMINATION' bei einem Systemfehler

5.3.3 Programmierhinweise

Das nachfolgende Kapitel beschreibt typische Stolpersteine bei der Programmierung und soll Ihnen die Arbeit mit einer DCAM-Anwendung erleichtern.

Verbindungsaufbau funktioniert nicht:

- EDIT=SYSTEM ist nicht erlaubt bei Kommunikation zwischen Anwendungen (auch bei APS-Anwendungen)
- EDIT=USER ist für MSN-Datensichtstationen nicht erlaubt
- beim YOPNCON OPTCD=ACQUIRE angegebene Edit-Parameter gelten nur als Vorschlag. Beim nächsten YOPNCON mit diesem YCCB wird das DEXP (Datenaustauschprotokoll) der letzten Verbindung ausgewertet und nicht der generierte Wert.

Datenflußprobleme:

- eine DCAM-Anwendung darf nur synchronisiert senden. Bei Verbindungen zu Druckern reicht eine Synchronisation über Transportquittung nicht aus. Es müssen Abdruckquittungen angegeben werden.

Nachricht geht verloren:

- die Verbindung ist im falschen CS/CA-Zustand. Beachten Sie, daß beim Empfang von Quittungen CS/CA nicht ausgewertet wird.

Negative Transportquittung (NACK) wird empfangen, obwohl keine Quittung verlangt wurde:

- beim YSEND wird ausschließlich über das Empfangen bzw. nicht Empfangen positiver Quittungen entschieden. Negative Transportquittungen können nur durch NOTACK beim YOPEN unterdrückt werden.
- bei EDIT=SYSTEM können Nachrichten nur von VTSU geteilt werden (z.B. RESET-Nachrichten). Es sind mehrere Quittungen pro Nachricht möglich.

Returncode X'1808' nach YOPNCON:

- die Verweisadresse auf den YCCB ist durch einen früheren YOPNCON zerstört worden. Sie müssen YOPNCON mit CCB=CCBADR absetzen.

Returncode X'20000000':

- DCAM garantiert keine Abwärtskompatibilität. Mit DCAM V8.0 übersetzte Phasen sind auch bei Verwendung des Makros YGENCB nicht in DCM V7.01 Umgebung ablauffähig.

Zeilenvorschub vor jedem YSEND:

- bei DCAM muß, im Gegensatz zu TIAM (WROUT), jederzeit mit einer Eingabe des Anwenders gerechnet werden. Darum wird vor jedem SEND auf eine neue Zeile positioniert.

Letzte Nachricht geht verloren:

- ein DISCON darf normale Nachrichten überholen. Vor dem Verbindungsabbau muß auf die Quittung für die letzte Nachricht gewartet werden.

DCAM baut eine Verbindung ab:

- dies kann passieren, wenn eine Anwendung eine Unzahl von Transportquittungen nicht abholt.

Hinweise zur Performance:

- der PRIM=TASK soll keinesfalls eine PASS oder VPASS-Schleife machen. Benutzen Sie statt dessen SOLSIG.
- ein VPASS nach dem YOPCON ist seit BS2000 V6.0 unnötig.
- bei Dateitransfer über MSV2-Verbindungen (z.B. Anschluß von Fernschreibern) soll möglichst mit ELEMENT gearbeitet werden.
- Verbindungen mit EDIT=USER erfordern wesentlich weniger Systemaufwand als Verbindungen mit EDIT=SYSTEM.

6 Anhang

6.1 DCAM-Aufrufe

Arten der Aufrufe an DCAM:

- Makroaufrufe (Assembler = A)
- Call-Aufrufe (COBOL = C)

Art	Aufruf	Funktion	Beschreibung
A	YACB	1)	Anwendungssteuerblock erzeugen
A	YAPPL	Namen- Zuweisung	Angaben zur DCAM-Anwendung in der CLT speichern; Löschen dieser Angaben
A	YCCB	1)	Verbindungssteuerblock erzeugen
A; C	YCHANGE	Verbindung	Eigenschaften einer bereits aufgebauten Verbindung ändern
A; C	YCLOSE	Existenz	DCAM-Anwendung schließen
A; C	YCLSCON	Verbindung	Aufforderung oder Abbau einer Verbindung zurücknehmen
A	YCONN	Namen- Zuweisung	Angaben zur Verbindung in der CLT speichern; Löschen dieser Angaben
A	YDCG 2)	1)	Verteilcode-Gruppenblock erzeugen
A	YDDACB	1)	(Pseudo-)Abschnitt für den Steuerblock ACB erzeugen
A	YDDCCB	1)	(Pseudo-)Abschnitt für den Steuerblock CCB erzeugen

Art	Aufruf	Funktion	Beschreibung
A	YDDDCG 2)	1)	(Pseudo-)Abschnitt für den Steuerblock DCG erzeugen
A	YDDDIP 2)	1)	(Pseudo-)Abschnitt für den Steuerblock DIP erzeugen
A	YDDENB	1)	(Pseudo-)Abschnitt für den Steuerblock ENB erzeugen
A	YDDRPB	1)	(Pseudo-)Abschnitt für den Steuerblock RPB erzeugen
A	YDIP 2)	1)	Verteilungsparameterblock erzeugen
A	YENB	1)	Ereignis-Meldungsblock erzeugen, der asynchrone Meldungen mit Contingency-Routinen verknüpft.
A; C	YFORBID 2)	Datenübermittlung	Zuordnung des Verteilungsnamens zu einer Verteilcodegruppe aufheben
A	YMODCB	1)	Felder in bestehenden Steuerblöcken ändern
A; C	YOPEN	Existenz	DCAM-Anwendung eröffnen
A; C	YOPNCON	Verbindung	Verbindung aufbauen
A; C	YPERMIT 2)	Datenübermittlung	Verteilungsnamen einer Verteilcodegruppe zuordnen
A; C	YRECEIVE	Datenübermittlung	Nachricht empfangen
A; C	YREJLOG	Verbindung	Aufforderung zum Verbindungsaufbau zurückweisen
A; C	YRESET	Datenübermittlung	Empfangsaufrufe zurücknehmen; CS/CA-Zustand ändern

Art	Aufruf	Funktion	Beschreibung
A	YRPB	1)	Anforderungsparameter-Block erzeugen
A; C	YSEND	Datenübermittlung	Nachricht senden
A	YSENDREC	Datenübermittlung	Nachricht senden und empfangen kombiniert
A; C	YSETLOG 2)	Existenz	Zustand einer Anwendung ändern
A	YSHOWCB	1)	Feldinhalte aus einem Steuerblock in den Benutzerbereich übertragen
A	YTESTCB	1)	Feldinhalt in einem Steuerblock mit einem vorgegebenen Wert vergleichen
C	YWAIT		Warten auf Beendigung von asynchronen CALL-Aufrufen
A	YGENCB	1)	Einen oder mehrerer Steuerblöcke beliebiger Art erzeugen
A; C	YINQUIRE	Existenz; Verbindung	Informationen über Anwendungen und Verbindungen abfragen

1) Steuerblock-Funktionen

2) nur für DCAM(NEA)-Transport-Service-Anwendungen

6.2 Grenzwerte

– Asynchrone Verarbeitung

Die folgende Tabelle zeigt, wieviele asynchrone Befehle gleichzeitig bearbeitet werden können.

Maximale Anzahl	Gültig für Aufruf-Typ	Funktion
8 je Anwendung	YOPNCON ACCEPT ANY	Aufforderung eines beliebigen Partners annehmen
8 je Anwendung	YOPNCON ACCEPT SPEC	Aufforderung eines bestimmten Partners annehmen
128 je Anwendung	YOPNCON ACQUIRE	Aufforderung ausgeben
8 je Task einer Anwendung	YRECEIVE ANY	Empfangen einer Nachricht von einem beliebigen Partner
8 je Verbindung	YRECEIVE SPEC	Empfangen einer Nachricht von einem bestimmten Partner

– Verteilcode-Verwendung, nur für NEA-Transport-Service

Die folgende Tabelle zeigt die Maximalwerte für die Definition und Zuordnung von Verteilcodes

Art	oberer Grenzwert	Gültig für
Verteilcode-Gruppen je logischer Verbindung	8 16	COBOL Assembler
Verteilcodes je Codegruppe	8	Assembler, COBOL
Task je Verteilungsnamen	8	Assembler, COBOL

– **Eine Task**

kann **gleichzeitig** eine bestimmte **Anzahl DCAM-Anwendungen** eröffnet haben. Wieviele es maximal sein können, geht aus der jeweiligen Freigabemitteilung des BS2000 hervor.

Zusätzlich ist die Zahl der 'nicht vordefinierten Anwendungen' begrenzt Sie kann auf eine Task oder auf das gesamte System bezogen sein. Diese Grenzwerte werden beim Systemstart festgelegt und können im Betrieb modifiziert werden.

Eine Anwendung kann gleichzeitig nur eine bestimmte Anzahl Verbindungen unterhalten. Zusätzlich ist die Zahl der Verbindungen, die eine 'nicht vordefinierte Anwendung' (vgl. oben) unterhalten darf, begrenzt.

– **Maximale Nachrichtenlänge (MAXLN)**, nur für NEA-Transport-Service

Lokaler Parameter, der die Ökonomie der vom System bereitgestellten Puffer beeinflusst. Er enthält die maximale Länge der zu sendenden Daten (Transport Service Data Unit, TSDU).

Bei EDIT = USER:

1 Nachricht = 1 TSDU (YSEND)

Bei EDIT = SYSTEM, EDITOUT = PHYS oder FORM:

DCAM sendet die vom System editierte Nachricht in Teilstücken, die abhängig sind von MAXLN und der Gerätekapazität. Die Beachtung der Geräte-Kapazität liegt beim Anwender.

Beim Editieren wird abgeschnitten, falls ein editierter Satz größer als MAXLN ist.

Größtmögliche Anwender-Nachricht je YSEND: 32767 bytes.

Hinweis

Ein editierter Satz ist stets länger als Anwenderdaten, da Kontrollzeichen umgewandelt und Protokollketten hinzugefügt werden.

Die folgende Tabelle zeigt die Maximalwerte für MAXLN:

Angeforderte MAXLN	←- 65530	Default-Wert	
DCAMVER	8.0	8.0	Default-Wert
mit DVR MAXLN	x = 65530(*)	4096	4096
mit DAST		4096	32767

(*) Die Resultate sind abhängig von der HW/SW-Konfiguration und von der Generierung.

– Betriebsmittel-Grenzwerte

In DCAM V11.0 gelten die folgenden statischen Maximalwerte:

GID/Anwendung (permitted)	32
GID/Anwendung (not permitted)	32
DID/Anwendung	32

Durch den Name Manager des BS2000 ist die Zahl der Anwendungen sowie die Zahl der P1-Ereignisse pro Task im BS2000 V9.5 auf etwa 500 und ab BS2000 V10 auf etwa 2000 begrenzt.

Durch BS2000 ist die Anzahl der für eine Task erzeugten und noch nicht abgelaufenen P1-Contingencies begrenzt.

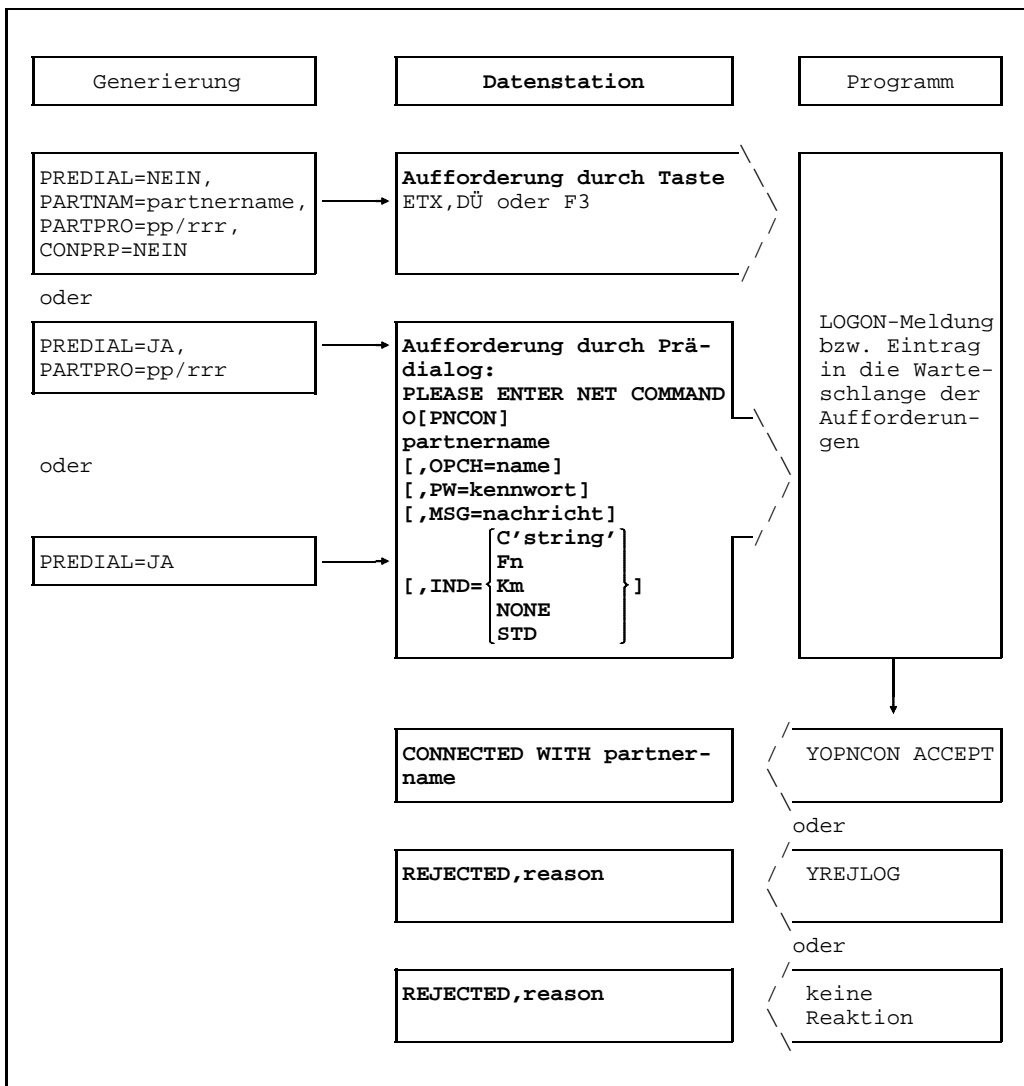
6.3 Verbindungsaufbau von der Datenstation



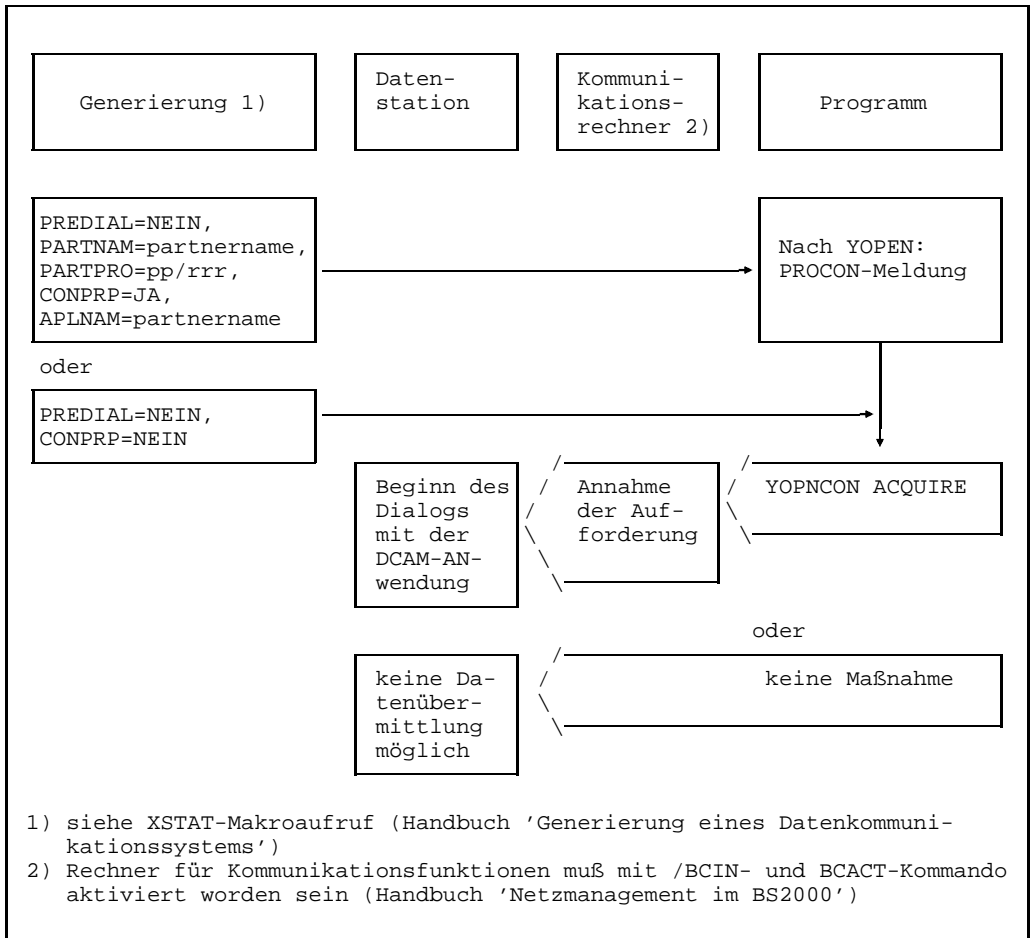
Dieser Abschnitt gilt nur für DCAM(NEA)-Transport-Service-Anwendungen.

Der Aufbau einer logischen Verbindung von einer Datenstation wird unterschiedlich erfolgen, je nach dem wie die Datenstation generiert wurde, welche Voreinstellungen vorgenommen wurden und wie der Kommunikationspartner DCAM-Anwendung den Verbindungsaufbau vorgesehen hat. Auf den folgenden Übersichtsbildern werden zwei grundsätzlich verschiedene Methoden gezeigt.

- Die DCAM-Anwendung, die als Partner benötigt wird, ist bereits eröffnet.
- Die Datenstation ist eingeschaltet und betriebsbereit.
- Die physikalische Verbindung (Leitung) wurde bzw. wird durch eine der drei folgenden Maßnahmen aufgebaut:
 - der Bediener der Datenstation hat sie aufgebaut
 - die Leitung ist eine Standleitung und braucht darum nicht aufgebaut zu werden
 - der Kommunikationsrechner, an dem die Datenstation angeschlossen ist, baut die Leitung auf, ausgelöst durch den Aufruf YOPNCON ACQUIRE im DCAM-Programm



Initiative bei der Datenstation (Attribut LOGON)



Initiative bei der DCAM-Anwendung (Attribut NLOGON)

Fachwörter

CMX-Anwendung

CMX application

Kommunikationsanwendung an einem SINIX- oder BS2000-Rechner, die von einem CMX-Anwendungsprogramm gesteuert wird.

Dateneinheit

Data unit

Datenmenge, die mit einem Aufruf an DCAM übergeben bzw. von DCAM empfangen werden kann.

Datenkommunikationssystem

Data communication system

Komplexe Einrichtung aus Hard- und Softwareprodukten, die es Kommunikationspartnern ermöglicht, Daten unter Beachtung bestimmter Regeln auszutauschen.

Datenstationsbenutzer

Terminal user

Mensch, der eine Datenstation benutzt, um mit einem Kommunikationspartner Daten auszutauschen.

[DCAM-]Anwendung

[DCAM] application

Kommunikationsanwendung, die von mindestens einem DCAM-Anwendungsprogramm gesteuert wird.

[DCAM-]Anwendungsprogramm

[DCAM] application program

Programm, das die Dienste der Zugriffsmethode DCAM benutzt. Es steuert eine oder mehrere DCAM-Anwendungen.

[DCAM-]Datenübermittlungsfunktion

[DCAM] data transmission function

DCAM-Funktion, die im Zusammenhang mit Senden und Empfangen von Nachrichten und Quittungen steht.

[DCAM-]Ereignis

[DCAM] event

DCAM-spezifisches Ereignis, das zur Koordinierung bestimmter Vorgänge im Datenkommunikationssystem verwendet werden kann. Es trifft vom Programmablauf zeitlich entkoppelt ein (= asynchron eintretendes Ereignis).

[DCAM-]Existenzfunktion

[DCAM] existence function

DCAM-Funktion, die im Zusammenhang mit der Erzeugung und Auflösung von DCAM-Anwendungen steht.

[DCAM-]Namen-Zuweisungsfunktion

[DCAM] name assignment function

DCAM-Funktion, die es dem Anwender erlaubt, die Anwendungsprogramme unabhängig von statischen Parameterwerten wie DCAM-Anwendungsname, Partnername usw. zu erstellen.

[DCAM-]Verbindungsfunktion

[DCAM] connection function

DCAM-Funktion, die im Zusammenhang mit Aufbau und Abbau von Verbindungen steht.

Expressnachricht

Express message

Nachricht begrenzter Länge, die mit höherer Priorität als Normalnachrichten zugestellt wird.

Format-Datenstation

Format terminal

Betriebsart einer logischen Datenstation, bei der die Nachricht aus einem Format (= Formular, Maske) besteht.

ICMX

Allgemeine C-Schnittstelle, die die OSI-Transportfunktionalität anbietet.

Kommunikationsanwendung

Communication application

Instanzen zur Verarbeitung von Nachrichten, die zwischen Kommunikationspartnern ausgetauscht werden. Sie wird vom Datenkommunikationssystem über ihren Zugangspunkt adressiert.

Kommunikationspartner

Communication partner

Instanzen, die Verbindungen unterhalten und Daten miteinander austauschen.

[Kommunikations-]Protokoll

[Communication] protocol

Beschreibung der Übergabebedingungen und Übergabeformate zwischen gleichartigen Funktionsschichten im Datenkommunikationssystem.

Kommunikationsrechner

Communication computer

Spezieller Rechner für Kommunikationsfunktionen.

Kommunikationszugriffsmethode

Communication access method

Software, die den Anwendungen Schnittstellen zur Kommunikation bietet.

Logische Datenstation

Virtual terminal

Modellvorstellung einer Datenstation, deren Funktionen auf die physikalischen Eigenschaften unterschiedlicher Datenstationen abgebildet werden.

Mehrfach benutzbare DCAM-Anwendung

Shareable DCAM application

DCAM-Anwendung, die von mehreren Tasks gleichzeitig benutzt werden kann.

Nachricht

Message

Eine logisch zusammengehörige Datenmenge, die an einen Kommunikationspartner gesendet werden soll bzw. von einem Kommunikationspartner empfangen werden soll.

Prozess

Process

Instanzen zur Ausführung eines Programms innerhalb einer Task.

Task

Task

Träger von Prozessen. Im BS2000 werden Tasks u.a. zum Abwickeln von Benutzerjobs (z.B. Batchjob, Dialogtask) oder zum Betreiben von (DCAM-, UTM-, TTX-) Anwendungen eingesetzt (Ablauf aller Aufgaben zwischen den BS2000-Kommandos LOGON und LOGOFF).

[Transportdienst

Transport service

Dienstleistung für den Datenaustausch zwischen Kommunikationspartnern. Der Transportdienst veranlaßt und kontrolliert den Transport der Nachricht durch das Datenkommunikationssystem und verwaltet Verbindungen.

Transportquittung

Transport acknowledgment

Ereignis, das über Abschluß oder Abbruch einer Datenübermittlung informiert.

Verbindung

Connection

Zuordnung zweier Kommunikationspartner, die es ihnen ermöglicht, Daten miteinander auszutauschen.

XHCS

Extended Host Code Support

Software zur Unterstützung von 8-bit-Datenstationen.

Zeilen-Datenstation

Line terminal

Betriebsart einer logischen Datenstation, bei der die Nachricht in Form von Zeilen strukturiert ist,

Literatur

- [1] **DCAM** (BS2000/OSD)
 Makroaufrufe
 Benutzerhandbuch
- [2] **DCAM** (BS2000/OSD)
 COBOL-Aufrufe
 Benutzerhandbuch
- [3] **CMX** (BS2000)
 Kommunikationsmethode im BS2000
 Benutzerhandbuch
- [4] **COBOL85** (BS2000)
 COBOL-Compiler
 Benutzerhandbuch
- [5] **COBOL85** (BS2000)
 COBOL-Compiler
 Beschreibung
- [6] **Assembler** (BS2000)
 Beschreibung
- [7] **Assemblerbefehle** (BS2000/OSD)
 Sprachbeschreibung
- [8] **ASSEMBH** (BS2000)
 Benutzerhandbuch
- [9] **TIAM** (BS2000/OSD)
 Benutzerhandbuch
- [10] **FHS** (BS2000/OSD)
 Benutzerhandbuch

- [11] **VTSU (BS2000/OSD)**
Virtual Terminal Support
Benutzerhandbuch
- [12] **XHCS**
(BS2000/OSD)
Extended Host Code Support für BS2000/OSD-BC V1.0
Benutzerhandbuch
- [13] **BS2000/OSD**
Makroaufrufe an den Ablaufteil
Benutzerhandbuch
- [14] **BS2000/OSD**
Dienstprogramme
Benutzerhandbuch
- [15] **BS2000/OSD**
Kommandos, Band 1-7
Benutzerhandbuch
- [16] **BS2000/OSD**
System-Exits
Benutzerhandbuch

- [24] **SOCKETS-DE** (BS2000)
Communication Method in BS2000
User Guide
- [25] **Wege zur offenen Kommunikation**
Das OSI-Referenzmodell im Umfeld der Kommunikation Druckschrift

Stichwörter

A

- A-Struktur 118
- ablaufinvariant 115, 125
- ablaufvariant 115, 125
- absenderspezifische Warteschlange 20, 84
- ACB 97
- Adressierung 19
- Anforderungsparameterblock 97
- Anwendung 17
 - abfragen 50
 - einfach benutzbar 42
 - eröffnen 29, 42
 - mehrfach benutzbar 42, 44
 - schließen 50
 - Zustand verändern 48
- Anwendungssteuerblock 97
- Anwendungsstruktur 118
- ANY 54
- ANYSTART 62
- APPSTART 62
- Assembler-Programm 97
- asynchrone Befehlsausführung 105
- asynchrone Meldung 111
- Aufforderung
 - annehmen 67
 - zurücknehmen 76
 - zurückweisen 76
- Aufrufe 24

B

- B-Struktur 118
- Befehlsausführung
 - asynchron 105
 - synchron 103
- Befehlsstruktur 118

Begleitinformation 58
Beschreibung der Verbindung 58

C

CA-Zustand 84
 ändern 89
CALL-Aufrufe 24, 135
 asynchrone Ausführung 123
 synchrone Ausführung 123
CCB 99
CLT 93
COBOL-Programm 118
Codelänge 70
Codeposition 70
COMEND 112
Communication Link Table 31
Contingency-Routine 52, 107
COPY-Anweisung 122
COPY-Elemente 119
CS-Zustand 84
 ändern 89

D

Daten 15
Dateneinheit 34
 empfangen 84
 senden 79
Datenfluß-Steuerung 26
Datenflußkontrolle 58
Datenlänge, maximale 62
Datenstation 17
 Format- 82
 logische 37, 95
 Zeilen- 81, 96
 Zugriff auf 37
Datenübermittlung 78ff
DCG 99
Dialog-Task 129
DIP 99
DISCO 46
DISNAME 46, 47
dynamische Namen-Zuweisung 31
dynamische Steuerblockerzeugung 102

E

EDIT 56
EDIT-Optionen 62, 96
Eigenschaften
 ändern 76
 der Verbindung 56
einfach benutzbare Anwendung 42
Einträge abfragen 74
empfängerglobale Warteschlange 20, 84
Empfangsaufruf zurücknehmen 89
ENACO 107, 112
ENB 97
Ereignis 105, 111, 123
Ereignis-Meldungsblock 97
Ereignisinformation 108
Eröffnen
 erstmalig 44, 46
 nachfolgendes 46
erstmaliges Eröffnen 44, 46
Existenzfunktion 41
EXPR 112
Expressnachricht 26, 84, 112

F

FCB 93
FDBK-Feld 109
Fehlerbehandlungsroutine 31
Fehlercode 124
FHS-Struktur 119
Format-Datenstation 82, 95

G

GO-Signal 26
GOSIGNAL 123
Grenzwerte 138

I

impliziter Verteilcode 24
Indikator 124
ISO-Attribut 42
ISO-Transport-Services 9

K

Kennwort 25
zum Verbindungsaufbau 62
Kommunikationspartner 16
Kopflängenbyte 82

L

Layer 10
LETTER 123
logische Datenstation 37, 95
logische Zeile 81
LOGON 48, 111
LOSCON 111, 123
LOSCON-ROUTINE 72

M

Makroaufrufe 24, 97, 135
maximale Datenlänge 62
maximale Nachrichtenlänge 59, 64
maximale Wartezeit 103
MAXLN 62
mehrfach benutzbare Anwendung 42, 44
Meldungen 24
asynchron 111
MF-Parameter 115
More Data-Funktion 34, 59

N

nachfolgendes Eröffnen 46
Nachricht 34
empfangen 29, 84
senden 29, 79
Nachrichtenaufbereitung 62, 81
Nachrichtencode 61
Nachrichtenkopf 82
Nachrichtenlänge, maximale 59, 64
Nachrichtenverteilung mit Verteilcodes 62
Namen-Zuweisung, dynamisch 31
Namen-Zuweisungsfunktion 93
NEA-Transport-Services 8
negative Transportquittung 72
NOEVENT 123
NSHARE 43

O

offenes System 9
OPENED 123
OSI 9
OSI-Referenzmodell 9, 10

P

Parallelverbindung 19, 56
Parameterwerte 98
Partnername 58
physikalische Programmierung 37
PLAM-Bibliothek 122
Primärtask 17, 32
PROC 56
PROCON 111
PROCON-Meldung 55
Programm 15
 ausführen 129
 reentrant-fähig 115, 125
Prozessorname 29, 58

Q

Quell-Bibliothek 122

R

RDF-Kennwort 25
reentrant-fähige Programme 115, 125
Referenzmodell 9
RLTH 59
Routenauswahl 61
RPB 97
Rückgabecode 124
Rückmeldung 31, 109, 124

S

Schicht 9
SECOND 112
Sekundärtask 17, 33
SHARE 46, 47
shared code 115
SOLSIG 105
SPEC 54
Stapeltask 129
START-Zustand 48
statische Steuerblockerzeugung 101
Statusabfrage 121

- Steuerblock 97
- Steuerblockerzeugung
 - dynamisch 102
 - statisch 101
- steuernde Task 32
- STOP-Zustand 49
- synchrone Befehlsausführung 103

T

- Tabelle, taskspezifisch 93
- TACK 112
- Task 15, 129
 - beenden 131
 - starten 129
 - steuernd 32
- taskspezifische Tabelle 93
- Terminalstatus 61
- TFT 93
- TIAMCTRC 96
- TIDU 34
- TIMEOUT 72
- Transportquittung 77, 88, 112
 - negative 72
- TSDU 34

U

- Überlagerungssegment 125
- UTM-Anwendung 39

V

- V-Struktur 118
- Verbindung 19
 - abbauen 77
 - abbrechen 29
 - aufbauen 29, 51
 - Beschreibung der 58
 - Eigenschaften der 56
 - explizit abbauen 29
 - implizit abbauen 29
 - vordefinierte 72
- Verbindungsaufbau 64
 - Kennwort zum 62
- Verbindungsfunktion 51
- Verbindungssteuerblock 99
- Verbindungsstruktur 118

Verteilcode, implizit 24
Verteilcode-Gruppe 89
Verteilcode-Gruppenblock 99
Verteilcode-Zuordnung 89
Verteilcodes 22, 46, 47
 Verwendung von 69
verteilcodespezifische Warteschlange 22
Verteilungsname 46
 zuordnen 91
Verteilungsparameter-Block 99
Verteilungsstruktur 118
Verwendung von Verteilcodes 69
vordefinierte Verbindung 72
VTCSET 96
VTLG-Struktur 118
VTSU-Control-Block 96, 118
VTSUCB-Struktur 118

W

W-Struktur 118
Warteschlange
 absenderspezifisch 20, 84
 empfängerglobal 20, 84
 verteilcodespezifisch 22
Wartestruktur 118
Wartezeit, maximale 103
WORKING STORAGE SECTION 118
Wurzel-Segment 125

X

X.25-Schnittstelle 99

Y

YAPPL 94
YCHANGE 76
YCLOSE 50
YCLSCON 76, 77
YFORBID 92
YGENCB 102
YINQUIRE 50, 74, 121
YOPEN 42
YOPNCON 51
YPERMIT 91
YRECEIVE 84
YREJLOG 76

YRESET 89
YSEND 79
YSENDREC 89
YWAIT 123

Z

Zeilen-Datenstation 81, 96
Zugriff auf Datenstation 37
Zugriffsschutz 25
Zuordnung auflösen 92

Inhalt

1	Einleitung	1
1.1	Wegweiser durch das Handbuch	5
1.2	Änderungen gegenüber Vorgängerausgabe	6
2	Einführung in die DCAM-Schnittstelle	7
2.1	Die Kommunikationszugriffsmethode DCAM	8
2.2	Verkehrsbeziehungen im Datenkommunikationssystem mit DCAM	12
2.3	Grundlegende Begriffe	15
2.3.1	Programm, Daten, Task	15
2.3.2	Kommunikationspartner	16
2.3.3	Adressierung	19
2.3.4	Verbindungen	19
2.4	Charakteristische Merkmale von DCAM	20
2.4.1	Verteilung ankommender Nachrichten	20
2.4.1.1	Absenderspezifische und empfängerglobale Warteschlange	20
2.4.1.2	Verteilcodespezifische Warteschlange	22
2.4.1.3	Impliziter Verteilcode	24
2.4.2	Aufrufe und Meldungen	24
2.4.3	Zugriffsschutz	25
2.4.4	Expresßnachrichten	26
2.4.5	Datenfluß-Steuerung	26
2.5	Programmaufbau	27
2.5.1	Aufgaben eines DCAM-Programms	27
2.5.2	Grundstruktur eines DCAM-Programms	29
2.5.3	Steuerung von Primär- und Sekundärtasks	32
2.5.4	Nachrichten und lokale Dateneinheiten - More Data-Funktion	34
2.5.5	Zugriff zu Datenstationen	37
2.6	Realisierung des Verarbeitungsverbundes	38
2.6.1	DCAM-Anwendung als Partner	38
2.6.2	UTM-Anwendung als Partner	39

3	Funktionen von DCAM	41
3.1	Existenzfunktion	41
3.1.1	DCAM-Anwendung eröffnen	42
3.1.1.1	Einfach benutzbare DCAM-Anwendung	43
3.1.1.2	Erstmaliges Eröffnen einer mehrfach benutzbaren DCAM-Anwendung	44
3.1.1.3	Erstmaliges Eröffnen - Verwendung von Verteilcodes	46
3.1.1.4	Nachfolgendes Eröffnen	46
3.1.1.5	Nachfolgendes Eröffnen - Verwendung von Verteilcodes	47
3.1.2	Zustand einer DCAM-Anwendung verändern	48
3.1.3	Zustand einer DCAM-Anwendung abfragen	50
3.1.4	DCAM-Anwendung schließen	50
3.2	Verbindungsfunktion	51
3.2.1	Verbindung aufbauen YOPNCON	51
3.2.1.1	Beschreibung der aufzubauenden Verbindung	58
3.2.1.2	Auffordern zum Aufbau	64
3.2.1.3	Annahme einer Aufforderung	67
3.2.1.4	Aufbau einer Verbindung - Verwendung von Verteilcodes	69
3.2.1.5	Anschließen an eine vordefinierte Verbindung	72
3.2.2	Einträge über Partner und Verbindungen abfragen	74
3.2.3	Aufforderung zum Verbindungsaufbau zurückweisen	76
3.2.4	Eigenschaften einer Verbindung ändern	76
3.2.5	Aufforderung zum Verbindungsaufbau zurücknehmen	76
3.2.6	Verbindung abbauen	77
3.3	Datenübermittlungsfunktion	78
3.3.1	Senden einer Nachricht/Dateneinheit	79
3.3.2	Nachricht/Dateneinheit empfangen	84
3.3.3	Senden und Empfangen kombiniert	89
3.3.4	Empfangsaufrufe zurücknehmen und CS/CA-Zustand ändern	89
3.3.5	Verteilcode-Zuordnung steuern	89
3.4	Namen-Zuweisungsfunktion	93
4	Unterstützung logischer Datenstationen	95
5	DCAM-Programme	97
5.1	Assembler-Programme	97
5.1.1	Makroaufrufe und Steuerblöcke	97
5.1.2	Synchrone Ausführung der Befehle an DCAM	103
5.1.3	Asynchrone Ausführung der Befehle an DCAM	105
5.1.4	Rückmeldung	109
5.1.5	Asynchrone DCAM-Meldungen	111
5.1.6	'Reentrant'-fähige Assembler-Programme	115

5.2	COBOL-Programme	118
5.2.1	COBOL-CALLS und Datenstrukturen	118
5.2.2	CALL-Aufrufe ausführen	123
5.2.2.1	Synchrone Ausführung	123
5.2.2.2	Asynchrone Ausführung	123
5.2.3	Rückmeldung	124
5.2.4	'Reentrant' -fähige COBOL-Programme	125
5.2.5	Verwendung weiterer Systemschnittstellen	128
5.3	DCAM-Programm ausführen: DCAM-Task	129
5.3.1	DCAM-Task starten	129
5.3.2	DCAM-Task beenden	131
5.3.3	Programmierhinweise	132
6	Anhang	135
6.1	DCAM-Aufrufe	135
6.2	Grenzwerte	138
6.3	Verbindungsaufbau von der Datenstation	141
	Fachwörter	145
	Literatur	149
	Stichwörter	153

DCAM (BS2000)

Programmschnittstellen

Stand der Beschreibung:
DCAM V11.0A

Kritik... Anregungen... Korrekturen...

Die Redaktion ist interessiert an Ihren Kommentaren zu diesem Handbuch. Ihre Rückmeldungen helfen uns, die Dokumentation zu optimieren und auf Ihre Wünsche und Bedürfnisse abzustimmen.

Sie können uns Ihre Kommentare per E-Mail an manuals@ts.fujitsu.com senden.

Zertifizierte Dokumentation nach DIN EN ISO 9001:2000

Um eine gleichbleibend hohe Qualität und Anwenderfreundlichkeit zu gewährleisten, wurde diese Dokumentation nach den Vorgaben eines Qualitätsmanagementsystems erstellt, welches die Forderungen der DIN EN ISO 9001:2000 erfüllt.

cognitas. Gesellschaft für Technik-Dokumentation mbH
www.cognitas.de

Copyright und Handelsmarken

Copyright © Fujitsu Technology Solutions 2010.

Alle Rechte vorbehalten.

Liefermöglichkeiten und technische Änderungen vorbehalten.

Alle verwendeten Hard- und Softwarenamen sind Handelsnamen und/oder Warenzeichen der jeweiligen Hersteller.

i

Zum 1. April 2009 ist Fujitsu Siemens Computers in den alleinigen Besitz von Fujitsu übergegangen. Diese neue Tochtergesellschaft von Fujitsu trägt seitdem den Namen Fujitsu Technology Solutions. Das vorliegende Dokument ist eine fachlich unveränderte Neuausgabe eines früheren Handbuchs zu einer bereits vor längerer Zeit freigegebene Produktversion.

Bitte beachten Sie, dass alle Firmenbezüge und Copyrights im vorliegenden Dokument rechtlich auf Fujitsu Technology Solutions übergegangen sind.

Kontakt- und Supportadressen werden nun von Fujitsu Technology Solutions angeboten und haben die Form ...@ts.fujitsu.com.

Die Internetseiten von Fujitsu Technology Solutions finden Sie unter <http://de.ts.fujitsu.com/>