1 Preface

DCAM (Data Communication Access Method) is an access method in the data communication system for communication between programs or between a program and a terminal. DCAM offers two different sets of functions:

– DCAM(NEA) transport service functions

– DCAM(ISO) transport service functions
  These permit data communication on the basis of the transport services standardized by ISO.

This manual is intended for programmers who use DCAM macros in their programs. You should, therefore, be familiar with the BS2000 ASSEMBLER language.

A further prerequisite is the basic knowledge of BS2000 necessary for compiling and linking programs.

This manual also assumes that you are familiar with the manual ’DCAM Program Interfaces’ and with the OSI Reference Model.

1.1 The concept of the DCAM manuals

The description of the DCAM ASSEMBLER interface is divided into two manuals to suit all types of users, regardless of whether they want to create programs or simply want information about DCAM (see the figure below).

DCAM Program Interfaces Chapter	**DCAM Macros** Chapter
Preface	Preface
Basic terminology and introduction to the use of the DCAM interface	DCAM ASSEMBLER interface
DCAM functions	Using the DCAM functions
Support for virtual terminals	—
Coding and executing DCAM programs	Macro catalog
—	Examples
—	Description of the DCAM system exit
Appendix	Appendix

Layout of the Reference Manual and User Guide

This manual contains the descriptions for both DCAM(ISO) transport service applications and DCAM(NEA) transport service applications. Differences between the two are discussed where applicable. Passages, sections and entire chapters that apply only to DCAM(NEA) transport service applications are indicated by a

at the start of the text.

How should the DCAM programmer use the manuals?

Start with a thorough study of the "DCAM Program Interfaces"manual. Only after you have mastered the contents of that manual will you be ready to tackle this User Guide.

"DCAM Program Interfaces" was written so that it can be read "sequentially", i.e. the user is introduced to DCAM in a step-by-step fashion.

The chapter **"DCAM ASSEMBLER interface"** expands on the introduction provided in "DCAM Program Interfaces"; it explains the special techniques for the use of the macros.

The chapter **"Using the DCAM functions"** contains a complete description of each macro, including information about filling out the various fields. In both manuals, the descriptions of the macros are arranged by the macro functions. The arrangement of the various sections is virtually identical, so that you can start with the detailed description of the function in "DCAM Program Interfaces" and find information on practical application in this manual.

The chapter **"Macro catalog"** contains all DCAM macros in alphabetical order. For each macro, this chapter also shows the feedback information which must be evaluated after execution of the macro.

The chapter **"Examples"** contains a few examples of DCAM(ISO) transport service applications and DCAM(NEA) transport service applications.

The chapter **"The DCAM system exit"** describes this user-specific exit routine.

Finally, the **Appendix** contains all important summaries and tables.

At the back of this User Guide you will find a glossary, a list of related publications, and an index.

A number of books and guides on computer networks and remote data processing with BS2000 deal with topics related to those discussed in this User Guide. Subjects such as generation and administration, programming communication processors and terminals, and support for virtual terminals are dealt with in separate manuals.

1.2 Changes since the last version of the manual

VTSU interface

This manual no longer describes the VTSU interface. The chapter that previously bore this name has been deleted. See the "VTSU User Guide" for a detailed description of the VTSU interface, the VTSU control block, the logical control characters and the status information.

Feedback information

The table has been expanded to include new feedback information messages.

README file

Information on any functional changes and additions to the current product version can be found in the product-specific README file. You will find the file on your BS2000 computer under the name `SYSDOC.product.version.READ-ME.E`. The user ID under which the README file is cataloged can be obtained from your system administrator. You can view the README file using the `/SHOW-FILE` command or an editor, and print it out on a standard printer using the following command:

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

Any functional changes and additions to the current product version can be found in the chapter "Manual supplements".

1.3 Notational conventions

In order to simplify the use of the manual, this description uses characters as so-called metasymbols. These are already used widely in other BS2000 user manuals and are explained in the following table.

Character	Explanation	Examples
UPPERCASE LETTERS	Denote constants which the user must specify in this form	"YES"
lowercase letters	Denote variables whose contents may vary from case to case. The user must replace them with the current values. The input format is determined by the type of PICTURE clause.	partnername
{ }	Alternative entries are listed below each other and enclosed within braces.	{"YES" / "NO"}
[ ]	Optional entries are enclosed within square brackets.	[password4] [{"YES" / "NO"}]
underscored	Default values are underscored. The default is the value assumed by the system if no value is specified by the user.	[{"YES" / "NO"}]
...	Ellipsis indicates repetition. It shows that the preceding specification may be given more than once.	(vsn,...)

Character	Explanation	Examples
()	An expression that is used to represent a variable is enclosed within parentheses. This notation is intended to show the value range at a glance. The parentheses are needed because several characters are necessary to show the range.	(0 < length < 9)
≤	The relationship between two values: the value on the left is less than or equal to that on the right; the value on the right is greater than or equal to that on the left.	number ≤ 2047 0 ≤ position
≥	The relationship between The value on the left is greater than or equal to the value on the right; the value on the right is less than or equal to the value on the left.	number ≥ 1
< >>	As above, but the case 'equal to' is not permitted.	0 < length count < 9

The following notational conventions also appear in the body of the text:

Note	precedes particularly useful or important items of information
boldface	highlights important terms
!	indicates passages that are applicable only to DCAM(NEA) transport service applications

2 The DCAM ASSEMBLER interface

An introduction to the characteristics of the ASSEMBLER interface has already been given in the description of the program interfaces.

All DCAM macros which do not directly serve to generate or manipulate control blocks refer to control blocks. These action macros are so designed that the control blocks and the instructions (with action calls) can be generated separately.

The following steps are required in a program run:

Step 1: **Provision of information** in control blocks by generating the control blocks and entering their contents.

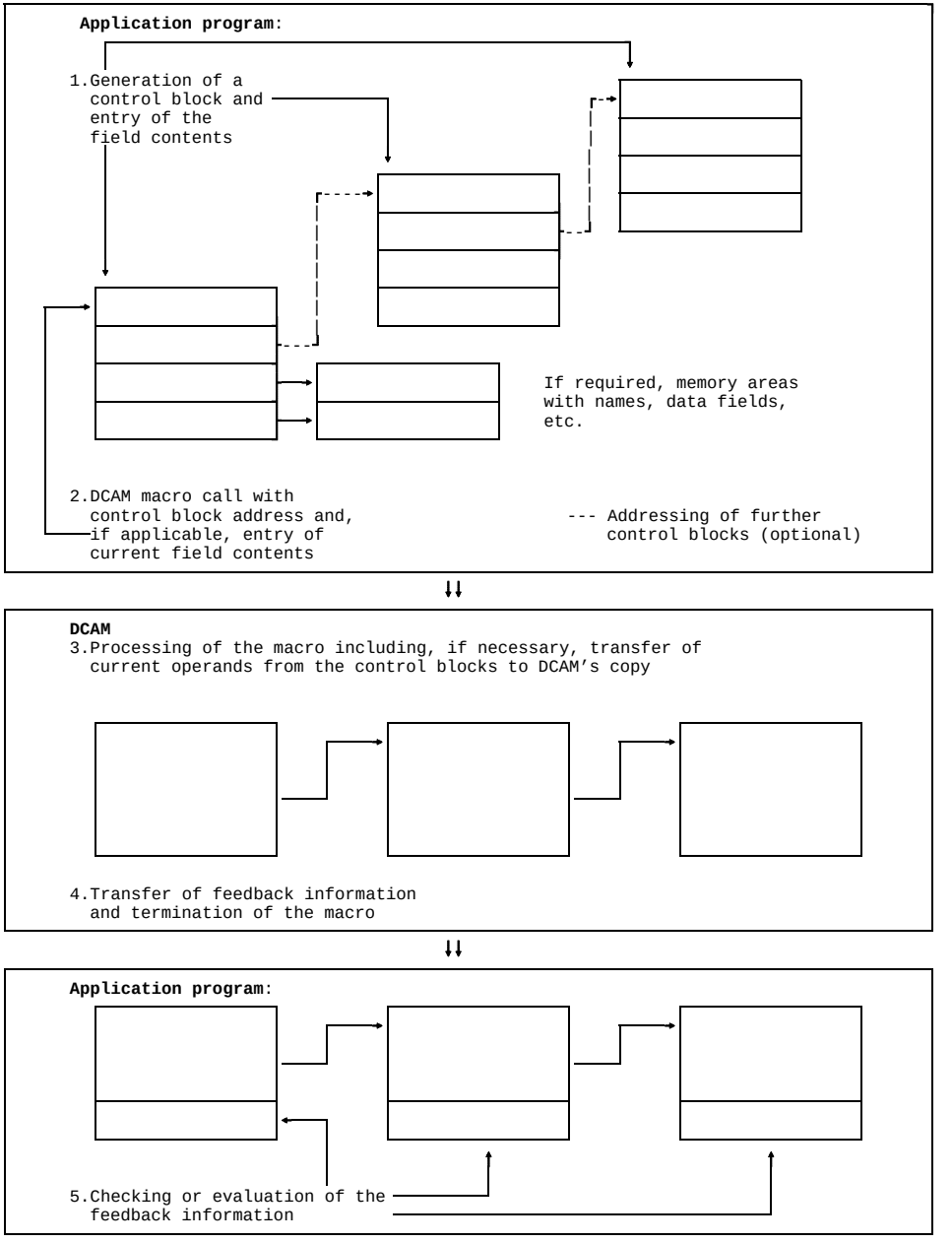
Step 2: When an **action macro is addressed to DCAM**, the control block that contains the required information is addressed.

Step 3: When the **macro is processed**, DCAM fetches the required fields from the specified control blocks and from variable fields addressed therein. DCAM sets up a copy of the control block with the current contents.

Step 4: DCAM enters **feedback information** (messages, identifiers etc.) into the user's control block and terminates the macro.

Step 5: The **user processes the feedback information** by accessing the control block fields or registers set by DCAM.

The following figure provides an overview of the basic processing steps.



DCAM ASSEMBLER interface

2.1 Generating control blocks

Control blocks are memory areas with predefined formats that contain operands for action macros. For each object that exists at the DCAM interface, there is a separate type of control block that contains the data which describes the object. An object may be: an application, a connection, an event, a request, a distribution parameter or a distribution code group.

There are three possible ways of generating control blocks:

– **static generation** (at assembly time);

– **dynamic generation** (by DCAM during program execution);

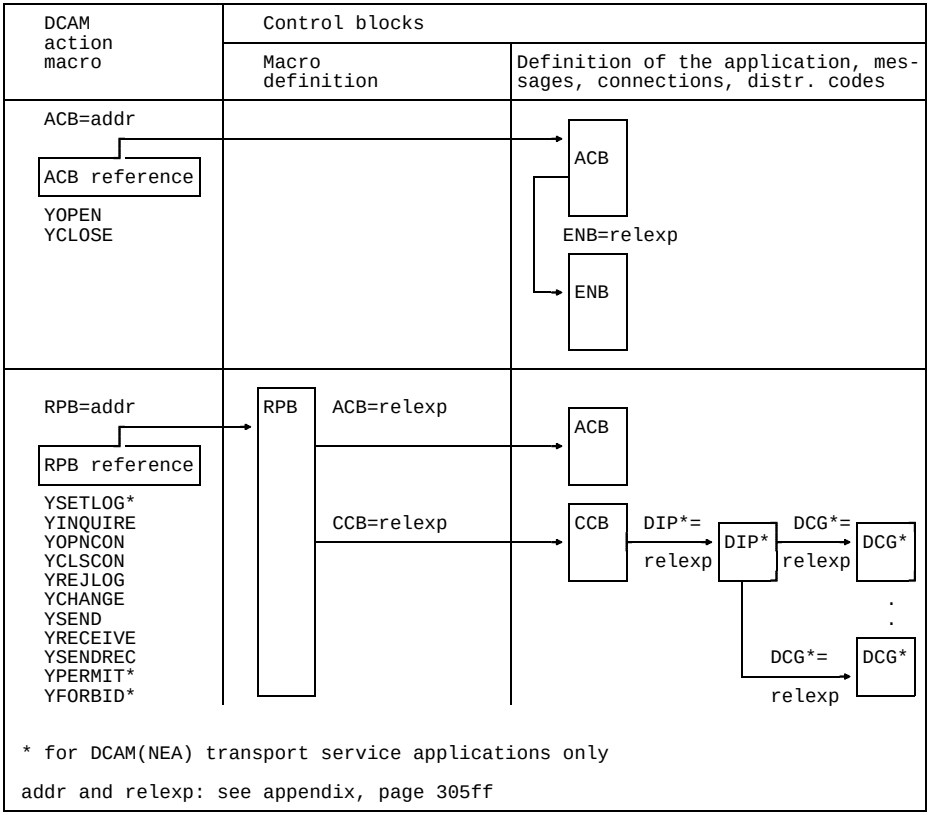
– **direct generation** (by the user).

The following table provides a summary of the macros for control block generation.

		Macros for control block generation		
Object	Control block	static	dynamic	direct
Application	ACB (application control block)	YACB	YGENCB BLK=ACB	YDDACB
Event	ENB (event notification block)	YENB	YGENCB BLK=ENB	YDDENB
Connection	CCB (connection control block)	YCCB	YGENCB BLK=CCB	YDDCCB
Request	RPB (request parameter block)	YRPB	YGENCB BLK=RPB	YDDRPB
Distribution parameters *	DIP (distribution parameter block)	YDIP	YGENCB BLKDIPB	YDDDIP
Distribution code group *	DCG (distribution code group block)	YDCG	YGENCB BLK=DCG	YDDDCG

* for DCAM(NEA) transport service applications only

The following figure shows the action macros and the control blocks.



Macros and control blocks

2.1.1 Generation of control blocks during assembly

Control blocks that are to be generated during assembly must be defined in the data section of the program. The appropriate macro is used for this purpose, with the result that the desired control block will be located at this point in the program after macro expansion.

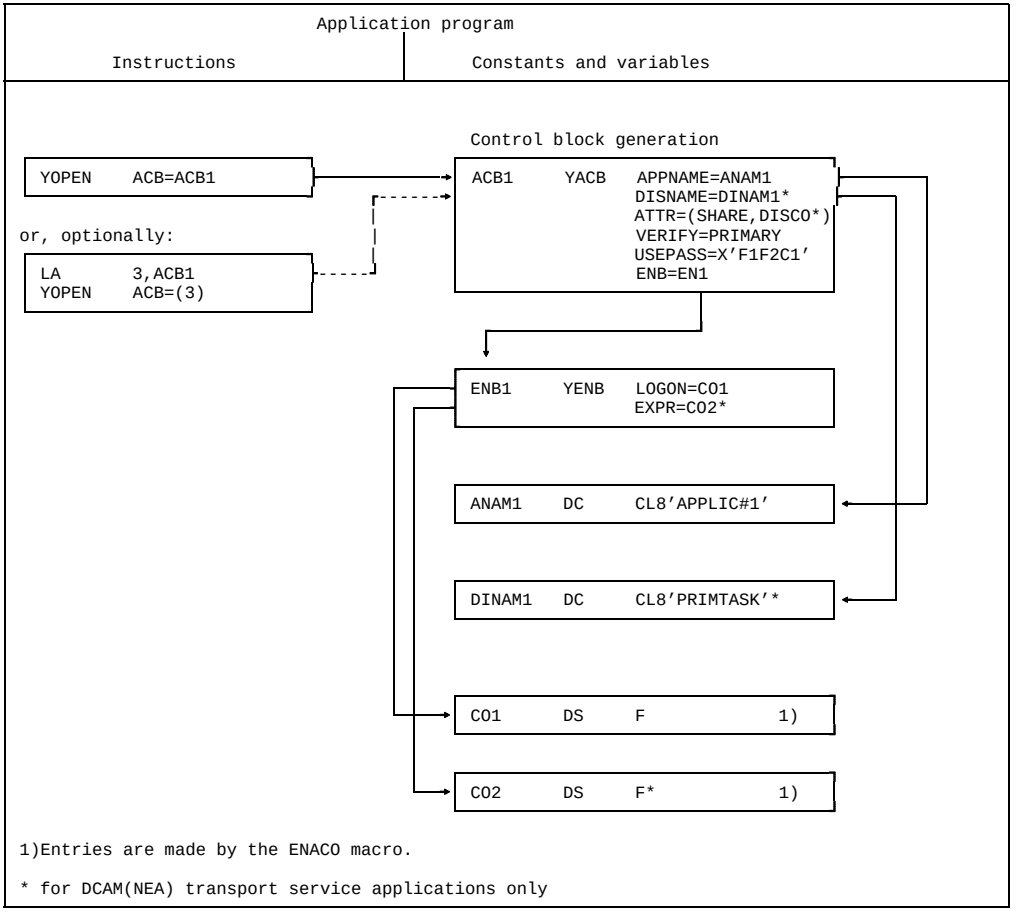
The contents of the individual fields are

- **specified by the user**, by combining appropriate values with a given keyword;
- **determined by the macro generator** if the user indicates, by omitting individual keywords, that the default values are to be used (but the default values can also be specified explicitly);
- **omitted** to allow DCAM to enter information after the execution of an action macro.

The entries in the control blocks are either self-defining ’code’ or ’constant’ terms or address references (’relexp’) to further control blocks or fields. These fields contain names or are used for data transfer.

The **address of the macro** generating the control block is later used to address the control block (’addr’). This address can also be passed in a register (see appendix, page 305).

The following figure summarizes the principle. An ACB and an ENB are generated to enable a DCAM application to be opened (see also the section Primary opening/use of distribution codes, page 54).



Example of static generation of control blocks

2.1.2 Generation of control blocks during program execution

The static generation of control blocks described above has the disadvantage of being dependent on the control block format. Also, reentrant programming is not possible because the control blocks generally contain addresses of variable fields.

In contrast, dynamic control block generation offers the following advantages:

- Control blocks can be established in **class 5 memory**, i.e. the user does not have to concern himself about the memory location, which is outside his program.
- However, it is possible to reserve memory space **in the program** and to generate the control block there.
- **Reference addresses** to fields and other control blocks can be specified in registers.
- **Several control blocks** of the same type can be generated with a single macro.
- All other field definitions are similar to those for static generation.

The **MF operand** determines the form of the dynamic generation macro YGENCB and of the control block handling macros YMODCB (see the section "Modification by means of a separate macro"), YSHOWCB (see the section Reading and saving control block fields) and YTESTCB (see the section Reading and testing control block fields).

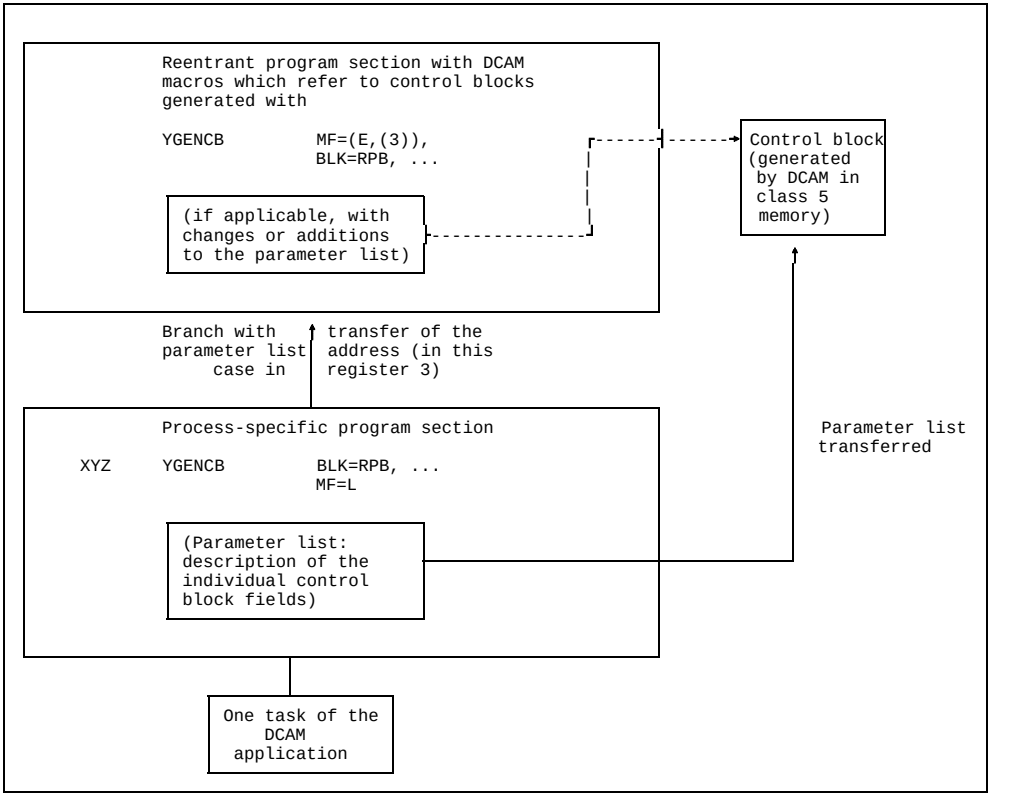
If the MF operand is omitted, the parameter list of the macro is used together with the instruction; however, this is undesirable, especially for reentrant programming.

The MF operand permits separate generation of the parameter list (list form or L form of the macro) and of the instruction (instruction or E form of the macro).

The YGENCB, YMODCB, YSHOWCB and YTESTCB calls can be used only when the DCAM subsystem has been loaded successfully. Note, too, that the DCAM subsystem status cannot be HOLD/DELETE when any of these macros is used. If a task successfully issued a DCAM command or a DCAM call before entering HOLD/DELETE, it can work with DCAM until the task is ended, despite a /HOLD subsystem or /DELETE subsystem (also applicable to %).

The YMODCB, YSHOWCB and YTESTCB macros can be replaced by shorter and faster instruction sequences addressing the fields by means of the YDDxxB DSECTs. Note, however, that this may necessitate recompilation in a later version.

The following figure shows the use of the MF parameter, using the YGENCB macro as an example.



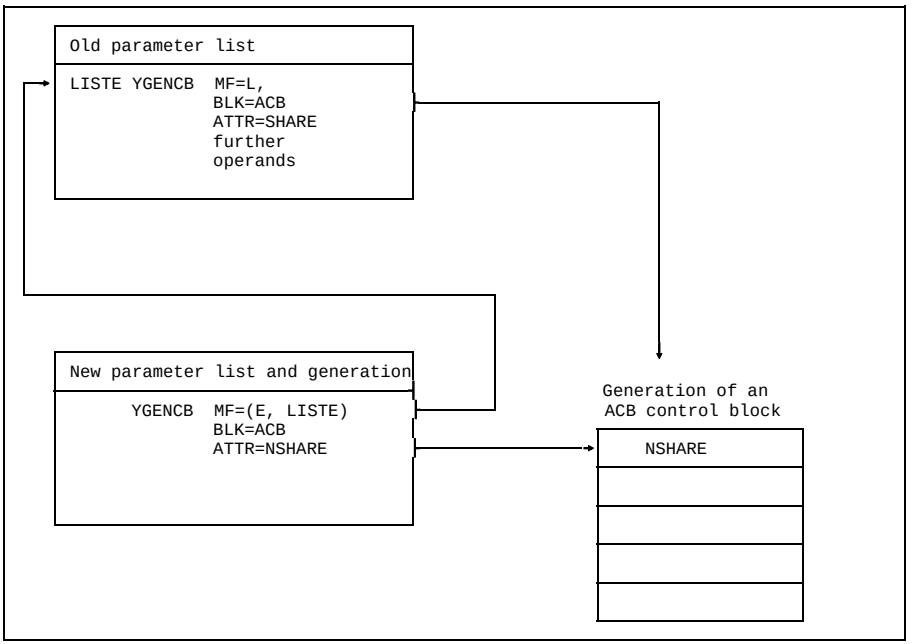
Use of the MF operand in YGENCB

The **list form (MF=L)** is used in the constants or variables section of the program. The macro is provided with a name to allow parameter list addressing. The parameter list is generated from the specified operands (constants, address constants etc.).

The **instruction form (MF=(E,addr))** is used in the instruction section of the program. It generates

– an instruction sequence comprising the SVC for DCAM and a reference to the parameter list whose address is specified in 'addr'. The address of the parameter list may also be specified in a register (registers 2 through 15). Register 1 may not be used.

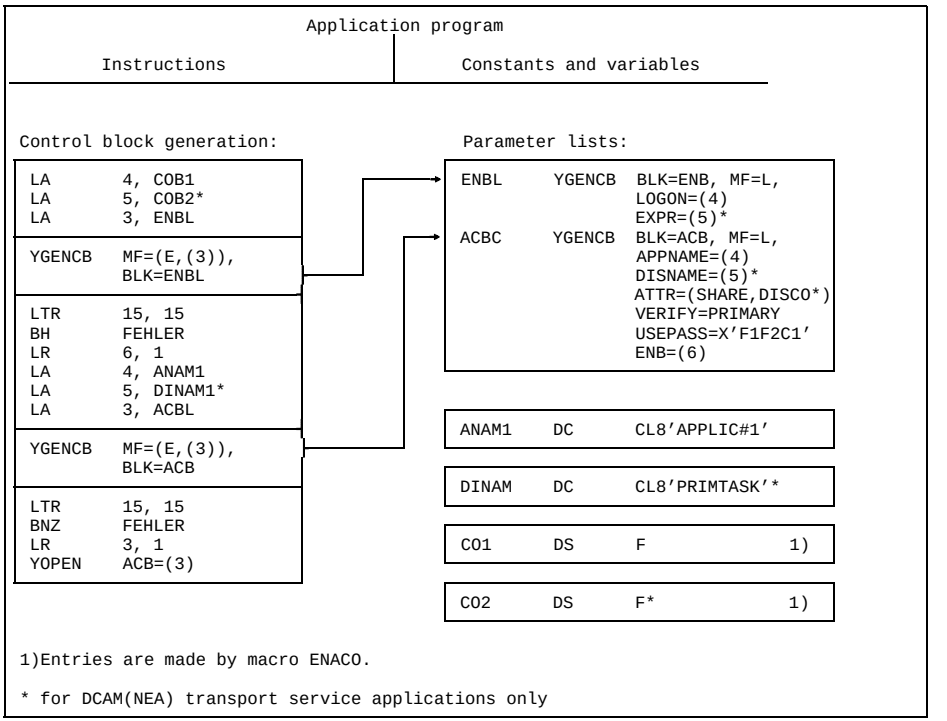
– an **optional parameter list** which supplements or updates the values in the addressed parameter list. This method has been introduced especially for DCAM to allow the universal use of parameter lists. The control blocks are generated from the operands of the separate list and the new values specified for it (see the figure below).



Example of modification during dynamic control block generation

The example in the figure below illustrates the use of the YGENCB macro. Control blocks are generated in class 5 memory (managed by DCAM). Only registers are used for the addressing (to demonstrate this option). The MF operand is used to achieve the same program layout as in the figure on page 12. This also shows how YGENCB is used in reentrant programming.

In this example an ENB and an ACB control block are generated dynamically to allow a DCAM application to be opened (see the section "Primary opening/ use of distribution codes", page 54)



Example of dynamic control block generation

2.1.3 Generation of control blocks by the user

With the aid of the macros YDDACB, YDDCCB, YDDENB, YDDRPB and, for DCAM(NEA) transport service applications, the macros YDDDCG and YDDDIP, the user can generate and modify control blocks directly (instead of using YGENCB, YMODCB, etc). These macros describe the formats of the control blocks (DSECT or CSECT); they are described in detail in the appendix, on page 309ff.

Note

Reserved or unused (unnamed) fields must always be filled with binary zeros.

2.2 Using control blocks during macro execution

After a control block has been generated, it can be used for the execution of an action macro. To this end, it must contain all the entries required for the execution of the macro. The chapter Using the DCAM functions, page 43ff, contains all the macros and their associated operands. A control block may contain entries that do not refer to the current action macro; only those entries that are required for the current macro are evaluated.

An RPB or CCB is flagged as active until the macro is terminated (synchronous execution) or until the instruction is terminated (asynchronous execution).

The flag is contained in the FLAG field of the control block (see the appendix, page 279):

FLAG=ACTIVE	The control block is currently being used (is reserved). This flag is used primarily to prevent a control block that is being used from being used for a further macro.
FLAG=INACTIVE	The control block is not being used at the moment. However, the contents of the control block have not necessarily been evaluated by the program. If it is necessary to indicate that a control block can actually be used for a further macro, suitable mechanisms must be provided within the application program.

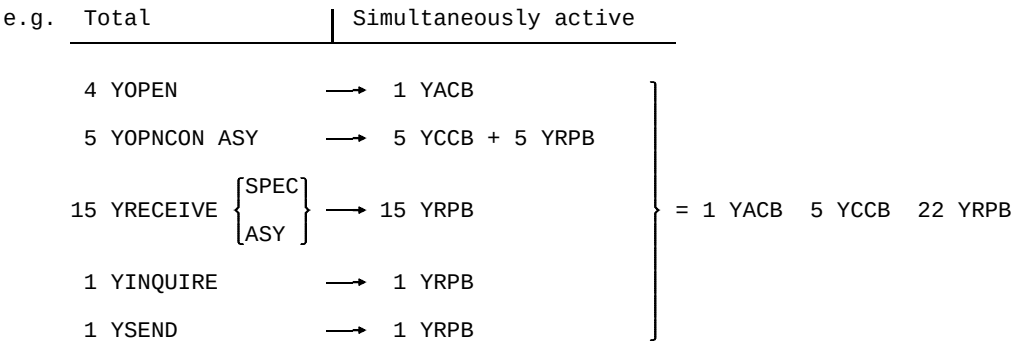
Thereafter, the user can process the control block fields set by DCAM, in particular the feedback information field and the identifier. Depending on the macro, further fields in the RPB or the CCB may be set (see the various functions in the chapter Using the DCAM functions, page 43ff).

A control block can be used for a number of calls, provided the individual fields are supplied with the current values.

Note

The macro YOPNCON (open a connection) destroys the reference address in the RPB which points to the CCB used. This is done to prevent an inadvertent attempt to open a connection twice.

If control blocks are used for several macros and the identifiers are saved and placed directly in the YRPB (see page 20), the result is a minimum number of control blocks (total number of control blocks required) for the maximum number of simultaneously active requests.



2.3 Use of identifiers

When DCAM creates a frequently used object which is used for a longer period (for more than one request), it assigns an identifier to it. The identifier is unique and is valid for the lifetime of the object. At the user interface, the object can then be accessed only with this identifier. Objects to which this applies are applications, connections, distribution structures and distribution code groups. The figure on page 23 shows a typical example of how the identifiers can be used.

The identifier returned by DCAM allows the user to reference the control block contents passed to DCAM. When a DCAM application is opened or a connection is established, the control blocks used for this purpose contain the valid identifier after execution of the macro.

Identifiers

Object	Name	Generated by:	Returned by DCAM in:	Can be entered by the user in: 1)	Invalid after:
Application	AID	YOPEN	AID field of the ACB control block	AID field of the RPB control block	YCLOSE or forced closure of the application
Connection	CID	YOPNCON	CID field of the CCB control block and CID field of the RPB control block used for macro YOPNCON	CID field of the RPB control block	YCLSCON or forced disconnection
Distribution structure*	DID	YOPNCON	DID field of the DIP control block and DID field of the CCB control block	DID field of the DIP control block and DID field of the CCB control block	YCLSCON or forced disconnection
Distribution code group*	GID	YOPNCON or YPERMIT	GID field of the DCG control block	GID field of the DCG control block	YCLSCON or forced disconnection or YFORBID

1) With YMODCB or with an action macro
* for DCAM(NEA) transport service applications only

The user now has two options:

– In further macros, he either addresses the applicable **control block ACB, CCB, DIP or DCG** and also, for DCAM(NEA) transport service applications, DIP and DCG containing the valid identifier, or

– he saves the returned identifiers in separate fields. The control blocks can then be used again or the areas can be deallocated.

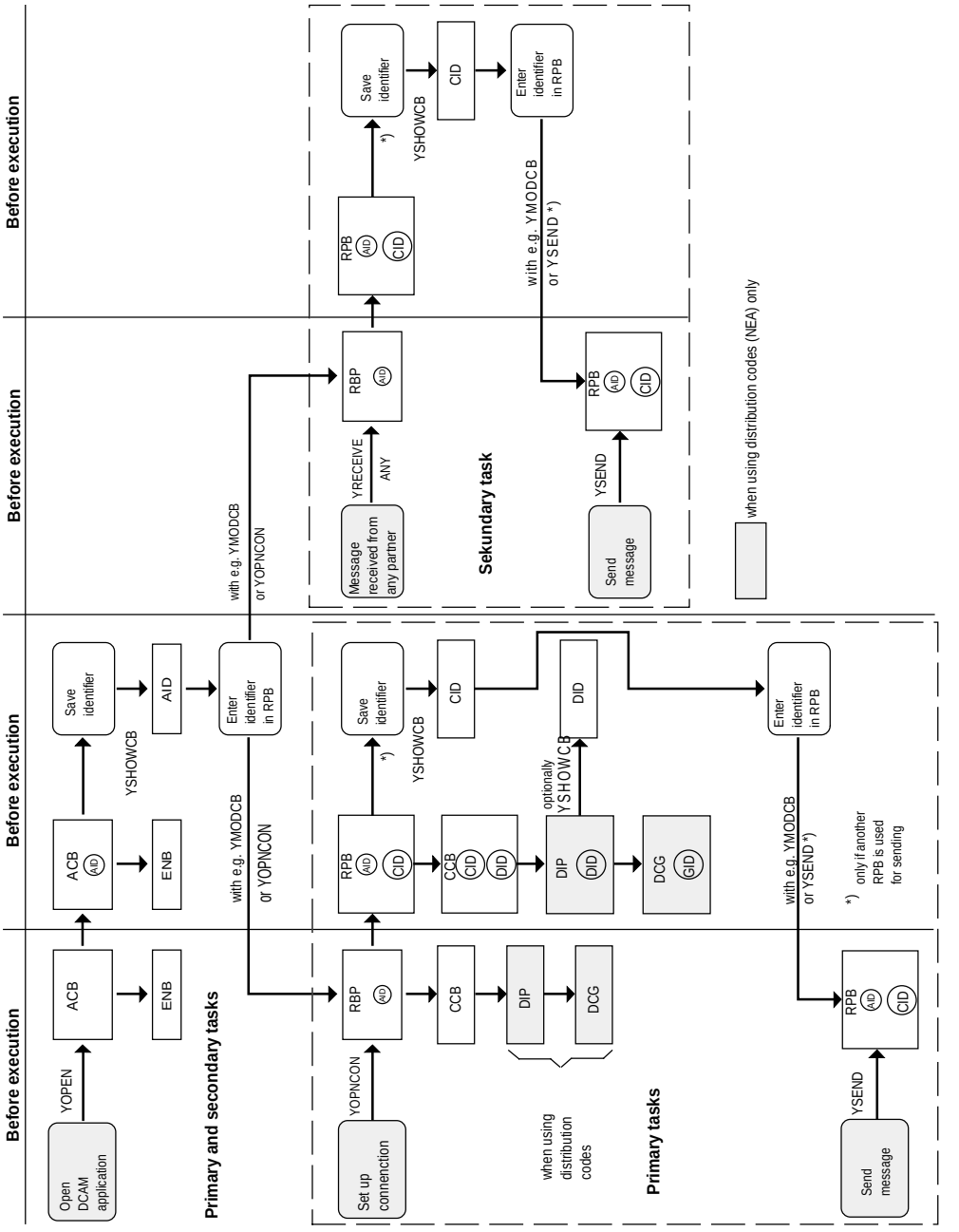
  If he wants to refer to the entries accessible by means of identifiers in DCAM, he must **enter the identifiers of the application (AID) and of the connection (CID) in the RPB used**.

Note

After a YOPNCON (connection setup) or a YRECEIVE (message reception), the connection identifier (CID) is also entered in the RPB. A consecutive YSEND (message transmission), YRECEIVE or YSENDREC may refer to this connection without having to enter the identifier beforehand.

If the identifiers AID and CID are entered directly in the YRPB control block with the aid of the DSECT YDDRPB, the corresponding flags YDDRAIFL YDDRAIFL and YDDRCIFL, respectively, must also be set. If the control block is modified by DCAM, then this is done automatically.

The entry in the RPB can be made in two different ways (see next section).

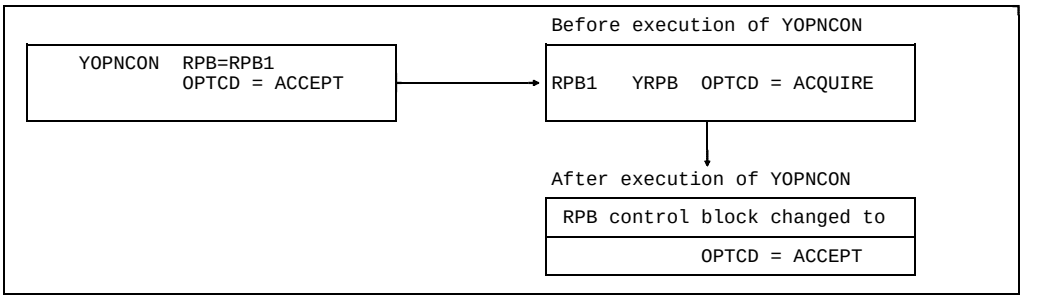

Before execution
Before execution
Before execution
Before execution
Open DCAM application
YOPEN
ACB
ENB
Primary and secondary tasks
ACB
AID
ENB
YSHOWCB
Save identifier
with e.g. YMODCB
or YOPNCON
Enter identifier in RPB
Set up connection
YOPNCON
RBP
AID
CCB
DIP
DCG
when using distribution codes
Primary tasks
RPB
AID
CID
*)
YSHOWCB
Save identifier
CID
CCB
CID
DID
optionally YSHOWCB
DIP
DID
DCG
GID
DID
with e.g. YMODCB
or YSEND *)
Enter identifier in RPB
Send message
YSEND
RPB
AID
CID
*) only if another RPB is used for sending
Message received from any partner
YRECEIVE
ANY
RBP
AID
Sekundary task
RPB
AID
CID
*)
YSHOWCB
Save identifier
CID
with e.g. YMODCB
or YSEND *)
Enter identifier in RPB
Send message
YSEND
RPB
AID
CID
when using distribution codes (NEA) only

2.4 Modifying the contents of control block fields

If the same control block is to be used as an operand field by different macros, the current values required for the macro must first be entered. To do this, or generally to make modifications in a control block, there are two methods available.

2.4.1 Modification during the use of the control block

This method is shown in the example in the figure below.



Example for modification during use of a control block

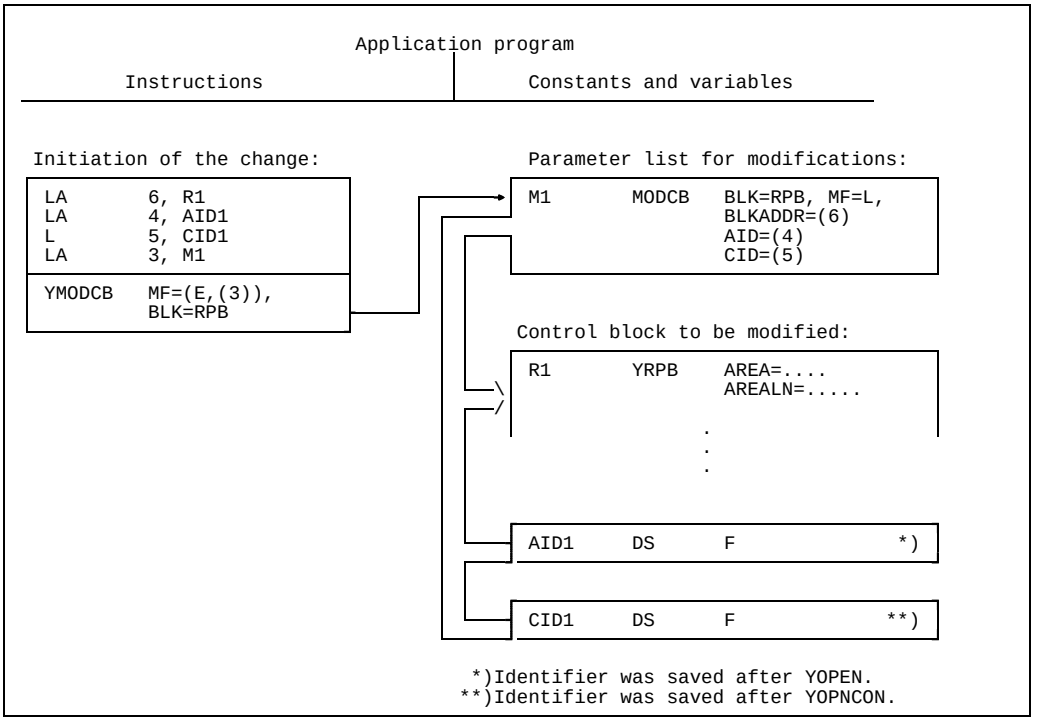
Only the contents of **RPB** control block fields can be modified during use. Modification is required when one RPB is used for several macros. The field contents to be modified can be specified in the macro together with the address of the RPB to be modified, with the result that old entries are overwritten in the control block. The original values specified during control block generation are lost. In contrast to static generation, addresses can also be specified in a register during modification just as with dynamic generation.

In the macro catalog (page 147 ff), the required fields are listed for all action macros. They can be modified in the way described above by means of the macro. In addition, all other RPB fields that can be modified by the user can be set (see appendix, page 279 ff).

Note that **default values cannot be used** here as shown in the chapter Using DCAM functions, page 43ff, where a separate control block is generated for each action macro so that default values need not be specified and the processing time is reduced.

2.4.2 Modification by means of a separate macro

This is illustrated by the figure below and the figure on page 22.



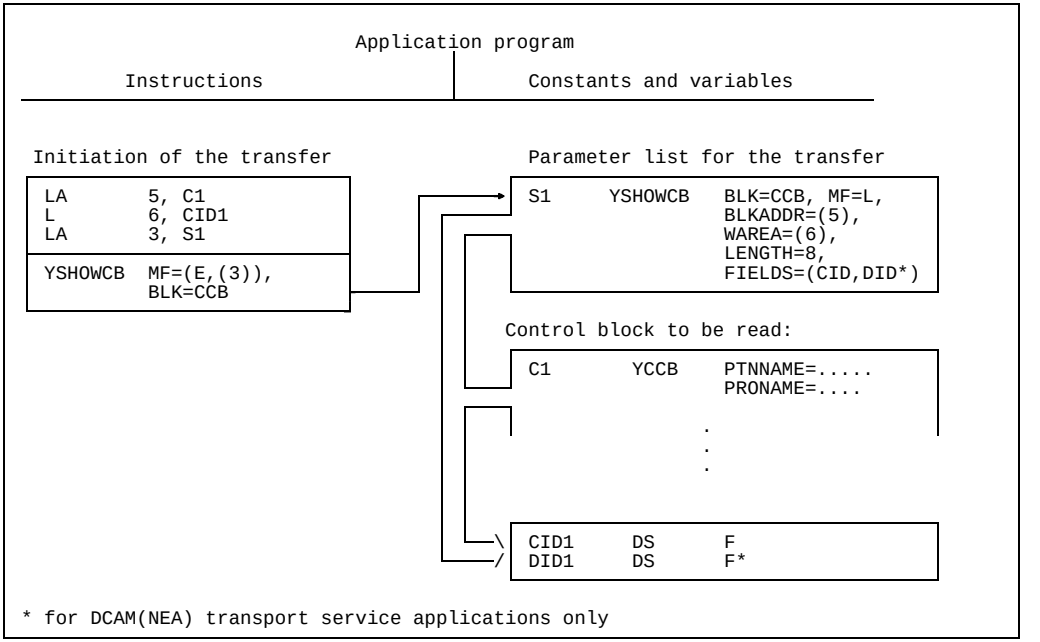
Example of modification of a control block with YMODCB

The **YMODCB** macro can be used to modify the contents of one or more fields of a control block. Bear in mind that the **YMODCB** macro can be used only when the DCAM subsystem has been loaded successfully. Note, too, that the DCAM subsystem status cannot be HOLD/DELETE when this macro is used. If a task successfully issued a DCAM command or a DCAM call before entering HOLD/DELETE, it can work with DCAM until the task is ended, despite a /HOLD subsystem or /DELETE subsystem (also applicable to %).
The fields that can be modified are listed in the appendix, page 279.

Note that a control block can be modified only when it is not being used for instruction execution (FLAG=ACTIVE, see page 18). The MF operand (see page 13) can be used to support reentrant programming. It allows the modification parameter list to be defined separately from the macro in the program.
The YMODCB macro can be replaced by shorter and faster instruction sequences addressing the fields by means of the YDDxxB DSECTs. Note, however, that this may necessitate recompilation in a later version.

2.5 Reading and saving control block fields

This is illustrated by the figure below.



Example of saving the contents of control block fields

To read individual control block fields and transfer them to other memory areas, the **YSHOWCB macro** should be used. It allows, for example, the identifiers returned by DCAM to be saved (see the figure on page 22). To allow access to the various fields, each field has a keyword (see the table in the appendix, page 279ff) and a fixed length. Bear in mind that the **YSHOWCB** macro can be used only when the DCAM subsystem has been loaded successfully. Note, too, that the DCAM subsystem status cannot be HOLD/DELETE when this macro is used. If a task successfully issued a DCAM command or a DCAM call before entering HOLD/DELETE, it can work with DCAM until the task is ended, despite a /HOLD subsystem or /DELETE subsystem (also applicable to %).

The MF operand can be used to support reentrant programming (see page 13ff). The parameter list of the fields to be read can then be defined separately from the macro in the program.

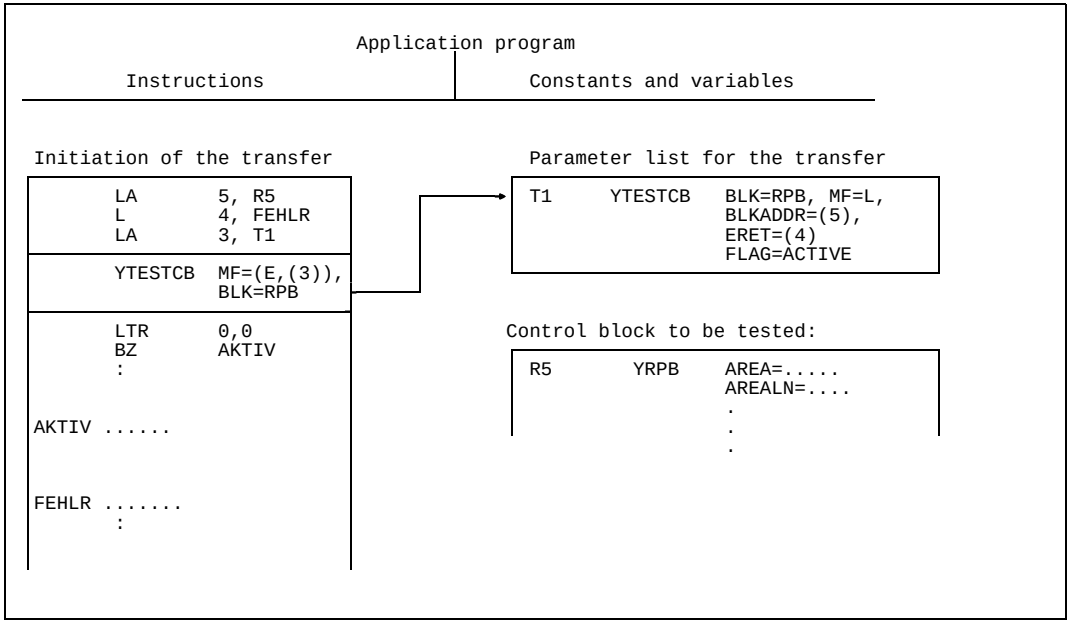
The YSHOWCB macro can be replaced by shorter and faster instruction sequences addressing the fields by means of the YDDxxB DSECTs. Note, however, that this may necessitate recompilation in a later version.

2.6 Reading and testing control block fields

The **YTESTCB macro** allows the contents of individual control block fields to be tested without having to transfer the fields to a separate memory area. The keywords and the test values which can be used for this are listed in the table of control block fields in the appendix, page 279ff. Bear in mind that the **YTESTCB** macro can be used only when the DCAM subsystem has been loaded successfully. Note, too, that the DCAM subsystem status cannot be HOLD/DELETE when this macro is used. If a task successfully issued a DCAM command or a DCAM call before entering HOLD/DELETE, it can work with DCAM until the task is ended, despite a /HOLD subsystem or /DELETE subsystem (also applicable to %).

The MF operand can be used to support reentrant programming (see page 13ff), thus enabling the parameter list to be defined separately from the macro in the program.

The following figure shows an example of its use: in a control block, the field which indicates whether the block is currently being used for instruction execution is tested.



Example of testing a control block field

The YTESTCB macro can be replaced by shorter and faster instruction sequences addressing the fields by means of the YDDxxB DSECTs. Note, however, that this may necessitate recompilation in a later version.

2.7 DCAM Instruction execution

Two methods are available to the user for executing DCAM instructions:

– **Synchronous execution:** Easy to handle, few specifications, but no optimum usage of wait times.

– **Asynchronous execution:** Optimum usage of wait times, but greater programming complexity (only with YOPNCON, YRECEIVE, YSENDREC and data flow control (YSEND-GO)).

Tables of register usage during execution are provided in the appendix, starting on page 301.

2.7.1 Synchronous execution

During synchronous execution (OPTCD=SYN operand), control is returned to the program only after the instruction has been executed. The return code is available in register 15 and in field FDBK of the RPB.

But what happens if the macro cannot be performed immediately, e.g. with YOPNCON, YRECEIVE or YSENDREC? In this case one can either wait for the event (e.g. the arrival of the message) to occur or have control returned immediately with an appropriate indication:

– **OPTCD=Q** (OPTCD = option command; Q = queue)

  The macro is placed **in a queue** until the expected event occurs. The maximum wait time can be specified with TOVAL=n (timeout value). The macro is terminated, i.e. control is returned to the user, when the event occurs or the wait time elapses.

– **OPTCD=NQ**

  The macro is **terminated immediately** even if the expected event has not occurred. In the case of a YOPNCON, OPTCD=ACCEPT (see the section "Connection-oriented macros", page 69), this may mean that a connection request was not received from the partner. In the case of a YRECEIVE or YSENDREC (see the section "Data communication-oriented macros", page 120), the message has not yet arrived. In this case, the user may have to repeat the macro at a later time.

Note

Only **one** synchronous macro which causes a waiting period (YOPNCON ACQUIRE, YOPNCON (ACCEPT, Q) YRECEIVE Q, YSENDREC Q) may be active at any time. A second synchronous macro with a waiting period (from a contingency task, started during the waiting period) is rejected.

2.7.2 Asynchronous execution

Instructions which may have to wait for the occurrence of an event can also be executed asynchronously. This means that the **OPTCD=ASY operand** must be specified, which is possible only for **YOPNCON** (see the section "Connection-oriented macros", page 69) and **YRECEIVE** or YSENDREC (see the section "Data communication-oriented macros", page 120). A special case in this respect is the data flow control with the GO signal. If the flow of data is impeded, the application program can specify that it is to be informed asynchronously when the data flow is resumed. This is the case for **YSEND** and **YSENDREC** if **PROC=SIGNAL** is specified in the **YOPNCON** in the CCB.

For asynchronous execution, DCAM interacts with the P1 Eventing/ Contingency Interface.

The following steps are required:

Step 1: A separate EID (**event item identifier**) must be defined for each asynchronous macro. The ENAEI (enable event item) macro is used for this. The identifier is used by the system to post the occurrence of the event.

Step 2: A **contingency routine** has to be defined if it is to be started upon the occurrence of the event (asynchronous SOLSIG, see step 4). The ENACO (enable contingency routine) macro specifies the routine and its start address. If a contingency routine is not to be performed or if its execution is to be dependent on the result of other routines, it can be given a specific priority from 1 (lowest) to 127 (highest). The priority can be modified during processing (LEVCO macro, see manual "Executive Macros").

The routine is also initiated if the maximum wait time elapses or an error occurs. When a contingency routine is started, the user is given information in various registers (see the register usage tables in the appendix, page 301). All other registers must be set by the user (including the base registers). The register contents of the interrupted routine or of the main routine can be accessed with the CONTXT macro (see "Executive Macros").

Step 3: The **event item identifier EID** has to be specified if the OPTCD=ASY operand is specified **in the YOPNCON, YRECEIVE or YSENDREC** macro. To this end, the address of the field in which the ENAEI macro has stored the identifier is specified. It is also necessary to indicate that **the macro is to wait (OPTCD=Q)** - except for YOPNCON, (OPTCD=ACQUIRE) - and, if applicable, **how long it is to wait (TOVAL)**. The wait time is reset when the instruction can be executed (e.g. message passed in YRECEIVE) or the maximum wait time expires

or in the event of an error (e.g. the partner clears down the connection).

The **event item identifier** must also be specified if a **YSEND** or **YSENDREC** is to be executed on a connection which was established with **YOPNCON PROC=SIGNAL**. If, due to a bottleneck on the connection, the message cannot be sent to the partner, DCAM terminates the macro with FDBK=X'10040C00' ("Wait for GO"). When the bottleneck has been cleared, DCAM issues the GO signal.

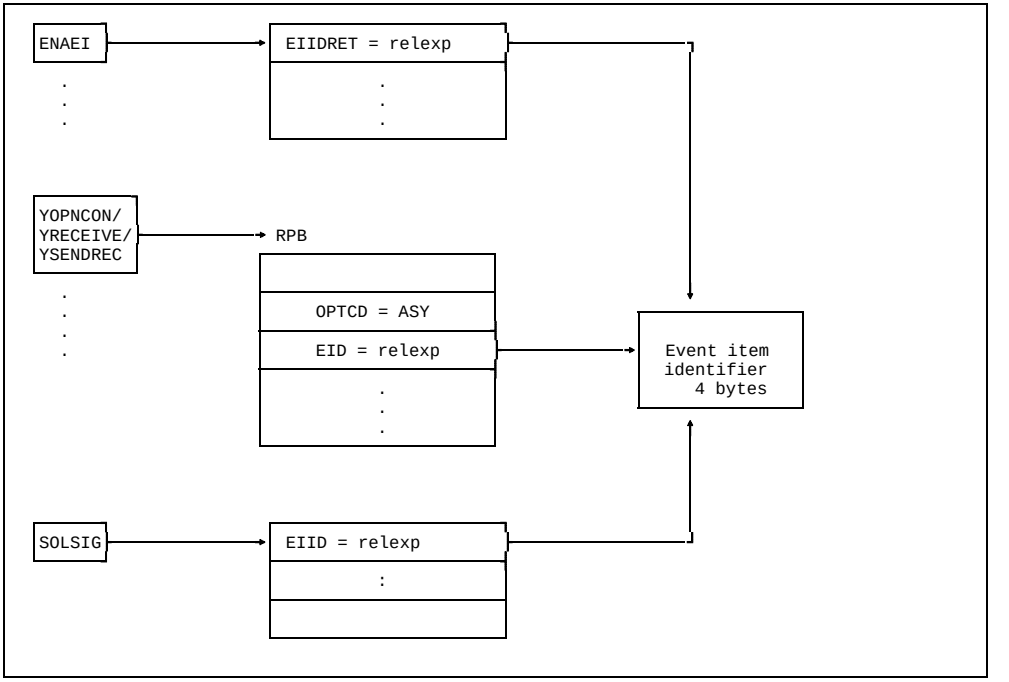
The RPB remains locked until the request is terminated or until the GO signal is received. Register 15 should be used for evaluation of the feedback, since the field FDBK in the RPB may already contain the result of the request.

Step 4: DCAM will make an entry ("event"; "signal") for the event item identifier EID as soon as execution, successful or not, has been completed (e.g. after the message has been entered in the user memory or the wait time has elapsed).
By means of the **SOLSIG** (solicit signal) **macro** the user specifies his or her readiness to process the requested event. This SOLSIG macro may again be processed either synchronously or asynchronously. The field FDBK in the RPB is available for evaluation of the feedback.

Synchronous SOLSIG (figure below):

A synchronous SOLSIG is placed at the point in the program at which event processing starts. Again, one of two methods can be selected:

- The system waits until the event occurs, if it has not yet occurred (COND=UNCOND operand); the maximum wait time is specified with the LIFETIM operand. If the wait time elapses without the event (successful or unsuccessful) having occurred and if no error occurred, the SOLSIG can be given again for the same event identifier.
- The macro is terminated immediately (COND=IMMED operand). If the event has not yet occurred, the macro may have to be repeated later.

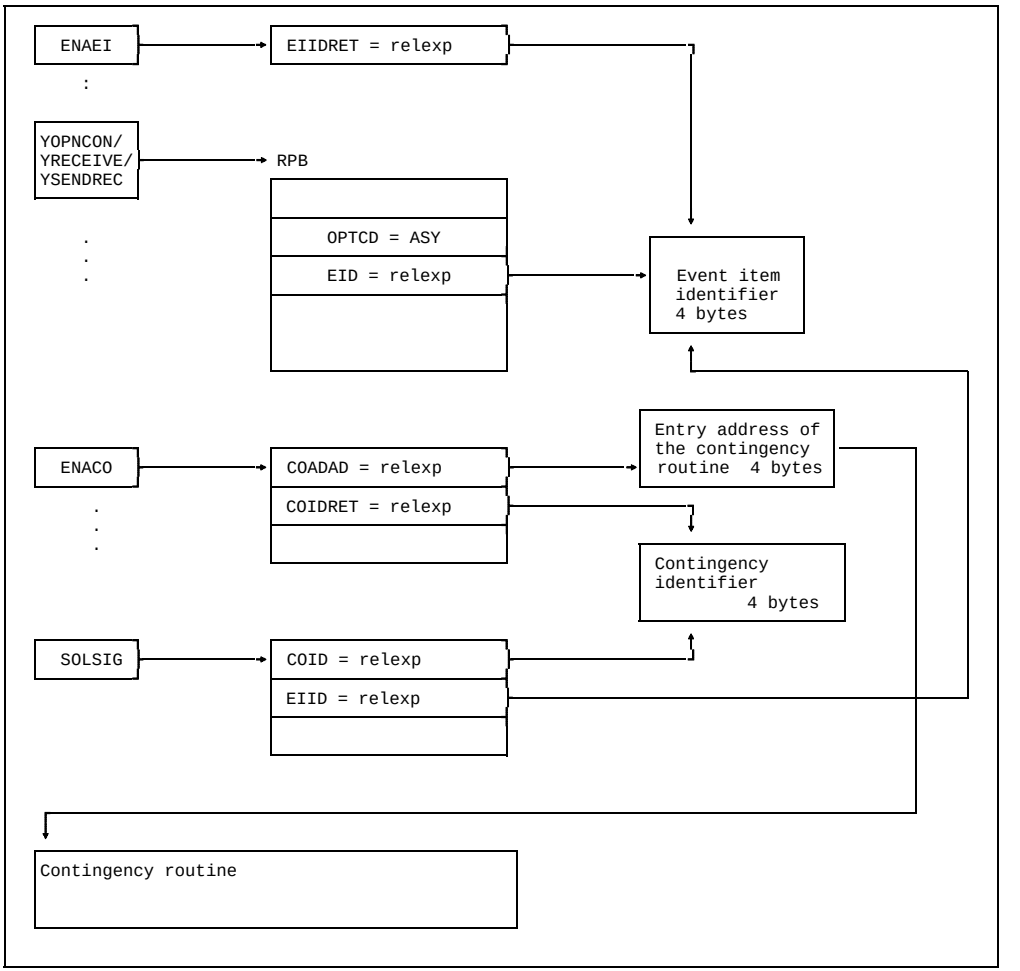


Specifications for synchronous execution of SOLSIG

Asynchronous SOLSIG (figure below):

An asynchronous SOLSIG can be placed immediately after step 1 or 2. The reference to the contingency identifier COID defined in step 2 (ENACO, see "Executive Macros") is established by specifying the address of the field in which the identifier is stored (COID operand).

If the (successful or unsuccessful) event occurs, the contingency routine whose identifier is specified here is initiated. This routine can be used to process the event.



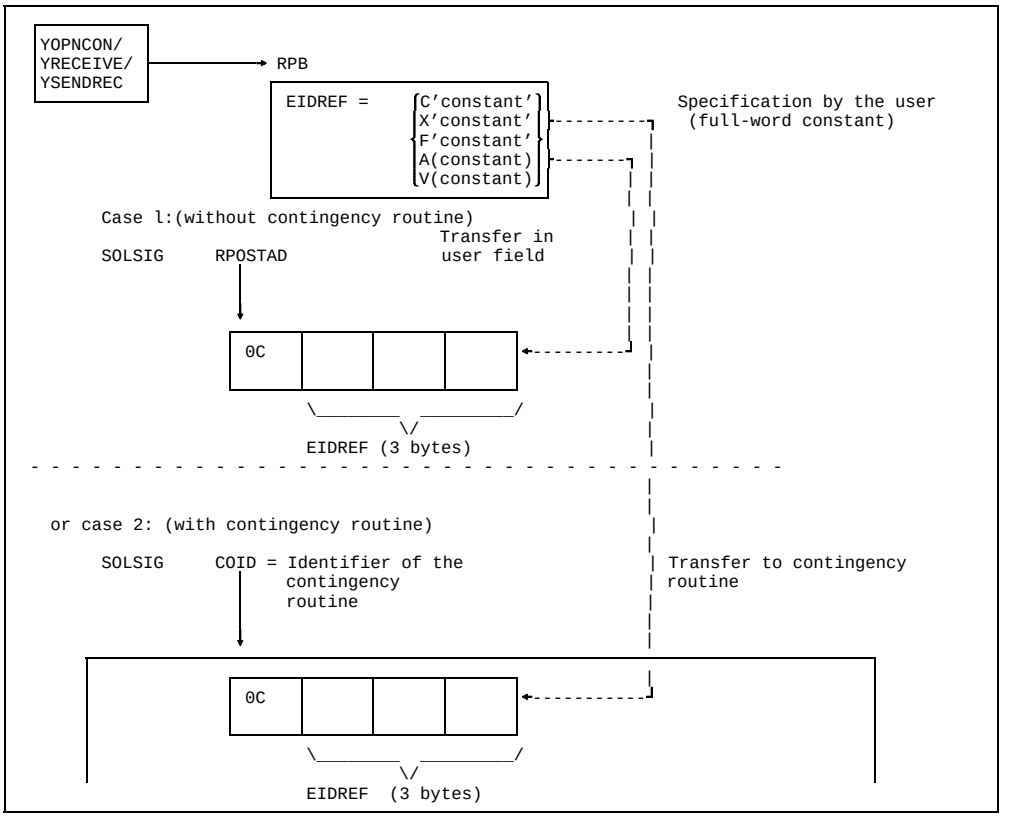
Specifications for asynchronous execution of SOLSIG

The DCAM macro is terminated immediately, i.e. at a time at which the instruction was accepted by DCAM but not necessarily executed. The **feedback information** in the control block used does not yet contain any information on instruction execution by

DCAM (see the appendix, page 286). Information on instruction execution and termination is given to the user only if he issues a SOLSIG macro for the relevant event identifier.

If he does not want to save it himself, the user can specify the address of the control block used as the macro reference in the **EIDREF** or **EIDREF2 operand** of the RPB control block. It will then be returned to him either in the RPOSTAD field (for the synchronous SOLSIG) or in register 3 or 4 of the contingency routine (asynchronous SOLSIG) for later access to the control block and processing of the feedback information (see also the figure below).

The high-order byte of the reference (EIDREF) is overwritten by DCAM with hexadecimal X'0C'. EIDREF2 remains unchanged. If an address is specified as the reference, it is advisable to use EIDREF2 as well (in 31-bit mode the addressX'0C000000' is rarely addressed). The RPOSTAD field must then be 8 bytes long and "RPOSTL=2" must be specified for SOLSIG.



Passing event information

2.8 Asynchronous DCAM notifications

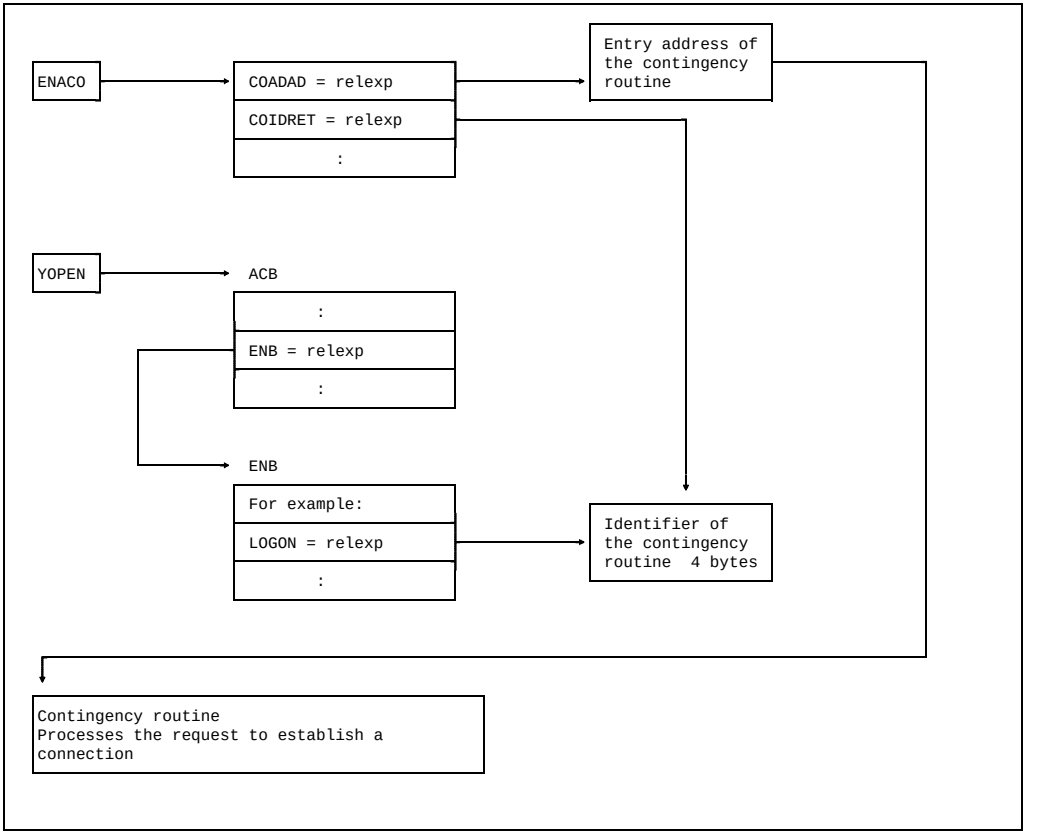
DCAM notifies the user of certain events occurring in the data communication system if the user has defined a **contingency routine** for this purpose. This is done during the opening of the DCAM application (see the section "Existence-oriented macros", page 45ff). The user enters in the ENB control block the contingency routine identifiers returned to him when the routines were defined with the ENACO macro.

The contingency routines are permanently assigned to a task, but their identifiers can be used by several DCAM applications if these were opened by the same task.

Once the assignment between the DCAM application and the contingency routine has been made, it cannot be modified again for the lifetime of the DCAM application.

If a contingency is not to be performed or if its execution is to be dependent on the results of other routines, it can be given a specific priority from 1 (lowest) to 127 (highest) by means of the LEVCO macro (see "Executive Macros"). The default value is 1, which still permits interruption of the main routine since the latter has the default priority 0. However, the main routine too can specify a higher priority and thus delay the interruption, if required.

The LOGON contingency routine example in the figure below shows which specifications are mandatory for definition.



Specifications for definition of a contingency routine for DCAM event notifications

When a contingency routine is started, the user is given **information in various registers** (see the register usage tables in the appendix, page 301). All other registers must be set by the user (including the base registers). Access to the register contents of the interrupted routine or of the main routine is possible with the CONTXT macro (see "Executive Macros").

The individual events of which the program can be informed by means of asynchronous notifications are described in detail below.

2.8.1 LOGON

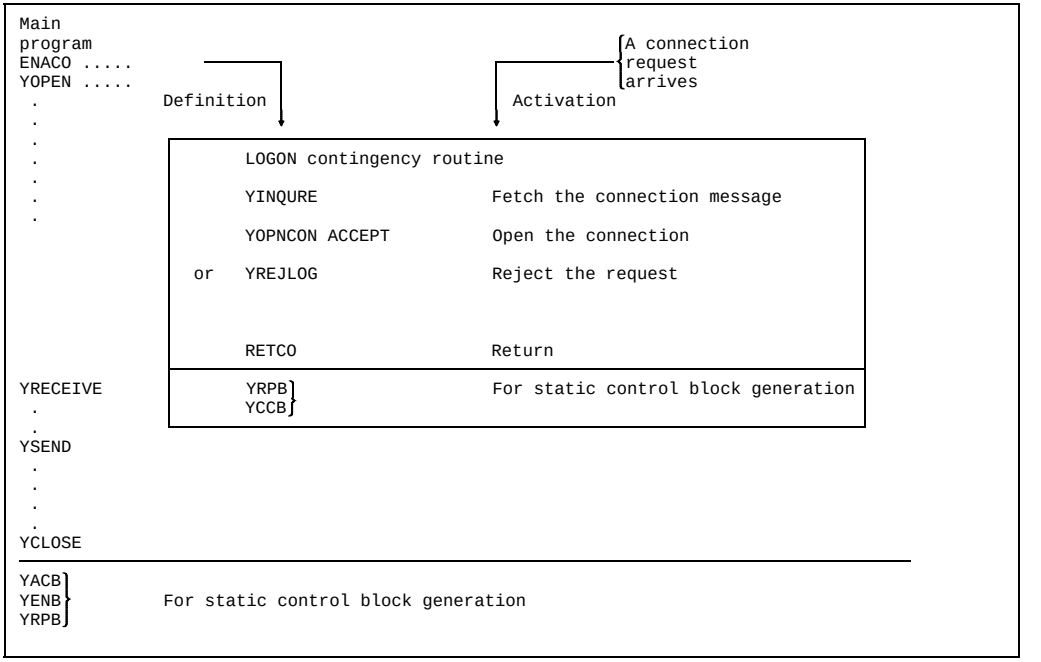
When a **connection request** with the LOGON attribute and the current state START is addressed to a DCAM application, this notification is issued (see DCAM Program Interfaces). DCAM expects a user response to this notification. 'No response' is interpreted as a rejection of the connection request (after a period of time defined in the communication system for such events). The function macros required for the response, if any, need not be issued in the LOGON contingency routine itself.

The connection request indicated by the LOGON notification can be accepted with a YOPNCON (OPTCD=ACCEPT) macro (see page 80) or rejected with a YREJLOG macro (see page 111). To obtain the information required for accepting (YOPNCON) the request, the user must issue an appropriate YINQUIRE macro (see page 89).

The following **information** is transferred when the LOGON contingency routine is entered:

- The identifier of the DCAM application (AID) in register 3.
- The identifier of the connection request (LID=logon identifier) in register 4.
- The length of the connection message transmitted by the partner in register 5.

The example in the figure below shows which macros are used when a LOGON contingency routine is provided.



Example of the use of a LOGON contingency routine

2.8.2 LOSCON

This notification is generated when the **connection** with a partner **is cleared** either by the operator of the communication system or by the partner. If there is no LOSCON contingency routine defined in a DCAM application, the loss of connection becomes known only when the next macro referring to this connection is issued.

The following **information** is transferred when the LOSCON contingency routine is entered:

- The identifier of the DCAM application (AID) in register 3
- The identifier of the connection (CID) in register 4
- The accompanying information stored by the user in USERFLD in register 5
- The reason for the loss of connection, or whether the connection was only lost after the warning period had expired, in register 6.

2.8.3 PROCON

As soon as the DCAM application is opened or the /BCIN command (see "Network Management in BS2000") is given at the console for the partner's processor, DCAM informs this application about the **partners predefined** in the communication system (XSTAT, see "Generating a Data Communication System"), i.e. the partners to whom connection requests are to be sent.

The PROCON contingency routine is performed once for each defined partner so that a connection request can be sent to each.

DCAM does not expect a response to this notification. The user must decide whether a connection is to be established.

The following **information** is transferred when the PROCON contingency routine is entered:

- The identifier of the DCAM application (AID) in register 3
- The symbolic name of the partner in registers 4 and 5
- The symbolic name of the processor to which the partner is connected in registers 6 and 7.

2.8.4 COMEND

The user receives this notification when the **communication access system is terminated** (SHUTDOWN) or when the **closure of the DCAM application** is forced (see the section "Closing a DCAM application", page 67). In the case of a shutdown or a forced closure the application is closed by DCAM and no user response is expected.

In the case of a pertinent warning, it is not possible to open another DCAM application or to establish another connection. Shutdown or forced closure occurs after a period of time that is predefined for the communication system (see "Network Management in BS2000").

The following **information** is transferred when the COMEND contingency routine is entered:

- The identifier of the DCAM application (AID) in register 3
- The reason (shutdown or forced closure, shutdown warning or forced closure warning) in register 4.

2.8.5 EXPR

This section applies only to DCAM(NEA) transport service applications.

When an **express message** arrives for the DCAM application, it can be transferred immediately to the relevant program by means of this notification. As a rule, the program controlling the primary task will receive the express message; this is the case when

- distribution codes are used;
- distribution codes are not used and there is no link with a secondary task for this connection (CS state).

A secondary task will receive this notification only if it has set the CS state on the relevant connection. Express messages are not transferred to the user on connections defined with EDIT=SYSTEM, i.e. those using virtual terminals; consequently, EXPR notifications are not generated either.

The following **information** is transferred when the EXPR contingency routine is entered:

- The identifier of the DCAM application (AID) in register 3
- The identifier of the connection on which the message arrived (CID) in register 4
- The accompanying information on the connection (USERFLD) in register 5
- The contents of the express message in registers 6 and 7
- The sequence number of the message in register 8.

2.8.6 TACK

This section applies only to DCAM(NEA) transport service applications.

When a **transport acknowledgment** arrives for the DCAM application, it can be transferred immediately to the pertinent program by means of this notification.

To identify the message, a sequence number (SEQNO) is assigned to it during transmission (YSEND) by the user.

The transport acknowledgments are distributed to a shareable DCAM application in accordance with the definition in the YOPEN of the primary task (ATTR operand in the ACB):

- PRIMTASK: All acknowledgments are transferred to the primary task, i.e. it is pointless to provide a TACK contingency routine for a secondary task.
- REQTASK: The acknowledgment is transferred to the task that sent the message.
- NOTACK: Acknowledgments are not transferred for this DCAM application. The definition of a contingency routine would be pointless.

The following **information** is transferred when the TACK contingency routine is entered:

- The identifier of the DCAM application (AID) in register 3
- The identifier of the connection (CID) in register 4
- The accompanying information on the connection (USERFLD) in register 5
- The positive/negative acknowledgment indicator in register 6
- The sequence number of the acknowledged message in register 7.

2.8.7 SECOND

This contingency routine enables a primary task to obtain information concerning a secondary task in the following three cases:

- Where a secondary task opens the application after a primary task.
- Where a secondary task logs off from the application.
- Where a message is received for a distribution code name which is not assigned to any secondary task (NEA transport service only).

The following information is transferred when the SECOND contingency routine is entered:

- The identifier of the DCAM application (AID) in register 3
- The reason for initiation of the SECOND routine (see above) in register 4
- The distribution code name (if any) in registers 5 and 6 (NEA transport service only).

3 Using the DCAM functions

Before you read this chapter, you must be familiar with the DCAM functions as described in the chapter ’DCAM functions’ of "DCAM Program Interfaces". The relevant chapter is organized the same way in both manuals, enabling you to consult "DCAM Program Interfaces" without having to use cross-references.

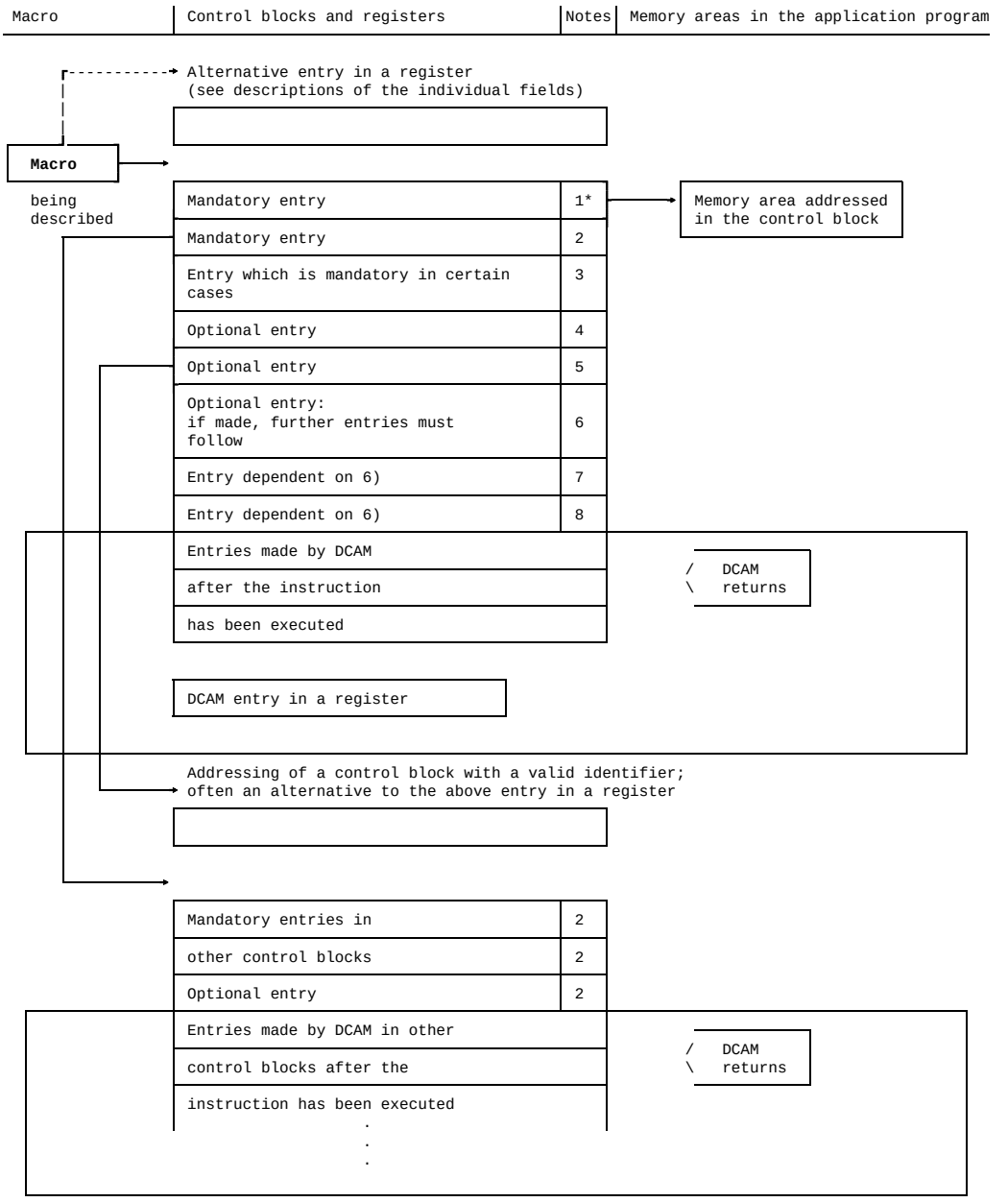
So you can get to know the interface quickly, particular attention has been paid to presenting it as simply as possible. For each action macro issued to DCAM **a separate, specially generated control block is used** (see page 9). As a result, **default values do not have to be specified.** Experienced users wishing to use control blocks more than once are referred to the alphabetic catalog (page 147 onwards). The contents of the individual control block fields and their interpretation, however, is described in this chapter.

For the meaning of abbreviations you should see the appendix, page 305 onwards.

A further aid is provided by the schematics. The figure below shows you how to interpret them.

In these diagrams no account is taken of how the control block is generated by the user (statically or dynamically).

Each macro description is followed by at least one example. The examples are coding skeletons of an implementation. They are not complete, nor are they to be regarded as models since various coding methods are possible.



*) The numbers refer to the field descriptions in the text (e.g. 1)2)3) etc.).

Explanation of the graphical representation

3.1 Existence-oriented macros

Existence-oriented macros are required for the generation and termination of DCAM applications. They also include the macros used for modifying or testing the state of DCAM applications.

Existence-oriented macros are:

- **YOPEN** **open DCAM application**
- **YCLOSE** **close DCAM application**
- **YINQUIRE** **query the state of a DCAM application (partner)**
- **YSETLOG** **set the state of a DCAM(NEA) transport service application**

Information on dynamic name assignment (LINK) is not included here; it is given in a separate section (page 139).

DCAM tasks can process the asynchronous notification **COMEND** 'terminate access methods and close DCAM application' (see page 39) with an appropriate contingency routine.

Notes

As soon as a batch or dialog task opens a DCAM application, DCAM changes its scheduling attribute (see the BS2000 "System Administrator's Guide") to "TP" if the corresponding JOIN entry authorizes the task for this.

As soon as the dialog task closes its last DCAM application, DCAM resets it to "DIALOG".

As soon as the batch task closes its last DCAM application, DCAM resets it to "BATCH".

3.1.1 Opening a DCAM application

A DCAM application can be

– non-shareable (it is opened by only one task) or

– shareable (it is opened by the primary task and by secondary tasks).

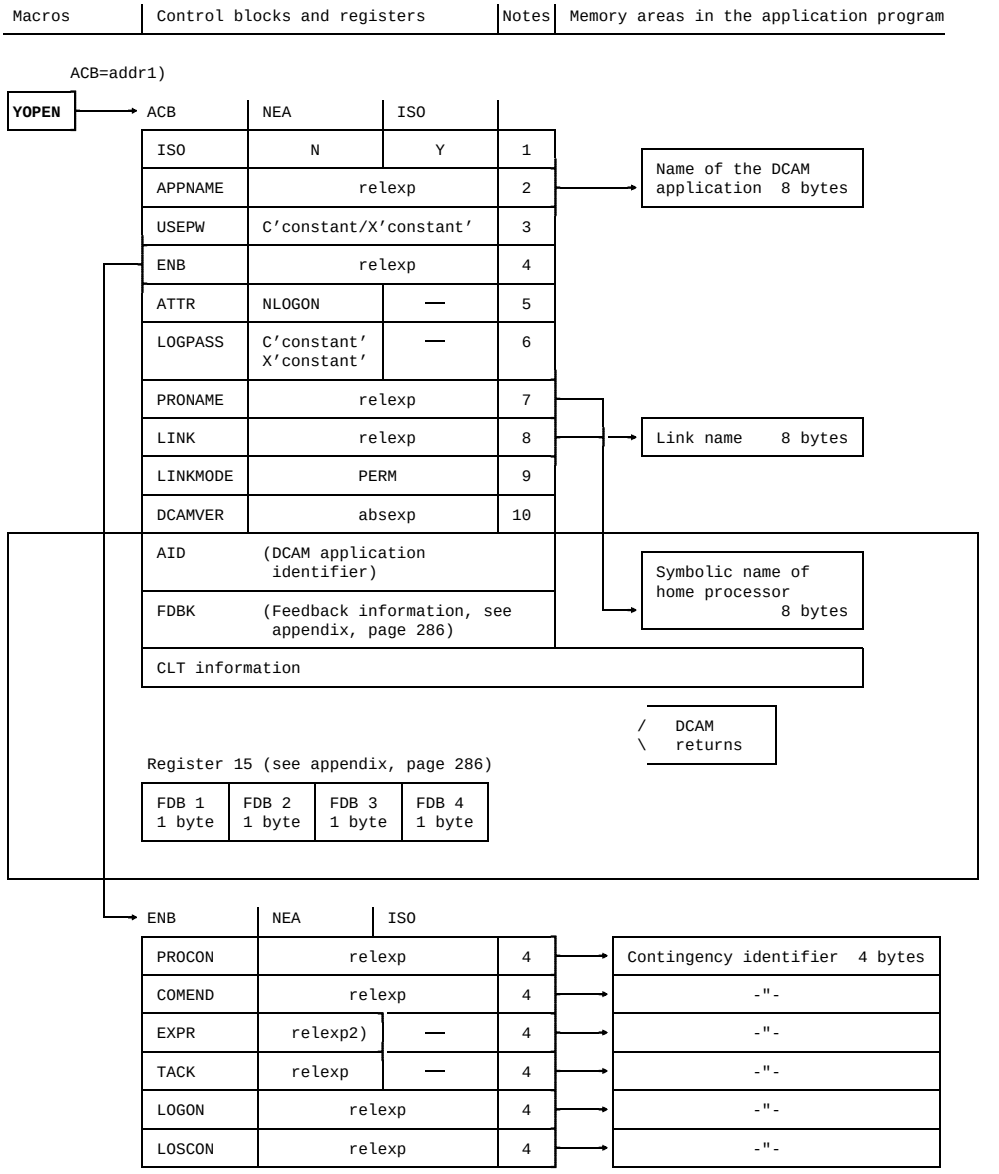
If it is shareable, messages can be distributed by the standard method (using the originator-oriented or the common receiver queue).

For DCAM(NEA) transport service applications, messages can also be distributed by means of distribution codes.

For all DCAM(ISO) transport service applications, the ISO attribute must be set in the application control block. The ISO attribute then applies to all connections maintained by this application.

The above possibilities result in five different versions of the YOPEN macro; these are described below.

3.1.1.1 Non-shareable DCAM application



1) Address of the control block that describes the DCAM application
2) only in conjunction with EDIT=USER

A non-shareable DCAM application is to be opened.

Mandatory specifications:

1) ISO=Y, to indicate that the ISO transport service is to be used. This specification is mandatory for DCAM(ISO) transport service applications. This function is not available for DCAM(NEA) transport service applications. If this specification is omitted, the NEA transport service is used as the default.

2) The name of the DCAM application, if an application predefined in the network file (RDF) is to be opened. If no name is specified, it is generated by DCAM. References to this application must then be established by means of the identifier (AID) returned by DCAM.

3) The password, if a predefined application is protected by a password.

Optional specifications:

4) The address of the ENB control block. If omitted, no DCAM notifications are processed (see page 34).

5) Connection requests are not to be processed (NLOGON). If omitted, requests are processed (LOGON) (NEA transport service only).

6) The password which has to be given by the partner when requesting connection. If omitted, no password check is performed (NEA transport service only).

7) The address of the field in which DCAM enters the symbolic name of its own processor.

8) The address of the field containing the link name.

9) That the information in the ACB is to be overwritten with the information from the CLT entry (CLT = communication link table) identified by the link name (PERM; permanent change). If this entry is omitted, the CLT entry information will be interpreted only for this call. The information in the ACB will not be changed (TEMP; temporary change).

10) The number of the DCAM version if the new functions introduced as of DCAM V8.0 are to be used in the task. 8.0 must be specified if the new functions are to be used. If this is not specified, or if the specification is $\neq$ 8.0, a DCAM version $\leq$ 7.0 is assumed.

Example 1

The DCAM(NEA) transport service application to be opened is to be non-shareable. Its name is generated by the system. The application is not to accept connection requests and does not process asynchronous notifications.

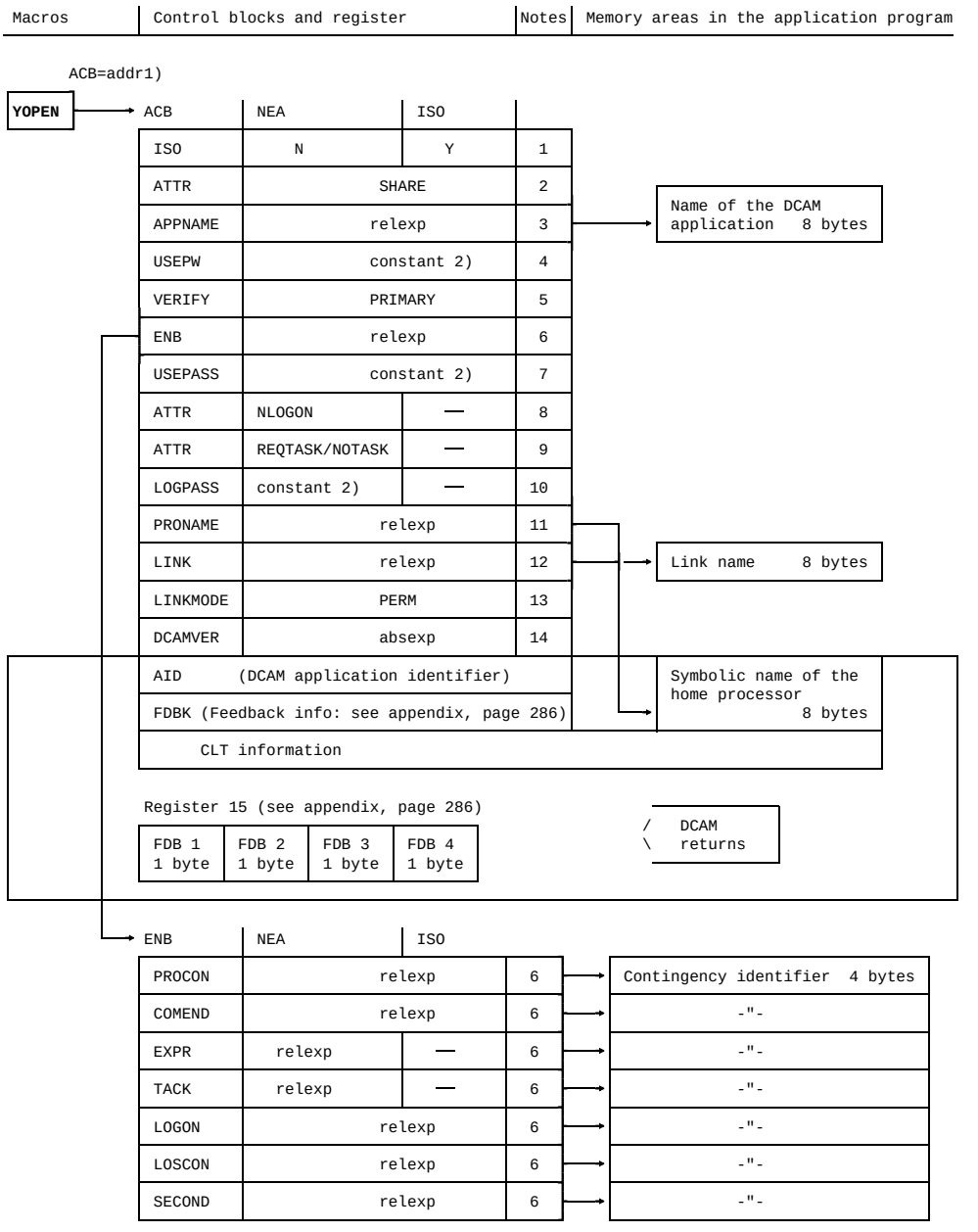
```
                .
                .
                .
                YOPEN       ACB=ACB1
                .
                .
                .
ACB1            YACB        ATTR=NLOGON
                .
                .
                .
```

Example 2

The DCAM(ISO) transport service application to be opened is to be non-shareable. Its name is generated by the user. The application does not process asynchronous notifications. DCAM is to return the symbolic name of the home processor.

```
                .
                .
                .
                YOPEN       ACB=ACB1
                .
                .
                .
ACB1            YACB        ISO=Y,APPNAME=ANAME,PRONAME=PNAME
ANAME           DC          CL8'APPL'
PNAME           DS          CL8
                .
                .
                .
```

3.1.1.2 Primary opening of a shareable DCAM application



1) Address of the control block that describes the DCAM application
2) constant may be defined as a character constant: C'constant' or a hexadecimal constant: X'constant'.

A DCAM application which is to be shareable and is not to use distribution codes is to be opened.

Mandatory specifications:

1) ISO=Y, to indicate that the ISO transport service is to be used. This specification is mandatory for DCAM(ISO) transport service applications. This function is not available for DCAM(NEA) transport service applications. If this specification is omitted, the NEA transport service is used as the default.
2) That the application is shareable. For DCAM(NEA) transport service applications, it is also necessary to specify that the application is not to use distribution codes.
3) The name of the DCAM application, which must be specified by the user. The address of an 8-byte memory area containing the name should be specified.
4) The password, if a predefined application is protected by a password in the RDF.

Optional specifications:

5) The task is to be a primary task (PRIMARY). If omitted, no check is to be made (NO).
6) DCAM notifications (see page 34) are to be processed. The address of the ENB should be specified for this purpose. If omitted, notifications are not processed.
7) The DCAM application is to be protected against unauthorized access by a secondary task by a password. The entry is only meaningful if the application has not been predefined and protected by a password in the RDF. If the entry is used anyway, the USEPASS password should be set to the same value as that in the RDF.
8) Connection requests are not to be processed (NLOGON). If this specification is omitted, connection requests are processed (NEA transport service only).
9) Transport acknowledgments are to be transferred to the requesting process (REQTASK) or not to be transferred (NOTACK). If this specification is omitted, transport acknowledgments are transferred to the primary task (PRIMTASK) (NEA transport service only).
10) A password is required for connection requests to this DCAM application, which is specified here. If this specification is omitted, a password is not required (NEA transport service only).

Further **optional** specifications:

11) The address of the field in which DCAM enters the symbolic name of its own processor.

12) The address of the field containing the link name.

13) That the information in the ACB is to be overwritten with the information from the CLT entry (CLT = communication link table) identified by the link name (PERM; permanent change). If this entry is omitted, the CLT entry information will be interpreted only for this call. The information in the ACB will not be changed (TEMP; temporary change).

14) The number of the DCAM version if the new functions introduced as of DCAM V8.0 are to be used in the task. 8.0 must be specified if the new functions are to be used. If this is not specified, or if the specification is ≠ 8.0, a DCAM version ≤ 7.0 is assumed.

Example

A DCAM(NEA) transport service application is to be opened for the first time and given the name APPL. It is to be shareable. A partner issuing a connection request is to specify the password 'WXYZ'. When a subsequent task opens the application, it is to specify the password 'ABCD'.

A check is to be made to ensure that this task is the first opening task.

```
                .
                .
                YOPEN       ACB=ACB1
                .
                .
ACB1            YACB        APPNAME=NAME,
                            ATTR=SHARE,
                            LOGPASS=C'WXYZ',
                            USEPASS=C'ABCD',
                            VERIFY=PRIMARY
NAME            DC          CL8'APPL'
```

Example

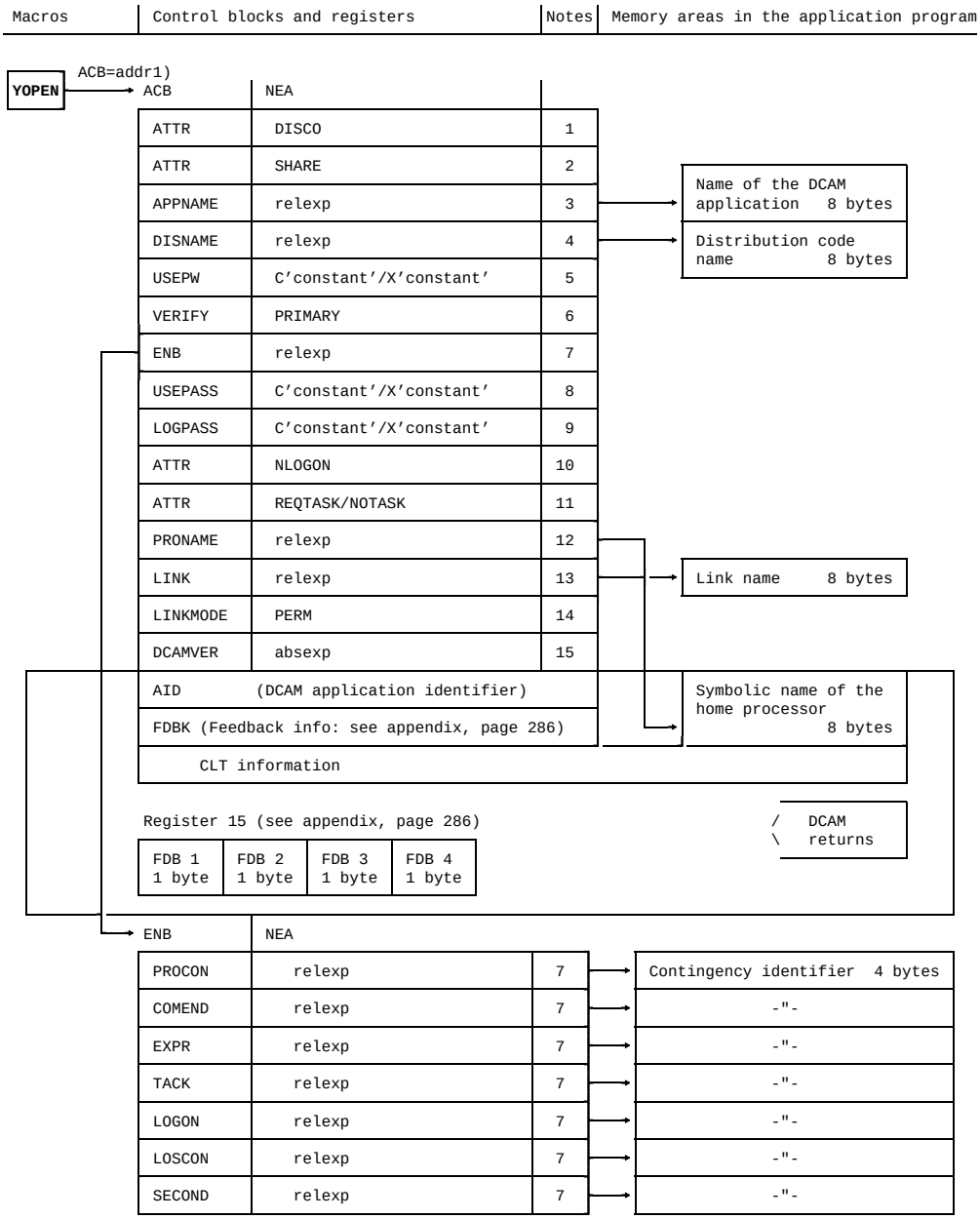
A DCAM(ISO) transport service application is to be opened for the first time and given the name APPL. It is to be shareable. When a subsequent task opens the application, it is to specify the password 'ABCD'.

A check is to be made to ensure that this task is the first opening task.

```
                .
                .
                YOPEN       ACB=ACB1
                .
                .
ACB1            YACB        ISO=Y,
                            APPNAME=NAME,
                            ATTR=SHARE,
                            USEPASS=C'ABCD',
                            VERIFY=PRIMARY
NAME            DC          CL8'APPL'
```

3.1.1.3 Primary opening - use of distribution codes

This section applies only to DCAM(NEA) transport service applications.



1) Address of the control block that describes the DCAM application

A DCAM application which uses distribution codes is to be opened for the first time.

Mandatory specifications:

1) Messages are to be distributed by means of a distribution code.
2) The DCAM application is to be shareable.
3) The user-defined DCAM application name, which is defined by the user. This must be placed in the 8-byte field addressed here.
4) The user-defined distribution code name, which is defined by the user. This must be placed in the 8-byte field addressed here.
5) The password, if a predefined application is protected by a password in the RDF.

Optional specifications:

6) The task is to be a primary task. If this specification is omitted, no check is to be made.
7) DCAM notifications (see page 34) are to be processed. The address of the ENB should be specified for this purpose. If it is omitted, DCAM notifications are not processed.
8) The DCAM application is to be protected by a password against unauthorized access by a secondary task. The entry is only meaningful if the application has not been predefined and protected by a password in the RDF. If the entry is used anyway, the USEPASS password should be set to the same value as that in the RDF.
9) A password is required for connection requests addressed to this DCAM application. If this specification is omitted, a password is not required.
10) Connection requests are not to be processed (NLOGON). If this specification is omitted, they are processed (LOGON).
11) Transport acknowledgments are to be transferred to the requesting process (REQTASK) or are not to be transferred (NOTACK). If this specification is omitted, transport acknowledgments are transferred to the primary task (PRIMTASK).

Further **optional** specifications:

12) The address of the field in which DCAM enters the symbolic name of its own processor.
13) The address of the field containing the link name.

14) That the information in the ACB is to be overwritten with the information from the CLT entry (CLT = communication link table) identified by the link name (PERM; permanent change). If this entry is omitted, the CLT entry information will be interpreted only for this call. The information in the ACB will not be changed (TEMP; temporary change).

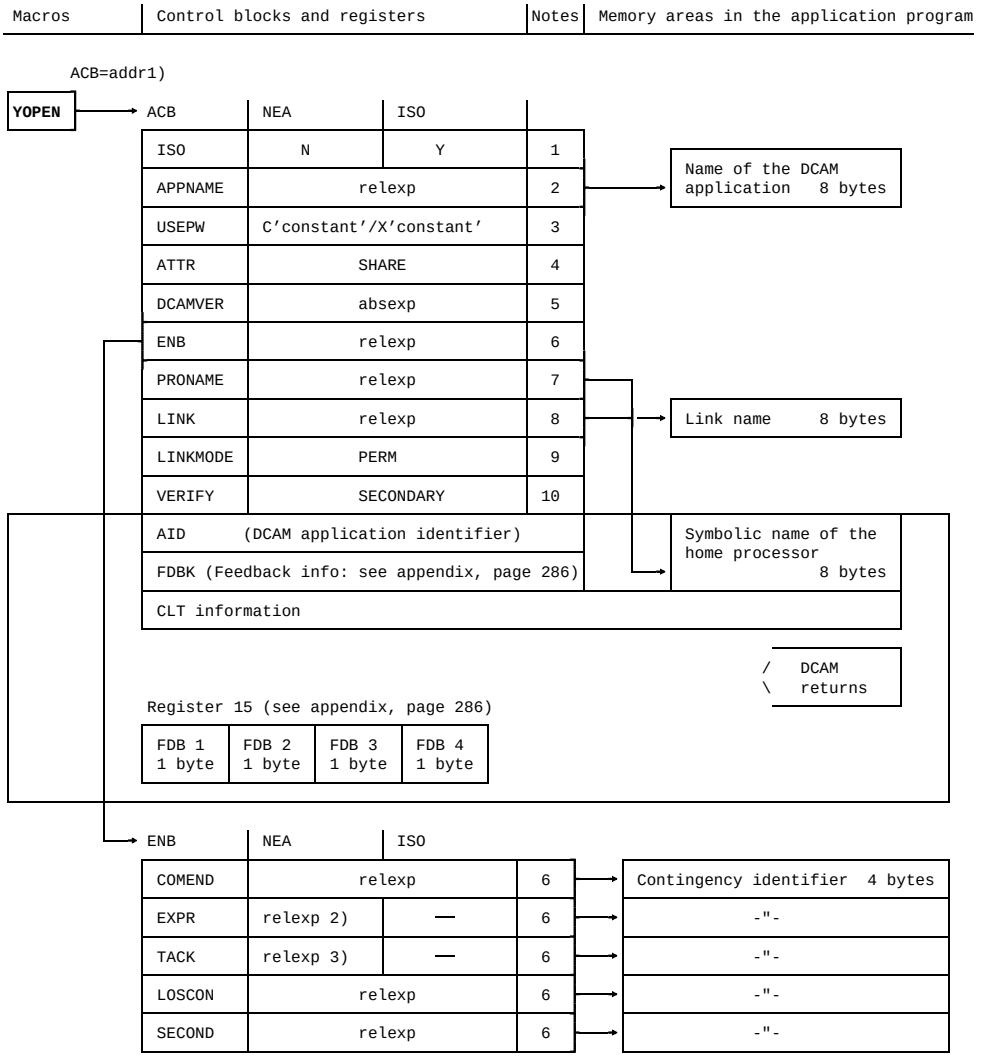
15) The number of the DCAM version if the new functions introduced as of DCAM V8.0 are to be used in the task. 8.0 must be specified if the new functions are to be used. If this is not specified, or if the specification is ≠ 8.0, a DCAM version ≤ 7.0 is assumed.

Example

A DCAM application using distribution codes is opened. A check is to be made to ensure that it is being opened for the first time. Secondary processes are to specify the password X'0000FFF0'.
Transport acknowledgments are to be transferred to the requesting task. The distribution code name for this task is to be "PRIMTSK". DCAM notifications are to be processed. Connections are to be established by the partners without the use of a password. The name of the application is to be "BOOKING".

```
Primary task:

               .
               .
               YOPEN       ACB=ACB1
               .
               .
ACB1           YACB        APPNAME=ADDR1,
                           VERIFY=PRIMARY,
                           ENB=ENBAD,
                           ATTR=(SHARE,DISCO,REQTASK),
                           DISNAME=DISAD,
                           USEPASS=X'0000FFF0'
ADDR1          DC          CL8'BOOKING'
ENBAD          YENB        ...
DISAD          DC          CL8'PRIMTSK'
```

3.1.1.4 Secondary opening



1) Address of the control block that describes the DCAM application
2) only in conjunction with EDIT = USER
3) only if the primary task specified ATTR = REQTASK

A DCAM application is to be opened by a secondary task. Messages are to be distributed by means of the common receiver queue or the originatororiented queue. The application must have been opened with the SHARE attribute in the primary task (see page 50).

Mandatory specifications:

1) ISO=Y, to indicate that the ISO transport service is to be used. If this is omitted, a DCAM(NEA) transport service application is used as the default.
2) The name of the DCAM application, which must be identical with that defined in the primary task. The name must be placed in the 8-byte field addressed here.
3) The password if the DCAM application is protected by a password (by specification of USEPASS in the primary task or by means of a password in the RDF).
4) The SHARE attribute, just as in the primary task.
5) The number of the DCAM version must be specified if this was done in the primary task. The DCAM version number specified in the primary and secondary tasks must be the same.

Optional specifications:

6) DCAM notifications are to be processed (see page 34). The address of an ENB control block is required. If this specification is omitted, DCAM notifications are not processed.
7) The address of the field in which DCAM enters the symbolic name of its own processor.
8) The address of the field containing the link name.
9) That the information in the ACB is to be overwritten with the information from the CLT entry (CLT = communication link table) identified by the link name (PERM; permanent change). If this entry is omitted, the CLT entry information will be interpreted only for this call. The information in the ACB will not be changed (TEMP; temporary change).
10) That the task is a secondary task (SECONDARY). If this specification is omitted, no check is made (NO), i.e. the task may also be a primary task.

Example

The DCAM(NEA) transport service application illustrated in the first example (see page 50) is to be opened by a secondary task.

Measures are to be taken to ensure that this task is a secondary task.

Secondary task:

```
                .
                .
                YOPEN       ACB=ACB1
                .
                .
ACB1            YACB        APPNAME=NAME,
                            ATTR=SHARE,
                            USEPW=C'ABCD',
                            VERIFY=SECONDARY
NAME            DC          CL8'APPL'
```

Example

The DCAM(ISO) transport service application illustrated in the first example (see page 50) is to be opened by a secondary task.

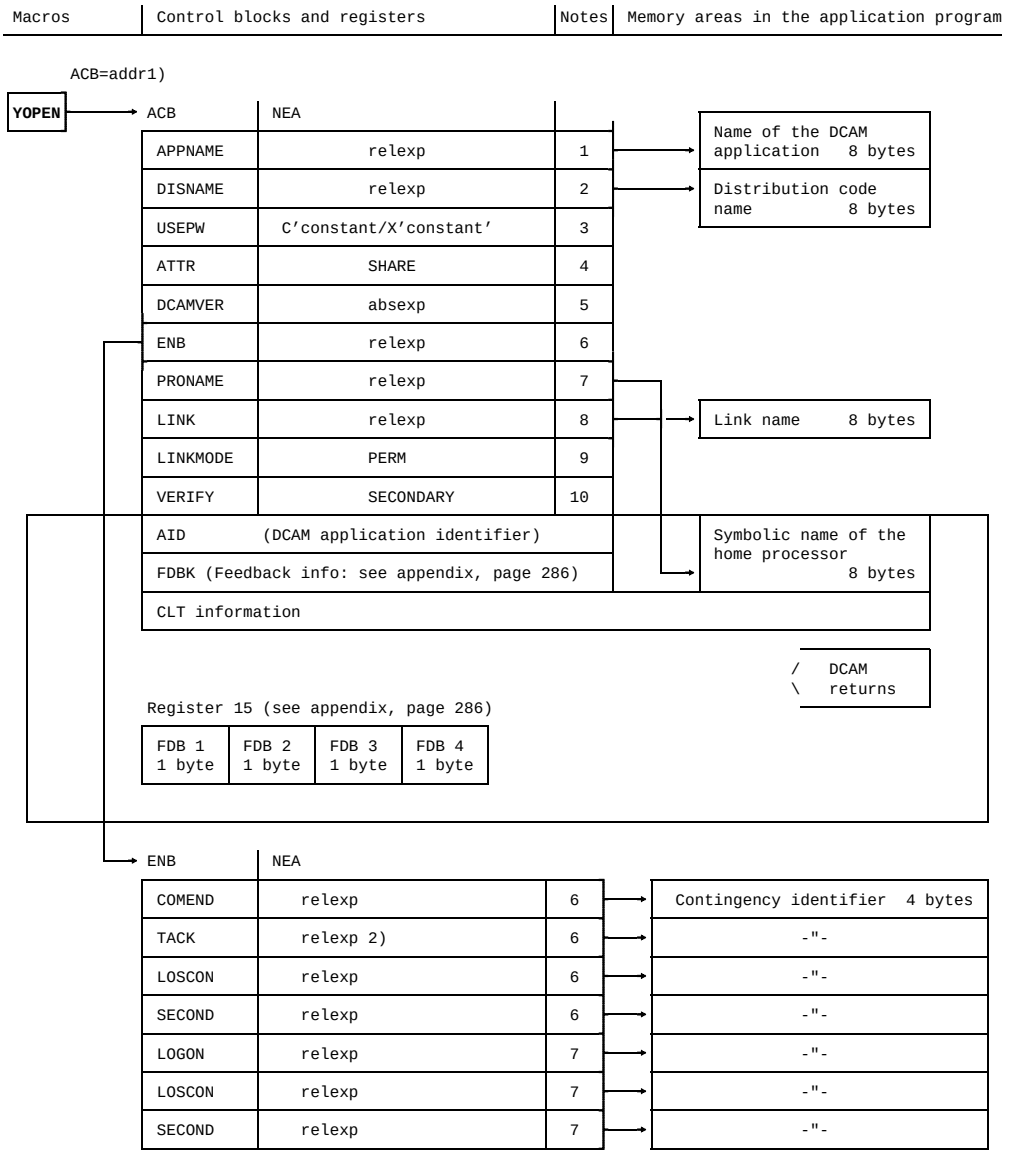
Measures are to be taken to ensure that this task is a secondary task.

Secondary task:

```
                .
                .
                YOPEN       ACB=ACB1
                .
                .
ACB1            YACB        ISO=Y,
                            APPNAME=NAME,
                            ATTR=SHARE,
                            USEPW=C'ABCD',
                            VERIFY=SECONDARY
NAME            DC          CL8'APPL'
```

3.1.1.5 Secondary opening - use of distribution codes

This section applies only to DCAM(NEA) transport service applications.



1) Address of the control block that describes the DCAM application
2) only if ATTR=REQTASK was specified in the primary task

A DCAM application using message distribution by means of distribution codes is to be opened as a secondary task. For this, the SHARE and DISCO attributes must have been specified in the primary task (see page 54).

Mandatory specifications:

1) The name of the DCAM application (which must be identical with the name specified in the primary task). This name must be placed in the 8-byte field addressed here.
2) The name for distribution code usage (distribution code name). This name must also be placed in the 8-byte field addressed here.
3) The password if the DCAM application is protected by a password (by specifying USEPASS in the primary task, or by means of a password in the RDF).
4) The SHARE attribute, just as in the primary task.
5) The number of the DCAM version if this was done in the primary task. The DCAM version number in the primary and secondary tasks must be the same.

Optional specifications:

6) DCAM notifications (see page 34) are to be processed. The ENB address must be specified for this purpose. If it is omitted, DCAM notifications are not processed.
7) The address of the field in which DCAM enters the symbolic name of its own processor.
8) The address of the field containing the link name.
9) That the information in the ACB is to be overwritten with the information from the CLT entry (CLT = communication link table) identified by the link name (PERM=permanent change). If no specification is made, the information from the CLT entry is interpreted for this call only. The ACB information will not be changed (TEMP=temporary change).
10) The task is to be a secondary task. If this specification is omitted, no check is made, i.e. the task may also be a primary task.

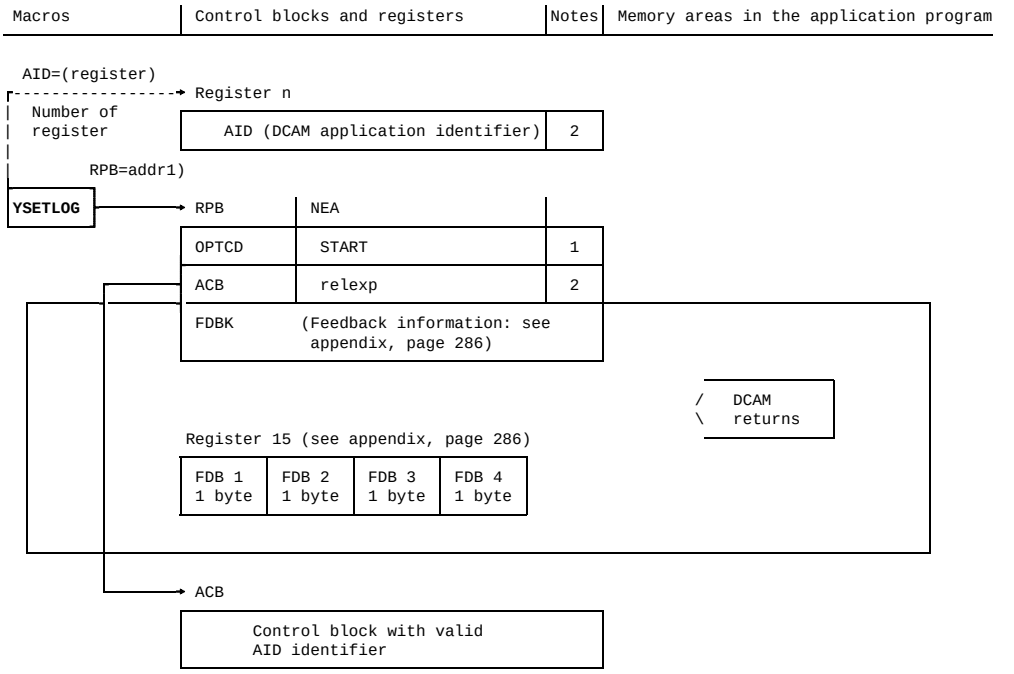
Example

A secondary task is to be linked to the application opened in an earlier example (see page 54) by the primary task. A check is to be made to ensure that this actually is a secondary opening. Asynchronous notifications are to be processed. The distribution code name for this example is "BOOK1".

```
Secondary task:
                .
                .
                .
                .
                YOPEN       ACB=ACB1
                .
                .
                .
ACB1            YACB        APPNAME=ADDR1,
                            VERIFY=SECONDARY,
                            ENB=ENBAD,
                            ATTR=SHARE,
                            DISNAME=DISAD,
                            USEPW=X'0000FFF0',
ADDR1           DC          CL8'BOOKING'
ENBAD           YENB        ...
DISAD           DC          CL8'BOOK1'
```

3.1.2 Changing the state of a DCAM application

This section applies only to DCAM(NEA) transport service applications.



1) Address of the control block that describes the DCAM application

The state of a DCAM application is to be changed. A prerequisite for this is that the DCAM application accepts connection requests, i.e. that the LOGON attribute was set in YOPEN (see pages 47 through 54).

Mandatory specifications:

1) In YSETLOG, the START state (requests are accepted) or the STOP state (requests are not accepted). The START state prevails before the first YSETLOG. If no specification is made here, the STOP state is set.

2) Either the address of the ACB control block containing the valid identifier AID or a register containing the valid identifier.

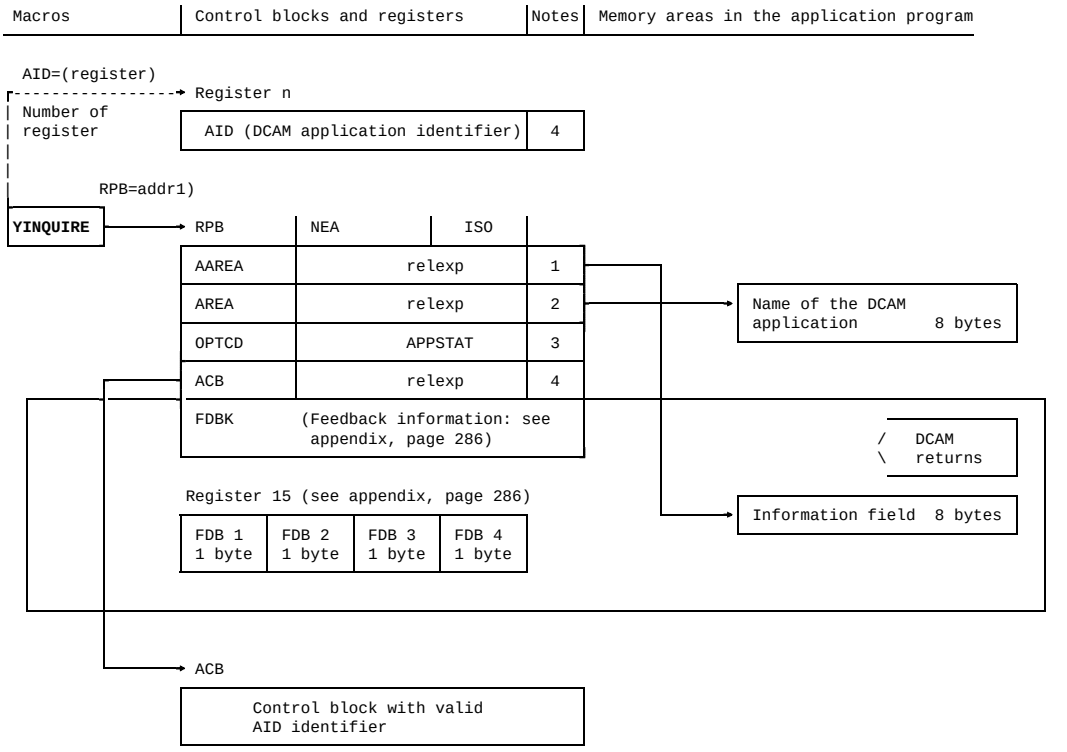
   This entry can be omitted if the identifier was previously entered in the RPB (e.g. with YMODCB).

Example

The primary task that opened the DCAM application APPL is to stop processing connection requests for a given period of time and to resume connection request processing after that time.

```
                                    Meaning:
       .
       .
       .
       YOPEN       ACB=ACBNAM
       .
       .
       .
       YSETLOG     RPB=RPBNAM      The state is set
       .                           to STOP
       .
       .
       YSETLOG     RPB=RPBNAM,     The state is set to
                   OPTCD=START     START (the RPB
                                   operand is set
                                   again).
       .
       .
       .
ACBNAM YACB        APPNAME=NAME,
                   ATTR=LOGON
RPBNAM YRPB        ACB=ACBNAM
NAME   DC          CL8'APPL'
```

3.1.3 Querying the state of a DCAM application



1) Address of the control block that describes the DCAM application

The state of a DCAM application is to be queried.

Mandatory specifications:

1) An 8-byte field in which the information is to be entered.

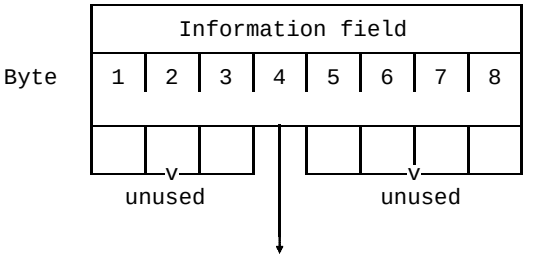
2) The name of the DCAM application to be queried. This must be an application in the same host computer. The application addressed in 4) can be specified.

3) The YINQUIRE version used here.

4) The address of the ACB control block containing the valid identifier (AID) or a register containing the valid identifier.

   This specification may be omitted if the identifier was previously entered in the RPB (e.g. with YMODCB).

The information comprises:



X’00’: The application is open and accepts connection requests (ATTR=LOGON; START state).

X’04’: The application is open and is not processing connection requests at present (ATTR=LOGON; STOP state) (NEA transport service only).

X’08’: The application is open and does not accept connection requests (ATTR=NLOGON) (NEA transport service only)

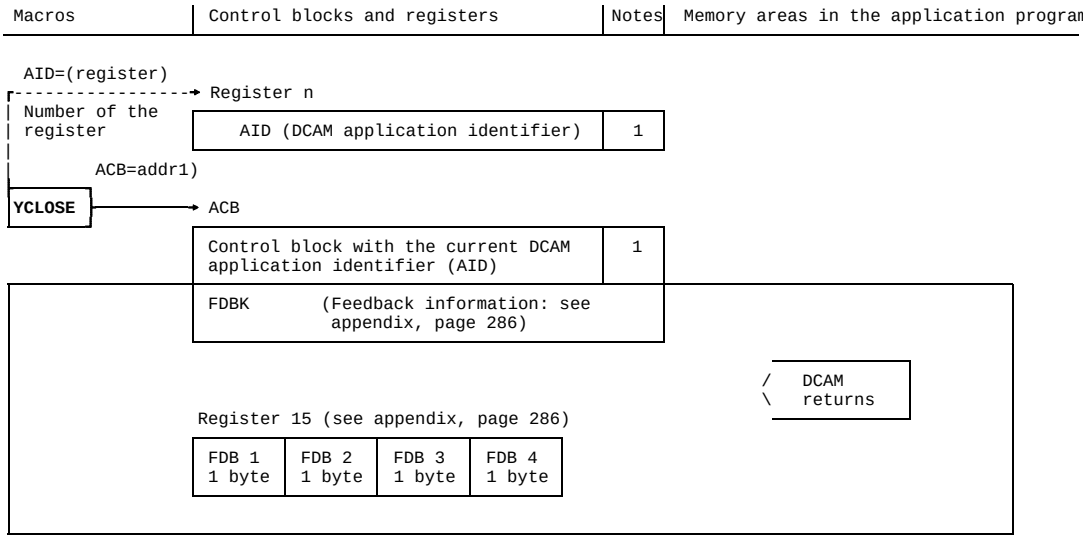
X’0C’: The application has not yet been created (opened by a primary task), i.e. it does not exist.

Example

The primary task of the DCAM application, whose identifier AID is contained in register 9, wants information on the state of the DCAM application with the symbolic name ’APPL’.

```
                .
                .
                .
                YINQUIRE    RPB=RPB2,AID=(9)
                .
                .
                .
RPB2            YRPB        OPTCD=APPSTAT,
                            AREA=APPNAME,
                            AAREA=INFFELD
APPNAME         DC          CL8’APPL’
INFFELD         DS          CL8
```

3.1.4 Closing a DCAM application



1) Address of the control block that describes the DCAM application

YCLOSE from a **secondary task**:

– The DCAM application is closed exclusively for this task. If a SECOND contingency routine has been defined (see page 41), this will be initiated.

YCLOSE from a **primary task**:

– The DCAM application is removed, i.e. it is closed for this task and for all the secondary tasks belonging to this application.

– All current connections are cleared.

– Pending requests for DCAM functions are reset.

– Received data that has not yet been transferred to the task is lost.

– All currently queued connection requests are rejected by DCAM.

The following applies to the **secondary tasks affected** by this YCLOSE:

– If the secondary task has defined a COMEND contingency routine, it is notified of the closing of the DCAM application by a branch to this routine.

– If no COMEND contingency routine is defined, the task is notified of application closure by the feedback information 'DCAM application not opened' (see appendix, page 286) when the next macro referring to this DCAM application is issued.

Mandatory specification:

1) Either the address of the ACB control block containing the valid identifier AID or a register containing the valid identifier.

Example

The DCAM application opened under the name 'APPL' is to be closed.

```
                .
                .
                .
                YOPEN       ACB=ACBNAM
                .
                .
                .
                YCLOSE      ACB=ACBNAM
ACBNAM          YACB        ISO=Y,APPNAME=NAME
NAME            DC          CL8'APPL'
```

3.2 Connection-oriented macros

Before communication partners can transfer messages, they must establish a connection. One of the partners requests connection. This connection request is transmitted to the other partner through the communication system. Information on the requesting partner can then be obtained, and the connection request accepted or rejected. Specific characteristics of the connection can be modified while the connection exists, and the connection can be cleared at any time by either partner.

The connection-oriented macros are:

- **YOPNCON** **establish a transport connection (request/acceptance)**
- **YINQUIRE** **request information on the connection and the communication partners**
- **YREJLOG** **reject a connection request**
- **YCHANGE** **modify the connection characteristics**
- **YCLSCON** **cancel a request/close a transport connection**

Information on dynamic name assignment (LINK) is provided in a separate section (page 139).

DCAM processes can process the following asynchronous notifications with appropriate contingency routines (see pages 29 and 34):

- **LOGON** **transport connection request or successful connection setup**
- **LOSCON** **lost transport connction**
- **PROCON** **proposed communication partner**
- **Successful transport connection setup**

3.2.1 Establishing a connection

In order to establish a connection:

- a request can be issued or
- a request can be accepted.

In addition, a link to a predefined connection can be established.

When a connection is established, either the type of queue (originator-oriented or common receiver) or, for DCAM(NEA) transport service applications, the assignment of the distribution codes to the connection is defined.

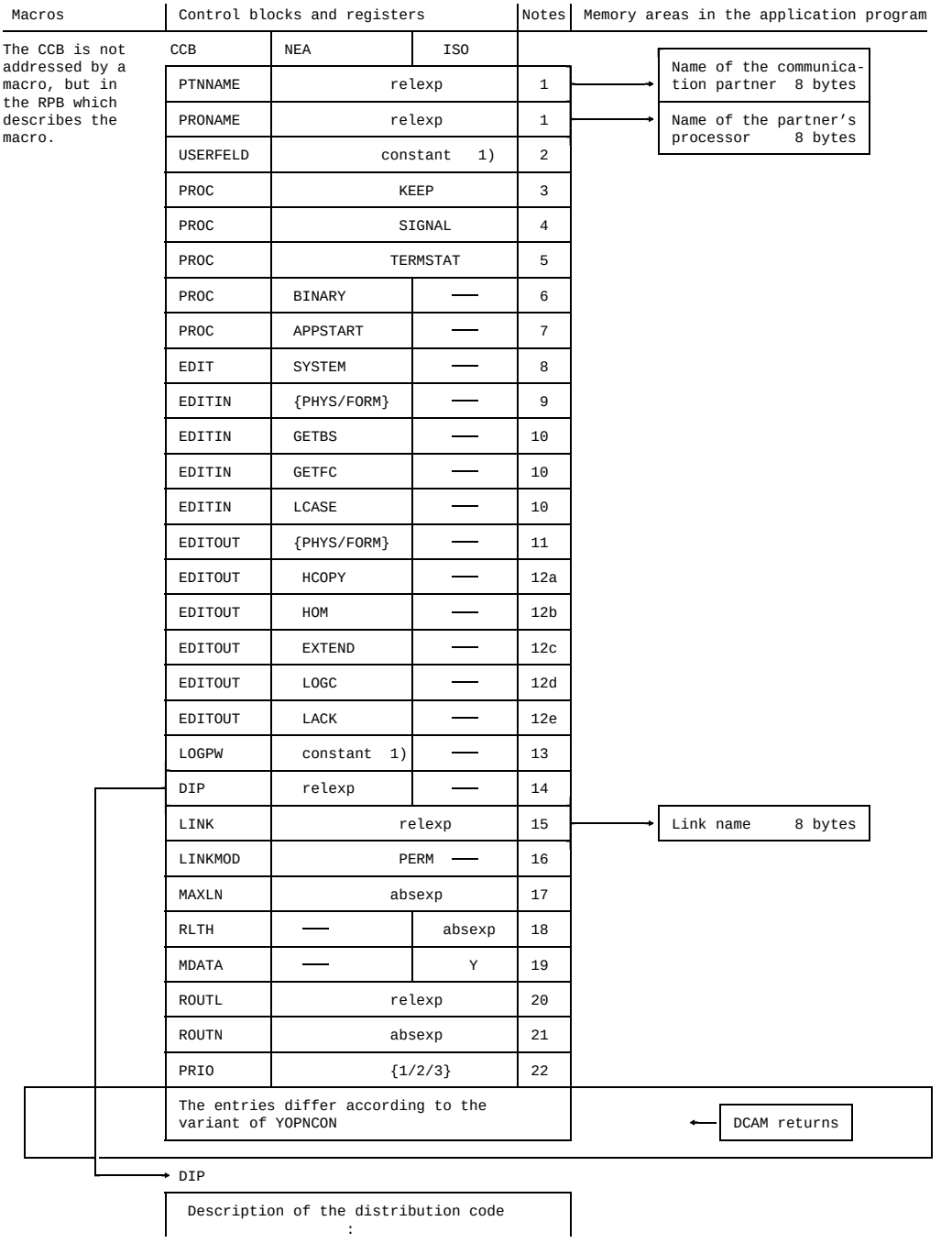
As a result, four variants of the YOPNCON macro are available. The differences arising from the synchronous or asynchronous processing of a macro are shown here not as separate variants (see page 80) but - in so far as they concern DCAM - within the three variants.

The connection to be established is described in a separate control block, the CCB. This is addressed in the RPB via YOPNCON.

Note

The reference address in the RPB to the CCB used is destroyed by YOPNCON. This is done to prevent an inadvertent attempt to open the connection twice. This can also be avoided by using YOPNCON explicitly with the keyword CCB=ccbnam.

3.2.1.1 Description of the connection to be established



```
1) constant may be specified as a character constant: C'constant', a
   hexadecimal constant: X'constant', a fixed-point constant: F'constant'
   or an address constant: A'constant', V(constant).
```

To allow a connection to be set up, the CCB control block must contain the values describing the connection. It is generated with YCCB or YGENCB and **may** contain the following entries:

1) The address of an 8-byte field containing the **name of the communication partner**, and the address of an 8-byte field containing the **processor name** of the partner. DCAM supplies these specifications if a YOPNCON with OPTCD=(ACCEPT,ANY) is used to specify that a request issued by any partner is to be accepted (see page 80).

2) The **accompanying information** (4 bytes) which is later to be transferred by DCAM to the program (upon YRECEIVE ANY, for LOSCON notifications and, for DCAM(NEA) transport service applications, for TACK or EXPR notifications).

Specifications for **message handling**:

3) Storage of the remaining portions of overlength messages (KEEP) pending another YRECEIVE.

   If this specification is omitted, the remaining portion of the message is discarded (TRUNC).

   This specification becomes effective only if requested in the YRECEIVE macro (OPTCD=CCBTK); otherwise, the specifications made there are used.

4) If the connection becomes overloaded during transmission of messages, the user should be requested by a GO signal to resume transmission (PROC=SIGNAL) after the backlog has been cleared.

   If the specification is omitted, no GO signal is issued.

5) The completeness of the status information from the 9763 Data Display Terminal is to be checked.
   If the status information is complete, feedback field FDBK 3 is set to X'80'; if it is incomplete, to X'40'. In neither case does the user receive a message.

6) Messages are to be transferred in the communication system as bit patterns without code conversion (BINARY). This requires that EDIT=USER was set (see below) (NEA transport service only).

   If this specification is omitted, DCAM expects and transfers the messages in the code used by its host computer (SYSCODE) (NEA transport service only).

7) Data transmission is to be initiated by this application (APPSTART). If omitted, no specification is made (ANYSTART) (NEA transport service only).

8) Virtual terminals are to be used for transmitting and receiving messages (EDIT=SYSTEM). If the partner is a DCAM application, this specification may be omitted (EDIT=USER) (NEA transport service only).

If **EDIT=SYSTEM** was set, the following may be specified:

9) The type of virtual terminal for message reception (NEA transport service only):

   EDITIN=

   – PHYS: The user receives the message in the form generated by the terminal. Message headers are received in the device code. Reassembly of blocked messages is assisted by the system and lowercase letters are transmitted if they are used.

   – FORM: Message format control is used.

   If this specification is omitted, EDITIN=LINE is assumed, i.e. virtual line terminals are used.

10) The handling of incoming messages (NEA transport service only):

   With LINE/FORM: Backspace characters are not to be interpreted, but are transferred to the user (GETBS).

   With LINE:

   – the logical function key code of the terminal is the first character to be transferred (GETFC).

   – lowercase characters are to be transferred (LCASE).

   If these specifications are omitted, backspace characters are analyzed and not transferred (NGETBS), the logical function key code is not transferred (NGETFC), and any lowercase letters sent by the partner are converted into upper case (NLCASE).

11) The type of virtual terminal to be used for message transmission (NEA transport service only):

   EDITOUT=

   – PHYS: The user transfers the message inclusive of the header length byte in a form acceptable to the terminal with the message header in device code.

   – FORM: Message format handling is used.

   If this entry is omitted, EDITOUT=LINE is assumed, i.e. line terminals are used.

12) The handling of the messages transferred to the system for transmission (NEA transport service only):

a) LINE or PHYS (8151, 8152, 816x, 975x, 976x, 3270 display terminals): The message is also to be output on the printer connected to the data display terminal (HCOPY). If this specification is omitted, the message is not printed (NHCOPY). A test has to be made to see whether such a printer is connected, if necessary by checking the partner characteristics (see page 95).

b) LINE: The message is to be printed unstructured (HOM). If this specification is omitted, the message is output in structured form, i.e. each logical line is treated as a unit (NHOM).

c) LINE (975x, 976x and 816x Data Display Terminals): By default, the entire screen is protected by the system (EXTEND).

d) No logical control characters of a message are evaluated (NLOGC). All characters < X'40' are replaced by SUB (smudge) (e.g. output of printer messages on terminal).

e) LINE:
Logical acknowledgments are requested from the printer terminal (LACK).

13) The password for establishing a connection (see YOPNCON ACQUIRE, page 76) (NEA transport service only).

14) The description of the distribution codes (see the separate section, page 84) (NEA transport service only).

15) The address of the field containing the link name.

16) The information in the CCB is to be overwritten with the information from the CLT entry (CLT = communication link table) identified by the link name (PERM; permanent change). If no specification is made, the CLT entry information will be interpreted only for this call. The information in the CCB will not be changed (TEMP; temporary change).

17) Maximum length of the messages to be transmitted by the DCAM application on this link. DCAM returns the actual available message length, which may be less than the value specified in MAXLN. The message length is not negotiated with the communication partner, nor is it passed on to the communication partner (see also "DCAM Program Interfaces").

18) The length of the expected receive messages for DCAM(ISO) transport service applications, but only if the more-data function is not used on this connection (MDATA=N). This value is used for optimizing space requirements and improving the system performance. It has only local significance and is neither passed on to nor negotiated with the communication partner. Specifying RLTH thus provides no guarantee that longer messages will not be received (see also "DCAM Program Interfaces").

19) The more-data function is to be used on this connection for DCAM(ISO) applications (MDATA=Y). If this is not specified, the more-data function is not used (MDATA=N) (see also "DCAM Program Interfaces").

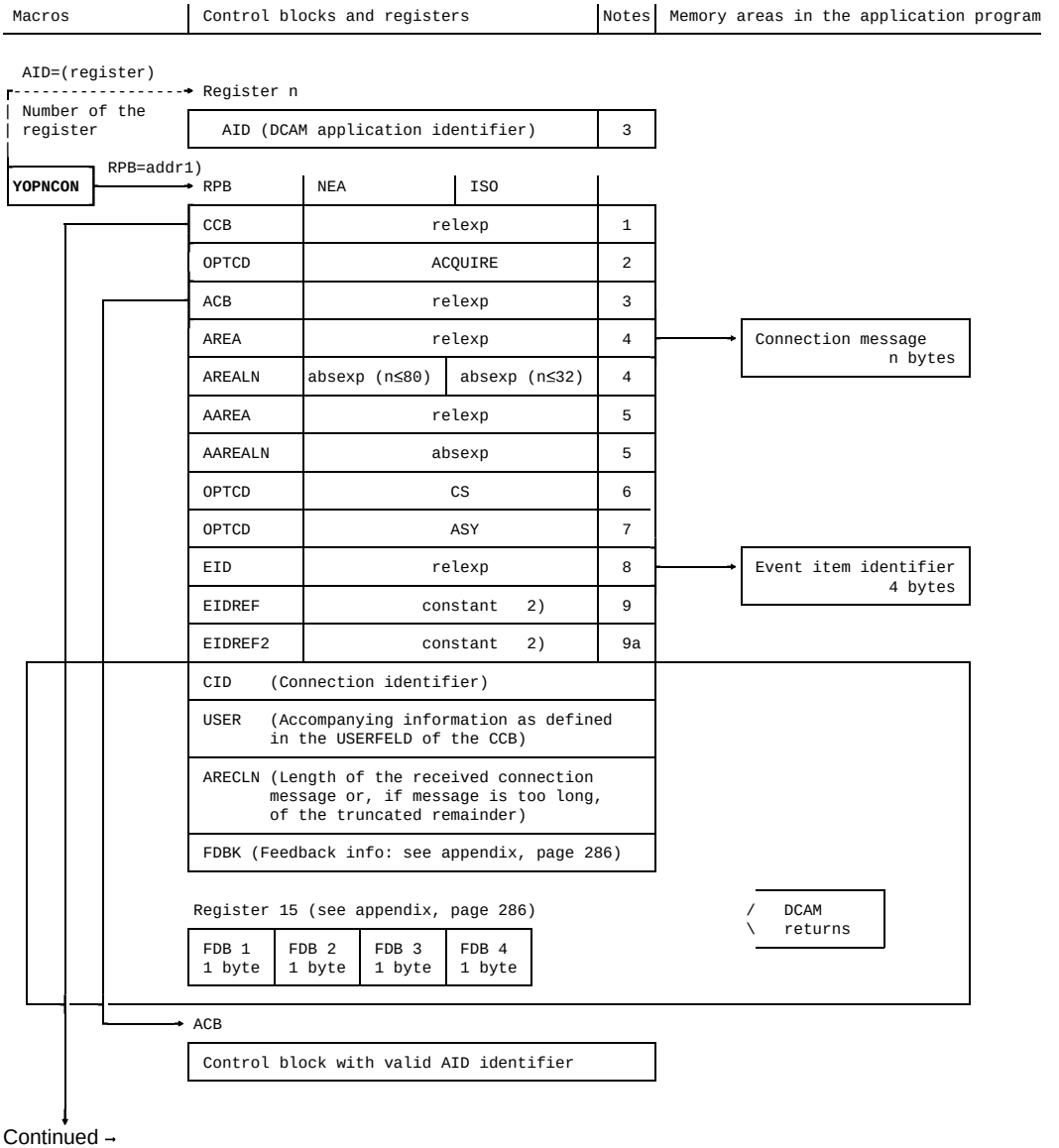
20) The address of a list containing the names of various routes to the partner. This is evaluated in YOPNCON ACQUIRE (page 76).

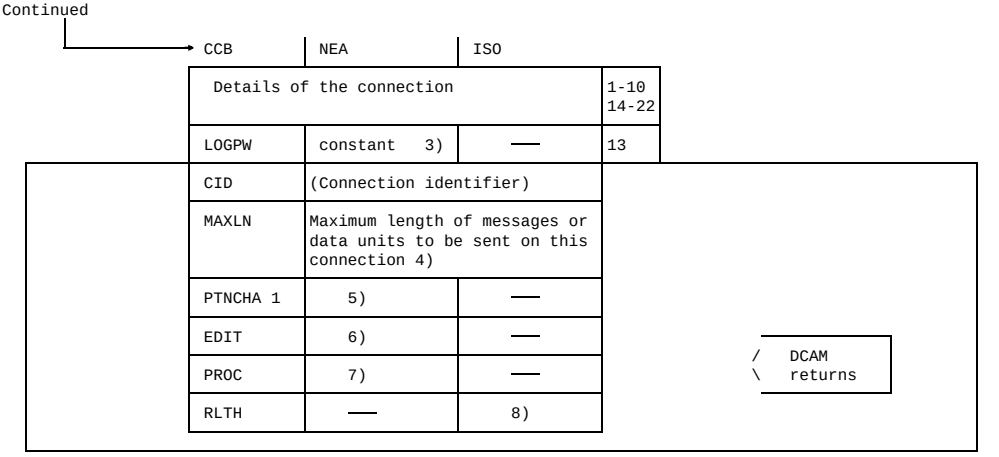
21) The number of routes to the specific partner. Up to 16 routes are possible. These are evaluated in YOPNCON ACQUIRE (page 76).

22) The definition of the priority of a connection. Highest priority PRIO=1 for system applications only) Lowest priority PRIO=3 (default).

The DCAM entries in the CCB depend on the YOPNCON variant and are therefore described in the relevant sections (see below).

3.2.1.2 Connection request





1) Address of the control block that describes the DCAM application
2) constant may be specified as a character constant: C'constant', a hexadecimal constant: X'constant', a fixed-point constant: F'constant' or an address constant': A'constant', V(constant).
3) constant may be specified as a character constant: C'constant' or a hexadecimal constant: X'constant'.
4) Dependent on the maximum packet length for a transfer to a connected computer or on a system constant for a transfer within the processing computer
5) The first four bytes of the partner's characteristics
6) The use of virtual terminals accepted by the partner; contains USER or SYSTEM
7) APPSTART/ANYSTART, as proposed by the partner
8) Maximum length of the received messages (only if MDATA = Y)

Establishment of a connection is to be requested. With this variant for DCAM(NEA) transport service applications (no use of distribution codes), the attributes NSHARE (see page 47) or SHARE with NDISCO (see page 50) must have been set when the DCAM application was opened.

Mandatory specifications:

1) The address of the CCB control block describing the connection to be established;
2) That a connection request is to be issued.

Optional specifications:

3) Either the address of the ACB control block containing the valid application identifier (AID) or a register containing the valid identifier.

   This specification may be omitted if the identifier was previously entered in the RPB (e.g. with YMODCB).

4) The address of an area containing the connection message to be transferred with the request and the length of the message (up to 80 bytes for DCAM(NEA) transport service applications or 32 bytes for DCAM(ISO) transport service applications). Longer messages will be truncated without prior warning. Note that connection messages or connection passwords should not be sent when IBM terminals are being used, since these are not converted by TRANSIT-CD.

5) The address of an area in which the connection message of the communication partner is to be stored by DCAM, together with the length of this message (up to 80 bytes for DCAM(NEA) transport service applications or 32 bytes for DCAM(ISO) transport service applications).

   This specification is evaluated only by DCAM Version 8.0 or higher (i.e. DCAMVER=8.0).

6) That message are to be distributed via the originator-oriented queue (CS). If this specification is omitted, the common receiver queue (CA) is used. This specification can be modified again at data transmission time.

7) That the instruction is to be processed asynchronously (ASY). If this specification is omitted, synchronous processing (SYN) is set.

8) The address of a 4-byte field containing - only if the instruction is to be processed asynchronously - the event item identifier, in which case the specification is mandatory (see also page 29).

9) The first word of the event information to be transferred when asynchronous processing of the instruction is completed. The first byte is overwritten with X'0C' when the information is returned.

   If this specification is omitted, the default value X'00000000' is passed.

9a)

   The second word of the event information to be transferred when asynchronous processing of the instruction is completed.

   If this specification is omitted or equal to the default value X'00000000', only the first word of the event information is transferred.

The connection has to be described in the CCB control block (see page 71); the following must be specified, if applicable:

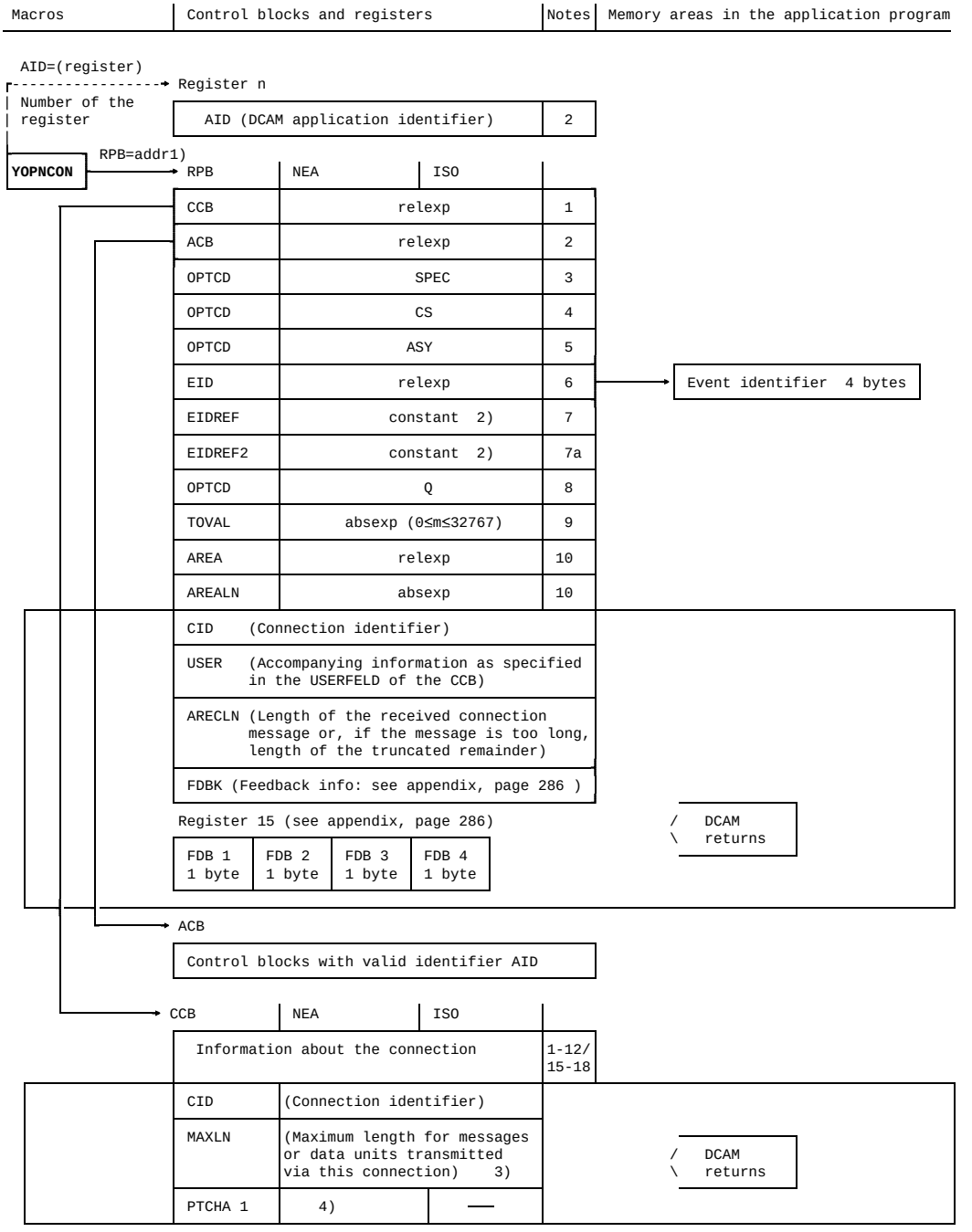
13) The connection request password (LOGPW) specified by the partner if the partner is a DCAM application (NEA transport service only).

Example

A connection is to be established to partner PTN1 attached to processor PROZ. The instruction is to be processed asynchronously. The user wants to be informed in a contingency routine when processing of the instruction has been completed. As a number of asynchronous instructions have already been issued, the user would like to be able to tell which instruction was terminated. He or she stores a private code (e.g. the string 'PT1') for this purpose in the EIDREF field of the RPB control block.

```
                .
                .
                ENAEI       EINAME=EVENT,
                            EIIDRET=EIDAD
                ENACO       CONAME=CONTGY,
                            COADAD=COAD,
                            COIDRET=COIDAD
                .
                .
                .
                YOPNCON     RPB=RPBNAM
                .
                .
                .
                SOLSIG      EIID=EIDAD,
                            COID=COIDAD
                .
                .
                .
RPBNAM          YRPB        ACB=ACBNAM,
                            CCB=CCBNAM,
                            OPTCD=(ACQUIRE,ASY),
                            EID=EIDAD,
                            EIDREF=C' PT1'
ACBNAM          YACB        ...
CCBNAM          YCCB        PTNNAME=NAM1,
                            PRONAME=NAM2
                            MDATA=N          (iso)
                            RLTH=N           (iso)
EIDAD           DS          F
COIDAD          DS          F
NAM1            DC          CL8'PTN1'
NAM2            DC          CL8'PROZ'
COAD            DS          A(CONTAD)
                .
                .
                .
CONTAD          .
                .
                .
                .
                RETCO
```

3.2.1.3 Request Acceptance



```
1) Address of the control block that describes the macro
2) constant may be specified as a character constant: C'constant', a hexadecimal constant: X'constant',
   a fixed-point constant: F'constant' or an address constant': A'constant', V(constant).
3) Dependent on the maximum packet length for a transfer to a connected computer or on a system constant
   for a transfer within the processing computer
4) The first four bytes of the partner characteristics (see the chapter 'Status information' in the
   'VTSU User Guide'.
```

A connection request is to be accepted.
For DCAM(NEA) transport service applications, acceptance of a connection request is required only if the application can process requests (LOGON attribute and START state). The YOPNCON version shown in this section also requires that the application was opened either with the NSHARE attribute (see page 47) or with the attributes SHARE, NDISCO (see page 50).

Mandatory specifications:

1) The address of the CCB control block describing the connection to be established (see page 71).
2) Either the address of the ACB control block that contains the valid application identifier (AID) or a register which contains the valid AID.

   This may be omitted if the identifier has already been placed in the RPB (e.g. with YMODCB).

Optional specifications:

3) Whether requests from a specific partner (SPEC) listed in the CCB are to be accepted. If this specification is omitted, any request is accepted (ANY).
4) Message distribution via the originator-oriented queue (CS). If this specification is omitted, the common receiver queue (CA) is used. This can be modified at data transmission time.
5) Asynchronous processing of the instruction (ASY). If this specification is omitted, synchronous processing (SYN) is set.
6) The address of a 4-byte field containing the event item identifier (only if the instruction is to be processed asynchronously, in which case the specification is required; see page 29).
7) The **first** word of the event information to be transferred when asynchronous processing of the instruction is completed. The first byte is overwritten with X'0C' during transfer.
   If this specification is omitted, the default value X'00000000' is passed.

   The **second** word of the event information to be transferred when asynchronous processing of the instruction is completed.

   If this specification is omitted or is equal to the default value X'00000000', only the first word of the event information is passed.

8) Whether the macro is to wait if no request is pending. If several macros are issued for a specific partner (SPEC) or for any partners (ANY), the macros are placed in a queue. This specification is mandatory for asynchronous execution (Q).

   If this specification is omitted, the macro is terminated immediately (NQ) even if no request is pending.

9) If the above specification was made, a maximum wait time can be defined. A zero means indefinite wait time. If this specification is omitted, a wait time of 600 seconds is set automatically.

10) Address of an area in which the connection message to be transferred is located, together with the length of this message (up to 80 bytes for DCAM(NEA) transport service applications or 32 bytes for DCAM(ISO) transport service applications). Longer connection messages will be truncated without prior warning.

    This specification is evaluated only in DCAM Version 8.0 or later (DCAMVER parameter for the ACB).

Example 1

A connection request from the partner with the name PTN1 connected to the processor with the name PROC is to be accepted.

```
                .
                .
                .
                YOPNCON     RPB=RPBNAM
                .
                .
                .
RPBNAM          YRPB        ACB=ACBNAM,
                            CCB=CCBNAM,
                            OPTCD=(SPEC,Q)
ACBNAM          YACB        ...
CCBNAM          YCCB        PTNNAME=PTNAD,
                            PRONAME=PROAD
                            MDATA=Y             (iso)
PTNAD           DC          CL8'PTN1'
PROAD           DC          CL8'PROC'
```

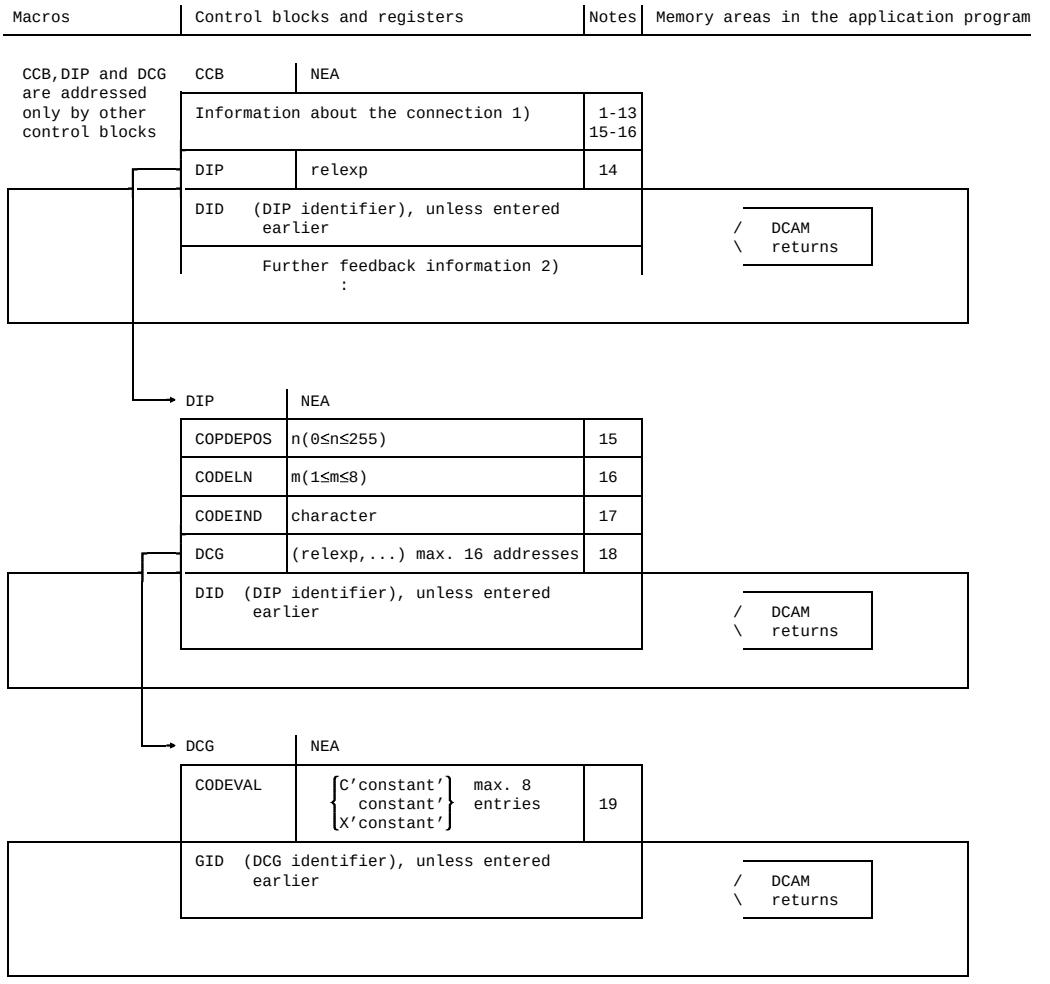
Example 2

A connection request from any partner is to be accepted asynchronously.

```
                .
                .
                .
                ENAEI       EINAME=EVENT,
                            EIIDRET=IDAD
                .
                .
                .
                YOPNCON     RPB=RPBNAM
                .
                .
                .
                SOLSIG      EIID=IDAD,
                            COND=IMMED
                .
                .
                .
RPBNAM          YRPB        ACB=ACBNAM,
                            CCB=CCBNAM,
                            OPTCD=(ASY,Q),
                            EID=IDAD
ACBNAM          YACB        ...
CCBNAM          YCCB        ...
IDAD            DS          F
PTNAD           DS          CL8
PROAD           DS          CL8
```

3.2.1.4 Connection setup - use of distribution codes

This section applies only to DCAM(NEA) transport service applications.



1) see page 71
2) see page 76

A connection which uses distribution codes is to be set up. The DCAM application must have been opened with the SHARE and DISCO attributes (see page 54 and the chapter "Macro catalog"). Only the differences with respect to the preceding sections (see page 76 and the chapter "Using the DCAM functions") are shown here; the OPTCD=CS specification is not required for either variant (ACCEPT or ACQUIRE).

Optional specification in the CCB:

14) The address of the DIP control block describing the distribution code operands. This specification is not required if a valid identifier DID is already contained in the CCB.

Mandatory specifications in the DIP control block:

15) The location of the distribution code in the message and

16) the length of the code(s) used.

18) The address of one or more (up to 16) control blocks DCG describing the distribution code(s) permitted on this connection.

Optional specifications:

17) The character indicating whether an implicit distribution code is used.

Optional specification in a DCG control block:

19) The distribution code(s) used (up to 8). This specification is not used if a valid identifier GID was already entered previously.

Example

The name of the DCAM application is ’APPL’.

A connection is to be established to a partner with the name ’PARTNER’ attached to a processor named ’PROZ1’.

The connection is to have the following characteristics:

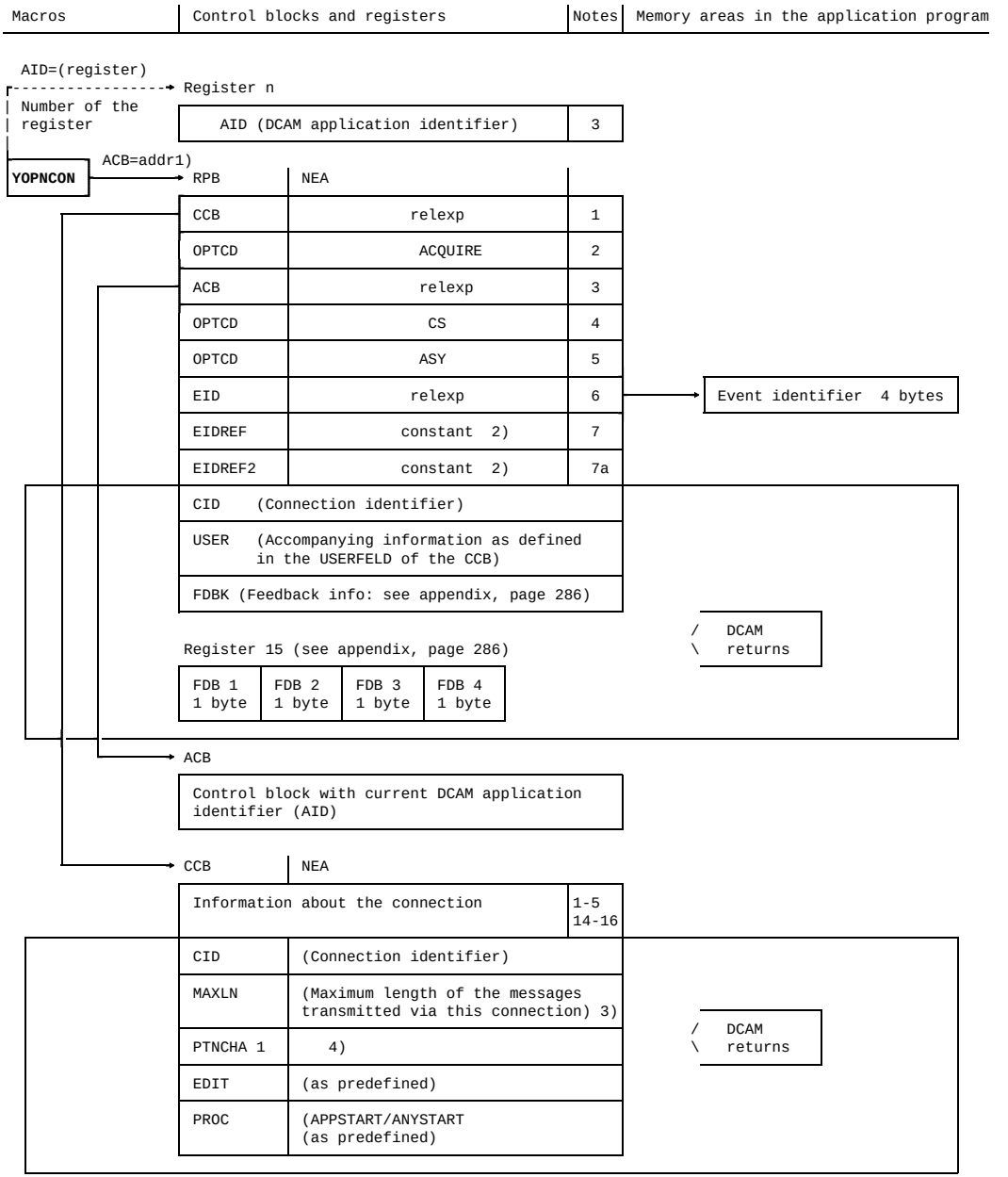
– Messages transferred over this connection are to be distributed on the basis of distribution codes.

– The code consists of one character (J, K or L) and is located at the third position in the message. The message text follows, starting at the fourth character.

```
                .
                .
                .
                YOPNCON     RPB=RPBNAM
                .
                .
                .
RPBNAM          YRPB        ACB=ACBNAM,
                            CCB=CCBNAM,
                            OPTCD=ACQUIRE
CCBNAM          YCCB        PTNNAME=PTNAD,
                            PRONAME=PROAD,
                            DIP=DIPNAM
ACBNAM          YACB        APPNAME=APPAD,
                            DISNAME=DISAD,
                            ATTR=(SHARE,DISCO)
DIPNAME         YDIP        CODEPOS=2,
                            CODELN=1,
                            DCG=DCGAD
DCGAD           YDCG        CODEVAL=(C’J’,C’K’,C’L’)
PTNAD           DC          CL8’PARTNER’
PROAD           DC          CL8’PROZ1’
DISAD           DC          CL8’PRIM’
APPAD           DC          CL8’APPL’
```

3.2.1.5 Linking to a predefined connection

This section applies only to DCAM(NEA) transport service applications.



```
1) Address of the control block that describes the macro
2) constant may be specified as a character constant: C'constant', a hexadecimal constant: X'constant',
   a fixed-point constant: F'constant' or an address constant': A'constant', V(constant).
3) dependent on the maximum packet length for transfer to a connected computer or on a system constant
   for transfer within the processing computer
4) The first four bytes of the partner characteristics
```

The DCAM application is to link itself to a predefined connection. This call is a variant of the calls described in the sections "Connection request", page 76 (without distribution code) and "Connection setup", page 84 (with distribution code).

Mandatory specifications:

1) The address of the CCB control block describing the connection to be established;
2) OPTCD=ACQUIRE to enable the link to the predefined connection to be established.
3) Either the address of the ACB control block containing the valid application identifier (AID) or a register containing the valid identifier.

   This specification may be omitted if the identifier was previously entered in the RPB (e.g. with YMODCB).

Optional specifications:

5) Message distribution through the originator-oriented queue (CS). If this specification is omitted, the common receiver queue (CA) is used. This specification can be modified again at data transmission time.
6) Asynchronous processing of the instruction (ASY). If this specification is omitted, synchronous processing (SYN) is set.
7) The address of a 4-byte field containing the event item identifier - but only if the instruction is to be processed asynchronously, in which case the specification is mandatory (also see page 29).
8) The **first** word of the event information to be transferred when asynchronous processing of the instruction is completed. The first byte is overwritten with X'0C' when the information is returned.

   If this specification is omitted, the default value X'00000000' is transferred.

8a) The **second** word of the event information to be transferred when asynchronous processing of the instruction is completed.

   If this specification is omitted or is equal to the default value X'00000000', only the first word of the event information is transferred.

The connection must be described in CCB control block (see page 71). However, not all of the options can be specified (see diagram).

When using distribution codes, the section "Secondary opening", page 57, also applies.

3.2.2 Obtaining partner and connection information

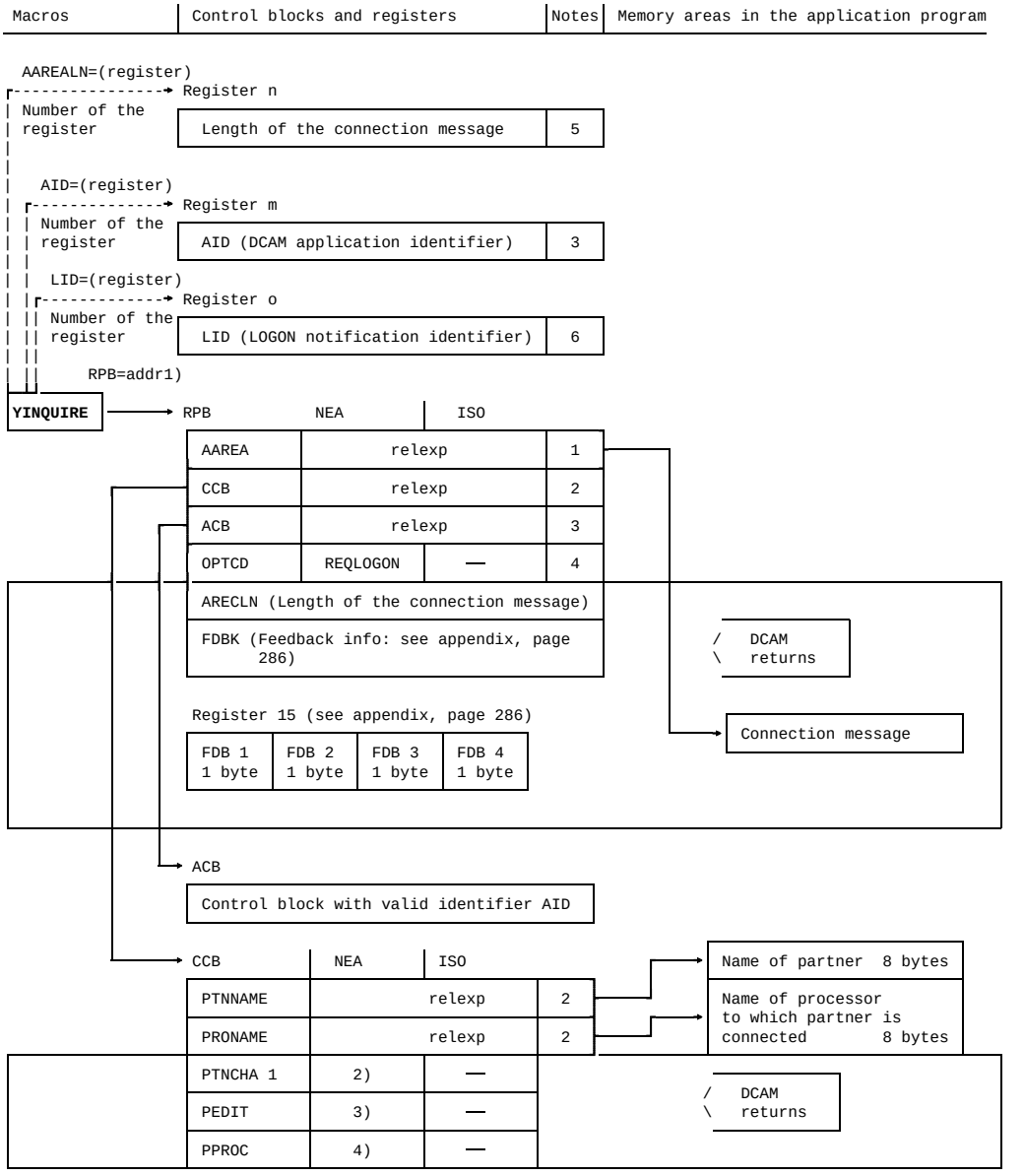
At connection setup, and at subsequent times, there are various reasons for requesting information about a partner or a connection:

- An unknown partner requests connection setup and proposes conditions for communication. He may have transmitted a connection message. This became known by the activation of the LOGON contingency routine (REQLOGON).
- No LOGON contingency routine was defined and consequently the first entry in a connection request queue, if any, is to be checked (TOPLOGON).
- The number of connected partners and of partners requesting connection is required (COUNTPTN).
- The names of the partner and of the processor are known but a possibly existing connection identifier (CID) is to be determined (NAMXLATE).
- The connection identifier (CID) is known and the names of the partner and of the processor are to be determined (CIDXLATE).

For DCAM(NEA) transport service applications, it is also possible to determine the characteristics of a partner (PTNCHAR), or to evaluate information about the terminal (BTERMINF), about the data display terminal and the character sets (MONCHARS), about the peripherals and about VTSUCB. See the ’VTSU User Guide’ for details on status information and the VTSU control block.

These different inquiries can be made with 6 variants of the YINQUIRE macro.

3.2.2.1 Partner information after a LOGON notification



1) Address of the control block that describes the macro
2) The first four bytes of the partner characteristics
3) Message editing, as proposed by the partner (see appendix, page 279)
4) If the partner wishes to start data transmission, this contains:
   PTNSTART: otherwise, it contains: ANYSTART.

The partner information is to be queried after a LOGON notification.

Mandatory specifications:

1) The address of an area in which the connection message is to be entered.
2) The address of a CCB control block in which various items of information are to be entered and which contains the field addresses for the partner and processor names.
3) Either the address of the ACB containing the valid application identifier AID or a register containing the identifier.

   This specification may be omitted only if the identifier was previously entered in the RPB (e.g. with YMODCB).

Optional specifications:

4) That information about the requesting partner is to be returned after a LOGON notification.
5) A register containing the length of the connection message; this length was passed in register 5 when the contingency routine was initiated.

   If this specification is omitted, the length (max. 80 bytes for a DCAM(NEA) transport service application or 32 bytes for a DCAM(ISO) transport service application) must have been entered previously in the AAREALN field of the RPB (e.g. with YMODCB).
6) A register containing the identifier LID that was transferred in register 4 when the LOGON contingency routine was initiated. This specification may be omitted only if the LID was entered previously in the RPB (e.g. with YMODCB).

Note

The connection message will be truncated if it is longer than the area whose length is specified in AAREALN. It is therefore recommended that the length of the area be specified as the length of the connection message transferred in register 5 at initiation of the contingency routine.

For DCAM(ISO) transport service applications, it is not possible to obtain information about the connection protocol which was actually used.

Example

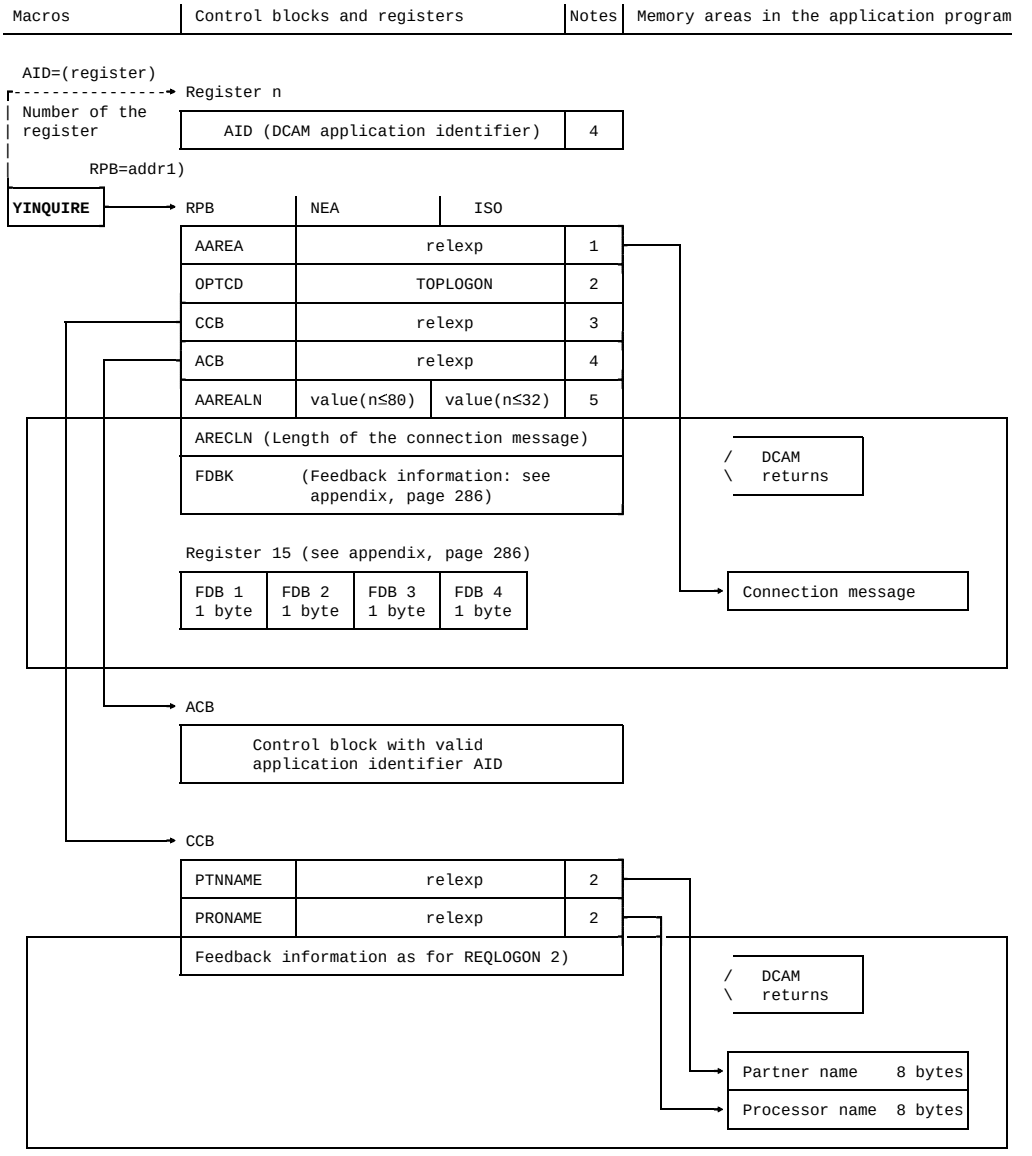
The user has defined a LOGON contingency routine (with the entry address LOGONADR). When it is initiated, he wants to know which partner has issued the connection request.

The following register information is passed to the user:

– AID in register 3
– LID in register 4
– Length of the connection message in register 5

```
LOGONADR        .
                .
                YINQUIRE    RPB=RPBCONT,
                            AID=(3),
                            LID=(4),
                            AAREALN=(5)
                .
                .
                RETCO
RPBCONT         YRPB        AAREA=BEREICH,
                            CCB=CCBCONT
CCBCONT         YCCB        PTNNAME=PTNAME,
                            PRONAME=PRNAME
PTNAME          DS          CL8
PRNAME          DS          CL8
BEREICH         DS          CL80          (32 bytes for ISO applications)
```

3.2.2.2 Partner information before connection setup



1) Address of the control block that describes the macro
2) For DCAM(NEA) transport service applications only, see page 90

The partner information is to be queried before connection setup.

Mandatory specifications:

1) The address of an area in which the connection message is to be entered.
2) Information is desired on the first partner requesting connection setup.
3) The address of a CCB control block in which various items of information are to be entered and which contains the field addresses for the partner and processor names.
4) Either the address of the ACB containing the valid application identifier (AID) or a register containing the valid identifier.

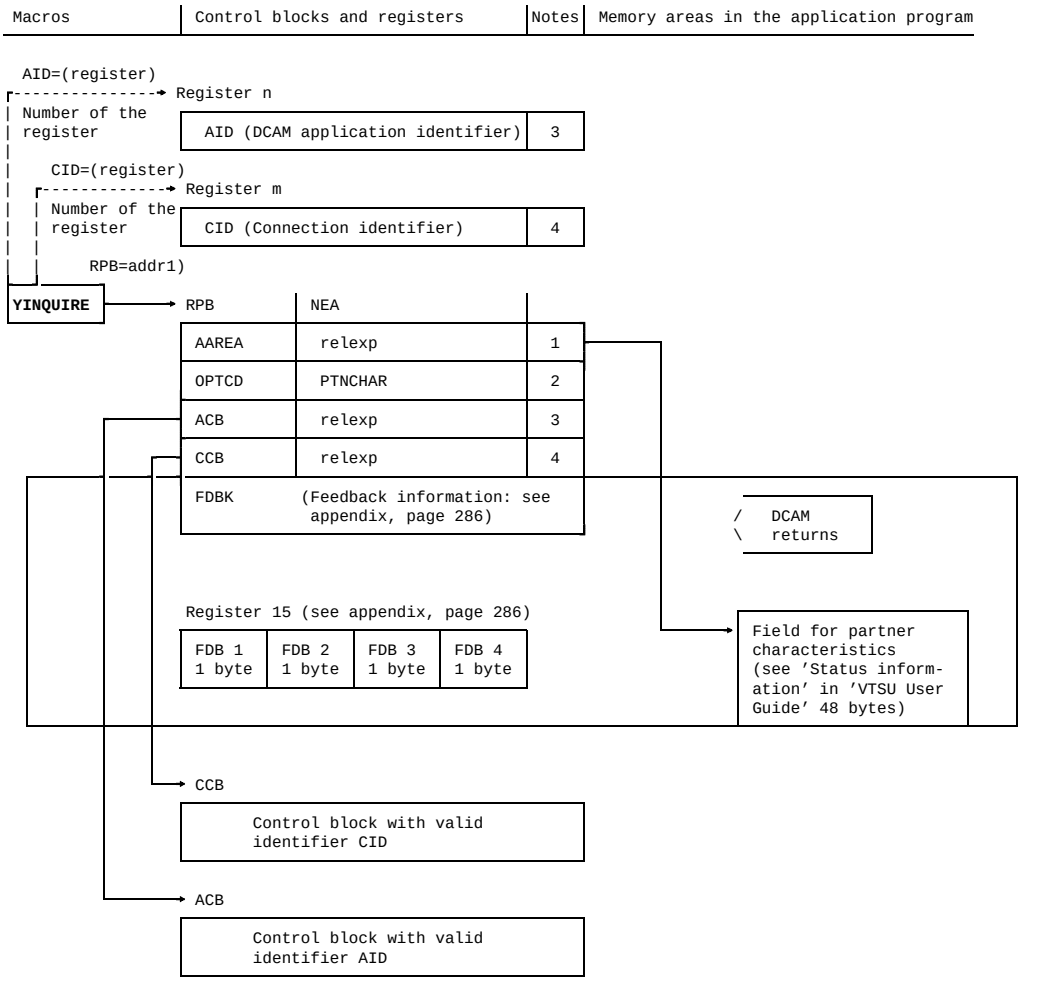
   If this specification is omitted, the valid identifier must previously have been entered in the RPB (e.g. with YMODCB).

Optional specification:

5) The length of the area in which DCAM enters the connection message (up to 80 bytes for a DCAM(NEA) transport service application or 32 bytes for a DCAM(ISO) transport service application). If the connection message is longer than this area it will be truncated and a warning will be issued in the feedback information.
   If this specification is omitted, the connection message is not transferred.

3.2.2.3 Partner characteristics

This section applies only to DCAM(NEA) transport service applications.



1) Address of the control block that describes the DCAM application

The partner characteristics are to be queried.

Mandatory specifications:

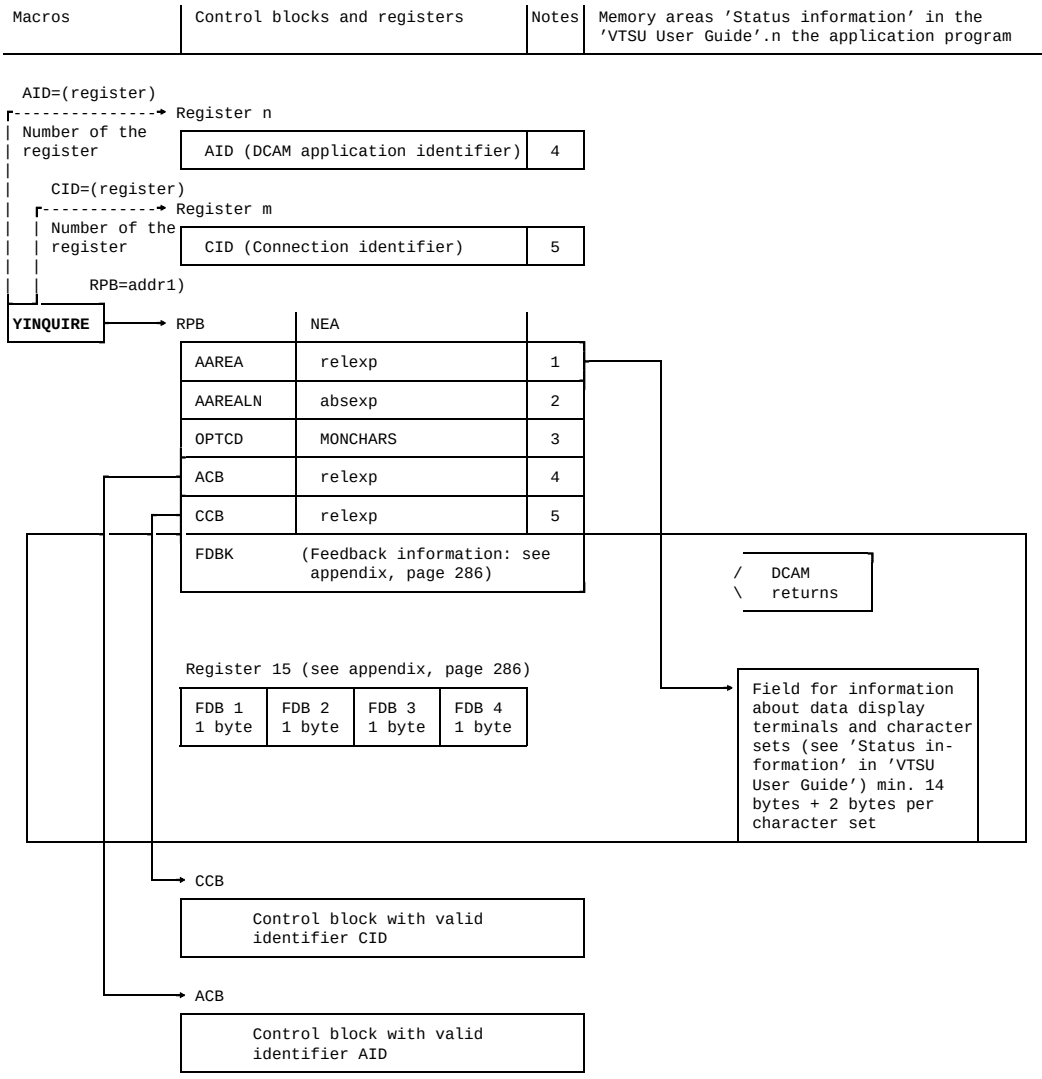
1) The address of a 48-byte area in which the partner characteristics are to be entered (see ’Status information’ in the ’VTSU User Guide’.
2) That the partner characteristics are to be queried.
3) Either the address of the ACB control block containing the valid application identifier AID or a register containing the valid identifier.

   If this specification is omitted, the valid identifier must previously have been entered in the RPB (e.g. with YMODCB).
4) Either the address of the CCB control block containing the valid connection identifier CID or a register containing the valid identifier.

   If this specification is omitted, the valid identifier must previously have been entered in the RPB (e.g. with YMODCB).

3.2.2.4 Data display terminal and character sets

This section applies only to DCAM(NEA) transport service applications.



1) Address of the control block that describes the DCAM application

The description of the data display terminal and the character sets is to be queried.

Mandatory specifications:

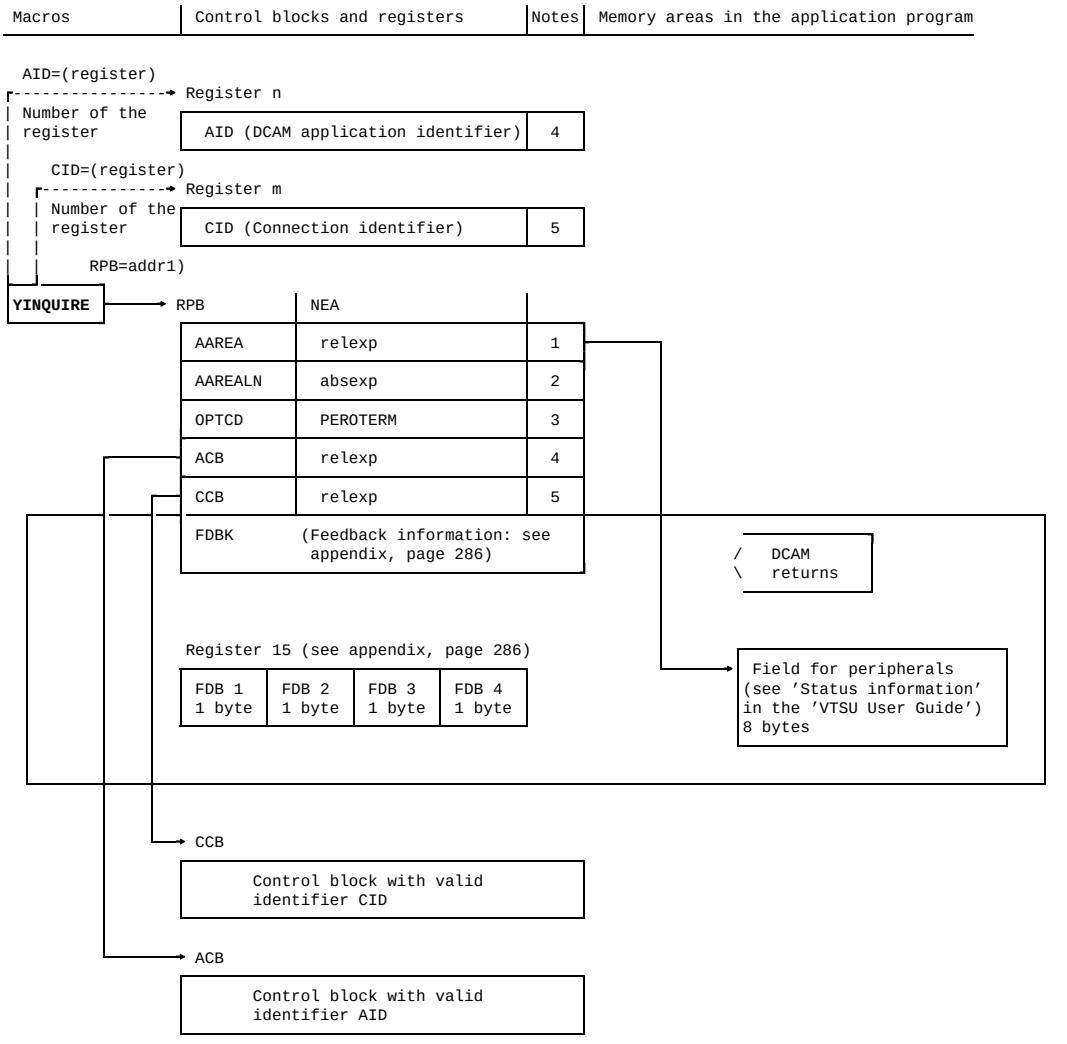
1) The address of an area with a length of at least 14 bytes in which the description of the data display terminal and the character sets is to be placed (see ’Status information’ in the ’VTSU User Guide’).
2) The length of the area into which DCAM is to write (at least 14 bytes + 2 bytes per character set). If the description is longer than this, the character sets are omitted.
3) That the description of the data display terminal and character sets is to be queried.
4) Either the address of the ACB control block containing the valid application identifier AID or a register containing the valid identifier.

   If this specification is omitted, the valid identifier must previously have been entered in the RPB (e.g. with YMODCB).
5) Either the address of the CCB control block containing the valid connection identifier CID or a register containing the valid identifier.

   If this specification is omitted, the valid identifier must previously have been entered in the RPB (e.g. with YMODCB).

3.2.2.5 Peripherals

This section applies only to DCAM(NEA) transport service applications.



1) Address of the control block that describes the DCAM application

Information about the peripherals is to be requested.

Mandatory specifications:

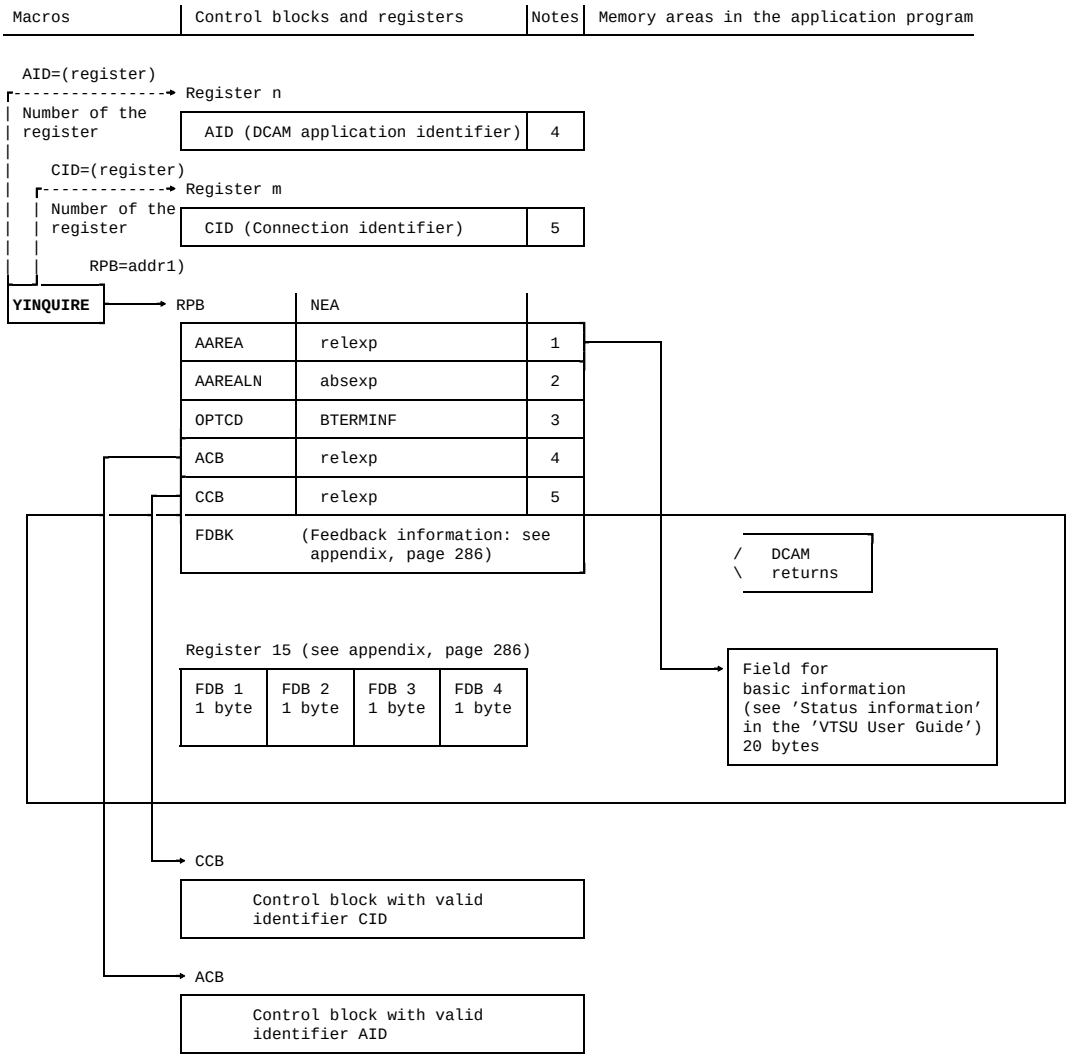
1) The address of an 8-byte area in which the information about the peripherals is to be placed (see ’Status Information’ in the ’VTSU User Guide’).
2) The length of the area into which DCAM is to write the information. If the information is longer than this, the YINQUIRE will be rejected; DCAM then returns the return code FDBK 04 00 28.
3) That information about the peripherals is required.
4) Either the address of the ACB control block containing the valid application identifier AID or a register containing the valid identifier.

   If this specification is omitted, the valid identifier must previously have been entered in the RPB (e.g. with YMODCB).
5) Either the address of the CCB control block containing the valid connection identifier CID or a register containing the valid identifier.

   If this specification is omitted, the valid identifier must previously have been entered in the RPB (e.g. with YMODCB).

3.2.2.6 Basic information about the terminal

This section applies only to DCAM(NEA) transport service applications.



1) Address of the control block that describes the DCAM application

The basic information about the terminal is to be queried.

Mandatory specifications:

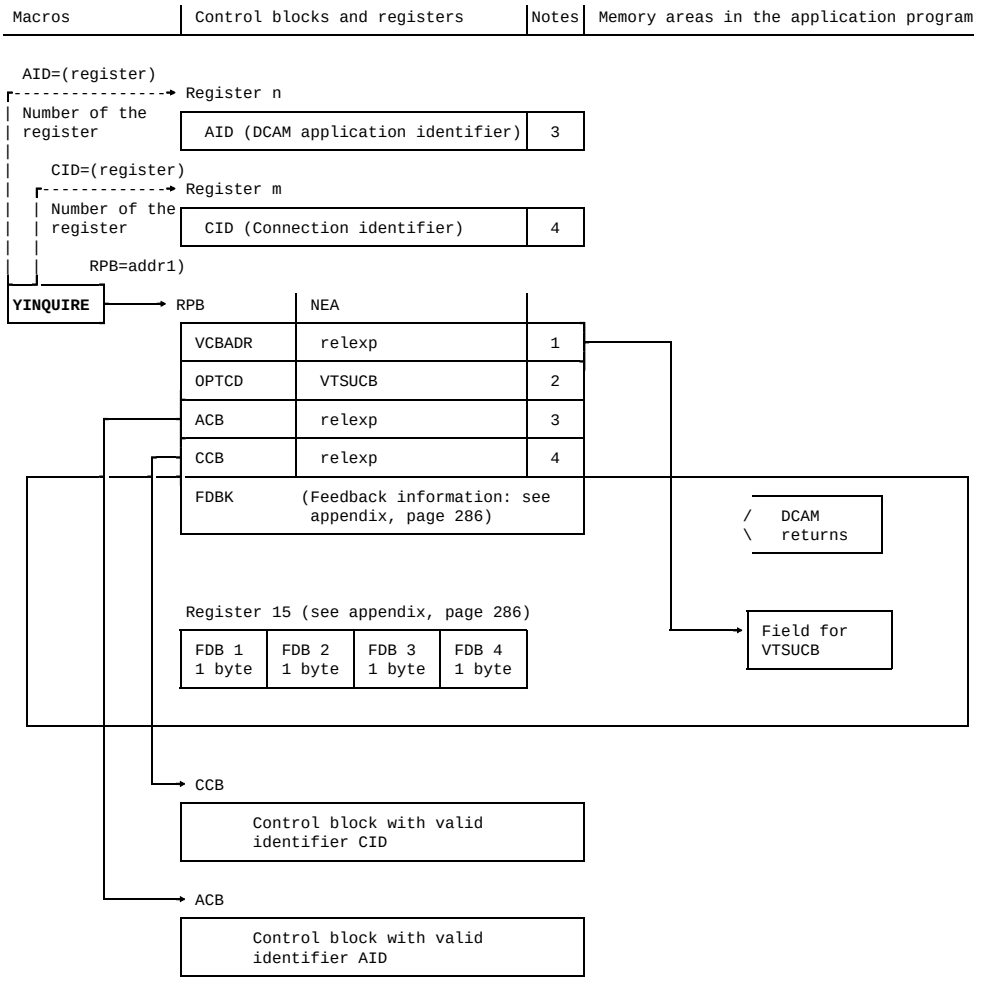
1) The address of an 20-byte area in which the basic information about the terminal is to be placed (see ’Status information’ in the VTSU User Guide’).
2) The length of the area into which DCAM is to write the information. If the information is longer than this, the YINQUIRE will be rejected; DCAM then returns the return code FDBK 04 00 28.
3) That the basic information about the terminal is required.
4) Either the address of the ACB control block containing the valid application identifier AID or a register containing the valid identifier.

   If this specification is omitted, the valid identifier must previously have been entered in the RPB (e.g. with YMODCB).
5) Either the address of the CCB control block containing the valid connection identifier CID or a register containing the valid identifier.

   If this specification is omitted, the valid identifier must previously have been entered in the RPB (e.g. with YMODCB).

3.2.2.7 VTSU control block

This section applies only to DCAM(NEA) transport service applications.



1) Address of the control block that describes the DCAM application

The contents of the VTSU control block are to be queried.

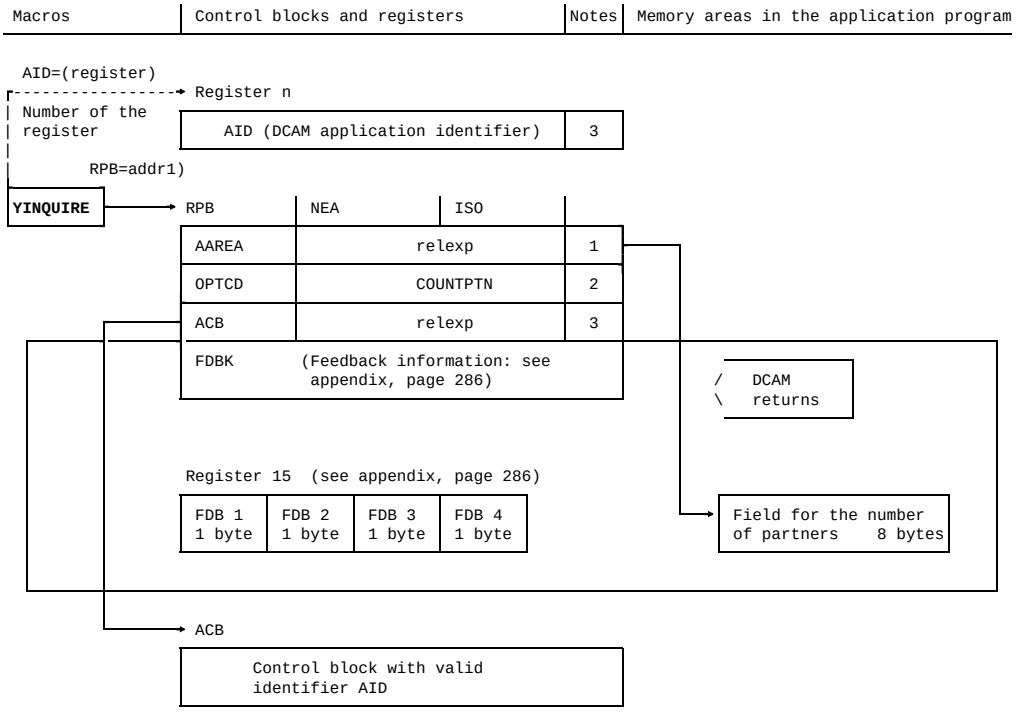
Mandatory specifications:

1) The address of an area in which the VTSU control block is to be placed (see page 'VTSU control block' in the 'VTSU User Guide').
2) That the VTSU control block is required.
3) Either the address of the ACB control block containing the valid application identifier AID or a register containing the valid identifier.

   If this specification is omitted, the valid identifier must previously have been entered in the RPB (e.g. with YMODCB).
4) Either the address of the CCB control block containing the valid connection identifier CID or a register containing the valid identifier.

   If this specification is omitted, the valid identifier must previously have been entered in the RPB (e.g. with YMODCB).

3.2.2.8 Number of partners

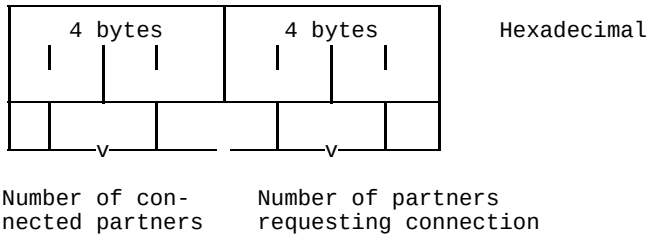


1) Address of the control block that describes the macro

The number of partners is to be queried.

Mandatory specifications:

1) The address of an 8-byte field in which the information is entered by DCAM as follows:

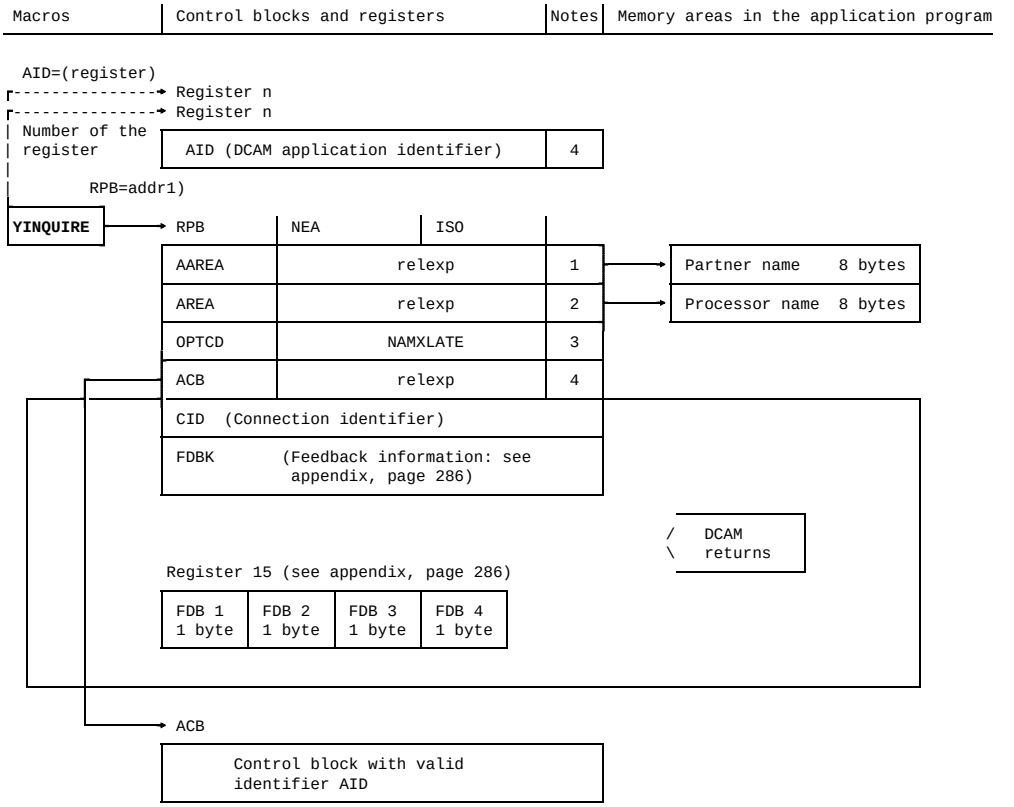


2) That the number of partners is required.

3) Either the address of the ACB control block containing the valid application identifier AID or a register containing the valid identifier.

   If this specification is omitted, the valid identifier must previously have been entered into the RPB (e.g. with YMODCB).

3.2.2.9 Connection identifier (CID)



1) Address of the control block that describes the macro

The connection identifier is to be queried.

Mandatory specifications:

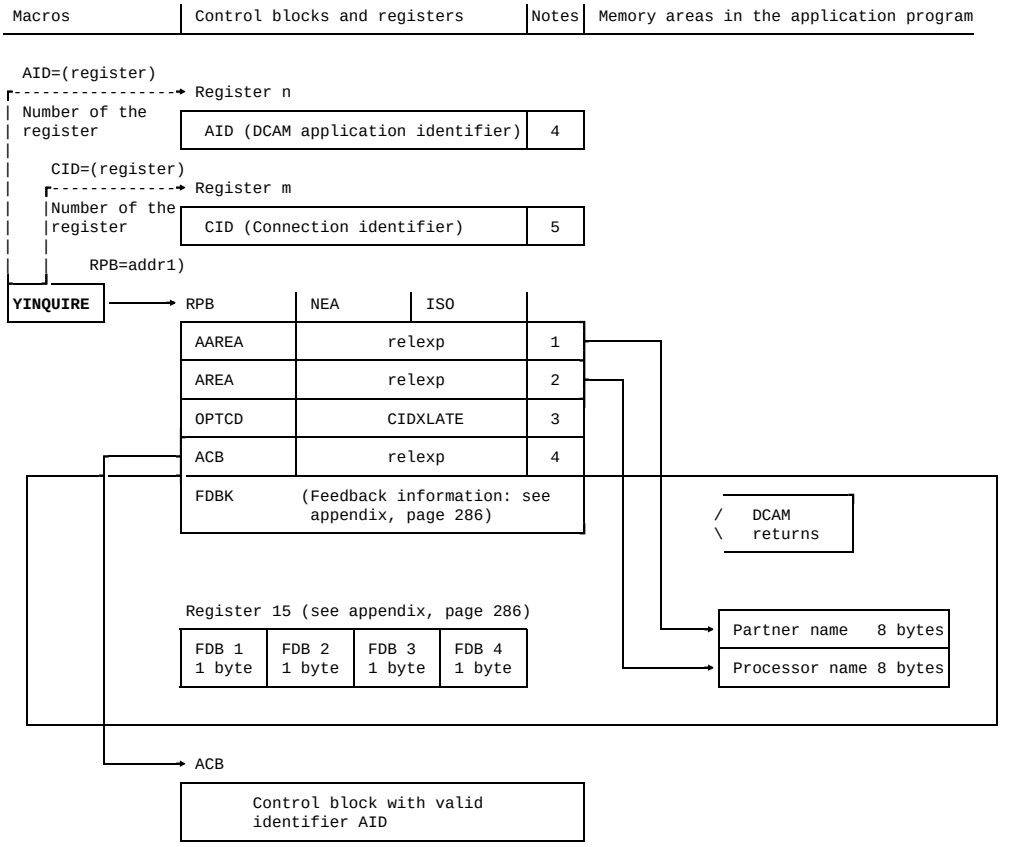
1) The address of an 8-byte field containing the partner name.
2) The address of an 8-byte field containing the processor name of the partner.
3) That the connection identifier is required.
4) Either the address of the ACB control block containing the valid application identifier AID or a register containing the valid identifier.

   If this specification is omitted, the valid identifier must previously have been entered in the RPB (e.g. with YMODCB).

Note

– Only the names of partners which are connected to the specified application, which was opened by this task, can be queried.
– If, for a DCAM(ISO) transport service application, there are several connections (parallel connections) to one partner, one of the valid identifiers is returned.

3.2.2.10 Partner and processor name



1) Address of the control block that describes the macro

The partner and processor names are to be queried.

Mandatory specifications:

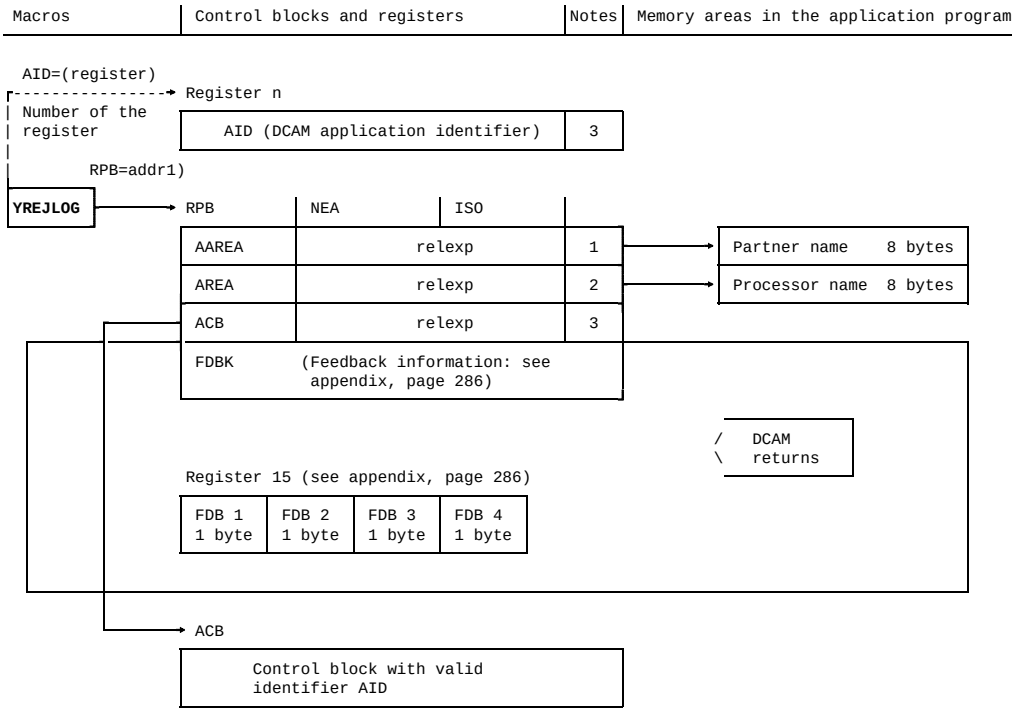
1) The address of an 8-byte field in which the partner name is to be placed.
2) The address of an 8-byte field in which the processor name of the partner is to be placed.
3) That the partner and processor names are to required.
5) A register containing the connection identifier (CID). If this specification is omitted, the valid identifier must previously have been entered in the RPB (e.g. with YMODCB).
4) The address of the ACB control block containing the valid application identifier (AID) or a register containing the valid identifier.

   If this specification is omitted, the valid identifier must previously have been entered in the RPB (e.g. with YMODCB).

Note

The first byte of the returned partner name is overwritten with a space if the specified identifier CID was invalid.

3.2.3 Rejecting a connection request



1) Address of the control block that describes the macro

A connection request is to be rejected.

Mandatory specifications:

1) The address of an 8-byte field containing the name of the partner with whom a connection is not to be established.
2) The address of an 8-byte field containing the processor name of the partner with whom a connection is not to be established.
3) Either the address of the ACB control block containing the valid identifier AID or a register containing the valid identifier.

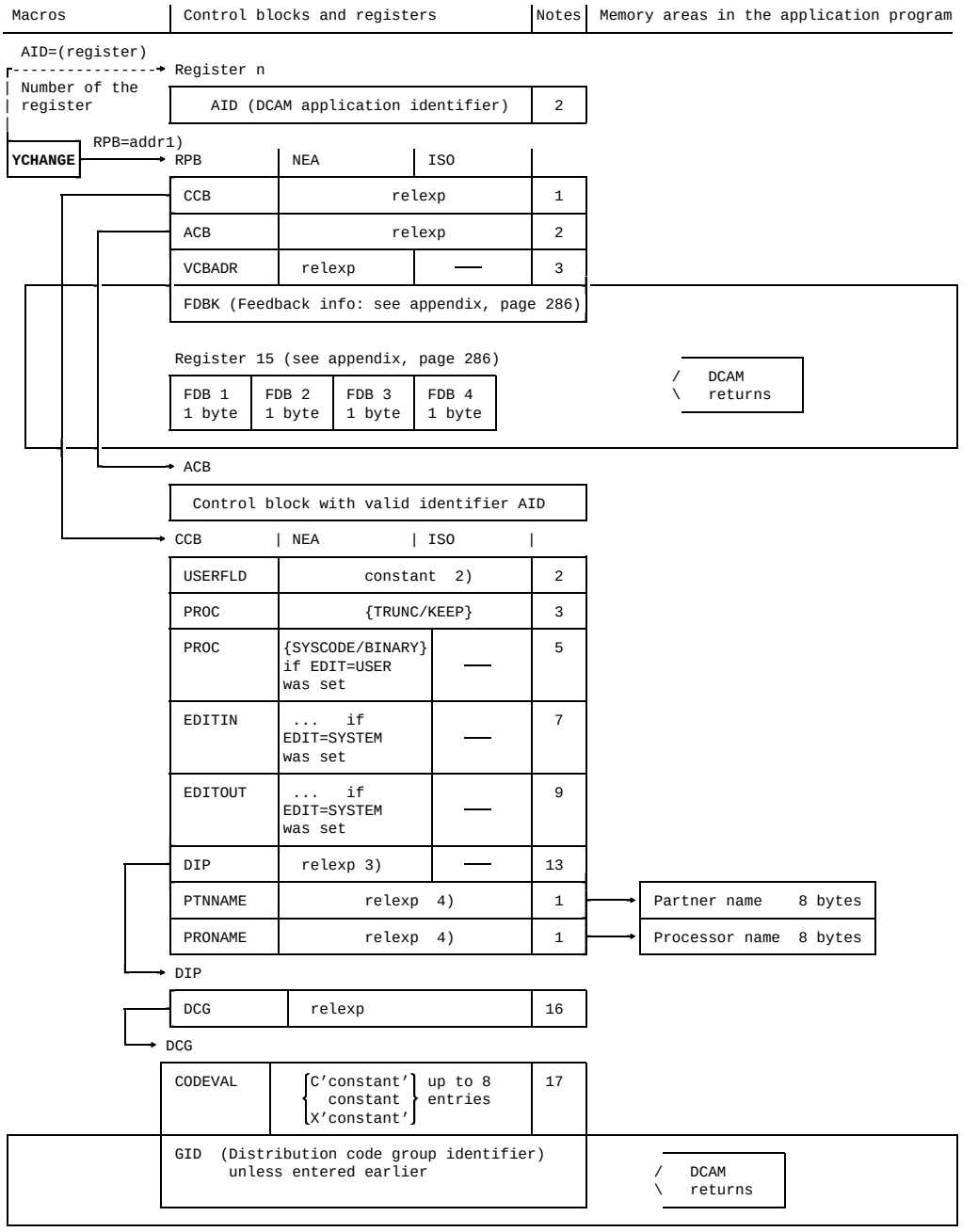
   If this specification is omitted, the valid identifier must previously have been entered in the RPB (e.g. with YMODCB).

Example

A connection request (notified by the activation of a LOGON contingency routine) from a communication partner with the name PARTNER connected to the processor with the name PROC is to be rejected.

```
*               LOGON contingency routine
                .
                .
                .
                YINQUIRE    RPB=RPB1,
                            AID=(3),
                            LID=(4),
                            AAREALN=(5),
                            AAREA=BEREICH
                .
                .
                .
                YREJLOG     RPB=RPB1,
                            AID=(3),
                            AAREA=PTNNAM,
                            AREA=PRONAM
RPB1            YRPB        CCB=CCB1
CCB1            YCCB        PRONAME=PRONAM,
                            PTNNAME=PTNNAM
PTNNAM          DC          CL8'PARTNER'
PRONAM          DC          CL8'PROC'
BEREICH         DS          CL80          (32 for ISO)
```

3.2.4 Changing the characteristics of a connection



```
1) Address of the control block that describes the macro
2) constant may be defined as a character constant: C'constant', a
   hexadecimal constant: X'constant', a fixed-point constant: F'constant'
   or an address constant': A'constant', V(constant).
3) If distribution codes are used and new codes (DCG) are to be assigned to
   this connection.
4) If the CID in the CCB control block is invalid.
```

The characteristics of a connection are to be changed.

Mandatory specifications:

1) The address of the CCB control block describing the connection whose characteristics are to be changed.
   This specification is required even if the RPB contains a valid connection identifier. If the CCB does not contain a valid identifier, the name and the processor name of the partner must be specified in the CCB instead.

2) Either the address of the ACB control block containing the valid application identifier (AID) or a register containing the valid identifier.

   If this specification is omitted, the identifier must previously have been entered in the RBP (e.g. with YMODCB).

Optional specification:

3) The address of an area in which VTSUCB is passed (NEA transport service only). If VCBADR≠NULL, VTSUCB is copied into the data structure within DCAM. If VCBADR=NULL, the internal DCAM copy of VTSUCB is deleted. Message editing is continued with the options specified in the CCB control block. The data is accepted at connection setup (YOPNCON) or at the last change of the connection characteristics (YCHANGE). The specification is evaluated only if OPTCD=VTSUCB was set.

The characteristics of the connection to be modified must be described in CCB control block. Only those fields which can be modified are evaluated. The following summary shows which fields these are. For details, see the description of the connection characteristics. The numbering of the individual field refers to this description (see page 71).

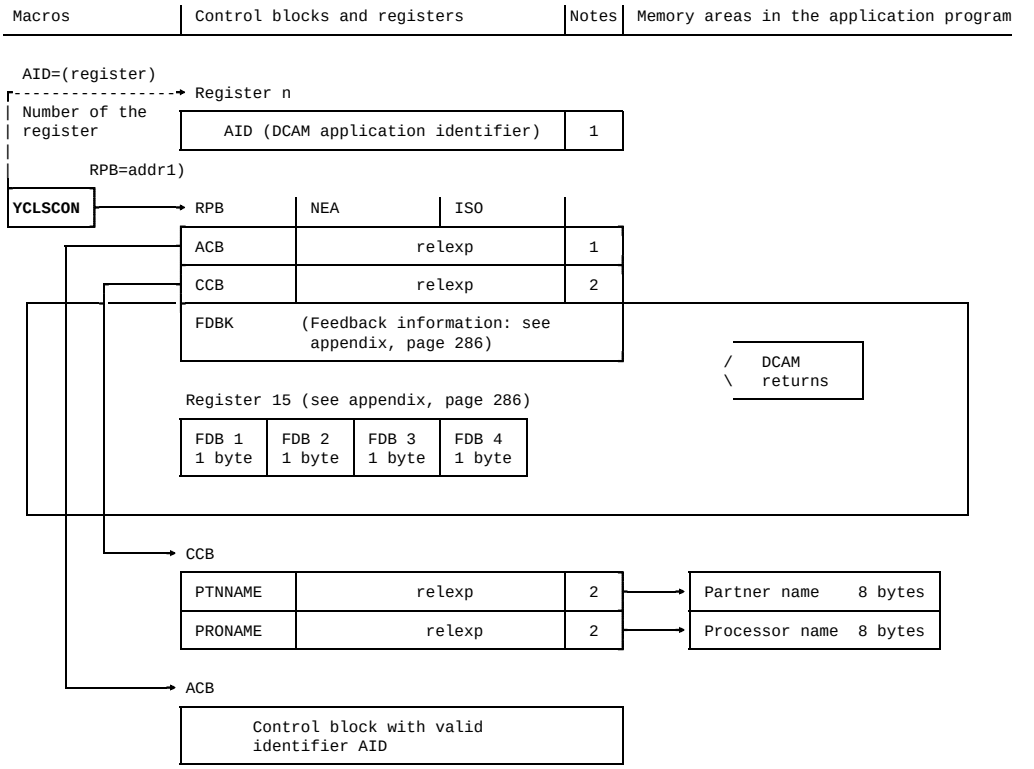
If, in a DCAM(NEA) transport service application, distribution codes are used, new control blocks DCG can be linked; modification of the code position (CODEPOS), code length (CODELN) and code indicator (CODEIND) entries in the DIP is not possible (see page 84).

Example

Input message editing (EDITIN) is to be changed from PHYS to LINE for the DCAM(NEA) transport service connection described by the CCB control block with the address CCB1.

```
                .
                .
                .
                YOPNCON     RPB=RPB1
                .
                .
                .
                YMODCB      BLK=CCB,
                            BLKADDR=CCB1,
                            EDITIN=LINE
                YCHANGE     RPB=RPB1
                .
                .
                .
RPB1            YRPB        ACB=ACB1,
                            CCB=CCB1
                            . further information relates to YOPNCON
                            .
                            .
CCB1            YCCB        PRONAME=NAM1,
                            PTNNAME=NAM2,
                            EDIT=SYSTEM,
                            EDITIN=PHYS
                .
                .
                .
ACB1            YACB        ...
```

3.2.5 Canceling a request



1) Address of the control block that describes the macro

A request is to be canceled.

Mandatory specifications:

1) Either the address of the ACB control block containing the valid application identifier AID or a register containing the valid identifier.

   If this specification is omitted, the valid identifier must previously have been entered in the RPB (e.g. with YMODCB).

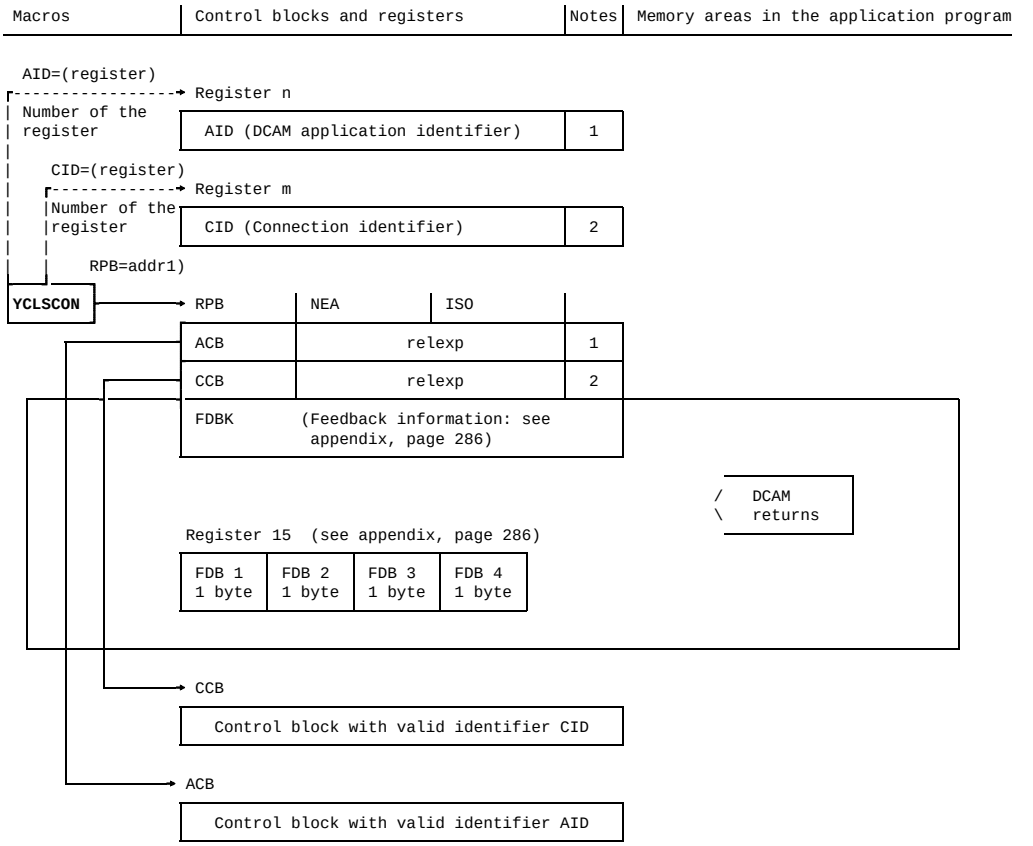
2) The address of the CCB control block containing the addresses of fields with the partner and processor names. A connection request waiting for acceptance by the partner addressed there is deleted.

Example

The connection request directed to a partner addressed in the control block CCB at the address CCBNAM is to be deleted.

```
                .
                .
                .
                YCLSCON    RPB=RPBNAM
                .
                .
                .
RPBNAM          YRPB        CCB=CCBNAM
CCBNAM          YCCB        PTNNAME=PTNAD,
                            PRONAME=PROAD
PTNAD           DC          CL8'PART1'
PROAD           DC          CL8'PROC0'
```

3.2.6 Clearing down a connection



1) Address of the control block that describes the macro

A connection is to be cleared down.

Mandatory specifications:

1) Either the address of the ACB control block containing the valid application identifier (AID) or a register containing the valid identifier.

   If this specification is omitted, the valid identifier must previously have been entered in the RPB (e.g. with YMODCB).

2) Either the address of the CCB control block containing the valid connection identifier (CID) or a register containing the valid identifier.

   If this specification is omitted, the valid identifier must previously have been entered in the RPB (e.g. with YMODCB).

Example

The connection whose identifier CID is given in register 3 is to be cleared down.

```
                .
                .
                .
                CLSCON      RPB=RPBNAM,CID=(3)
                .
                .
                .
RPBNAM          YRPB        ...
```

3.3 Data communication-oriented macros

The communication partners can transfer data after successful completion of connection setup.
A send or receive macro always means data transfer from the user area into the data storage of the communication system or vice versa. A send macro, for example, is completed successfully when the data to be transmitted has been placed in the data storage of the communication system.

By means of a combined send/receive macro, the user can save one program interrupt.

Distribution code assignment for DCAM(NEA) transport service applications is controlled with 2 macros.

Data communication-oriented macros are:

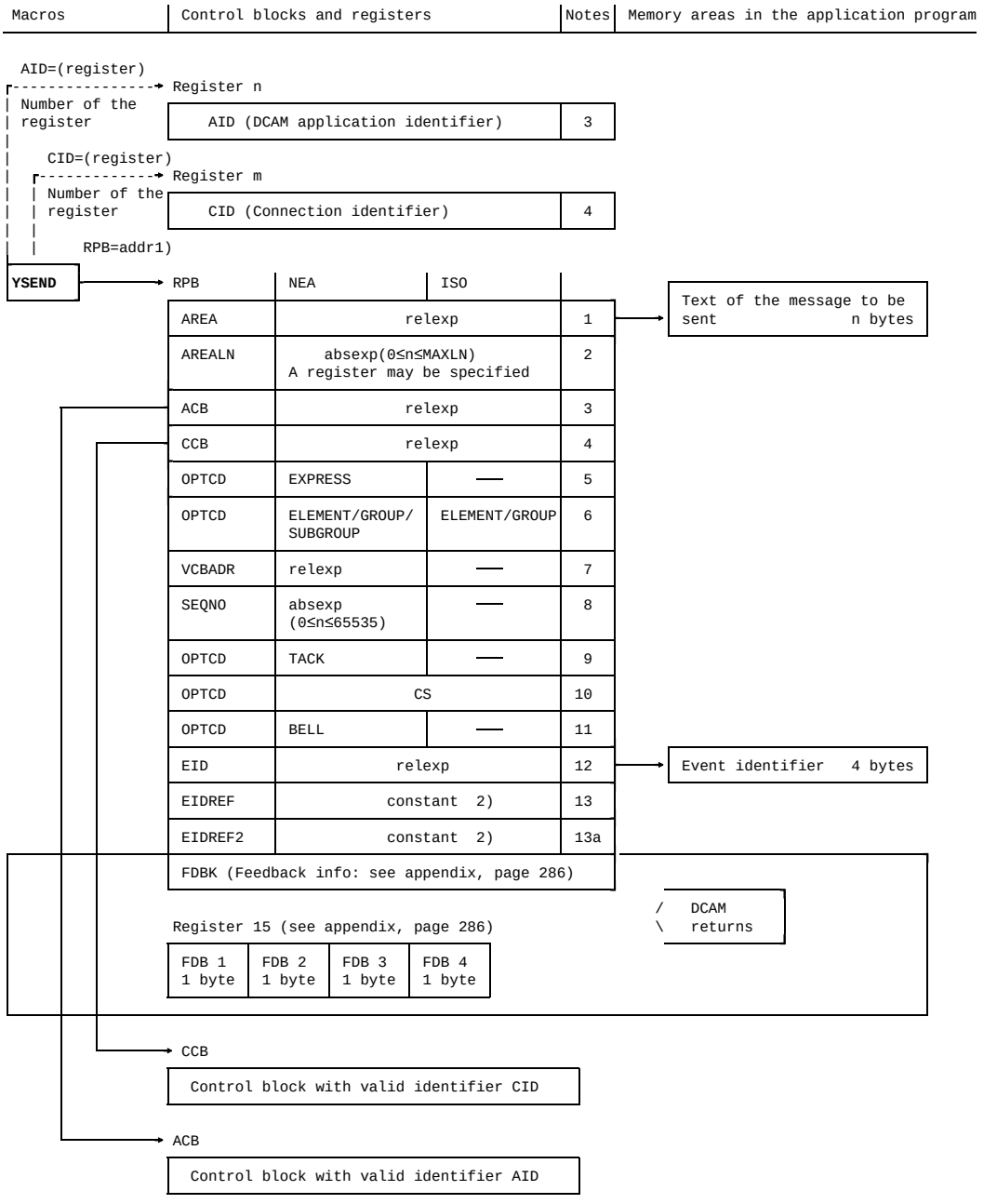
- **YSEND** **send a message**
- **YRECEIVE** **receive a message**
- **YSENDREC** **send a message to a partner and then receive a message from the same partner**
- **YRESET** **cancel YRECEIVE macros and change the CS/CA state of the connection**
- **YPERMIT** **permit a task of the task group to receive messages with specific distribution codes** (NEA transport service only)
- **YFORBID** **forbid a task of the task group to receive messages with specific distribution codes** (NEA transport service only)

For DCAM(NEA) transport service applications, DCAM tasks can process the following asynchronous notifications by means of an suitable contingency routine (see pages 29 and 34):

- **EXPR** **express message received**
- **TACK** **transport acknowledgment received**
- **message received successfully**

The terms "message/data unit" and "more-data function" for DCAM(ISO) transport service applications are defined in the manual "DCAM Program Interfaces" and are not explained in detail in this manual.

3.3.1 Sending a message



```
1) Address of the control block that describes the macro
2) constant may be defined as a character constant: C'constant', a
   hexadecimal constant: X'constant', a fixed-point constant: F'constant'
   or an address constant': A'constant', V(constant).
```

A message is to be sent.

Mandatory specifications:

1) The address of the field containing the data to be transmitted.
2) The length of the message to be transmitted.

   In a DCAM(NEA) transport service application, this specification may be omitted when an express message is to be transmitted, in which case the maximum length of 8 bytes is assumed.
3) The address of the ACB control block containing the valid application identifier (AID) or a register containing the valid identifier. This specification may be omitted if the valid identifier was previously entered in the RPB (e.g. with YMODCB).
4) The address of the ACB control block containing the valid connection identifier (CID) or a register containing the valid identifier. This specification may be omitted if the valid identifier was previously entered in the RPB (e.g. with YMODCB).

Optional specifications:

5) That the message is to be an express message, i.e. it is to bypass all message flow controls and is to be able to pass other messages addressed to the same partner (NEA transport service only).
   This specification includes the condition that the maximum length is 8 bytes. For normal messages this specification is not required.
6) That, for a DCAM(ISO) transport service application, the data unit being sent is the the last unit of a message (OPTCD=GROUP) or that further data units follow. This operand is evaluated only if the connection was opened with MDATA=Y.

   For message structuring in a DCAM(NEA) transport service application, which part of the message is to be sent. Possible entries are ELEMENT and SUBGROUP. If nothing is specified, GROUP is assumed. This specification is possible only if EDITOUT=PHYS was specified in the CCB for EDIT=USER or EDIT=SYSTEM.

7) The address of an area in which VTSUCB is passed (NEA transport service only). If VCBADR≠NULL, VTSUCB is copied into the data structure within DCAM. If VCBADR=NULL, the internal DCAM copy of VTSUCB is deleted. Message editing is continued with the options specified in the CCB control block. The data is accepted at connection setup (YOPNCON) or at the last change of the connection characteristics (YCHANGE). The specification is evaluated only if OPTCD=VTSUCB was set.

8) A message sequence number specified by the user (NEA transport service only). This specification is required if a transport acknowledgment is requested. If it is omitted, the number 0 is entered.

9) That a transport acknowledgment is requested for this message (NEA transport service only).

10) That messages arriving from this partner after macro execution are to be placed in the originator-oriented queue (CS). If this specification is omitted, the common receiver queue is used (CA).

    The CS specification is required if CS was set previously and receive macros (e.g. YRECEIVE) have already been issued and have not yet been completed.

    The CS specification cannot be made more than once: If it has already been included in another macro (e.g. YRECEIVE), but the CS state has not yet been set (asynchronous execution), it cannot be specified again.

    For a DCAM(NEA) transport service application, this specification (CS) is meaningful only when distribution codes are not used (ATTR=NSHARE or (SHARE, NDISCO) specified in YOPEN).

11) That the message is to be linked to the terminal with an audible alarm; interpreted only if EDIT=SYSTEM (NEA transport service only).

12) The address of the field in which the event identifier for GO signals is located, but only if the connection is established with PROC=SIGNAL.

13) The **first** word of the event information to be transferred when the GO signal arrives.
    This specification is required only in the case of PROC=SIGNAL.
    The first byte is overwritten with X'0C' when the information is returned.
    If this specification is omitted, the default value X'00000000' is passed.

13a) The **second** word of the event information to be transferred when the GO signal arrives.
This specification is required only in the case of PROC=SIGNAL.
If this specification is omitted or is set equal to the default value X'00000000', only the first word of the event information is passed.

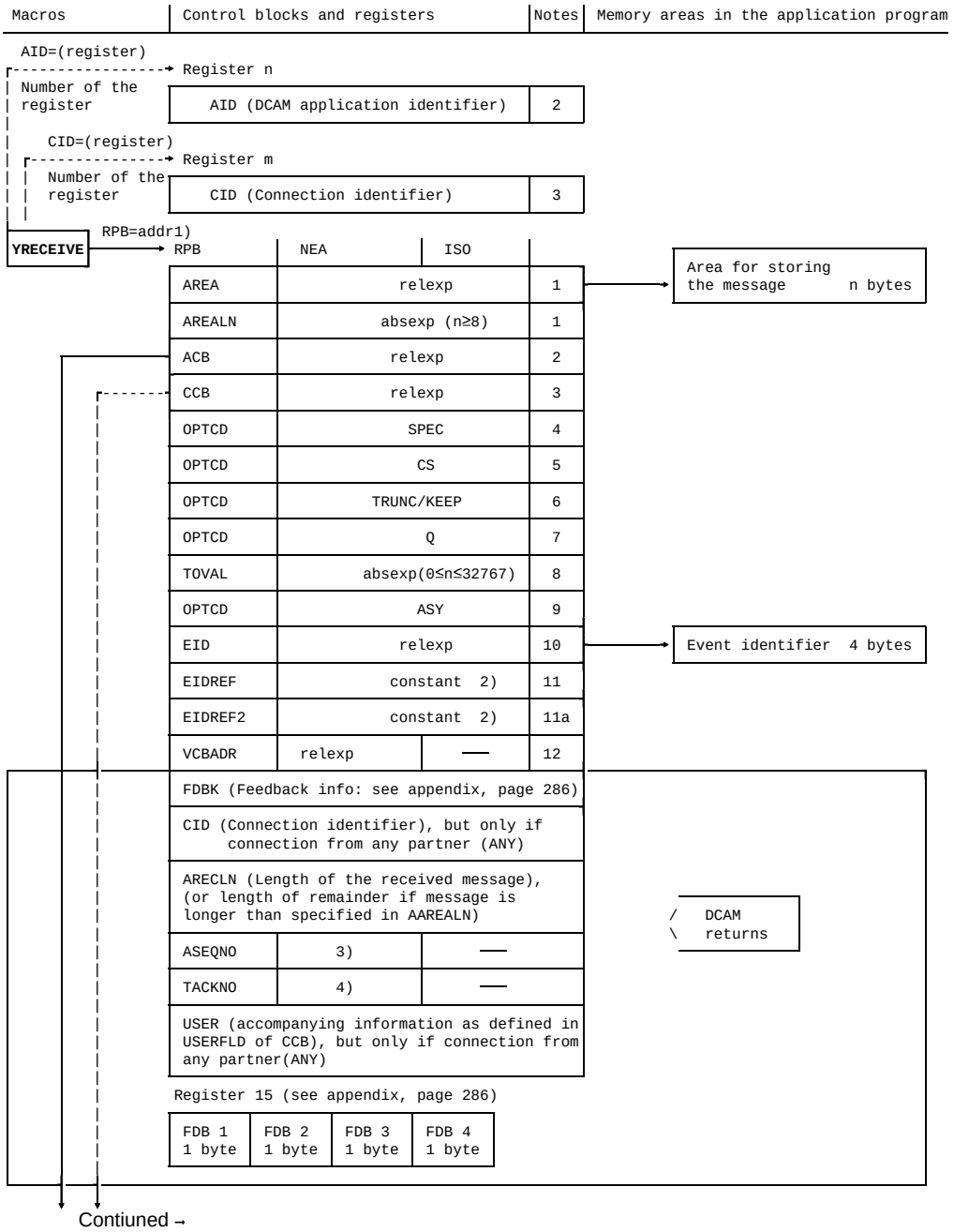
See the notes on page 229.

Example

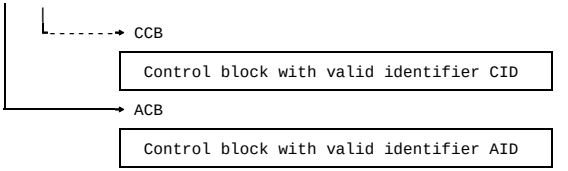
A normal message with the sequence number 15 (contained in register 5) is to be transmitted over the connection described in CCB control block with the address CCBAD. The message text is 'HALLO PARTNER'.

```
                .
                .
                YSEND       RPB=RPBAD,
                            SEQNO=(5)              (NEA transport service only)
                .
                .
RPBAD           YRPB        ACB=ACBAD,
                            CCB=CCBAD,
                            AREA=TEXT,
                            AREALN=13
ACBAD           YACB        ...
CCBAD           YCCB        ...
TEXT            DC          C'HALLO PARTNER'
```

3.3.2 Receiving a message or transport acknowledgment



Contiuned




```
1) Address of the control block that describes the macro
1) Address of the control block that describes the macro
2) constant may be defined as a character constant: C'constant', a hexadecimal constant: X'constant',
   a fixed-point constant: F'constant' or an address constant': A'constant', V(constant).
3) Sequence number as defined in SEQNO of the partner's DCAM application or as defined in the data
   communication system if the partner is a terminal.
4) Sequence number of the message being acknowledged if a transport acknowledgment is received.
```

A message or a transport acknowledgment is to be received.

Mandatory specifications:

1) The address of the area in which the data is to be entered, and the length of the area, which must be at least 8 bytes.

2) Either the address of the ACB control block containing the valid application identifier (AID) or a register containing the valid identifier.

   If this specification is omitted, the valid identifier must previously have been entered in the RPB (e.g. with YMODCB).

3) Either the address of the CCB control block containing the valid connection identifier (CID) or a register containing the valid identifier.

   If this specification is omitted, the valid identifier must previously have been entered in the RPB (e.g. with YMODCB). This specification may be omitted if a message is to be received from any partner (ANY).

Optional specifications:

4) That a message is to be received from a specific partner (SPEC).

   This specification may not be made when the originator-oriented queue (CS) has been set and a YRECEIVE or YSENDREC with SPEC and switchover to the common receiver queue (CA) has already been issued but not yet been completed.

   This specification may also not be made when the common receiver queue has been set.

   If this specification is omitted, a message from any partner is accepted (ANY).

5) That, after execution of the macro, messages from this partner are to be entered in the originator-oriented queue (CS).

   If this specification is omitted, the common receiver queue (CA) is set.

   This operand is not evaluated if a transport acknowledgment is received with YRECEIVE in a DCAM(NEA) transport service application. This specification is evaluated only when distribution codes are not used (ATTR=NSHARE or (SHARE, NDISCO) specified in YOPEN).

6) Handling of excess-length messages (messages that are longer than specified in AAREALN) - either truncation (TRUNC) with the excess-length part being lost, or storage of the excess-length part (KEEP) for another YRECEIVE.

   If this specification is omitted, the specification made during connection setup is used (CCBTK, see page 71).

   A requirement for KEEP is that the originator-oriented queue (CS) is set here at the latest.

7) That the macro is to be placed in a queue if it cannot be processed immediately (Q).

   If this specification is omitted, the macro is terminated immediately, if applicable with an error message, i.e. it is issued only "just in case" and is not placed in a queue (NQ).

   This specification is required when the instruction is to be processed asynchronously.

8) The total time (TOVAL) the YRECEIVE is to remain in the queue. The macro is terminated after this time if the desired data has not yet arrived.

   If no entry is made, a time of 600 s is set. The entry ’0’ specifies an indefinite wait time.

   This specification is not evaluated if entry in a queue is not desired (NQ).

9) That the instruction is to be processed asynchronously (ASY). If this specification is omitted, the instruction is processed synchronously (SYN).

10) The address of the field containing - only if the instruction is to be processed asynchronously - the event identifier, in which case the specification is mandatory (see page 29).

11) The **first** word of the event information to be transferred when asynchronous processing of the instruction is completed. The first byte is overwritten with X'0C' when the information is returned.

If this specification is omitted, the default value X'00000000' is passed.

11a) The **second** word of the event information to be transferred when asynchronous processing of the instruction is completed.

If this specification is omitted or is set equal to the default value X'00000000', only the first word of the event information is passed.

12) The address in the application program into which the internal VTSUCB is to be copied (NEA transport service only).
This specification is evaluated only if OPTCD=VTSUCB was set.

Note for DCAM(ISO) transport service applications

Feedback information from DCAM(ISO) transport service applications:

For MDATA=Y

Flag X'21' in FDB4 →	further data units to follow: the message is not yet complete.
or	
Flag X'81' in FBD4 →	no further data units to follow: the message is complete.

For MDATA=N, only X'81' (message) is flagged in FDB4.

In DCAM(NEA) transport service applications, a second, negative transport acknowledgment may follow YSEND if EDIT=SYSTEM is set. For SHARE applications, this is sent to the primary task, regardless of the setting of REQTASK.

Example 1 (synchronous processing)

A message with a maximum length of 100 bytes is to be received over the connection described in the CCB control block with the address CCBAD. If a longer message arrives, the excess-length part is to be stored. The message is to be placed in the area with the address EMPFANG.

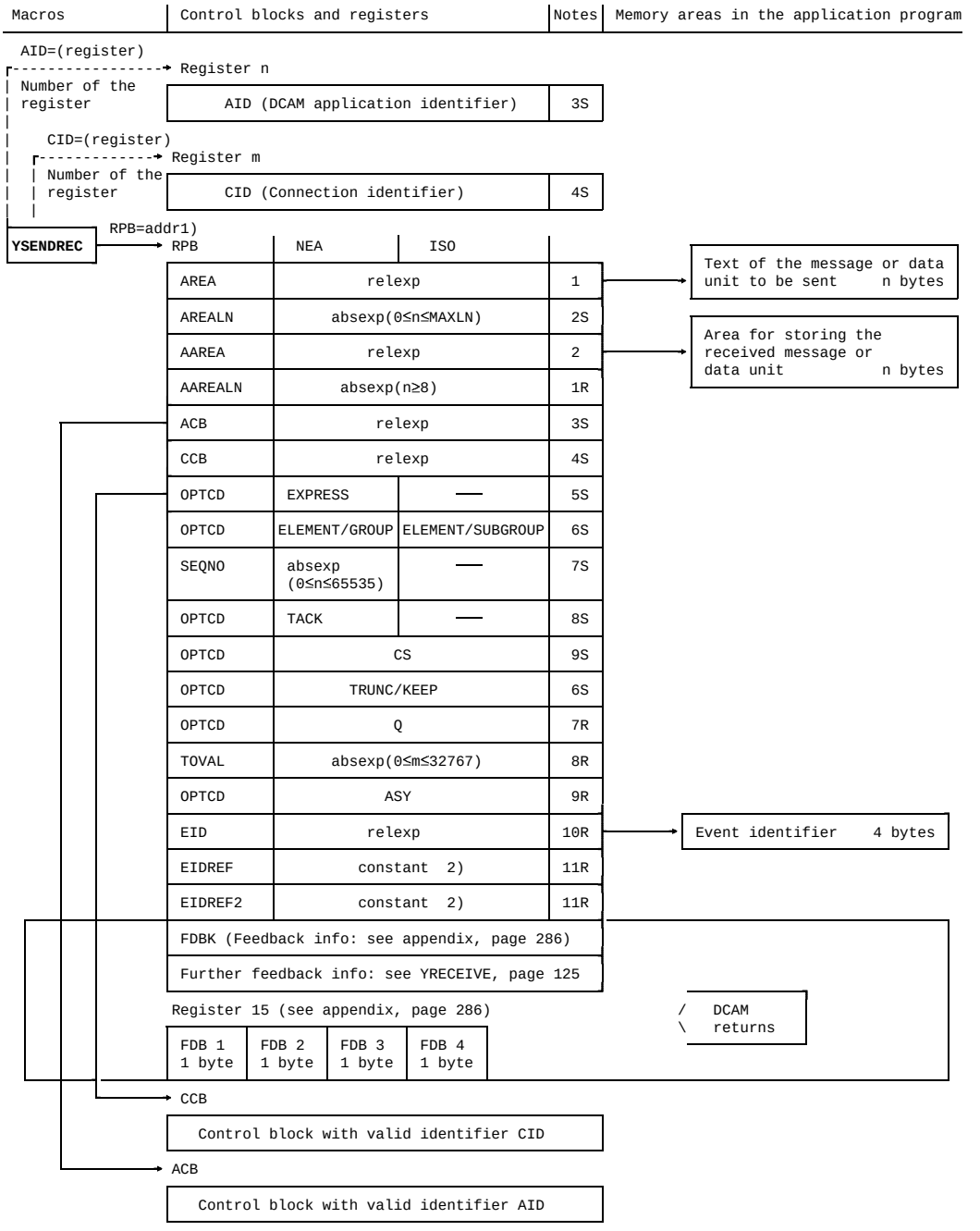
```
                .
                .
                YRECEIVE    RPB=RPBAD
                .
                .
RPBAD           YRPB        ACB=ACBAD,
                            CCB=CCBAD,
                            AAREA=EMPFANG,
                            AAREALN=100,
                            OPTCD=(KEEP,CS)
ACBAD           YACB        ...
CCBAD           YCCB        ...
EMPFANG         DS          CL100
```

Example 2 (asynchronous processing)

A message with a maximum length of 100 bytes is to be received asynchronously over the connection described in the CCB control block with the address CCBAD. If a longer message arrives, the excess-length part is to be discarded. The message is to be placed in the area with the address EMPFANG.

```
                .
                .
                ENAEI       EINAME=EVENT,
                            EIIDRET=IDAD
                .
                .
                YRECEIVE    RPB=RPBNAM
                .
                .
                SOLSIG      EIID=IDAD,
                            COND=IMMED
                .
                .
RPBNAM          YRPB        ACB=ACBNAM,
                            CCB=CCBAD,
                            AAREA=EMPFANG,
                            AAREALN=100,
                            OPTCD=(TRUNC,ASY,Q),
                            EID=IDAD
ACBNAM          YACB        ...
CCBAD           YCCB        ...
EMPFANG         DS          CL100
IDAD            DS          F
```

3.3.3 Combined transmission and reception



```
1) Address of the control block that describes the macro
2) constant may be defined as a character constant: C'constant', a
   hexadecimal constant: X'constant', a fixed-point constant: F'constant'
   or an address constant': A'constant', V(constant).
```

Messages are to be sent and received.

Mandatory specifications:

1) The address of the field containing the data to be transmitted.
2) The address of the field in which the incoming data is to be placed, and the length of this field, which must be at least 8 bytes.

   This macro must not be issued if the originator-oriented queue (CS) was set but another YRECEIVE or YSENDREC which has not yet been completed has already requested switchover to the common receiver queue (CA). This specification is never possible when the common receiver queue is set.

Optional specification:

3) The required YSEND and YRECEIVE specifications (see pages 121 and 125). The fields are marked with S for YSEND and with R for YRECEIVE. The individual operands are described in the sections on the relevant macros.

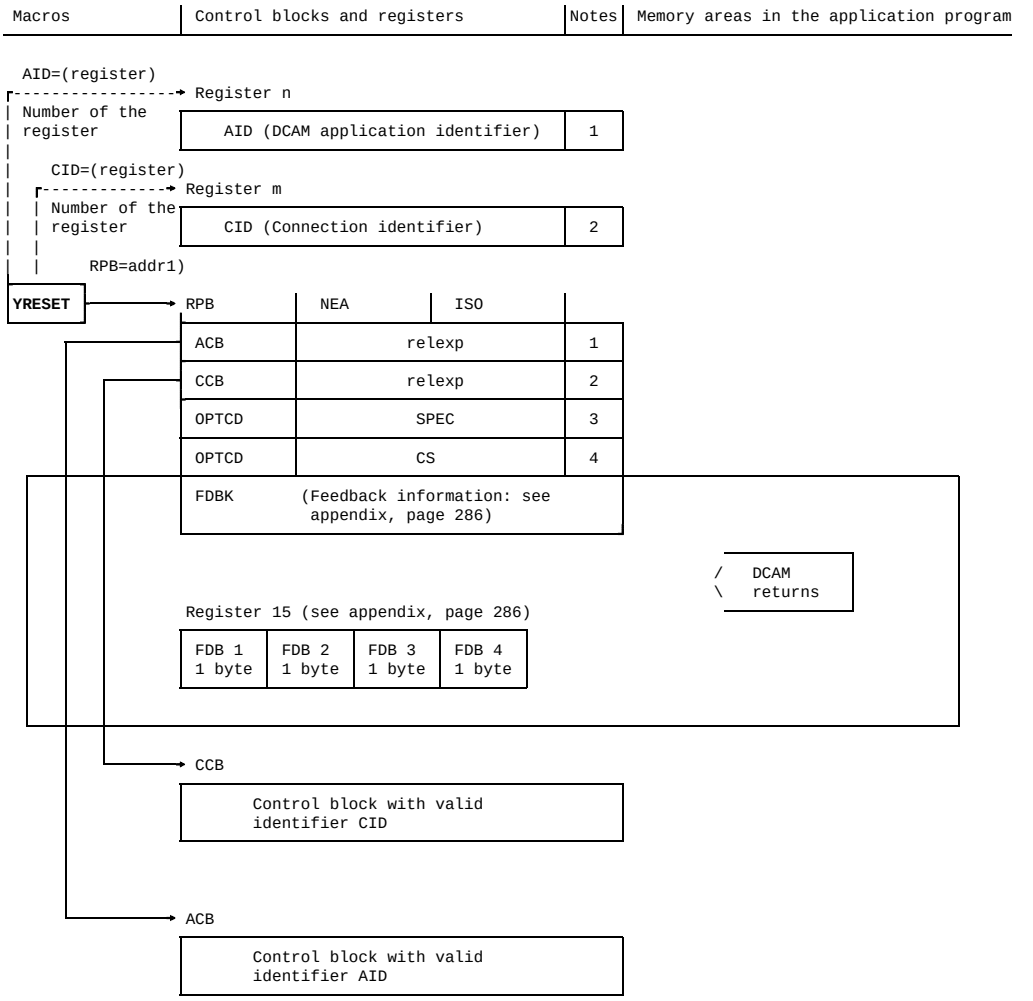
Example

A program contains a loop which is processed until a specified event occurs. The program is then to be continued at the address which was loaded into register 3. During each pass through the loop, a message is issued whose length is contained in register 6.

The program waits for a message with the same length for up to 3 minutes. Before the loop is entered, the identifier of the connection to be serviced is loaded into register. The same connection is always to be serviced during all loop passes. Excess-length messages are ignored and transport acknowledgments are not processed. The macro is to be processed synchronously. Since only one application was opened by the program and the same RPB is always used, the application identifier AID was entered in it at the beginning.

```
                          .
                          .
           LOOP           YSENDREC    RPB=RPBADDR,
                                      CID=(5),
                                      AREA=SENDETXT,
                                      AREALN=(6),
                                      AAREA=EMPFTEXT,
                                      AAREALN=(6),
                                      OPTCD=(SPEC,CS,Q,TRUNC),
                                      TOVAL=180
Event occurred?
        If so:            BR          3
        If not:           B           LOOP
           RPBADDR        YRPB
           SENDETXT       DS          0F
           EMPFTEXT       EQU         *
```

3.3.4 Terminating receive macros and changing the CS/CA state



1) Address of the control block that describes the macro

Receive macros are to be terminated and the CS/CA state is to be changed.

Mandatory specifications:

1) Either the address of the ACB control block containing the valid application identifier (AID) or a register containing the valid identifier.

   This specification may be omitted if the valid identifier was previously entered in the RPB (e.g. with YMODCB).

2) Either the address of the CCB control block containing the valid connection identifier (CID) or a register containing the valid identifier.

   This specification may be omitted if the valid identifier was previously entered in the RPB (e.g. with YMODCB).

   This specification may be omitted if 3) is not specified.

Optional specifications:

3) That all pending YRECEIVE SPEC macros of the connection identified in the CCB are to be canceled. If this specification is omitted, all pending YRECEIVE ANY macros of this task within the application specified in the ACB will be canceled.

4) That messages arriving from this connection after execution of the YRESET SPEC macro are to be placed in the originator-oriented queue.

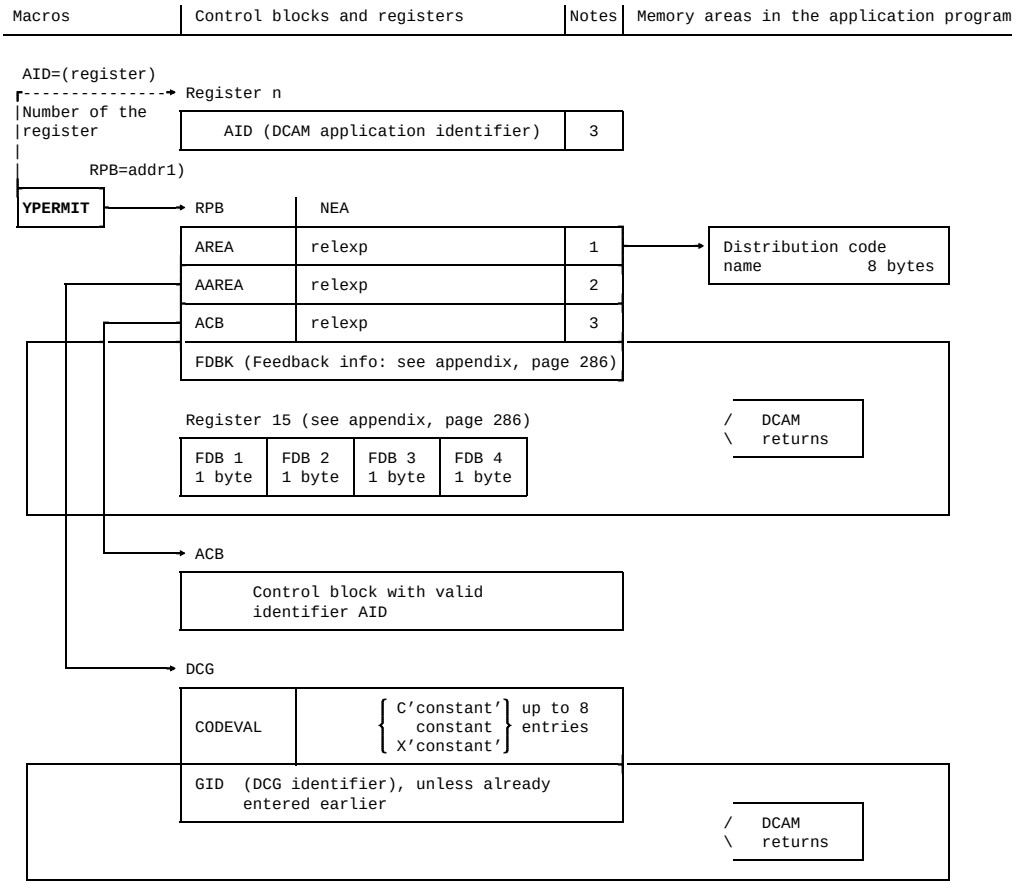
   For DCAM(NEA) transport service applications, this specification is appropriate only when distribution codes are not used (ATTR=NDISCO with YOPEN in the ACB and OPTCD=SPEC in 3).

3.3.5 Control of distribution code assignment

This section applies only to DCAM(NEA) transport service applications.

Two macros are available to the primary task for controlling distribution code assignment. One serves to assign distribution code names to distribution code group (DCG) blocks (YPERMIT) and the other (YFORBID) serves to cancel this assignment without creating a new one.

3.3.5.1 Assigning a distribution code name to a distribution code group block



1) Address of the control block that describes the macro
up to 8 entries

Assignment of distribution codes is to be controlled.

Mandatory specifications:

1) The address of the field containing the distribution code name specified in YOPEN.
2) The address of the distribution code group block (DCG) to be assigned.
3) Either the address of the ACB control block containing the valid application identifier (AID) or a register containing the valid identifier.

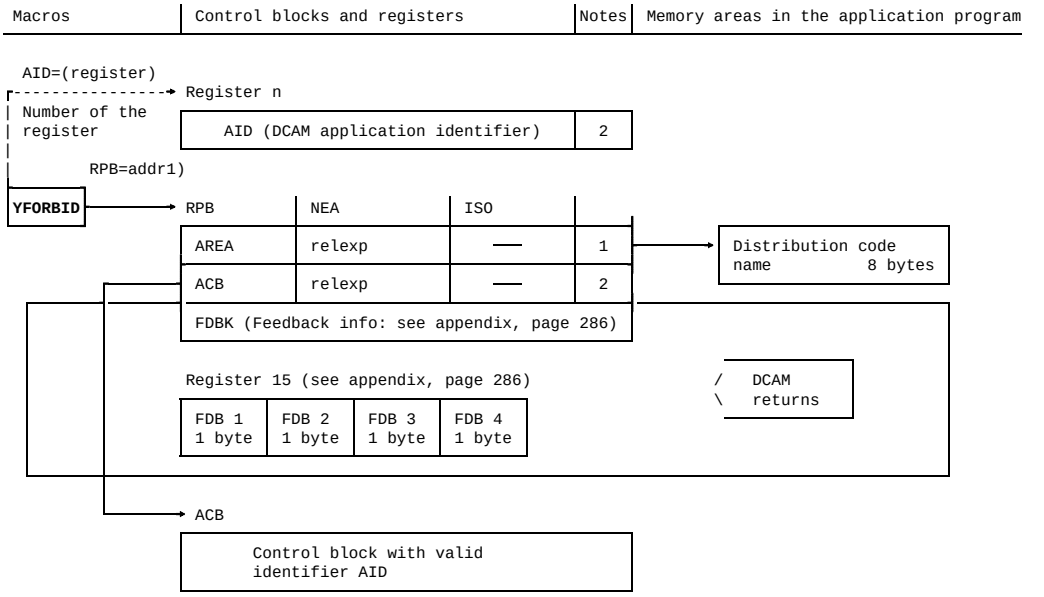
   This specification may be omitted if the valid identifier AID has previously been entered in the RPB (e.g. with YMODCB).

Example

In a DCAM application, the task that specified the distribution code name BETA during opening is to receive messages using the distribution codes described in the distribution code group block (DCG) at address DCGAD.

```
                .
                .
                .
                YPERMIT     RPB=RPBAD
                .
                .
                .
RPBAD           YRPB        ACB=ACBAD,
                            AREA=DISNAM2,
                            AAREA=DCGAD
ACBAD           YACB
DCGAD           YDCG        ...     (Description as defined e.g.
                                     in YOPNCON)
DISNAM2         DC          CL8'BETA'
```

3.3.5.2 Assignment cancellation



1) Address of the control block that describes the macro

The assignment of distribution code names is to be canceled.

Mandatory specifications:

1) The address of the field containing the distribution code name whose assignment is to be canceled.
2) Either the address of the ACB control block containing the valid application identifier (AID) or a register containing the valid identifier.

   This specification may be omitted if the valid identifier has been entered previously in the RPB (e.g. with YMODCB).

Example

The assignment of the distribution code name ALPHA to the associated DCG control block is to be canceled.

```
                .
                .
                YFORBID     RPB=RPBAD
                .
                .
                .
RPBAD           YRPB        ACB=ACBAD,
                            AREA=DISNAM
ACBAD           YACB        ...
DISNAM          DC          CL8'ALPHA'
```

3.4 Name assignment function macros

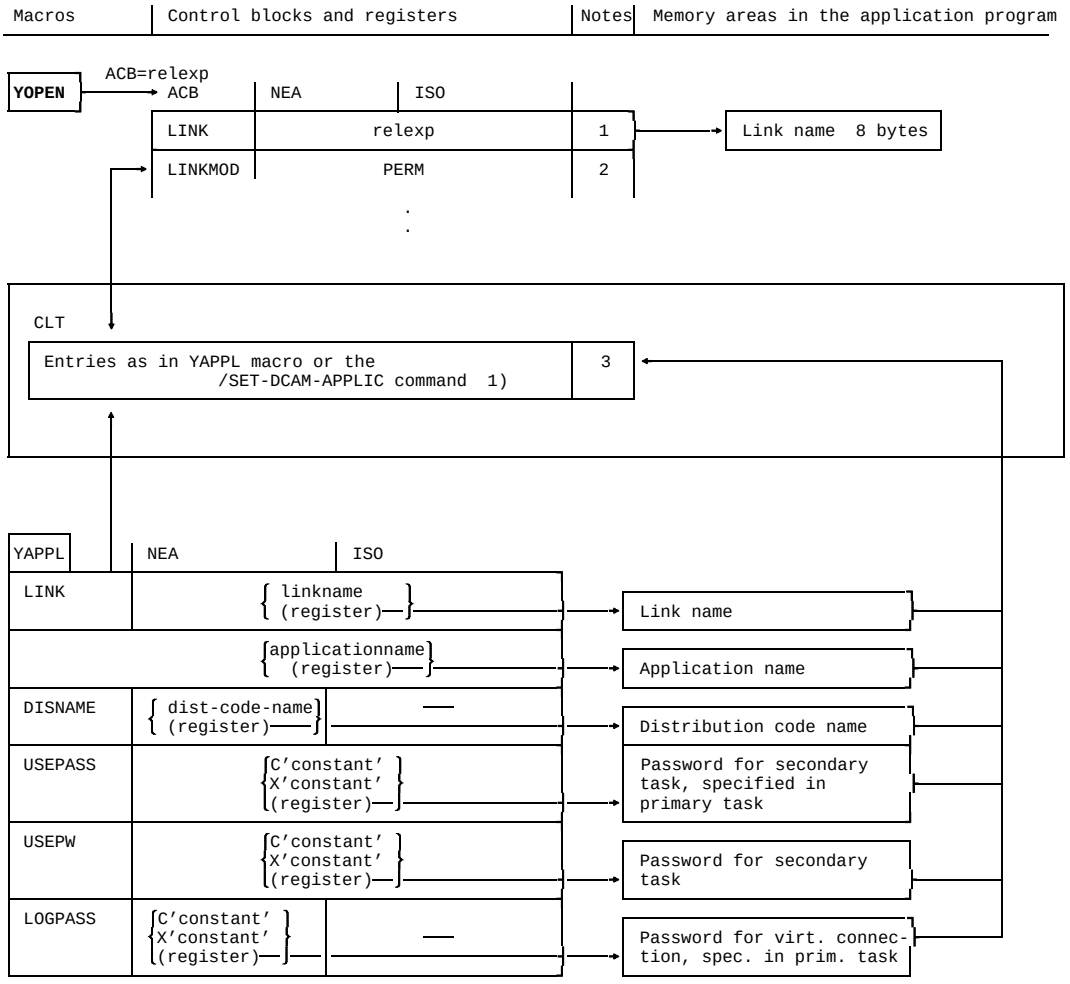
The name assignment function can be used to make programs independent of current operand values.

Two macros are available for making entries in the task-oriented communication link table (CLT):

– YAPPL for entries concerning the DCAM application

– YCONN for entries concerning the virtual connection.

3.4.1 Assignment for the DCAM application



1) see the BS2000 manual "Utility Routines"

The assignment for the DCAM application is to be defined.

Mandatory specification:

1) A link name in the ACB (for further entries refer to the YOPEN description, description, page 46).

Optional specification:

2) That the information of the CLT entries overwrites the data in the ACB. If this specification is omitted, the information is available only during the macro.

Further **mandatory** specifications are:

3) Entries in the CLT, made either with the YAPPL macro (for details refer to the YOPEN description, page 46) or with the /SET-DCAM-APPLICATION-LINK command (see appendix, page 343).

Bear in mind that the YAPPL macro and the /SET-DCAM-APPLICATION-LINK command can be used only when the DCAM subsystem has been loaded successfully. Note, too, that the DCAM subsystem status cannot be HOLD/DELETE when these calls are used. If a task successfully issued a DCAM command or a DCAM call before entering HOLD/DELETE, it can work with DCAM until the task is ended, despite a /HOLD subsystem or /DELETE subsystem (also applicable to %).

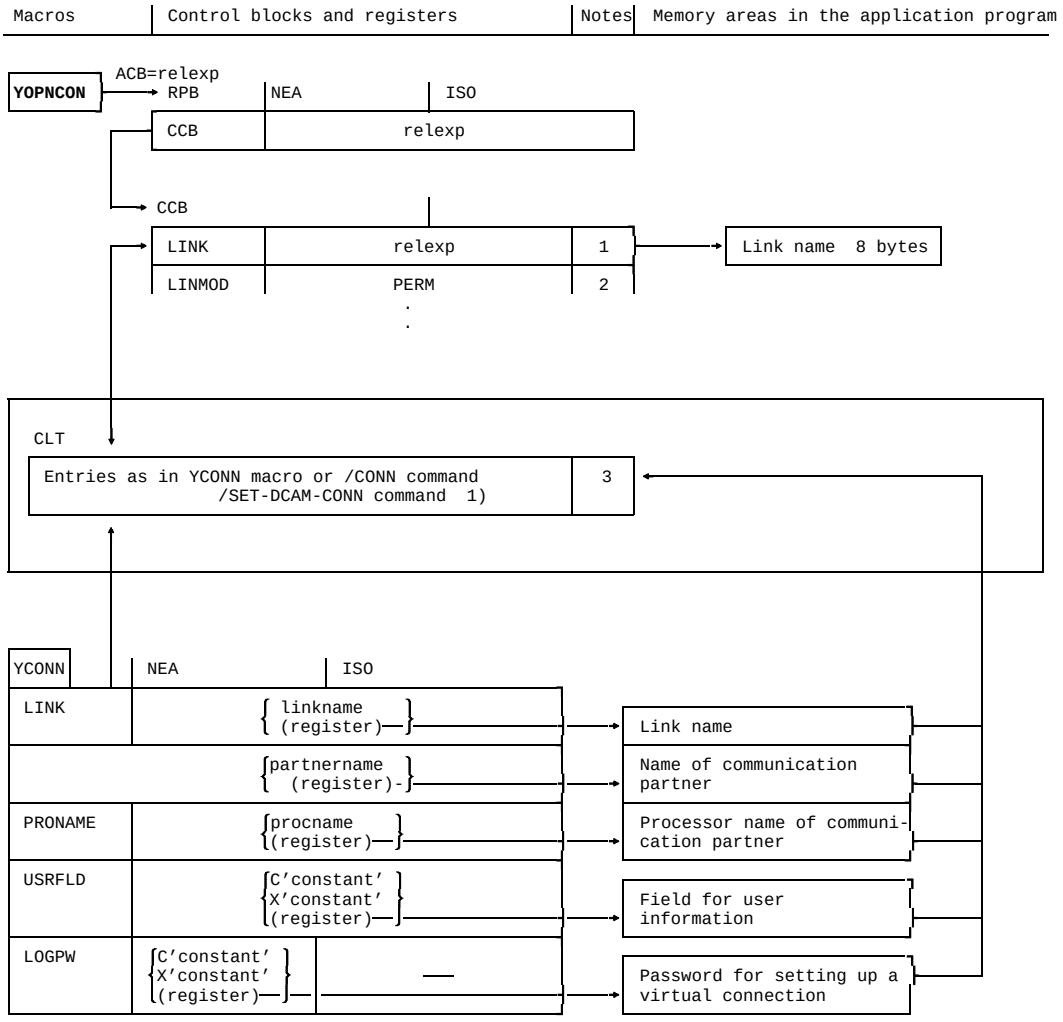
Example

The password for virtual connections is to be updated each time by a separate program before the actual DCAM program is started. SYSDTA is used for input.

Leader program:

```
CURRENT         START
                .
                .
                .
* Read information from SYSDTA:

                RDATA       AREA,ERRADDR
                .
                .
                .
* Entry in the CLT:

                LA          6,USEPASS
                YAPPL       LINK=LINKNAME,
                            USEPASS=(6)
                .
                .
ERRADDR         DS          0Y

* ERROR ROUTINE FOR RDATA
                .
                .
                DS          0F
AREA            DS          0CL84
HEADER          DS          F
USEPASS         DS          80C

DCAM processing program:
                .
                .
                .
                YOPEN       ACB=ACBAD
                .
                .
                .
ACBAD           YACB        USEPASS=C'UNDF',
                            LINK=LINKAD
LINKAD          DC          C'LINKNAME'
```

3.4.2 Assignment for the connection



The assignment for the connection is the be defined.

Mandatory specification:

1) A link name in the CCB (for further entries refer to the YOPNCON description, page 70 through to the chapter 'Macro catalog').

Optional specification:

2) That the information of the CLT entries overwrites the data in the ACB. If this specification is omitted, the information is available only during the macro.

Further **mandatory** specification:

3) Entries in the CLT, made either by means of the YCONN macro (for details of the individual entries, see the YOPNCON macro description, page 70 through to the chapter 'Macro catalog') or with the /SET-DCAM-CONNECTION-LINK command (see appendix, page 343).

Bear in mind that the YCONN macro and the /SET-DCAM-CONNECTION-LINK command can be used only when the DCAM subsystem has been loaded successfully. Note, too, that the DCAM subsystem status cannot be HOLD/DELETE when these calls are used. If a task successfully issued a DCAM command or a DCAM call before entering HOLD/DELETE, it can work with DCAM until the task is ended, despite a /HOLD subsystem or /DELETE subsystem (also applicable to %).

Example

The USERFELD for transport connections is to be specified again by the requesting task for each session. A leader program reads the current password from, for example, SYSDTA.

Leader program:

```
CURRENT         START
                .
                .
                .
* Read information from SYSDTA:

                RDATA       AREA,ERRADDR
                .
                .
                .
* Entry in the CLT:

                LA          6,USERFLD
                YCONN       LINK=LINKNAME,
                            USERFLD=(6)
                .
                .
ERRADDR         DS          0Y

* ERROR ROUTINE FOR RDATA
                .
                .
                DS          0F
AREA            DS          0CL84
HEADER          DS          F
USERFLD         DS          80C

DCAM processing program:
                .
                .
                .
                YOPNCON     RPB=RPBAD
                .
                .
                .
RPBAD           YRPB        CCB=CCBAD,
                            OPTCD=ACQUIRE,...
CCBAD           YCCB        USERFLD=C' USR',
                            LINK=LINKAD,...
LINKAD          DC          C'LINKNAME'
```

4 Macro catalog

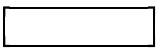
This chapter lists all the DCAM macros and provides details fo their formats.

In the tables, some operands are shown more than once within a macro format, since they may have different applications. However, each operand should be specified only once in a given macro, since correct the function of the macro cannot otherwise be guaranteed. The user should, therefore, place all parameter values for one operand after this operand, rather than specifying them separately.

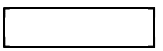
The following table summarizes the macros described and their relationship with the DCAM functions, and contains a short functional description.

For ease of reference, the macros are arranged **alphabetically** in this chapter without regard to their functions.

The shadings used in the macro formats have the following meaning:



These operands apply only to DCAM(ISO) transport service applications.



These operands apply only to DCAM(NEA) transport service applications.

Operands which are not shaded apply to both DCAM(ISO) and DCAM(NEA) transport service applications.

The following layout is used for describing the macros:

Name	Operation	Operands
symbolic name	macro	keyword=operand,...

- The **sequence** of the keyword operands is immaterial.
- Operands enclosed in **brackets** '[ ]' are optional.
- **Braces** '{ }' enclose alternative entries, written one above the other.
- An **underlined operand**, e.g. "LOGON", is used as the default value if no specification is made.
- **Parentheses** '( )' enclose an operand sublist of one or more operands separated by commas. The parentheses may be omitted if the sublist comprises only one operand. The length of the sublist inclusive of the parentheses and commas must not exceed 127 characters.
- **Ellipses** (three periods) after a comma indicate that the preceding operand can be repeated a number of times (keyword=new-value,...).
- **Mnemonic values** (names used in several macros):

  For the following mnemonic values, the same conventions apply as for the Executive Macros (see appendix, page 305):

  - relexp
  - absexp
  - value (=register or absexp)
  - addr (=register or relexp)
  - character
  - symbol
  - integer
  - code
  - (register)

  '(register)' must be replaced by the number of a general register enclosed in parentheses has to be specified.
  All further values occur only once and are therefore explained in the sections on the relevant macros.

Macro	Function	Description
YACB		Generates an application control block
YAPPL	Name assignment	Stores information on the DCAM application in the CLT or deletes this information
YCCB	Connection	Generates a connection control block
YCHANGE	Connection	Modifies the characteristics of an established connection
YCLOSE	Existence	Closes a DCAM application
YCLSCON	Connection	Withdraws a connection request or clears a virtual connection
YCONN	Name assignment	Stores information on the connection in the CLT or deletes such information
YDCG		Generates a distribution code group block for a DCAM(NEA) transport service application
YDIP		Generates a distribution operand block for a DCAM(NEA) transport service application
YENB		Generates an event notification block linking asynchronous notifications with contingency routines
YFORBID	Data com-munication	Cancels the link between a distribution code name and a distribution code group for a DCAM(NEA) transport service application
YGENCB		Generates one or more control blocks of any kind
YINQUIRE	Existence Connection	Retrieves information on applications and virtual connections
YMODCB		Modifies fields in existing control blocks
YOPEN	Existence	Opens a DCAM application
YOPNCON	Connection	Establishes a connection

Macro	Function	Description
YPERMIT	Data communication	Assigns distribution code name to distribution code group for a DCAM(NEA) transport service application
YRECEIVE	Data communication	Receives a message, express message or transport acknowledgment for a DCAM(NEA) transport service application. Receives a message for a DCAM(ISO) transport service application.
YREJLOG	Connection	Rejects a connection request
YRESET	Data communication	Cancels receive macros and changes the CS/CA state of a connection
YRPB		Generates a request parameter block
YSEND	Data communication	Transmits a message or express message for a DCAM(NEA) transport service application. Transmits a message for a DCAM(ISO) transport service application.
YSENDREC	Data communication	Combines transmission of a message or express message with reception of a message, express message or transport acknowledgment for a DCAM(NEA) application. Combines transmission and reception of a message for a DCAM(ISO) transport service application.
YSETLOG	Existence	Modifies the state of a DCAM(NEA) transport service application
YSHOWCB		Transfers individual field contents from a control block to the user area
YTESTCB		Compares the contents of a control block field with a specified value

4.1 YACB

Function

An application control block (ACB) is generated (see also page 11).

Format

Name	Operation	Operands
[symbol]	YACB	ISO={Y / N} [,APPNAME=relexp] [,ATTR=([{SHARE / NSHARE}] [,PRONAME=relexp] [,ENB=relexp] [,LINK=relexp] [,LINKMOD={PERM / TEMP}] [,USEPASS=password1] [,USEPW=password2] [,VERIFY={NO / PRIMARY / SECONDARY}] [,DCAMVER=absexp]
		[,ATTR=([,{DISCO / NDISCO}][,{PRIMTASK / REQTASK / NOTACK}][,{LOGON / NLOGON}])] [,DISNAME=relexp] [,LOGPASS=password3]

Operands

ISO={Y / N}

Y — Mandatory operand for ISO transport-service-compatible applications. The DCAM(NEA) functions are not available for these applications.

N — A DCAM(NEA) transport service application is being used.

This operand is evaluated in YOPEN issued by a primary or secondary task.

APPNAME=relexp

Address of the field containing the name of the DCAM application.

This operand is evaluated in

- YOPEN by a primary task. This operand can be used optionally for ATTR=NSHARE. If no name is specified, a unique name is generated in the host processor in which the task issuing YOPEN exists, without the user being able to learn this name.
- YOPEN by a secondary task.

The area to which the operand refers must satisfy the following conditions:

- The area must be an 8-byte field
- The characters must be left-justified, the first of them being alphabetic (A-Z, @, #, $). In other words, the name must conform to the conventions for ASSEMBLER names.
- '$' in the first byte is reserved for system applications.
- Unused bytes of the field must be filled with blanks (X'40').

ATTR={SHARE | NSHARE}

The DCAM application can either

SHARE be shared by several tasks (shareable) or

NSHARE be used by only one task (non-shareable).

This operand is evaluated in

– YOPEN by a primary task

– YOPEN by a secondary task. In this case ATTR=SHARE is mandatory. If SHARE is not specified, the YOPEN macro is not executed and appropriate feedback information is returned (see appendix, page 286).

PRONAME=relexp

– The address of an 8-byte field in which DCAM enters the symbolic name of its own processor.

If this specification is omitted, the name is not returned.

This operand is evaluated in YOPEN.

ENB=relexp

Address of the event notification block ENB. It links the ACB control block with the ENB control block. The ENB control block contains the addresses of the identifiers of the contingency routines to be activated when specific events are notified (see pages 34 and 187).

The same ENB control block can be referenced in several ACB control blocks.

This operand is evaluated in YOPEN by a primary or secondary task.

LINK=relexp

Address of the field containing the link name (see also page 269).

This operand is evaluated in YOPEN by a primary or secondary task.

The area referenced by this operand must satisfy the following conditions:

– The area must be an 8-byte field.
– The characters must be left-justified.
– The name must be alphanumeric and left-justified, with the first byte being alphabetic (A-Z, @, #, $). In other words the name must conform to the conventions for ASSEMBLER names.
– '$' in the first byte is reserved for system applications.
– Unused bytes of the field must be filled with blanks (X'40').

LINKMOD={PERM / TEMP}

PERM The specifications in the ACB are to be overwritten by the information of the CLT entry identified by the link-name.

The user must specify a field for an application name. DCAM writes the actual application name in this field during YOPEN.

TEMP The information of the CLT entry is available only during the macro call. The specifications in the ACB are not changed.

This operand is evaluated in YOPEN.

USEPASS=password1

Specification of a password for connection of secondary tasks.

Password function: each secondary task must specify this password (USEPW in the ACB control block) when it opens a DCAM application. If, with a password defined, a secondary task omits the password or specifies an invalid when it opens an application, DCAM rejects the YOPEN macro. The secondary task is informed of this in the feedback information (see the appendix, page 286).

Specifying this password is pointless if the application is already protected in the network file RDF; if the password is specified in this case, it must be the same as the password in the RDF.

This operand is evaluated in YOPEN by a primary task, but only if ATTR=SHARE was set.

password1 may be specified as a character constant (C'constant') or a hexadecimal constant (X'constant') with a length of 4 bytes.

The default value is X'00000000', which is equivalent to 'no password defined'.

USEPW=password2

Specification of the password for linkage to a shareable DCAM application.

Password function: The password is entered by the primary task by means of the USEPASS operand or there is an RDF password. Every secondary task opening this DCAM application must specify this password in the USEPW operand. If these is an RDF password, it must also be specified by the primary task.

This operand is only evaluated in

– YOPEN by a primary task USEPW must contain the RDF password (if there is one)

– YOPEN by a secondary task USEPW must contain the RDF password (if there is one), or the password (if any) specified by the primary task by means of USEPASS.

Note

USEPASS and USEPW have the same meaning in the YAPPL or APPLICATION macro. The /APPLICATION command may also be entered in SDF format (see the appendix, page 343)

password2 may be specified as a character constant (C'constant') or a hexadecimal constant (X'constant') with a length of 4 bytes.

The default value is X'00000000', which is equivalent to 'no password defined'.

VERIFY={NO | PRIMARY | SECONDARY}

The task opening the DCAM application is to be

NO	Any task
PRIMARY	The primary task
SECONDARY	A secondary task.

This operand is evaluated in YOPEN by a primary or secondary task.

If the above condition is not satisfied, the macro is not executed and an appropriate notification is returned to the user (see the appendix, page 286).

DCAMVER=absexp

Specification of the DCAM version number.

This operand ensures that a program is compatible as regards the newly added functions of a DCAM version.

If this operand is omitted, a DCAM version ≤ 7.0 is assumed as default.

8.0 This must be specified if the program uses the new functions of DCAM version 8.0. New functions are transfer of a connection notification for YOPNCON with OPTCD=ACCEPT and the specification of the maximum notification length by means of the MAXLN operand in YOPNCON.

Secondary tasks must open an application with the same DCAMVER as the primary task.

This operand is evaluated in YOPEN.

ATTR={DISCO / NDISCO} NEA transport service only

DISCO Distribution of messages to a specific task of the DCAM application is performed by means of a distribution code contained in the message. This specification is meaningful only if ATTR=SHARE is set.

NDISCO Standard message distribution (using the common receiver and the originator-oriented queues).

This operand is evaluated in YOPEN by a primary task.

ATTR={PRIMTASK / REQTASK / NOTACK} NEA transport service only

The acknowledgment for message transport is

PRIMTASK transferred to the primary task;

REQTASK transferred to the task requesting the acknowledgment. This specification is meaningful only with ATTR=SHARE.

NOTACK not transferred (destroyed), even if a positive acknowledgment is requested in the YSEND or the YSENDREC macro (OPTCD=TACK).

This operand is evaluated in YOPEN by a primary task.

ATTR={LOGON / NLOGON} NEA transport service only

Connection requests are

LOGON	processed
NLOGON	not processed.

This operand is evaluated in YOPEN by a primary task.

DISNAME=relexp NEA transport service only

Address of the field containing the distribution code name.

This operand is used with:

- YOPEN by a primary task if ATTR=(SHARE,DISCO) is set;
- YOPEN by a secondary task if the primary task specifies the ATTR=(SHARE,DISCO) operand.

Up to 8 tasks can specify the same distribution code name.

The area to which this operand refers must satisfy the following conditions:

- The area must be an 8-byte field.
- The name must be left-justified and consist of alphanumeric characters, the first byte being alphabetic (A-Z, @, #, $). In other words, the name must conform to the conventions for ASSEMBLER names.
- Unused bytes of the field must be filled with blanks (X'40').

LOGPASS=password3 NEA transport service only

Specification of a password for connection setup.

Password function: each time a connection request is addressed to this DCAM application the requesting partner must specify the password (LOGPW in the CCB control block).
This password cannot be changed during an existing connection. If, when a password is required, the requesting task or the terminal user does not specify the password or specifies a wrong password, DCAM rejects the request and informs the requesting task of the rejection by means of the feedback information (see the appendix, page 286).

This operand is only evaluated in YOPEN by a primary task if ATTR=LOGON was set in the ACB.

password3 may be specified as a character constant (C'constant') or a hexadecimal constant (X'constant') with a length of 4 bytes. The default value is X'00000000', which is equivalent to 'no password defined'.

4.2 YAPPL

Function

The YAPPL macro is used to store or delete information on a DCAM application in the task-oriented communication link table.

This macro uses the name assignment function (see page 139).

Format

Name	Operation	Operands
[symbol]	YAPPL	[{applicationname / (register)}] ,LINK={linkname / (register)} [,USEPASS={password1 / (register)}] [,USEPW={password2 / (register)}]
		[,DISNAME={distributioncodename / (register)}] [,LOGPASS={password3 / (register)}]

Operands

```
{applicationname}
{(register)     }
```

applicationname — specifies the name of the DCAM application.
It may be up to 8 bytes long, must consist of alphanumeric characters, and the first character must be alphabetic (A-Z, @, #, $), i.e. it must conform to the conventions for ASSEMBLER names. '$' as the first character is reserved for system applications, e.g. $TSOS.

(register) — contains the address of an 8-byte field in which the application name is stored left-justified. Unused bytes on the right must be filled with blanks (X'40').

```
LINK={linkname  }
     {(register)}
```

linkname — specifies the name of the link.
It may be up to 8 bytes long, must consist of alphanumeric characters, and the first character must be alphabetic (A-Z, @, #, $), i.e. it must conform to the conventions for ASSEMBLER names.

(register) — contains the address of an 8-byte field in which the link name is stored left-justified. Unused bytes on the right must be filled with blanks (X'40').

If this operand is specified alone, earlier entries in the CLT under the same link name are deleted.

```
USEPASS={password1 }
        {(register)}
```

password1 — is used by a primary task to define the password for linking a secondary task to an application (see also YACB macro). The password is 4 bytes long and consists of a character constant: C'constant' or a hexadecimal constant: X'constant'.

(register) — contains the address of a 4-byte address constant in which the password is stored right-justified.

```
USEPW=⎧password2 ⎫
     ⎨          ⎬
     ⎩(register)⎭
```

password2 specifies a password for linking a secondary task to an application; this password was specified in the secondary task or is the RDF password (see also YACB macro).

The password is 4 bytes long and consists of a character constant: C'constant' or a hexadecimal constant: X'constant'.

(register) contains the address of a 4-byte address constant in which the password is stored right-justified.

```
DISNAME=⎧distributioncodename⎫
        ⎨                    ⎬  NEA transport service only
        ⎩(register)          ⎭
```

distributioncodename

specifies the name under which a distribution code group can be assigned to this task. The name may be up to 8 bytes long, must consist of alphanumeric characters, and the first character must be alphabetic (A-Z, @, #, $), i.e. the name must conform to the conventions for ASSEMBLER names.

(register) contains the address of an 8-byte field in which the distribution code name is stored left-justified. Unused bytes on the right must be filled with blanks (X'40').

```
LOGPASS=⎧password3 ⎫
       ⎨          ⎬        NEA transport service only
       ⎩(register)⎭
```

password3 contains the connection setup password specified in the primary task for use by the communication partners. The password is 4 bytes long and consists of a character constant: C'constant' or a hexadecimal constant: X'constant'.
This password cannot be modified during an existing application.

(register) contains the address of a 4-byte address constant in which the password is stored right-justified.

Feedback

Information returned by DCAM:

Register	Information	Condition
15 (left-justified)	X’00’	Macro was executed as requested.
	X’04’	Macro was not executed because insufficient memory space was available.
	X’08’	Specified linkname was not used.
	X’0C’	Macro could not be executed due to a system error.
	X’24’	DCAM subsystem is not available.

4.3 YCCB

Function

Generates a connection CCB control block (see also page 70).

Format

Name	Operation	Operands
[symbol]	YCCB	[PTNNAME=relexp] [,PRONAME=relexp] [,LINK=relexp] [,LINKMOD={PERM / TEMP}] [,PROC=({TRUNC / KEEP}][,{SIGNAL / NSIGNAL}])] [,PRIO ={3 / 2 / 1}] [,ROUTL=relexp] [,ROUTN=absexp] [,USERFLD=userfield] [,MAXLN=absexp]
		[,MDATA={N / Y}] [,RLTH=absexp] [,DIP=relexp] [,EDIT={USER / SYSTEM}]

Name	Operation	Operands
		[,EDITIN=([{PHYS / LINE / FORM}][,{GETBS / NGETBS}][,{GETFC / NGETFC}][,{LCASE / NLCASE}])]
		[,EDITOUT=([{PHYS / LINE / FORM}][,{HCOPY / NHCOPY}][,{HOM / NHOM}][,{EXTEND / NEXTEND}]
		[,{LOGC / NLOGC}][,{LACK / NLACK}])]
		[,LOGPW=password4]
		[,PROC=([,{SYSCODE / BINARY}][,{APPSTART / ANYSTART}][,{TERMSTAT / NTERMSTAT}])]

Operands

PTNNAME=relexp

Address of the field containing the name of the communication partner.

This operand is evaluated in

- YOPNCON if OPTCD=ACQUIRE or OPTCD=(ACCEPT, SPEC) was set in the RPB control block. If YOPNCON is used with OPTCD=(ACCEPT, ANY), DCAM enters the name in the field.
- YINQUIRE if OPTCD=REQLOGON or =TOPLOGON was set in the RPB control block. DCAM enters the name in the field. If YINQUIRE REQLOGON is specified but a request cannot be found, then DCAM enters X’00’ in the first byte.
- YCHANGE if the CID identifier in the CCB control block is invalid, or if the CCB control block is not yet known to DCAM.

The area referenced by this operand must satisfy the following conditions:

– The area must be an 8-byte field.

– The name must be alphanumeric and left-justified, the first byte being alphabetic (A-Z, @, #, $). In other words the name must conform to the conventions for ASSEMBLER names.

– Unused bytes of the field must be filled with blanks (X'40').

PRONAME=relexp

Address of the field containing the processor name of the communication partner.

This operand has the same format as PTNNAME. X'40'as the first character means "own processor" with YOPNCON ACQUIRE macros.

LINK=relexp

Address of a field containing the link name (see also page 181).

This operand is evaluated in YOPNCON.

The area to which this operand refers must satisfy the following conditions:

– The area must be an 8-byte field.
– The characters must be left-justified.
– The first character must be alphabetic (A-Z, @, #, $).
– Unused bytes in the field must be filled with blanks (X'40').

LINKMOD={PERM | TEMP}

PERM The specifications in the CCB are to be overwritten by the information of the CLT entry identified by the link name.

TEMP The information of the CLT entry is available only during the macro call. The specifications in the CCB are not changed.

This operand is evaluated in YOPNCON, but only if the operand LINK is specified.

PROC={TRUNC | KEEP}

TRUNC — If the message is longer than specified in the YRECEIVE macro, the excess length part is truncated and deleted.

The YRECEIVE macro is executed, but a feedback message (see the appendix, page 286) is issued to indicate the overflow. The length of the remainder of the data is indicated in the ARECLN field (see the appendix, page 279) of the RPB control block.

KEEP — If the message is longer than specified in the YRECEIVE macro, the excess-length part is stored for a following YRECEIVE macro (OPTCD=SPEC).

The YRECEIVE macro is executed, but a feedback message (see the appendix, page 286) is generated to indicate that an overflow occurred and that the data was stored. The length of the remainder of the data is indicated in the ARECLN field (see the appendix, page 279) of the RPB control block.

This operand is evaluated in

- YOPNCON, YCHANGE
- YRECEIVE if OPTCD=CCBTK was set in the RPB and the length of the transmitted message is greater than specified in the AAREALN field.
- YSENDREC if OPTCD=CCBTK was set in the RPB and the length of the transmitted message is greater than the value specified in the AAREALN field.

The specification for data overflow handling made in the YRECEIVE macro (OPTCD=TRUNC or =KEEP in the RPB) takes precedence over the specification made here.

Notes

In non-shareable DCAM applications or in shareable DCAM applications (with ATTR=NDISCO in the ACB for DCAM(NEA) transport service applications), the KEEP function is performed only if the partner is in the CS state (see page 214) or is set to the CS state by the YRECEIVE macro used to receive the first part of the message. The remainder of the message can only be received with a YRECEIVE macro (OPTCD=SPEC in the RPB) issued by the task that caused the CS state. If this task is closed prematurely, the remainder of the message is lost.

In DCAM(NEA) transport service applications with the attribute ATTR=DISCO in the ACB, the remainder of the message can be received with a YRECEIVE macro with OPTCD=SPEC in the RPB issued by any task of the DCAM application. It is the user’s responsibility to ensure that the proper task is serviced.

PROC={SIGNAL | NSIGNAL}

SIGNAL — DCAM controls communication by means of a GO signal: after an overload condition on the connection, the transmitting task is informed by a GO signal that it can continue transmission.

The user must specify a valid EID for the GO signal with each YSEND call; an invalid EID will result in a forced closure of the application when the GO signal arrives.

NSIGNAL — No GO signal is sent after the overload of the connection has been cleared.

This operand is evaluated in YSEND.

PRIO={3 | 2 | 1}

defines the priority of the connection.
PRIO=1 is the highest priority, but this is reserved for system applications. PRIO=3 is the lowest priority.

This operand is evaluated in YOPNCON if OPTCD=ACQUIRE was set in the RPB control block.

ROUTL=relexp

The address of a list containing the names of various connection paths (routes) to the partner. The area to which ROUTL refers must satisfy the following conditions:

- The area must be a contiguous list of 8-byte fields.
- Each field contains one connection name.
- The connection name must be left-justified.
- The first character must be alphabetic (A-Z, @, #, $).
- Unused bytes in the field must be filled with blanks (X’40’).

ROUTN=absexp

The number of routes to a specific partner (see ROUTL). Up to 16 routes may be specified.

USERFLD=benutzerfeld

The user always receives the accompanying information specified in this field when events related to this connection occur or messages arrive.

This operand is evaluated in YOPNCON and YCHANGE.

userfield — may be defined as a character constant: C'constant', a hexadecimal constant: X'constant', a fixed-point constant: F'constant' or an address constant: A'constant', V(constant).

The default value is X'00000000', which means that no accompanying information was defined.

When a macro refers to an RPB control block, the user receives the accompanying information in the USER field of the RPB control block (see the appendix, page 279).

If the user has defined a LOSCON, EXPR or TACK contingency routine (see the chapter 'DCAM ASSEMBLER interface, page 7) and such an event is notified, he receives the content of USERFLD in register 5 of the relevant contingency routine.

MAXLN=absexp

Maximum length of the data (TSDU = transport service data unit) to be transmitted by the DCAM application on this connection.

This value is not passed on to the communication partner. It merely serves to optimize the buffers provided by the system.

DCAM returns the value which has either been accepted or reduced by the system in the MAXLN field of the CCB (see the appendix, page 279). If you specify MAXLN, take care that the value is specified again after each YOPNCONmacro with the same CCB, since the value previously entered by DCAM will otherwise be interpreted as the MAXLN specification.

The following applies to DCAM(NEA) transport service applications:

- In the case of EDIT=USER, a message sent by YSEND corresponds to 1 TSDU.
- In the case of EDIT=SYSTEM and EDITOUT=PHYS or FORM, the physical length of the message must be at least 9 bytes shorter than MAXLN. The user or FHS is responsible for observing the capacity of the device.
- In the case of EDIT=SYSTEM and EDITOUT=LINE, DCAM can send a message edited by this system in sections of minimum size (MAXLN, device capacity).
- During editing, truncation is performed (FDBK: 04 00 2C 00) or editing is rejected (FDBK 18 5C 00 00) if an edited logical record is longer than MAXLN.
- The message sent with an YSEND call must not exceed a maximum of 32767 bytes in length with EDIT=SYSTEM.

Notes

If one CCB is used for several YOPNCON macros, the MAXLN operand must be updated each time, since the value previously entered by DCAM will otherwise be interpreted as the desired MAXLN value.

Note

Edited data is always longer than user data since the control characters are converted and protocol labels are added.

Requested MAXLN	≤ 65530	No specification	
DCAMVER	8.0	8.0	No specification
Result in MAXLN field with an FEP with a DXC	≤ 65530 *)	 4096 4096	 4096 32767

*) The results depend on the HW/SW configuration (see the manual "Generating a Data Communication System").

MDATA={N | Y}

N In DCAM(ISO) transport service applications, the "more-data" function is not used for this connection.

Y In DCAM(ISO) transport service applications, the "more-data" function is used for this connection. This determines how the data units are passed at the own local DCAM (ISO) interface, but says nothing about the division of the "physical" data blocks for the remote transport system.

This operand is evaluated in YOPNCON.

MDATA cannot be changed once a connection has been set up.

More detailed information about the more-data function can be found in the manual "DCAM Program Interfaces".

RLTH=absexp

Maximum expected message length (default: 65535) for DCAM(ISO) transport service applications.
This value is not passed on to the communication partner; it serves merely to optimize the buffers provided by the system. The DCAM(ISO) operand is dependent on MDATA:

– If MDATA=N, the maximum expected message length may be specified in the RLTH operand, thus saving system memory space. In this case, however, RLTH offers no guarantee that longer messages will not occur.

– If MDATA=Y, the maximum possible length for messages to be received is entered after a connection has been established. Any value specified for RLTH is ignored and overwritten. The communication system ensures that no longer messages can occur.

This operand is evaluated in YOPNCON.

RLTH has no effect on the message length in the send direction (dependent on MAXLN).

Further information about the RLTH operand can be found in the manual "DCAM Program Interfaces".

DIP=relexp NEA transport service only

Address of the DIP control block. It contains:

– The location and length of the distribution code in the message transmitted over the connection described by this CCB control block.

– The address of the DCG control block.

This operand is evaluated in

– YOPNCON and
– YCHANGE, but only if the messages are distributed by means of distribution codes.

EDIT={USER | SYSTEM} NEA transport service only

USER means that message editing during input is performed by the user. Virtual terminals are not used. The message is converted to EBCDIC if the PROC=SYSCODE operand is also specified.

Note:

Not for locally connected terminals.

SYSTEM means that the communication system performs message editing either in accordance with the EDITIN or EDITOUT operands set in the CCB control block or in accordance with the parameters specified in VTSUCB (see 'VTSU control block' in the 'VTSU User Guide').

A prerequisite for this is that the message is coded in EBCDIC.

This specification is required if virtual terminals are to be used.

This operand is evaluated in YOPNCON.

EDIT=SYSTEM need not be specified if OPTCD=ACQUIRE is specified. The actual type of message handling is then entered by DCAM.

EDITIN={PHYS / LINE / FORM} NEA transport service only

This operand specifies the type of message editing during input.

PHYS The system restricts message editing to any necessary deblocking and, possibly, to transfer of lowercase letters. The system supports reassembly of blocked messages and, in some cases, transfers lowercase letters. In the case of YRECEIVE, the user area (AAREALN) must be at least large enough to accept the message header.

LINE The system edits messages with the aid of virtual terminals.

Note:

The logical control characters can also be specified with symbolic names (see page 158).

FORM Message editing is performed by the system program 'FHS' or 'FORM' (format terminals).

DCAM evaluates this operand in YOPNCON and YCHANGE, but only if EDIT=SYSTEM was set.

EDITIN={GETBS / NGETBS} NEA transport service only

GETBS The user processes the 'underline' character (X'6D') himself.

NGETBS The system removes all 'underline' characters and the characters to be deleted by this.

This operand is evaluated in YOPNCON and YCHANGE, but only if EDIT=SYSTEM was set.

This function is useful only with communication terminals which use the underline character, e.g. the 8103 Printer Terminal.

EDITIN={GETFC / NGETFC} NEA transport service only

The logical function key code transmitted by the terminal

GETFC is transferred as the first character of the message; (for the values of the function key codes see 'Table of standard function Key code' in the 'VTSU User Guide').

NGETFC is not transferred.

This operand is evaluated in YOPNCON and YCHANGE, but only if EDIT=SYSTEM and EDITIN=LINE were set.

EDITIN={LCASE / NLCASE} NEA transport service only

LCASE The user processes lowercase characters himself.

NLCASE The system converts lowercase letters to uppercase letters.

This operand is evaluated in YOPNCON and YCHANGE, but only if EDIT=SYSTEM was set.

EDITOUT={PHYS / LINE / FORM} NEA transport service only

This operand specifies the type of message editing for output.

PHYS The system does not perform message editing except for transfer of lowercase letters (EDITIN=LCASE).

LINE The system edits messages with the help of virtual terminals.

The logical control characters should be specified with symbolic name (see 'Logical control characters' in the 'VTSU User Guide').

FORM Message editing is performed by the system program 'FHS' or 'FORM' (format terminals).

This operand is evaluated in YOPNCON and YCHANGE, but only if EDIT=SYSTEM was set.

EDITOUT={HCOPY | NHCOPY} NEA transport service only

HCOPY The output message on the data display terminal is also output on a hardcopy device (printer), provided such a device is available and ready.

NHCOPY No hardcopy is printed.

This operand is evaluated in YOPNCON and YCHANGE, but only if

It is used only with EDITOUT=LINE when 8151, 8152, 816x, 975x and 976x Terminals are used.

Note

With EDITOUT=PHYS, HCOPY the user himself is responsible for ensuring the message header is formatted correctly. If the format is incorrect, a negative acknowledgment is issued if a central hardcopy device is used. When a local hardcopy device is used, the hardcopy request is ignored.

EDITOUT={HOM | NHOM} NEA transport service only

The structuring of the message is specified.

HOM An unstructured message is to be output, i.e. all logical lines of a message are treated as a single unit.

NHOM A structured message is to be output, i.e. each individual logical line of the message is a unit.

This operand is evaluated in YOPNCON and YCHANGE, but only if EDIT=SYSTEM and EDITOUT=LINE were set.

EDITOUT={EXTEND | NEXTEND} NEA transport service only

EXTEND The screen is protected as a default. Unprotected fields must be defined by logical control characters and are all transferred to the computer when entered.

NEXTEND The output data is not protected.

This operand is only used if EDITOUT=LINE has been set and if type 975x, 976x, 816x and 3270 terminals are used.

With the exception of BELL, LCASE and GETFC, no EDIT functions are allowed for.

The keys RU, EFZ, AFZ and LSP on the keyboard are disabled.

The control character NL is not permitted in the input with EXTEND (FDBK 04 00 48).

If YSEND is used with EXTEND, DCAM automatically sets EDITOUT=EXTEND for the next YRECEIVE.

This operand is evaluated in YSEND.

EDITOUT={LOGC / NLOGC} NEA transport service only

LOGC All logical control characters of a message are evaluated and converted to device control characters (printer support).
Specific physical control characters are evaluated (see 'Logical control characters' in the 'VTSU User Guide'). Other characters < X'40' are replaced by SUB (smudge).

NLOGC No logical control characters are evaluated. All characters < X'40' in EBCDIC are replaced by SUB (smudge).

This operand is used only if EDITOUT=LINE is set.

EDITOUT={LACK / NLACK} NEA transport service only

Logic acknowledgments are requested by the printer terminal.

LACK Positive and negative logical acknowledgments are requested.

Operand OPTCD=TACK is not affected, i.e. logical acknowledgments and transport acknowledgments may arrive.

The control block field SEQNO in the RPB is used as a reference field for logical acknowledgments and for transport acknowledgments. The two bytes of the SEQNO must be convertible into ASCI code and the following must apply to each of the two bytes:
$20 \leq SEQNO \leq 7F$ (e.g. the printable characters).

NLACK Neither positive nor negative logical acknowledgments arrive.

Positive and negative transport acknowledgments arrive corresponding to the request identifier in the RPB (TACK/NACK).

This operand is used only if EDITOUT=LINE is set.

Logical acknowledgments can be received via macro YRECEIVE in the same way as normal messages.

Structure of a logical acknowledgment:

Byte 0: Logical information
Bytes 1-2: Reference field: SEQNR
Byte 3: Information on printer status

This operand is evaluated in YSEND.

LOGPW=password4 NEA transport service only

Specification of the password for the establishment of a connection.

Function of the password: This password was entered by the communication partner of a DCAM application in the appropriate ACB control block by means of the LOGPASS operand. Every partner requesting connection must specify this password.

This operand is evaluated in YOPNCON with OPTCD=ACQUIRE in the RPB.

password4 may be specified as a character constant: C'constant' or a hexadecimal constant: X'constant' with a length of 4 bytes.

The default value is X'00000000', which means that no password is required.

PROC={SYSCODE | BINARY} NEA transport service only

SYSCODE The user data is coded in EBCDIC, i.e. messages to be transmitted are in EBCDIC and received messages are expected to be in EBCDIC. If the received message is in another code, DCAM forwards the message and terminates the YRECEIVE macro with a warning in the feedback information (see the appendix, page 286).

Data transport in the communication system is not transparent. If required, code conversion to the code used by the communication partner is performed before the data is forwarded (e.g. for output to a terminal).

BINARY The user data is in any code, i.e. messages to be transmitted are in any code and received messages may be in any code.

Data transport in the communication system is transparent. Code conversion is not performed by the system. If the partner is a terminal supported by a procedure other than HDLC, the following rules apply:

– Control characters in the bit patterns are deactivated by DLE characters.

– Deactivation is effective only if the terminal can handle transparent data transmission. Otherwise, the user must ensure that the bit pattern does not contain any undesired control characters.

This operand is evaluated in YOPNCON, YCHANGE, YSEND, YRECEIVE and YSENDREC, but only if EDIT=USER was set.

PROC={APPSTART / ANYSTART} NEA transport service only

APPSTART The DCAM application which requested connection setup or which accepts the request will initiate data transmission. The specification of the accepting partner defeats that of the requesting partner.

This operand is evaluated in YOPNCON.

The partner can obtain information on the proposed version by testing the PROC field in the CCB (see pages 90 and 93).

The requesting partner can obtain information on the final definition by testing the PROC field in the CCB.

PROC={TERMSTAT / NTERMSTAT} NEA transport service only

TERMSTAT Immediately after establishment of a connection to a data display terminal capable of returning status information (such as the 9763), DCAM queries the current terminal partner. When the response to the status query arrives, DCAM signals the message on a YRECEIVE macro with FDB1 = X'00' and FDB3 = X'80'. The status response is processed internally and is not passed on to the program (ARECLN = 0). The information about the terminal status is complete only after arrival of the status response, and it can then be queried with YINQUIRE.
For data display terminals without the status capability, the procedure is simulated and DCAM behaves, from the viewpoint of the application program, in the same manner.
This operand is useful if a program wishes to utilize special features of the data display terminal that are not available on all data display terminals.

NTERMSTAT DCAM does not request a status message.
All message editing is based on the assumption that only the basic configuration of the data display terminal is available.

This operand is evaluated in YOPNCON, but only if EDIT = SYSTEM is set.

4.4 YCHANGE

Function

This macro modifies specific characteristics of an existing connection which are described in the CCB control block (see also page 113).

This macro may be issued by a primary task only.

Format

Name	Operation	Operands
[symbol]	YCHANGE	RPB=addr [,keyword=new-value,...]

Operands

RPB=addr

Address of the RPB control block describing the macro.

keyword=new-value

This operand allows RPB control block values to be updated as with the YMODCB macro.

The following specifications are relevant during the processing of the YCHANGE macro:

keyword	new-value
ACB [1)]	addr
AID [1)]	(register)
CCB	addr
VCBADR [*]	relexp

[1)] Either ACB or AID must be specified. If both ACB and AID are specified and ACB and AID are not compatible, the specification actually evaluated depends on the macro being used.

[*] For DCAM(NEA) transport service applications only

For a description of the individual fields refer to YRPB (page 214) or to the table of control block fields (see the appendix, page 279).

Feedback

The feedback information is given in the functional description (see page 113) and in the table of messages (see the appendix, page 286).

4.5 YCLOSE

Function

The YCLOSE macro closes a DCAM application for a task, i.e. it removes it (see also page 67).

Format

Name	Operation	Operands
[symbol]	YCLOSE	{ACB=addr | AID=(register)}

Operands

ACB=addr

Address of the ACB control block containing the identifier of the DCAM application to be closed.

AID=(register)

Number of the register containing the identifier AID of the DCAM application to be closed.

4.6 YCLSCON

Function

The YCLSCON macro withdraws a previous connection request (YOPNCON ACQUIRE) if the connection is not yet established, or clears down the connection with a communication partner (see also page 116).

YCLSCON may only be issued by a primary task. Data still in system buffers after this macro has been issued cannot be accessed.

Format

Name	Operation	Operands
[symbol]	YCLSCON	RPB=addr [,keyword=new-value,...]

Operands

RPB=addr

Address of the RPB control block describing the macro.

keyword=new-value

This operand allows RPB control block values to be updated as with the YMODCB macro.

The following specifications are relevant during the processing of YCLSCON:

keyword	new-value
ACB [1]	addr
AID [1]	(register)
CCB [2]	addr
CID [2]	(register)

[1] Either ACB or AID must be specified. If both ACB and AID are specified and ACB and AID are not compatible, the specification actually evaluated depends on the macro being used.

[2] Either CCB or AID must be specified. If both CCB and AID are specified and CCB and AID are not compatible, the specification actually evaluated depends on the macro being used.

For a description of the individual fields refer to YRPB (page 214) or to the table of control block fields (see the appendix, page 279).

Feedback

The feedback information is given in the functional description (see page 116) and in the table of messages (see the appendix, page 286).

4.7 YCONN

Function

The YCONN macro is used to store or delete information on a virtual connection in the task-oriented communication link table CLT.

This macro uses the name assignment function (see page 139).

Format

Name	Operation	Operands
[symbol]	YCONN	[{partnername / (register)}] ,LINK={linkname / (register)} [,PRONAME={procname / (register)}] [,USERFLD={userfield / (register)}]
		[,LOGPW={password4 / (register)}]

Operands

```
{partnername }
{            }
{(register)  }
```

partnername — specifies the name of the communication partner.
The name may be up to 8 bytes long, must consist of EBCDIC characters, and the first character must be alphabetic (A-Z, @, #, $), i.e. the name must conform to the conventions for ASSEMBLER names.

(register) — contains the address of an 8-byte field in which the partner name is stored left-justified. Unused bytes to the right must be filled with blanks (X'40').

```
LINK={linkname  }
     {          }
     {(register)}
```

linkname — specifies the link name.
The name may be up to 8 bytes long, must consist of EBCDIC characters, and the first character must be alphabetic (A-Z, @, #), i.e. the name must conform to the conventions for ASSEMBLER names.

(register) — contains the address of an 8-byte field in which the link name is stored left-justified. Unused bytes to the right must be filled with blanks (X'40').

If only this operand is specified, earlier entries made in the CLT under the same link name are deleted.

```
PRONAME={procname  }
        {          }
        {(register)}
```

procname — specifies the processor name of the partner.
The name may be up to 8 bytes long, must consist of EBCDIC characters, and the first character must be alphabetic (A-Z, @, #, $), i.e. the name must conform to the conventions for ASSEMBLER names.

(register) — contains the address of an 8-byte field in which the processor name is stored left-justified. Unused bytes to the right must be filled with blanks (X'40').

USERFLD={userfield / (register)}

userfield — specifies a field in which the user stores accompanying information. This field may be up to 4 bytes long and is a character constant: C'constant' or a hexadecimal constant: X'constant'.

(register) — contains the address of a 4-byte address constant in which the accompanying information is stored right-justified.

LOGPW={password4 / (register)} NEA transport service only

password4 — specifies the connection setup password given by the communication partner issuing the request. The password has a length of 4 bytes and is a character constant: C'constant' or a hexadecimal constant: X'constant'.

(register) — contains the address of a 4-byte address constant in which the password is stored right-justified.

Feedback

Information returned by DCAM:

Register	Information	Condition
15 (left-justified)	X'00'	Macro was executed as required.
	X'04'	Macro was not executed, as there was insufficient memory space available.
	X'08'	Specified link name was not used.
	X'0C'	Macro could not be executed due to a system error.
	X'24'	DCAM subsystem not available.

4.8 YDCG

This section applies only to DCAM(NEA) transport service applications.

Function

A distribution code group block (DCG) is generated. It contains the description of one or more distribution codes (see also pages 84 and 113).

Format

Name	Operation	Operands
[symbol]	YDCG	CODEVAL=(codevalue,...)

Operands

CODEVAL=(codevalue,...)

List of distribution code values. The list may contain up to 8 values.

codevalue is a string of up to 8 characters specified as a

character constant: { C'constant' | constant }

or a hexadecimal constant: X'constant'.

This operand is evaluated in

- YOPNCON if the DCG control block addressed in the DIP control block does not contain a valid identifier GID.
- YCHANGE if the DCG control block addressed in the DIP control block does not contain a valid identifier GID.
- YPERMIT if the DCG control block addressed in AAREA does not contain a valid identifier GID.
- YPERMIT if the DCG contains a valid identifier GID and the operand CODEVAL was modified with YMODCB prior to the granting of authorization.

The distribution code values for a connection must be unique, i.e. the same distribution code value must not be entered in different DCG control blocks addressed by one DIP control block.

Messages containing invalid distribution code values are passed on to the primary task.

4.9 YDIP

This section applies only to DCAM(NEA) transport service applications.

Function

A distribution parameter block DIP is generated. It contains the location and length of the distribution code in a message and references to DCG control blocks (see also pages 84 and 113).

Format

Name	Operation	Operands
[symbol]	YDIP	[CODELN=absexp]
		[,CODEPOS=absexp]
		[,CODEIND=character]
		[,DCG=(relexp,...)]

Operands

CODELN=absexp

Length of the distribution code in bytes. The default value is zero. The distribution code may be up to 8 characters long.

This operand is evaluated in YOPNCON.

CODEPOS=absexp

Displacement of the distribution code relative to the first byte (from the left) of the received data. The first character has a displacement of zero. The default value is zero.

The displacement must not be

– greater than 255

– greater than the total message length (message + distribution code).

This operand is evaluated in YOPNCON.

CODEIND=character

specifies a character which indicates that the following characters (up to 7) are to be interpreted as a distribution code. It can be coded as a 1-byte character constant:

C':constant:' or :constant: or

a hexadecimal constant: X':constant:'.

The default value is X'00'. This means that:

– no implicit distribution code is used;

– an explicit distribution code is expected in every message.

This operand is evaluated in YOPNCON.

Note

The code indicator is located at the point specified in CODEPOS.

DCG=(relexp,...)

List of DCG control block addresses. Up to 16 addresses may be specified.

This operand is evaluated in YOPNCON and YCHANGE.

4.10 YENB

Function

An event notification block ENB is generated. This contains the addresses of the contingency identifiers (see the chapter 'DCAM ASSEMBLER interface', page 7).

Format

Name	Operation	Operands
[symbol]	YENB	[COMEND=relexp] [,LOGON=relexp] [,LOSCON=relexp] [,PROCON=relexp] [,SECOND=relexp]
		[,EXPR=relexp] [,TACK=relexp]

Operands

COMEND=relexp

Address of a field containing the COMEND contingency identifier.

This operand is evaluated in YOPEN.

LOGON=relexp

Address of a field containing the LOGON contingency identifier.

This operand is evaluated in YOPEN by a primary task.

LOSCON=relexp

Address of a field containing the LOSCON contingency identifier.

This operand is evaluated in YOPEN.

PROCON=relexp

Address of a field containing the PROCON contingency identifier.

This operand is evaluated in YOPEN by a primary task.

SECOND=relexp

Address of the field containing the SECOND contingency identifier.

This operand is evaluated in YOPEN by the primary task.

EXPR=relexp NEA transport service only

Address of a field containing the EXPR contingency identifier.

This operand is evaluated in

– YOPEN by a primary task;
– YOPEN by a secondary task with ATTR=NDISCO in the ACB.

TACK=relexp NEA transport service only

Address of a field containing the TACK contingency identifier.

This operand is evaluated in

– YOPEN by a primary task with ATRR=PRIMTASK in the ACB;
– YOPEN by a secondary task with ATTR=REQTASK in the ACB.

Note

The appropriate contingency identifier is returned after the ENACO (enable contingency) macro has been executed (COIDRET operand).

4.11 YFORBID

This section applies only to DCAM(NEA) transport service applications.

Function

This macro cancels the link between a distribution code group as described in a DCG control block and a distribution code name. The task that defined the relevant distribution code name can then receive no more messages (see also page 137).

YFORBID may be issued only by a primary task.

Format

Name	Operation	Operands
[symbol]	YFORBID	RPB=addr [,keyword=new-value,...]

Operands

RPB=addr

Address of the RPB control block describing the macro.

keyword=new-value

This operand permits RPB control block values to be modified in the same manner as with the YMODCB macro.

The following specifications are relevant to the processing of YFORBID:

keyword	new-value
ACB [1)]	addr
AID [1)]	(register)
AREA	addr

1) Either ACB or AID must be specified. If both ACB and AID are specified and ACB and AID are not compatible, the specification actually evaluated depends on the macro being used.

For a description of the individual fields refer to YRPB (page 214) and the table of control block fields (appendix, page 279).

Feedback

The feedback information is given in the functional description (see page 137) and in the table of messages in the appendix (page 286).

4.12 YGENCB

Function

The YGENCB macro generates one or more identical control blocks during program execution. Bear in mind that the YGENCB macro can be used only when the DCAM subsystem has been loaded successfully. Note, too, that the DCAM subsystem status cannot be HOLD/DELETE when this macro is used. If a task successfully issued a DCAM command or a DCAM call before entering HOLD/DELETE, it can work with DCAM until the task is ended, despite a /HOLD subsystem or /DELETE subsystem (also applicable to %).

If a memory area (WAREA) and the length of this area (LENGTH) are specified, the control blocks will be generated at this point in the program (class 6 memory). If no specifications are made, the control blocks are generated in an area managed by DCAM (class 5 memory).

Format

Name	Operation	Operands
[symbol]	YGENCB	BLK={ACB / CCB / ENB / RPB} [,COPIES={1 / integer}] [,WAREA=addr] [,LENGTH=value] [,MF={L / (E,addr)}] ,keyword=initial-value,...
		BLK={DCG / DIP}

Operands

BLK={ACB | CCB | | ENB | RPB}

This operand specifies the type of control block(s) to be generated.

COPIES={1 | integer}

This operand specifies the number of control blocks to be generated. A maximum of 255 control blocks is permissible.

WAREA=addr

Start address of the memory area in the program in which the block(s) is (are) to be generated.

The area must start on a word boundary. When this specification is made, the next one is required too.

If WAREA is not specified, the following applies:
If the SVC instruction generated by the YGENCB macro is below the 16-Mbyte memory limit, the control blocks are also generated below this limit and any program sections running in 24-bit mode can access the control blocks. If not, however, the control blocks will be initialized above 16 Mbytes and hence not accessible in 24-bit mode.

LENGTH=value

specifies the length (in bytes) of the area specified by WAREA. The following minimum lengths are required:

```
ACB   64 bytes
CCB   88 bytes
RPB  100 bytes
ENB   36 bytes
DCG   80 bytes  NEA transport service only
DIP   80 bytes  NEA transport service only
```

Since expansions are expected in future versions, we recommend that larger areas be selected.

MF={L | (E,addr)}

specifies the form of the macro (see page 13).

keyword=initial-value

specifies the initial values of one or more control block fields. The default values listed in the description of the macros for static control block generation are entered into those fields for which no values are specified.

The specifications for "keyword" and "initial-value" are given in the table of control block fields (see the appendix, page 279).

BLK={DCG | DIP} NEA transport service only

This operand specifies the type of control block(s) to be generated.

Feedback

Information returned by DCAM:

Register	Information	Condition
0	(Total) length of the control block(s) in bytes	The control block was generated in dynamically assigned memory area (WAREA and LENGTH operands not set).
	Required length in the user area	The user area is too small (LENGTH operand set).
1	Address of the first byte of the generated control block(s)	The control block was generated in a dynamically assigned memory area.
15	X’00’	Control block generation was successful.
(left-justified)	X’04’	Either the user area was too small (see register 0) or insufficient (dynamically assigned) memory was available.
	X’08’	A mandatory operand is missing.
	X’0C’	The area specified by WAREA does not begin on a word boundary.
	X’14’	The control block field is not a field of the control block specified with BLK.
	X’18’	The address in the MF operand does not refer to an L form parameter list.
	X’1C’	The user area address is invalid.
	X’20’	Invalid operand list or register number.
	X’24’	DCAM subsystem not available.

Symbolic addresses can be generated for the contents of register 15 by using the YDDFDB macro (see the appendix, page 309).

4.13 YINQUIRE

Function

This macro allows a task to request information about a DCAM application (see page 65) or about communication partners and connections (see page 89).

Format

Name	Operation	Operands
[symbol]	YINQUIRE	RPB=addr [,keyword=new-value,...]

Operands

RPB=addr

Address of the RPB control block describing the macro.

keyword=new-value

This operand permits RPB control block values to be modified just as with the YMODCB macro.

The following specifications are relevant during the processing of YINQUIRE:

keyword	new-value
ACB [1)]	addr
AID [1)]	(register)
CCB [2)]	addr
CID [2)]	(register)

1) Either ACB or AID must be specified. If both ACB and AID are specified and ACB and AID are not compatible, the specification actually evaluated depends on the macro being used.

2) Either CCB or AID must be specified. If both CCB and AID are specified and CCB and AID are not compatible, the specification actually evaluated depends on the macro being used.

keyword	new-value
AREA	addr
AREALN	value
AAREA	addr
AAREALN	value
LID	(register)
VCBADR	relexp *
OPTCD	APPSTAT
OPTCD	REQLOGON
OPTCD	TOPLOGON
OPTCD	COUNTPTN
OPTCD	PTNCHAR *
OPTCD	CIDXLATE
OPTCD	NAMXLATE
OPTCD	MONCHARS *
OPTCD	PEROTERM *
OPTCD	BTERMINF *
OPTCD	VTSUCB *

* for DCAM(NEA) transport service applications only

For a description of the individual fields refer to YRPB (page 214) and the table of control block fields (appendix, page 279).

Feedback

The feedback information is given in the functional description (see pages 137 and 89) and in the table of messages in the appendix (page 286).

4.14 YMODCB

Function

This macro allows one or more fields in user area control blocks to be modified (see page 24). The modification does not affect the copies made by DCAM.

The following restrictions apply when YMODCB is used:

– The YMODCB macro can be used only when the DCAM subsystem has been loaded successfully. Note, too, that the DCAM subsystem status cannot be HOLD/DELETE when this macro is used. If a task successfully issued a DCAM command or a DCAM call before entering HOLD/DELETE, it can work with DCAM until the task is ended, despite a /HOLD subsystem or /DELETE subsystem (also applicable to %).

– An RPB control block may not be modified while it is active, e.g. while it is being used for an asynchronous instruction that has not yet been terminated.

– A CCB control block may not be modified while its address is contained in the CCB field of an active RPB control block.

Format

Name	Operation	Operands
[symbol]	YMODCB	BLK={ACB / CCB / ENB / RPB} ,BLKADDR=addr [,MF={L / (E,addr)}] ,keyword=new-value,...
		BLK={DCG / DIP}

Operands

BLK={ACB | CCB | | ENB | RPB}

specifies the type of control block to be modified.

BLKADDR=addr

specifies the address of the control block to be modified.

MF={L | | (E,addr)}

specifies the form of the macro (see page 13).

keyword=new-value

modifies the values in the control block fields. If a "code" type operand comprises several possible specifications (e.g. OPTCD operand), only the specified ones are modified. The specifications for "keyword" and "new value" are given in the table of control field values (see the appendix, page 279).

BLK={DCG | | DIP} NEA transport service only

specifies the type of control block to be modified.

Feedback

Information returned by DCAM:

Register	Information	Condition
15	X’00’	YMODCB was performed successfully.
(left-justified)	X’04’	The location specified by BLKADDR does not contain a valid control block.
	X’08’	A mandatory operand is missing.
	X’0C’	A control block field was to be modified that may not be modified.
	X’10’	The control block addressed by BLKADDR is not of the type specified in BLK.
	X’14’	The control block field is not a field of the specified control block.
	X’18’	The address in the MF operand does not refer to an L form parameter list.
	X’1C’	The control block address is invalid.
	X’20’	RPB/CCB is already being used by an asynchronous macro, or the operand list or register number is invalid.
	X’24’	DCAM subsystem not available.

Symbolic addresses can be generated for the contents of register 15 by using the YDDFDB macro (see the appendix, page 309).

4.15 YOPEN

Function

This macro opens a DCAM application for a task, and generates the DCAM application if it does not yet exist (see also page 46).

Format

Name	Operation	Operand
[symbol]	YOPEN	ACB=addr

Operand

ACB=addr

Address of the ACB control block describing the application to be opened.

Feedback

The feedback information is given in the functional description (see page 46) and in the table of messages in the appendix (see page 286).

4.16 YOPNCON

Function

This macro requests a connection (OPTCD=ACQUIRE; see page 76) or accepts a connection request (OPTCD=ACCEPT: see page 80).

YOPNCON may be issued only by a primary task.

Format

Name	Operation	Operands
[symbol]	YOPNCON	RPB=addr [,keyword=new-value,...]

Operands

RPB=addr

Address of the RPB control block describing the macro.

keyword=new-value

This operand permits the RPB control block to be modified just as with the YMODCB macro.

The following information is relevant during the processing of the YOPNCON macro:

keyword	new-value
ACB [1)]	addr
AID [1)]	(register)
CCB	addr
OPTCD	[{ACCEPT / ACQUIRE}] [,{BELL / NBELL}]* [,{SPEC / ANY}] [,{Q / NQ}] [,{SYN / ASY}] [,{CS / CA}]

1) Either ACB or AID must be specified. If both ACB and AID are specified and ACB and AID are not compatible, the specification actually evaluated depends on the macro being used.

* for DCAM(NEA) transport service applications only

keyword	new-value
EID	addr
EIDREF	{information / (register)}
EIDREF2	{information / (register)}
AREA	addr
AREALN	value
TOVAL	value

For a description of the individual fields refer to YRPB (see page 214) and to the table of the control block fields (see the appendix, page 279).

Restriction

For DCAM applications, one task may issue
- up to 128 YOPNCON (ACQUIRE)
- up to 8 YOPNCON (ACCEPT/ANY)
- up to 8 YOPNCON (ACCEPT/SPEC)

simultaneously per application.

For DCAM(ISO) transport service applications, only one YOPNCON macro may be active to any one partner at any one time (parallel connections may not be established).

Note

If a CCB is used for more than one YOPNCON call, the MAXLN operand of the CCB must be updated every time, as otherwise the value previously entered by DCAM would be interpreted as the desired MAXLN.

Feedback

The feedback information is given in the functional description (see page 70) and in the table of messages in the appendix (page 286).

4.17 YPERMIT

This section applies only to DCAM(NEA) transport service applications.

Function

This macro establishes a link between the distribution code group (as described in a DCG control block) and a distribution code name. The task which has defined the relevant distribution code name will receive, from this point on, all messages containing one of the distribution codes described in the associated DCG control block.

Since up to 8 tasks can define the same distribution code name, this macro does not necessarily establish a unique link between the messages and the task.

If YPERMIT is issued again, an existing link can be replaced by a new one (see also page 135), or a distribution code can be modified.

YPERMIT may be issued only by a primary task.

Format

Name	Operation	Operands
[symbol]	YPERMIT	RPB=addr [,keyword=new-value,...]

Operands

RPB=addr

Address of the RPB control block describing the macro.

keyword=new-value

This operand permits the RPB control block to be modified just as with the YMODCB macro.

The following information is relevant during the processing of the YPERMIT macro:

keyword	new-value
ACB [1)]	addr
AID [1)]	(register)
AAREA	addr
AREA	addr

1) Either ACB or AID must be specified. If both ACB and AID are specified and ACB and AID are not compatible, the specification actually evaluated depends on the macro being used.

For a description of the individual fields refer to YRPB (page 214) and the table of control block fields (appendix, page 279).

Feedback

The feedback information is given in the functional description (see page 135) and in the table of messages in the appendix (page 286).

4.18 YRECEIVE

Function

This macro initiates the transfer (reception) of data to the user memory area. Messages and, in DCAM(NEA) transport service applications, express messages and transport acknowledgments can be received (see also page 125).

Format

Name	Operation	Operands
[symbol]	YRECEIVE	RPB=addr [,keyword=new-value,...]

Operands

RPB=addr

Address of the RPB control block describing the macro.

keyword=new-value

This operand permits the RPB control block to be modified just as with the YMODCB macro.

The following information is relevant during the processing of the YRECEIVE macro:

keyword	new-value
ACB [1]	addr
AID [1]	(register)
CCB [2]	addr
CID [2]	(register)
AAREA	addr
AAREALN	value
OPTCD	[{SPEC / ANY}] [,{Q / NQ}] [,{SYN / ASY}] [,{CS / CA}] [,{TRUNC / KEEP / CCBTK}]

1) Either ACB or AID must be specified. If both ACB and AID are specified and ACB and AID are not compatible, the specification actually evaluated depends on the macro being used.

2) Either CCB or AID must be specified. If both CCB and AID are specified and CCB and AID are not compatible, the specification actually evaluated depends on the macro being used.

keyword	new-value
EID	addr
EIDREF	[information] {(register)}
EIDREF2	[information] {(register)}
TOVAL	value
VCBADR	relexp*

* for DCAM(NEA) transport service applications only

For a description of the individual fields, refer to YRPB (page 214) and to the table of control block fields (appendix, page 279).

Notes

A YRECEIVE with OPTCD=ANY may be issued before a connection has been established, but successful termination requires at least one connection to be established.

If OPTCD=Q is used and OPTCD=SYN, an interrupt may be caused during the wait time by the activation of a contingency routine. If a YRECEIVE with OPTCD=Q is issued in this too, it is rejected.

In the case of block-by-block transmission of an input message from a terminal, each block must be received with a YRECEIVE call.

On reception of the first block of an input message, this connection is set by DCAM to the CS state. This means that the following blocks as far as the last block must be received with OPTCD=SPEC.

The connection remains in CS state until the last block is fetched, i.e. the CS/CA option of the YRECEIVE which fetches the last block is evaluated.

Regardless of reception in block mode, the user must specify the CS option with an YRECEIVE, OPTCD=KEEP.

With block-by-block input at the terminal, GROUP indicates that input is terminated. SUBGROUP/ELEMENT signify that input has not yet been terminated. The next block must be fetched by YRECEIVE. DCAM indicates the data grouping in the FDBK4 of the feedback information (see page 286).

If a task specifies that VTSUCB is to be returned even if the connection does not use VTSUCB, DCAM places an entry in the feedback field FDBK3 (see page 286).

In applications (SHARE/NDISCO) in which messages are distributed via the common receiver queue (CA), it is not possible to predict which task will receive the status information.

In applications (SHARE/DISCO) which use distribution codes, the primary task always receives the status information.

AAREALN must be at least large enough to accept the message header in case of EDITIN=PHYS.

Restriction
A single task can only issue the following simultaneously:

– no more than eight YRECEIVE ANY with OPTCD=ASY per application and task
– no more than eight YRECEIVE SPEC with OPTCD=ASY per connection.

Feedback

The feedback information is given in the functional description (see page 125) and in the table of messages (see the appendix, page 286).

4.19 YREJLOG

Function

This macro rejects a connection request (see also page 111).

It may be issued only in a primary task.

Format

Name	Operation	Operands
[symbol]	YREJLOG	RPB=addr [,keyword=new-value,...]

Operands

RPB=addr

Address of the RPB control block describing the macro.

keyword=new-value

This operand permits the RPB control block to be modified just as with the YMODCB macro.

The following information is relevant during the processing of the YREJLOG macro:

keyword	new-value
ACB [1)]	addr
AID [1)]	(register)
AAREA	addr
AREA	addr

[1)] Either ACB or AID must be specified. If both ACB and AID are specified and ACB and AID are not compatible, the specification actually evaluated depends on the macro being used.

For a description of the individual fields, refer to YRPB (page 214) and to the table of control block fields (appendix, page 279).

Feedback

The feedback information is given in the functional description (see page 111) and in the table of messages (see the appendix, page 286).

4.20 YRESET

Function

Any pending YRECEIVE macros can be canceled. The CS/CA state can be changed for a particular connection.

Format

Name	Operation	Operands
[symbol]	YRESET	RPB=addr [,keyword=new-value,...]

Operands

RPB=addr

Address of the RPB control block describing the macro.

keyword=new-value

This operand permits the RPB control block to be modified just as with the YMODCB macro.

The following information is relevant during the processing of the YRESET macro:

keyword	new-value
ACB [1)]	addr
AID [1)]	(register)
CCB [2)]	addr
CID [2)]	(register)
OPTCD	[{SPEC / ANY}] [,{CS / CA}]

1) Either ACB or AID must be specified. If both ACB and AID are specified and ACB and AID are not compatible, the specification actually evaluated depends on the macro being used.

2) Either CCB or AID must be specified. If both CCB and AID are specified and CCB and AID are not compatible, the specification actually evaluated depends on the macro being used.

For a description of the individual fields, refer to YRPB (page 214) and to the table of control block fields (appendix, page 279).

Feedback

The feedback information is given in the functional description (see page 133) and in the table of messages (see the appendix, page 286).

4.21 YRPB

Function

This macro generates a request parameter block (RPB).

Format

Name	Operation	Operands
[symbol]	YRPB	[AAREA=relexp] [,AAREALN=absexp] [,ACB=relexp] [,AREA=relexp] [,AREALN=absexp] [,CCB=relexp] [,EID=relexp] [,EIDREF=information] [,EIDREF2=information] [,OPTCD=([{ACCEPT | ACQUIRE}][,{ | CS | CA}][,{Q | NQ}][,{REQLOGON | TOPLOGON | NAMXLATE | , | COUNTPTN | APPSTAT | CIDXLATE}] [,{SPEC | , | ANY}][,{SYN | ASY}][,{TRUNC | KEEP | CCBTK}])] [,TOVAL=absexp]
		[,OPTCD=([,{ELEMENT | GROUP}]

Name	Operation	Operands
[symbol]	YRPB	[,OPTCD=([,{BELL / NBELL}][,{ELEMENT / SUBGROUP / GROUP}][,{NORMAL / EXPRESS}] [,{PTNCHAR / MONCHARS / PEROTERM / BTERMINF / VTSUCB}][,{START / STOP}][,{TACK / NTACK}] [,SEQNO=absexp] [,VCBADR=relexp]

Operands

AAREA=relexp

Address of an area in which the user provides data for DCAM and in which DCAM enters data for the user. The information to be provided or entered in this area depends on the macro currently referencing this RPB control block.

The user provides:

- the name of the partner for YINQUIRE with OPTCD=NAMXLATE;
- the address of the area for the connection message in the case of YINQUIRE with OPTCD=REQLOGON;
- the symbolic name of the partner for YREJLOG;
- and, in the case of DCAM(NEA) transport service applications, also the address of the DCG control block for YPERMIT.

DCAM enters:

- the name of the partner for YINQUIRE with OPTCD=CIDXLATE.
- the requested information for YINQUIRE.
- the received message for YRECEIVE and YSENDREC.
- as from DCAMVER 8.0, the connection message of the partner if operand OPTCD=ACQUIRE has been set with YOPNCON.

AAREALN=absexp

Length (in bytes) of the area to which AAREA points.

This may be the length of the

- connection message (up to 80 bytes for DCAM(NEA) transport service applications or up to 32 bytes for DCAM(ISO) transport service applications) for YINQUIRE with OPTCD=REQLOGON, TOPLOGON. It must be borne in mind that connection messages should not be sent if IBM terminals are used, since these are not converted by TRANSIT-CD.
- received data for YRECEIVE and YSENDREC (at least 8 bytes).
- connection message of the partner in the case of YOPNCON with operand OPTCD=ACQUIRE (as from DCAMVER 8.0).

If the length of the received data is greater than the length specified here, the user is given a warning in the FDBK field (see the appendix, page 286) of the RPB control block referenced by the receive macro.

ACB=relexp

Address of an ACB control block that contains the identifier of the application referenced by the macro described here.

This operand is evaluated in YOPNCON, YCLSCON, YREJLOG, YINQUIRE, YCHANGE, YSEND, YRECEIVE , YSENDREC and, for DCAM(NEA) transport service applications, in YSETLOG, YPERMIT and YFORBID.

AREA=relexp

Address of an area in which the user provides data for DCAM and in which DCAM enters data for the user. The data provided or entered in this area depends on the macro currently referencing this RPB control block.

The user provides:

- the connection message (up to 80 bytes for DCAM(NEA) transport service applications or up to 32 bytes for DCAM(ISO) transport service applications) for YOPNCON with OPTCD=ACQUIRE or (as from DCAMVER 8.0) OPTCD=ACCEPT. The AREA operand is evaluated in this case only if the value for AREALN is not equal to O.
- the name of the partner’s processor for YREJLOG and YINQUIRE with the operand OPTCD=NAMXLATE;
- the name of the DCAM application for YINQUIRE with OPTCD=APPSTAT;

– the message to be transmitted for YSEND and YSENDREC;
– and, for DCAM(NEA) transport service applications, also the distribution code name for YPERMIT and YFORBID.

DCAM enters the name of the partner's processor for YINQUIRE with OPTCD=CIDXLATE.

AREALN=absexp

Length of the area referenced by AREA: This may be the length of:

– the connection message for YOPNCON (up to 80 bytes for DCAM(NEA) applications or up to 32 bytes for DCAM(ISO) transport service applications);
– the message to be transmitted for YSEND and YSENDREC. DCAM transmits the data with the length specified here, starting at the address specified in the AREA operand. The maximum length of a message that can be transmitted over a specified connection is given to the user by DCAM in the MAXLN field of the CCB control block.

If AREA contains a name (e.g. a distribution code name), the implied length is 8 bytes, i.e. AREALN is not interpreted.

CCB=relexp

Address of the CCB control block describing the connection to which the macro described here refers.

This operand is evaluated in YOPNCON, YCLSCON, YCHANGE, YSEND, YRECEIVE and YSENDREC. It is also evaluated in YINQUIRE, if OPTCD=REQLOGON/TOPLOGON/PTNCHAR/MONCHARS/PEROTERM or BTERMINF was set.

EID=relexp

Address of the event item identifier returned to the user after an ENAEI with the EIIDRET operand.

The event identified here has occurred if a response was given to a connection request or connection acceptance or if a message or GO signal arrived.

This operand is evaluated in YOPNCON, YRECEIVE, YSEND and YSENDREC.

In YOPNCON, YRECEIVE and YSENDREC this operand is evaluated only if OPTCD=(ASY,Q) was set. In this case, this operand is mandatory (see OPTCD=ASY, OPTCD=Q).

In YSEND this operand is evaluated only if the connection characteristic PROC=SIGNAL has been defined in the CCB, in which case the operand must again be specified (see YCCB, PROC=SIGNAL).

EIDREF=information

This specification is the first word of a user reference which is returned to the user when the event occurs. The first (most significant) byte is then overwritten with X'0C'.

In a synchronous SOLSIG call the user references the area in which the information is to be entered by means of the RPOSTAD operand.

In an asynchronous SOLSIG call the information is passed in register 3 of the associated contingency routine.

This operand is evaluated in YOPNCON, YRECEIVE, YSEND and YSENDREC.

In YOPEN, YRECEIVE and YSENDREC this operand is evaluated only if OPTCD=(ASY,Q) was set. In this case, this operand is mandatory (see OPTCD=ASY, OPTCD=Q).
In YSEND this operand is evaluated only if the connection characteristic PROC=SIGNAL has been defined in the CCB, in which case the operand must again be specified (see YCCB, PROC=SIGNAL).

information may be specified as a character constant: C'constant', a hexadecimal constant: X'constant', a fixed-point constant: F'constant' or an address constant: A(constant) or V(constant).
Character constants and hexadecimal constants are right-justified and padded on the left with X'0'. The default value is X'00000000'. In all cases. the first byte is overwritten with X'0C'.

EIDREF2=information

This specification is the second word of the user reference which is returned to the user when the event has occurred. EIDREF2 is not altered by DCAM.

Note

If EIDREF2 is 4X'00', only **one** user reference is returned.

In a synchronous SOLSIG call the user references the area in which EIDREF and EIDREF2 are to be entered by means of the RPOSTAD and RPOSTL operands. If EIDREF2 is used and RPOSTL=1, an appropriate return code is issued for the SOLSIG call.
In an asynchronous SOLSIG call the second word of the user information is passed in register 4 of the contingency routine.

This operand is evaluated in YOPNCON, YRECEIVE, YSEND and YSENDREC.

In YOPEN, YRECEIVE and YSENDREC this operand is evaluated only if OPTCD=(ASY,Q) was set. In this case, this operand is mandatory (see OPTCD=ASY, OPTCD=Q).
In YSEND this operand is evaluated only if the connection characteristic PROC=SIGNAL has been defined in the CCB, in which case the operand must again be specified (see YCCB, PROC=SIGNAL).

information may be specified as a character constant: C'constant', a hexadecimal constant: X'constant', a fixed-point constant: F'constant' or an address constant: A(constant) or V(constant).
Character constants and hexadecimal constants are right-justified and padded on the left with X'0'. The default value is X'00000000'.

OPTCD={ACCEPT / ACQUIRE}

ACCEPT: A connection request is to be accepted.

ACQUIRE: A connection request is to be issued.

This operand is evaluated in YOPNCON.

OPTCD={CS / CA}

CS After termination of the macro described here, the communication partner is set to the CS state.

CS state means: Messages arriving from the partner can only be received with OPTCD=SPEC because they were entered in the originator-oriented queue. In the case of a shareable DCAM application, only the task that caused the CS state can receive subsequent messages.

CA After the termination of the macro described here, the partner is set to the CA state.

CA state means: Subsequent messages from this partner can only be received with OPTCD=ANY because they were entered in the common receiver queue.

This operand is evaluated in YOPNCON, YSEND, YRESET, YRECEIVE and YSENDREC.

For DCAM(NEA) transport service applications, this operand is interpreted only if distribution codes are not used (ATTR=(SHARE, NDISCO) or NSHARE). This operand is ignored when a transport acknowledgment is received.

YRECEIVE with OPTCD=SPEC is not permitted if the partner is in the CS state and a YRECEIVE with OPTCD=(SPEC,CA) is still pending.

YRECEIVE with OPTCD=SPEC is not permitted if the partner is in the CA state.

YSEND with OPTCD=CA is not permitted if the partner is in the CS state and YRECEIVE macros for the partner are still pending or block-by-block input has not yet been terminated.

YSEND with OPTCD=CS is not permitted if the partner is in the CA state and switchover to the CS state is currently taking place.

Notes

In DCAM(NEA) transport service applications, DCAM can ignore the CS/CA option if block-by-block input is being performed at the terminal.

The communication partner is set to CS state in DCAM(NEA) transport service applications for as long as input is not terminated.

In DCAM(NEA) transport service applications, the CS/CA option of the YRECEIVE is not operative until input is terminated (see page 206).

OPTCD={Q / NQ}

Q — A macro referencing this RPB control block is placed in a queue if it cannot be completed immediately.

This is possible with:

- YRECEIVE if it is necessary to wait for the arrival of data.
- YOPNCON (OBTCD=ACCEPT) if it is necessary to wait for a connection request.

NQ — A macro referencing this RPB control block is not placed in a queue if it cannot be completed immediately.

This operand is evaluated in YOPNCON, YRECEIVE and YSENDREC.

OPTCD=Q is mandatory if OPTCD=ASY is also set.

OPTCD={REQLOGON | TOPLOGON | NAMXLATE | COUNTPTN | APPSTAT | CIDXLATE}

This operand specifies the type of YINQUIRE macro.

REQLOGON — After a LOGON message, information is desired on the partner requesting connection.

TOPLOGON — Information is desired on the first waiting partner requesting connection.

NAMXLATE — The partner and processor names are to be converted to the connection identifier CID.

COUNTPTN — The number of connected partners and the number of partners requesting connection are desired.

APPSTAT — Information is desired on whether an application is open or not and whether it is in the START or STOP state.

CIDXLATE — A connection identifier CID is to be converted to the partner and processor names.

OPTCD={SPEC | ANY}

SPEC — A YOPNCON with OPTCD=ACCEPT or a YRECEIVE or YRESET refers to a specific partner.

ANY — A YOPNCON with OPTCD=ACCEPT or a YRECEIVE or YRESET refers to any partner.

This operand is evaluated in YRECEIVE, YRESET and YSENDREC. It is also evaluated in YOPNCON with OPTCD=ACCEPT if ATTR=LOGON is set in the ACB control block describing the DCAM application.

OPTCD={SYN / ASY}

SYN — Program control is returned to the user after macro termination and execution of the instruction contained in the macro.

ASY — Program control is returned to the user after the acceptance of the macro by DCAM. After execution of the instruction contained in the macro, DCAM issues a user-defined notification for an event item identifier (EID). In this case, OPTCD=Q and EID=relexp must also be set.

This operand is evaluated in YOPNCON, YRECEIVE and YSENDREC.

OPTCD={TRUNC / KEEP / CCBTK}

TRUNC — If the length of the message is greater than that specified for the receive area in AAREALN of the receive macro, the excess-length part of the message is truncated and deleted. The receive macro is terminated successfully, but the data overflow is indicated in the FDBK field (see the appendix, page 286) of the RPB control block to which the receive macro refers. The length of the remaining data is indicated in the ARECLN field (see the appendix, page 279) of the same control block.

KEEP — If the message is longer than that specified for the receive area in the receive macro, the excess-length part is stored for a subsequent YRECEIVE macro (OPTCD=SPEC).

The receive macro is terminated successfully, but the FDBK field (see the appendix, page 286) of the RPB control block to which the receive macro refers indicates that the remaining data has been stored. The length of the remaining data is indicated in the ARECLN field (see the appendix, page 279) of the same control block.

CCBTK — The value (TRUNC or KEEP) specified in the PROC operand of the CCB control block applies.

This operand is evaluated in YRECEIVE and YSENDREC, but only if a data overflow occurs. In case of differences, the specifications made here take precedence over the specifications in the CCB.

In non-shareable DCAM applications or in shareable DCAM applications (with ATTR=NDISCO in the case of DCAM(NEA) transport service applications), the KEEP function is performed only if the partner is in the CS state or is set to the CS state by the receive macro with which the first part of the message is received.

The remainder of the message can only be received with a YRECEIVE (OPTCD=SPEC) macro issued by the task that caused the CS state. If this task is terminated beforehand, the remainder of the message is lost.

In DCAM(NEA) transport service applications with ATTR=DISCO, the remainder of the message can be received with the next YRECEIVE (OPTCD=SPEC) macro issued by any task of the DCAM application.

Note

The OPTCD groups are placed in the RPB independently of each other.

TOVAL=absexp

This operand specifies the time a macro is kept in a queue if it cannot be processed immediately.

The value is specified in seconds.

The default value is 600.

The maximum value is 32767.

Specifying value zero signifies unrestricted wait time.

Monitoring precision is in the order of seconds.

This operand is evaluated in YOPNCON with OPTCD=ACCEPT, YRECEIVE and YSENDREC, but only if OPTCD=Q is also set.

OPTCD={ELEMENT / GROUP}

ELEMENT In DCAM(ISO) transport service applications, the data unit to be transmitted receives the identifier "further data to follow" (= TIDU: Transport Interface Data Unit).

GROUP In DCAM(ISO) transport service applications, the message/data unit to be transmitted receives the identifier "no further data to follow", i.e. it is the last data unit of a message or it is a message which is not passed in individual data units.

This operand is evaluated in YSEND and YSENDREC.

In DCAM(ISO) transport service applications, this operand is evaluated only if the connection uses the more-data function. If the connection was opened with MDATA=N, the data is always transmitted with OPTCD=GROUP, however, it should be noted that the division of the recipient's message is not necessarily identical with the division of the originator's message (see the section "Messages and local data units" in the manual "DCAM Program Interfaces").

OPTCD={BELL / NBELL} NEA transport service only

BELL Output of a message on the terminal is to be indicated by an audible alarm.

NBELL Output is without an audible alarm.

OPTCD={ELEMENT / SUBGROUP / GROUP} NEA transport service only

ELEMENT The data is an element of a group or subgroup of data.

SUBGROUP The data is the last element of a subgroup of data.

GROUP The data is the last element of a group of data.

If the partner is a DCAM application it receives the information in FDB4 of the feedback message (see the appendix, page 286)

- In the case of EDIT=USER, DCAM only ensures that the information on the data stage (block, last block of a message, last block of transmission) is also transmitted. If possible, this specification is entered in the control information for terminals.
- In the case of EDIT=SYSTEM, DCAM performs the task, if necessary, of outputting the message for the user in segments.

This operand is evaluated in YSEND and YSENDREC.

OPTCD={NORMAL / EXPRESS} NEA transport service only

NORMAL The message to be transmitted has normal priority.

EXPRESS The message to be transmitted has express (highest) priority: is to be transported in the communication system with high priority, and is to be passed on immediately to the partner (with an asynchronous EXPR notification, if necessary).

This operand is evaluated in YSEND and YSENDREC, but only if the connection (CCB) was defined with the EDIT=USER. Otherwise all messages have the same priority.

Notes

Express messages should only be sent in exceptional cases. DCAM signals an overload of the connection (wait for go) if an express message is transmitted before a preceding express message has arrived at the system of the communication partner (refer to YCCB: PROC=SIGNAL/NSIGNAL).

Message flow control (wait for go - GO signal) operates independently both for normal messages and for express messages.

OPTCD={PTNCHAR / MONCHARS / PEROTERM / BTERMINF} NEA transport service only

This operand specifies the type of YINQUIRE macro.

PTNCHAR The characteristics of a partner are requested.

MONCHARS Information about the data display terminal and its character sets is requested.

PEROTERM Information about the peripherals of the terminal is requested.

BTERMINF Basic information about the terminal is requested.

This operand is evaluated in YINQUIRE.

OPTCD=VTSUCB NEA transport service only and EDIT=SYSTEM

VTSUCB The VTSU control block is used for message editing on this connection.

This operand may be used together with VCBADR in:

– YCHANGE, to generate, modify (VCBADR=A(VTSUCB)) or delete (VCBADR=0) the internal DCAM copy of VTSUCB;

– YINQUIRE and YRECEIVE, to copy the internal DCAM VTSUCB into the program memory: these macros write the current VTUSCB back to the address specified in the RPB control block, permitting the user to evaluate the feedback information returned in the VTUSCB;

– YSEND and YSENDREC, to generate, delete (VCBADR=0) or modify (VCBADR=A(VTSUCB)) the internal DCAM copy of VTSUCB and then transfer it back to program memory after processing by VTSU.

OPTCD={START / STOP} NEA transport service only

START The DCAM application state is set to START.

STOP The DCAM application state is set to STOP.

This operand is evaluated in YSETLOG, but only if the application was opened with ATTR=LOGON.

The state indicates whether connection requests can be processed (START) or not (STOP).

After YOPEN, and before the first YSETLOG is issued, the state is set to START.

OPTCD={TACK / NTACK} NEA transport service only

TACK A transport acknowledgment is requested for messages transmitted with YSEND or YSENDREC.

NTACK No transport acknowledgment is requested for messages transmitted with YSEND or YSENDREC.

This operand is evaluated in YSEND and YSENDREC.

A transport acknowledgment may be positive or negative. This means:

– If the receiver is a terminal, proper or improper termination of the communication procedure.

– If the receiver is a task, proper or improper reception of a message by this task with the YRECEIVE or YSENDREC macro.

If a TACK contingency routine was defined, the type of transport acknowledgment (positive or negative) is indicated by the contents of register 6.

If no TACK contingency routine was defined, the type of acknowledgment received with YRECEIVE can be found in FDB4 of the feedback field (see the appendix, page 286).

SEQNO=absexp NEA transport service only

The sequence number specified here is assigned to the message to be transmitted. The default value is 0, and the maximum value 32767.

This operand is evaluated in YSEND and YSENDREC.

Notes

The sequence number can also be supplied via a register, if the register number is specified directly in YSEND:

```
YSEND...,SEQNO=(register),...
```

The relationship between a message and the associated transport acknowledgment is established by means of this number. If a TACK contingency routine has been specified, the number is returned in register 7. Otherwise, when the acknowledgment is received, the number can be fetched from the TACKNO field (see the appendix, page 279) of the RPB control block to which the receive macro refers.

The message number is available to the partner in the ASEQNO field (see the appendix, page 279) of the RPB control block to which the receive macro refers. This field contains either the number (DCAM application) specified in SEQNO or a number generated by the communication system if the originator was a terminal.

VCBADR=relexp NEA transport service only and EDIT=SYSTEM

Address of an area in which VTSUCB is passed. If VCBADR^x07ZERO, the user's VTSUCB is copied into the internal DCAM data structure. Message editing is carried out in accordance with the options defined in the VTSUCB.
If VCBADR=ZERO, the internal copy of VTSUCB is deleted. Message editing is then carried out in accordance with the options defined in the CCB.

This operand is evaluated in YCHANGE, YINQUIRE, YSEND, YRECEIVE and YSENDREC with OPTCD=VTSUCB

Note

For unchanging EDIT options, it is sufficient to pass VTSUCB only once to DCAM (e.g. with YCHANGE RPB=, OPTCD=VTSUCB, VCBADR=). However, the EDIT options of VTSUCB may also be modified in each YSEND or queried in each YRECEIVE.

4.22 YSEND

Function

This macro initiates the transfer (transmission) of data from the user area to the data storage of the communication system. Normal and, in DCAM(NEA) transport service applications, express messages can be transmitted (see also page 121).

Format

Name	Operation	Operands
[symbol]	YSEND	RPB=addr [,keyword=new-value,...]

Operands

RPB=addr

Address of the RPB control block describing the macro.

keyword=new-value

This operand may be used to update RPB control block values, just as with the YMODCB macro.

The following specifications are relevant during YSEND processing:

keyword	new-value
ACB [1)]	addr
AID [1)]	(register)
CCB [2)]	addr
CID [2)]	(register)
AREA	addr
AREALN	value
OPTCD	[,{NORMAL / EXPRESS}]* [,{TACK / NTACK}]* [,{CS / CA}] [,{ELEMENT / SUBGROUP / GROUP}]** [,{BELL / NBELL}]* VTSUCB*
SEQNO	value

1) Either ACB or AID must be specified. If both ACB and AID are specified and ACB and AID are not compatible, the specification actually evaluated depends on the macro being used.

2) Either CCB or AID must be specified. If both CCB and AID are specified and CCB and AID are not compatible, the specification actually evaluated depends on the macro being used.

** SUBGROUP applies only to DCAM(NEA) transport service applications

* for DCAM(NEA) transport service applications only

keyword	new-value
EID	addr
EIDREF	{information / (register)}
EIDREF2	{information / (register)}
VCBADR	relexp*

* for DCAM(NEA) transport service applications only

For a description of the individual fields refer to YRPB (page 214) and to the table of control block fields (appendix, page 279).

Notes

If PROC=SIGNAL was specified in YOPNCON for a connection and there is a backlog in the flow of messages on the connection, the feedback information "wait for go" (FDBK 10 04 0C) is returned to the transmitting task for the first YSEND macro it issues. Until the appropriate GO signal (entry for the event item identifier EID) is issued, further YSEND macros issued by this task will for the same connection will result in the feedback information "DCAM: lack of memory space" (FDBK 10 04 08).

Further tasks which attempt to transmit data on an overloaded connection are treated in precisely the same way as the first transmitting task.

In the case of feedback "wait for go", the request parameter block (RPB) is blocked until a GO signal has been issued. DCAM enters the FDBK information in this block for the GO signal (see the appendix, page 286, ’Feedback Information: FDB4’).

As soon as the connection overload has been rectified, a GO signal is issued to each task that has received a "wait for go".

A GO signal does not guarantee that the next YSEND call will be successful.

If a connection or application is closed, a GO signal is transmitted to each task that is waiting for such a signal for the connection in question.

In DCAM(NEA) transport service applications, VTSUCB contains further details of any errors which occur. This information can be obtained with the macro YINQUIRE OPTCD=VTSUCB.

Feedback

The feedback information is given in the functional description (see page 121) and in the table of messages (see the appendix, page 286).

4.23 YSENDREC

Function

This macro requests the combined YSEND and YRECEIVE functions (see also page 130).

Data is transmitted to a specified partner and data is then expected from this partner.

Format

Name	Operation	Operands
[symbol]	YSENDREC	RPB=addr [,keyword=new-value,...]

Operands

RPB=addr

Address of the RPB control block describing the macro.

keyword=new-value

This operand may be used to update RPB control block values, just as with the YMODCB macro.

The following specifications are relevant during YSENDREC processing:

keyword	new-value
ACB [1]	addr
AID [1]	(register)
CCB [2]	addr
CID [2]	(register)
AREA	addr
AREALN	value
AAREA	addr
AAREALN	value
OPTCD	[{Q / NQ}] [,{SYN / ASY}] [,{CS / CA}] [,{TRUNC / KEEP / CCBTK}] [,{NORMAL / EXPRESS}]* [,{TACK / NTACK}]*

1) Either ACB or AID must be specified. If both ACB and AID are specified and ACB and AID are not compatible, the specification actually evaluated depends on the macro being used.

2) Either CCB or AID must be specified. If both CCB and AID are specified and CCB and AID are not compatible, the specification actually evaluated depends on the macro being used.

* for DCAM(NEA) transport service applications only

keyword	new-value
OPTCD	[,{ELEMENT / SUBGROUP / GROUP}]** [,{BELL / NBELL}]* VTSUCB*
EID	addr
EIDREF	{information / (register)}
EIDREF2	{information / (register)}
TOVAL	value
SEQNO	value *
VCBADR	relexp*

** SUBGROUP applies only to DCAM(NEA) transport service applications

* for DCAM(NEA) transport service applications only

For a description of the individual fields refer to YRPB (page 214) and to the table of control block fields (appendix, page 279).

Notes

A YSENDREC corresponds to a YSEND followed by a YRECEIVE on the same connection. This is a way of reducing the number of instructions and saving runtime. The data is received (corresponds to YRECEIVE) only if the send (corresponds to YSEND) returns an RC X’00xxxxxx’ (executed successfully) or RC X’04xxxxxx’ (warning); the RC of YSENDREC then shows the result of the YRECEIVE (also in VTSUCB, if applicable).

Note that the data is received even after "Output truncated" (YSEND-RC=X’04002C00’).

If the send returns neither RC X’00xxxxxx’ nor RC X’04xxxxxx’, the YSENDREC is terminated with a corresponding return code. For example, a "wait for go" may be issued in the event of overload of the connection. Reception of data is then no longer initiated.

Feedback

The feedback information is given in the functional description (see page 130) and in the table of messages (see the appendix, page 286).

4.24 YSETLOG

This section applies only to DCAM(NEA) transport service applications.

Function

This macro causes the state of a DCAM application to be changed. The START state can be changed to the STOP state and the STOP state to the START state (see also page 70).

This macro may be issued only in a primary task.

Format

Name	Operation	Operands
[symbol]	YSETLOG	RPB=addr [,keyword=new-value,...]

Operands

RPB=addr

Address of the RPB control block describing the macro.

keyword=new-value

This operand may be used to update RPB control block values, just as with the YMODCB macro.

The following specifications are relevant during YSETLOG processing:

keyword	new-value
ACB [1]	addr
AID [1]	(register)
OPTCD	[START] {STOP}

[1] Either ACB or AID must be specified. If both ACB and AID are specified and ACB and AID are not compatible, the specification actually evaluated depends on the macro being used.

[2] Either CCB or AID must be specified. If both CCB and AID are specified and CCB and AID are not compatible, the specification actually evaluated depends on the macro being used.

For a description of the individual fields refer to YRPB (page 214) and to the table of control block fields (appendix, page 279).

Feedback

The feedback information is given in the functional description (see page 63) and in the table of messages (see the appendix, page 286).

4.25 YSHOWCB

Function

The YSHOWCB macro transfers the contents of one or more ACB, CCB, DIP, ENB or RPB fields to a user-specified area. Only the fields of one specified control block can be transferred with one YSHOWCB macro (in the order in which they were specified in the FIELDS operand). The required length of the area is given in the table of control block fields. If the user area is too small to store all fields, YSHOWCB places an error code in register 15 and the required length in register 0.

The YSHOWCB macro can be used only when the DCAM subsystem has been loaded successfully. Note, too, that the DCAM subsystem status cannot be HOLD/DELETE when this macro is used. If a task successfully issued a DCAM command or a DCAM call before entering HOLD/DELETE, it can work with DCAM until the task is ended, despite a /HOLD subsystem or /DELETE subsystem (also applicable to %).

Format

Name	Operation	Operands
[symbol]	YSHOWCB	BLK={ACB / CCB / ENB / RPB} ,BLKADDR=addr ,WAREA=addr ,LENGTH=value [,MF={L / (E,addr)}] ,FIELDS={keyword / (keyword,...)}
		BLK={DCG / DIP}

Operands

```
     ⌈ACB⌉
     |CCB|
BLK=⌊   ⌋
     |ENB|
     ⌊RPB⌋
```

specifies the type of control block from which fields are to be read.

BLKADDR=addr

specifies the address of the control block whose fields are to be processed by YSHOWCB.

WAREA=addr

specifies the address of the memory area in the application program in which the control block fields are to be entered.

This area must begin at a word boundary.

LENGTH=value

specifies the length of the area defined by the WAREA operand.

```
    ⌈L       ⌉
MF=⌊(E,addr)⌋
```

specifies the form of the macro (see page 13).

```
        ⌈keyword      ⌉
FIELDS=⌊(keyword,...)⌋
```

specifies one or more control block fields whose contents are to be transferred. The names of the fields can be found in the appropriate table (see the appendix, page 279).

```
     ⌈DCG⌉
BLK=⌊   ⌋                          NEA transport service only
     ⌊DIP⌋
```

specifies the type of control block from which fields are to be read.

Feedback

DCAM returns the following information:

Register	Information	Condition
0	Length of required user memory area	The value specified for LENGTH is too small
15	X’00’	YSHOWCB was performed successfully.
(left-justified)	X’04’	The address specified in BLKADDR is not a control block address.
	X’08’	A mandatory operand is missing.
	X’0C’	Memory area for field contents is too small. The required length is given in register 0.
	X’10’	The control block whose address was specified in BLKADDR is not of the type specified in BLK.
	X’14’	The control block field is not a field of the specified control block.
	X’18’	The address in the MF operand does not refer to an L form parameter list.
	X’1C’	The address of the user area is invalid.
	X’20’	Invalid parameter list or register number.
	X’24’	DCAM subsystem not available.

Symbolic addresses can be generated for the contents of register 15 with the YDDFDB macro (see the appendix, page 309).

4.26 YTESTCB

Function

The YTESTCB macro compares the contents of a specified ACB, CCB, DCG, DIP, ENB or RPB control block field with a test value specified by the user. The YTESTCB performs a logical comparison. The result is made available to the user in register O (if the field value is A and the test value is B, then register O contains O if A equals B, a value less than O if A is less than B, and a value greater than O if A is greater than B).

The YTESTCB macro can be used only when the DCAM subsystem has been loaded successfully. Note, too, that the DCAM subsystem status cannot be HOLD/DELETE when this macro is used. If a task successfully issued a DCAM command or a DCAM call before entering HOLD/DELETE, it can work with DCAM until the task is ended, despite a /HOLD subsystem or /DELETE subsystem (also applicable to %).

Format

Name	Operation	Operands
[symbol]	YTESTCB	BLK={ACB / CCB / ENB / RPB} ,BLKADDR=addr [,ERET=addr] [,MF={L / (E,addr)}] ,keyword=test-value
		BLK={DCG / DIP}

Operands

BLK={ACB / CCB / ENB / RPB}

specifies the type of control block whose field is to be tested.

BLKADDR=addr

specifies the address of the control block.

ERET=addr

specifies the entry address of a user error routine to be executed if the YTESTCB cannot be processed or if YTESTCB processing cannot be completed. In such a case, the contents of register O are meaningless.

MF={L | | (E,addr)}

specifies the form of the macro (see page 13).

keyword=test-value

specifies the control block field (keyword) to be compared with the specified value (test-value).

The names of the control block fields and the test values can be found in the appropriate table (see the appendix, page 279).

BLK={DCG | | DIP} NEA transport service only

specifies the type of control block from whose field is to be tested.

Feedback

DCAM returns the following information:

Register	Information	Condition
O	> 0	The current control block field value is greater than the user-specified value.
	< 0	The current value is less than the user-specified value.
	= 0	The current value is equal to the user-specified value.

Register	Information	Condition
15	X’00’	YTESTCB was performed successfully.
(left-justified)	X’04’	The address specified in BLKADDR is not a control block address.
	X’08’	A mandatory operand is missing.
	X’0C’	Invalid ERET address.
	X’10’	The control block whose address was specified in BLKADDR is not of the type specified in BLK.
	X’14’	The control block field is not a field of the specified control block.
	X’18’	The address of the MF operand does not refer to an L form parameter list.
	X’1C’	The control block address is invalid.
	X’20’	Invalid parameter list or register number.
	X’24’	DCAM subsystem not available.

Symbolic addresses can be generated for the contents of register 15 with the YDDFDB macro (see the appendix, page 309).

5 Examples

5.1 DCAM(NEA) transport service

Example 1

```
       TITLE 'TERMSTA1: OPNCON (ACCEPT,ANY,SYN)(TERMSTAT)'
*-------------------------------------------------------------------
* THIS PROGRAM OPENS THE APPLICATION TERMSTA1 AND WAITS FOR A
* CONNECTION REQUEST FROM ANOTHER TERMINAL.
* AFTER RECEIVING THE STATUS RESPONSE AND CHECKING THE AVAILABLE
* CHARACTER SETS, THE PROGRAM RETURNS THE TERMINAL INPUTS TO THE
* TERMINAL.
* ONLY THE DCAM RETURN CODES WHICH ARE ABSOLUTELY NECESSARY FOR
* PROGRAM EXECUTION ARE CHECKED, BUT ALL DCAM RETURN CODES ARE
* SENT BY THE PROGRAM TO SYSOUT.
*-------------------------------------------------------------------
TERMSTA1 START
         BASR  10,0
         USING *,10
         B     START
* APPNAME,PRONAME  UND PTNNAME PTNPRO
OWNNAME  DC    CL8'TERMSTA1'
OWNPROZ  DS    CL8
PTNNAME  DS    CL8
PTNPROZ  DS    CL8
START    DS    0H
*-------------------------------------------------------------------
*        OPEN THE APPLICATION
*-------------------------------------------------------------------
OPEN     YOPEN ACB=ACB1
         MVI   AKTDIST+1,TOPEN
         BAL   14,FDBKTRC
*-------------------------------------------------------------------
*        ESTABLISH THE CONNECTION
*-------------------------------------------------------------------
OPNCON   YOPNCON RPB=RPB1,TOVAL=45
         MVI   AKTDIST+1,TOPNCON
         BAL   14,FDBKTRC
*-------------------------------------------------------------------
*        WAIT FOR YRECEIVE WITH FDBK 'STATUS MESSAGE RECEIVED'
*-------------------------------------------------------------------
         LA    2,3              STATUS AFTER MAX 3 YRECEIVE'S
WAIT     YRECEIVE RPB=RPB1,AAREALN=L'AAREA,OPTCD=SPEC,TOVAL=30
         MVI   AKTDIST+1,TRECEIVE
```

```
         BAL   14,FDBKTRC
         CLM   15,8,=AL1(YDDFSUCC)
         BNE   WAIT01
         CLM   15,2,=AL1(YDDFTSTA)
         BE    CONTINUE
*  THE TERMINAL USER HAS ENTERED A MESSAGE BEFORE THE STATUS
*  MESSAGE WAS EVALUATED. THIS WAS RECEIVED ALTHOUGH THE STATUS
*  MESSAGE IS MISSING AND COULD BE EVALUATED HERE.
WAIT01   BCT   2,WAIT
*  STATUS MESSAGE NOT RECEIVED: CLOSE THE CONNECTION AGAIN
         B     CLSCON
CONTINUE DS    0H
*-------------------------------------------------------------------
*        DETERMINE THE AVAILABLE CHARACTER SETS
*-------------------------------------------------------------------
         YINQUIRE RPB=RPB1,OPTCD=MONCHARS,AAREA=STATUS,AAREALN=STALENG
         MVI   AKTDIST+1,TINQUIRE
         BAL   14,FDBKTRC
         CLI   STADIM4,'Y'
         BE    DIM02
DIM01    DS    0H
         MVI   AREA,NP           NEW SCREEN PAGE
         B     DIM03
DIM02    DS    0H
         MVC   AREA(8),D27X132   NEW PAGE & DIM 27X132
DIM03    DS    0H
*-------------------------------------------------------------------
*        SEND AND RECEIVE MESSAGES
*-------------------------------------------------------------------
         MVC   AAREA(8),OWNNAME
         MVC   AAREA+9(8),OWNPROZ
         MVC   AAREA+32(8),PTNNAME
         MVC   AAREA+41(8),PTNPROZ
LOOP     DS    0H
*-------------------------------------------------------------------
*        SEND A MESSAGE
*-------------------------------------------------------------------
SEND     YSEND RPB=RPB1,AREA=AREA,AREALN=L'AREA+L'AAREA
         MVI   AKTDIST+1,TSEND
         BAL   14,FDBKTRC
         MVI   AAREA,C' '
         MVC   AAREA+1(L'AAREA-1),AAREA
*-------------------------------------------------------------------
*        RECEIVE AN INPUT MESSAGE
*-------------------------------------------------------------------
RECEIVE  YRECEIVE RPB=RPB1,AAREA=AAREA,AAREALN=L'AAREA,TOVAL=120
         CLM   15,8,=AL1(YDDFSUCC)
         BE    RECOK
         MVC   AAREA(3),=C'END'
RECOK    DS    0Y
         MVI   AKTDIST+1,TRECEIVE
         BAL   14,FDBKTRC
*
         CLC   AAREA(3),=C'END'
         BE    END
         AP    ZAEHLER,EINS
         UNPK  AREA(5),ZAEHLER(3)
         MVC   AREA+3(5),BDPB
```

```
         B     LOOP
*-------------------------------------------------------------------
*        CLEAR DOWN THE CONNECTION
*-------------------------------------------------------------------
END      DS    0H
CLSCON   YCLSCON RPB=RPB1
         MVI   AKTDIST+1,TCLSCON
         BAL   14,FDBKTRC
*-------------------------------------------------------------------
*        CLOSE THE APPLICATION
*-------------------------------------------------------------------
CLOSE    YCLOSE ACB=ACB1
         MVI   AKTDIST+1,TCLOSE
         BAL   14,FDBKTRC
*
TERM     TERM
*-------------------------------------------------------------------
FDBKTRC  DS    0H
         ST    15,YDDFDBK
         LH    8,AKTDIST
         LA    8,MSGTAB(8)
         MVC   MESSTEXT,0(8)
         UNPK  MESSCODE(9),YDDFDBK(5)
         TR    MESSCODE,TRTAB-C'0'
         MVI   MESSEND,HKOMMA
         WROUT MESSAGE,TERM,PARMOD=31
         BR    14
*-------------------------------------------------------------------
AKTDIST  DC    H'0'
MSGTAB   DS    0H
TOPEN    EQU   *-MSGTAB
         DC    CL8'YOPEN'
TCLOSE   EQU   *-MSGTAB
         DC    CL8'YCLOSE'
TOPNCON  EQU   *-MSGTAB
         DC    CL8'YOPNCON'
TCLSCON  EQU   *-MSGTAB
         DC    CL8'YCLSCON'
TINQUIRE EQU   *-MSGTAB
         DC    CL8'YINQUIRE'
TRECEIVE EQU   *-MSGTAB
         DC    CL8'YRECEIVE'
TSEND    EQU   *-MSGTAB
         DC    CL8'YSEND'
MSGTABE  EQU   *
*-------------------------------------------------------------------
YDDFDBK  YDDFDB C
MESSAGE  DC    Y(LMESSAGE)
         DC    CL3' '
MESSTEXT DC    CL8' '
         DC    C' FDBK = X'''
MESSCODE DC    CL8' '
MESSEND  DC    C''''
LMESSAGE EQU   *-MESSAGE
HKOMMA   EQU   C''''
TRTAB    DC    C'0123456789ABCDEF'
```

```
*-------------------------------------------------------------------------------
*         DCAM  CONTROL BLOCKS
*-------------------------------------------------------------------------------
ACB1     YACB  APPNAME=OWNNAME,PRONAME=OWNPROZ,LINK=LNK0,LINKMOD=TEMP
CCB1     YCCB  PTNNAME=PTNNAME,PRONAME=PTNPROZ,LINK=LNK0,LINKMOD=PERM, -
               EDIT=SYSTEM,PROC=TERMSTAT
RPB1     YRPB  ACB=ACB1,CCB=CCB1,                                      -
               AREA=AREA,                                            -
               AAREA=AAREA,                                          -
               OPTCD=(ANY,Q,CS)
*-------------------------------------------------------------------------------
*         CONSTANTS
*-------------------------------------------------------------------------------
         DS    0F
LNK0     DC    CL8'TERMSTA1'
AREA     DC    CL8' '
AAREA    DC    CL50'        /         CONNECTED TO           /'
*
ZAEHLER  DC    PL2'0'
EINS     DC    P'1'
BDPB     DC    C' :   '
D27X132  DS    0CL8
         DC    AL1(NP),AL1(EXT),AL1(DIM)
         DC    C'27132'
*
STATUS   DCSTA C,,TYPE=MONCS
STALENG  EQU  *-STATUS
         VTCSET
         END
```

Example 2

This example demonstrates communication with the system application ’$CONSOLE’ and the use of the YGENCB macro.

The program opens the application ’HUGO’ and attempts, after input of the LOGON MESSAGE, to establish a connection to ’$CONSOLE’.

If this is successful, a STXIT routine is activated to handle break and escape interrupts; in the case of an error, the user is again requested to enter the LOGON MESSAGE.

A synchronous YRECEIVE macro is then issued. If this is completed without errors, the length of the message is fetched with YSHOWCB and the message is displayed on the screen. In the case of a timeout, a new YRECEIVE macro is issued in a loop; for all other errors, the program is terminated.

If the STXIT event occurs, an input is expected from the user. If the user enters ’*END’, the connection is cleared down, the application ’HUGO’ is closed and the program is terminated; otherwise, the user input is sent to ’$CONSOLE’ with YSEND.

After each DCAM macro, an error handling routine is called; in the event of an error, this routine displays the return code and then terminates the program.

```
CONSOLE  START
         PRINT NOGEN
         SPACE 3
         BALR  10,0            BASE REGISTER
         USING *,10
         L     7,FDBCHK        SUBROUTINE FOR CHECKING FDBK
         SPACE 3
*-------------------------------------------------------------------------
*   GENERATE AN ACB
*-------------------------------------------------------------------------
         YGENCB BLK=ACB,MF=(E,YACB0)
         ST    1,AACB
         MVI   SVCIND,DGENACB
         BALR  6,7
         SPACE 3
*-------------------------------------------------------------------------
*   GENERATE A CCB
*-------------------------------------------------------------------------
         YGENCB BLK=CCB,MF=(E,YCCB0)
         ST    1,ACCB
         MVI   SVCIND,DGENCCB
         BALR  6,7
         SPACE 3

*-------------------------------------------------------------------------
*   GENERATE AN RPB FOR YRECEIVE
*-------------------------------------------------------------------------
         YGENCB BLK=RPB,MF=(E,YRPB0)
         ST    1,ARECRPB
         MVI   SVCIND,DGENRPB
         BALR  6,7
```

```
         SPACE 3
*--------------------------------------------------------------------
*   GENERATE AN RPB FOR YSEND
*--------------------------------------------------------------------
         YGENCB BLK=RPB,MF=(E,YRPB0)
         ST     1,ASNDRPB
         BALR   6,7
         TM     @PROGSTA,@ENDPROG
         BNZ    TERM
*--------------------------------------------------------------------
*   OPEN THE APPLICATION 'HUGO    '
*--------------------------------------------------------------------
         L      3,AACB
         YOPEN ACB=(3)
         MVI    SVCIND,DOPEN
         BALR   6,7
         SPACE 3
*--------------------------------------------------------------------
*   SAVE AID IN 'AID0'
*--------------------------------------------------------------------
         YSHOWCB BLK=ACB,BLKADDR=(3),WAREA=AID0,LENGTH=4,FIELDS=(AID)
         MVI    SVCIND,DSHOWCB
         BALR   6,7
         EJECT
         MVC    STARTTYP,STARTIND
RETRY    WROUT STARTMSG,TERM
         MVC    STARTTYP,RETRYIND
         NI     @PROGSTA,255-@NOCONN-@TIMEOUT RUECKSETZEN
*--------------------------------------------------------------------
*   READ THE LOGON MESSAGE
*--------------------------------------------------------------------
         RDATA INBUF,TERM,100,MODE=LINE,ICFD=YES
         CLC    INDATA(4),ENDIND
         BE     CLSAPPL
*--------------------------------------------------------------------
*   ESTABLISH THE CONNECTION TO $CONSOLE
*--------------------------------------------------------------------
         L      2,ARECRPB
         L      3,AID0
         L      4,ACCB
         LH     5,INLEN
         SH     5,CORRLEN
         YOPNCON RPB=(2),AID=(3),CCB=(4),AREALN=(5),TOVAL=30
         MVI    SVCIND,DOPNCON
         BALR   6,7
         TM     @PROGSTA,@ENDPROG
         BNZ    CLSAPPL
         TM     @PROGSTA,@NOCONN+@TIMEOUT
         BNZ    RETRY
*--------------------------------------------------------------------
*   SAVE CID IN 'CID0'
*--------------------------------------------------------------------
         MVI    SVCIND,DSHOWCB
         YSHOWCB BLK=CCB,BLKADDR=(4),WAREA=CID0,LENGTH=4,FIELDS=(CID)
         BALR   6,7
*--------------------------------------------------------------------
*   PLACE AID AND CID IN SEND-RPB
*--------------------------------------------------------------------
```

```
         L     4,CID0
         L     5,ASNDRPB
         YMODCB BLK=RPB,BLKADDR=(5),AID=(3),CID=(4)
         MVI   SVCIND,DMODCB
         BALR  6,7
         EJECT
*-------------------------------------------------------------------
*   DECLARE STXIT ROUTINE
*-------------------------------------------------------------------
         STXIT ESCPBRK=(STX,0)
         EJECT
LOOP     DS    0H
         XC    RECLEN,RECLEN
         NI    @PROGSTA,255-@TIMEOUT  RUECKSETZEN
         SPACE 3
*-------------------------------------------------------------------
*   YRECEIVE FROM '$CONSOLE'
*-------------------------------------------------------------------
         YRECEIVE RPB=(2),TOVAL=600
         MVI   SVCIND,DRECEIV
         BALR  6,7
         TM    @PROGSTA,@NOCONN+@ENDPROG
         BNZ   CLSAPPL
         TM    @PROGSTA,@TIMEOUT
         BNZ   LOOP

*-------------------------------------------------------------------
*   DETERMINE THE LENGTH OF THE RECEIVED DATA
*-------------------------------------------------------------------
         YSHOWCB BLK=RPB,BLKADDR=(2),WAREA=RECLEN,LENGTH=4,            -
               FIELDS=(ARECLN)
         MVI   SVCIND,DSHOWCB
         BALR  6,7
         SPACE 3
*-------------------------------------------------------------------
*   DISPLAY MESSAGE AND ISSUE NEW YRECEIVE
*-------------------------------------------------------------------
         L     8,RECLEN
         LTR   8,8
         BZ    LOOP
         LA    8,5(0,8)
         STH   8,WRLEN
         WROUT RECAREA,TERM
         B     LOOP
         SPACE 3
*-------------------------------------------------------------------
*   CLOSE THE APPLICATION 'HUGO    '
*-------------------------------------------------------------------
CLSAPPL  MVI   SVCIND,DCLOSE
         YCLOSE AID=(3)
         BALR  6,7
*
TERM     TERM
         EJECT
         SPACE 3
*-------------------------------------------------------------------
*   STXIT ROUTINE (ESCPBRK HANDLING)
*-------------------------------------------------------------------
```

```
STX      DS    0H
         BALR  11,0            BASE REGISTER
         USING *,11
         L     7,FDBCHK        ADDRESS OF FDBK EVALUATION
         L     2,ASNDRPB       ADDRESSE OF RPB FOR SENDING
         RDATA INBUF,TERM,100,MODE=LINE
         CLC   INDATA(4),ENDIND
         BE    STX0

*--------------------------------------------------------------------
*   SEND INPUT TOT '$CONSOLE'
*--------------------------------------------------------------------
         LH    5,INLEN
         SH    5,CORRLEN
         YSEND RPB=(2),AREALN=(5)
         MVI   SVCIND,DSEND
         BALR  6,7
         TM    @PROGSTA,@ENDPROG+@NOCONN
         BZ    EXIT
*--------------------------------------------------------------------
*   TERMINATE THE PROGRAM,
*   RESET ANY PENDING YRECEIVE TO ENSURE
*   THAT THE BASE ROUTINE DOES NOT CONTINUE WAITING
*--------------------------------------------------------------------
STX0     OI    @PROGSTA,@ENDPROG
         YRESET RPB=(2)
         MVI   SVCIND,DRESET
         BALR  6,7
*--------------------------------------------------------------------
*   EXIT FROM STXIT ROUTINE
*--------------------------------------------------------------------
EXIT     EXIT
         DROP  11
         EJECT
*--------------------------------------------------------------------
*   CHECKERR: SUBROUTINE WHICH EVALUATES DCAM-FDBK
*--------------------------------------------------------------------
CHECKERR DS 0H
         USING CHECKERR,7
         STCM    15,15,YDDFDB
*--------------------------------------------------------------------
*   IF FDBK = 0 : RETURN
*--------------------------------------------------------------------
         CLI   YDDFFDB1,YDDFSUCC
         BE    CHECKEND
*--------------------------------------------------------------------
*   IF TIMEOUT: SET INDICATOR
*--------------------------------------------------------------------
         CLI   YDDFFDB1,YDDFSUWA
         BNE   CHECK02
         CLI   YDDFFDB3,YDDFTOUT
         BNE   CHECK01
         OI    @PROGSTA,@TIMEOUT
         B     CHECKEND
         SPACE 3

*--------------------------------------------------------------------
*   IF YRECEIVE WAS RESET WITH YRESET: RETURN
```

```
*-------------------------------------------------------------------
CHECK01  CLI   YDDFFDB3,YDDFTRES
         BE    CHECKEND
*-------------------------------------------------------------------
*   OUTPUT THE RETURN CODE
*-------------------------------------------------------------------
CHECK02  DS    0H
         LH    8,SVCDIST
         LA    8,ERRMSG1(8)
         MVC   ERRIND,0(8)
         UNPK  RCODE(9),YDDFDB(5)
         TR    RCODE,TRTAB-C'0'
         WROUT ERRMSG,TERM
CHECK05  DS 0H
*-------------------------------------------------------------------
*   IF WARNING: CONTINUE
*-------------------------------------------------------------------
         CLI   YDDFFDB1,YDDFSUWA
         BE    CHECKEND
*-------------------------------------------------------------------
*   IF NO CONNECTION: SET INDICATOR
*-------------------------------------------------------------------
         CLI   YDDFFDB1,YDDFRPTS
         BNE   CHECK08
         OI    @PROGSTA,@NOCONN
         B     CHECKEND
*-------------------------------------------------------------------
*   ALL OTHER ERRORS: END OF PROGRAM
*-------------------------------------------------------------------
CHECK08  OI    @PROGSTA,@ENDPROG
CHECKEND BR    6
         EJECT
*-------------------------------------------------------------------
*    DATA DECLARATIONS
*-------------------------------------------------------------------
FDBCHK   DC    A(CHECKERR)
AACB     DS    A
ACCB     DS    A
ARECRPB  DS    A
ASNDRPB  DS    A
AID0     DS    A
CID0     DS    A
CORRLEN  DC    H'4'
@PROGSTA DC    XL1'00'          PROGRAM STATUS
@ENDPROG EQU   X'80'
@NOCONN  EQU   X'40'
@TIMEOUT EQU   X'20'
@OK      EQU   X'00'
YDDFDB   YDDFDB C
         SPACE 3
ERRMSG   DC    Y(ERRMSGE-ERRMSG)
         DC    C'   UNRECOVERABLE '
ERRIND   DS    CL6
         DC    C'-ERROR,CODE='
RCODE    DC    C'????????'
ERRMSGE  EQU   *
         SPACE 3
ERRMSG1  DS    0F
```

```
SVCDIST  DS    0H
         DS    X
SVCIND   DS    X
DGENACB  EQU   *-ERRMSG1
         DC    C'GENACB'
DGENCCB  EQU   *-ERRMSG1
         DC    C'GENCCB'
DGENRPB  EQU   *-ERRMSG1
         DC    C'GENRPB'
DMODCB   EQU   *-ERRMSG1
         DC    C'MODRPB'
DOPEN    EQU   *-ERRMSG1
         DC    C'YOPEN '
DSHOWCB  EQU   *-ERRMSG1
         DC    C'SHOWCB'
DOPNCON  EQU   *-ERRMSG1
         DC    C'OPNCON'
DRECEIV  EQU   *-ERRMSG1
         DC    C'RECEIV'
DSEND    EQU   *-ERRMSG1
         DC    C'YSEND '
DRESET   EQU   *-ERRMSG1
         DC    C'YRESET'
DCLOSE   EQU   *-ERRMSG1
         DC    C'YCLOSE'
         SPACE 3
TRTAB    DC    C'0123456789ABCDEF'
         SPACE 3
*
INBUF    DS    0H
INLEN    DS    H
         DC    2C' '
INDATA   DS    XL100
         SPACE 3
RECLEN   DS    F
*
RECAREA  DS    0H
WRLEN    DS    H
         DC    3C' '
RECDATA  DS    XL200
         SPACE 3
STARTMSG DC    Y(STARTMSE-STARTMSG)
         DC    C'   CONSOLE SERVICE: '
STARTTYP DC    CL5' '
         DC    AL1(LOGNL)      NEUE ZEILE
         DC    C'PLEASE ENTER LOGON MESSAGE FOR $CONSOLE OR *END'
STARTMSE EQU   *
         SPACE 3
STARTIND DC    C'START'
RETRYIND DC    C'RETRY'
ENDIND   DC    C'*END'
         SPACE 3
YACB0    YGENCB BLK=ACB,DCAMVER=8.0,ATTR=NLOGON,LINKMOD=PERM,          -
               APPNAME=OWNAPPL,PRONAME=OWNPROC,LINK=LINK,MF=L
YCCB0    YGENCB BLK=CCB,EDIT=SYSTEM,LINK=LINK,LINKMOD=PERM,            -
               PTNNAME=PTNAPPL,PRONAME=PTNPROC,MF=L
YRPB0    YGENCB BLK=RPB,AREA=INDATA,AREALN=(5),AAREA=RECDATA,          -
               AAREALN=L'RECDATA,OPTCD=(ACQUIRE,SYN,Q,SPEC,CS),MF=L
```

```
         SPACE 3
OWNAPPL  DC    C'HUGO    '
OWNPROC  DC    C'        '
PTNPROC  DC    C'        '
PTNAPPL  DC    C'$CONSOLE'
LINK     DC    C'CONSOLE '
         SPACE 3
         VTCSET LOG
         END CONSOLE
```

5.2 DCAM(ISO) transport service

Example 1

This example shows a normal ISO program.

```
         PRINT NOGEN
ISOBS01  START
         TITLE  'TRANSMIT ON AN ISO CONNECTION'
*-------------------------------------------------------------------
* THE PROGRAM GENERATES THE CONTROL BLOCKS ACB, CCB AND RPB
* DYNAMICALLY IN THE PROGRAM MEMORY. THE AREA FOR THE ACB IS USED,
* AFTER THE APPLICATION HAS BEEN OPENED, FOR THE CCB.
* THE DCAM-ISO APPLICATION 'ISOBS01 ' IS OPENED AND WAITS FOR A
* CONNECTION REQUEST FROM 'ISOBS02 ' -> APPLICATION 'ISOBS01 '
* MUST BE OPENED BEFORE THE CONNECTION REQUEST FROM 'ISOBS02 '
* ARRIVES. WITH THE AID OF THE LINK NAME, THE LOCAL AND REMOTE
* NAMES CAN BE CHANGED BEFORE EXECUTION. IF NO /SET-DCAM-CONNECTION-
* LINK IS ISSUED, THE PROGRAM EXPECTS THE PARTNER TO BE ON THE
* LOCAL COMPUTER. AFTER THIS, MESSAGES OF VARIOUS LENGTHS ARE
* SENT TO THE PARTNER.
* THE DCAM RETURN CODES ARE DELIBERATELY NOT EVALUATED; THEY ARE
* SIMPLY LOGGED. THIS MAKES IT EASY TO SEE HOW DCAM REACTS TO
* VARIOUS EVENTS (E.G. NO CONNECTION REQUEST FROM THE PARTNER,
* CLEARING DOWN OF THE CONNECTION, ETC.):
*-------------------------------------------------------------------
         SPACE  3
         BALR  10,0
         USING *,10
         B     ANFANG
*-------------------------------------------------------------------
*        NAMES OF THE COMMUNICATION PARTNERS
*-------------------------------------------------------------------
OWNAPPL  DC    CL8'ISOBS01 '
OWNPROZ  DC    CL8' '
PTNAPPL  DC    CL8'ISOBS02 '
PTNPROZ  DC    CL8' '
LINKNAME DC    CL8'ISOBS01 '
*-------------------------------------------------------------------
ANFANG   DS    0H
*-------------------------------------------------------------------
*              GENERATE AN ACB
*-------------------------------------------------------------------
         LA    5,A#CCB
         YGENCB BLK=ACB,WAREA=(5),LENGTH=L'A#CCB,ISO=Y,ATTR=NSHARE,   -
               APPNAME=OWNAPPL,PRONAME=OWNPROZ,                       -
               LINK=LINKNAME,LINKMOD=PERM
         MVI   AKTIND,DGENCB
         MVC   ZUSATZ,TXTACB
         BAL   14,DUMPRC
*-------------------------------------------------------------------
*              OPEN THE APPLICATION
*-------------------------------------------------------------------
         YOPEN ACB=(5)
         MVI   AKTIND,DOPEN
         MVC   ZUSATZ,OWNAPPL
         BAL   14,DUMPRC
```

```
*-------------------------------------------------------------------
*               SAVE THE AID
*-------------------------------------------------------------------
         YSHOWCB BLK=ACB,BLKADDR=(5),WAREA=AIDSAVE,LENGTH=4,FIELDS=AID
         MVI   AKTIND,DSHOWCB
         MVC   ZUSATZ,TXTACB
         BAL   14,DUMPRC
*-------------------------------------------------------------------
*               GENERATE A CCB (IN THE AREA USED FOR THE ACB)
*-------------------------------------------------------------------
         LA    5,A#CCB
         YGENCB BLK=CCB,WAREA=(5),LENGTH=L'A#CCB,                    -
               PTNNAME=PTNAPPL,PRONAME=PTNPROZ,                      -
               PROC=(TRUNC,SIGNAL),MAXLN=600,MDATA=Y,                -
               LINK=LINKNAME,LINKMOD=PERM
         MVI   AKTIND,DGENCB
         MVC   ZUSATZ,TXTCCB
         BAL   14,DUMPRC
*-------------------------------------------------------------------
*               GENERATE AN RPB
*-------------------------------------------------------------------
         LA    6,RPB
         YGENCB BLK=RPB,WAREA=(6),LENGTH=L'RPB,CCB=(5),              -
               AREA=AREA1,                                           -
               OPTCD=(Q,CS,ACCEPT,SPEC,SYN),TOVAL=60
         MVI   AKTIND,DGENCB
         MVC   ZUSATZ,TXTRPB
         BAL   14,DUMPRC
*-------------------------------------------------------------------
*               GENERATE AN EVENT IDENTIFIER
*-------------------------------------------------------------------
         ENAEI EINAME=ISOBS01,EIIDRET=GOEID,PARMOD=31
         MVI   AKTIND,DENAEI
         BAL   14,DUMPRC

*-------------------------------------------------------------------
*               ESTABLISH THE CONNECTION
*        ALTHOUGH EID AND EIDREF ARE NOT ACTUALLY REQUIRED FOR A
*        SYNCHRNOUS YOPNCON; THEY CAN STILL BE PLACED IN THE RPB
*        AT THIS TIME FOR LATER USE BY YSEND (GO SIGNAL).
*-------------------------------------------------------------------
         L     2,AIDSAVE
         YOPNCON RPB=(6),AID=(2),EID=GOEID,EIDREF2=C'GOGO'
         MVI   AKTIND,DOPNCON
         MVC   ZUSATZ,PTNAPPL
         BAL   14,DUMPRC
*-------------------------------------------------------------------
*        AFTER SUCCESSFUL EXECUTION OF YOPNCON, THE AID AND
*        THE CID ARE STORED IN THE RPB.
*        SEND FOURTEEN MESSAGES TO THE PARTNER.
*-------------------------------------------------------------------
         XR    9,9
         LA    2,14                    SEND 14 MESSAGES
         LA    4,100
         ST    4,SENDLEN
SENDEN   L     4,SENDLEN
         LA    4,50(0,4)               MSG LENGTHS : 150, 200, 250,..
         ST    4,SENDLEN
```

```
         AR    9,4                      R9 = NUMBER OF BYTES SENT
*                                       GENERATE THE MESSAGE
         CVD   4,DW
         UNPK  AREA1,DW
         OI    AREA1+L'AREA1-1,X'F0'
         MVC   AREA1+8(256),AREA1
         MVC   AREA1+264(256),AREA1
         MVC   AREA1+520(256),AREA1
         MVC   AREA1+776(24),AREA1
SEND1    DS    0H                       SEND THE MESSAGE
         YSEND RPB=(6),AREALN=(4)
         MVI   AKTIND,DSEND
         MVC   ZUSATZ,AREA1
         BAL   14,DUMPRC
         CLC   DCAMFDBK(3),WAITGO
         BNE   WEITER
*-------------------------------------------------------------------
*              WAIT FOR THE GO SIGNAL
*-------------------------------------------------------------------
         XC    EIREF,EIREF
         SOLSIG EIID=GOEID,COND=UNCOND,LIFETIM=600,                    -
               RPOSTAD=EIREF,RPOSTL=2,PARMOD=31
         MVI   AKTIND,DSOLSIG
         MVC   ZUSATZ(4),EIREF2
         BAL   14,DUMPRC
*        WITHOUT EVALUATING THE SOLSIG-RTC, IT IS ASSUMED THAT THE
*        GO SIGNAL HAS ARRIVED -> REPEAT THE YSEND
         B     SEND1
WEITER   DS    0H
         BCT   2,SENDEN
*-------------------------------------------------------------------
*        EXCHANGE FINAL MESSAGE
*        AAREALN IS DELIBERATELY TOO SHORT
*-------------------------------------------------------------------
         MVI   AREA1,C'E'        SET END INDICATOR
         YSENDREC RPB=(6),AAREA=AAREA1,AAREALN=8,AREALN=1
         MVI   AKTIND,DSENDREC
         CVD   9,DW
         UNPK  ZUSATZ,DW         NUMBER OF BYTES SENT
         OI    ZUSATZ+L'ZUSATZ-1,X'F0'
         MVC   ZUSATZ(4),AAREA1+4 NUMBER OF BYTES RECEIVED
         BAL   14,DUMPRC
ENDE     DS    0H
*-------------------------------------------------------------------
*              CLEAR DOWN THE CONNECTION
*-------------------------------------------------------------------
         YCLSCON RPB=(6)
         MVI   AKTIND,DCLSCON
         BAL   14,DUMPRC
*-------------------------------------------------------------------
*              CLOSE THE APPLICATION
*-------------------------------------------------------------------
         L     2,AIDSAVE
         YCLOSE AID=(2)
         MVI   AKTIND,DCLOSE
         BAL   14,DUMPRC
TERM     TERM
         EJECT
```

```
*-------------------------------------------------------------------
*  SUBROUTINE: OUTPUT THE RETURN CODE
*-------------------------------------------------------------------
DUMPRC   DS    0H
         ST    15,DCAMFDBK
         LA    8,AKTDIST
         AH    8,AKTDIST
         MVC   MESSTEXT,0(8)           INSERT TEXT
         UNPK  MESSCODE(9),DCAMFDBK(5) CONVERT DCAM FDBK
         MVI   HK,C''''
         TR    MESSCODE,TRTAB-C'0'
         WROUT MESSAGE,TERM,PARMOD=31  SEND FDBK TO SYSOUT
         MVC   ZUSATZ,TXTNIL
         BR    14
***************************** DATA ***********************
TXTNIL   DC    CL8' '
TXTACB   DC    CL8'ACB'
TXTCCB   DC    CL8'CCB'
TXTRPB   DC    CL8'RPB'
MESSAGE  DS    0H                      MESSAGE TEXT
         DC    Y(MESSEND-MESSAGE)
         DC    CL5' '
MESSTEXT DC    CL8' '
         DC    CL1' '
ZUSATZ   DC    CL8' '
         DC    C'  FDBK = X'''
MESSCODE DC    CL8' '
HK       DC    CL2' '
MESSEND  EQU   *
TRTAB    DC    C'0123456789ABCDEF'
AKTDIST  DS    0H
         DS    X
AKTIND   DS    X
DOPEN    EQU   *-AKTDIST
         DC    C'YOPEN   '
DCLOSE   EQU   *-AKTDIST
         DC    C'YCLOSE  '
DOPNCON  EQU   *-AKTDIST
         DC    C'YOPNCON '
DCLSCON  EQU   *-AKTDIST
         DC    C'YCLSCON '
DGENCB   EQU   *-AKTDIST
         DC    C'YGENCB  '
DSHOWCB  EQU   *-AKTDIST
         DC    C'YSHOWCB '
DSEND    EQU   *-AKTDIST
         DC    C'YSEND   '
DSENDREC EQU   *-AKTDIST
         DC    C'YSENDREC'
DENAEI   EQU   *-AKTDIST
         DC    C'ENAEI   '
DSOLSIG  EQU   *-AKTDIST
         DC    C'SOLSIG  '
DCAMFDBK YDDFDB C
         DS    X
         EJECT
DW       DS    D
AIDSAVE  DC    F'0'
```

```
GOEID    DC    F'0'
SENDLEN  DC    F'0'
EIREF    DS    0CL8
         DS    CL4
EIREF2   DS    CL4
WAITGO   DC    AL1(YDDFRDCS),AL1(YDDFSHOR),AL1(YDDFWTGO)
         DS    0F                ALIGNMENT ON WORD BOUNDARY
A#CCB    DS    CL100             AREA FOR ACB AND CCB
         DS    0F                ALIGNMENT ON WORD BOUNDARY
RPB      DS    CL120             AREA FOR RPB
AAREA1   DS    CL32
AREA1    DS    100CL8            SEND AND RECEIVE AREA
AREA1LN  EQU   *-AREA1
         END
```

Example 2

```
         PRINT NOGEN
ISOBS02  START
*-------------------------------------------------------------------------
* THIS PROGRAM GENERATES THE CONTROL BLOCKS ACB, CCB AND RPB
* STATICALLY IN THE PROGRAM MEMORY AND MODIFIES THEM WITH THE AID
* OF DSECTS. THIS MEANS THAT THE PROGRAM MAY HAVE TO BE RECOMPILED
* WHEN A NEWER VERSION OF DCAM IS INTRODUCED. THE PROGRAM OPENS THE
* DCAM-ISO APPLICATION 'ISOBS02 ' AND SENDS A CONNECTION REQUEST
* TO THE PARTNER 'ISOBS01 ' AB. -> APPLICATION 'ISOBS01 ' MUST BE
* OPENED BEFORE THE CONNECTION REQUEST FROM 'ISOBS02' ARRIVES.
* WITH THE AID OF THE LINK NAME, THE LOCAL AND REMOTE NAMES CAN BE
* CHANGED BEFORE EXECUTION. IF NO /SET-DCAM-CONNECTION-LINK IS
* ISSUED, THE PROGRAM EXPECTS THE PARTNER TO BE ON THE LOCAL
* COMPUTER. AFTER THIS, MESSAGES ARE RECEIVED. WHEN THE END
* INDICATOR IS RECEIVED, THE PROGRAM SENDS A MESSAGE AND WAITS
* FOR THE PARTNER TO CLEAR DOWN THE CONNECTION.
* THE FDBK VALUES ARE OUTPUT ON SYSOUT.
*-------------------------------------------------------------------------
         YDDACB D,EQU=N
         EJECT
         YDDCCB D
         EJECT
         YDDRPB D
         EJECT
ISOBS02  CSECT
         BALR  10,0
         USING *,10
         USING YDDACB,5
         USING YDDCCB,6
         USING YDDRPB,7
         LA    5,ACB1
         LA    6,CCB1
         LA    7,RPB1
*-------------------------------------------------------------------------
*              OPEN THE APPLICATION
*-------------------------------------------------------------------------
         YOPEN ACB=(5)
         MVI   AKTIND,DOPEN
         BAL   14,DUMPRC
*-------------------------------------------------------------------------
*              ESTABLISH THE CONNECTION
*-------------------------------------------------------------------------
         L     2,YDDAAID
         YOPNCON RPB=(7),AID=(2)
         MVI   AKTIND,DOPNCON
         BAL   14,DUMPRC

*-------------------------------------------------------------------------
*              RECEIVE MESSAGES
*-------------------------------------------------------------------------
         XR    9,9
RECCYCLE DS    0H
         YRECEIVE RPB=(7)
         L     4,YDDRARCL
         CVD   4,DW
         UNPK  RECLN,DW
```

```
         OI    RECLN+L'RECLN-1,X'F0'
         MVI   AKTIND,DRECEIVE
         BAL   14,DUMPRC
         CLI   YDDRFDB1,YDDRSUWA      ERROR DETECTED ?
         BH    ENDERR                 EVALUATE DCAM FDBK
         CLI   AAREA1,C'E'            END OF DATA ?
         BE    ENDREC
*-------------------------------------------------------------------
*              PROCESS RECEIVED MESSAGE
         AR    9,4                    RECEIVED BYTES
         VPASS 1
*-------------------------------------------------------------------
         B     RECCYCLE
*-------------------------------------------------------------------
*        SEND AN ACKNOWLEDGEMENT AND WAIT FOR THE PARTNER
*        TO CLEAR DOWN THE CONNECTION (YSENDREC FDB1 =  X'0C')
*-------------------------------------------------------------------
ENDREC   CVD   9,DW
         UNPK  AREA1(8),DW
         OI    AREA1+7,X'F0'
         YSENDREC RPB=(7),AREA=AREA1,AREALN=16,TOVAL=30
         MVI   AKTIND,DSENDREC
         MVC   RECLN,AREA1
         BAL   14,DUMPRC
         B     ENDE
*-------------------------------------------------------------------
*              HANDLING ROUTINE FOR UNEXPECTED FDBK
*-------------------------------------------------------------------
ENDERR   DS    0H
*        FDB1 = X'08'   APPLICATION CLEARED DOWN
*        FDB1 = X'0C'   CONNECTION ABORTED
*        ................
*-------------------------------------------------------------------
*              CLOSE THE APPLICATION
*-------------------------------------------------------------------
ENDE     DS    0H
         MVI   AKTIND,DCLOSE
         YCLOSE ACB=(5)
         BAL   14,DUMPRC
TERM     TERM
*-------------------------------------------------------------------
* SUBROUTINE: OUTPUT THE RETURN CODE
*-------------------------------------------------------------------
DUMPRC   DS    0H
         ST    15,SAV15
         LH    8,AKTDIST
         LA    8,AKTDIST(8)
         MVC   MESSTEXT,0(8)
         UNPK  MESSCODE(9),SAV15(5)
         TR    MESSCODE,TRTAB-C'0'
         MVI   FILLER,C''''
         WROUT MESSAGE,TERM
         MVC   RECLN,TXTNIL
         BR    14
***************************** DATA ***********************
TXTNIL   DC    CL8' '
MESSAGE  DS    0H
         DC    Y(MESSEND-MESSAGE)
```

```
         DC    CL5' '
MESSTEXT DC    CL8' '
         DC    CL1' '
RECLN    DC    CL8' '
         DC    C' FDBK=X'''
MESSCODE DC    C'       '
MESSEND  EQU   *
FILLER   DS    X
TRTAB    DC    C'0123456789ABCDEF'
SAV15    DS    F
AKTDIST  DS    0H
         DS    X
AKTIND   DS    X
DOPEN    EQU   *-AKTDIST
         DC    C'YOPEN   '
DCLOSE   EQU   *-AKTDIST
         DC    C'YCLOSE  '
DOPNCON  EQU   *-AKTDIST
         DC    C'YOPNCON '
DCLSCON  EQU   *-AKTDIST
         DC    C'YCLSCON '
DGENACB  EQU   *-AKTDIST
         DC    C'YGENACB '
DGENCCB  EQU   *-AKTDIST
         DC    C'YGENCCB '
DGENRPB  EQU   *-AKTDIST
         DC    C'YGENRPB '
DRECEIVE EQU   *-AKTDIST
         DC    C'YRECEIVE'
DSENDREC EQU   *-AKTDIST
         DC    C'YSENDREC'
DW       DS    D
ACBADR   DS    A
CCBADR   DS    A
ENBADR   DS    A
RPBADR   DS    A
OWNAPPL  DC    C'ISOBS02 '
OWNPROZ  DC    C'       '
PTNAPPL  DC    C'ISOBS01 '
PTNPROZ  DC    C'       '
LINKNAME DC    C'ISOBS02 '
ACB1     YACB  APPNAME=OWNAPPL,PRONAME=OWNPROZ,DCAMVER=8.0,            -
               ISO=Y,LINK=LINKNAME,LINKMOD=PERM
CCB1     YCCB  PTNNAME=PTNAPPL,PRONAME=PTNPROZ,RLTH=500,               -
               MDATA=Y,LINK=LINKNAME,LINKMOD=PERM
RPB1     YRPB  AAREA=AAREA1,AAREALN=L'AAREA1,ACB=ACB1,CCB=CCB1,        -
               OPTCD=(Q,CS,ACQUIRE,SPEC,SYN),TOVAL=240
AREA1    DS    CL32
AAREA1   DS    CL1024
         END
```

6 The DCAM system exit

Introduction

The DCAM system exit provides the user with the option of extending and changing DCAM functions and adding his own special functions. Examples of such functions are:

– data protection extending beyond TRANSDATA with extended authorization checks (e.g. permitting applications to be opened dependent upon the user ID or only permitting connection setup with specific partners or processors for an application)

– initiating additional actions (starting jobs, outputting messages).

The user-specific exit routine is called when opening and closing applications and when opening and closing connections. When opening applications and connections, the exit routine can accept, reject, or if applicable, modify the call. The calls when closing only serve to inform the exit routine.

DCAM exit events

The DCAM exit is called in the case of the events explained below. Each event is clearly identified by the main event name and subevent name.

Main event name	
'APP'	Event relating to the existence of applications
'CON'	Event relating to the existence of connections
Subevent name	Defines the individual events (opening, closing, ...)

The exit main event 'APP' is called each time an application is opened (primary or secondary) and each time it is closed (explicitly by YCLOSE or implicitly at the end of the program, shutdown).

The exit main event 'CON' is called during connection setup/connection cleardown in the following cases:

– request for connection setup (YOPNCON ACQUIRE):

  1. in the case of YOPNCON ACQUIRE call by the user
  2. after successful connection setup

– acceptance of a connection request

  1. if a request for connection setup arrives

  2. after acceptance of the request (YOPNCON ACCEPT)

– in the case of explicit connection cleardown by the user or partner or after a DCAM error.

The exit main event ’CON’ is not called in the case of implicit closure of all existing connections when an application is closed.

1) Exit event ’APPOPN’

   The exit routine is called

   – for each YOPEN call, after the user information has been checked and before the call is processed.

   Functions of the exit routine:

   – additional checks

     (e.g. – application may only be opened by specific user IDs
     – reloading tables for the exit routine)

   – accepting the YOPEN call

   – rejecting the YOPEN call

   – modifying specific YOPEN parameters

   – initiating additional actions

     (e.g. – starting jobs (ENTER)
     – outputting messages)

2) Exit event ’APPCLS’

   The exit routine is called

   – for each YCLOSE call, before cleardown of the DCAM data structure

   Functions of the exit routine:

   – initiating additional actions

     (e.g. – outputting messages
     – updating exit-specific tables)

3) Exit event ’APPSHU’

The exit routine is called

– for each internal closure of an application after input of a ’BCAPPL’, ’BCEND’ or ’SHUTDOWN’ command by the operator.

Notes

– The exit routine is called only when an application is closed, not when a warning (if any) is issued beforehand.

– The ’APPSHU’ exit routine is not called if TIAM termination takes place before DCAM has been terminated, i.e. if the DCAM program was started as an interactive process. In this case, the task is terminated by TIAM. The exit is called with ’APPTRM’.

4) Exit event ’APPTRM’

The exit routine is called

– in the case of internal closure of an application when a program or task is terminated

Functions of the exit routine:

– new start of an alternate task

5) Exit event ’APPFCL’

The exit routine is called

– in the case of forced closure of the application owing to a DCAM error.

6) Exit event ’CONACQ’

The exit routine is called

– when a request for connection setup is sent (YOPNCON ACQUIRE), after the user information has been checked, before the call is executed (i.e. before the system issues the REQCON message).

Functions of the exit routine:

– additional checks (e.g. connection message (LOGON message) in accordance with specs ?)

– accepting the call

– rejection of the call

– modifying specific YOPNCON parameters

7) Exit event ’CONACC’

The exit routine is called

– each time a connection request is accepted (YOPNCON ACCEPT), after the user information has been checked, before the call is processed.

Functions of the exit routine:

– additional checks
– accepting the call
– rejecting the call
– modifying specific YOPNCON parameters

8) Exit event ’CONCPL’

The exit routine is called

– after successful connection setup, i.e. if a YOPNCON ACQUIRE has been accepted by the partner.

9) Exit event ’CONREQ’

The exit routine is called

– if a connection request arrives, after checking the information, prior to processing the connection request.

Functions of the exit routine:

– additional checks
– accepting the connection request
– rejecting the connection request

10) Exit event ’CONCLS’

The exit routine is called

– each time a connection is cleared down by the user (YCLSCON), before releasing the DCAM data structure.

11) Exit event ’CONFCL’

The exit routine is called

– each time a connection is forcibly cleared down owing to a NEABI protocol error or owing to acknowledgment overflow.

12) Exit event ’CONBAD’

The exit routine is called

– each time a connection is cleared down, initiated by the partner.

6.1 System exits operation

With DSSM as a subsystem, the system administrator can

– load

– activate

– and deactivate the DCAM exit routines and

– query the status of the exit routines.

The number of the DCAM exit is ’075’.

Management of the exits is described in the BS2000 manual "System Exits".

6.2 System exits program interface

The first section of this chapter covers the general description of the DCAM exit program interface (i.e. valid for all exit events).

The second section describes, for each exit event, which fields in the parameter lists are supplied with data and how the return fields have to be or can be supplied with data.

The third section describes the restrictions which must be noted when programming a DCAM exit routine.

The fourth section describes the formats of the macros with which the DSECTs of the DCAM data structure can be generated.

6.2.1 DCAM exit program interface

6.2.1.1 Register data input

Upon entry to the exit routine, registers 4-11 are undefined and the remaining registers contain the following values

R0 = ’075’ number of the DCAM exit
R1 = A (YDDEXPL) DCAM-exit parameter list
R2 = A (task control block)
R3 = A (executive vector table)
R12 = A (P2 program manager)
R13 = A (18-word register save area)
R14 = A (indirect return)
R15 = A (exit routine)

Registers 12, 13, 14 must not be destroyed by the exit routine.

6.2.1.2 DCAM-Exit parameter list

A DSECT can be generated for the DCAM exit parameter list with macro YDDEXPL.

Format

Name	Operation	Operands
[symbol]	YDDEXPL	D [,{prefix / YDDX}]

Operands

symbol — symbolic address for the macro. If no address is specified, the address YDDEXPL is issued.

D — A dummy section (DSECT) is generated.

prefix — This specifies a character string of up to 4 characters used as a prefix for the name.
The default is YDDX.

Example

```
YDDEXPL  DSECT
*****************************************************************************
*                                                                           *
*    DCAM  EXIT PARAMETER LIST                                              *
*                                                                           *
*****************************************************************************
*
* INPUT PARAMETERS
*
YDDXDPAR DS    A             A(DARPPAR)
*                            DCAM INTERNAL WORK AREA
YDDXMAIN DS    X             EXIT-MAIN-CASE
YDDXAPPL EQU   1               APPL
YDDXCONN EQU   2               CONN
YDDXSUB  DS    X             EXIT-SUB-CASE
YDDXAOPN EQU   1               APPOPN
YDDXACLS EQU   2               APPCLS
YDDXASHU EQU   3               APPSHU
YDDXATRM EQU   4               APPTRM
YDDXAFCL EQU   5               APPFCL
YDDXCACQ EQU   6               CONACQ
YDDXCACC EQU   7               CONACC
YDDXCCPL EQU   8               CONCPL
YDDXCREQ EQU   9               CONREQ
YDDXCCLS EQU   10              CONCLS
YDDXCFCL EQU   11              CONFCL
YDDXCBAD EQU   12              CONBAD
*
* INPUT/RETURN PARAMETERS
*
YDDXLOGL DS    H               LENGTH OF THE LOGON MESSAGE
YDDXLOGM DS    CL80            LOGON MESSAGE
*
* RETURN PARAMETERS
*
YDDXPW   DS    XL4             PASSWORD
YDDXL    EQU   *-YDDEXPL         LENGTH OF THE DCAM EXIT PARAMETER LIST
               *,YDDEXPL      450    840126   55647211
```

Description of the fields in parameter list YDDEXPL:

YDDXPAR A (DARPPAR)
Address of a DCAM-internal work area. Further call-specific information is transferred to the exit routine in this work area.

YDDXMAIN Exit main event (main-case)
Indicator for the exit routine
YDDXAPPL: event related to the existence of applications.
YDDXCONN: event related to the existence of connections.

YDDXSUB Exit subevent (sub-case)
Indicator for the exit routine
YDDXAOPN: exit event APPOPN
YDDXACLS: exit event APPCLS
YDDXASHU: exit event APPSHU
YDDXATRM: exit event APPTRM
YDDXAFCL: exit event APPFCL
YDDXCACQ: exit event CONACQ
YDDXCACC: exit event CONACC
YDDXCCPL: exit event CONCPL
YDDXCREQ: exit event CONREQ
YDDXCCLS: exit event CONCLS
YDDXCFCL: exit event CONFCL
YDDXCBAD: exit event CONBAD

YDDXLOGL Length of LOGON message
The length of the connection message in field YDDXLOGM is specified in this field (≤ 80 bytes for DCAM(NEA) transport service applications, ≤32 bytes for DCAM(ISO) transport service applications). YDDXLOGL = 0 signifies that no connection message has been specified. This field can be changed by the exit routine.

YDDXLOGM LOGON message
The exit routine can transfer a connection message (LOGON message) in this field during return. This connection message replaces the connection message specified by the user in field AREA in the YOPNCON macro.

YDDXPW Password
In this field, the exit routine can transfer a password to DCAM during return. This password replaces the RDF password specified by the user in the YOPEN macro or the LOGON password specified in the YOPNCON ACCEPT macro.

6.2.1.3 Return information

The exit routine returns return information to the calling DCAM component in the following format in register 15:

Register 15
byte 0: reserved
bytes 1-2: FDBK
byte 3: RC

RC: return code of the exit routine to the calling DCAM component. The following values are possible:

00 : accept call
04 : reject call
08 : modify call

FDBK: feedback information from the exit routine to the DCAM user. This field must be supplied with data if RC=04 has been set. This information is transferred to the DCAM user in field FDBK1/FDBK2. FDBK can be taken either from an area 'CC00'-'CCFF', reserved for the exit routine, or may be one of the valid DCAM return codes. FDBK must not be '0000'.

If RC=08 has been specified, the exit routine transfers the valid data for modification in the parameter list YDDEXPL.

6.2.2 DCAM exit events

6.2.2.1 DCAM data structure

The most important elements of the DCAM data structure are described below:

DARPPAR internal DCAM work area:
contains task-specific data, e.g. the current addresses of the DCAM control blocks.

Addresses/fields which are not specified are initialized with X'00'.

ACB
CCB
RPB
ENB
DIP (NEA)
DCG (NEA)

} **control blocks specified by the user**

DACB DCAM application control block

The DACB contains application-specific data, e.g. application name, attributes,... . The DACB is created during YOPEN and is retained until the application is closed.

DCCB DCAM connection control block

The DCCB contains connection-specific data, e.g. partner names, processor names. One DCCB is created per connection.

DRPB DCAM request parameter block

contains the call-specific information from the RPB in the case of asynchronous calls.

When the exit routine is called, the following fields of the DCAM data structure are filled, depending on the exit event:

Field name	Field contents	Exit event APP- OPN	APP- CLS	APP- SHU	APP- TRM	APP- FCL	CON- ACQ	CON- ACC	CON- CPL	CON- REQ	CON- CLS	CON- FCL	CON- BAD
YDDEXPL: Exit parameter list													
YDDXMAIN	Exit main event	x	x	x	x	x	x	x	x	x	x	x	x
YDDXSUB	Exit subevent	x	x	x	x	x	x	x	x	x	x	x	x
YDDXDPAR	A (DARPPAR) = internal work area	x	x	x	x	x	x	x	x	x	x	x	x
YDDXLOGL	Length of the connection message						x	x	x	x			
YDDXLOGM	Address of the connection message						x	x	x	x			
YDDDVEC: Internal work area													
YDDVXTID	TID task identifier	x	x	x	x	x	x	x	x	x	x	x	x
YDDVADAC	A(DACB) DCAM application control block	x	x	x	x	x	x	x	x	x	x	x	x
YDDVADCC	A(DCCB) DCAM connection control block						x	x	x			x	x
YDDVADRP	A(DRPB) DCAM request control block						x	x	x				
YDDVARPB	A(RPB) user request control block						x	x	x		x		
YDDVAPTN	A(PTN)							x		x			
YDDVAPCN	A(PCN)							x		x			

See page 277 for data entry for the other fields of the internal work area and the control blocks (DACB, DCCB, ...). Addresses and areas for which data has not been entered are initialized with X’00’.

6.2.2.2 Return information for the exit events

In the case of the exit events which are used to inform the exit routine, the exit routine cannot influence processing in DCAM. For this reason, RC=00 must always be returned in register 15 in the case of the following exit events:

```
APPCLS
APPSHU
APPTRM
APPFCL
CONCPL
CONCLS
CONFCL
CONBAD.
```

In the case of exit events APPOPN, CONACQ, CONACC and CONREQ, the exit routine can influence further processing in DCAM. The table below specifies which return information can be set in the case of these events.

Return information		Condition	Exit events APPOPN	CONACQ	CONACC	CONREQ
Register 15	RC = 00		x	x	x	x
	RC = 04		x	x	x	x
	RC = 08		x	x	x	-
	FDBK	RC = 04	+	+	+	+
YDDEXPL	YDDXPW	RC = 08	+	+	-	-
	YDDXLOGL	RC = 08	-	+	+	-
	YDDXLOGM	YDDXLOGL>0	-	+	+	-

```
x: the value may be specified
-: the value/the item may not be specified
+: the field must be specified if the specified condition applies.
```

The exit routine can initiate modification of certain user specifications by setting RC=08 in register 15. The new specifications are transferred to DCAM via the return fields in parameter list YDDEXPL.

The following modifications can be initiated:

Exit Case	Field in YDDEXPL	Modification
APPOPN	YDDXPW	The RDF password (parameter USEPW in the ACB) is replaced by the transferred password in the case of the BCAM call
CONACC CONACQ	YDDXLOGM YDDXLOGL	The connection message (LOGON MESSAGE) of the user (contents of AREA) is replaced by the transferred data prior to the BCAM call. Connection message deleted in the event of YDDXLOGL=32767
CONACQ	YDDXPW	For DCAM(NEA) transport service applications, the LOGON password (parameter LOGPW in the CCB) is replaced by the transferred password prior to the BCAM call

6.2.3 Restrictions applying to DCAM exit routines

The restrictions described in the BS2000 manual "System Exits" apply to programming of DCAM exit routines. In addition, the following restrictions apply to DCAM exit routines in order to avoid inconsistencies in the DCAM data structure and deadlock situations.

1) Only read access is permitted to all fields of the DCAM data structure (DARPPAR, control blocks).
   Write access is permitted only to the return fields in the parameter list YDDEXPL.
2) No BCAM calls are permitted.
3) The exit routine may not set any locks.
4) The process level (PCB level) at which the exit routine runs may not be changed.

6.2.4 DSECT macros for the DCAM control blocks

Compatibility cannot be guaranteed for the layout of the DCAM control blocks. For this reason, the DCAM data structure may be accessed only via the symbolic field names. New compilation may be necessary in the case of any change in the DCM version.

1. DCAM internal work area (DARPPAR):

Format

Name	Operation	Operands
[symbol]	YDDDVEC	{D / C} [,{prefix / YDDV}]

2. DCAM application control block (DACB)

Format

Name	Operation	Operands
[symbol]	YDDDACB	{D / C} [,{prefix / YDDH}]

3. DCAM connection control block (DCCB)

Format

Name	Operation	Operands
[symbol]	YDDDCCB	{D / C} [, {prefix / YDDL}]

4. DCAM request parameter block (DRPB)

Format

Name	Operation	Operands
[symbol]	YDDDRPB	{D / C} [, {prefix / YDDD}]

Explanation of the operands

[symbol] — symbolic address of the macro. If no address is specified, the macro name is used as the address.

D — specifies that a dummy section is to be generated.

C — specifies that a code section is to be generated.

prefix — a character string of up to four characters which is to be used as a prefix for the name. If this specification is omitted, the underscored character string is used as the prefix in each case.

7 Appendix

7.1 Table of control block fields

The following table lists all control block fields which can be accessed or modified by control block or action macros.
The value or mnemonic code which can be used with each particular field is specified. The YGENCB, YMODCB, YSHOWCB and YTESTCB calls can be used only when the DCAM subsystem has been loaded successfully. Note, too, that the DCAM subsystem status cannot be HOLD/DELETE when any of these macros is used. If a task successfully issued a DCAM command or a DCAM call before entering HOLD/DELETE, it can work with DCAM until the task is ended, despite a /HOLD subsystem or /DELETE subsystem (also applicable to %).

For the YSHOWCB macro, the length of the user area required for each field that can be processed with YSHOWCB is specified (in bytes).

The mnemonic codes

– value and
– addr

in the fields listed below also allow register specification.

Control block	Keyword of control block field	Value or mnemonic code				Field length for YSHOWCB	Explanations
		Macro for static control block gener.	YGENCB	YMODCB	YTESTCB		
ACB	APPNAME DISNAME ENB LINK PRONAME	relexp	addr	addr	addr	4	Address of: {application name | distribution code | name | event notification block | link name | symbolic processor name}
	ISO	{Y | N}					Setting the ISO attribute
	LINKMOD	{PERM | TEMP}					Transfer of CLT entry
	ATTR	([{SHARE | NSHARE}][,{DISCO | NDISCO}][,{PRIMTASK | REQTASK | NOTACK}][,{LOGON | NLOGON}])					Characteristics of DCAM application
	VERIFY	{NO | PRIMARY | SECONDARY}					Test specification
	USEPASS USEPW LOGPASS	password	{password | (register)}	{password | (register)}	{password | (register)}	4	Application password (macro) Application password (user) Connection password (macro)
	DCAMVER	8.0					Number of the DCAM version
	AID			(register)	(register)	4	DCAM application identifier
	FDBK					4	Feedback information field
	ACBLN				value	2	Length of ACB control block

Control block	Keyword of control block field	Value or mnemonic code				Field length for YSHOWCB	Explanations
		Macros for static control block generation	YGENCB	YMODCB	YTESTCB		
CCB	LINK PRONAME PTNNAME DIP APTNCH ROUTL	relexp	addr	addr	addr	4	Address of: link name / processor name / partner name / distribution parameter block / partner characteristics / connection routes
	MDATA	[N / Y]					MORE-DATA mechanism
	RLTH	absexp	value	value	value	4	Maximum length of the messages/data units for receive (local)
	MAXLN	absexp	value	value	value	2	Maximum length of a message on this connection
	PTNCHLN	absexp	value	value	value	2	Length of the partner characteristics
	ROUTN	absexp	value	value	value	2	Number of connection routes (up to 16)
	LINKMOD	[PERM / TEMP]					Transfer of CLT entries
	USERFLD	user field	[user field / (register)]	[user field / (register)]	[user field / (register)]	4	Accompanying connection information
	LOGPW	password	[password / (register)]	[password / (register)]	[password / (register)]	4	Register password (user)
	EDIT	[USER / SYSTEM]					Message editing
	EDITIN	([PHYS / LINE / FORM][,GETBS / NGETBS][,GETFC / NGETFC][,LCASE / NLCASE])					Input message editing

Control block	Keyword of control block field	Value or mnemonic code				Field length for YSHOWCB	Explanations
		Macros for static control block generation	YGENCB	YMODCB	YTESTCB		
	EDITOUT	([{PHYS / LINE / FORM}][,{HCOPY / NHCOPY}][,{HOM / NHOM}][,{EXTEND / NEXTEND}] [,{LOGC / NLOGC}][,{LACK / NLACK}])					Output message editing
	PROC	([{TRUNC / KEEP}][,{SYSCODE / BINARY}][,{APPSTART / ANYSTART}][,{SIGNAL / NSIGNAL}]) [,{TERMSTAT / NTERMSTAT}])					Connection parameters
	DID		(register)	(register)	(register)	4	Message distribution identifier
	CID			(register)	(register)	4	Connection identifier
	PTNCHA1				1)	4	Partner characteristics, part 1
	PEDIT				{USER / SYSTEM}		Message editing proposed by the partner
	PPROC				{PTNSTART / ANYSTART}		Start of data transmission proposed by the partner
	FLAG				{ACTIVE / INACTIVE}		Flag for use of a control block by a macro
	PRIO	absexp	value	value	value	1	Transport priority
	CCBLN				value	2	Length of the CCB

Control block	Keyword of control block field	Value or mnemonic code				Field length for YSHOWCB	Explanations
		Macros for static control block generation	YGENCB	YMODCB	YTESTCB		
DCG	CODEVAL	(codevalue,...)				64	Up to 8 distribution code values (each entered left-justified in an 8-byte field)
	GID			(register)	(register)	4	Distribution code group identifier (DCG)
	DCGLN				value	2	Length of distribution code group block
DIP	CODELN CODEPOS	absexp	value	value	value	2 [1]	Length of distribution code Posit. of distribution code
	CODEIND	character	character	character	character	2 [1]	Indicator for implicit distribution code
	DCG	(relexp,...)	(relexp,...)			64	Up to 16 addresses of DCG control blocks
	DID			(register)	(register)	4	Message distribution identifier
	DIPLN				value	2	Length of DIP control block
ENB	LOGON LOSCON PROCON COMEND EXPR TACK SECOND	relexp	addr	addr	addr	4	Identifier address of {LOGON, LOSCON, PROCON, COMEND, EXPR, TACK, SECOND} contingency routine
	ENBLN				value	2	Length of ENB control block

[1] Right-justified byte

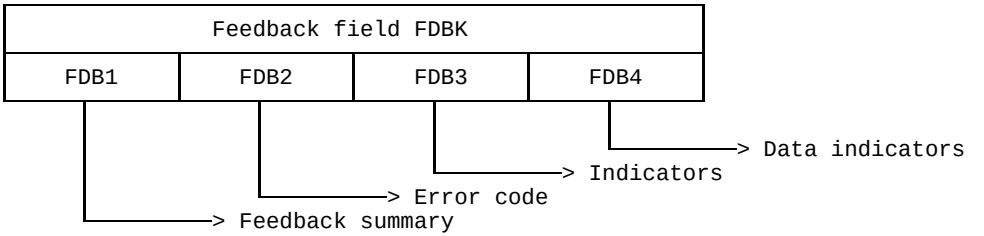
Control block	Keyword of control block field	Value or mnemonic code				Field length for YSHOWCB	Explanations
		Macros for static control block generation	YGENCB	YMODCB	YTESTCB		
RPB	ACB CCB AREA AAREA VCBADR EID	relexp	addr	addr	addr	4	Address of {ACB control block / CCB control block / buffer area / alternative buffer / VTSUCB area / event identifier}
	AREALN AAREALN	absexp	value	value	value	4 4	Length of buffer area Length of alternative buffer area
	TOVAL					2	Time interval for macro queueing
	SEQNO					2	Sequence number of message to be transmitted
	EIDREF	information	{infor-mation (register)}	{infor-mation (register)}	{infor-mation (register)}	4	First word of user information after termination of an asynchronous call
	EIDREF2	information	{infor-mation (register)}	{infor-mation (register)}	{infor-mation (register)}	4	Second word of user information after termination of an asynchronous call
	OPTCD	([{ACCEPT / ACQUIRE}][,{CS / CA}][,{ELEMENT / SUBGROUP / GROUP}][,{NORMAL / EXPRESS}] [,{Q / NQ}][,{REQLOGON / TOPLOGON / NAMXLATE / COUNPTN / APPSTAT / CIDXLATE / PTNCHAR / VTSUCB / MONCHARS / PEROTERM / BTERMINF}][,{SPEC / ANY}][,{START / STOP}] [,{SYN / ASY}][,{TACK / NTACK}][,{TRUNC / KEEP / CCBTK}][,{BELL / NBELL}])					Request identifiers

Control block	Keyword of control block field	Value or mnemonic code				Field length for YSHOWCB	Explanations
		Macros for static control block generation	YGENCB	YMODCB	YTESTCB		
	LID		(register)	(register)	(register)	4	Connection request identifier
	AID CID		(register)	(register)	(register)	4	DCAM application identifier Connection identifier
	ARECLN					4	Length of data entered in alternative buffer area or the length of the remainder if the message is longer than the buffer area
	ASEQNO				value	2	Sequence number of received message
	TACKNO					2	Sequence number of received transport acknowledgment
	USER				{user field (register)}	4	Accompanying connection information as specified in the CCB
	REQTYPE				1)		Macro type
	FLAG				{ACTIVE INACTIVE}		Flag for control block usage by a macro
	FDBK					4	Feedback information field
	RPBLN				value	2	Length of RPB control block

1) Name of macro without Y prefix character.

7.2 Table of feedback information

The feedback field is located in the RPB or ACB control block and has the name FDBK. It has a length of 4 bytes:



All entries are in hexadecimal. Symbolic addresses can be generated for the contents of register 15 (YDDFDB macro, see the appendix, page 309).

The feedback information is also supplied in register 15.

If the ACB or RPB control block is no longer addressable, the feedback information is supplied only in register 15.

With asynchronous instruction execution, register 15 may sometimes only contain information on the acceptance or rejection of the instruction (cf. table below).

Feedback info / Macro status	Control block: Valid: FDBK in ACB/RPB	Control block: Valid: Register 15	Control block: Invalid: Register 15 3)	Control block: Invalid: Execution confirmed?
Synchronous macro rejected immediately	FDB1 FDB2	FDB1 FDB2	X’20’	-
executed immediately	FDB1 to FDB4	FDB1 to FDB4	-	-
executed/rejected after a delay	FDB1 to FDB4	FDB1 to FDB4	X’20’ 1)	-
Asynchronous macro rejected immediately	FDB1 FDB2	FDB1 FDB2	X’20’	No
accepted and executed immediately	FDB1 to FDB4	FDB1 4)	-	Yes
executed/rejected after a delay	FDB1 to FDB4	-	2)	Yes

1) The wait status can be interrupted by a contingency routine; it is thus possible that the control block was destroyed by the user.

2) The user receives control before the macro is executed; it is thus possible that the control block was destroyed by the user. In this case, the user cannot receive any feedback information and cannot subsequently access the invalid control block with YSHOWCB or YTESTCB.

3) Left-justified byte.

4) X’00’ = "accepted".

FDB1	FDB2	FDB3	FDB4	Macro status	Remark
X'00'	-	-	-	Executed successfully	-
X'00'	-	Primary/secondary task indicator	-	Executed successfully	Only for YOPEN macro
X'00'	-	Message length, data characteristics and grouping indicator		Executed successfully	Only for YSENDREC and YRECEIVE
X'00'	-	-	-	Accepted	Only for asynchronous YOPNCON, YRECEIVE and YSENDREC
X'04'	-	Warning indicator	Data characteristics and grouping indicator	Executed	-
X'08'	Cause	-	-	Rejected	Caused by DCAM application status
X'0C'	Cause	-	-	Rejected	Caused by partner status
X'10'	Cause	-	-	Rejected	Caused by data transmission system status
X'14'	Error code	-	-	Rejected	Invalid use of a macro
X'18'	Error code	-	-	Rejected	Wrong macro operands
X'CC'	Error code	-	-	Rejected	Rejection by the system exit

Register 15	-	-	-	Macro Status	Remark
X'20'	-	-	-	Rejected	Wrong address or register specification
X'24'	-	-	-	Rejected	DCAM subsystem not available

Feedback field				Meaning	Macro													
FDB1	FDB2	FDB3	FDB4		YOPEN	YCLOSE	YOPNCON	YCLSCON	YREJLOG	YSETLOG	YCHANGE	YINQUIRE	YPERMIT	YFORBID	YSEND	YRECEIVE	YSENDREC	YRESET
00				Macro terminated successfully	x	x	x	x	x	x	x	x	x	x	x	x	x	x
00		00		Task is the primary task } of the DCAM application	x													
00		04		Task is a secondary task } of the DCAM application	x													
00				Macro accepted (OPTCD=ASY)			x									x	x	
00		00	*	Message does not exceed the buffer length											x	x	x	
00		2^2=1	*	Message exceeds the buffer length; excess data deleted											x	x	x	
00		2^3=1	*	Message exceeds the buffer length; excess data retained; length of the remainder is in field ARECLN of the RPB control block											x	x	x	
00		2^4=1	*	The requested VTSUCB could not be transferred to the user area (OPTCD=VTSUCB); the connection does not use VTSUCB.												x		
00		2^6=1	*	Status of 9763 Data Display Terminal incomplete (PROC=TERMSTAT)											x	x	x	
00		2^7=1	*	Status of data display terminal 9763 received (PROC=TERMSTAT)												x	x	
			2^0=1	Message											*	x	x	
			2^1=1	GO signal											x		x	
			2^2=1	Transport acknowledgment - positive												x	x	
			2^3=1	Transport acknowledgment - negative												x	x	
			2^4=1	Express message											*	x	x	
			2^5=1	Element } Data grouping in DCAM(NEA) transport service												x	x	
			2^6=1	Subgroup } Data grouping in DCAM(NEA) transport service												x	x	
			2^7=1	Group } Data grouping in DCAM(NEA) transport service												x	x	
			2^5=1	Further data units to follow } in DCAM(ISO) applications												x	x	
			2^7=1	No further data units to follow } in DCAM(ISO) applications												x	x	

* only when set together with 2^1=1 (GO signal)

Feedback field				Meaning	Macro													
FDB1	FDB2	FDB3	FDB4		YOPEN	YCLOSE	YOPNCON	YCLSCON	YREJLOG	YSETLOG	YCHANGE	YINQUIRE	YPERMIT	YFORBID	YSEND	YRECEIVE	YSENDREC	YRESET
04				Macro terminated with warning			x		x		x	x			x	x	x	
04		04	*	Mutually exclusive data coding information (BINARY/SYSCODE)												x	x	
			2^0=1 2^7=1	} as for FDB1=X'00'												x	x	
04		08		No input data present (OPTCD=NQ)												x	x	
04		0C		No connection request or no request with appropriate EDIT-OPTION in queue			x		x									
04		10		Macro terminated because of timeout (TOVAL)			x									x	x	
04		14		Data loss because of (system) timeout or because the message was identified as garbled												x	x	
04		18		Queued connection request deleted because of (system) timeout			x											
04		20		Connection message truncated			x					x						
04		24		Message editing error											x	x	x	
04		28		Information not completely transferred; YINQUIRE aborted.								x						
04		2C		Output truncation (edited message was too long)											x		x	
04		30		Invalid message editing information (modification was not performed)							x							
04		34		Macro terminated by YRESET												x	x	
04		44		No printer operational for data output			x				x							
04		48		Invalid check character (NL) has been entered (with EXTEND); with 3270 and EXTEND, also message indicating not all fields transferred; message truncated												x	x	

Feedback field				Meaning	Macro													
FDB1	FDB2	FDB3	FDB4		YOPEN	YCLOSE	YOPNCON	YCLSCON	YREJLOG	YSETLOG	YCHANGE	YINQUIRE	YPERMIT	YFORBID	YSEND	YRECEIVE	YSENDREC	YRESET
08				Macro rejected because of current DCAM application status	x	x	x	x	x	x	x	x	x	x	x	x	x	x
08	04			DCAM application not opened by calling task (AID invalid)		x	x	x	x	x	x	x	x	x	x	x	x	x
08	08			DCAM application already open (ATTR=NSHARE set)	x													
08	0C			The DCAM application is already open and is not shareable.	x													
08	10			The DCAM application has already been opened by the requesting task.	x													
08	14			The DCAM application is already open (VERIFY=PRIMARY set).	x													
08	18			The DCAM application is not open (VERIFY=SECONDARY set).	x													
08	1C			Distribution code name/distribution codes already permitted									x					
08	20			Warning: forced termination of DCAM application	x		x						x					
08	24			Forced termination of DCAM application	x	x	x	x	x	x	x	x	x	x	x	x	x	x
08	28			Forced termination of DCAM application due to DCAM error	x	x	x	x	x	x	x	x	x	x	x	x	x	x
08	2C			Forced termination of DCAM application due to specification of an invalid contingency/ event identifier by the primary task			x								x	x	x	
08	30			Forced termination of the DCAM application due to specification of an invalid contingency/ event identifier by the secondary task											x	x	x	
08	34			Forced termination of the DCAM application due to termination of the primary task	x	x						x			x	x	x	
08	38			Termination of the DCAM application through a primary task request	x	x	x					x			x	x	x	

Feedback field				Meaning	Macro													
FDB1	FDB2	FDB3	FDB4		YOPEN	YCLOSE	YOPNCON	YCLSCON	YREJLOG	YSETLOG	YCHANGE	YINQUIRE	YPERMIT	YFORBID	YSEND	YRECEIVE	YSENDREC	YRESET
08	3C			Termination of the DCAM application through a secondary task request												x	x	
08	40			Too many macros of the same type were issued for this task simultaneously (up to 8 permitted, except for YOPNCON ACQUIRE where 128 permitted)			x									x	x	
08	44			The DCAM application is not active.	x													
08	48			Invalid password (USEPW)	x													
08	4C			The task has opened too many applications.	x													
08	50			Too many undefined applications for a task	x													
08	54			Invalid password with existing RDF password	x													
08	58			The application was opened by another subsystem.	x													
08	5C			Too many undefined applications	x													
08	60			Too many connections for the undefined application			x											
08	64			The DCAM version does not correspond to that specified by the primary task.	x													

Feedback field				Meaning	Macro													
FDB1	FDB2	FDB3	FDB4		YOPEN	YCLOSE	YONCON	YCLSCON	YREJLOG	YSETLOG	YCHANGE	YINQUIRE	YPERMIT	YFORBID	YSEND	YRECEIVE	YSENDREC	YRESET
0C				The macro was rejected due to the current partner status.			x	x			x	x			x	x	x	x
0C	04			The partner is not linked with the application (invalid CID).				x			x	x			x	x	x	x
0C	08			The partner is already linked with the DCAM application. In DCAM(ISO) transport service applications, no parallel connection is possible, due to the transport protocol being used (NEA). Note: CID is supplemented by DCAM in the RPB and CCB as with a return code 00 00.			x x											
0C	0C			A request from this partner has already been queued (no ACQUIRE possible).			x											
0C	10			The DIP control block is not active (invalid DID).			x				x							
0C	14			Invalid LID								x						
0C	18			The connection was cleared by a user request or a YCLSCON.			x									x	x	x
0C	1C			The connection with the partner was cleared forcibly or the partner has cleared the connection.											*	x	x	x
0C	20			The partner is in the CS state and transition to the CA state is not completed.											x			
0C	24			The partner is in the CS state and YRECEIVE instructions are not yet completed. The requested transfer to the CA state is not permissible.											x		x	
0C	28			The partner is in the CS state for another task.											x	x	x	x
0C	2C			The partner is in the CA state. YRECEIVE (with OPTCD=SPEC) is not permitted.												x	x	
0C	30			The message is too long (> MAXLN).											x		x	
0C	34			The position of the distribution code exceeds the maximum message length.			x				x							
0C	38			This sequence number has already been used; the acknowledgment procedure is not yet completed.											x		x	

* only when set together with 2^1=1 and 2^0=1 or 2^4=1 in FDB4 (GO signal)

Feedback field				Meaning	Macro													
FDB1	FDB2	FDB3	FDB4		YOPEN	YCLOSE	YOPNCON	YCLSCON	YREJLOG	YSETLOG	YCHANGE	YINQUIRE	YPERMIT	YFORBID	YSEND	YRECEIVE	YSENDREC	YRESET
0C	3C			OPTCD=EXPRESS not possible when EDIT=SYSTEM											x		x	
0C	40			The partner has rejected the connection without specifying a reason.			x											
0C	44			The partner has requested the wrong protocol.			x											
0C	48			System timeout for connection request			x											
0C	4C			The partner is not accessible (if the partner is a DCAM application, then this is not yet open).			x											
0C	50			The partner is not processing connection requests (the DCAM application is in the STOP state).			x											
0C	54			The partner does not want to accept connection requests (the DCAM application has ATTR=NLOGON set).			x											
0C	58			Invalid password (LOGPW)			x											
0C	5C			The connection request was rejected by the partner (e.g. REJLOG macro). For DCAM(NEA) transport service applications only			x											
		see bel.		This byte contains details of the reason why the connection request was rejected (for diagnosis). For DCAM(ISO) transport service applications only														
		18		Invalid connection request (e.g. partner unknown, protocol error, invalid references)														
		2C		Partner has rejected connection request.														
		40		Connection establishment rejected (e.g. proposed protocol class not supported)														
0C	60			The partner characteristics have not been accepted by the partner.			x											
0C	64			Error in the terminal service protocol			x											
0C	68			The partner is not processing requests for connection setup; a request for connection setup by the partner is to follow.			x											
0C	6C			Error in activation of the partner’s VTSU support			x											

Feedback field				Meaning	Macro													
FDB1	FDB2	FDB3	FDB4		YOPEN	YCLOSE	YOPNCON	YCLSCON	YREJLOG	YSETLOG	YCHANGE	YINQUIRE	YPERMIT	YFORBID	YSEND	YRECEIVE	YSENDREC	YRESET
0C	70			The partner belongs to another DCAM application.			x											
0C	74			The DIP control block does not address a DCG control block			x				x							
0C	78			Connection cleared down by partner or system immediately after connection setup			x											
0C	7C			Reserved														
0C	80			Syntax error in connection message [1)]			x											
0C	84			Unknown authorization name [1)]			x											
0C	88			The application is already linked [1)]			x											
0C	8C			Invalid password [1)]			x											
0C	90			The proposed transport service class has not been accepted by the partner.			x											
0C	94			The proposed data network priority has not been accepted by the partner.			x											
0C	98			Processing error (e.g. fault on X.25)			x											
0C	9C			The connection setup request has been rejected by the administration. In this case, establish a connection with the system administrator.			x											
0C	A0			Connection cleared down by communication partner due to protocol error or inconsistency									x	x	x			
0C	A4			OPTCD=EXPRESS not permitted since it has not been agreed with the communication partner								x		x				
0C	A8			Fatal UCON error [1)]			x											
0C	AC			Invalid application name: first character not $ [1)]			x											
0C	B0			Invalid processor (not own processor) [1)]			x											
0C	B4			Task could not be created for checking [1)]			x											
0C	B8			Invalid CID for this OP-ID [1)]			x											
0C	BC			Internal UCON error (authentication not possible [1)]			x											

[1)] These messages are returned when a connection request addressed to a system application (usually "$CONSOLE") is rejected by the latter.

Feedback field				Meaning	Macro													
FDB1	FDB2	FDB3	FDB4		YOPEN	YCLOSE	YOPNCON	YCLSCON	YREJLOG	YSETLOG	YCHANGE	YINQUIRE	YPERMIT	YFORBID	YSEND	YRECEIVE	YSENDREC	YRESET
0C	C0			No further entry available in ECRNAM table [1]			x											
0C	C4			DCAM version is less than 10 [1])			x											
0C	C8			Not a chip card terminal [1]			x											
0C	CC			Protocol inconsistent [1]			x											
0C	D0			Chip card subsystem not available [1]			x											
0C	D4			Error in KVP protocol [1])			x											
10				The macro was rejected because of the current condition of the Data Communication Method DCM	x	x	x	x	x	x	x	x	x	x	x	x	x	x
10	04	s.b.		Lack of DCAM storage space	x		x				x		x		x	x	x	
		s.b.		Reasons														
		04		BCAM: Try input/output later														
		08		BCAM: Lack of storage space														
		0C		BCAM: Wait for GO signal														
		20		DCAM: No control block available														
		24		DCAM: No ID entry available														
		28		DCAM: No storage space available														
		2C		DCAM: Overload in partner’s system			x											
		30		DCAM: Too many DCAM events not fetched			x											
10	08			Warning: DCM termination	x		x				x		x					
10	0C			DCM termination	x	x	x	x	x	x	x	x	x	x	x	x	x	x
10	10			DCM not active	x	x	x	x	x	x	x	x	x	x	x	x	x	x
10	14	s.b.		DCM error	x	x	x	x	x	x	x	x	x	x	x	x	x	x
10	14	00		DCM error														
10	14	04	xx*	BCAM: Unknown BCINF return code														
10	14	08	xx*	BCAM: Unknown APINF return code														
10	14	0C	xx*	BCAM: Unknown STINF return code														
10	20			DCAM blocked temporarily	x													

[1] These messages are returned when a connection request addressed to a system application (usually "$CONSOLE") is rejected by the latter.

* BCAM return code

Feedback field				Meaning	Macro														
FDB1	FDB2	FDB3	FDB4		YOPEN	YCLOSE	YOPNCON	YCLSCON	YREJLOG	YSETLOG	YCHANGE	YINQUIRE	YPERMIT	YFORBID	YSEND	YRECEIVE	YSENDREC	YRESET
14				Illegal use of macro			x	x	x	x	x	x	x	x		x	x	
14	04			The macro may not be issued by a secondary task.			x	x	x	x	x	x	x	x				
14	08			The macro may not be used with DCAM applications having ATTR=NSHARE or ATTR=(SHARE,NDISCO) set.									x	x				
14	0C			This macro may not be used with DCAM application having ATTR=NLOGON set.			x			x								
14	10			Equivalent macros are already pending (OPTCD=(ACQUIRE,ASY) or OPTCD=(ACCEPT,SPEC,ASY)			x											
14	14			The DCAM application is not allowed to use this macro.			x											
14	18			A synchronous macro is already queued for this task OPTCD=(SYN,Q).			x									x	x	

Feedback field				Meaning	Macro													
FDB1	FDB2	FDB3	FDB4		YOPEN	YCLOSE	YOPNCON	YCLSCON	YREJLOG	YSETLOG	YCHANGE	YINQUIRE	YPERMIT	YFORBID	YSEND	YRECEIVE	YSENDREC	YRESET
18				The macro was rejected because of wrong operands.	x		x	x	x	x	x	x	x	x	x	x	x	x
18	04			Invalid ACB control block address			x	x	x	x	x	x	x	x	x	x	x	x
18	08			Invalid CCB control block address			x	x	x		x	x			x	x	x	x
18	0C			Invalid DCG control block address			x				x							
18	10			Invalid DIP control block address			x				x							
18	14			Invalid ENB control block address	x													
18	18			Invalid APPNAME address	x													
18	1C			Invalid DISNAME address	x													
18	20			Invalid PTNNAME address			x	x			x	x						
18	24			Invalid address of contingency identifier	x													
18	28			Invalid address of event identifier			x									x	x	
18	2C			Invalid AREA address			x					x	x	x	x			
18	30			Invalid AAREA address					x			x	x			x	x	
18	34			Invalid DCAM application name	x													
18	38			Invalid distribution code name	x								x	x				
18	3C			Invalid partner name			x	x	x		x	x						
18	40			Invalid processor name			x	x	x		x	x						
				Processor not activated (no /BCACT)			x	x	x		x	x						
				Processor unknown (not generated)			x	x	x		x	x						
				Route(s) to processor not active			x	x	x		x	x						
				Specified route to processor unknown (not generated)			x	x	x		x	x						
18	44			The CCB control block is being used for an asynchronous macro (CCB active).			x		x		x	x						

Feedback field				Meaning	Macro													
FDB1	FDB2	FDB3	FDB4		YOPEN	YCLOSE	YONCON	YCLSCON	YREJLOG	YSETLOG	YCHANGE	YINQUIRE	YPERMIT	YFORBID	YSEND	YRECEIVE	YSENDREC	YRESET
18	48			The sequence number exceeds the upper limit.											x		x	
18	4C			AAREALN is less than 8 bytes/AREALN=0											x	x	x	
18	50			Invalid subfunction								x						
18	54			Invalid PRONAME address	x		x	x			x	x						
18	58			Negative AREALN			x											
18	5C	s.b.		Unrecoverable message editing error:			x				x				x	x	x	
		00		EDIT parameter error: in the case of YRECEIVE and EDITIN=PHYS, the user area may be too small to accommodate the message header														
		04		Invalid device header														
		08		Message length is 0														
		0C		VTSU is not available														
		10		Error in VTSUCB														
		14		Message with error in NEABT protocol received												x	x	
18	60			Invalid length of distribution code (CODELN)			x											
18	64			Reserved														
18	68			Reserved														
18	6C			Specification of ACB parameters conflicts with ISO = Y	x													
18	70			There is no VTSUCB which can be returned to the user.								x						
18	74			VTSUCB not permitted (EDIT=USER)							x	x			x	x	x	
18	78			Invalid VCBADR address							x	x			x	x	x	
18	7C			Inconsistency between ROUTL and ROUTN			x											
18	80			Invalid ROUTL address			x											

Feedback field				Meaning	Macro													
FDB1	FDB2	FDB3	FDB4		YOPEN	YCLOSE	YOPNCON	YCLSCON	YREJLOG	YSETLOG	YCHANGE	YINQUIRE	YPERMIT	YFORBID	YSEND	YRECEIVE	YSENDREC	YRESET
20				The macro was rejected because of a wrong address or register specification.	x	x	x	x	x	x	x	x	x	x	x	x	x	x
20				Invalid ACB control block address	x	x												
20				Invalid RPB control block address or RPB control block is being used for an asynchronous instruction (RPB active).			x	x	x	x	x	x	x	x	x	x	x	x
20				Invalid address or invalid contents of operand list	x	x	x	x	x	x	x	x	x	x	x	x	x	x
20				Invalid register number	x	x	x	x	x	x	x	x	x	x	x	x	x	x
24				The macro was not executed because the DCAM subsystem either was not loaded or its status is HOLD/DELETE and the task in question has not yet issued a DCAM macro.	x	x	x	x	x	x	x	x	x	x	x	x	x	x
CC	XX			Macro has been rejected by system exit routine.	x		x											
	XX			Contents are defined by a special exit routine. Send queries to system administrator														

7.3 Register usage

1. Register usage at DCAM macro termination

Symbolic addresses can be generated for the contents of register 15 (YDDFDB macro, see the appendix, page 309).

	Register 0	Register 1	Registers 2 thru 14	Register 15
Feedback from a macro referring to an ACB control block	*)	Modified	*)	FDBK field
Feedback from a macro referring to an RPB control block	*)	Modified	*)	FDBK field
Feedback from YGENCB macro	Length of control block	Address of control block	*)	Feedback information
Feedback from YMODCB macro	*)	Modified	*)	Feedback information
Feedback from YSHOWCB macro	Required memory size	Modified	*)	Feedback information
Feedback from YTESTCB macro	Test result	Modified	*)	Feedback information

*) means "unchanged"

2. Register usage at activation of a contingency routine due to an asynchronous DCAM notification

Reg.	Contents
1	Contingency information specified with the COMAD operand of the ENACO macro
2	Reason for contingency routine activation in two bytes: ES: Event switch (right-justified) II: Information indicator (left-justified) The following information is hexadecimal: II ES 00 00 No contingency information (COMAD) 08 00 available 04 00 Contingency information (COMAD) 0C 00 available

	LOGON	LOSCON	PROCON	COMEND	EXPR *)	TACK *)	SECOND
3	AID	AID	AID	AID	AID	AID	AID
4	LID	CID	Name of proposed partner	See page 302	CID	CID	See page 302
5	Length of connec-tion message	Contents of USERFLD in CCB control block			Contents of USERFLD in CCB control block	Contents of USERFLD in CCB control block	Distribu-tion name
6		See page 302 table	Name of proces-sor to which proposed partner is con-nected		Express message data	Acknowl-edgment: 0: Positive 4: Negative	Distribu-tion name
7						Seq. no. of acknowl-edgment	
8					Seq. no. of message		

*) Columns EXPR and TACK apply only to DCAM(NEA) transport service applications

Symbolic addresses can be generated for the register contents listed below (YDDFDB macro, see the appendix, page 309).

Contents	(right-justified) of register 6 for LOSCON:
X’00’	Connection cleardown was requested by the user
X’08’	Processing error: e.g. owing to fault on X.25
X’0C’	The partner is no longer available, or the terminal is defective.
X’10’	Connection cleardown through external intervention (system operator)
X’14’	Connection interruption in the network. A new connection request should be expected
X’18’	Connection cleardown due to a failure or crash in the communication network
X’1C’	Reserved
X’20’	Warning of impending connection cleardown
X’24’	Reserved
X’28’	The connection of the partner has been cleared down after a request by the system administration
X’2C’	Error in the linkage element of the station services
X’30’	Reserved
X’34’	Error in the station services protocol
X’38’	Error in the transport system
X’3C’	Reserved
X’40’	Error in the connection message from the station services for the user
X’44’	Reserved
X’48’	Reserved
X’4C’	Reserved

Contents	(right-justified) of register 6 for LOSCON:
X’50’	The proposed class of transport services has not been accepted by the communication partner.
X’54’	The proposed priority in the data network has not been accepted by the partner.
X’58’	The connection has been cleared down owing to DCM shortage of memory space (too many negative transport acknowledgments not fetched).
X’5C’	The partner’s VTSU is not active.
X’60’	The connection has been cleared down due to a protocol error by the communication partners.
X’64’-X’90’	reserved; see YDFDB D,,CONT

Contents	(right-justified) of register 4 for COMEND:
X’00’	Warning of impending termination of communication access system
X’04’	Termination of communication access system
X’08’	Warning of impending forced termination of the DCAM application
X’0C’	Forced termination of DCAM application
Contents	(right-justified) of register 4 for SECOND:
X’00’	Opening of application in secondary task
X’04’	Closure of application in secondary task
X’08’	Messages for distribution name without secondary task.

The other registers do not contain defined values and must, therefore, be set by the user (base register included).

7.4 Macro mnemonics

Macro mnemonics help the user to remember the form that a specific operand may assume. Eight mnemonics are used in this description, to which eight different operand forms can be related (see table below).

Mnemonic			Operand forms					
	Relative ex-pression	Register desig-nation	Symbol	Decimal integer	Absolute ex-pression	Code	Text	Char-acter
relexp.	x							
absexp.					x			
addr	x	x						
integer				x				
value		x			x			
text							x	
code						x		
characters			x					x

Relative expression

A relative expression is an expression whose value will change by n when the program is relocated n bytes from its original location. The value of all relative expressions must be positive. A relative expression may also consist of an arithmetic combination of terms. The combination must comprise only relative and absolute terms and must satisfy the following conditions:

1. The number of relative terms must be odd.
2. All relative terms must be paired and 1 additional relative term must be provided. Pairing is described under "Absolute expressions".
3. The non-paired term may not include a minus sign.
4. A relative term may not occur as an operand in a multiplication or division.

A relative expression is assigned a single value by the assembler.

In the following examples of relative expressions, SAM, JOE and FRANK occur in the same program section and are relative terms, while PT is an absolute term.

```
SAM
SAM-JOE+FRANK
JOE-PT*5
SAM+3
```

Note

SAM-JOE is not a relative term because the difference between two relative addresses represents a constant value.

Register designation

Register designations are written as absolute terms enclosed in parentheses. The absolute terms must assume values between 2 and 12, which are the associated general registers.

In the following example of register designations, SAM and JOE are relocatable terms and PAL is an absolute term.

```
(5)         indicates register 5
(SAM-JOE)
(PAL)
(PAL+3)
```

Symbol

This operand is written as a string of up to eight characters, the first of which must be alphabetic. Commas and blanks are not permitted. Symbols beginning with a dollar sign ($) are not permitted either, because such symbols are reserved for system use. Examples of symbols are:

```
LEE
MARGIE3
BIL8SAM
DEBDEB
```

Decimal integer

These operands are written as decimal numbers with up to eight digits, e.g. 5, 31, 127 etc.

Absolute expression

An absolute expression may be an absolute term or any arithmetic combination of absolute terms. An absolute term may be an absolute symbol or any self-defining term. All arithmetic operators are permitted between absolute terms.

An absolute expression may comprise relative terms or relative terms combined with absolute terms only under the following conditions:

1. The expression must comprise an even number of relative terms.
2. The relative terms must be paired, and each pair must have the same relative attribute, i.e. the terms must appear in the same program section in the ASSEMBLER code. Each pair must comprise terms with opposite signs. The term pairs need not be contiguous, e.g. RT+AT=RT, where RT is relative and AT is absolute.
3. A relative term may not occur in a multiplication or division.

Pairing of relative terms (with opposite signs and the same relative attribute) cancels the effect of relocation. The value represented by the paired terms remains constant even if the program is relocated.

For example, A in the absolute expression A-Y+X is an absolute term, while X and Y are relative terms with the same relative attribute. If A=50, Y=25 and X=10, the value of the expression is 35. If X and Y are relocated by 100, their values become 125 and 110, but the value of the expression is still 35 (50-125+110). An absolute expression is reduced to a single absolute value.

In the following examples of absolute expressions, JOE and SAM are relative and defined in the same program section, while BERNY and DAVE are absolute:

```
BERNY+DAVE-83
JOE-SAM
DAVE*4+BERNY
```

Code

A code is written like a macro, e.g.

Name	Operation	Operands
[symbol]	FTBAL	target-code

target-code specifies the desired action:

GC: ground contact
GL: goal
HT: half-time

The macro could be written in a program as follows:

```
SAM     FTBAL   GC
        FTBAL   GL
UME     FTBAL   HT
```

Text

A text operand is written as a sequence of alphanumeric characters enclosed in apostrophes. Blanks and special characters are permitted. Text operands may not contain more than 127 characters including the quotes, e.g.

```
'AREA,PCB,132,    ,1256'
```

Character

The character operand is written as a character string. Commas and blanks are not permitted, and the character string must not be enclosed in apostrophes. Example:

```
CUBTDAVE+HEINZ+JOHN*83OPMOT
```

7.5 Control blocks and feedback messages: names of fields and values

Macro YDDACB

The macro YDDACB enables the user to create a (dummy) section (CSECT or DSECT) for the ACB control block.

Format

Name	Operation	Operands
[symbol]	YDDACB	[{D / C}] [,prefix] [,EQU={Y / N}]

Operands

symbol specifies the symbolic address for the macro. If no address is specified, the address "YDDACB" is generated.

specifies what is to be generated:

{**D** / C}

D requests the generation of a dummy section.

C requests the generation of a control section.

prefix specifies a string of up to 4 characters which is to precede the name. In the absence of this entry the character string "YDDA" will be used in each case to prefix the symbolic name.

EQU=Y indicates that symbolic values are assigned to all possible elements in the feedback field (see also YDDFDB).

EQU=N No symbolic values are assigned.

Dummy section DSECT for the ACB control block

```
                        EXTERNAL SYMBOL DICTIONARY
          START
  YDDACB  YDDACB D,EQU=Y
1 YDDACB  DSECT
1 *
1 *              HEADER OF CONTROL BLOCK (ACB)
1 *
1 YDDAHDTY DC    CL3'ACB'                TYPE OF CONTROL BLOCK
1 YDDAHDD  DC    CL1'D'                  DCAM CONTROL BLOCK
1 YDDAHDCB DC    '07'                  DCAM INTERFACE VERSION#
1 YDDAHDLN DC    AL2(YDDALEN-(*+2-YDDAHDTY)) LENGTH OF BODY OF CNT. BLOCK
1 *
1 *              BODY OF CONTROL BLOCK (ACB)
1 *
1 YDDAAPPN DS    A                       A(APPNAME)
1 YDDADISN DS    A                       A(DISNAME)
1 YDDAENB  DS    A                       A(ENB)
1 YDDAAID  DS    F                       APPLICATION IDENTIFIER (AID)
1 *
1 YDDAAT1  DS    XL1                     BYTE 1 FOR ATTR
1 *
1 YDDAATSH EQU   X'80'                   ATTR = NSHARE (IF BIT=1: N..!)
1 YDDAATLO EQU   X'40'                   ATTR = NLOGON (IF BIT=1: N..!)
1 YDDAATDI EQU   X'20'                   ATTR = NDISCO (IF BIT=1: N..!)
1 YDDAATIS EQU   X'02'                   ATTR = ISO
1 YDDAAHNM EQU   X'04'                   ATTR = HOSTNAM
1 *
1 YDDAATTA DS    XL1                     BYTE 2 FOR ATTR
1 *
1 YDDARECT EQU   X'04'                   ATTR = REQTASK
1 YDDAPRIT EQU   X'02'                   ATTR = PRIMTASK
1 YDDANOTK EQU   X'01'                   ATTR = NOTACK
1 *
1 YDDAVERI DS    XL1                     BYTE FOR VERIFY
1 *
1 YDDANO   EQU   X'01'                   VERIFY = NO
1 YDDAPRIM EQU   X'02'                   VERIFY = PRIMARY
1 YDDASEC  EQU   X'04'                   VERIFY = SECONDARY
1 *
1 YDDAOPT  DS    CL1
1 *
1 YDDAPERM EQU   X'02'                   LINKMOD = PERM
1 YDDAACL5 EQU   X'04'                   FOR INTERNAL USE
1 YDDADCL5 EQU   X'08'                       - - " - -
1 *
1 YDDALINK DS    A                       A(LINK)
1 YDDALGPA DS    A                       LOGPASS
1 YDDAPSSO DS    F                       USEPASS
1 YDDAPWO  DS    F                       USEPW
1 *
1 YDDAFDBK YDDFDB YDDA
2 YDDAFDBK DS    0F
2 *
2 ************** FEEDBACK FIELD 1 ************************************
2 YDDAFDB1 DS    XL1                     GENERAL RETURN CODE (IN R15 TOO)
2 *
```

```
2 YDDASUCC EQU   X'00'                  REQUEST SUCCESSFULLY
2 YDDASUWA EQU   X'04'                  REQUEST COMPLETED WITH WARNING
2 YDDARAPS EQU   X'08'                  REQUEST REJ. DUE TO APP. STATE
2 YDDARPTS EQU   X'0C'                  REQUEST REJ. DUE TO PART. STATE
2 YDDARDCS EQU   X'10'                  REQUEST REJ. DUE TO DCS STATE
2 YDDAINRU EQU   X'14'                  INVALID REQUEST USAGE
2 YDDARPAR EQU   X'18'                  REQUEST REJ. DUE TO BAD PARAM
2 YDDABCBR EQU   X'20'                  BAD CONTROL BLOCK/PL (REFERENCE)
2 YDDACONS EQU   X'24'                  CANNOT CONNECT TO SS
2 YDDASYSX EQU   X'CC'                  RESERVED FOR SYSTEM EXITS
2 *
2 ************** FEEDBACK FIELD 2 ************************************
2 YDDAFDB2 DS    XL1                    REASON FOR REJECTION
2 *                                     X'**NN****'
2 YDDAFD2N EQU   X'00'                  NO INDICATION IN FDBK-FIELD 2
2 *
2 *              THE FOLLOWING EQUATES ARE VALID IF RAPS IS SET
2 *              IN FEEDBACK FIELD 1    X'08NN****'
2 YDDANOTO EQU   X'04'                  APPLICATION NOT OPENED
2 YDDAALRO EQU   X'08'                  APPLICATION ALREADY OPENED
2 YDDANOSH EQU   X'0C'                  APPLICATION IS NON SHARABLE
2 YDDAONEO EQU   X'10'                  NO MORE THAN ONE OPEN IS ALLOWED
2 YDDAVERO EQU   X'14'                  WRONG OPEN FOR PRIMARY TASK
2 YDDASECO EQU   X'18'                  WRONG OPEN FOR SECONDARY TASK
2 YDDADIS  EQU   X'1C'                  DISTRIBUTION ALREADY PERMITTED
2 YDDAFCLW EQU   X'20'                  FORCED APPL. CLOSING WARNING
2 YDDAFCL  EQU   X'24'                  FORCED APPL. CLOSING
2 YDDAFCLD EQU   X'28'                  FORCED CLOSING - DCAM ERROR
2 YDDAFCEP EQU   X'2C'                  FORCED CLOSING-INV CONT BY PRIM
2 YDDAFCSE EQU   X'30'                  FORCED CLOSING-INV CONT BY SEC
2 YDDAFCLT EQU   X'34'                  FORCED CLOSING - TERM OF PRIMARY
2 YDDAFCLP EQU   X'38'                  FORCED CLOSING BY PRIMARY TASK
2 YDDAFCRS EQU   X'3C'                  FORCED CLOSING BY SECONDARY TASK
2 YDDATMAR EQU   X'40'                  TOO MANY REQUESTS PENDING
2 YDDANACT EQU   X'44'                  APPLICATION NOT ACTIVE
2 YDDAIUPW EQU   X'48'                  INVALID USEPW
2 YDDAANUM EQU   X'4C'                  TOO MANY APPL. OPENED
2 YDDANPRE EQU   X'50'                TOO MANY NON-PREDEFINED APPL./TASK
2 YDDAIRDF EQU   X'54'                  INVALID PASSWORD FOR RDF
2 YDDAOPSS EQU   X'58'                 APPL. OPENED BY ANOTHER SUBSYSTEM
2 YDDANPRD EQU   X'5C'                 TOO MANY NON-PREDEF. APPL.
2 YDDACNPA EQU   X'60'                  TOO MANY CONN./NONPREDEF.APPL.
2 YDDAISVR EQU   X'64'                  INVALID DCAMVER OF SECONDARY
2 *
2 *              THE FOLLOWING EQUATES ARE VALID IF RPTS IS SET
2 *              IN FEEDBACK FIELD 1    X'0CNN****'
2 YDDANCON EQU   X'04'                  PARTNER NOT CONNECTED TO APPL.
2 YDDAACON EQU   X'08'                  PARTNER ALREADY CONNECTED
2 YDDALOGQ EQU   X'0C'                  LOGON REQUEST QUEUED
2 YDDAIDID EQU   X'10'                  INVALID DID
2 YDDAILID EQU   X'14'                  INVALID LID
2 YDDACLUR EQU   X'18'                  CNNECTION CLOSED BY USER REQUEST
2 YDDAFDIS EQU   X'1C'                  PARTNER FORCED DISCONNECTED
2 YDDAPCSP EQU   X'20'                  PARTNER IN CS STATE PENDING
2 YDDAPCSS EQU   X'24'                  CHANGE TO CA NOT ALLOWED
2 YDDAPCST EQU   X'28'                  PARTNER IN CS FOR ANOTHER TASK
2 YDDAPCA  EQU   X'2C'                  PARTNER IN CA STATE
2 YDDAMESL EQU   X'30'                  MESSAGE IS TOO LONG
```

```
2 YDDAWRCP EQU   X'34'                  WRONG CODE POSITION
2 YDDASQUS EQU   X'38'                  SEQUENCE# ALREADY USED
2 YDDANOEX EQU   X'3C'                  EXPRESS NOT ALLOWED
2 YDDANUSD EQU   X'40'                  NO USER DATA WITH REJLOG
2 YDDAIPAR EQU   X'44'                  INVALID DEPROT/EDIT
2 YDDASYTI EQU   X'48'                  SYSTEM TIMEOUT
2 YDDAPNAV EQU   X'4C'                  PARTNER NOT AVAILABLE
2 YDDAPSTP EQU   X'50'                  PARTNER IN STOP STATE
2 YDDAPNLG EQU   X'54'                  PARTNER IN NLOGON STATE
2 YDDAILPW EQU   X'58'                  INVALID LOGPW
2 YDDAREJL EQU   X'5C'                  REQUEST REJECTED
2 YDDAPCNA EQU   X'60'                  PTNCHAR NOT ACCEPTED BY PTN
2 YDDAISSE EQU   X'64'                  INVALID STAT.SERV.ELMT DATA
2 YDDAPSTR EQU   X'68'                  PTN IN STOP; REQCON FOLLOWS
2 YDDAPNVT EQU   X'6C'                  ERR IN ACTIVATING PTN'S VTSU
2 YDDAPDED EQU   X'70'                  PARTNER ALREADY DEDICATED
2 YDDANDCG EQU   X'74'                  NO DCG
2 YDDAIDIS EQU   X'78'                  IMMEDIATELY DISCONNECTED
2 *                                     AFTER ACCEPTANCE
2 YDDASYER EQU   X'80'                  SYNTAX ERROR IN USER MSG
2 YDDANONA EQU   X'84'                  UNKNOWN AUTHORIZATION NAME
2 YDDAAPCO EQU   X'88'                  APPLICATION ALREADY CONNECTED
2 YDDAIPSW EQU   X'8C'                  INVALID PASSWORD
2 YDDARCOS EQU   X'90'                  PROPOSED GROS REJECTED BY PTN
2 YDDARPRI EQU   X'94'                  PROP. NETW.PRIO REJ'D BY PTN
2 YDDAPPRE EQU   X'98'                  PTN PROCESS ERR (X.25EVENT,..)
2 YDDARADM EQU   X'9C'                  CONN.REQ. REJECTED BY ADMIN.
2 YDDAPERD EQU   X'A0'                  PROTOCOL INCONSISTENCY BY PTN
2 YDDAEXNA EQU   X'A4'                  EXPEDITED NOT ALLOWED
2 YDDAUPER EQU   X'A8'                  UNRECOVERABLE UCON ERROR
2 YDDAUSNP EQU   X'AC'                  STATION NOT PRIVILEGED
2 YDDAUPNP EQU   X'B0'                  PROCESSOR NOT PRIVILEGED
2 YDDAUNOT EQU   X'B4'                  NO TASK FOR PW CHECK
2 YDDAUCID EQU   X'B8'                  WRONG CID FOR OP-ID
2 YDDAUIER EQU   X'BC'                  INTERNAL UCON ERROR
2 YDDAUNOE EQU   X'C0'                  NO ECRNAM ENTRY AVAILABLE
2 YDDAUVER EQU   X'C4'                  DCAM VERSION < 10
2 YDDAUNCT EQU   X'C8'                  NO CHIPCARD TERMINAL
2 YDDAUNNP EQU   X'CC'                  NOT NEW PROTOCOL
2 YDDAUNCS EQU   X'D0'                  NO CHIPCARD SUBSYSTEM
2 YDDAUKVP EQU   X'D4'                  ERROR IN KVP PROTOCOL
2 *
2 *              THE FOLLOWING EQUATES ARE VALID IF RDCS IS SET
2 *              IN FEEDBACK FIELD 1     X'10NN****'
2 YDDASHOR EQU   X'04'                  DCS SHORTAGE OF RESOURCES
2 YDDASHUT EQU   X'08'                  DCS SHUTDOWN WARNED
2 YDDAQSHU EQU   X'0C'                  DCS QUICK SHUTDOWN
2 YDDAIACT EQU   X'10'                  DCS INACTIVE
2 YDDADCSE EQU   X'14'                  DCS ERROR
2 YDDADCLK EQU   X'20'                  DCAM IS LOCKED
2 *
2 *              THE FOLLOWING EQUATES ARE VALID IF INRU IS SET
2 *              IN FEEDBACK FIELD 1     X'14NN****'
2 YDDAIRSE EQU   X'04'                  INVALID REQUEST FOR SECONDARY
2 YDDAIRNS EQU   X'08'                  NOT ALLOWED FOR NONSHARE APPL.
2 YDDAIRNL EQU   X'0C'                  NOT ALLOWED WITH ATTR. NLOGON
2 YDDAERPE EQU   X'10'                  EQUIVALENT REQUEST PENDING
2 YDDANAUT EQU   X'14'                  APPICATION NOT AUTHORIZED
```

```
2 YDDASYNQ EQU   X'18'                  SYN REQUEST ALREADY QUEUED
2 *
2 *              THE FOLLOWING EQUATES ARE VALID IF RPAR IS SET
2 *              IN FEEDBACK FIELD 1     X'18NN****'
2 YDDAIACB EQU   X'04'                  INVALID ACB ADDRESS
2 YDDAICCB EQU   X'08'                  INVALID CCB ADDRESS
2 YDDAIDCG EQU   X'0C'                  INVALID DCG ADDRESS
2 YDDAIDIP EQU   X'10'                  INVALID DIP ADDRESS
2 YDDAIENB EQU   X'14'                  INVALID ENB ADDRESS
2 YDDAIAAD EQU   X'18'                  INVALID APPNAME ADDRESS
2 YDDAIDAD EQU   X'1C'                  INVALID DISNAME ADDRESS
2 YDDAIPAD EQU   X'20'                  INVALID PTNNAME ADDRESS
2 YDDAICOI EQU   X'24'                  INVALID CONTINGENCY ID. ADDRESS
2 YDDAIEVI EQU   X'28'                  INVALID EVENT ITEM ID. ADDRESS
2 YDDAIARA EQU   X'2C'                  INVALID AREA ADDRESS
2 YDDAIAAR EQU   X'30'                  INVALID AAREA ADDRESS
2 YDDAIAPN EQU   X'34'                  INVALID APPLICATION NAME
2 YDDAIDIN EQU   X'38'                  INVALID DISTRIBUTION NAME
2 YDDAIPTN EQU   X'3C'                  INVALID PARTNER NAME
2 YDDAIPRO EQU   X'40'                  INVALID PROCESSOR NAME
2 YDDACCBA EQU   X'44'                  CCB REFERRED TO BY ASYNCHR.
2 YDDASEQH EQU   X'48'                  SEQUENCE NUMBER TOO HIGH
2 YDDAWRLN EQU   X'4C'                  AAREALN LESS THAN 8 / AREALN = 0
2 YDDAISUB EQU   X'50'                  INVALID SUBFUNCTION
2 YDDAIPRN EQU   X'54'                  INVALID PRONAME ADDRESS
2 YDDANGAR EQU   X'58'                  NEGATIVE AREALN
2 YDDAEDER EQU   X'5C'                  EDITING ERROR
2 YDDAICDL EQU   X'60'                  INVALID CODELN
2 YDDAIPCL EQU   X'64'                  PTNCHLN LESS 4 BYTES
2 YDDAIPCA EQU   X'68'                  APTNCH INVALID
2 *              THE FOLLOWING EQUATE IS VALID IN CASE OF
2 *              PROBLEMS WITH ISO-APPLICATIONS
2 *
2 YDDABATR EQU   X'6C'                  CONTRADICTION ISO/ATTR
2 *
2 YDDANVCB EQU   X'70'                  VTSUCB NOT USED ON THIS CONN
2 YDDABVCB EQU   X'74'                  VTSUCB NOT ALLOWED (EDIT=USER)
2 YDDAIVCB EQU   X'78'                  INVALID VTSUCB ADDRESS
2 YDDAIROU EQU   X'7C'                  ICONSISTENT ROUT PARAM SPECIFIED
2 YDDAIRLN EQU   X'80'                  INVALID ROUTLIST
2 *
2 ************** FEEDBACK FIELD 3 ************************************
2 YDDAFDB3 DS    XL1                    INDICATORS
2 *
2 YDDAFD3N EQU   X'00'                  NO INDICATION IN FDBK-FIELD 3
2 *
2 *              THE FOLLOWING EQUATES ARE VALID IF SUCC IS SET
2 *              IN FEEDBACK FIELD 1     X'00**NN**'
2 YDDAPTSK EQU   X'00'                  TASK IS PRIMARY   (YOPEN)
2 YDDASTSK EQU   X'04'                  TASK IS SECONDARY (YOPEN)
2 *
2 YDDANORM EQU   X'00'                  MESSAGE NOT TOO LONG
2 YDDAMTRN EQU   X'04'                  MESSAGE TRUNCATED
2 YDDAMKEP EQU   X'08'                  REMAINDER OF MESSAGE IS KEPT
2 *
2 YDDARVCB EQU   X'10'                  RECEIVE OK - NO VTSUCB
2 *
2 YDDANSTA EQU   X'40'                  TERMINAL STATUS INCOMPLETE
```

```
2 YDDATSTA EQU   X'80'                   TERMINAL STATUS COMPLETED
2 *
2 *              THE FOLLOWING EQUATES ARE VALID IF SUWA IS
2 *              SET IN FEEDBACK FIELD 1 X'04**NN**'
2 YDDACODE EQU   X'04'                   CONTRADICTION IN DATA CODE
2 YDDANOIN EQU   X'08'                   NO INPUT AVAILABLE
2 YDDANOLO EQU   X'0C'                   NO LOGON REQUEST QUEUED
2 YDDATOUT EQU   X'10'                   REQUEST CANCELED BY TIMEOUT
2 YDDALDAT EQU   X'14'                   LOSS OF DATA DUE TO TIMEOUT
2 YDDALQCT EQU   X'18'                   LOGON REQUEST CANCELED - TIMEOUT
2 YDDALMTR EQU   X'20'                   LOGON MESSAGE TRUNCATED
2 YDDAEDTE EQU   X'24'                   EDIT ERROR OCCURRED
2 YDDAPTTR EQU   X'28'                   PTNCHAR TRUNCATED
2 YDDAOUTR EQU   X'2C'                   OUTPUT TRUNCATED
2 YDDAEDIV EQU   X'30'                   INVALID EDIT OPTIONS
2 YDDATRES EQU   X'34'                   REQUEST TERMINATED BY YRESET
2 YDDAILHC EQU   X'44'                   LOCAL HARDCOPY NOT ASSIGNED
2 YDDAINLC EQU   X'48'                   NEW LINE CHAR. WHILE EXTEND=Y
2 *
2 *              THE FOLLOWING EQUATES ARE VALID IF
2 *              RDCS IS SET IN FEEDBACK FIELD 1 AND
2 *              SHOR IS SET IN FEEDBACK FIELD 2  X'1004NN**'
2 YDDATRYL EQU   X'04'                   BCAM: TRY I/O LATER
2 YDDABSHO EQU   X'08'                   BCAM: SHORTAGE OF RESOURCES
2 YDDAWTGO EQU   X'0C'                   BCAM: WAIT FOR GO
2 YDDANOCB EQU   X'20'                   DCAM: NO CONTROL BLOCK AVAILABLE
2 YDDANOID EQU   X'24'                   DCAM: NO ID-ENTRY AVAILABLE
2 YDDANMEM EQU   X'28'                   DCAM: NO MEMORY AVAILABLE
2 YDDAPTSH EQU   X'2C'                   DCAM: SH.RES. AT PTN'S SYSYTEM
2 YDDANMAX EQU   X'30'                   NAME MANAGER: MAX NAME #
2 *
2 *              THE FOLLOWING EQUATES ARE VALID IF
2 *              RPAR IS SET IN FEEDBACK FIELD 1 AND
2 *              EDER IS SET IN FEEDBACK FIELD 2  X'185CNN**'
2 YDDAEDPE EQU   X'00'                   EDIT PARAM ERROR
2 YDDADVHD EQU   X'04'                   INVALID DEVICE HEADER
2 YDDAMSGL EQU   X'08'                   LENGTH OF RECEIVED MESSAGE = 0
2 YDDAVTNA EQU   X'0C'                   VTSU NOT AVAILABLE
2 YDDAEVTS EQU   X'10'                   ERROR IN VTSUCB
2 YDDAENPT EQU   X'14'                   ERROR IN NEABT PROTOCOL
2 *
2 *              THE FOLLOWING EQUATES ARE VALID IF
2 *              RDCS IS SET IN FEEDBACK FIELD 1 AND
2 *              DCSE IS SET IN FEEDBACK FIELD 2  X'1014NN**'
2 YDDABCAI EQU   X'04'                   UNEXPECTED BCINF RC
2 YDDABCAA EQU   X'08'                   UNEXPECTED APINF RC
2 YDDABCAS EQU   X'0C'                   UNEXPECTED STINF RC
2 YDDABCAO EQU   X'10'                   OTHER UNEXPECTED BCAM RC
2 *
2 *
2 ************** FEEDBACK FIELD 4 ************************************
2 YDDAFDB4 DS    XL1                     DATA INDICATORS
2 *
2 YDDAFD4N EQU   X'00'                   NO INDICATION IN FDBK-FIELD 4
2 *
2 *              THE FOLLOWING EQUATES ARE VALID IF SUCC OR SUWA
2 *              ARE SET IN FEEDBACK FIELD 1 X'00****NN'/X'04****NN'
2 YDDAMSG  EQU   X'01'                   MESSAGE
```

```
2 YDDAGO   EQU   X'02'                  GO-SIGNAL
2 YDDAPTCK EQU   X'04'                  POSITIVE TACK
2 YDDANTCK EQU   X'08'                  NEGATIVE TACK
2 YDDAEXPD EQU   X'10'                  EXPRESS DATA
2 YDDAELMT EQU   X'20'                  DATA ITEM IS A ELEMENT
2 YDDASGRP EQU   X'40'                  LAST ELEMENT OF SUBGROUP
2 YDDAGRUP EQU   X'80'                  LAST ELEMENT OF GROUP
2 YDDALNF  EQU   *-YDDAFDB1             LENGTH OF FEEDBACK INFO
2 *
2 *
2                *,YDDFDB     034    920624   55616014
1 *
1 YDDAPCN  DS    A                      A(PRONAME)
1 YDDADVER DS    Y                      DCAM FUNCTIONAL VERSION#
1 YDDAV70  EQU   X'0000'                     VERSIONS LESS/EQUAL 7.0
1 YDDAV80  EQU   X'0800'                     VERSION 8.0
1 YDDARES5 DS    10XL1'00'              RESERVED FOR FURTHER EXTENSIONS
1 YDDALEN  EQU   *-YDDAHDTY             LENGTH OF CONTROL BLOCK
1                *,YDDACB     030    910919   55616011
           END
```

Macro YDDCCB

The macro YDDCCB enables the user to create a (dummy) section (CSECT or DSECT) for the CCB control block.

Format

Name	Operation	Operands
[symbol]	YDDCCB	[{D / C}] [,prefix]

Operands

symbol specifies the symbolic address for the macro. If no address is specified, the address "YDDCCB" is generated.

{**D** / C} specifies what is to be generated:

D requests the generation of a dummy section.

C requests the generation of a control section.

prefix specifies a string of up to 4 characters which is to precede the name. In the absence of this entry the character string "YDDC" will be used in each case to prefix the symbolic name.

Dummy section DSECT for the CCB control block

```
                          EXTERNAL SYMBOL DICTIONARY
           START
  YDDCCB   YDDCCB D
1 YDDCCB   DSECT
1 *
1 *              HEADER OF CONTROL BLOCK (CCB)
1 *
1 YDDCHDTY DC    CL3'CCB'                TYPE OF CONTROL BLOCK
1 YDDCHDD  DC    CL1'D'                  DCAM CONTROL BLOCK
1 YDDCHDCB DC    '07'                  DCAM INTERFACE VERSION#
1 YDDCHDLN DC    AL2(YDDCLEN-(*+2-YDDCHDTY)) LGTH OF BODY OF CNT. BLOCK
1 *
1 *              BODY OF CONTROL BLOCK (CCB)
1 *
1 YDDCCID  DS    XL4                     CID
1 YDDCPTN  DS    A                       A(PTNNAME)
1 YDDCPAWO DS    A                       LOGPW
1 YDDCUSER DS    XL4                     USERFLD
1 YDDCPCN  DS    A                       A(PRONAME)
1 *
1 YDDCITRA DS    XL1                     BYTE 1 FOR EDITIN
1 *
1 YDDCITLI EQU   X'01'                   EDITIN = LINE
1 YDDCITPH EQU   X'02'                   EDITIN = PHYS
1 YDDCITFO EQU   X'04'                   EDITIN = FORM
1 *
1 YDDCIT2  DS    XL1                     BYTE 2 FOR EDITIN
1 *
1 YDDCISPA EQU   X'80'                   EDITIN = GETBS
1 YDDCILCA EQU   X'40'                   EDITIN = LCASE
1 YDDCIGET EQU   X'20'                   EDITIN = GETFC
1 *
1 YDDCOTRA DS    XL1                     BYTE 1 FOR EDITOUT
1 *
1 YDDCOTLI EQU   X'01'                   EDITOUT = LINE
1 YDDCOTPH EQU   X'02'                   EDITOUT = PHYS
1 YDDCOTFO EQU   X'04'                   EDITOUT = FORM
1 *
1 YDDCOCOP DS    XL1                     BYTE 2 FOR EDITOUT
1 *
1 YDDCOHCP EQU   X'80'                   EDITOUT = HCOPY
1 YDDCOHOM EQU   X'40'                   EDITOUT = HOM
1 YDDCOEXT EQU   X'20'                   EDTITOUT=EXTEND
1 YDDCOLOG EQU   X'10'                   EDTITOUT=NLOGC
1 YDDCOLAK EQU   X'08'                   EDTITOUT=LACK
1 *
1 YDDCDIP  DS    A                       A(DIP)
1 YDDCDID  DS    0F                      DID
1 YDDCDSN  DS    H
1 YDDCDLN  DS    H
1 *
1 YDDCFLG1 DS    XL1                     FLAGBYTE 1
1 YDDCMDAT EQU   X'10'                   MDATA = YES
1 *
1 YDDCTRUN EQU   X'04'                   PROC = TRUNC
1 YDDCKEEP EQU   X'02'                   PROC = KEEP
```

```
1 YDDCCBK  EQU   X'01'
1 *
1 YDDCFLG2 DS    XL1                    FLAGBYTE 2
1 *
1 *
1 YDDCPRCO EQU   X'80'                  PROC = BINARY
1 YDDCPRST EQU   X'40'                  PROC = APPSTART
1 YDDCPRSI EQU   X'20'                  PROC = SIGNAL
1 YDDCPRTE EQU   X'10'                  PROC = TERMSTAT
1 YDDCPRI1 EQU   X'02'                  PRIO = 1
1 YDDCPRI2 EQU   X'04'                  PRIO = 2
1 YDDCPRI3 EQU   X'06'                  PRIO = 3
1 *
1 YDDCFLAG DS    XL1                    ACTIVE FLAG
1 *
1 YDDCFLG3 DS    XL1                    FLAGBYTE 3
1 *
1 YDDCPDXP EQU   X'80'                  PDPROT = SYSTEM
1 YDDCPED  EQU   X'40'                  PEDIT = SYSTEM
1 YDDCPPST EQU   X'20'                  PPROC = PTNSTART
1 YDDCPCL5 EQU   X'08'                  FOR INTERNAL USE
1 YDDCACL5 EQU   X'04'                     - - " - -
1 YDDCPERM EQU   X'02'                  LINKMOD = PERM
1 YDDCDIFL EQU   X'01'                  DID IS USED
1 *
1 YDDCLINK DS    A                      A(LINK)
1 YDDCMXLN DS    Y                      MAXIMAL LENGTH OF MESSAGE
1 *
1 *
1          DS    XL1                    RESERVED
1 *
1 YDDCED   DS    XL1                    BYTE FOR EDIT
1 *
1 YDDCEDUS EQU   X'01'                  EDIT = USER
1 YDDCEDSY EQU   X'04'                  EDIT = SYSTEM
1 YDDCEDSM EQU   X'08'                  EDIT = DSSIM
1 *
1 YDDCPCH1 DS    0CL8                   PARTNER CHARACTERISTIC
1 YDDCPTYP DS    XL1                    PARTNER TYP
1 YDDCPDEV DS    XL3                    PARTNER DEVICE
1 YDDCRES  DS    XL4
1          DS    F                      RESERVE
1 YDDCAPTC DS    A                      FOR DSSIM: A(PTNCHAR)
1 YDDCLPTC DS    H                      FOR DSSIM: L'PTNCHAR
1 YDDCROTN DS    H                      ROUTENUMBER
1 YDDCRLTH DS    F                      RLTH FOR OPNCON
1 YDDCROTL DS    A                      A(ROUTELIST)
1          DS    8XL1'00'               RESERVE
1 YDDCLEN  EQU   *-YDDCHDTY              LENGTH OF CONTROL BLOCK
1                *,YDDCCB     030    910919   55616012
           END
```

Macro YDDDCG

The macro YDDDCG enables the user to create a (dummy) section (CSECT or DSECT) for the DCG control block.

Format

Name	Operation	Operands
[symbol]	YDDDCG	[{D|C}] [,prefix]

Operands

symbol specifies the symbolic address for the macro. If no address is specified, the address "YDDDCG" is generated.

{D|**C**} specifies what is to be generated:

D requests the generation of a dummy section.

C requests the generation of a control section.

prefix specifies a string of up to 4 characters which is to precede the name. In the absence of this entry the character string "YDDG" will be used in each case to prefix the symbolic name.

Dummy section DSECT for the DCG control block

```
                          EXTERNAL SYMBOL DICTIONARY
           START
  YDDDCG   YDDDCG D
1 YDDDCG   DSECT
1 *
1 *              HEADER OF CONTROL BLOCK (DCG)
1 *
1 YDDGHDTY DC    CL3'DCG'              TYPE OF CONTROL BLOCK
1 YDDGHDD  DC    CL1'D'                DCAM CONTROL BLOCK
1 YDDGHDCB DC    '07'                  DCAM INTERFACE VERSION#
1 YDDGHDLN DC    AL2(YDDGLEN-(*+2-YDDGHDTY)) LENGTH OF BODY
1 *
1 *              BODY OF CONTROL BLOCK (DCG)
1 *
1 YDDGNCOD DS    H                     NUMBER OF DISTRIBUTION CODES
1 YDDGFLG1 DS    CL1                   FLAGBYTE
1 *
1 YDDGCOM  EQU   X'80'                 CODE MODIFICATION
1 *
1 YDDGRES1 DS    CL1
1 YDDGCODE DS    0CL8                  DISTRIBUTION CODES:
1 YDDGCOD1 DS    XL8                   1. CODEVAL
1 YDDGCOD2 DS    XL8                   2. CODEVAL
1 YDDGCOD3 DS    XL8                   3. CODEVAL
1 YDDGCOD4 DS    XL8                   4. CODEVAL
1 YDDGCOD5 DS    XL8                   5. CODEVAL
1 YDDGCOD6 DS    XL8                   6. CODEVAL
1 YDDGCOD7 DS    XL8                   7. CODEVAL
1 YDDGCOD8 DS    XL8                   8. CODEVAL
1 *
1 YDDGGID  DS    0F                    IDENTIFIER OF DCG
1 YDDGGSN  DS    H
1 YDDGGLN  DS    H
1 *
1 YDDGLEN  EQU   *-YDDGHDTY             LENGTH OF CONTROL BLOCK
1                *,YDDDCG     030    910919   55616017
           END
```

Macro YDDDIP

The macro YDDDIP enables the user to create a (dummy) section (CSECT or DSECT) for the DIP control block.

Format

Name	Operation	Operands
[symbol]	YDDDIP	[{D|C}] [,prefix]

Operands

symbol specifies the symbolic address for the macro. If no address is specified, the address "YDDDIP" is generated.

{D|**C**} specifies what is to be generated:

D requests the generation of a dummy section.

C requests the generation of a control section.

prefix specifies a string of up to 4 characters which is to precede the name. In the absence of this entry the character string "YDDD" will be used in each case to prefix the symbolic name.

Dummy section DSECT for the DIP control block

```
                          EXTERNAL SYMBOL DICTIONARY
           START
  YDDDIP   YDDDIP D
1 YDDDIP   DSECT
1 *
1 *              HEADER OF CONTROL BLOCK (DIP)
1 *
1 YDDDHDTY DC    CL3'DIP'              TYPE OF CONTROL BLOCK
1 YDDDHDD  DC    CL1'D'                DCAM CONTROL BLOCK
1 YDDDHDCB DC    '07'                  DCAM INTERFACE VERSION#
1 YDDDHDLN DC    AL2(YDDDLEN-(*+2-YDDDHDTY)) LENGTH OF BODY
1 *
1 *              BODY OF CONTROL BLOCK (DIP)
1 *
1 YDDDCDPS DS    XL1                   CODEPOS
1 YDDDCDLN DS    XL1                   CODELN
1 YDDDCIND DS    XL1                   CODEIND
1 YDDDRES  DS    XL1                   RESERVED
1 *
1 YDDDDID  DS    0F                    DISTRIBUTION IDENTIFIER
1 YDDDDSN  DS    H
1 YDDDDLN  DS    H
1 *
1 YDDDDCG  DS    0F                    ADDRESSES OF DCG'S
1 YDDDDCG1 DS    F                     A(DCG1)
1 YDDDDCG2 DS    F                     A(DCG2)
1 YDDDDCG3 DS    F                     A(DCG3)
1 YDDDDCG4 DS    F                     A(DCG4)
1 YDDDDCG5 DS    F                     A(DCG5)
1 YDDDDCG6 DS    F                     A(DCG6)
1 YDDDDCG7 DS    F                     A(DCG7)
1 YDDDDCG8 DS    F                     A(DCG8)
1 YDDDDCG9 DS    F                     A(DCG9)
1 YDDDDCGA DS    F                     A(DCG10)
1 YDDDDCGB DS    F                     A(DCG11)
1 YDDDDCGC DS    F                     A(DCG12)
1 YDDDDCGD DS    F                     A(DCG13)
1 YDDDDCGE DS    F                     A(DCG14)
1 YDDDDCGF DS    F                     A(DCG15)
1 YDDDDCGG DS    F                     A(DCG16)
1 *
1 YDDDLEN  EQU   *-YDDDHDTY            LENGTH OF CONTROL BLOCK
1                *,YDDDIP     030    910919   55616018
           END
```

Macro YDDENB

The macro YDDENB enables the user to create a (dummy) section (CSECT or DSECT) for the ENB control block.

Format

Name	Operation	Operands
[symbol]	YDDENB	[{D / C}] [,prefix]

Operands

symbol specifies the symbolic address for the macro. If no address is specified, the address "YDDENB" is generated.

{D / **C**} specifies what is to be generated:

D requests the generation of a dummy section.

C requests the generation of a control section.

prefix specifies a string of up to 4 characters which is to precede the name. In the absence of this entry the character string "YDDE" will be used in each case to prefix the symbolic name.

Dummy section DSECT for the ENB control block

```
                          EXTERNAL SYMBOL DICTIONARY
           START
  YDDENB   YDDENB D
1 YDDENB   DSECT
1 *
1 *              HEADER OF CONTROL BLOCK (ENB)
1 *
1 YDDEHDTY DC    CL3'ENB'                TYPE OF CONTROL BLOCK
1 YDDEHDD  DC    CL1'D'                  DCAM CONTROL BLOCK
1 YDDEHDCB DC    '07'                  DCAM INTERFACE VERSION#
1 YDDEHDLN DC    AL2(YDDELEN-(*+2-YDDEHDTY)) LENGTH OF BODY OF CNT. BLOCK
1 *
1 *              BODY OF CONTROL BLOCK (ENB)
1 *
1 YDDECEND DS    A                       A(COMEND CONTINGENCY ID)
1 YDDEEXPR DS    A                       A(EXPR   CONTINGENCY ID)
1 YDDELOGO DS    A                       A(LOGON  CONTINGENCY ID)
1 YDDELOSC DS    A                       A(LOSCON CONTINGENCY ID)
1 YDDEPROC DS    A                       A(PROCON CONTINGENCY ID)
1 YDDETACK DS    A                       A(TACK   CONTINGENCY ID)
1 YDDESECO DS    A                       A(SECOND CONTINGENCY ID)
1 *
1 YDDELEN  EQU   *-YDDEHDTY               LENGTH OF CONTROL BLOCK
1                *,YDDENB     030    910919   55616016
           END
```

Macro YDDRPB

The macro YDDRPB enables the user to create a (dummy) section (CSECT or DSECT) for the RPB control block.

Format

Name	Operation	Operands
[symbol]	YDDRPB	[{D / C}] [,prefix] [,EQU={Y / N}]

Operands

symbol specifies the symbolic address for the macro. If no address is specified, the address "YDDRPB" is generated.

specifies what is to be generated:

{D / **C**}

D requests the generation of a dummy section.

C requests the generation of a control section.

prefix specifies a string of up to 4 characters which is to precede the name. In the absence of this entry the character string "YDDR" will be used in each case to prefix the symbolic name.

EQU=Y indicates that symbolic values are assigned to all possible elements in the feedback field (see also YDDFDB).

EQU=N No symbolic values are assigned.

Dummy section DSECT for the RPB control block

```
                          EXTERNAL SYMBOL DICTIONARY
          START
  YDDRPB  YDDRPB D,EQU=Y
1 YDDRPB  DSECT
1 *
1 *              HEADER OF CONTROL BLOCK (RPB)
1 *
1 YDDRHDTY DC    CL3'RPB'                TYPE OF CONTROL BLOCK
1 YDDRHDD  DC    CL1'D'                  DCAM CONTROL BLOCK
1 YDDRHDCB DC    '07'                  DCAM INTERFACE VERSION#
1 YDDRHDLN DC    AL2(YDDRLEN-(*+2-YDDRHDTY)) LGTH OF BODY OF CNT. BLOCK
1 *
1 *              BODY OF CONTROL BLOCK (RPB)
1 *
1 YDDRRQT  DS    XL1                     REQUEST TYPE:
1 *
1 YDDRNORQ EQU   X'00'                   NO REQUEST
1 YDDROPNC EQU   X'01'                   OPEN CONNECTION REQUEST
1 YDDRCLSC EQU   X'02'                   CLOSE CONNECTION REQUEST
1 YDDRRSET EQU   X'03'                   RESET REQUEST
1 YDDRREJC EQU   X'04'                   REJECT CONNECTION REQUEST
1 YDDRSETL EQU   X'05'                   SET LOGON REQUEST
1 YDDRCHAN EQU   X'06'                   CHANGE REQUEST
1 YDDRINQU EQU   X'07'                   INQUIRE REQUEST
1 YDDRPMIT EQU   X'08'                   PERMIT REQUEST
1 YDDRFBID EQU   X'09'                   FORBID REQUEST
1 YDDRSD   EQU   X'0A'                   SEND REQUEST
1 YDDRRC   EQU   X'0B'                   RECEIVE REQUEST
1 YDDRSDRC EQU   X'0C'                   SEND/RECEIVE REQUEST
1 YDDRSESS EQU   X'0D'                   SESSION REQUEST
1 *
1 YDDRFLAG DS    XL1                     STATUS BYTE FOR CONTROL BLOCK
1 *
1 YDDRACTI EQU   X'01'                   RPB IS USED BY REQUEST
1 *
1 YDDRSWIT DS    XL1                     USED PARAMETER FOR R EQUEST
1 *
1 YDDRCIFL EQU   X'80'                   CID IS USED
1 YDDRAIFL EQU   X'40'                   AID IS USED
1 *
1 YDDROPC1 DS    XL1                     BYTE 1 FOR OPTCD
1 *
1 YDDRSPEC EQU   X'80'                   OPTCD = SPEC
1 YDDRCS   EQU   X'40'                   OPTCD = CS
1 YDDRQ    EQU   X'20'                   OPTCD = Q
1 YDDRASY  EQU   X'10'                   OPTCD = ASY
1 YDDRACQ  EQU   X'08'                   OPTCD = ACQUIRE
1 YDDRPASS EQU   X'04'
1 YDDRRLRQ EQU   X'02'
1 YDDRSTAR EQU   X'01'                   OPTCD = START
1 *
1 YDDROPC2 DS    XL1                     BYTE 2 FOR OPTCD
1 *
1 YDDRPCHA EQU   X'00'                   OPTCD = PTNCHAR
1 YDDRLOMS EQU   X'01'                   OPTCD = REQLOGON
1 YDDRTOPL EQU   X'02'                   OPTCD = TOPLOGON
```

```
1 YDDRCOUN EQU   X'03'                  OPTCD = COUNTPN
1 YDDRASTA EQU   X'04'                  OPTCD = APPSTAT
1 YDDRCIDX EQU   X'05'                  OPTCD = CIDXLATE
1 YDDRNAMX EQU   X'06'                  OPTCD = NAMXLATE
1 YDDRMODX EQU   X'09'                  OPTCD = MODXLATE
1 YDDRVTCB EQU   X'0A'                  OPTCD = VTSUCB
1 YDDRMONC EQU   X'0B'                  OPTCD = MONCHARS
1 YDDRPOTE EQU   X'0C'                  OPTCD = PEROTERM
1 YDDRBTIN EQU   X'0D'                  OPTCD = BTERMINF
1 *
1 YDDROPC3 DS    XL1                    BYTE 3 FOR OPTCD
1 *
1 YDDRELEM EQU   X'04'                  OPTCD = ELEMENT
1 YDDRSUBG EQU   X'02'                  OPTCD = SUBGROUP
1 YDDRGRP  EQU   X'01'                  OPTCD = GROUP
1 *
1 YDDROPC4 DS    XL1                    BYTE 4 FOR OPTCD
1 *
1 YDDRTRUN EQU   X'04'                  OPTCD = TRUNC
1 YDDRKEEP EQU   X'02'                  OPTCD = KEEP
1 YDDRCCBT EQU   X'01'                  OPTCD = CCBTK
1 *
1 YDDROPC5 DS    XL1                    BYTE 5 FOR OPTCD
1 *
1 YDDREXPR EQU   X'80'                  OPTCD = EXPR
1 YDDRTACK EQU   X'40'                  OPTCD = TACK
1 YDDRBELL EQU   X'20'                  OPTCD = BELL
1 YDDRFHSY EQU   X'10'                  FHS = YES (FOR DCUS ONLY !)
1 *
1 YDDRACB  DS    A                      A(ACB)
1 YDDRAID  DS    XL4                    APPLICATION IDENTIFIER (AID)
1 YDDRCCB  DS    A                      A(CCB)
1 YDDRCID  DS    XL4                    CONNECTION IDENTIFIER
1 *
1 YDDRLID  DS    XL4                    LOGON IDENTIFIER
1 YDDRAR   DS    A                      A(AREA)
1 YDDRARLN DS    F                      AREALN
1 YDDRAA   DS    A                      A(AAREA)
1 YDDRAALN DS    F                      AAREALN
1 YDDRARCL DS    F                      LENGTH OF DATA IN AAREA (ARECLN)
1 YDDRUSER DS    F                      USER FIELD
1 YDDRTOVA DS    H                      TOVAL
1 YDDRTCKN DS    H                      SEQUENCE# OF RECEIVED TACK
1 YDDRSQN  DS    H                      SEQUENCE# FOR OUTPUT (SEQNO)
1 YDDRASQN DS    H                      SEQUENCE# FOR INPUT (ASEQNO)
1 YDDREID  DS    A                      A( EVENT ITEM IDENTIFIER )
1 YDDREIDR DS    F                      EIDREF1
1 *
1 YDDRFDBK YDDFDB YDDR
2 YDDRFDBK DS    0F
2 *
2 ************** FEEDBACK FIELD 1 ************************************
2 YDDRFDB1 DS    XL1                    GENERAL RETURN CODE (IN R15 TOO)
2 *
2 YDDRSUCC EQU   X'00'                  REQUEST SUCCESSFULLY
2 YDDRSUWA EQU   X'04'                  REQUEST COMPLETED WITH WARNING
2 YDDRRAPS EQU   X'08'                  REQUEST REJ. DUE TO APP. STATE
2 YDDRRPTS EQU   X'0C'                  REQUEST REJ. DUE TO PART. STATE
```

```
2 YDDRRDCS EQU   X'10'                  REQUEST REJ. DUE TO DCS STATE
2 YDDRINRU EQU   X'14'                  INVALID REQUEST USAGE
2 YDDRRPAR EQU   X'18'                  REQUEST REJ. DUE TO BAD PARAM
2 YDDRBCBR EQU   X'20'                  BAD CONTROL BLOCK/PL (REFERENCE)
2 YDDRCONS EQU   X'24'                  CANNOT CONNECT TO SS
2 YDDRSYSX EQU   X'CC'                  RESERVED FOR SYSTEM EXITS
2 *
2 ************** FEEDBACK FIELD 2 ************************************
2 YDDRFDB2 DS    XL1                    REASON FOR REJECTION
2 *                                     X'**NN****'
2 YDDRFD2N EQU   X'00'                  NO INDICATION IN FDBK-FIELD 2
2 *
2 *              THE FOLLOWING EQUATES ARE VALID IF RAPS IS SET
2 *              IN FEEDBACK FIELD 1    X'08NN****'
2 YDDRNOTO EQU   X'04'                  APPLICATION NOT OPENED
2 YDDRALRO EQU   X'08'                  APPLICATION ALREADY OPENED
2 YDDRNOSH EQU   X'0C'                  APPLICATION IS NON SHARABLE
2 YDDRONEO EQU   X'10'                  NO MORE THAN ONE OPEN IS ALLOWED
2 YDDRVERO EQU   X'14'                  WRONG OPEN FOR PRIMARY TASK
2 YDDRSECO EQU   X'18'                  WRONG OPEN FOR SECONDARY TASK
2 YDDRDIS  EQU   X'1C'                  DISTRIBUTION ALREADY PERMITTED
2 YDDRFCLW EQU   X'20'                  FORCED APPL. CLOSING WARNING
2 YDDRFCL  EQU   X'24'                  FORCED APPL. CLOSING
2 YDDRFCLD EQU   X'28'                  FORCED CLOSING - DCAM ERROR
2 YDDRFCEP EQU   X'2C'                  FORCED CLOSING-INV CONT BY PRIM
2 YDDRFCSE EQU   X'30'                  FORCED CLOSING-INV CONT BY SEC
2 YDDRFCLT EQU   X'34'                  FORCED CLOSING - TERM OF PRIMARY
2 YDDRFCLP EQU   X'38'                  FORCED CLOSING BY PRIMARY TASK
2 YDDRFCRS EQU   X'3C'                  FORCED CLOSING BY SECONDARY TASK
2 YDDRTMAR EQU   X'40'                  TOO MANY REQUESTS PENDING
2 YDDRNACT EQU   X'44'                  APPLICATION NOT ACTIVE
2 YDDRIUPW EQU   X'48'                  INVALID USEPW
2 YDDRANUM EQU   X'4C'                  TOO MANY APPL. OPENED
2 YDDRNPRE EQU   X'50'                 TOO MANY NON-PREDEFINED APPL./TASK
2 YDDRIRDF EQU   X'54'                   INVALID PASSWORD FOR RDF
2 YDDROPSS EQU   X'58'                  APPL. OPENED BY ANOTHER SUBSYSTEM
2 YDDRNPRD EQU   X'5C'                  TOO MANY NON-PREDEF. APPL.
2 YDDRCNPA EQU   X'60'                   TOO MANY CONN./NONPREDEF.APPL.
2 YDDRISVR EQU   X'64'                   INVALID DCAMVER OF SECONDARY
2 *
2 *              THE FOLLOWING EQUATES ARE VALID IF RPTS IS SET
2 *              IN FEEDBACK FIELD 1    X'0CNN****'
2 YDDRNCON EQU   X'04'                  PARTNER NOT CONNECTED TO APPL.
2 YDDRACON EQU   X'08'                  PARTNER ALREADY CONNECTED
2 YDDRLOGQ EQU   X'0C'                  LOGON REQUEST QUEUED
2 YDDRIDID EQU   X'10'                  INVALID DID
2 YDDRILID EQU   X'14'                  INVALID LID
2 YDDRCLUR EQU   X'18'                  CNNECTION CLOSED BY USER REQUEST
2 YDDRFDIS EQU   X'1C'                  PARTNER FORCED DISCONNECTED
2 YDDRPCSP EQU   X'20'                  PARTNER IN CS STATE PENDING
2 YDDRPCSS EQU   X'24'                  CHANGE TO CA NOT ALLOWED
2 YDDRPCST EQU   X'28'                  PARTNER IN CS FOR ANOTHER TASK
2 YDDRPCA  EQU   X'2C'                  PARTNER IN CA STATE
2 YDDRMESL EQU   X'30'                  MESSAGE IS TOO LONG
2 YDDRWRCP EQU   X'34'                  WRONG CODE POSITION
2 YDDRSQUS EQU   X'38'                  SEQUENCE# ALREADY USED
2 YDDRNOEX EQU   X'3C'                  EXPRESS NOT ALLOWED
2 YDDRNUSD EQU   X'40'                  NO USER DATA WITH REJLOG
```

```
2 YDDRIPAR EQU   X'44'                  INVALID DEPROT/EDIT
2 YDDRSYTI EQU   X'48'                  SYSTEM TIMEOUT
2 YDDRPNAV EQU   X'4C'                  PARTNER NOT AVAILABLE
2 YDDRPSTP EQU   X'50'                  PARTNER IN STOP STATE
2 YDDRPNLG EQU   X'54'                  PARTNER IN NLOGON STATE
2 YDDRILPW EQU   X'58'                  INVALID LOGPW
2 YDDRREJL EQU   X'5C'                  REQUEST REJECTED
2 YDDRPCNA EQU   X'60'                  PTNCHAR NOT ACCEPTED BY PTN
2 YDDRISSE EQU   X'64'                  INVALID STAT.SERV.ELMT DATA
2 YDDRPSTR EQU   X'68'                  PTN IN STOP; REQCON FOLLOWS
2 YDDRPNVT EQU   X'6C'                  ERR IN ACTIVATING PTN'S VTSU
2 YDDRPDED EQU   X'70'                  PARTNER ALREADY DEDICATED
2 YDDRNDCG EQU   X'74'                  NO DCG
2 YDDRIDIS EQU   X'78'                  IMMEDIATELY DISCONNECTED
2 *                                     AFTER ACCEPTANCE
2 YDDRSYER EQU   X'80'                  SYNTAX ERROR IN USER MSG
2 YDDRNONA EQU   X'84'                  UNKNOWN AUTHORIZATION NAME
2 YDDRAPCO EQU   X'88'                  APPLICATION ALREADY CONNECTED
2 YDDRIPSW EQU   X'8C'                  INVALID PASSWORD
2 YDDRRCOS EQU   X'90'                  PROPOSED GROS REJECTED BY PTN
2 YDDRRPRI EQU   X'94'                  PROP. NETW.PRIO REJ'D BY PTN
2 YDDRPPRE EQU   X'98'                  PTN PROCESS ERR (X.25EVENT,..)
2 YDDRRADM EQU   X'9C'                  CONN.REQ. REJECTED BY ADMIN.
2 YDDRPERD EQU   X'A0'                  PROTOCOL INCONSISTENCY BY PTN
2 YDDREXNA EQU   X'A4'                  EXPEDITED NOT ALLOWED
2 YDDRUPER EQU   X'A8'                  UNRECOVERABLE UCON ERROR
2 YDDRUSNP EQU   X'AC'                  STATION NOT PRIVILEGED
2 YDDRUPNP EQU   X'B0'                  PROCESSOR NOT PRIVILEGED
2 YDDRUNOT EQU   X'B4'                  NO TASK FOR PW CHECK
2 YDDRUCID EQU   X'B8'                  WRONG CID FOR OP-ID
2 YDDRUIER EQU   X'BC'                  INTERNAL UCON ERROR
2 YDDRUNOE EQU   X'C0'                  NO ECRNAM ENTRY AVAILABLE
2 YDDRUVER EQU   X'C4'                  DCAM VERSION < 10
2 YDDRUNCT EQU   X'C8'                  NO CHIPCARD TERMINAL
2 YDDRUNNP EQU   X'CC'                  NOT NEW PROTOCOL
2 YDDRUNCS EQU   X'D0'                  NO CHIPCARD SUBSYSTEM
2 YDDRUKVP EQU   X'D4'                  ERROR IN KVP PROTOCOL
2 *
2 *              THE FOLLOWING EQUATES ARE VALID IF RDCS IS SET
2 *              IN FEEDBACK FIELD 1     X'10NN****'
2 YDDRSHOR EQU   X'04'                  DCS SHORTAGE OF RESOURCES
2 YDDRSHUT EQU   X'08'                  DCS SHUTDOWN WARNED
2 YDDRQSHU EQU   X'0C'                  DCS QUICK SHUTDOWN
2 YDDRIACT EQU   X'10'                  DCS INACTIVE
2 YDDRDCSE EQU   X'14'                  DCS ERROR
2 YDDRDCLK EQU   X'20'                  DCAM IS LOCKED
2 *
2 *              THE FOLLOWING EQUATES ARE VALID IF INRU IS SET
2 *              IN FEEDBACK FIELD 1     X'14NN****'
2 YDDRIRSE EQU   X'04'                  INVALID REQUEST FOR SECONDARY
2 YDDRIRNS EQU   X'08'                  NOT ALLOWED FOR NONSHARE APPL.
2 YDDRIRNL EQU   X'0C'                  NOT ALLOWED WITH ATTR. NLOGON
2 YDDRERPE EQU   X'10'                  EQUIVALENT REQUEST PENDING
2 YDDRNAUT EQU   X'14'                  APPICATION NOT AUTHORIZED
2 YDDRSYNQ EQU   X'18'                  SYN REQUEST ALREADY QUEUED
2 *
2 *              THE FOLLOWING EQUATES ARE VALID IF RPAR IS SET
2 *              IN FEEDBACK FIELD 1     X'18NN****'
```

```
2 YDDRIACB EQU   X'04'                  INVALID ACB ADDRESS
2 YDDRICCB EQU   X'08'                  INVALID CCB ADDRESS
2 YDDRIDCG EQU   X'0C'                  INVALID DCG ADDRESS
2 YDDRIDIP EQU   X'10'                  INVALID DIP ADDRESS
2 YDDRIENB EQU   X'14'                  INVALID ENB ADDRESS
2 YDDRIAAD EQU   X'18'                  INVALID APPNAME ADDRESS
2 YDDRIDAD EQU   X'1C'                  INVALID DISNAME ADDRESS
2 YDDRIPAD EQU   X'20'                  INVALID PTNNAME ADDRESS
2 YDDRICOI EQU   X'24'                  INVALID CONTINGENCY ID. ADDRESS
2 YDDRIEVI EQU   X'28'                  INVALID EVENT ITEM ID. ADDRESS
2 YDDRIARA EQU   X'2C'                  INVALID AREA ADDRESS
2 YDDRIAAR EQU   X'30'                  INVALID AAREA ADDRESS
2 YDDRIAPN EQU   X'34'                  INVALID APPLICATION NAME
2 YDDRIDIN EQU   X'38'                  INVALID DISTRIBUTION NAME
2 YDDRIPTN EQU   X'3C'                  INVALID PARTNER NAME
2 YDDRIPRO EQU   X'40'                  INVALID PROCESSOR NAME
2 YDDRCCBA EQU   X'44'                  CCB REFERRED TO BY ASYNCHR.
2 YDDRSEQH EQU   X'48'                  SEQUENCE NUMBER TOO HIGH
2 YDDRWRLN EQU   X'4C'                  AAREALN LESS THAN 8 / AREALN = 0
2 YDDRISUB EQU   X'50'                  INVALID SUBFUNCTION
2 YDDRIPRN EQU   X'54'                  INVALID PRONAME ADDRESS
2 YDDRNGAR EQU   X'58'                  NEGATIVE AREALN
2 YDDREDER EQU   X'5C'                  EDITING ERROR
2 YDDRICDL EQU   X'60'                  INVALID CODELN
2 YDDRIPCL EQU   X'64'                  PTNCHLN LESS 4 BYTES
2 YDDRIPCA EQU   X'68'                  APTNCH INVALID
2 *              THE FOLLOWING EQUATE IS VALID IN CASE OF
2 *              PROBLEMS WITH ISO-APPLICATIONS
2 *
2 YDDRBATR EQU   X'6C'                  CONTRADICTION ISO/ATTR
2 *
2 YDDRNVCB EQU   X'70'                  VTSUCB NOT USED ON THIS CONN
2 YDDRBVCB EQU   X'74'                  VTSUCB NOT ALLOWED (EDIT=USER)
2 YDDRIVCB EQU   X'78'                  INVALID VTSUCB ADDRESS
2 YDDRIROU EQU   X'7C'                  ICONSISTENT ROUT PARAM SPECIFIED
2 YDDRIRLN EQU   X'80'                  INVALID ROUTLIST
2 *
2 ************** FEEDBACK FIELD 3 ************************************
2 YDDRFDB3 DS    XL1                    INDICATORS
2 *
2 YDDRFD3N EQU   X'00'                  NO INDICATION IN FDBK-FIELD 3
2 *
2 *              THE FOLLOWING EQUATES ARE VALID IF SUCC IS SET
2 *              IN FEEDBACK FIELD 1    X'00**NN**'
2 YDDRPTSK EQU   X'00'                  TASK IS PRIMARY   (YOPEN)
2 YDDRSTSK EQU   X'04'                  TASK IS SECONDARY (YOPEN)
2 *
2 YDDRNORM EQU   X'00'                  MESSAGE NOT TOO LONG
2 YDDRMTRN EQU   X'04'                  MESSAGE TRUNCATED
2 YDDRMKEP EQU   X'08'                  REMAINDER OF MESSAGE IS KEPT
2 *
2 YDDRRVCB EQU   X'10'                  RECEIVE OK - NO VTSUCB
2 *
2 YDDRNSTA EQU   X'40'                  TERMINAL STATUS INCOMPLETE
2 YDDRTSTA EQU   X'80'                  TERMINAL STATUS COMPLETED
2 *
2 *              THE FOLLOWING EQUATES ARE VALID IF SUWA IS
2 *              SET IN FEEDBACK FIELD 1 X'04**NN**'
```

```
2 YDDRCODE EQU   X'04'                  CONTRADICTION IN DATA CODE
2 YDDRNOIN EQU   X'08'                  NO INPUT AVAILABLE
2 YDDRNOLO EQU   X'0C'                  NO LOGON REQUEST QUEUED
2 YDDRTOUT EQU   X'10'                  REQUEST CANCELED BY TIMEOUT
2 YDDRLDAT EQU   X'14'                  LOSS OF DATA DUE TO TIMEOUT
2 YDDRLQCT EQU   X'18'                  LOGON REQUEST CANCELED - TIMEOUT
2 YDDRLMTR EQU   X'20'                  LOGON MESSAGE TRUNCATED
2 YDDREDTE EQU   X'24'                  EDIT ERROR OCCURRED
2 YDDRPTTR EQU   X'28'                  PTNCHAR TRUNCATED
2 YDDROUTR EQU   X'2C'                  OUTPUT TRUNCATED
2 YDDREDIV EQU   X'30'                  INVALID EDIT OPTIONS
2 YDDRTRES EQU   X'34'                  REQUEST TERMINATED BY YRESET
2 YDDRILHC EQU   X'44'                  LOCAL HARDCOPY NOT ASSIGNED
2 YDDRINLC EQU   X'48'                  NEW LINE CHAR. WHILE EXTEND=Y
2 *
2 *              THE FOLLOWING EQUATES ARE VALID IF
2 *              RDCS IS SET IN FEEDBACK FIELD 1 AND
2 *              SHOR IS SET IN FEEDBACK FIELD 2  X'1004NN**'
2 YDDRTRYL EQU   X'04'                  BCAM: TRY I/O LATER
2 YDDRBSHO EQU   X'08'                  BCAM: SHORTAGE OF RESOURCES
2 YDDRWTGO EQU   X'0C'                  BCAM: WAIT FOR GO
2 YDDRNOCB EQU   X'20'                  DCAM: NO CONTROL BLOCK AVAILABLE
2 YDDRNOID EQU   X'24'                  DCAM: NO ID-ENTRY AVAILABLE
2 YDDRNMEM EQU   X'28'                  DCAM: NO MEMORY AVAILABLE
2 YDDRPTSH EQU   X'2C'                  DCAM: SH.RES. AT PTN'S SYSYTEM
2 YDDRNMAX EQU   X'30'                  NAME MANAGER: MAX NAME #
2 *
2 *              THE FOLLOWING EQUATES ARE VALID IF
2 *              RPAR IS SET IN FEEDBACK FIELD 1 AND
2 *              EDER IS SET IN FEEDBACK FIELD 2  X'185CNN**'
2 YDDREDPE EQU   X'00'                  EDIT PARAM ERROR
2 YDDRDVHD EQU   X'04'                  INVALID DEVICE HEADER
2 YDDRMSGL EQU   X'08'                  LENGTH OF RECEIVED MESSAGE = 0
2 YDDRVTNA EQU   X'0C'                  VTSU NOT AVAILABLE
2 YDDREVTS EQU   X'10'                  ERROR IN VTSUCB
2 YDDRENPT EQU   X'14'                  ERROR IN NEABT PROTOCOL
2 *
2 *              THE FOLLOWING EQUATES ARE VALID IF
2 *              RDCS IS SET IN FEEDBACK FIELD 1 AND
2 *              DCSE IS SET IN FEEDBACK FIELD 2  X'1014NN**'
2 YDDRBCAI EQU   X'04'                  UNEXPECTED BCINF RC
2 YDDRBCAA EQU   X'08'                  UNEXPECTED APINF RC
2 YDDRBCAS EQU   X'0C'                  UNEXPECTED STINF RC
2 YDDRBCAO EQU   X'10'                  OTHER UNEXPECTED BCAM RC
2 *
2 *
2 ************** FEEDBACK FIELD 4 ************************************
2 YDDRFDB4 DS    XL1                    DATA INDICATORS
2 *
2 YDDRFD4N EQU   X'00'                  NO INDICATION IN FDBK-FIELD 4
2 *
2 *              THE FOLLOWING EQUATES ARE VALID IF SUCC OR SUWA
2 *              ARE SET IN FEEDBACK FIELD 1 X'00****NN'/X'04****NN'
2 YDDRMSG  EQU   X'01'                  MESSAGE
2 YDDRGO   EQU   X'02'                  GO-SIGNAL
2 YDDRPTCK EQU   X'04'                  POSITIVE TACK
2 YDDRNTCK EQU   X'08'                  NEGATIVE TACK
2 YDDREXPD EQU   X'10'                  EXPRESS DATA
```

```
2 YDDRELMT EQU   X'20'                   DATA ITEM IS A ELEMENT
2 YDDRSGRP EQU   X'40'                   LAST ELEMENT OF SUBGROUP
2 YDDRGRUP EQU   X'80'                   LAST ELEMENT OF GROUP
2 YDDRLNF  EQU   *-YDDRFDB1              LENGTH OF FEEDBACK INFO
2 *
2 *
2                *,YDDFDB     034    920624   55616014
1 *
1 *
1 YDDRUREA DS    X                       ISO: USERDATA FOR REJ/DISCON
1          DS    CL3                     RESERVE
1 *
1 *        NEW PARAMETER EIDREF2 (FIRST USE IN DCM V8.9)
1 *
1 YDDREIR2 DS    F                       EIDREF2
1 *
1 *        NEW PARAMETER AVTSUCB (FIRST USE DCM V10)
1 *
1 YDDRVCBA DS    A
1          DS    8XL1'00'                RESERVE
1 *
1 YDDRLEN  EQU   *-YDDRHDTY             LENGTH OF CONTROL BLOCK
1                *,YDDRPB     030    910919   55616013
           END
```

Macro YDDFDB

The macro YDDFDB generates a 4-byte area intended for the contents of various feedback messages. A dummy section can optionally be created. The symbolic names can be used as symbolic addresses for the interpretation of feedback information.

Format

Name	Operation	Operands
[symbol]	YDDFDB	[{D / C}] [,prefix] [,{FDBK / CONT / CBRC}]

Operands

symbol specifies the symbolic address for the macro. If no address is specified, the address "YDDFFDBK" is generated.

specifies what is to be generated:

{D / C}

D requests the generation of a dummy section.

C requests the generation of an area 4 bytes long.

prefix specifies a string of up to 4 characters which is to precede the name. In the absence of this entry the character string "YDDF" will be used in each case to prefix the symbolic name.

[FDBK]
{CONT}
[CBRC]

Addresses can be generated for one of the three different feedback messages.

FDBK specifies that the names for feedback messages are to be generated after termination of a macro call (contents of the FDBK field or of register 15).

CONT requests the generation of names which reference the contents of register 6 (for LOSCON) or register 4 (for COMEND and SECOND). The values are ascertained as soon as DCAM has initiated a contingency routine of the type specified.

CBRC requests the generation of names which refer to the contents of register 15. The values are ascertained after completion of one of the following macro calls: YGENCB; YMODCB; YTESTCB; YSHOWCB.

The following pages show expansions of the three variants of the YDDFDB macro.

Variant 1

```
                          EXTERNAL SYMBOL DICTIONARY
          START
  YDDFDB  YDDFDB D
1 YDDFDB  DSECT
1 *
1 ************** FEEDBACK FIELD 1 ************************************
1 YDDFFDB1 DS    XL1                    GENERAL RETURN CODE (IN R15 TOO)
1 *
1 YDDFSUCC EQU   X'00'                  REQUEST SUCCESSFULLY
1 YDDFSUWA EQU   X'04'                  REQUEST COMPLETED WITH WARNING
1 YDDFRAPS EQU   X'08'                  REQUEST REJ. DUE TO APP. STATE
1 YDDFRPTS EQU   X'0C'                  REQUEST REJ. DUE TO PART. STATE
1 YDDFRDCS EQU   X'10'                  REQUEST REJ. DUE TO DCS STATE
1 YDDFINRU EQU   X'14'                  INVALID REQUEST USAGE
1 YDDFRPAR EQU   X'18'                  REQUEST REJ. DUE TO BAD PARAM
1 YDDFBCBR EQU   X'20'                  BAD CONTROL BLOCK/PL (REFERENCE)
1 YDDFCONS EQU   X'24'                  CANNOT CONNECT TO SS
1 YDDFSYSX EQU   X'CC'                  RESERVED FOR SYSTEM EXITS
1 *
1 ************** FEEDBACK FIELD 2 ************************************
1 YDDFFDB2 DS    XL1                    REASON FOR REJECTION
1 *                                     X'**NN****'
1 YDDFFD2N EQU   X'00'                  NO INDICATION IN FDBK-FIELD 2
1 *
1 *              THE FOLLOWING EQUATES ARE VALID IF RAPS IS SET
1 *              IN FEEDBACK FIELD 1    X'08NN****'
1 YDDFNOTO EQU   X'04'                  APPLICATION NOT OPENED
1 YDDFALRO EQU   X'08'                  APPLICATION ALREADY OPENED
1 YDDFNOSH EQU   X'0C'                  APPLICATION IS NON SHARABLE
1 YDDFONEO EQU   X'10'                  NO MORE THAN ONE OPEN IS ALLOWED
1 YDDFVERO EQU   X'14'                  WRONG OPEN FOR PRIMARY TASK
```

```
1 YDDFSECO EQU   X'18'                  WRONG OPEN FOR SECONDARY TASK
1 YDDFDIS  EQU   X'1C'                  DISTRIBUTION ALREADY PERMITTED
1 YDDFFCLW EQU   X'20'                  FORCED APPL. CLOSING WARNING
1 YDDFFCL  EQU   X'24'                  FORCED APPL. CLOSING
1 YDDFFCLD EQU   X'28'                  FORCED CLOSING - DCAM ERROR
1 YDDFFCEP EQU   X'2C'                  FORCED CLOSING-INV CONT BY PRIM
1 YDDFFCSE EQU   X'30'                  FORCED CLOSING-INV CONT BY SEC
1 YDDFFCLT EQU   X'34'                  FORCED CLOSING - TERM OF PRIMARY
1 YDDFFCLP EQU   X'38'                  FORCED CLOSING BY PRIMARY TASK
1 YDDFFCRS EQU   X'3C'                  FORCED CLOSING BY SECONDARY TASK
1 YDDFTMAR EQU   X'40'                  TOO MANY REQUESTS PENDING
1 YDDFNACT EQU   X'44'                  APPLICATION NOT ACTIVE
1 YDDFIUPW EQU   X'48'                  INVALID USEPW
1 YDDFANUM EQU   X'4C'                  TOO MANY APPL. OPENED
1 YDDFNPRE EQU   X'50'                 TOO MANY NON-PREDEFINED APPL./TASK
1 YDDFIRDF EQU   X'54'                  INVALID PASSWORD FOR RDF
1 YDDFOPSS EQU   X'58'                 APPL. OPENED BY ANOTHER SUBSYSTEM
1 YDDFNPRD EQU   X'5C'                 TOO MANY NON-PREDEF. APPL.
1 YDDFCNPA EQU   X'60'                  TOO MANY CONN./NONPREDEF.APPL.
1 YDDFISVR EQU   X'64'                  INVALID DCAMVER OF SECONDARY
1 *
1 *              THE FOLLOWING EQUATES ARE VALID IF RPTS IS SET
1 *              IN FEEDBACK FIELD 1     X'0CNN****'
1 YDDFNCON EQU   X'04'                  PARTNER NOT CONNECTED TO APPL.
1 YDDFACON EQU   X'08'                  PARTNER ALREADY CONNECTED
1 YDDFLOGQ EQU   X'0C'                  LOGON REQUEST QUEUED
1 YDDFIDID EQU   X'10'                  INVALID DID
1 YDDFILID EQU   X'14'                  INVALID LID
1 YDDFCLUR EQU   X'18'                  CNNECTION CLOSED BY USER REQUEST
1 YDDFFDIS EQU   X'1C'                  PARTNER FORCED DISCONNECTED
1 YDDFPCSP EQU   X'20'                  PARTNER IN CS STATE PENDING
1 YDDFPCSS EQU   X'24'                  CHANGE TO CA NOT ALLOWED
1 YDDFPCST EQU   X'28'                  PARTNER IN CS FOR ANOTHER TASK
1 YDDFPCA  EQU   X'2C'                  PARTNER IN CA STATE
1 YDDFMESL EQU   X'30'                  MESSAGE IS TOO LONG
1 YDDFWRCP EQU   X'34'                  WRONG CODE POSITION
1 YDDFSQUS EQU   X'38'                  SEQUENCE# ALREADY USED
1 YDDFNOEX EQU   X'3C'                  EXPRESS NOT ALLOWED
1 YDDFNUSD EQU   X'40'                  NO USER DATA WITH REJLOG
1 YDDFIPAR EQU   X'44'                  INVALID DEPROT/EDIT
1 YDDFSYTI EQU   X'48'                  SYSTEM TIMEOUT
1 YDDFPNAV EQU   X'4C'                  PARTNER NOT AVAILABLE
1 YDDFPSTP EQU   X'50'                  PARTNER IN STOP STATE
1 YDDFPNLG EQU   X'54'                  PARTNER IN NLOGON STATE
1 YDDFILPW EQU   X'58'                  INVALID LOGPW
1 YDDFREJL EQU   X'5C'                  REQUEST REJECTED
1 YDDFPCNA EQU   X'60'                  PTNCHAR NOT ACCEPTED BY PTN
1 YDDFISSE EQU   X'64'                  INVALID STAT.SERV.ELMT DATA
1 YDDFPSTR EQU   X'68'                  PTN IN STOP; REQCON FOLLOWS
1 YDDFPNVT EQU   X'6C'                  ERR IN ACTIVATING PTN'S VTSU
1 YDDFPDED EQU   X'70'                  PARTNER ALREADY DEDICATED
1 YDDFNDCG EQU   X'74'                  NO DCG
1 YDDFIDIS EQU   X'78'                  IMMEDIATELY DISCONNECTED
1 *                                     AFTER ACCEPTANCE
1 YDDFSYER EQU   X'80'                  SYNTAX ERROR IN USER MSG
1 YDDFNONA EQU   X'84'                  UNKNOWN AUTHORIZATION NAME
1 YDDFAPCO EQU   X'88'                  APPLICATION ALREADY CONNECTED
1 YDDFIPSW EQU   X'8C'                  INVALID PASSWORD
```

```
1 YDDFRCOS EQU   X'90'                  PROPOSED GROS REJECTED BY PTN
1 YDDFRPRI EQU   X'94'                  PROP. NETW.PRIO REJ'D BY PTN
1 YDDFPPRE EQU   X'98'                  PTN PROCESS ERR (X.25EVENT,..)
1 YDDFRADM EQU   X'9C'                  CONN.REQ. REJECTED BY ADMIN.
1 YDDFPERD EQU   X'A0'                  PROTOCOL INCONSISTENCY BY PTN
1 YDDFEXNA EQU   X'A4'                  EXPEDITED NOT ALLOWED
1 YDDFUPER EQU   X'A8'                  UNRECOVERABLE UCON ERROR
1 YDDFUSNP EQU   X'AC'                  STATION NOT PRIVILEGED
1 YDDFUPNP EQU   X'B0'                  PROCESSOR NOT PRIVILEGED
1 YDDFUNOT EQU   X'B4'                  NO TASK FOR PW CHECK
1 YDDFUCID EQU   X'B8'                  WRONG CID FOR OP-ID
1 YDDFUIER EQU   X'BC'                  INTERNAL UCON ERROR
1 YDDFUNOE EQU   X'C0'                  NO ECRNAM ENTRY AVAILABLE
1 YDDFUVER EQU   X'C4'                  DCAM VERSION < 10
1 YDDFUNCT EQU   X'C8'                  NO CHIPCARD TERMINAL
1 YDDFUNNP EQU   X'CC'                  NOT NEW PROTOCOL
1 YDDFUNCS EQU   X'D0'                  NO CHIPCARD SUBSYSTEM
1 YDDFUKVP EQU   X'D4'                  ERROR IN KVP PROTOCOL
1 *
1 *              THE FOLLOWING EQUATES ARE VALID IF RDCS IS SET
1 *              IN FEEDBACK FIELD 1    X'10NN****'
1 YDDFSHOR EQU   X'04'                  DCS SHORTAGE OF RESOURCES
1 YDDFSHUT EQU   X'08'                  DCS SHUTDOWN WARNED
1 YDDFQSHU EQU   X'0C'                  DCS QUICK SHUTDOWN
1 YDDFIACT EQU   X'10'                  DCS INACTIVE
1 YDDFDCSE EQU   X'14'                  DCS ERROR
1 YDDFDCLK EQU   X'20'                  DCAM IS LOCKED
1 *
1 *              THE FOLLOWING EQUATES ARE VALID IF INRU IS SET
1 *              IN FEEDBACK FIELD 1    X'14NN****'
1 YDDFIRSE EQU   X'04'                  INVALID REQUEST FOR SECONDARY
1 YDDFIRNS EQU   X'08'                  NOT ALLOWED FOR NONSHARE APPL.
1 YDDFIRNL EQU   X'0C'                  NOT ALLOWED WITH ATTR. NLOGON
1 YDDFERPE EQU   X'10'                  EQUIVALENT REQUEST PENDING
1 YDDFNAUT EQU   X'14'                  APPICATION NOT AUTHORIZED
1 YDDFSYNQ EQU   X'18'                  SYN REQUEST ALREADY QUEUED
1 *
1 *              THE FOLLOWING EQUATES ARE VALID IF RPAR IS SET
1 *              IN FEEDBACK FIELD 1    X'18NN****'
1 YDDFIACB EQU   X'04'                  INVALID ACB ADDRESS
1 YDDFICCB EQU   X'08'                  INVALID CCB ADDRESS
1 YDDFIDCG EQU   X'0C'                  INVALID DCG ADDRESS
1 YDDFIDIP EQU   X'10'                  INVALID DIP ADDRESS
1 YDDFIENB EQU   X'14'                  INVALID ENB ADDRESS
1 YDDFIAAD EQU   X'18'                  INVALID APPNAME ADDRESS
1 YDDFIDAD EQU   X'1C'                  INVALID DISNAME ADDRESS
1 YDDFIPAD EQU   X'20'                  INVALID PTNNAME ADDRESS
1 YDDFICOI EQU   X'24'                  INVALID CONTINGENCY ID. ADDRESS
1 YDDFIEVI EQU   X'28'                  INVALID EVENT ITEM ID. ADDRESS
1 YDDFIARA EQU   X'2C'                  INVALID AREA ADDRESS
1 YDDFIAAR EQU   X'30'                  INVALID AAREA ADDRESS
1 YDDFIAPN EQU   X'34'                  INVALID APPLICATION NAME
1 YDDFIDIN EQU   X'38'                  INVALID DISTRIBUTION NAME
1 YDDFIPTN EQU   X'3C'                  INVALID PARTNER NAME
1 YDDFIPRO EQU   X'40'                  INVALID PROCESSOR NAME
1 YDDFCCBA EQU   X'44'                  CCB REFERRED TO BY ASYNCHR.
1 YDDFSEQH EQU   X'48'                  SEQUENCE NUMBER TOO HIGH
1 YDDFWRLN EQU   X'4C'                  AAREALN LESS THAN 8 / AREALN = 0
```

```
1 YDDFISUB EQU   X'50'                  INVALID SUBFUNCTION
1 YDDFIPRN EQU   X'54'                  INVALID PRONAME ADDRESS
1 YDDFNGAR EQU   X'58'                  NEGATIVE AREALN
1 YDDFEDER EQU   X'5C'                  EDITING ERROR
1 YDDFICDL EQU   X'60'                  INVALID CODELN
1 YDDFIPCL EQU   X'64'                  PTNCHLN LESS 4 BYTES
1 YDDFIPCA EQU   X'68'                  APTNCH INVALID
1 *              THE FOLLOWING EQUATE IS VALID IN CASE OF
1 *              PROBLEMS WITH ISO-APPLICATIONS
1 *
1 YDDFBATR EQU   X'6C'                  CONTRADICTION ISO/ATTR
1 *
1 YDDFNVCB EQU   X'70'                  VTSUCB NOT USED ON THIS CONN
1 YDDFBVCB EQU   X'74'                  VTSUCB NOT ALLOWED (EDIT=USER)
1 YDDFIVCB EQU   X'78'                  INVALID VTSUCB ADDRESS
1 YDDFIROU EQU   X'7C'                  ICONSISTENT ROUT PARAM SPECIFIED
1 YDDFIRLN EQU   X'80'                  INVALID ROUTLIST
1 *
1 ************** FEEDBACK FIELD 3 ************************************
1 YDDFFDB3 DS    XL1                    INDICATORS
1 *
1 YDDFFD3N EQU   X'00'                  NO INDICATION IN FDBK-FIELD 3
1 *
1 *              THE FOLLOWING EQUATES ARE VALID IF SUCC IS SET
1 *              IN FEEDBACK FIELD 1     X'00**NN**'
1 YDDFPTSK EQU   X'00'                  TASK IS PRIMARY   (YOPEN)
1 YDDFSTSK EQU   X'04'                  TASK IS SECONDARY (YOPEN)
1 *
1 YDDFNORM EQU   X'00'                  MESSAGE NOT TOO LONG
1 YDDFMTRN EQU   X'04'                  MESSAGE TRUNCATED
1 YDDFMKEP EQU   X'08'                  REMAINDER OF MESSAGE IS KEPT
1 *
1 YDDFRVCB EQU   X'10'                  RECEIVE OK - NO VTSUCB
1 *
1 YDDFNSTA EQU   X'40'                  TERMINAL STATUS INCOMPLETE
1 YDDFTSTA EQU   X'80'                  TERMINAL STATUS COMPLETED
1 *
1 *              THE FOLLOWING EQUATES ARE VALID IF SUWA IS
1 *              SET IN FEEDBACK FIELD 1 X'04**NN**'
1 YDDFCODE EQU   X'04'                  CONTRADICTION IN DATA CODE
1 YDDFNOIN EQU   X'08'                  NO INPUT AVAILABLE
1 YDDFNOLO EQU   X'0C'                  NO LOGON REQUEST QUEUED
1 YDDFTOUT EQU   X'10'                  REQUEST CANCELED BY TIMEOUT
1 YDDFLDAT EQU   X'14'                  LOSS OF DATA DUE TO TIMEOUT
1 YDDFLQCT EQU   X'18'                  LOGON REQUEST CANCELED - TIMEOUT
1 YDDFLMTR EQU   X'20'                  LOGON MESSAGE TRUNCATED
1 YDDFEDTE EQU   X'24'                  EDIT ERROR OCCURRED
1 YDDFPTTR EQU   X'28'                  PTNCHAR TRUNCATED
1 YDDFOUTR EQU   X'2C'                  OUTPUT TRUNCATED
1 YDDFEDIV EQU   X'30'                  INVALID EDIT OPTIONS
1 YDDFTRES EQU   X'34'                  REQUEST TERMINATED BY YRESET
1 YDDFILHC EQU   X'44'                  LOCAL HARDCOPY NOT ASSIGNED
1 YDDFINLC EQU   X'48'                  NEW LINE CHAR. WHILE EXTEND=Y
1 *
1 *              THE FOLLOWING EQUATES ARE VALID IF
1 *              RDCS IS SET IN FEEDBACK FIELD 1 AND
1 *              SHOR IS SET IN FEEDBACK FIELD 2  X'1004NN**'
1 YDDFTRYL EQU   X'04'                  BCAM: TRY I/O LATER
```

```
1 YDDFBSHO EQU   X'08'                  BCAM: SHORTAGE OF RESOURCES
1 YDDFWTGO EQU   X'0C'                  BCAM: WAIT FOR GO
1 YDDFNOCB EQU   X'20'                  DCAM: NO CONTROL BLOCK AVAILABLE
1 YDDFNOID EQU   X'24'                  DCAM: NO ID-ENTRY AVAILABLE
1 YDDFNMEM EQU   X'28'                  DCAM: NO MEMORY AVAILABLE
1 YDDFPTSH EQU   X'2C'                  DCAM: SH.RES. AT PTN'S SYSYTEM
1 YDDFNMAX EQU   X'30'                  NAME MANAGER: MAX NAME #
1 *
1 *              THE FOLLOWING EQUATES ARE VALID IF
1 *              RPAR IS SET IN FEEDBACK FIELD 1 AND
1 *              EDER IS SET IN FEEDBACK FIELD 2  X'185CNN**'
1 YDDFEDPE EQU   X'00'                  EDIT PARAM ERROR
1 YDDFDVHD EQU   X'04'                  INVALID DEVICE HEADER
1 YDDFMSGL EQU   X'08'                  LENGTH OF RECEIVED MESSAGE = 0
1 YDDFVTNA EQU   X'0C'                  VTSU NOT AVAILABLE
1 YDDFEVTS EQU   X'10'                  ERROR IN VTSUCB
1 YDDFENPT EQU   X'14'                  ERROR IN NEABT PROTOCOL
1 *
1 *              THE FOLLOWING EQUATES ARE VALID IF
1 *              RDCS IS SET IN FEEDBACK FIELD 1 AND
1 *              DCSE IS SET IN FEEDBACK FIELD 2  X'1014NN**'
1 YDDFBCAI EQU   X'04'                  UNEXPECTED BCINF RC
1 YDDFBCAA EQU   X'08'                  UNEXPECTED APINF RC
1 YDDFBCAS EQU   X'0C'                  UNEXPECTED STINF RC
1 YDDFBCAO EQU   X'10'                  OTHER UNEXPECTED BCAM RC
1 *
1 *
1 ************** FEEDBACK FIELD 4 ************************************
1 YDDFFDB4 DS    XL1                    DATA INDICATORS
1 *
1 YDDFFD4N EQU   X'00'                  NO INDICATION IN FDBK-FIELD 4
1 *
1 *              THE FOLLOWING EQUATES ARE VALID IF SUCC OR SUWA
1 *              ARE SET IN FEEDBACK FIELD 1 X'00****NN'/X'04****NN'
1 YDDFMSG  EQU   X'01'                  MESSAGE
1 YDDFGO   EQU   X'02'                  GO-SIGNAL
1 YDDFPTCK EQU   X'04'                  POSITIVE TACK
1 YDDFNTCK EQU   X'08'                  NEGATIVE TACK
1 YDDFEXPD EQU   X'10'                  EXPRESS DATA
1 YDDFELMT EQU   X'20'                  DATA ITEM IS A ELEMENT
1 YDDFSGRP EQU   X'40'                  LAST ELEMENT OF SUBGROUP
1 YDDFGRUP EQU   X'80'                  LAST ELEMENT OF GROUP
1 YDDFLNF  EQU   *-YDDFFDB1             LENGTH OF FEEDBACK INFO
1 *
1 *
1                *,YDDFDB     034    920624   55616014
           END
```

Variant 2

```
                          EXTERNAL SYMBOL DICTIONARY
          START
  YDDFDB  YDDFDB D,,CONT
1 YDDFDB  DSECT
1 *
1 *       REASONS FOR COMEND (R4 - RIGHTMOST BYTE)
1 *
1 YDDFNUSE DS    CL3                    UNUSED
1 YDDFRC  DS    CL1                    REASON FOR CONTINGENCY
1 YDDFSHWR EQU   X'00'                  SHUTDOWN WARNING
1 YDDFSHDW EQU   X'04'                  SHUTDOWN
1 YDDFFCWR EQU   X'08'                  FORCED CLOSURE WARNING
1 YDDFFCLS EQU   X'0C'                  FORCED CLOSURE
1 *
1 *
1 *
1 *       REASONS FOR LOSCON (R6 - RIGHTMOST BYTE)
1 *
1 YDDFDSUS EQU   X'00'                  DISCONNECTION BY USER
1 YDDFIPRM EQU   X'04'                  INVALID DEPROT/EDIT COMBINATION
1 YDDFPPER EQU   X'08'                  PTN PROCESS ERR(X.25EVENT,..)
1 YDDFPTNA EQU   X'0C'                  PARTNER NOT AVAILABLE
1 YDDFDSSY EQU   X'10'                  DISCONNECTION BY SYSTEM OPERATOR
1 YDDFLOST EQU   X'14'                  PARTNER LOST CONNECTION
1 YDDFNETW EQU   X'18'                  DISCONNECTION DUE TO NETWORK
1 YDDFDSWR EQU   X'20'                  DISCONNECTION WARNING
1 YDDFRPTC EQU   X'24'                  PTNCHAR NOT ACCEPTED BY PARTNER
1 YDDFADMS EQU   X'28'                  PTN DISCONN. SUMMONED BY ADM.
1 YDDFESSE EQU   X'2C'                  ERR IN STAT.SERV. CONN.ELMT.
1 YDDFESSP EQU   X'34'                  ERR IN STAT.SERV.PROTOCOL
1 YDDFTSER EQU   X'38'                  ERR IN TRANSPORT SYSTEM
1 YDDFCUME EQU   X'40'                  STAT.SERV. CONN. USERMESS. ERR
1 YDDFGOSN EQU   X'50'                  GROS NOT ACCEPTED BY PTN
1 YDDFPRIN EQU   X'54'                  NETW.PRIO NOT ACC'D BY PTN
1 YDDFDSSH EQU   X'58'                  SHORT.RES AT PTN'S SYSTEM
1 *                                     (TOO MANY NTACK'S,...)
1 YDDFVTSI EQU   X'5C'                  PARTNER'S VTSU INACTIVE
1 YDDFPERL EQU   X'60'                  PROTOCOL INCONSISTENCY BY PTN
1 YDDFLSPE EQU   X'64'                  UNRECOVERABLE UCON ERROR
1 YDDFLSSP EQU   X'68'                  STATION NOT PRIVILEGED
1 YDDFLSPP EQU   X'6C'                  PROCESSOR NOT PRIVILEGED
1 YDDFLSNT EQU   X'70'                  NO TASK FOR PW CHECK
1 YDDFLSCI EQU   X'74'                  WRONG CID FOR OP-ID
1 YDDFLSIE EQU   X'78'                  INTERNAL UCON ERROR
1 YDDFLSEC EQU   X'7C'                  NO ECRNAM ENTRY AVAILABLE
1 YDDFLSDC EQU   X'80'                  DCAM VERSION < 10
1 YDDFLSCT EQU   X'84'                  NO CHIPCARD TERMINAL
1 YDDFLSNP EQU   X'88'                  NOT NEW PROTOCOL
1 YDDFLSCS EQU   X'8C'                  NO CHIPCARD SUBSYSTEM
1 YDDFLSKV EQU   X'90'                  ERROR IN KVP PROTOCOL
1 *
1 *
1 *
1 *       REASONS FOR SECOND (R5 - RIGHTMOST BYTE)
1 *
1 YDDFSOPN EQU   X'00'                  OPEN BY SECONDARY TASK
```

```
1 YDDFSCLS EQU   X’04’                  CLOSE BY SECONDARY TASK
1 YDDFSDIS EQU   X’08’                  SECONDARY TASK MISSING
1 YDDFLNCO EQU   *-YDDFNUSE             LENGTH
1 *
1                *,YDDFDB     034    920624   55616014
          END
```

Variant 3

```
                          EXTERNAL SYMBOL DICTIONARY
          START
  YDDFDB  YDDFDB D,,CBRC
1 YDDFDB  DSECT
1 *       RETURNCODE FOR MANIPULATIVE REQUESTS
1 *       (R15 - LEFTMOST BYTE)
1 *
1 YDDFRC15 DS    CL1                    LEFTMOST BYTE OF R15
1 *
1 YDDFOK  EQU   X'00'                  REQUEST SUCCESSFUL
1 *
1 *
1 YDDFARSM EQU   X'04'                 USER/REQM - AREA TOO SMALL
1 *                                    (YGENCB)
1 YDDFINCB EQU   X'04'                 BLKADDR POINTS TO INVALID CB
1 *                                    (YMODCB, YSHOWCB, YTESTCB)
1 *
1 *
1 YDDFMISS EQU   X'08'                 OBLIGATORY PARAM MISSING
1 *                                    (YGENCB,YMODCB,YSHOWCB,YTESTCB)
1 *
1 *
1 YDDFNOFW EQU   X'0C'                 WAREA NOT ON FULLWORD BOUNDARY
1 *                                    (YGENCB)
1 YDDFNOMD EQU   X'0C'                 MODIFICATION NOT ALLOWED
1 *                                    (YMODCB)
1 YDDFUSAR EQU   X'0C'                 USER AREA TOO SMALL
1 *                                    (YSHOWCB)
1 YDDFERET EQU   X'0C'                 INVALID ERET ADDRESS
1 *                                    (YTESTCB)
1 *
1 *
1 YDDFIBLK EQU   X'10'                 INVALID BLK/BLKADDR COMBINATION
1 *                                    (YMODCB, YSHOWCB, YTESTCB)
1 *
1 *
1 YDDFIFLD EQU   X'14'                 INVALID FIELD/BLK COMBINATION
1 *
1 *
1 YDDFILST EQU   X'18'                 INVALID LIST-ADDRESS (MF-PARAM)
1 *
1 *
1 YDDFIUSA EQU   X'1C'                 INVALID ADDRESS OF USER AREA
1 *                                    (YGENCB, YSHOWCB)
1 YDDFICBA EQU   X'1C'                 INVALID CB-ADDRESS
1 *                                    (YMODCB, YTESTCB)
1 *
1 YDDFIREF EQU   X'20'                 INVALID CB REFERENCE
1 *                                    INVALID PL REFERENCE/CONTENTS
1 *                                    (YMODCB, YTESTCB)
1 *
1 YDDFNOSS EQU   X'24'                 CANNOT CONNECT TO SS
1 *
1 *
1          DS    CL3                    UNUSED
1 YDDFLNC  EQU   *-YDDFRC15             LENGTH
```

```
1 *
1 *
1               *,YDDFDB     034    920624   55616014
          END
```

7.6 Name assignment commands

There are two name assignment commands for the BS2000 control system. A short functional description of each command, including the respective formats, is given below.

1. APPLICATION command

The APPLICATION command is employed by DCAM users to store or delete information about a DCAM application in a task-specific table, the communication link table (CLT).

When the DCAM application is opened by a DCAM application program, the values in this table replace the corresponding entries in the application control block ACB (ASSEMBLER) or in the application structure (COBOL). The linkage between the CLT entry and this program area is established by means of the link name, which must be specified both in the command and in the program.

The /APPLICATION command can be used only when the DCAM subsystem has been loaded successfully. Note, too, that the DCAM subsystem status cannot be HOLD/DELETE when this command is used. If a task successfully issued a DCAM command or a DCAM call before entering HOLD/DELETE, it can work with DCAM until the task is ended, despite a /HOLD subsystem or /DELETE subsystem (also applicable to %).

Format

Name	Operation	Operands
	APPLICATION APPL	[applicationname] ,LINK=linkname [,DISNAME=distributionname] [,USEPASS=password1] [,USEPW=password2] [,LOGPASS=password3] (NEA)

The significance of the operands is described in the description of macro YAPPL.

2. CONNECTION command

The CONNECTION command is used by DCAM users to store or delete information about a virtual connection in a task-specific table, the communication link table (CLT).

When such a connection is established, the values contained in this table supplement or replace the relevant entries in the connection control block CCB (ASSEMBLER) or in the connection structure (COBOL). The linkage between the CLT entry and this program area is provided by the link name, which must be specified both in the command and in the program.

The /CONNECTION command can be used only when the DCAM subsystem has been loaded successfully. Note, too, that the DCAM subsystem status cannot be HOLD/DELETE when this command is used. If a task successfully issued a DCAM command or a DCAM call before entering HOLD/DELETE, it can work with DCAM until the task is ended, despite a /HOLD subsystem or /DELETE subsystem (also applicable to %).

Format

Name	Operation	Operands
	{CONNECTION / CONN}	[partnername]
		,LINK=linkname
		[,PRONAME=taskname]
		[,USERFLD=userfield]
		[,LOGPW=password4] (NEA)

The significance of the operands is described in the description of the YCONN macro.

These commands may also be entered in SDF syntax, as follows. A detailed description of the SDF syntax can be found in the manual "User Commands (SDF Format)".

REMOVE-DCAM-APPLICATION-LINK

Deletes information about a DCAM application from the CLT.

Format

```
REMOVE-DCAM-APPLICATION-LINK

 LINK-NAME = <name 1..8>
```

REMOVE-DCAM-CONNECTION-LINK

Deletes information about a DCAM connection from the CLT.

Format

```
REMOVE-DCAM-CONNECTION-LINK

 LINK-NAME = <name 1..8>
```

SET-DCAM-APPLICATION-LINK

Stores information about a DCAM application in the CLT. This information is then used instead of the corresponding specifications in the program when the DCAM application is opened.

Format

```
SET-DCAM-APPLICATION-LINK


 LINK-NAME = <name 1..8>

,APPLICATION-NAME = *BY-PROGRAM / <name 1..8>

,DISTRIBUTION-NAME = *BY-PROGRAM / <name 1..8>

,PROTECTION = PARAMETERS(...)

   PARAMETERS(...)
      |
      | CONNECTION-PASSWORD = BY-PROGRAM / <c-string 1..4> /
      |                                    <x-string 1..8> / SECRET
      |
      |,SHARE-PASSWORD = BY-PROGRAM / <c-string 1..4> / <x-string 1..8> /
      |                  SECRET

,SHARE-PASSWORD = BY-PROGRAM / <c-string 1..4> / <x-string 1..8> / SECRET
```

SET-DCAM-CONNECTION-LINK

Stores information about a DCAM connection in the CLT. This information is then used instead of the corresponding specifications in the program when the connection is established.

Format

```
SET-DCAM-CONNECTION-LINK

 LINK-NAME = <name 1..8>
,PARTNER-ADDRESS = PARAMETERS(...)
   PARAMETERS(...)
     PARTNER-NAME = *BY-PROGRAM / <name 1..8>
     ,PROCESSOR-NAME = *BY-PROGRAM / <name 1..8>
,CONNECTION-PASSWORD = BY-PROGRAM / <c-string 1..4> / <x-string 1..8> /
                       SECRET
,USER-DATA = BY-PROGRAM / <c-string 1..4> / <x-string 1..8>
```

7.7 MNOTEs

Description of the MNOTEs for DCAM macros

MACRO	MNOTE
All	MNOTE *,macroname, versionnumber
YACB	MNOTE 10,ATTR INCORRECT MNOTE 10,ISO INCORRECT MNOTE 10,ATTR SUBOPERAND CONFLICT MNOTE 160,VALUE OF ISO IGNORED BECAUSE OF INVALID ATTR MNOTE 10,VERIFY INCORRECT MNOTE 10,LINKMOD INCORRECT MNOTE 10,LOGPASS INCORRECT MNOTE 10,USEPASS INCORRECT MNOTE 10,USEPW INCORRECT MNOTE 10,DCAMVER INCORRECT
YCCB	MNOTE 10,LOGPW INCORRECT MNOTE 10,USERFLD INCORRECT MNOTE 10,EDITIN INCORRECT MNOTE 10,EDITIN SUBOPERAND CONFLICT MNOTE 10,EDITOUT INCORRECT MNOTE 10,EDITOUT SUBOPERAND CONFLICT MNOTE 10,PROC INCORRECT MNOTE 10,MDATA INCORRECT MNOTE 10,PROC SUBOPERAND CONFLICT MNOTE 10,LINKMOD INCORRECT MNOTE 0,PARAMETER DEPROT REMOVED FROM THE DCAM-INTERFACE MNOTE 10,EDIT INCORRECT MNOTE 10,PTNCHLN OVERFLOW MNOTE 160,VALUE OF MDATA IGNORED BECAUSE OF INVALID PROC MNOTE 10,PRIO INCORRECT
YRPB	MNOTE 1O,OPTCD INCORRECT MNOTE 5,TOVAL OUT OF RANGE MNOTE 1O,EIDREF INCORRECT MNOTE 1O,EIDREF2 INCORRECT MNOTE 1O,OPTCD SUBOPERAND CONFLICT
YPLI	MNOTE 10,MF INCORRECT. STANDARD FORM ASSUMED MNOTE 10,FIELDS-SUBOPERAND INCORRECT MNOTE 10,FIELDS-SUBOPERAND NOT COMPATIBLE WITH BLK MNOTE 10,USEPASS INCORRECT MNOTE 10,USEPW INCORRECT MNOTE 10,LOGPASS INCORRECT MNOTE 10,ISO INCORRECT MNOTE 10,ATTR INCORRECT MNOTE 10,ATTR SUBOPERAND CONFLICT MNOTE 10,VERIFY INCORRECT MNOTE 10,DCAMVER INCORRECT MNOTE 10,LOGPW INCORRECT

MACRO	MNOTE
	MNOTE 0,PARAMETER DEPROT REMOVED FROM THE DCAM-INTERFACE MNOTE 10,EDIT INCORRECT MNOTE 10,EDITIN INCORRECT MNOTE 10,EDITIN SUBOPERAND CONFLICT MNOTE 10,EDITOUT INCORRECT MNOTE 10,EDITOUT SUBOPERAND CONFLICT MNOTE 10,MAXLN OVERFLOW MNOTE 10,MDATA INCORRECT MNOTE 10,PRIO INCORRECT MNOTE 10,PTNTYPE INCORRECT MNOTE 0,PARAMETER PDEPROT REMOVED FROM THE DCAM-INTERFACE MNOTE 10,PEDIT INCORRECT MNOTE 10,PPROC INCORRECT MNOTE 10,PROC INCORRECT MNOTE 10,PROC SUBOPERAND CONFLICT MNOTE 10,PTNCHLN OVERFLOW MNOTE 10,USER INCORRECT MNOTE 10,OPTCD INCORRECT MNOTE 10,OPTCD SUBOPERAND CONFLICT MNOTE 10,EIDREF INCORRECT MNOTE 1O,EIDREF2 INCORRECT MNOTE 10,TOVAL OUT OF RANGE MNOTE 160,REQTYPE INCORRECT MNOTE 10,SEQNO OUT OF RANGE MNOTE 10,USERFLD INCORRECT MNOTE 10,GID NOT IN REGISTER NOTATION MNOTE 10,NUMBER OF CODEVALUE’S EXCEEDS LIMIT (=8) MNOTE 10,CODEVAL INCORRECT MNOTE 10,NUMBER OF DCG’S EXCEEDS LIMIT (=16) MNOTE 10,CODEIND INCORRECT MNOTE 10,LID NOT IN REGISTER NOTATION MNOTE 10,PRONAME NOT COMPATIBLE WITH BLK-VALUE MNOTE 10,LINKMOD INCORRECT MNOTE 10,LINKMOD NOT COMPATIBLE WITH BLK-VALUE MNOTE 10,LINK NOT COMPATIBLE WITH BLK-VALUE MNOTE 10,DID NOT COMPATIBLE WITH BLK-VALUE MNOTE 10,DID NOT IN REGISTER NOTATION MNOTE 10,FLAG INCORRECT MNOTE 10,FLAG NOT COMPATIBLE WITH BLK-VALUE MNOTE 10,AID NOT COMPATIBLE WITH BLK-VALUE MNOTE 10,AID NOT IN REGISTER NOTATION MNOTE 10,CID NOT COMPATIBLE WITH BLK-VALUE MNOTE 10,CID NOT IN REGISTER NOTATION MNOTE 10,PARAM NOT COMPATIBLE WITH BLK-VALUE

MACRO	MNOTE
YAPPL	MNOTE 160,LINK MISSING MNOTE 160,LINK INCORRECT MNOTE 10,APPNAME INCORRECT MNOTE 10,DISNAME INCORRECT MNOTE 10,USEPASS INCORRECT MNOTE 10,USEPW INCORRECT MNOTE 10,LOGPASS INCORRECT
YCHANGE	MNOTE 160,RPB PARAM MISSING
YCLOSE	MNOTE 160,PARAMETER MISSING MNOTE 160,AID NOT IN REGISTER-NOTATION
YCLSCON	MNOTE 160,RPB PARAM MISSING
YCONN	MNOTE 160,LINK PARAMETER MISSING MNOTE 160,LINK INCORRECT MNOTE 10,PTNNAME INCORRECT MNOTE 10,PRONAME INCORRECT MNOTE 10,USERFLD INCORRECT MNOTE 10,LOGPW INCORRECT
YDCG	MNOTE 10,CODEVAL INCORRECT MNOTE 10,NUMBER OF CODEVAL'S EXCEEDS LIMIT (=8)
YDIP	MNOTE 10,CODEIND INCORRECT MNOTE 10,NUMBER OD DCG'S EXCEEDS LIMIT (=16)
YFORBID	MNOTE 160,RPB PARAM MISSING
YGENCB	MNOTE 160,BLK PARAM MISSING MNOTE 10,CB-IDENTIFIER NOT ALLOWED FOR GIVEN BLK
YINQUIRE	MNOTE 160,RPB PARAM MISSING
MACRO	MNOTE
YMODCB	MNOTE 160,BLK AND/OR BLKADDR MISSING
YOPEN	MNOTE 160,ACB MISSING
YOPNCON	MNOTE 160,RPB PARAM MISSING
YPERMIT	MNOTE 160,RPB PARAM MISSING
YRECEIVE	MNOTE 160,FORMAT PARAM INCORRECT MNOTE 160,RPB PARAM MISSING
YREJLOG	MNOTE 160,RPB PARAM MISSING
YRESET	MNOTE 160,RPB PARAM MISSING

YSEND	MNOTE 160,FORMAT PARAM INCORRECT MNOTE 160,RPB PARAM MISSING
YSENDREC	MNOTE 160,FORMAT PARAM INCORRECT MNOTE 160,RPB PARAM MISSING
YSETLOG	MNOTE 160,RPB PARAM MISSING
YSHOWCB	MNOTE 160,PARAM MISSING
YTESTCB	MNOTE 160,BLK AND/OR BLKADDR MISSING

7.8 Effects of the CCITT X.25 Recommendation on the IDCAM user interface

The CCITT X.25 Recommendation allows data terminal equipment (DTE) to be connected to packet switching networks. In TRANSDATA, connection is made using the software product X.25PORT in the PDN of the computer providing the X.25 port (hereafter referred to as the 'XIC'). This allows a heterogeneous network to be set up with TRANSDATA or third-party-systems. The term 'heterogeneous' indicates that at a level higher than X.25 no TRANSDATA protocols can be exchanged between the XIC and a DTE on the other side of the packet switching network (X.25 interfacing computer or third-party-system).

Restrictions on the use of the IDCAM interface

If both communication partners are embedded in a TRANSDATA environment, X.25-specific functions do not need to be controlled.

This is not the case in station mode: in this case, a DCAM application is embedded in a TRANSDATA environment and the other communication partner is embedded in non-TRANSDATA environment. This results in a number of restrictions as regards the IDCAM interface:

IDCAM function	X.25 function	Restriction as compared with TRANSDATA
Connection function:		
Connection request (YOPNCON ACQUIRE)	Setting up the virtual connection	Dependent on the generation, see XSTAT macro, IX25 operand
Connection message (AREA in the YCCB or LOGON contingency)	Control of the optional performance features	Specific structure is prescribed [1)]
Message editing (EDIT=USER in the YCCB)	Is not transmitted	Must always be performed by the application itself
Message code (PROC=SYSCODE in the YCCB)	Is not transmitted	Only EBCDI code possible
Initiative during data transmission (PROC=ANYSTART)	Is not transmitted	Is always optional
Partner characteristics (PTNCHA1 in the YCCB)	Is not transmitted	Partner application always receives the entry X’0133’.
Connection cleardown (LOSCON contingency)	Reset request (RESET) or clearing down the virtual connection	The transport connection is cleared down (reasons specified in LOSCON).

1) See format of the DCAM connection message

IDCAM function	X.25 function	Restriction as compared with TRANSDATA
Data transmission function:		
Sequence number of the message (SEQNO in the YRPB)	Is not transmitted	Not possible
Message structuring (OPTCD=ELEMENT, SUBGROUP in the YRPB)	Is not transmitted	Not possible
Transport acknowledgment (OPTCD=TACK in the YRPB)	Is not transmitted	Only transfer to the PSN is acknowledged.
First byte of the message (AREA/AAREA in the YRPB) 1)	Q bit, TQ bit, reset packet, D bit, M bit	Usage of individual bits is defined in the XSTAT macro, IX25 operand
Express message (OPTCD=EXPRESS in the YRPB)	Interrupt packet	Only 1 byte is relevant

1) The meaning of the bits in the first message byte is as follows:

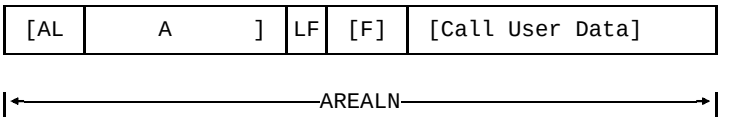
Bit 2^0: Q bit
Bit 2^1: TQ bit
Bit 2^2 and 2^3: Reset packet
Bit 2^4: D bit
Bit 2^5: M bit

Q bit, D bit

The Q(ualifier) bit permits two data types to be differentiated by the DTE.
D bit support permits end-to-end control but provides no information on the transport of the message to the application. A DTE sets the D bit in the packet header to 1 in order to request an acknowledgment for this data packet. If the D bit is set to zero, the acknowledgment has only local significance, i.e. a data packet has been accepted free of errors by the packet switching network. Use of the D bit must be agreed with the network administration.

Format of the DCAM connection message

AREA



AL = Length of local and remote DTE address if the operand IX25=(DTEADR) was specified in the XSTAT macro when the station was generated. One half-byte is used for each length specification.

A = Remote and local DTE address if the operand IX25=(DTEADR) was specified in the XSTAT macro when the station was generated. The addresses are half-bytes, if necessary filled to whole bytes with X'0'.

LF = Length of F, i.e. X'00' if F is not specified (LF itself is one byte long).

F = X.25 facilities (optional) with a length of LF bytes The X.25 format is described in the DATEX-P manual (available in German only from the Fernmeldetechnisches Zentralamt, Darmstadt).

Call User Data
X.25 user message.

Call User Data

The X.25 user message at connect time can be a maximum of 16 bytes long. The first byte of the user message has the following format:

Bits 2^0 to 2^5: undefined;

Bits 2^6 and 2^7:

00 A part of the user message contains the identification of a protocol corresponding to other CCITT recommendations, e.g. X.29.

01 A part of the user message is used for identifying the protocol as stipulated by the network administration.

10 A part of the user message is used for identifying a protocol as stipulated by international user organizations.

11 No restrictions apply to the use of the rest of the user message by the DTE.

There are two methods of process selection via the X.25 user message over a heterogeneous interface:

– Process selection by alphanumeric process code
  In this case the user message has the following format:

  X'D5000001''12'*lla*

  ll: length of the alphanumeric process code
  a: alphanumeric process code

– Process selection by numeric process code
  In this case the user message has the following format:

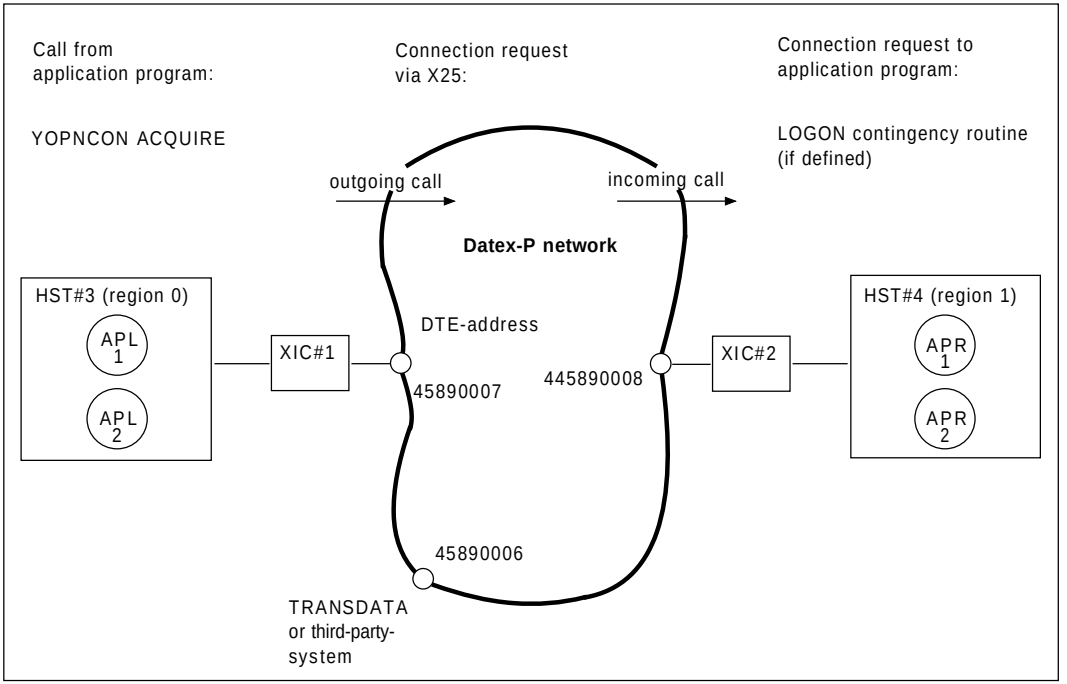
  X'D5000001''1301'*h*

  h: hexadecimal process code (1 byte)

Note

If a dial-up virtual call is established between two packet-oriented DTEs, the network does not respond to the user message nor does it influence it.

Example of connection setup via the X.25 interface (heterogeneous network)



XIC : X.25 interfacing computer with PDN
HST : host computer with BS2000
} TRANSDATA networks

DTE : data terminal equipment

APPxx : name of application

Note

Connections via the X.25 interface can also be predefined (see "Generating a Data Communication System"). In this case the DCAM application can be linked to the connection via YOPNCON ACQUIRE (see page 87). The present description however deals with the dynamic establishment of connections via the X.25 interface.

X.25 Station Tables in the XICs

XIC#1

remote DTE address	remote application	local partner	parallel connection?
45890008	APPR1 /2/	-	-
45890008	APPR2	-	-

XIC#2

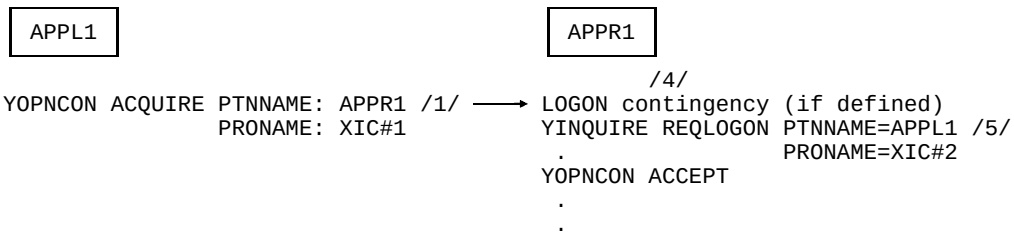
remote DTE address	remote application	local partner	parallel connection?
45890007	APPL1 /3/	APPR1 4/1	/13/
45890007	APPL2	(irrelevant)	-
45890006	APPM1 /8/	APPR2 4/1	-

During generation of the PDN, the entries are defined by means of:

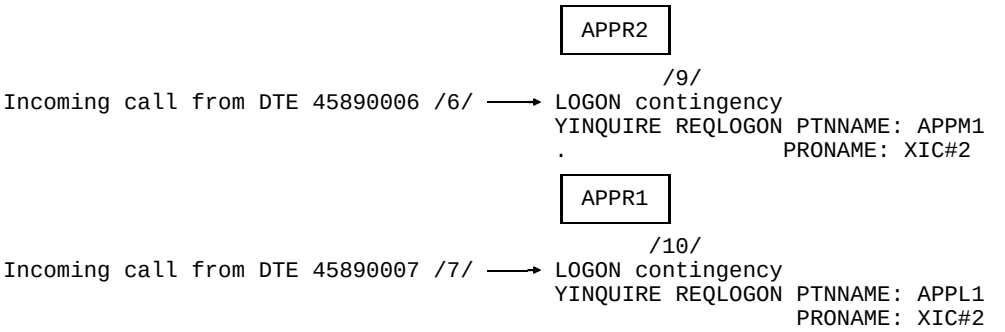
XSTAT parameter:	DTEADR	STATNAM	PARTNAM, PARTPRO	PARVER
XKON parameter:			PARTNAM, PARTID	

Example 1: direct connection establishment

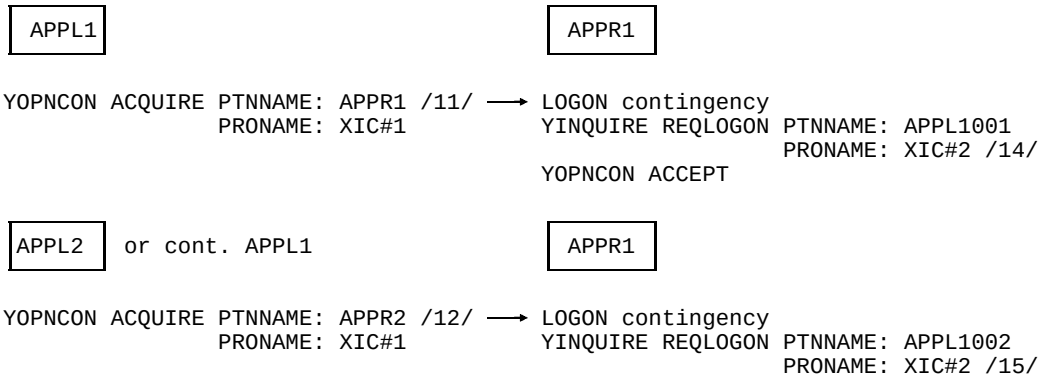
1. Single connection



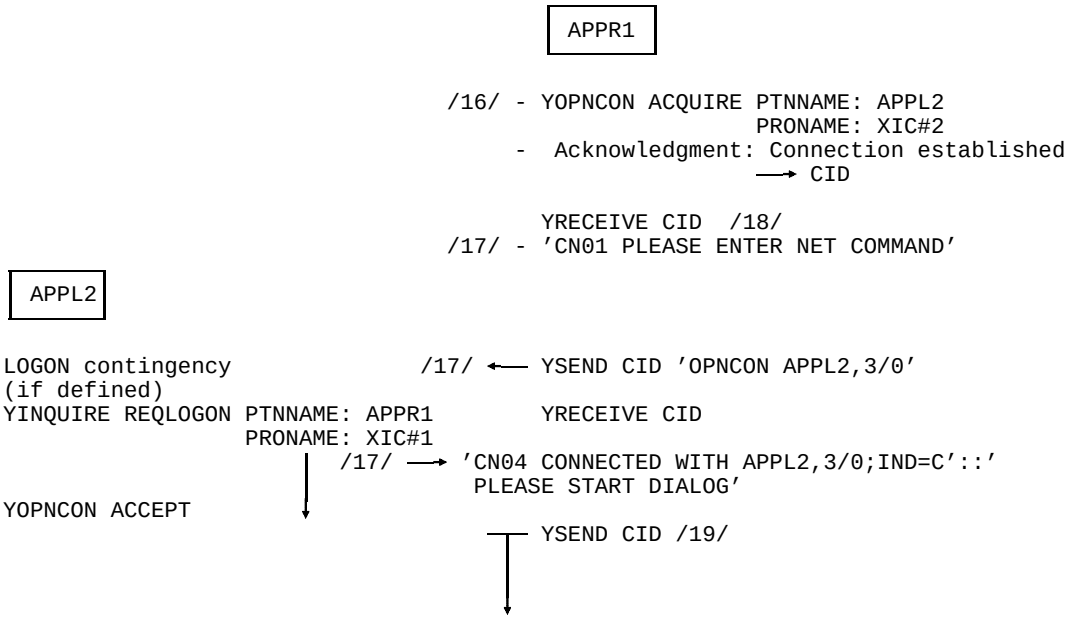
2. Connection request to various local partners of a DTE



3. Parallel connections



Example 2: connection establishment with predialog



Explanation of example 1

1. Single connection

outgoing call:

/1/ Request for connection establishment via X.25: the local X.25 computer XIC#1 is specified as the processor in the partner connection.

/2/ In the X.25 station table of XIC#1 the name of the partner to be connected (APPR1) represents a remote application and refers to the remote DTE address 45890008 which is to be accessed. This is the destination address for the outgoing call.

incoming call:

The incoming call via X.25 contains the source DTE address 45890007.

/3/ This identifies in the X.25 station table of the local X.25 computer XIC#2 the (representative) name APPL1 of the remote partner and the local TRANSDATA partner APPR1 which is to be allocated.

/4/ The local partner application receives a connection request.

/5/ Information concerning the connection request: the local X.25 computer XIC#2 appears as the processor of the partner application.

/3/ The partner name APPL1 derives from the entry in the X.25 station table of XIC#2.

2. Connection requests to various local partners of a DTE

/6/ /7/ Incoming calls from various remote DTE addresses or various private extensions having one and the same DTE address.

/8/ /3/ Each individual DTE address or private extension identifies the entry belonging to it in the X.25 station table.

/9/ /10/ The connection requests are sent to various local TRANSDATA partners in accordance with the entries in the X.25 station table.

3. Parallel connections

/11/ /12/

Several incoming calls from one and the same remote DTE address (45890007) (no private extensions) in response to the connection request(s) of one or more remote applications.

/3/ The parallel connections give rise in each case to connection requests which are sent to one and the same local TRANSDATA partner APPR1 in accordance with the first entry for the DTE address 45890007 in the X.25 station table of XIC#2.

/13/ If (the representative) APPL1 is generated in XIC#2 with PARVER=JA, several parallel connections to APPR1 can be established.

/14/ /15/

In each case the name of the requesting partner (from the X.25 station table of XIC#2) is modified with a three-figure sequence number when parallel connections are being used.

Explanation of example 2

/16/ /2/ No local partner is predefined in XIC#1 for the incoming call from the remote DTE address 45890008.

/17/ In the predialog with the connection handler of the remote X.25 computer XIC#1, APPR1 nominates the desired remote partner.

/18/ The predialog is carried out as a data exchange via the connection which has already been established with respect to DCAM.

/19/ The data exchange between APPL2 and APPR1 can now begin.

Connection handling by the X.25 port

Reasons for rejection of a connection request

Rejection by the X.25 network	Acknowledgment after YOPNCON (see the appendix, page 286)
Remote station busy/malfunction	0C4C
Remote station refuses to accept charges	0C5C
Invalid connection request	0C4C
Access inhibited (resource bottleneck)	0C4C
Network overloaded (resource bottleneck)	0C4C
Remote station not accessible	0C4C

Rejection by the X.25 port	Acknowledgment after YOPNCON (see the appendix, page 286)
Line does not lead to the network specified in the directory (DATEX-P, Euronet,...)	0C4C
Network inactive	0C4C
Generated DTE address of the recipient is invalid	0C4C
No free supervisor call (SVC)	0C4C
Name of the recipient is not in the X.25 directory	0C4C
Local resource bottleneck	0C4C

Reasons for connection cleardown

Reason	Value in register 6 with LOSCON
Cleardown by remote DTE	X’00’
Run error of the remote station/local	X’08’
X.25 reset	X’08’

7.9 Reasons for negative transport acknowledgments

This appendix covers the most frequent, typical reasons for negative transport acknowledgments for DCAM(NEA) transport service applications which may occur in DCAM application programs.

Generation of negative transport acknowledgments

Negative transport acknowledgments can be generated by terminal systems or transit systems.

Terminal systems are systems that contain at least one of the connection partners.

Transit systems are systems that interconnect terminal systems.

Transport systems

Transit systems send negative transport acknowledgments if a message or an express message could not be conveyed, e.g. because

– the line or the port to the remote system is not ready for operation

Note: One reason for "line inactive" may also be a resource bottleneck in the receive system.

– a message is longer than the maximum permissible packet length.

Handling of negative transport acknowledgments is dependent upon the relevant product version (refer to table):

Home system	Remote system		Handling of negative transport acknowledgments
	DCAM version	PDN version	
≥ 8.0	≤ 7.0	≤ 8.0	All negative transport acknowledgments are sent to the DCAM application program
≥ 8.0	≥ 8.0	≥ 8.1	Negative transport acknowledgments are sent to the DCAM user program only by the terminal system. Type "resource bottleneck" does not occur. It is replaced by the data flow mechanism.

Terminal systems

PDN and DCAM/BS2000 handle negative transport acknowledgments differently (refer to table):

Reason	DCAM/BS2000	PDN
Data terminal, line or port not active	No negative transport acknowledgment, connection cleardown after repeated attempt (data terminal connected locally via MSN)	Negative transport acknowledgment
Application not available	Negative transport acknowledgment (not in the case of predefined connections)	Negative transport acknowledgment: - Loading error in the program - Error in the application program - Addressing error
Application program	Negative transport acknowledgment: - The user does not accept the signalled data ("time out")	Negative transport acknowledgment: Only by the user program
Resource bottleneck *)	Negative transport acknowledgment: - Bottleneck in the queue for express messages	No negative transport acknowledgment from the terminal system
Fault in the network	-	Negative transport acknowledgment X.25 reset

*) Generation of a negative transport acknowledgment depends on the product versions.

The negative transport acknowledgments listed are sent to the DCAM user program.

8 Manual supplements

This chapter is an update for the present manual valid for DCAM V13.3.

8.1 Opening an application - PRONAME parameter

Addition to YOPEN and YACB on pages 48, 52, 55, 58, 61 and 153 in the manual:

After successfully opening the application and specifying PRONAME, the name of the processor in which the application is opened is always returned. If there is an entry in the configuration file for the application, it is always the name of this virtual host. The default name of the configuration file is $TSOS.SYSDAT.BCAM.APPLICATIONS.

If the application is to be opened on changing hosts, the (virtual) host must be entered in the CLT with the /SET-DCAM-APPLICATION-LINK or /APPL command. In this case, there must be no entry present in the configuration file.

Detailed information on virtual hosts and the configuration file can be found in the BCAM User Guide, volumes 1 and 2.

8.2 Connection setup

Addition to page 70 in the manual:

New paragraph after Note:

The CID return field mentioned below is only filled by DCAM after a successful connection setup.

The maximum wait time when requesting connection setup is defined via the /BCTIMES CONN= command.

8.3 YRPB macro

Correction to page 221 in the manual:

Changed paragraph under ANY:

This operand is evaluated by YRECEIVE and YRESET. It is also evaluated by YOPNCON with OPTCD=ACCEPT, if ... was set.

YSENDREC is always executed with OPTCD=SPEC regardless of the entry in the RPB, i.e. a YSENDREC macro for a partner in the CA state is always rejected with return code 0C2C0000.

Correction to page 223 in the manual:

Additional paragraph under TOVAL=absexp:

With YOPNCON OPTCD=ACQUIRE the value is that which was set with /BCTIMES CONN=.

8.4 DCAM system exit

Addition to page 269 in the manual:

New paragraph after the last line of 12) Exit event 'CONBAD':

The return code is only passed on to the user after refusal by the system exit in the local computer (events APPOPN, CONACQ and CONACC, points 1), 6) and 7)).

If refusal is by the system exit of the partner (event CONREQ, point 9), the DCAM return code is X'0C40' regardless of what the partner specified in the feedback field.

8.5 Feedback messages

Various supplements and corrections:

Page 289: Additional meaning for FDB3=2**2=1:

With YRECEIVE and EDITIN=PHYS, the user area is possibly too small to take the message header.

Page 292: meaning for 08 58:

The application has already been opened on another (virtual) host.

Page 293:
meaning for 0C 2C:
The partner refuses connection setup or the partner is unknown.

Page 294:
meaning for 0C 3C:
The partner has already issued YCLSCON.

Page 295/296: Correction to footnote 1):

These messages are output if a connection setup request to the "$CONSOLE" system application is rejected by it and the syntax of the connection message corresponds to a logon as logical console with generated authorization name. $CONSOLE behaves differently for logon of a logical console with a dynamic authorization name.

Detailed information on the format of the connection message can be found in the manual "Introduction to System Administration".

Page 296:
meaning for 10 04 00:
Lack of memory with EDIT=SYSTEM: no space available for the rest of the buffer.

meaning for 10 14 50:
Host incorrectly configured

Page 298:

– Correction to return code 18 30:

  18 30 can also occur with YOPNCON, therefore "x" in the YOPNCON column.

– Correction to return code 18 34:

  Invalid name for DCAM application or invalid processor name.

Page 299: New line for return code 18 5C:

FDB3=18: unknown device type for VTSU" ("x" in column YOPNCON)

8.6 Register setting

Page 302/303: New note for register 6:

Setting register 6 to a value not equal to X'00' is not a guaranteed service. With some transport service applications, register 6 is always set to X'00', i.e. the value X'00' then has no real meaning.

Page 304: correction of the last line in table:

X'64'-X'A8' | reserved; see YDDFDB D,,CONT

8.7 DSECTs for control blocks

Various additions and corrections:

Page 310 - 342: all macros:

The command line with the version number and date may have changed for all macros."

Page 312, YDDACB macro:

Insert after line "2 YDDAUKVP EQU X'D4'...":

```
2 YDDAULET EQU   X'D8'            LETTER LOST
2 YDDAUDUP EQU   X'DC'            DUPLICATE CID
2 YDDAUBAD EQU   X'EO'            BAD OPERATOR ID
2 YDDAVMR1 EQU   X'E4'            $VMCONS REJECT 1
2 YDDAVMR2 EQU   X'E8'            $VMCONS REJECT 2
```

Page 314, YDDACB macro:

Insert after line "2 YDDAENPT EQU X'14'...":

```
2 YDDAIDEV EQU   X'18'            INVALID DEVICE
```

Page 329, YDDRPB macro:

Insert after line "2 YDDRUKVP EQU X'D4'...":

```
2 YDDRULET EQU   X'D8'            LETTER LOST
2 YDDRUDUP EQU   X'DC'            DUPLICATE CID
2 YDDRUBAD EQU   X'EO'            BAD OPERATOR ID
2 YDDRVMR1 EQU   X'E4'            $VMCONS REJECT 1
2 YDDRVMR2 EQU   X'E8'            $VMCONS REJECT 2
```

Page 331, YDDRPB macro:

Insert after line "2 YDDRENPT EQU X'14'...":

```
2 YDDRIDEV EQU   X'18'            INVALID DEVICE
```

Page 336, YDDFDB macro:

Insert after line "1 YDDFUKVP EQU X'D4'...":

```
1 YDDFULET EQU   X'D8'            LETTER LOST
1 YDDFUDUP EQU   X'DC'            DUPLICATE CID
1 YDDFUBAD EQU   X'E0'            BAD OPERATOR ID
1 YDDFVMR1 EQU   X'E4'            $VMCONS REJECT 1
1 YDDFVMR2 EQU   X'E8'            $VMCONS REJECT 2
```

Page 338, YDDFDB macro:

Insert after line "1 YDDFENPT EQU X'14'...":

```
1 YDDFIDEV EQU   X'18'            INVALID DEVICE
```

8.8 Commands for name assignment

Page 343: Additional operand for APPLICATION command:

[,HOSTNAME=processorname]

processorname can be used to specify the name of a virtual host in which the application is to be opened.

Page 346: Additional operand for SET-DCAM-APPLIC command:

,HOST-NAME=*PROGRAM/<name 1..8>
<name 1..8> can be used to specify the name of a virtual host in which the application is to be opened.

Page 346 - 347: General correction:

"*BY-PROGRAM" must always be used instead of "BY-PROGRAM".

Glossary

CMX application
A communication application running on a SINIX or BS2000 computer and controlled by a CMX application program.

communication application
A facility for processing the messages exchanged by communication partners. It is addressed by the data communication system via its access point.

communication partners
Entities that maintain connections and exchange data with each other.

[communication] protocol
A description of the conditions and formats for transfer of information between equivalent functional layers in the data communication system.

communication computer
A computer designed specially for communication functions.

communication access method
The software that provides applications with an interface to the communication facility.

connection
A relationship between two communication partners that permits them to exchange data.

data unit
The quantity of data that can be passed to or received from DCAM with one call.

data communication system
A complex combination of hardware and software products that permits communication partners to exchange data in accordance with certain rules.

[DCAM] application
A communication application that is controlled by at least one DCAM application program.

[DCAM] application program
A program that uses the services of the DCAM access method; it controls one or more DCAM applications.

[DCAM] data communication function
A DCAM function that is related to the transmission and reception of messages and acknowledgments.

[DCAM] event
A DCAM-specific event that can be used for coordination of certain operations in the data communication system. There is no specific time relationship between its arrival and the execution of the program (= asynchronous event).

[DCAM] existence function
A DCAM function that is related to the generation and cancellation of DCAM applications.

[DCAM] name assignment function
A DCAM function that permits the user to generate application programs independently of current parameter values, such as the DCAM application name, the partner name, etc.

[DCAM] connection function
A DCAM function that is related to the establishment and clearing down of connections.

express message
A message, with a restricted length, that is transmitted with a higher priority than normal messages.

format terminal
An operating mode of a virtual terminal where the message consists of a format (= entry form, screen mask).

line terminal
An operating mode of a virtual terminal where the message is structured in the form of lines.

logical terminal
⟶ virtual terminal

message
A logically related set of data that is to be transmitted to or received from a communication partner.

process
A facility for executing a program within a task.

shareable DCAM application
A DCAM application that can be used simultaneously by more than one task.

task
The carrier for processes. In BS2000, tasks are used, amongst other things, for execution of user jobs (e.g. batch job, interactive task) or for operation of (DCAM, UTM, TTX) applications (execution of all procedures specified between the BS2000 commands LOGON and LOGOFF).

terminal user
A person who uses a terminal to exchange data with a communication partner.

transport service
A service for the exchange of data between communication partners. The transport service initiates and monitors the transport of messages through the data communication system and manages connections.

transport acknowlegment
An event that provides information about the successful or unsuccessful execution of a data transfer.

virtual terminal (logical terminal)
A terminal model whose functions are mapped on the physical characteristics of various terminal types.

Related publications

FHS (BS2000/OSD)
User Guide
User Guide

DCAM (BS2000/OSD)
COBOL Calls
User Guide

DCAM (BS2000/OSD)
Program Interfaces
Reference Manual

BS2000/OSD
Utility Routines
User Guide

BS2000/OSD
User Commands
Volume 1-7
User Guide

BS2000/OSD
Executive Macros
User Guide

TIAM (BS2000/OSD)
User Guide

VTSU (BS2000/OSD)
User Guide

XHCS
Extended Host Code Support for BS2000/OSD
User Guide

Related publications

Index

Contents

User Guide - English

FUJITSU

DCAM (BS2000)

Macros

Valid for
DCAM V11.0A
With Supplement chapter for DCAM V13.3A

Edition April 2010

Comments… Suggestions… Corrections…

The User Documentation Department would like to know your opinion on this manual. Your feedback helps us to optimize our documentation to suit your individual needs.

Feel free to send us your comments by e-mail to:
manuals@ts.fujitsu.com

Certified documentation according to DIN EN ISO 9001:2000

To ensure a consistently high quality standard and user-friendliness, this documentation was created to meet the regulations of a quality management system which complies with the requirements of the standard DIN EN ISO 9001:2000.

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

Copyright and Trademarks

All hardware and software names used are trademarks of their respective manufacturers.

On April 1, 2009, Fujitsu became the sole owner of Fujitsu Siemens Computers. This new subsidiary of Fujitsu has been renamed Fujitsu Technology Solutions.
This document is a new edition of an earlier manual for a product version which was released a considerable time ago in which changes have been made to the subject matter.
Please note that all company references and copyrights in this document have been legally transferred to Fujitsu Technology Solutions.
Contact and support addresses will now be offered by Fujitsu Technology Solutions and have the format …@ts.fujitsu.com.
The Internet pages of Fujitsu Technology Solutions are available at http://ts.fujitsu.com/...