User Guide - English

# ADAM V17

## Abstract Device Access Method

Edition June 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 9001:2008

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 9001:2008.

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

## Copyright and Trademarks

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/...

# Contents

# 1 Preface

## 1.1 Summary of contents

This manual describes the access method ADAM (**A**bstract **D**evice **A**ccess **M**ethod). ADAM is executable under BS2000/OSD-BC.

This manual provides you with the following information:

*Chapter 1* "Preface"
This chapter provides a summary of the contents of the manual and describes the changes made since ADAM V11.0A.

*Chapter 2* "Product description"
This chapter describes the functions provided by ADAM.

*Chapter 3* "Program interface"
This chapter contains a description of the macros with the macro call format, a description of the operands, and the return code table. At the beginning of the chapter you will find a description of the metasyntax used in the macro call format.

*Chapter 4* "ADAM subsystem"
This chapter explains how to install, load and unload the ADAM subsystem.

*Chapter 5* "System generation"
This chapter contains information which you should bear in mind at system generation for devices supported by ADAM.

*Chapter 6* "Device descriptions"
This chapter contains two examples of the structure of a device description for a device supported by ADAM.

*Chapter 7* "Sample program"
This chapter contains a sample program which illustrates the use of the new macros.

*Chapter 8* "Appendix"
This chapter contains a brief description of the macros that are available for use in user programs up to and including ADAM V11.0A. These macros continue to be supported for reasons of compatibility.

## 1.2 Target group

This manual is intended for Assembler programmers who want to include in their programs I/O access for devices that are not supported by the logical access methods of the BS2000 operating system.

## 1.3 README file

Information on any functional changes and additions to the current product version described in this manual can be found in the product-specific README file. You will find the README file on your BS2000 computer under the file name **SYSRME.ADAM.<version>.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, or print it out on a standard printer using the following command:

```
/PRINT-DOCUMENT SYSRME.ADAM.<version>.E, LINE-SPACING=*BY-EBCDIC-CONTROL
```

## 1.4 Changes since the last version of the manual

This new edition of the manual incorporates the following changes with regard to its predecessor:

- Description of the ATTR operanden ATTR in the FDEFIO macro, see page 40.
- With the aid of the FDEFIO macro it is possible to specify a TIC device job, see page 41.

# 2 Product description

The access method ADAM enables devices that are not supported by the logical access methods of the BS2000 operating system (Data Management System, SPOOL, etc.) to be attached and used on BS2000 installations.

Such devices are referred to in this manual as "physically supported devices". Access to these devices is achieved exclusively by means of macro calls in Assembler programs.

Physically supported devices typically include the following:

- plotters
- impact printers
- graphics terminals
- page readers
- document readers
- magnetic stripe readers
- automatic testers
- cartridge devices
- photo-composing machines

**ADAM enables you to create jobs independently of the CPU and the channel:**

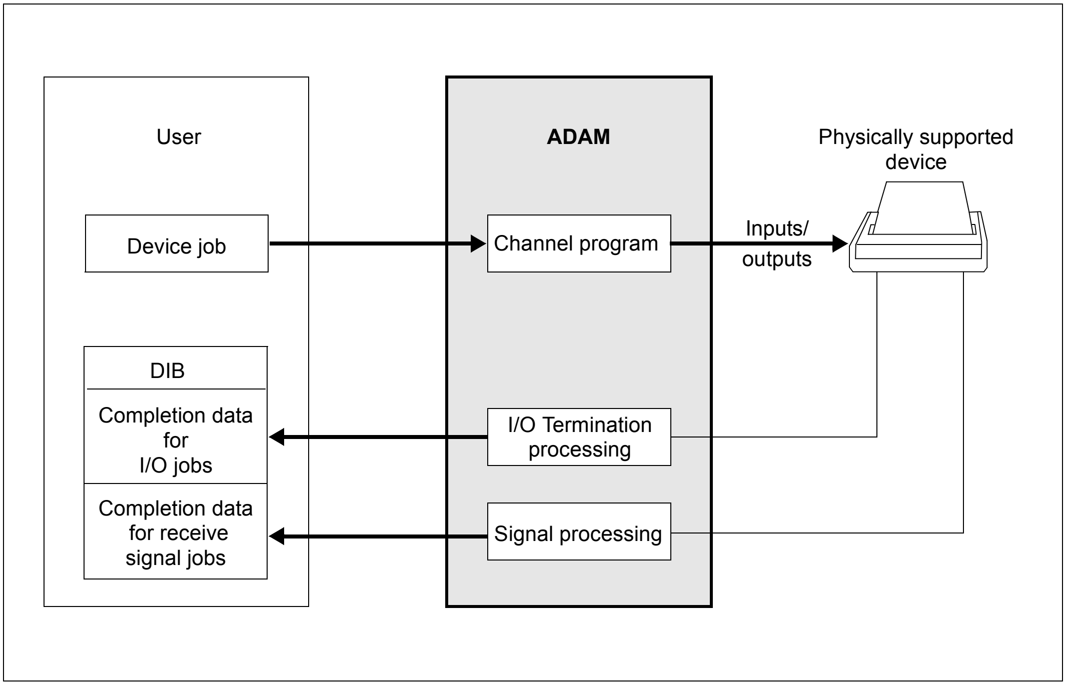
- Instead of having to write channel programs, you simply formulate device jobs and leave it to ADAM to translate these into channel programs.
- Completion data provided by the channel on termination of an I/O operation is converted by ADAM into channel-independent return information for the user program.

For each physically supported device there is an ADAM device description that contains all the information required for the ADAM interfaces. You will find two examples of device descriptions starting on page 87.

**Support requests of specific devices may be addressed to you local representative of Fujitsu Technology Solutions.**

## 2.1 Functions

The diagram below illustrates in simplified form how you can access physically supported devices with the help of the ADAM access method.
You formulate device jobs, which ADAM converts into channel programs. Completion data from I/O operations or receive signal jobs are made available to you in a separate storage area, the device information block (DIB).
A description of the individual ADAM functions and the macro calls required for their implementation follows the diagram.


User
Device job
DIB
Completion data for I/O jobs
Completion data for receive signal jobs
ADAM
Channel program
I/O Termination processing
Signal processing
Inputs/ outputs
Physically supported device

### 2.1.1 Initiating and terminating device access

Device access under ADAM control must be initiated by means of the **FOPNDV** macro. In this device control open macro you specify the device you wish to use.

ADAM then grants an exclusive access right to the specified device until the **FCLSDV** (device control close) macro is issued. Unless you call the FCLSDV macro explicitly, the device control close is performed automatically at program or task termination.

ADAM is not able to reserve a device beyond the end of a program run.

### 2.1.2 Defining device jobs

Input/output operations are described by so-called device jobs. These jobs are lists of parameters with which ADAM generates channel programs.

A device job is defined and modified by means of the **FDEFIO** macro. Device jobs can be generated and supplied with data in the data area of the user program at assembly time (i.e. statically; MF=L, MF=C operand) and can be modified at program runtime (MF=M).

You may link together a maximum of 63 device jobs in a device job chain (CHAIN operand of the FDEFIO macro).

### 2.1.3 Generating channel programs

The device jobs are converted into channel programs with the help of device-specific information from the device modules which form part of the ADAM selectable unit.

Channel programs can be generated in two ways:

1. Implicitly by starting an I/O operation with the **FEXCIO** or **FEXIOW** macro if this references a device job (IOPRG operand of the FEXCIO and FEXIOW macros). In this case, the channel program is generated with the help of the device job chain prior to actual input/output and is deleted again automatically on completion of the I/O operation.
2. Explicitly by calling the **FDECCP** macro. In this operation, a channel program is generated in resident memory and is preserved there until it is deleted again by the **FDELCP** macro or until device access is terminated.
   If a channel program generated in this way is referenced at the start of an I/O operation (CPID operand of the FEXCIO and FEXIOW macros), input/output will be faster because the device job chain is not converted again. Up to 63 FDECCP-generated channel programs are allowed per device.

## 2.1.4 Creating device information blocks

Upon completion of an I/O operation, ADAM stores the completion data in the device information block (DIB), a separate data area of the user program.
Interrupt data is also stored here when a receive signal job is completed.

You create a device information block by means of the **FDEVIB** macro. You must specify the address of the DIB when you call a macro which terminates, or which may terminate, an I/O operation (FEXIOW, FCEKIO, FWFTIO, FCEKSG and FWFSG macros).
You must initialize the DIB before you call a macro, i.e. you must supply the standard header with the correct values.
ADAM checks the values stored in the FCTU (function unit number), FCT (function number) and FCTV (function interface version) fields of the standard header (see page 21).

## 2.1.5 Initiating, controlling and terminating I/O operations

An I/O operation involves executing a device job or device job chain.

I/O operations are started and controlled by means of the following macros:

**FEXIOW** : initiates a synchronous I/O operation. You are only given control again when the I/O operation is terminated.

**FEXCIO** : initiates an asynchronous I/O operation. You are immediately given control again, even if the I/O operation is still running. You must call the FCEKIO or FWFTIO macro for termination processing.
In an FCEKIO or FWFTIO macro, you refer to a specific I/O operation by specifying the appropriate I/O identifier (IOID operand of the FCEKIO and FWFTIO macros).
When you initiate an asynchronous I/O operation, this identifier is the value returned to the field whose address you specified when you called the FEXCIO macro.

**FCEKIO** : checks the status of an asynchronous I/O operation. If this operation has already ended, termination processing is performed.

**FWFTIO** : waits for the end of an asynchronous I/O operation and performs termination processing.

**FHLTIO** : aborts a currently running asynchronous I/O operation.

**FTIMOT** : defines a period of time (timeout) after which an I/O operation is terminated if it has not already ended. The default specified in the device module for each device can be modified with this macro.

I/O operations are terminated

- by the arrival of the end message from the device or controller for input/output (contingent upon the IOTERM operand of the FEXCIO and FEXIOW macros).
- by expiry of the specified maximum time interval before the arrival of the end message (timeout)
- by abortion of input/output by means of the FHLTIO macro.

ADAM stores the result data of an I/O operation in a device information block. Information on the success or failure of an I/O operation is stored here along with information on the cause of any error that occurred. You must specify the address of the device information block when you call the FEXIOW, FCEKIO and FWFTIO macros.

### 2.1.6 Processing device signals

ADAM enables you to process device signals (attention interrupts) and the associated interrupt data in your program. A device signal is an interrupt which is output by the device and which does not relate directly to an I/O operation. Device signals serve to report changes in the state of a device that are due to external influences or to some situation in the device itself (e.g. ready to operate or ready to send/receive data).

The following macros are provided for this purpose:

**FRECSG**: You can use the FRECSG macro to have the next device signal sent to yourself.

**FCEKSG**: You can use the FCEKSG macro after an FRECSG macro to check whether a device signal has arrived.

**FWFSG**: You can use the FWFSG macro to wait for the arrival of the device signal.

A receive job placed with FRECSG is concluded when a FCEKSG or FWFSG macro confirms the arrival of the device signal.

The interrupt data belonging to a device signal is stored in the device information block when a receive signal job is concluded. You must specify the address of the device information block when you call the FCEKSG or FWFSG macro.

The following steps illustrate how device signals are typically used in an ADAM application:

```
FRECSG              Place a receive job for the next device signal

................
................    other operations
................

FCEKSG              Check the state of a receive job
oder
FWFSG               Wait for as device signal
```

## 2.1.7 Participating in user eventing

ADAM supports BS2000 user eventing, which is a mechanism for coordinating the execution of two or more user programs in different tasks.

To participate in user eventing, you must request an event item by means of the BS2000 macro ENAEI. An identifier for the event item (event item ID) is returned. You must specify this event item ID when you start an asynchronous I/O operation with the FEXCIO macro, or when you place a receive signal job for a device signal with the FRECSG macro (EIID operand of the FEXCIO and FRECSG macros).
In addition, a freely selectable information byte (1 byte) can be defined (POSTBYT operand of the FEXCIO and FRECSG macros).
Termination of input/output or the arrival of the device signal is then reported to the user program as an event relating to the specified event item.
Calling the BS2000 macro SOLSIG allows the user program to wait for the event (termination of input/output or device signal) synchronously or asynchronously. Arrival of the event is reported in the first byte of the post code (4 bytes). You can request this information by means of the POSTBYT operand.

You will find descriptions of the SOLSIG and ENAEI macros in the "Executive Macros" manual [2].

## 2.1.8 Performing device error recovery

If a device error is reported when an I/O operation or receive signal job is completed, you must ensure that device-specific error recovery is performed.
For this purpose, ADAM provides you with the necessary error flags and error information in the return code and in the device information block of the macro involved.

ADAM does not perform error recovery when input/output is terminated with an error. Device errors are only logged in the HEL file under the circumstances defined in the device module.

### 2.1.9 System privileges for device access

Any device that can be accessed by means of ADAM can be assigned system privileges, which are stored in the device module of the associated device.

Any task that is to access a device protected in this way must possess the appropriate system privileges. If this is not the case, initiation of device access (FOPNDV macro) is rejected.
If a system privilege is assigned to a device, successful initiation attempts and initiation attempts that fail due to the missing system privilege are recorded in a SAT (security audit trail) entry (provided that the SAT component of the SECOS software product has been installed, see the "SECOS" manual [3]).

## 2.2 Performance characteristics of ADAM

ADAM offers the following performance characteristics:

| Performance characteristic | Maximum value |
|---|---|
| Parallel I/O operations | 2 |
| Pre-generated channel programs | 63 |
| Device jobs per input/output | 63 |
| Amount of data per device job | 65535 bytes |

Table 1: Performance characteristics of ADAM

## 2.3 Compatibility

To ensure the source code compatibility of existing programs so that their assembly is still possible, the old Assembler interfaces remain available under their old names and with their old functionality. However, they will not be further developed.

The old ADAM Assembler interfaces remain object-compatible in order to ensure that application software that has already been assembled can still be run without any loss in functionality.

# 3 Program interface

## 3.1 Action and definition macros

The macros provided by the ADAM access method belong to one the following two groups of macros, depending on their function:

- action macros
- definition macros

### 3.1.1 Action macros

An action macro is a macro that is expected to perform a certain function.
Due to the way in which their operands are passed, action macros are S-type macros (operands are passed in memory), i.e. the operand values specified in the macro call are passed to ADAM in the form of a data area. An ADAM function is subsequently called. Register R1 is overwritten with the address of the associated data area before a macro is called. The remaining registers remain unchanged.

The following action macros are available:

**Action macros for device access**

| **Macro name** | **Macro description** |
|---|---|
| FOPNDV | Open device control. |
| FCLSDV | Close device control. |

**Action macros for device jobs**

| **Macro name** | **Macro description** |
|---|---|
| FDECCP | Generate and store channel program. |
| FDELCP | Delete channel programs. |

### Action macros for I/O operations

| Macro name | Macro description |
|---|---|
| FTIMOT | Define maximum waiting time for I/O termination. |
| FEXIOW | Initiate an I/O operation and wait for termination (synchronous I/O operation). |
| FEXCIO | Initiate an I/O operation and do not wait for termination (asynchronous I/O operation). |
| FHLTIO | Force termination of an asynchronous I/O operation. |
| FCEKIO | Check the status of an asynchronous I/O operation;<br>execute termination processing if the I/O operation has been completed. |
| FWFTIO | Wait for termination of an asynchronous I/O operation and perform termination processing. |

### Action macros for signal processing

| Macro name | Macro description |
|---|---|
| FRECSG | Request delivery of the next device signal. |
| FCEKSG | Check the status of a requested device signal and perform termination processing if the signal has arrived. |
| FWFSG | Wait for the arrival of the next device signal and perform termination processing. |

## 3.1.2 Definition macros

Definition macros are O-type macros (macros without type assignments) and enable you to create data areas and enter operand values in these areas. Definition macros do not call any ADAM functions. No registers are updated.

Macros for defining data areas

| Macro name | Macro description |
|---|---|
| FDEFIO | Define a device job. |
| FDEVIB | Create a device information block. |

## 3.2 Forms of the macro calls

The MF operand determines the type of macro generation.
For action macros (see page 17), the MF operand can be assigned the value C, D, E, M or L.
In the case of definition macros (see page 18), you may specify the value C, D, L or M.

A distinction is made between five forms of the macro call, depending on the value specified for the MF operand:

| Macro name | Operands |
|---|---|
| `<macroname>` | `MF=C [,PREFIX=p][,MACID=mac]`<br>`MF=D [,PREFIX=p]`<br>`MF=E [,PARAM=<addr> / (<reg>)]`<br>`MF=L [,operand` $_1$`,...,operand` $_n$`]`<br>`MF=M ,operand` $_1$`,...,operand` $_n$ `[,PREFIX=p][,MACID=mac]` |

MF=C [,PREFIX=p][,MACID=mac]

Only the data area is generated. Each field has a field name and explanatory equates, if necessary. The data area ends with a length equate. The standard header (see page 21) is normally initialized by the user.
The PREFIX operand allows you to define the first character of the field names and the equates. p = 1 letter.
The MACID operand allows you to define the second, third and fourth characters of the field names and equates. mac = 1..3 characters.

No other operands are evaluated in the C form.

MF=D [,PREFIX=p]

A DSECT is generated. Each field has a field name and explanatory equates, if necessary. The location counter is set to zero. The DSECT ends with a length equate. The system does not switch to the original location counter.
The **DSECT** describes the structure of a memory area without occupying any memory space itself. The symbolic name specified for the DSECT can be used when assigning a base register (USING statement). The location counter is set to zero.
You can use the prefix p to define the first character of the field names and equates. p = 1 letter.

No other operands are evaluated in the D form.

MF=E [,PARAM=<addr> / (<reg>)]

Only the commands required to call the ADAM function are generated. The instruction segment ends with an SVC. You must specify the address of the data area with the operand values in the macro call.

The PARAM operand indicates the address of the data area. addr = address of the data area.
reg = register containing the address of the data area. You must load the register with this address value before you execute the macro. If you do not specify otherwise, the default value is PARAM = (1).

No other operands are evaluated in the E form.

MF=L [,operand 1,...,operand n]

Only the data area is generated, taking the operand values specified in the macro call into account. The data area does not contain any field names or explanatory equates. The standard header (see page 21) is initialized. The macro call is issued in the definition segment of the program.
In the case of shared code programming, this call must not be located in the invariable program segment if it contains variable data. The data area is initialized with constant values in the invariable program segment, copied to a data area specific to this run before the E form of the macro is called, and updated there if necessary. The update operation is implemented with the M form.

MF=M ,operand 1,...,operand n [,PREFIX=p][,MACID=mac]

Instructions (e.g. MVCs) are generated. During program execution, fields in a data area previously initialized with MF=L or, in the case of shared code programming, in a local copy of the data area initialized with MF=L, are overwritten by these instructions with the operand values specified when the macro was called. This means that the M form of the macro call provides a convenient means of **dynamically** tailoring the operand values with which an ADAM function is called to the program run.

If you specify MF=M, no default values are assumed for function operands, i.e. you must specify all operands explicitly.
Since the instructions generated for MF=M use the symbolic addresses and equates of the C or D form, you must ensure in the M form that these names are available for addressing the data area to be modified. In particular, you must ensure that any values specified for the PREFIX and MACID operands in a macro call with MR=M are the same as in the corresponding MF=C or MF=D call.
You can use the PREFIX operand to define the first character of the field names and equates. p = 1 letter.
The MACID operand allows you to define the second, third and fourth characters of the field names and equates. mac = 1..3 characters.

## 3.3 Macro return codes

After each macro call, a return code is stored in the standard header. The return code contains information on the success or failure of the macro called.
The standard header has the same structure for all the macros. The only difference is to be found in the symbolic field names, the first character (PREFIX=<p>) and subsequent three characters (MACID=<mac>) of which may vary.
The names of the return code fields are MRET, SR1 and SR2; they are printed in bold type below.

*Standard header*

```
<p><mac>HDR  DS    0A
<p><mac>FHE  DS    0XL8            0   GENERAL PARAMETER AREA HEADER
*
<p><mac>IFID DS    0A              0   INTERFACE IDENTIFIER
<p><mac>FCTU DS    AL2             0   FUNCTION UNIT NUMBER
*                                  BIT 15    HEADER FLAG BIT,
*                                  MUST BE RESET UNTIL FURTHER NOTICE
*                                  BIT 14-12 UNUSED, MUST BE RESET
*                                  BIT 11-0  REAL FUNCTION UNIT NUMBER
<p><mac>FCT  DS    AL1             2   FUNCTION NUMBER
<p><mac>FCTV DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
*
<p><mac>RET  DS    0A              4   GENERAL RETURN CODE
<p><mac>SRET DS    0AL2            4   SUB RETURN CODE
<p><mac>SR2  DS    AL1             4   SUB RETURN CODE 2
<p><mac>SR1  DS    AL1             5   SUB RETURN CODE 1
<p><mac>MRET DS    0AL2            6   MAIN RETURN CODE
<p><mac>MR2  DS    AL1             6   MAIN RETURN CODE 2
<p><mac>MR1  DS    AL1             7   MAIN RETURN CODE 1
<p><mac>FHL  EQU   8               8   GENERAL OPERAND LIST HEADER LENGTH
```

### Meaning of subcode 1 (SR1)

Subcode 1 divides the return codes into classes.

**00** **No error.**
The job was processed in its entirety and without error.

**01** **Parameter error**
The job data contains an error. This is usually a programming error. Once the job data has been corrected, the job can be repeated.

**20** **Internal error**
An error occurred during job processing. This error is located in one of the system components required by ADAM. You cannot resolve an error of this class yourself. Usually, the only reasonable course of action is to abort the user program.

**40** **Other error**
This class of error encompasses errors to which you must respond individually, depending on the nature of the particular error involved.

### General return codes for ADAM macros

| SC2 | SC1 | Maincode | Meaning |
|---|---|---|---|
| 00 | 00 | 0000 | The job was processed successfully. |
| 01 | 00 | 0000 | There was no action to be taken.<br>The job results existed before the macro was called. |
| 02 | 00 | xxxx | The job was processed successfully.<br>The user's attention is drawn to exceptional situations that occurred. |
| 00 | 02 | FFFF | ADAM is not available. |
| 00 | 03 | FFFF | The ADAM version in the standard header of the caller is not supported. |
| 00 | 41 | FFFF | Error when loading the ADAM subsystem. |

ADAM does not issue any messages when an error occurs.

If calling a system interface from ADAM leads to an error and the return code indicates that the subordinate function has not already written a SERSLOG entry, ADAM causes a SERSLOG entry to be generated. This enables the ADAM development to diagnose system errors.

## 3.4 Representation of macro syntax

The macros in this manual are represented using a uniform macro call format. This macro call format consists of two columns, the first of which contains the macro name while the second column contains the possible operands.

**Macro call format**

| Macro name | Operands |
|---|---|
| <macro name> | <operand $_1$><br>,<operand $_2$><br>...<br>,<operand $_n$> |

When issuing a macro you must separate the macro name from the first operand by at least one blank column. If you specify more than one operand, you must separate the operands with commas.

Certain characters (metacharacters) are used in the macro call format. These characters are explained in the table below.

### 3.4.1 Elements of the metasyntax

| **Formal notation** | **Meaning** | **Examples** |
|---|---|---|
| UPPERCASE LETTERS | Uppercase letters denote keywords or constants that the user must enter exactly as shown. Keywords must start with * if both keywords and, as an alternative, the names of constants or variables can be specified. | DIB<br>FORCED=*YES |
| lowercase letters | Lowercase letters denote the data type of values or variables that can be specified by the user. | DIB = <var: pointer> |
| < > | Angle brackets denote variables whose set of values is described by the data types indicated. | <var: pointer> |
| Underscoring | Underscoring indicates the default value for an operand. If an operand does not have a default value, an operand value must be specified. | FORCED = *NO / *YES |

Table 2: Elements of the metasyntax

| **Formal notation** | **Meaning** | **Examples** |
|---|---|---|
| = | The equals sign links an operand name with the associated operand values. | DATA = <var: pointer> |
| / | The slash separates alternative operand values. | FORCED = *NO / *YES |
| list-poss(n) | A list can be created from the operand values that follow list-poss. n is the maximum number of elements in a list. If a list contains more than one element, it must be enclosed in parentheses. | FLAG=list-poss(3):<br>*SLI / *SKIP / *DC<br><br>Specify:<br>FLAG=*SKIP<br>FLAG=(*SLI,*DC) |

Table 2: Elements of the metasyntax

An operand is assigned an operand value using an equals sign. The operand value is one of a set of possible values.
The set of possible operand values is determined by a data type. table 3contains the data types of the operand values.

## 3.4.2 Data types of the operand values

| **Data type** | **Character set** | **Comments** |
|---|---|---|
| c-string | EBCDIC characters | must be enclosed in single quotes |
| integer | $-2^{31}$ to $2^{31}$ -1<br>(-2147483648 to 2147483647) | is a decimal number |
| var: | Indicates the start of a variable specification. The variable type is indicated after the colon (see table 5) | <var:var-type> |
| reg: | Register 0..15 | Specify:<br>(<reg:var-type>) |

Table 3: Data types of the operand values

### 3.4.3 Suffixes for the data types of operand values

| Suffix | Meaning |
|---|---|
| n..m | For the data type integer, n..m indicates a range of values;<br>n: minimum value<br>m: maximum value |
| | For the data type c-string, n..m is a length specification in bytes;<br>n: minimum length<br>m: maximum length<br>where n < m |
| n | For the data type c-string, n is a length specification in bytes;<br>n must be observed exactly. |

Table 4: Suffixes for data types

The operand values can be specified directly as a string or integer (see data types "c-string" and "integer") or indirectly by means of a variable (see data type "var:"). table 5lists the data types that are possible for variables.

### 3.4.4 Data types of the variables (var-type)

| Data type | Description | Definition in program |
|---|---|---|
| char:n | The variable is a string with n characters. If the length specification is omitted, n=1 is used. | CLn |
| int:n | The variable is an integer that occupies n bytes in binary form. If n is not specified, n=1 is used.<br>Prerequisite: $n \leq 4$ | FLn |
| pointer | The variable is an address or an address value. | A |

Table 5: Data types of the variables

*Examples*

Various register specifications and their meanings:

| | |
|---|---|
| <reg: int> | corresponds to <reg: int:1><br>The register contains an integer from the range - $2^7$ to $2^7$ -1 in the least significant byte. |
| <reg: int:2> | The register contains an integer from the range -$2^{15}$ to $2^{15}$ -1 in the two least significant bytes. |
| <reg: int:4> | The register contains an integer from the range -$2^{31}$ to $2^{31}$ -1 that occupies 4 bytes. |

# 3.5 Macro overview

## 3.5.1 ADAM macros for device access

| Macro name | Function description | Page |
|---|---|---|
| FOPNDV | Open device control. | 65 |
| FCLSDV | Close device control. | 34 |
| FDEVIB | Create a device information block. | 48 |

## 3.5.2 ADAM macros for device jobs

| Macro name | Function description | Page |
|---|---|---|
| FDEFIO | Define a device job. | 40 |
| FDECCP | Generate and store a channel program. | 37 |
| FDELCP | Delete channel programs. | 45 |

## 3.5.3 ADAM macros for I/O operations

| Macro name | Function description | Page |
|---|---|---|
| FTIMOT | Define a maximum time for waiting for I/O termination. | 72 |
| FEXIOW | Initiate an I/O operation and wait for termination (synchronous I/O operation). | 58 |
| FEXCIO | Initiate an I/O operation and do not wait for termination (asynchronous I/O operation). | 51 |
| FHLTIO | Force termination of an asynchronous I/O operation. | 63 |
| FCEKIO | Check the status of an asynchronous I/O operation and perform termination processing if the I/O operation has been completed. | 28 |
| FWFTIO | Wait for termination of an asynchronous I/O operation and perform termination processing. | 78 |

### 3.5.4 ADAM macros for signal processing

| **Macro name** | **Function description** | **Page** |
|---|---|---|
| FRECSG | Request delivery of the next device signal. | 68 |
| FCEKSG | Check the status of a requested device signal and perform termination processing if the signal has arrived. | 31 |
| FWFSG | Wait for the arrival of the next device signal and perform termination processing. | 75 |

All the macros are described in alphabetical order in the following sections.

## 3.6 Macros

### FCEKIO
### Check I/O status

Macro type: S (action macro)

#### Macro description

This macro allows you to check the status of a specific asynchronous I/O operation that was started by means of the FEXCIO macro. If the I/O operation has been completed, ADAM termination processing is performed.

Information on the result of the I/O operation is stored in the device information block (DIB). You must initialize the DIB by means of the FDEVIB macro before calling the FCEKIO macro.

#### Macro call format

| Macro name | Operands |
|---|---|
| FCEKIO | MF = C / D / E / L / M<br>,DEVID = <var: int:4> / (<reg: int:4>)<br>,IOID = <var: int:4> / (<reg: int:4>)<br>,DIB = <var: pointer> / (<reg: pointer>) |

#### Operand description

**MF = C / D / E / L / M**
See page 19 for the meanings of the operand values.

**DEVID =**
Indicates the device whose I/O status is to be checked. The device ID is returned when you open device control by means of the FOPNDV macro.

> **<var: int:4>**
> Specification of a symbolic field name; the field contains the device ID as an integer (4 bytes).
>
> **(<reg: int:4>)**
> Specification of a register; the register contains the device ID as an integer (4 bytes).

**IOID =**
Indicates a specific I/O operation whose status is to be checked.
The ID for an I/O operation is returned when you start an asynchronous I/O operation by means of the FEXCIO macro.
If the I/O operation has not yet been completed, you are informed of this by means of the return code stored in the standard header.
If the I/O operation has been completed, ADAM performs termination processing. This means that the I/O operation is ended in the normal manner.

**<var: int:4>**
Specification of a symbolic field name; the field contains the I/O identifier as an integer (4 bytes).

**(<reg: int:4>)**
Specification of a register; the register contains the I/O identifier as an integer (4 bytes).

**DIB =**
Indicates the device information block (DIB). You must specify the address of the DIB that you generated with the FDEVIB macro. If the I/O operation is terminated with an error, supplementary information is stored in the DIB.

**<var: pointer>**
There are two ways of specifying the address of the DIB:

- Specification of a symbolic field name: the field contains the address.
  This specification is only possible in conjunction with MF=M. In addition, an operand list must be generated with MF=C or MF=D.
- Direct specification in the form A(field).

**(<reg: pointer>)**
Specification of a register containing the address of the DIB.

### Return codes

| SC2 | SC1 | Maincode | Meaning |
|---|---|---|---|
| 00 | 00 | 0000 | I/O terminated without error. |
| 00 | 01 | 0400 | Invalid value for I/O identifier (IOID operand). |
| 00 | 01 | 0408 | Invalid value for device ID (DEVID operand). |
| 00 | 01 | 0418 | Invalid DIB address (DIB operand). |
| xx | 20 | 0C00 | System error:<br>The subcode 2 (SC2) contains a system error code intrinsic to ADAM. |
| 00 | 40 | 0010 | I/O not yet terminated. |
| 00 | 40 | 0014 | I/O terminated with error. |
| 00 | 40 | 0018 | I/O terminated by means of FHLTIO. |
| 00 | 40 | 0820 | I/O identifier refers to an I/O operation for which ADAM termination processing has already been performed. |
| 00 | 40 | 084C | I/O could not be started - wait and start again. |

## Assembler DSECT

```
         FCEKIO MF=D
         MFTST MF=D,PREFIX=N,MACID=DA5,ALIGN=F,                        C
               DMACID=DA5,SUPPORT=(E,D,C,M,L),DNAME=DA5CKIO
NDA5CKIO DSECT ,
               *,##### PREFIX=N, MACID=DA5 ####
*   **** ===== FCEKIO PARAMETER AREA =====
NDA5HDR  FHDR  MF=(C,NDA5),EQUATES=NO        ---- FHDR -----------------
NDA5HDR  DS    0A
NDA5FHE  DS    0XL8            0   GENERAL PARAMETER AREA HEADER
*
NDA5IFID DS    0A              0   INTERFACE IDENTIFIER
NDA5FCTU DS    AL2             0   FUNCTION UNIT NUMBER
*                                  BIT 15    HEADER FLAG BIT,
*                                  MUST BE RESET UNTIL FURTHER NOTICE
*                                  BIT 14-12 UNUSED, MUST BE RESET
*                                  BIT 11-0  REAL FUNCTION UNIT NUMBER
NDA5FCT  DS    AL1             2   FUNCTION NUMBER
NDA5FCTV DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
*
NDA5RET  DS    0A              4   GENERAL RETURN CODE
NDA5SRET DS    0AL2            4   SUB RETURN CODE
NDA5SR2  DS    AL1             4   SUB RETURN CODE 2
NDA5SR1  DS    AL1             5   SUB RETURN CODE 1
NDA5MRET DS    0AL2            6   MAIN RETURN CODE
NDA5MR2  DS    AL1             6   MAIN RETURN CODE 2
NDA5MR1  DS    AL1             7   MAIN RETURN CODE 1
NDA5FHL  EQU   8               8   GENERAL OPERAND LIST HEADER LENGTH
*
*   main return codes
NDA5NOE  EQU   0                         No error
NDA5III  EQU   1024                      Invalid value of IOID
NDA5IDI  EQU   1032                      Invalid value of DEVID
NDA5IDB  EQU   1048                      Invalid address of DIB
NDA5SYE  EQU   3072                      System error
NDA5INT  EQU   16                        IO not yet terminated
NDA5IOE  EQU   20                        IO terminated with error
NDA5IOH  EQU   24                        IO halted
NDA5INR  EQU   2080                      IO identifier refers not to an
*                                        incomplete IO
NDA5INS  EQU   2124                      IO could not be started - wait
*                                        and retry
*
*
NDA5INP  DS    0XL24                     ---- Input Parameters -----
NDA5DEVI DS    F                         Device-ID
NDA5FLG1 DS    XL1                       *** reserved ***
NDA5FLG2 DS    XL1                       *** reserved ***
NDA5FLG3 DS    XL1                       *** reserved ***
NDA5FLG4 DS    XL1                       *** reserved ***
NDA5IOID DS    F                         IO identifier
NDA5DIB@ DS    A                         Address of DIB
NDA5CIB@ DS    A                         Address of CIB
NDA5RSV1 DS    XL4                       *** reserved ***
*
NDA5#    EQU   *-NDA5HDR
```

# FCEKSG
# Check device signal status

Macro type: S (action macro)

## Macro description

This macro allows you to check whether a device signal (attention interrupt) has arrived and initiate termination processing for this signal. You must have submitted a request to receive a device signal to ADAM beforehand (FRECSG macro).

If a device signal has arrived, ADAM transfers the interrupt data to the device information block (DIB) and performs termination processing for this signal.You must initialize the DIB by means of the FDEVIB macro before you call the FCEKSG macro.
You transfer the DIB address in the DIB operand.

If no device signal has been received, an appropriate return code is issued. You must call the FCEKSG or FWFSG macro again in order to complete the receive signal job. Otherwise no new receive signal job (FRECSG macro) can be requested.

## Macro call format

| Macro name | Operands |
|---|---|
| `FCEKSG` | `MF = C / D / E / L / M`<br>`,DEVID = <var: int:4> / (<reg: int:4>)`<br>`,DIB = <var: pointer> / (<reg: pointer>)` |

## Operand description

**MF = C / D / E / L / M**
See page 19 for the meanings of the operand values.

**DEVID =**
Indicates the device whose signals are to be checked. The device ID is returned when you open device control by means of the FOPNDV macro.

**<var: int:4>**
Specification of a symbolic field name; the field name contains the device ID as an integer (4 bytes).

**(<reg: int:4>)**
Specification of a register; the register contains the device ID as an integer (4 bytes).

**DIB =**
Indicates the device information block (DIB). You must specify the address of the DIB that you generated by means of the FDEVIB macro.
The interrupt data is stored in the DIB.

**<var: pointer>**
There are two ways of specifying the address of the DIB:

- Specification of a symbolic field name: the field contains the address.
  This specification is only possible in conjunction with MF=M. In addition, an operand list must be generated with MF=C or MF=D.
- Direct specification in the form A(field)

**(<reg: pointer>)**
Specification of a register containing the address of the DIB.

### Return codes

| SC2 | SC1 | Maincode | Meaning |
|---|---|---|---|
| 00 | 00 | 0000 | No error; device signal has arrived. |
| 00 | 01 | 0408 | Invalid device ID (DEVID operand). |
| 00 | 01 | 0418 | Invalid DIB address (DIB operand). |
| xx | 20 | 0C00 | System error:<br>The subcode 2 (SC2) contains a system error code intrinsic to ADAM. |
| 00 | 40 | 0010 | Signal not yet arrived. |
| 00 | 40 | 0820 | No receive signal job (FRECSG macro) requested. |

## Assembler DSECT

```
         FCEKSG MF=D
         MFTST MF=D,PREFIX=N,MACID=DAV,ALIGN=F,                        C
               DMACID=DAV,SUPPORT=(E,D,C,M,L),DNAME=DAVCKSG
NDAVCKSG DSECT ,
               *,##### PREFIX=N, MACID=DAV ####
*   **** ===== FCECSG PARAMETER AREA =====
NDAVHDR  FHDR  MF=(C,NDAV),EQUATES=NO        --- FHDR -----------------
NDAVHDR  DS    0A
NDAVFHE  DS    0XL8            0   GENERAL PARAMETER AREA HEADER
*
NDAVIFID DS    0A              0   INTERFACE IDENTIFIER
NDAVFCTU DS    AL2             0   FUNCTION UNIT NUMBER
*                                  BIT 15    HEADER FLAG BIT,
*                                  MUST BE RESET UNTIL FURTHER NOTICE
*                                  BIT 14-12 UNUSED, MUST BE RESET
*                                  BIT 11-0  REAL FUNCTION UNIT NUMBER
NDAVFCT  DS    AL1             2   FUNCTION NUMBER
NDAVFCTV DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
*
NDAVRET  DS    0A              4   GENERAL RETURN CODE
NDAVSRET DS    0AL2            4   SUB RETURN CODE
NDAVSR2  DS    AL1             4   SUB RETURN CODE 2
NDAVSR1  DS    AL1             5   SUB RETURN CODE 1
NDAVMRET DS    0AL2            6   MAIN RETURN CODE
NDAVMR2  DS    AL1             6   MAIN RETURN CODE 2
NDAVMR1  DS    AL1             7   MAIN RETURN CODE 1
NDAVFHL  EQU   8               8   GENERAL OPERAND LIST HEADER LENGTH
*
*   main return codes
NDAVNOE  EQU   0                         No error
NDAVIDI  EQU   1032                      Invalid value of DEVID
NDAVIDB  EQU   1048                      Invalid address of DIB
NDAVSYE  EQU   3072                      System error
NDAVSNA  EQU   16                        Signal not yet arrived
NDAVNSR  EQU   2080                      No receive signal job
*                                        requested
*
*
NDAVINP  DS    0XL24                     --- Input Parameters -----
NDAVDEVI DS    F                         Device-ID
NDAVFLG1 DS    XL1                       *** reserved ***
NDAVFLG2 DS    XL1                       *** reserved ***
NDAVFLG3 DS    XL1                       *** reserved ***
NDAVFLG4 DS    XL1                       *** reserved ***
NDAVRSV1 DS    XL4                       *** reserved ***
NDAVDIB@ DS    A                         Address of DIB
NDAVRSV2 DS    XL4                       *** reserved ***
NDAVRSV3 DS    XL4                       *** reserved ***
*
NDAV#    EQU   *-NDAVHDR
```

# FCLSDV
# Close device control

Macro type: S (action macro)

## Macro description

This macro allows you to close device control.

Once all I/O operations and receive signal jobs have been completed, device control is terminated, and any channel programs declared by means of the FDECCP macro are automatically deleted.

If I/O operations are still running, or a device signal has been received but the receive job has not yet been completed (FCEKSG or FWFSG macro), the FCLSDV macro is executed as defined in the FORCED operand:

- FORCED=*NO
  Default value; device control is not terminated, an appropriate return code is transferred. In this case, you must terminate any current I/O operations or receive signal jobs (FCEKSG macro) and issue the FCLSDV macro again.
- FORCED=*YES
  Any current I/O operations in programs and requested receive signal jobs for device signals are aborted. Device control is terminated and all channel programs declared by means of the FDECCP macro are automatically deleted.

## Macro call format

| Macro name | Operands |
|---|---|
| FCLSDV | MF = C / D / E / L / M<br>,DEVID = <var: int:4> / (<reg: int:4>)<br>,FORCED = *NO / *YES |

## Operand description

**MF = C / D / E / L / M**
See page 19 for the meanings of the operand values.

**DEVID =**
Indicates the device for which device control is to be terminated. The device ID is returned when you open device control by means of the FOPNDV macro.

**<var: int:4>**
Specification of a symbolic field name; the field contains the device ID as an integer (4 bytes).

**(<reg: int:4>)**
Specification of a register; the register contains the device ID as an integer (4 bytes).

**FORCED =**
Indicates whether device control is to be terminated even if I/O operations are still in progress or receive signal jobs have not yet been completed.

***NO**
Device control is only terminated if there are no I/O operations in progress and no receive signal jobs that have yet to be completed. This is the default value.

***YES**
Device control is terminated. Any current I/O operations and receive signal jobs in progress are aborted. Information regarding I/O operations or device signals is lost.

### Return codes

| SC2 | SC1 | Maincode | Meaning |
|---|---|---|---|
| 00 | 00 | 0000 | No error; device control closed. |
| 02 | 00 | 0230 | Function executed with warnings.<br>I/O operation or receive signal job aborted because FORCED=*YES was specified. |
| 00 | 01 | 0408 | Invalid device ID (DEVID operand). |
| xx | 20 | 0C00 | System error:<br>The subcode 2 (SC2) contains a system error code intrinsic to ADAM. |
| 00 | 40 | 0830 | I/O or signal processing incomplete, device control has not been terminated. |

### Assembler DSECT

```
         FCLSDV MF=D
         MFTST MF=D,PREFIX=N,MACID=DA2,ALIGN=F,                        C
               DMACID=DA2,SUPPORT=(E,D,C,M,L),DNAME=DA2CLDV
NDA2CLDV DSECT ,
               *,##### PREFIX=N, MACID=DA2 ####
*   **** ===== FCLSDV PARAMETER AREA =====
NDA2HDR  FHDR  MF=(C,NDA2),EQUATES=NO        --- FHDR -----------------
NDA2HDR  DS    0A
NDA2FHE  DS    0XL8            0   GENERAL PARAMETER AREA HEADER
*
NDA2IFID DS    0A              0   INTERFACE IDENTIFIER
NDA2FCTU DS    AL2             0   FUNCTION UNIT NUMBER
*                                  BIT 15    HEADER FLAG BIT,
*                                  MUST BE RESET UNTIL FURTHER NOTICE
*                                  BIT 14-12 UNUSED, MUST BE RESET
*                                  BIT 11-0  REAL FUNCTION UNIT NUMBER
NDA2FCT  DS    AL1             2   FUNCTION NUMBER
NDA2FCTV DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
*
NDA2RET  DS    0A              4   GENERAL RETURN CODE
NDA2SRET DS    0AL2            4   SUB RETURN CODE
NDA2SR2  DS    AL1             4   SUB RETURN CODE 2
NDA2SR1  DS    AL1             5   SUB RETURN CODE 1
NDA2MRET DS    0AL2            6   MAIN RETURN CODE
NDA2MR2  DS    AL1             6   MAIN RETURN CODE 2
NDA2MR1  DS    AL1             7   MAIN RETURN CODE 1
NDA2FHL  EQU   8               8   GENERAL OPERAND LIST HEADER LENGTH
*
*   main return codes
NDA2NOE  EQU   0                         No error
NDA2FPS  EQU   560                       Function processed with
*                                        warnings
NDA2IDI  EQU   1032                      Invalid value of DEVID
NDA2SYE  EQU   3072                      System error
NDA2IOE  EQU   2096                      IO or signal processing
*                                        incomplete, device operation
*                                        not terminated
*
*
NDA2INP  DS    0XL8                      --- Input Parameters -----
NDA2DEVI DS    F                         Device-ID
NDA2FLG1 DS    AL1                       Flag-byte 1
NDA2FRC  EQU   X'80'                     - FORCED=*YES
NDA2RS1  EQU   X'7F'                     - reserved
NDA2FLG2 DS    XL1                       *** reserved ***
NDA2FLG3 DS    XL1                       *** reserved ***
NDA2FLG4 DS    XL1                       *** reserved ***
*
NDA2#    EQU   *-NDA2HDR
```

# FDECCP
## Generate and store channel program

Macro type: S (action macro)

### Macro description

This macro makes a device job chain, from which ADAM generates a channel program, known to ADAM.
The device job chain is defined by means of the FDEFIO macro.
The channel program ID is returned in a field. You use the CPIDRET operand to specify the address under which the contents of this field are to be stored. You can reference the generated channel program in the FEXCIO or FEXIOW macro by means of the channel program ID (CPID operand).
No more than 63 FDECCP-generated channel programs are permitted per device.

### Macro call format

| Macro name | Operands |
|---|---|
| `FDECCP` | ` MF = C / D / E / L / M`<br>`,DEVID = <var: int:4> / (<reg: int:4>)`<br>`,IOPRG = <var: pointer> / (<reg: pointer>)`<br>`,CPIDRET = <var: pointer> / (<reg: pointer>)` |

### Operand description

**MF = C / D / E / L / M**
See page 19 for the meanings of the operand values.

**DEVID =**
Indicates the device for which a channel program is to be generated. The device ID is returned when you open device control by means of the FOPNDV macro.

**<var: int:4>**
Specification of a symbolic field name; the field name contains the device ID as an integer (4 bytes).

**(<reg: int:4>)**
Specification of a register; the register contains the device ID as an integer (4 bytes).

**IOPRG =**
Indicates the address of a specific device job or the address of the first job in the device job chain. The I/O job indicated here refers to the device job (or device job chain) defined by means of the FDEFIO macro.

**<var: pointer>**
There are two ways of specifying the address of the device job:

- Specification of a symbolic field name: the field contains the address.
  This specification is only possible in conjunction with MF=M. In addition, an operand list must be generated with MF=C or MF=D.
- Direct specification in the form A(field).

**(<reg: pointer>)**
Specification of a register containing the address of the device job.

**CPIDRET =**
Indicates the address of a field in the user address space. ADAM stores the channel program ID in this field.

**<var: pointer>**
There are two ways of specifying the address of this field:

- Specification of a symbolic field name: the field contains the address.
  This specification is only possible in conjunction with MF=M. In addition, an operand list must be generated with MF=C or MF=D.
- Direct specification in the form A(field).

**(<reg: pointer>)**
Specification of a register containing the address of the field.

### Return codes

| SC2 | SC1 | Maincode | Meaning |
|---|---|---|---|
| 00 | 00 | 0000 | No error; channel program generated. |
| 00 | 01 | 0404 | Invalid device job address (IOPRG operand). |
| 00 | 01 | 0408 | Invalid device ID (DEVID operand). |
| 00 | 01 | 040C | Invalid address for the field in which the channel program ID (CPIDRET) is stored. |
| 00 | 01 | 0420 | Invalid I/O buffer address (DATA operand). |
| 00 | 01 | 0424 | Invalid data length (LENGTH operand). |
| 00 | 01 | 0428 | Invalid instruction code (CMD operand). |
| 00 | 01 | 042C | Invalid device job option (FLAG operand). |
| 00 | 01 | 0430 | Device job chain is too long. |
| 00 | 01 | 0434 | Invalid continuation address in device job (CHAIN operand). |
| 00 | 01 | 0438 | Device job option DC not permitted for this device. |
| 00 | 01 | 043C | Device job option DC not permitted for this operation. |
| 00 | 01 | 0444 | Device job option DC incompatible with continuation address. |
| xx | 20 | 0C00 | System error:<br>The subcode 2 (SC2) contains a system error code intrinsic to ADAM. |
| 00 | 40 | 0834 | Maximum number of channel programs exceeded. |

## Assembler DSECT

```
         FDECCP MF=D
         MFTST MF=D,PREFIX=N,MACID=DA8,ALIGN=F,                        C
               DMACID=DA8,SUPPORT=(E,D,C,M,L),DNAME=DA8DCCP
NDA8DCCP DSECT ,
               *,##### PREFIX=N, MACID=DA8 ####
*   **** ===== FDECCP PARAMETER AREA =====
NDA8HDR  FHDR  MF=(C,NDA8),EQUATES=NO        --- FHDR -----------------
NDA8HDR  DS    0A
NDA8FHE  DS    0XL8            0   GENERAL PARAMETER AREA HEADER
NDA8IFID DS    0A              0   INTERFACE IDENTIFIER
NDA8FCTU DS    AL2             0   FUNCTION UNIT NUMBER
*                                  BIT 15    HEADER FLAG BIT,
*                                  MUST BE RESET UNTIL FURTHER NOTICE
*                                  BIT 14-12 UNUSED, MUST BE RESET
*                                  BIT 11-0  REAL FUNCTION UNIT NUMBER
NDA8FCT  DS    AL1             2   FUNCTION NUMBER
NDA8FCTV DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
*
NDA8RET  DS    0A              4   GENERAL RETURN CODE
NDA8SRET DS    0AL2            4   SUB RETURN CODE
NDA8SR2  DS    AL1             4   SUB RETURN CODE 2
NDA8SR1  DS    AL1             5   SUB RETURN CODE 1
NDA8MRET DS    0AL2            6   MAIN RETURN CODE
NDA8MR2  DS    AL1             6   MAIN RETURN CODE 2
NDA8MR1  DS    AL1             7   MAIN RETURN CODE 1
NDA8FHL  EQU   8               8   GENERAL OPERAND LIST HEADER LENGTH
*   main return codes
NDA8NOE  EQU   0                         No error
NDA8IDJ  EQU   1028                      Invalid address of Device Job
*                                        (IOPRG)
NDA8IDI  EQU   1032                      Invalid value of DEVID
NDA8IIR  EQU   1036                      Invalid address CPIDRET
NDA8IIB  EQU   1056                      Invalid address of IO buffer
*                                        (DATA)
NDA8IDL  EQU   1060                      Invalid data length (LENGTH)
NDA8IOP  EQU   1064                      Invalid ADAM Op-Code (CMD)
NDA8IJO  EQU   1068                      Invalid Device Job option (FLAG)
NDA8CTL  EQU   1072                      Device job chain too long
NDA8ICA  EQU   1076                      Invalid continuation address
*                                        in Device Job (CHAIN)
NDA8IDD  EQU   1080                      Invalid option DC, not allowed
*                                        for this device
NDA8IDO  EQU   1084                      Invalid option DC, not allowed
*                                        for this operation
NDA8IDC  EQU   1092                      Invalid option DC,
*                                        incompatible with
*                                        continuation address
NDA8SYE  EQU   3072                      System error
NDA8MCE  EQU   2100                      Maximum number of channel
*                                        programs exceeded
NDA8INP  DS    0XL24                     --- Input Parameters -----
NDA8DEVI DS    F                         Device-ID
NDA8FLG1 DS    XL1                       *** reserved ***
NDA8FLG2 DS    XL1                       *** reserved ***
NDA8FLG3 DS    XL1                       *** reserved ***
NDA8FLG4 DS    XL1                       *** reserved ***
NDA8DJB@ DS    A                         Address of Device Job
NDA8CPI@ DS    A                         Address of return-field for CPID
NDA8RSV1 DS    XL4                       *** reserved ***
NDA8RSV2 DS    XL4                       *** reserved ***
NDA8#    EQU   *-NDA8HDR
```

# FDEFIO
# Define device job

Macro type: O (definition macro)

### Macro description

This macro allows you to define and describe the data area of an I/O job.

### Macro call format

| Macro name | Operands |
|---|---|
| FDEFIO | MF = C / D / L / M<br>,CMD = <integer 1..255> / <var: int:1> / (<reg: int:1>)<br>,DATA = <var: pointer> / (<reg: pointer>)<br>,LENGTH = <integer 0..65535> / <var: int:2> / (<reg: int:2>)<br>,FLAG = list-poss(3): *SLI / *SKIP / *DC<br>,CHAIN = *NO / <var: pointer> / (<reg: pointer>)<br>,ATTR = *NO / *RESERVE / *RELEASE |

*Illegal operands*

The operand FLAG=*DC (data chaining) cannot be specified in conjunction with CHAIN=*NO.

### Operand description

**MF = C / D / L / M**
See page 19 for the meanings of the operand values.

**CMD =**
Indicates the ADAM instruction code of the I/O job. Depending on the device type, the ADAM instruction code corresponds to the physical instruction code of the device. If this is not the case, the ADAM instruction code is assigned to the physical instruction code using device-specific information from the device modules.
Please refer to the device description of the device being referenced for additional information on the instruction code.

**<integer 1..255>**
Direct specification of the instruction code as an integer (in decimal form).

**<var: int:1>**
Specification of a symbolic field name; the field name contains the instruction code as an integer (1 byte).

**(<reg: int:1>)**
Specification of a register; the register contains the instruction code as an integer (1 byte) in the least significant byte.

**DATA =**
Indicates the address of the I/O buffer for transfer data.
The address specification is ignored if the device description contains the information "no data transfer", or if the LENGTH operand is assigned the value 0).
If the Search-TIC permission bit is set in the device module, it is possible to specify a TIC device job with the aid of the FDEFIO macro.
This TIC device job is defined not only by the TIC instruction code but also by the fact that DATA contains the address of the previous device job.

**<var: pointer>**
There are two ways of specifying the address of the I/O buffer:

– Specification of a symbolic field name: the field contains the address. This specification is only possible in conjunction with MF=M. In addition, an operand list must be generated with MF=C or MF=D.
– Direct specification in the form A(field).

**(<reg: pointer>)**
Specification of a register containing the address of the I/O buffer.

**LENGTH =**
Indicates the number of bytes (data length) to be transferred during the I/O operation. This specification is ignored if the device description contains the information "no data transfer".
In operations of the "read backward" type, the data is read into the buffer in reverse, starting at the address DATA+LENGTH-1.
If you specify LENGTH=0, no data is transferred and specifications made for the DATA operand are therefore ignored.
The maximum number of transferred bytes for a device job is 65535. In the case of devices with a bus channel, up to 32768 bytes can be transferred.

**<integer 0..65535>**
Direct specification of the data length as an integer.

**<var: int:2>**
Specification of a symbolic field name; the field (2 bytes) contains the data length as an integer.

**(<reg: int:2>)**
Specification of a register; the register contains the data length as an integer (2 bytes) in the two least significant bytes.

**FLAG =**
Indicates an I/O option.

***SLI**
SLI stands for SUPPRESS LENGTH INDICATION.
If the number of bytes to be transferred is not the same as the physical record length, execution of the job chain is not to be aborted and no error flag is to be generated.

***SKIP**
SKIP stands for SKIP DATA TRANSFER.
In an operation of the type "read", no data is transferred to the I/O buffer.

***DC**
DC stands for DATA CHAINING.
The instruction code in the next device job in the device job chain is ignored. The data being sent to or from the device belongs to the same data block.
This I/O option is only permitted if it has been included in the device description.

**CHAIN =**
Indicates the address of the next device job in the device job chain.
You can use the CHAIN operand to create a device job chain by specifying the address of a device job that is to be executed after the current device job within the same I/O operation. A device job chain is terminated by a device job in which the CHAIN operand has the value *NO or points to the start address of the same device job chain.
The start address of a device job chain is specified by the IOPRG operand in the FDECCP, FEXCIO and FEXIOW macros.
It is also possible to form loops for certain devices. To do this, the address specified in the CHAIN operand must point to the start of the device job chain. A note to the effect that loop formation is possible must be included in the device module.

<u>***NO**</u>
There is no further device job.

**<var: pointer>**
There are two ways of specifying the address of the next device job:

- Specification of a symbolic field name: the field contains the address.
  This specification is only possible in conjunction with MF=M. In addition, an operand list must be generated with MF=C or MF=D.
- Direct specification in the form A(field).

**(<reg: pointer>)**
Specification of a register containing the address of the next device job.

**ATTR=**
This operand enables the user to specify a specific instruction attribute of the ADAM instruction code. In the device module there is a permission bit for each instruction attribute. If this bit is not set, the device job is rejected with the appropriate return code.
This operand may only be specified for particular devices.

***NO**
The ADAM instruction code is specified without an instruction attribute.

***RESERVE**
The input/output operation specified by the ADAM instruction code is a DEVICE-RESERVE operation.

***RELEASE**
The input/output operation specified by the ADAM instruction code is a DEVICE-RELEASE operation.

### Return codes

None.

## Assembler DSECT

```
        FDEFIO MF=D
         MFTST MF=D,PREFIX=N,MACID=DAI,ALIGN=F,                       C
               DMACID=DAI,SUPPORT=(D,C,M,L),DNAME=DAIDFIO
NDAIDFIO DSECT ,
               *,##### PREFIX=N, MACID=DAI ####
NDAICHNO EQU   -1                         Value for CHAIN=*NO
*
NDAICHNX EQU   -2                         Value for CHAIN=NEXT (A-intf.)*
*
*   **** ===== ADAM DEVICE JOB DEFINITION =====
NDAIIND  DS    XL1                        *** reserved ***
NDAIFLG2 DS    XL1                        *** reserved ***
NDAIFLG3 DS    XL1                        *** reserved ***
NDAICMD  DS    X                          ADAM Op-Code (CMD)
NDAIDATA DS    A                          Address of data buffer (DATA)
NDAIFLAG DS    AL1                        Device Job Option (FLAG)
NDAIDTCH EQU   X'80'                      - DC__: Data Chaining
NDAICMCH EQU   X'40'                      - reserved -
NDAISLI  EQU   X'20'                      - SLI_: Suppress Lgt. Indic.
NDAISKIP EQU   X'10'                      - SKIP: Skip Data Transfer
NDAIRS2  EQU   X'0F'                      - reserved -
NDAIFLG4 DS    AL1                        Op-Code attribute (FLAG)
NDAIRES  EQU   X'80'                      - ATTR=*RESERVE
NDAIREL  EQU   X'40'                      - ATTR=*RELEASE
NDAIRS3  EQU   X'3F'                      - reserved -
NDAILENG DS    H                          Byte count (LENGTH)
NDAIREP  DS    XL4                        *** reserved ***
NDAIDJB@ DS    A                          Address of next device job
*                                         (CHAIN)
NDAI#    EQU   *-NDAIIND
```

# FDELCP
# Delete channel programs

Macro type: S (action macro)

### Macro description

This macro allows you to delete channel programs generated by means of the FDECCP macro.

### Macro call format

| Macro name | Operands |
|---|---|
| `FDELCP` | `MF = C / D / E / L / M`<br>`,DEVID = <var: int:4> / (<reg: int:4>)`<br>`,CPID = *ALL / <var: int:4> / (<reg: int:4>)` |

### Operand description

**MF = C / D / E / L / M**
See page 19 for the meanings of the operand values.

**DEVID =**
Indicates the device whose predefined channel programs are to be deleted. The device ID is returned when you open device control by means of the FOPNDV macro.

**<var: int:4>**
Specification of a symbolic field name; the field name contains the device ID as an integer (4 bytes).

**(<reg: int:4>)**
Specification of a register; the register contains the device ID as an integer (4 bytes).

**CPID =**
Indicates the channel program to be deleted. The channel program ID is returned after execution of the FDECCP macro.

***ALL**
All the channel programs generated for the specified device are deleted.

**<var: int:4>**
Only the channel program specified here is deleted.
Specification of a symbolic field name; the field name contains the channel program ID as an integer (4 bytes).

**(<reg: int:4>)**
Only the channel program specified here is deleted.
Specification of a register; the register contains the channel program ID as an integer (4 bytes).

### Return codes

| SC2 | SC1 | Maincode | Meaning |
|---|---|---|---|
| 00 | 00 | 0000 | No error; channel program deleted. |
| 01 | 00 | 0000 | No deletion.<br>No predefined channel program exists. |
| 00 | 01 | 0408 | Invalid device ID (DEVID operand). |
| 00 | 01 | 0410 | Invalid channel program ID (CPID operand). |
| xx | 20 | 0C00 | System error:<br>The subcode 2 (SC2) contains a system error code intrinsic to ADAM. |
| 00 | 40 | 000C | One or more channel programs are still active. |

## Assembler DSECT

```
         FDELCP MF=D
         MFTST MF=D,PREFIX=N,MACID=DA9,ALIGN=F,                        C
               DMACID=DA9,SUPPORT=(E,D,C,M,L),DNAME=DA9DLCP
NDA9DLCP DSECT ,
               *,##### PREFIX=N, MACID=DA9 ####
*   **** ===== FDELCP PARAMETER AREA =====
NDA9HDR  FHDR  MF=(C,NDA9),EQUATES=NO        --- FHDR -----------------
NDA9HDR  DS    0A
NDA9FHE  DS    0XL8            0   GENERAL PARAMETER AREA HEADER
*
NDA9IFID DS    0A              0   INTERFACE IDENTIFIER
NDA9FCTU DS    AL2             0   FUNCTION UNIT NUMBER
*                                  BIT 15    HEADER FLAG BIT,
*                                  MUST BE RESET UNTIL FURTHER NOTICE
*                                  BIT 14-12 UNUSED, MUST BE RESET
*                                  BIT 11-0  REAL FUNCTION UNIT NUMBER
NDA9FCT  DS    AL1             2   FUNCTION NUMBER
NDA9FCTV DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
*
NDA9RET  DS    0A              4   GENERAL RETURN CODE
NDA9SRET DS    0AL2            4   SUB RETURN CODE
NDA9SR2  DS    AL1             4   SUB RETURN CODE 2
NDA9SR1  DS    AL1             5   SUB RETURN CODE 1
NDA9MRET DS    0AL2            6   MAIN RETURN CODE
NDA9MR2  DS    AL1             6   MAIN RETURN CODE 2
NDA9MR1  DS    AL1             7   MAIN RETURN CODE 1
NDA9FHL  EQU   8               8   GENERAL OPERAND LIST HEADER LENGTH
*
*   main return codes
NDA9NOE  EQU   0                         No error
NDA9NTD  EQU   0                         Nothing to do
NDA9IDI  EQU   1032                      Invalid value of DEVID
NDA9ICI  EQU   1040                      Invalid value of CPID
NDA9SYE  EQU   3072                      System error
NDA9CIU  EQU   12                        One or more channel programs
*                                        are still active
NDA9INP  DS    0XL24                     --- Input Parameters -----
NDA9DEVI DS    F                         Device-ID
NDA9FLG1 DS    AL1                       Flag-byte 1
NDA9ALL  EQU   X'80'                     : CPID=*ALL specified
NDA9RS1  EQU   X'7F'                     : - reserved -
NDA9FLG2 DS    XL1                       *** reserved ***
NDA9FLG3 DS    XL1                       *** reserved ***
NDA9FLG4 DS    XL1                       *** reserved ***
NDA9CPID DS    F                         Channel program ID
NDA9RSV1 DS    XL4                       *** reserved ***
NDA9RSV2 DS    XL4                       *** reserved ***
NDA9RSV3 DS    XL4                       *** reserved ***
*
NDA9#    EQU   *-NDA9HDR
```

# FDEVIB
# Create device information block

Macro type: O (definition macro)

### Macro description

This macro allows you to define the data structure of the device information block (DIB).

After an I/O or receive signal job has been executed, the following *device-specific data* is written to the DIB.

- Sense bytes (if any exist)
- Diagnostic status bytes

After execution of an I/O job, *I/O-specific data* is also stored in the DIB (FEXIOW, FCEKIO, FWFTIO macros).

- I/O status bytes
- Address of the first device job in the device job chain.

  The following data is transferred in the event of an I/O error:

  – Address of the last device job executed
  – Remainder byte counter for the last device job executed

### Macro call format

| Macro name | Operands |
|---|---|
| FDEVIB | MF = C / D / L / M |

### Operand description

**MF = C / D / L / M**
See page 19 for the meanings of the operand values.

The FEDVIB macro has no other operands; it does not call any function.

### Return codes

| SC2 | SC1 | Maincode | Meaning |
|---|---|---|---|
| 02 | 00 | 0001 | Valid device-specific and I/O-specific data, DIB data from I/O processing |
| 02 | 00 | 0002 | Valid device-specific data.<br>DIB data from signal processing. |

## Assembler DSECT

```
         FDEVIB MF=D
         MFTST MF=D,PREFIX=N,MACID=DAJ,ALIGN=F,                        C
               DMACID=DAJ,SUPPORT=(D,C,L),DNAME=DAJDIBL
NDAJDIBL DSECT ,
               *,##### PREFIX=N, MACID=DAJ ####
*   **** ===== ADAM DEVICE INFORMATION BLOCK =====
NDAJFHDR FHDR  MF=(C,NDAJ),EQUATES=NO                              FHDR
NDAJFHDR DS    0A
NDAJFHE  DS    0XL8            0   GENERAL PARAMETER AREA HEADER
*
NDAJIFID DS    0A              0   INTERFACE IDENTIFIER
NDAJFCTU DS    AL2             0   FUNCTION UNIT NUMBER
*                                  BIT 15    HEADER FLAG BIT,
*                                  MUST BE RESET UNTIL FURTHER NOTICE
*                                  BIT 14-12 UNUSED, MUST BE RESET
*                                  BIT 11-0  REAL FUNCTION UNIT NUMBER
NDAJFCT  DS    AL1             2   FUNCTION NUMBER
NDAJFCTV DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
*
NDAJRET  DS    0A              4   GENERAL RETURN CODE
NDAJSRET DS    0AL2            4   SUB RETURN CODE
NDAJSR2  DS    AL1             4   SUB RETURN CODE 2
NDAJSR1  DS    AL1             5   SUB RETURN CODE 1
NDAJMRET DS    0AL2            6   MAIN RETURN CODE
NDAJMR2  DS    AL1             6   MAIN RETURN CODE 2
NDAJMR1  DS    AL1             7   MAIN RETURN CODE 1
NDAJFHL  EQU   8               8   GENERAL OPERAND LIST HEADER LENGTH
*
*   main return codes
NDAJIOE  EQU   1                         Device- and IO specific data
*                                        valid
NDAJSIG  EQU   2                         Device specific data valid
*
*
NDAJOUT  DS    0XL56                     === Output Parameters ====
*
NDAJDEV  DS    0XL40                     --- Device specific data ----
*
NDAJSENS DS    0XL32                     Sense Bytes
*
NDAJOPSB DS    0XL3                      - Operational Status Bytes
NDAJSB1  DS    XL1                       - - Sense Byte 1
NDAJSB2  DS    XL1                       - - Sense Byte 2
NDAJSB3  DS    XL1                       - - Sense Byte 3
*
NDAJDSBS DS    XL29                      - Diagnostic Status Bytes
*
NDAJSDB1 DS    AL1                       Standard device byte 1 / Unit
*                                        status byte
NDAJEDRI EQU   X'80'                     : Attent.interrupt request/
*                                        :_Attention
NDAJTRIN EQU   X'40'                     : Termin. interrupt pend. /
*                                        :_Status modifier
NDAJDVBS EQU   X'20'                     : Device busy _ _ _ _ _ _ /
*                                        :_Control unit end
NDAJCNBS EQU   X'10'                     : Control unit busy _ _ _ /
*                                        :_Busy
NDAJDVEN EQU   X'08'                     : Device end __ _ _ _ _ _ /
*                                        :_Channel end
NDAJSCIN EQU   X'04'                     : Secondary indicator _ _ /
*                                        :_Device end
NDAJDVIN EQU   X'02'                     : Device inoperable _ _ _ /
```

```
*                                         :_Unit check
NDAJSTMO EQU   X'01'                      : Status modifier _ _ _ _ /
*                                         :_Unit exception
NDAJSDB2 DS    XL1                        Standard Device Byte 2
NDAJRES1 DS    XL2                        *** reserved ***
NDAJRES2 DS    XL4                        *** reserved ***
*
*
NDAJIOD  DS    0XL16                      ——— IO specific data —————
NDAJIOS1 DS    AL1                        IO Status Byte 1
NDAJRNME EQU   X'80'                      : DEVICE RESERVE no more
*                                         effective
NDAJCHER EQU   X'40'                      : Channel error
NDAJINCL EQU   X'20'                      : Incorrect length indicator
NDAJTRRS EQU   X'10'                      : Trunk reset
NDAJRE6  EQU   X'08'                      : - reserved -
NDAJCTNA EQU   X'04'                      : CC=3 : Control unit not
*                                         available, operator message
*                                         issued
NDAJCTBS EQU   X'02'                      : CC=2 : Control unit
*                                         permanently locked, operator
*                                         message issued
NDAJCTC1 EQU   X'01'                      : CC=1
NDAJIOS2 DS    AL1                        IO Status Byte 2
NDAJUNDF EQU   X'80'                      : Device status undefined
NDAJTIM  EQU   X'40'                      : Termin. interrupt missing
NDAJSBP  EQU   X'20'                      : Sense bytes generated
NDAJSNC  EQU   X'10'                      : Sense bytes incomplete
NDAJDBSY EQU   X'08'                      : Device busy
NDAJDINO EQU   X'04'                      : Device inoperable
NDAJHLT  EQU   X'02'                      : IO aborted (HLTIO)
NDAJETOP EQU   X'01'                      : Device detached by operator,
*                                         : device access right lost
NDAJRES3 DS    XL2                        *** reserved ***
NDAJGA   DS    A                          Addr. of first device job in
*                                         chain
NDAJGAAD DS    A                          Addr. of last device job
*                                         executed
NDAJRBYT DS    F                          Residual byte count of last
*                                         device job executed
*
*
NDAJ#    EQU   *-NDAJFHDR
```

# FEXCIO
# Initiate asynchronous I/O

Macro type: S (action macro)

### Macro description

This macro initiates execution of an asynchronous I/O operation, i.e. control is returned to the user immediately after initiation.
Up to two I/O operations can execute in parallel. Once the maximum number has been reached, any further I/O initiation attempt is rejected.
Once an I/O operation has been initiated successfully, the I/O identifier is stored in a field whose address you specified via the IOIDRET operand for this macro.
No I/O operation should be started while another is still in progress unless the two operations can be executed fully independently of each other. An I/O operation terminated as a result of an error causes no delay to other I/O operations.

To support user eventing, you must call the BS2000 macro ENAEI and provide it with the name for the event item and the scope. ENAEI sets up this event item for your task and returns an identifier for the event item (4 bytes, EIIDRET operand). You can use this event item ID to establish a relationship between your task and user eventing in the course of further processing with the FEXCIO macro (EIID operand).
You use the BS2000 macro SOLSIG to request a signal for your event item. The BS2000 macro POSSIG reports the arrival of an event to user eventing. The post code returned to user eventing by POSSIG informs you whether the event is relevant to you. If this is not the case, you must issue SOLSIG repeatedly until an event reported by POSSIG is the one you are waiting for.

Information on the occurrence of the "end of I/O" event is stored in the first byte of the post code. You can request this information via the POSTBYT operand of the FEXCIO macro.

If you wish to terminate an asynchronous I/O operation that you initiated by means of the FEXCIO macro, you must call the FCEKIO or FWFTIO macro. You use the IOID operand of these macros to reference the asynchronous I/O operation you initiated.

Information on the status or result of the I/O operation is stored in the device information block (DIB).

### Macro call format

| Macro name | Operands |
|---|---|
| FEXCIO | MF = C / D / E / L / M<br>,DEVID = <var: int:4> / (<reg: int:4>)<br>,IOPRG = <var: pointer> / (<reg: pointer>)<br>,CPID = <var: int:4> / (<reg: int:4>)<br>,IOTERM = *DEVEND / *CHNEND<br>,EVNTMSG = *NO / *UEVNT<br>,EIID = <var: int:4> / (<reg: int:4>)<br>,POSTBYT = <c-string 1..1> / <var: char:1> / (<reg: char:1>)<br>,IOIDRET = <var: pointer> / (<reg: pointer>) |

*Illegal operands*

- The IOPRG and CPID operands must not be specified together.
- You may only supply the EIID and POSTBYT operands with values if you specify EVNTMSG=*UEVNT.

### Operand description

**MF = C / D / E / L / M**
See page 19 for the meanings of the operand values.

**DEVID =**
Indicates the device for which the asynchronous I/O operation is to be initiated. The device ID is returned when you open device control by means of the FOPNDV macro.

**<var: int:4>**
Specification of a symbolic field name; the field name contains the device ID as an integer (4 bytes).

**(<reg: int:4>)**
Specification of a register; the register contains the device ID as an integer (4 bytes).

**IOPRG =**
Indicates the address of a specific device job or the address of the first job in the device job chain. The I/O job specified here refers to the device job (or device job chain) defined by means of the FDEFIO macro.

ADAM automatically generates the channel program for this device job when the I/O operation is initiated. The channel program is automatically deleted when the I/O operation is terminated.

**<var: pointer>**
There are two ways of specifying the address of the device job:

– Specification of a symbolic field name: the field contains the address.
This specification is only possible in conjunction with MF=M. In addition, an operand list must be generated with MF=C or MF=D.
– Direct specification in the form A(field).

**(<reg: pointer>)**
Specification of a register containing the address of the device job.

**CPID =**
Indicates the channel program generated by the I/O operation (FDECCP macro).

**<var: int:4>**
Specification of a symbolic field name; the field name contains the channel program ID as an integer (4 bytes).

**(<reg: int:4>)**
Specification of a register; the register contains the channel program ID as an integer (4 bytes).

**IOTERM =**
Determines termination behavior when the I/O operation is completed.

***DEVEND**
Control is returned to you when the end-of-message signal arrives from the device. This is the default value.

***CHNEND**
Control is returned to you when the end-of-message signal arrives from the controller.

**EVNTMSG =**
Indicates the level of information provided for the "end of I/O" event. You can use this operand to define whether you wish to be informed of the arrival of the "end of I/O" event. If you wish to receive information on the end of the I/O operation, you must terminate the initiated asynchronous I/O operation by means of the FCEKIO or FWFTIO macro.

***NO**
You are not informed of the end of the I/O operation. In this case, you do not need to execute either the FCEKIO or FWFTIO macro. This is the default value.

***UEVNT**
User eventing informs you of the end of the I/O operation. The information byte (see the POSTBYT operand) indicates whether the I/O operation has been completed or whether you still have to wait for this event.

**EIID =**
Indicates the event item ID returned after execution of the ENAEI macro.

**<var: int:4>**
Specification of a symbolic field name; the field contains the event item ID as an integer (4 bytes).

**(<reg: int:4>)**
Specification of a register; the register contains the event item ID as an integer (4 bytes).

**POSTBYT =** *(this operand is optional)*
Indicates the first byte of the post code in which a short message regarding the "end of I/O" event is stored (information byte).
When you call the BS2000 macro SOLSIG, you address the post code field.
You can use the short message to identify the I/O operation that has been terminated.

**<c-string 1..1>**
Direct specification of the information byte as a string (1 byte).

**<var: char:1>**
Specification of a symbolic field name; the field name contains the information byte as a string (1 byte).

**(<reg: char: 1>)**
Specification of a register; the register contains the information byte as a string (1 byte).

**IOIDRET =**
Indicates the address of the field in which the identifier of the I/O operation that has been initiated is stored. The field must be aligned on a word boundary.

**<var: pointer>**
There are two ways of specifying the address of the field in which the I/O identifier is stored:

- Specification of a symbolic field name: the field contains the address.
  This specification is only possible in conjunction with MF=M. In addition, an operand list must be generated with MF=C or MF=D.
- Direct specification in the form A(field).

**(<reg: pointer>)**
Specification of a register containing the address of the field.

### Return codes

| SC2 | SC1 | Maincode | Meaning |
|---|---|---|---|
| 00 | 00 | 0000 | No error; I/O initiated. |
| 00 | 01 | 0404 | Invalid device job address (IOPRG operand). |
| 00 | 01 | 0408 | Invalid device ID (DEVID operand). |
| 00 | 01 | 040C | Invalid address for the field containing the I/O identifier (IOIDRET operand). |
| 00 | 01 | 0410 | Invalid channel program ID (CPID operand). |
| 00 | 01 | 0414 | Invalid value for the level of information for the user (EVNTMSG operand). |
| 00 | 01 | 0420 | Invalid address for the I/O buffer (DATA operand). |
| 00 | 01 | 0424 | Invalid number of bytes to be transferred (LENGTH operand). |
| 00 | 01 | 0428 | Invalid instruction code (CMD operand). |
| 00 | 01 | 042C | Invalid device job option (FLAG operand). |
| 00 | 01 | 0430 | Device job chain is too long. |
| 00 | 01 | 0434 | Invalid continuation address in device chain (CHAIN operand). |
| 00 | 01 | 0438 | Device job option DC not permitted for this device. |
| 00 | 01 | 043C | Device job option DC not permitted for this operation. |
| 00 | 01 | 0444 | Device job option DC incompatible with continuation address. |
| 00 | 01 | 0448 | Invalid event item ID (EIID operand) |
| xx | 20 | 0C00 | System error:<br>The subcode 2 (SC2) contains a system error code intrinsic to ADAM. |
| 00 | 40 | 0834 | Maximum number of parallel I/O operations reached. |
| 00 | 40 | 0848 | Resources temporarily unavailable, wait and try again. |
| 00 | 40 | 000C | Referenced channel program still in use. |

## Assembler DSECT

```
         FEXCIO MF=D
         MFTST MF=D,PREFIX=N,MACID=DA4,ALIGN=F,                         C
               DMACID=DA4,SUPPORT=(E,D,C,M,L),DNAME=DA4XCIO
NDA4XCIO DSECT ,
               *,##### PREFIX=N, MACID=DA4 ####
*   **** ===== FEXCIO PARAMETER AREA =====
NDA4HDR  FHDR  MF=(C,NDA4),EQUATES=NO        --- FHDR -----------------
NDA4HDR  DS    0A
NDA4FHE  DS    0XL8            0   GENERAL PARAMETER AREA HEADER
*
NDA4IFID DS    0A              0   INTERFACE IDENTIFIER
NDA4FCTU DS    AL2             0   FUNCTION UNIT NUMBER
*                                  BIT 15    HEADER FLAG BIT,
*                                  MUST BE RESET UNTIL FURTHER NOTICE
*                                  BIT 14-12 UNUSED, MUST BE RESET
*                                  BIT 11-0  REAL FUNCTION UNIT NUMBER
NDA4FCT  DS    AL1             2   FUNCTION NUMBER
NDA4FCTV DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
*
NDA4RET  DS    0A              4   GENERAL RETURN CODE
NDA4SRET DS    0AL2            4   SUB RETURN CODE
NDA4SR2  DS    AL1             4   SUB RETURN CODE 2
NDA4SR1  DS    AL1             5   SUB RETURN CODE 1
NDA4MRET DS    0AL2            6   MAIN RETURN CODE
NDA4MR2  DS    AL1             6   MAIN RETURN CODE 2
NDA4MR1  DS    AL1             7   MAIN RETURN CODE 1
NDA4FHL  EQU   8               8   GENERAL OPERAND LIST HEADER LENGTH
*
*   main return codes
NDA4NOE  EQU   0                         No error
NDA4IDJ  EQU   1028                      Invalid address of Device Job
*                                        (IOPRG)
NDA4IDI  EQU   1032                      Invalid value of DEVID
NDA4IIO  EQU   1036                      Invalid address IOIDRET
NDA4ICI  EQU   1040                      Invalid value of CPID
NDA4IEV  EQU   1044                      Invalid value of EVNTMSG
NDA4IIB  EQU   1056                      Invalid address of IO buffer
*                                        (DATA)
NDA4IDL  EQU   1060                      Invalid data length (LENGTH)
NDA4IOP  EQU   1064                      Invalid ADAM Op-Code (CMD)
NDA4IJO  EQU   1068                      Invalid Device Job option
*                                        (FLAG)
NDA4CTL  EQU   1072                      Device job chain too long
NDA4ICA  EQU   1076                      Invalid continuation address
*                                        in Device Job (CHAIN)
NDA4IDD  EQU   1080                      Invalid option DC, not allowed
*                                        for this device
NDA4IDO  EQU   1084                      Invalid option DC, not allowed
*                                        for this operation
NDA4IDC  EQU   1092                      Invalid option DC,
*                                        incompatible with
*                                        continuation address
NDA4IEI  EQU   1096                      Invalid Event Item ID
NDA4SYE  EQU   3072                      System error
NDA4MIR  EQU   2100                      Maximum number of parallel
*                                        IO's reached
NDA4RBN  EQU   2120                      System resource temporarily
*                                        not available, wait and retry
NDA4CIU  EQU   12                        Referenced channel program
*                                        still in use
*
*
```

```
NDA4INP  DS    0XL24                     --- Input Parameters -----
NDA4DEVI DS    F                         Device-ID
NDA4FLG1 DS    AL1                       Flag-byte 1
NDA4CPS  EQU   X'80'                     : CPRG specified
NDA4CHE  EQU   X'40'                     : IOTERM=*CHNEND
NDA4EIS  EQU   X'20'                     : Event information specified
NDA4RS1  EQU   X'1F'                     : - reserved -
NDA4FLG2 DS    AL1                       Flag-byte 2
NDA4EVB  EQU   X'80'                     : EVNTMSG=*UEVNT
NDA4EVF  EQU   X'40'                     : EVNTMSG=*FITC
NDA4RS2  EQU   X'3F'                     : - reserved -
NDA4FLG3 DS    XL1                       *** reserved ***
NDA4FLG4 DS    XL1                       *** reserved ***
NDA4DJOB DS    0XL4                      Device job specification
NDA4DJB@ DS    A                         - Address of device job
         ORG   NDA4DJOB
NDA4CPID DS    F                         - Channel program ID
         ORG   NDA4DJOB+4
NDA4EVID DS    0XL4                      Event ID
NDA4EIID DS    F                         - Event Item ID
         ORG   NDA4EVID
NDA4POID DS    H                         - FITC Port ID
         ORG   NDA4EVID+4
NDA4EVIN DS    0XL4                      Eventing information
NDA4MSG  DS    CL4                       - short message
         ORG   NDA4EVIN
NDA4POC  DS    CL1                       - post code
         ORG   NDA4EVIN+4
NDA4IOI@ DS    A                         Pointer to return-field for
*                                        IO-id
*
NDA4#    EQU   *-NDA4HDR
```

# FEXIOW
# Initiate synchronous I/O

Macro type: S (action macro)

## Macro description

This macro initiates an I/O operation and waits until it has terminated. Control is returned to you once termination processing has been completed.

Information on the result of the synchronous I/O operation is stored in the device information block (DIB). You must initialize the DIB by means of the FDEVIB macro before you call the FEXIOW macro. The DIB address is transferred in the DIB operand of the FEXIOW macro.

## Macro call format

| Macro name | Operands |
|---|---|
| `FEXIOW` | `MF = C / D / E / L / M`<br>`,DEVID = <var: int:4> / (<reg: int:4>)`<br>`,IOPRG = <var: pointer> / (<reg: pointer>)`<br>`,CPID = <var: int:4> / (<reg: int:4>)`<br>`,IOTERM = *DEVEND / *CHNEND`<br>`,DIB = <var: pointer> / (<reg: pointer>)` |

*Illegal operands*

The IOPRG and CPID operands must not be specified together.

## Operand description

**MF = C / D / E /  L / M**
See page 19 for the meanings of the operand values.

**DEVID =**
Indicates the device for which the synchronous I/O operation is to be initiated. The device ID is returned when you open device control by means of the FOPNDV macro.

**<var: int:4>**
Specification of a symbolic field name; the field name contains the device ID as an integer (4 bytes).

**(<reg: int:4>)**
Specification of a register; the register contains the device ID as an integer (4 bytes).

**IOPRG =**
Indicates the address of a specific device job or the address of the first job in the device job chain. The I/O operation specified here refers to the device job (or device job chain) defined by means of the FDEFIO macro.
ADAM automatically generates the channel program for this device job when the I/O operation is initiated. The channel program is automatically deleted when the I/O operation is terminated.

**<var: pointer>**
There are two ways of specifying the address of the device job:

– Specification of a symbolic field name: the field contains the address. This specification is only possible in conjunction with MF=M. In addition, an operand list must be generated with MF=C or MF=D.
– Direct specification in the form A(field).

**(<reg: pointer>)**
Specification of a register containing the address of the device job.

**CPID =**
Indicates the channel program generated from the I/O operation (FDECCP macro).

**<var: int:4>**
Specification of a symbolic field name; the field name contains the channel program ID as an integer (4 bytes).

**(<reg: int:4>)**
Specification of a register; the register contains the channel program ID as an integer (4 bytes).

**IOTERM =**
Determines termination behavior when the I/O operation is completed.

**<u>*DEVEND</u>**
Control is returned to you when the end-of-messages signal arrives from the device. This is the default value.

***CHNEND**
Control is returned to you when the end-of-message signal arrives from the controller.

**DIB =**
Indicates the device information block (DIB). Here you must specify the address of the DIB that you generated by means of the FDEVIB macro.
If the I/O operation is terminated with an error, the data is stored in the DIB.

**<var: pointer>**
There are two ways of specifying the address of the DIB:

- Specification of a symbolic field name: the field contains the address.
  This specification is only possible in conjunction with MF=M. In addition, an operand list must be generated with MF=C or MF=D.
- Direct specification in the form A(field).

**(<reg: pointer>)**
Specification of a register containing the address of the DIB.

### Return codes

| SC2 | SC1 | Maincode | Meaning |
|---|---|---|---|
| 00 | 00 | 0000 | No error; I/O terminated without error. |
| 00 | 01 | 0404 | Invalid device job address (IOPRG operand). |
| 00 | 01 | 0408 | Invalid device ID (DEVID operand). |
| 00 | 01 | 0410 | Invalid channel program ID (CPID operand). |
| 00 | 01 | 0418 | Invalid DIB address (DIB operand). |
| 00 | 01 | 0420 | Invalid address for the I/O buffer (DATA operand). |
| 00 | 01 | 0424 | Invalid number of bytes to be transferred (LENGTH operand). |
| 00 | 01 | 0428 | Invalid instruction code (CMD operand). |
| 00 | 01 | 042C | Invalid device job option (FLAG operand). |
| 00 | 01 | 0430 | Device job chain is too long. |
| 00 | 01 | 0434 | Invalid continuation address in device chain (CHAIN operand). |
| 00 | 01 | 0438 | Device job option DC not permitted for this device. |
| 00 | 01 | 043C | Device job option DC not permitted for this operation. |
| 00 | 01 | 0444 | Device job option DC incompatible with continuation address. |
| xx | 20 | 0C00 | System error:<br>The subcode 2 (SC2) contains a system error code intrinsic to ADAM. |
| 00 | 40 | 0834 | Maximum number of parallel I/O operations reached. |
| 00 | 40 | 000C | Referenced channel program still in use. |
| 00 | 40 | 0014 | I/O terminated with error. |

### Assembler DSECT

```
         FEXIOW MF=D
         MFTST MF=D,PREFIX=N,MACID=DA3,ALIGN=F,                        C
               DMACID=DA3,SUPPORT=(E,D,C,M,L),DNAME=DA3XIOW
NDA3XIOW DSECT ,
               *,##### PREFIX=N, MACID=DA3 ####
*   **** ===== FEXIOW PARAMETER AREA =====
NDA3HDR  FHDR  MF=(C,NDA3),EQUATES=NO        --- FHDR -----------------
NDA3HDR  DS    0A
NDA3FHE  DS    0XL8            0   GENERAL PARAMETER AREA HEADER
*
NDA3IFID DS    0A              0   INTERFACE IDENTIFIER
NDA3FCTU DS    AL2             0   FUNCTION UNIT NUMBER
*                                  BIT 15    HEADER FLAG BIT,
*                                  MUST BE RESET UNTIL FURTHER NOTICE
*                                  BIT 14-12 UNUSED, MUST BE RESET
*                                  BIT 11-0  REAL FUNCTION UNIT NUMBER
NDA3FCT  DS    AL1             2   FUNCTION NUMBER
NDA3FCTV DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
*
NDA3RET  DS    0A              4   GENERAL RETURN CODE
NDA3SRET DS    0AL2            4   SUB RETURN CODE
NDA3SR2  DS    AL1             4   SUB RETURN CODE 2
NDA3SR1  DS    AL1             5   SUB RETURN CODE 1
NDA3MRET DS    0AL2            6   MAIN RETURN CODE
NDA3MR2  DS    AL1             6   MAIN RETURN CODE 2
NDA3MR1  DS    AL1             7   MAIN RETURN CODE 1
NDA3FHL  EQU   8               8   GENERAL OPERAND LIST HEADER LENGTH
*
*   main return codes
NDA3NOE  EQU   0                         No error
NDA3IDJ  EQU   1028                      Invalid address of Device Job
*                                        (IOPRG)
NDA3IDI  EQU   1032                      Invalid value of DEVID
NDA3ICI  EQU   1040                      Invalid value of CPID
NDA3IDB  EQU   1048                      Invalid address of DIB
NDA3IIB  EQU   1056                      Invalid address of IO buffer
*                                        (DATA)
NDA3IDL  EQU   1060                      Invalid data length (LENGTH)
NDA3IOP  EQU   1064                      Invalid ADAM Op-Code (CMD)
NDA3IJO  EQU   1068                      Invalid Device Job option
*                                        (FLAG)
NDA3CTL  EQU   1072                      Device Job chain too long
NDA3ICA  EQU   1076                      Invalid continuation address
*                                        in Device Job (CHAIN)
NDA3IDD  EQU   1080                      Invalid option DC, not allowed
*                                        for this device
NDA3IDO  EQU   1084                      Invalid option DC, not allowed
*                                        for this operation
NDA3IDC  EQU   1092                      Invalid option DC,
*                                        incompatible with
*                                        continuation address
NDA3SYE  EQU   3072                      System error
NDA3MIR  EQU   2100                      Maximum number of parallel
*                                        IO's reached
NDA3CIU  EQU   12                        Referenced channel program
*                                        still in use
NDA3IOE  EQU   20                        IO terminated with error
*
*
NDA3INP  DS    0XL24                     --- Input Parameters -----
NDA3DEVI DS    F                         Device-ID
NDA3FLG1 DS    AL1                       Flag-byte 1
```

```
NDA3CPS  EQU   X'80'                      - CPRG specified
NDA3CHE  EQU   X'40'                      - IOTERM=*CHNEND
NDA3RS1  EQU   X'3F'                      - reserved -
NDA3FLG2 DS    XL1                        *** reserved ***
NDA3FLG3 DS    XL1                        *** reserved ***
NDA3FLG4 DS    XL1                        *** reserved ***
NDA3DJOB DS    0XL4                       Device Job specification
NDA3DJB@ DS    A                          - Address of Device Job
         ORG   NDA3DJOB
NDA3CPID DS    F                          - Channel program ID
         ORG   NDA3DJOB+4
NDA3DIB@ DS    A                          Address of DIB
NDA3CIB@ DS    A                          Address of CIB
NDA3RSV1 DS    XL4                        *** reserved ***
*
NDA3#    EQU   *-NDA3HDR
```

# FHLTIO
# Force termination of asynchronous I/O operation

Macro type: S (action macro)

## Macro description

This macro allows you to force the termination of all I/O operations of an active device which had not yet been terminated when the macro was issued. The I/O operations are aborted. The reaction to the aborted I/O operations depends on the device involved.
The execution status of an I/O operation terminated in this manner is not defined.
An I/O operation must be terminated by means of the FCEKIO or FWFTIO macro even after execution of the FHLTIO macro. An indicator is set in I/O status byte 2 in the device information block (DIB), which is referenced by the FCEKIO and FWFTIO macros. Further device-specific error flags are likewise stored in the DIB.

The system forces termination of the I/O operation at the latest after expiration of the timeout period (see the FTIMOT macro).

## Macro call format

| Macro name | Operands |
|---|---|
| FHLTIO | MF = C / D / E / L / M<br>,DEVID = <var: int:4> / (<reg: int:4>) |

## Operand description

**MF = C / D / E / L / M**
See page 19 for the meanings of the operand values.

**DEVID =**
Indicates the device whose asynchronous I/O operations are to be forcibly terminated. The device ID is returned when you open device control by means of the FOPNDV macro.

**<var: int:4>**
Specification of a symbolic field name; the field name contains the device ID as an integer (4 bytes).

**(<reg: int:4>)**
Specification of a register; the register contains the device ID as an integer (4 bytes).

### Return codes

| SC2 | SC1 | Maincode | Meaning |
|---|---|---|---|
| 00 | 00 | 0000 | No error; I/O terminated. |
| 01 | 00 | 0000 | No action.<br>All I/O operations have already been terminated, but ADAM has not yet performed termination processing. |
| 00 | 01 | 0408 | Invalid device ID (DEVID operand) |
| xx | 20 | 0C00 | System error:<br>The subcode 2 (SC2) contains a system error code intrinsic to ADAM. |
| 00 | 40 | 0820 | No asynchronous I/O operation initiated. |

### Assembler DSECT

```
         FHLTIO MF=D
         MFTST MF=D,PREFIX=N,MACID=DA7,ALIGN=F,                        C
               DMACID=DA7,SUPPORT=(E,D,C,M,L),DNAME=DA7HLIO
NDA7HLIO DSECT ,
               *,##### PREFIX=N, MACID=DA7 ####
*   **** ===== FHLTIO PARAMETER AREA =====
NDA7HDR  FHDR  MF=(C,NDA7),EQUATES=NO        —— FHDR —————————————————
NDA7HDR  DS    0A
NDA7FHE  DS    0XL8            0   GENERAL PARAMETER AREA HEADER
NDA7IFID DS    0A              0   INTERFACE IDENTIFIER
NDA7FCTU DS    AL2             0   FUNCTION UNIT NUMBER
*                                  BIT 15    HEADER FLAG BIT,
*                                  MUST BE RESET UNTIL FURTHER NOTICE
*                                  BIT 14-12 UNUSED, MUST BE RESET
*                                  BIT 11-0  REAL FUNCTION UNIT NUMBER
NDA7FCT  DS    AL1             2   FUNCTION NUMBER
NDA7FCTV DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
*
NDA7RET  DS    0A              4   GENERAL RETURN CODE
NDA7SRET DS    0AL2            4   SUB RETURN CODE
NDA7SR2  DS    AL1             4   SUB RETURN CODE 2
NDA7SR1  DS    AL1             5   SUB RETURN CODE 1
NDA7MRET DS    0AL2            6   MAIN RETURN CODE
NDA7MR2  DS    AL1             6   MAIN RETURN CODE 2
NDA7MR1  DS    AL1             7   MAIN RETURN CODE 1
NDA7FHL  EQU   8               8   GENERAL OPERAND LIST HEADER LENGTH
*
*   main return codes
NDA7NOE  EQU   0                         No error
NDA7IDI  EQU   1032                      Invalid value of DEVID
NDA7SYE  EQU   3072                      System error
NDA7CIU  EQU   20                        All IOs already terminated but
*                                        not completed by ADAM
NDA7NII  EQU   2080                      No asyncronous IO initiated
*
NDA7INP  DS    0XL8                      —— Input Parameters —————
NDA7DEVI DS    F                         Device-ID
NDA7FLG1 DS    XL1                       *** reserved ***
NDA7FLG2 DS    XL1                       *** reserved ***
NDA7FLG3 DS    XL1                       *** reserved ***
NDA7FLG4 DS    XL1                       *** reserved ***
*
NDA7#    EQU   *-NDA7HDR
```

# FOPNDV
# Open device control

Macro type: S (action macro)

### Macro description

This macro initiates device control and gives you control over the specified device. Your user task has access to the desired device until the FCLSDV macro is executed or until the program or task is terminated.

### Macro call format

| Macro name | Operands |
|---|---|
| `FOPNDV` | ` MF = C / D / E / L / M`<br>`,DEVIDRE = <var: pointer> / (<reg: pointer>)`<br>`,SYMBTYP = <c-string 1..8> / <var: char:8> / (<reg: char:8>)`<br>`,MNEMON = <c-string 2..4> / <var: char:4> / (<reg: char:4>)` |

*Illegal operands*

The SYMBTYP and MNEMON operands must not be specified together.

### Operand description

**MF = C / D / E / L / M**
See page 19 for the meanings of the operand values.

**DEVIDRE =**
Indicates the name of the field in which the device ID is stored.

**<var: pointer>**
There are two ways of specifying the address of the field:

- Specification of a symbolic field name: the field contains the address.
  This specification is only possible in conjunction with MF=M. In addition, an operand list must be generated with MF=C or MF=D.
- Direct specification in the form A(field).

**(<reg: pointer>)**
Specification of a register containing the address of the field.

**SYMBTYP =**
Indicates the symbolic device type. You are granted access to all available devices of the specified type.

**<c-string 1..8>**
Direct specification of the symbolic device type as a string (8 bytes).

**<var: char:8>**
Specification of a symbolic field name; the field name contains the symbolic device type as a string (8 bytes).

**(<reg: char:8>)**
Specification of a register; the register contains the symbolic device type as a string (8 bytes).

**MNEMON =**
Indicates the mnemonic device name that you can use to reference the device. You are granted access to the device with the specified name, provided it is available.

**<c-string 2..4>**
Direct specification of the mnemonic device name as a string (minimum length: 2 bytes, maximum length: 4 bytes).

**<var: char:4>**
Specification of a symbolic field name; the field name contains the mnemonic device name as a string (4 bytes).

**(<reg: char:4>)**
Specification of a register; the register contains the mnemonic device name as a string (4 bytes).

### Return codes

| SC2 | SC1 | Maincode | Meaning |
|---|---|---|---|
| 00 | 00 | 0000 | No error; device control opened. |
| 00 | 01 | 0408 | Invalid device ID (DEVIDRE operand). |
| 00 | 01 | 0458 | The operands MNEMON and SYMBTYP are missing or have both been specified. |
| 00 | 01 | 0464 | Invalid mnemonic device name (MNEMON operand). |
| 00 | 01 | 0468 | Invalid symbolic device type (SYMBTYP operand). |
| xx | 20 | 0C00 | System error:<br>The subcode 2 (SC2) contains a system error code intrinsic to ADAM. |
| 00 | 40 | 0850 | Device not accessible: device is reserved for a different task, is permanently disabled or has not been generated. |
| 00 | 40 | 0854 | Device module not accessible: invalid version or channel type. |
| 00 | 40 | 0858 | Device access not permitted: privilege missing. |

## Assembler DSECT

```
         FOPNDV MF=D
         MFTST MF=D,PREFIX=N,MACID=DA1,ALIGN=F,                        C
               DMACID=DA1,SUPPORT=(E,D,C,M,L),DNAME=DA1OPDV
NDA1OPDV DSECT ,
               *,##### PREFIX=N, MACID=DA1 ####
*   **** ===== FOPNDV PARAMETER AREA =====
NDA1FHDR FHDR  MF=(C,NDA1),EQUATES=NO        --- FHDR -----------------
NDA1FHDR DS    0A
NDA1FHE  DS    0XL8            0   GENERAL PARAMETER AREA HEADER
*
NDA1IFID DS    0A              0   INTERFACE IDENTIFIER
NDA1FCTU DS    AL2             0   FUNCTION UNIT NUMBER
*                                  BIT 15    HEADER FLAG BIT,
*                                  MUST BE RESET UNTIL FURTHER NOTICE
*                                  BIT 14-12 UNUSED, MUST BE RESET
*                                  BIT 11-0  REAL FUNCTION UNIT NUMBER
NDA1FCT  DS    AL1             2   FUNCTION NUMBER
NDA1FCTV DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
*
NDA1RET  DS    0A              4   GENERAL RETURN CODE
NDA1SRET DS    0AL2            4   SUB RETURN CODE
NDA1SR2  DS    AL1             4   SUB RETURN CODE 2
NDA1SR1  DS    AL1             5   SUB RETURN CODE 1
NDA1MRET DS    0AL2            6   MAIN RETURN CODE
NDA1MR2  DS    AL1             6   MAIN RETURN CODE 2
NDA1MR1  DS    AL1             7   MAIN RETURN CODE 1
NDA1FHL  EQU   8               8   GENERAL OPERAND LIST HEADER LENGTH
*
*   main return codes
NDA1NOE  EQU   0                         No error
NDA1IDA  EQU   1032                      Invalid value of DEVIDRE
NDA1MSE  EQU   1112                      MNEMON and SYMBTYP missing or
*                                        specified together
NDA1IMN  EQU   1124                      Invalid value of MNEMON
NDA1ISY  EQU   1128                      Invalid value of SYMBTYP
NDA1SYE  EQU   3072                      System error
NDA1DNA  EQU   2128                      Device not accessible:
*                                        reserved for another task,
*                                        permanently disabled or not
*                                        generated
NDA1DMI  EQU   2132                      ADAM device module not
*                                        accessible, invalid version
*                                        or channel type
NDA1PRV  EQU   2136                      Device access not permitted:
*                                        Privilege missing
*
*
NDA1INP  DS    0XL12                     --- Input Parameters -----
NDA1DID@ DS    A                         Address of return-field for
*                                        Device-ID
NDA1FLG1 DS    XL1                       *** reserved ***
NDA1FLG2 DS    XL1                       *** reserved ***
NDA1FLG3 DS    XL1                       *** reserved ***
NDA1FLG4 DS    XL1                       *** reserved ***
NDA1INF@ DS    A                         Address of Info bytes for
*                                        IOTRACE
*
*
NDA1INO  DS    0XL12                     --- Inout Parameters -----
NDA1SYMB DS    CL8                       Symbolic device type
NDA1MNEM DS    CL4                       Device mnemonic
*
NDA1#    EQU   *-NDA1FHDR
```

## FRECSG
## Request next device signal

Macro type: S (action macro)

### Macro description

You use this macro to direct ADAM to inform you of the arrival of the next device signal (attention interrupt). You receive the signal from the device specified in the DEVID operand. You can only use the FRECSG macro to transfer a single device signal. Any attempt to issue a FRECSG macro is rejected until the previous receive signal job has been terminated.
You must confirm the arrival of the device signal by executing the FCEKSG or FWFSG macro before you can terminate the receive signal job and receive the interrupt data returned with the device signal.

To support user eventing, you must call the BS2000 macro ENAEI and provide it with the name for the event item and the scope. ENAEI sets up this event item for your task and returns an identifier for the event item (4 bytes, EIIDRET operand). You can use this event item ID to establish a relationship between your task and user eventing in the course of further processing with the FRECSG macro (EIID operand).
You use the BS2000 macro SOLSIG to request a signal for your event item. The BS2000 macro POSSIG reports the arrival of an event to user eventing. The post code returned to user eventing by POSSIG informs you whether the event is relevant to you. If this is not the case, you must issue SOLSIG repeatedly until POSSIG reports an event that is relevant for your purposes.

Information on the occurrence of the "device signal received" event is stored in the first byte of the post code. You can request this information via the POSTBYT operand of the FRECSG macro.

### Macro call format

| Macro name | Operands |
|---|---|
| FRECSG | MF = C / D / E / L / M<br>,DEVID = <var: int:4> / (<reg: int:4>)<br>,EVNTMSG = *NO / *UEVNT<br>,EIID = <var: int:4> / (<reg: int:4>)<br>,POSTBYT = <c-string 1..1> / <var: char:1> / (<reg: char:1>) |

*Illegal operands*

You may only specify the EIID and POSTBYT operands in conjunction with EVNTMSG=*UEVNT.

**Operand description**

**MF = C / D / E / L / M**
See page 19 for the meanings of the operand values.

**DEVID =**
Indicates the device whose device signals you are requesting. The device ID is returned when you open device control by means of the FOPNDV macro.

**<var: int:4>**
Specification of a symbolic field name; the field name contains the device ID as an integer (4 bytes).

**(<reg: int:4>)**
Specification of a register; the register contains the device ID as an integer (4 bytes).

**EVNTMSG =**
Indicates the level of information provided on the "device signal received" event. You can use this operand to define whether you wish to be informed of the arrival of the "device signal received" event. You must terminate the receive signal job by means of the FCEKSG or FWFSG macro in order to receive interrupt data.

**<u>*NO</u>**
You are not informed of the arrival of a device signal. In this case, you do not need to execute either the FCEKSG or FWFSG macro. This is the default value.

***UEVNT**
User eventing informs you of the arrival of a device signal. The information byte (see the POSTBYT operand) indicates whether a device signal has arrived or whether you still have to wait for this event.

**EIID =**
Indicates the event item ID returned after execution of the ENAEI macro.

**<var: int:4>**
Specification of a symbolic field name; the field contains the event item ID as an integer (4 bytes).

**(<reg: int:4>)**
Specification of a register; the register contains the event item ID as an integer (4 bytes).

**POSTBYT =** *(this operand is optional)*
Indicates the first byte of the post code in which a short message regarding the "device signal received" event is stored (information byte).
When you call the BS2000 macro SOLSIG, you address the post code field.
You can use the short message to find out whether a device signal has arrived.
The information byte is also stored in the first byte of the sense byte. The sense byte is stored in the device information block when you call the FCEKSG macro.

**<c-string 1..1>**
Direct specification of the information byte as a string (1 byte).

**<var: char:1>**
Specification of a symbolic field name; the field name contains the information byte as a string (1 byte).

**(<reg: char: 1>)**
Specification of a register; the register contains the information byte as a string (1 byte).

### Return codes

| SC2 | SC1 | Maincode | Meaning |
|---|---|---|---|
| 00 | 00 | 0000 | No error; device signal received. |
| 00 | 01 | 0408 | Invalid device ID (DEVID operand). |
| 00 | 01 | 0414 | Invalid value for the level of information for the user (EVNTMSG operand). |
| 00 | 01 | 0448 | Invalid event item ID. |
| xx | 20 | 0C00 | System error:<br>Subcode 2 (SC2) contains a system error code intrinsic to ADAM. |
| 00 | 40 | 0848 | Resources temporarily unavailable, wait and try again. |
| 00 | 40 | 0830 | Preceding receive signal job not yet completed. |

### Assembler DSECT

```
         FRECSG MF=D
         MFTST MF=D,PREFIX=N,MACID=DAU,ALIGN=F,                        C
               DMACID=DAU,SUPPORT=(E,D,C,M,L),DNAME=DAURCSG
NDAURCSG DSECT ,
               *,##### PREFIX=N, MACID=DAU ####
*   **** ===== FRECSG PARAMETER AREA =====
NDAUHDR  FHDR  MF=(C,NDAU),EQUATES=NO        --- FHDR -----------------
NDAUHDR  DS    0A
NDAUFHE  DS    0XL8            0   GENERAL PARAMETER AREA HEADER
*
NDAUIFID DS    0A              0   INTERFACE IDENTIFIER
NDAUFCTU DS    AL2             0   FUNCTION UNIT NUMBER
*                                  BIT 15    HEADER FLAG BIT,
*                                  MUST BE RESET UNTIL FURTHER NOTICE
*                                  BIT 14-12 UNUSED, MUST BE RESET
*                                  BIT 11-0  REAL FUNCTION UNIT NUMBER
NDAUFCT  DS    AL1             2   FUNCTION NUMBER
```

```
NDAUFCTV DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
*
NDAURET  DS    0A              4   GENERAL RETURN CODE
NDAUSRET DS    0AL2            4   SUB RETURN CODE
NDAUSR2  DS    AL1             4   SUB RETURN CODE 2
NDAUSR1  DS    AL1             5   SUB RETURN CODE 1
NDAUMRET DS    0AL2            6   MAIN RETURN CODE
NDAUMR2  DS    AL1             6   MAIN RETURN CODE 2
NDAUMR1  DS    AL1             7   MAIN RETURN CODE 1
NDAUFHL  EQU   8               8   GENERAL OPERAND LIST HEADER LENGTH
*
*   main return codes
NDAUNOE  EQU   0                         No error
NDAUIDI  EQU   1032                      Invalid value of DEVID
NDAUIEV  EQU   1044                      Invalid value of EVNTMSG
NDAUIEI  EQU   1096                      Invalid Event Item ID
NDAUSYE  EQU   3072                      System error
NDAURBN  EQU   2120                      System resource temporarily
*                                        not available, wait and retry
NDAURNC  EQU   2096                      Preceding signal receive job
*                                        not yet completed
*
*
NDAUINP  DS    0XL24                     --- Input Parameters -----
NDAUDEVI DS    F                         Device-ID
NDAUFLG1 DS    AL1                       Flag-byte 1
NDAURS1  EQU   X'80'                     - reserved -
NDAURS2  EQU   X'40'                     - reserved -
NDAUEIS  EQU   X'20'                     - Event information specified
NDAURS3  EQU   X'1F'                     - reserved -
NDAUFLG2 DS    AL1                       Flag-byte 2
NDAUEVB  EQU   X'80'                     - EVNTMSG=*UEVNT
NDAUEVF  EQU   X'40'                     - EVNTMSG=*FITC
NDAURS4  EQU   X'3F'                     - reserved -
NDAUFLG3 DS    XL1                       *** reserved ***
NDAUFLG4 DS    XL1                       *** reserved ***
NDAUEVID DS    0XL4                      Event ID
NDAUEIID DS    F                         Event Item ID
         ORG   NDAUEVID
NDAUPOID DS    H                         FITC Port ID
         ORG   NDAUEVID+4
NDAUEVIN DS    0XL4                      Eventing information
NDAUMSG  DS    CL4                       - short message
         ORG   NDAUEVIN
NDAUPOC  DS    CL1                       - post code
         ORG   NDAUEVIN+4
NDAURSV1 DS    XL4                       *** reserved ***
NDAURSV2 DS    XL4                       *** reserved ***
*
NDAU#    EQU   *-NDAUHDR
```

# FTIMOT
# Define waiting time for I/O termination

Macro type: S (action macro)

## Macro description

This macro controls the timeout period for an I/O operation.

The timeout period determines the maximum length of time allowed from the start of an I/O operation to the arrival of the end-of-message signal from the device. After this time, the operating system forces termination of the I/O operation.
A default timeout value and a maximum timeout value are predefined for each device type in the appropriate device module. The standard timeout value is used when device control is opened (FOPNDV macro).

You can use the FTIMOT macro to vary the timeout period as desired, provided that you do not exceed the maximum value. The newly defined value remains valid for all subsequent I/O operations until the next time the FTIMOT macro is executed or until device control is closed.

The operating system’s I/O management system can increase the timeout period in keeping with I/O-specific characteristics.

## Macro call format

| Macro name | Operands |
|---|---|
| `FTIMOT` | ` MF = C / D / E / L / M`<br>`,DEVID = <var: int:4> / (<reg: int:4>)`<br>`,TIMEOUT = <integer 0..65534> / <var: int:4> / (<reg: int:4>) / *OFF` |

## Operand description

**MF = C / D / E / L / M**
See page 19 for the meanings of the operand values.

**DEVID =**
Indicates the device whose timeout period you wish to change. The device ID is returned when you open device control by means of the FOPNDV macro.

**<var: int:4>**
Specification of a symbolic field name; the field name contains the device ID as an integer (4 bytes).

**(<reg: int:4>)**
Specification of a register; the register contains the device ID as an integer (4 bytes).

**TIMEOUT =**
Indicates the timeout value. If the value specified exceeds the maximum value, the maximum value is used and transferred to the parameter area. If you specify zero, this value is replaced by the default timeout value stored in the device module.

**<integer 0..65534>**
Direct specification of the timeout value as an integer. The value is expressed in seconds.

**<var: int:4>**
Specification of a symbolic field name; the field contains the timeout value as an integer (4 bytes). The value is expressed in seconds.

**(<reg: int:4>)**
Specification of a register; the register contains the timeout value as an integer (4 bytes). The value is expressed in seconds.

***OFF**
The keyword *OFF indicates that timeout monitoring is to be disabled. Permission for this (the appropriate flag) must be entered in the device module of the specified device.

### Return codes

| SC2 | SC1 | Maincode | Meaning |
|---|---|---|---|
| 00 | 00 | 0000 | No error; timeout value modified. |
| 00 | 01 | 0408 | Invalid device ID (DEVID operand). |
| 00 | 01 | 0414 | Specification of TIMEOUT=*OFF not permitted for this device. |
| 00 | 20 | 0C00 | System error:<br>Subcode 2 (SC2) contains a system error code intrinsic to ADAM. |

## Assembler DSECT

```
         FTIMOT MF=D
         MFTST MF=D,PREFIX=N,MACID=DAT,ALIGN=F,                        C
               DMACID=DAT,SUPPORT=(E,D,C,M,L),DNAME=DATTMOT
NDATTMOT DSECT ,
               *,##### PREFIX=N, MACID=DAT ####
NDATTOFF EQU   -1                        Value for TIMEOUT = *OFF
*
*   **** ===== FTIMOT PARAMETER AREA =====
NDATHDR  FHDR  MF=(C,NDAT),EQUATES=NO        --- FHDR --------------------
NDATHDR  DS    0A
NDATFHE  DS    0XL8            0   GENERAL PARAMETER AREA HEADER
*
NDATIFID DS    0A              0   INTERFACE IDENTIFIER
NDATFCTU DS    AL2             0   FUNCTION UNIT NUMBER
*                                  BIT 15    HEADER FLAG BIT,
*                                  MUST BE RESET UNTIL FURTHER NOTICE
*                                  BIT 14-12 UNUSED, MUST BE RESET
*                                  BIT 11-0  REAL FUNCTION UNIT NUMBER
NDATFCT  DS    AL1             2   FUNCTION NUMBER
NDATFCTV DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
*
NDATRET  DS    0A              4   GENERAL RETURN CODE
NDATSRET DS    0AL2            4   SUB RETURN CODE
NDATSR2  DS    AL1             4   SUB RETURN CODE 2
NDATSR1  DS    AL1             5   SUB RETURN CODE 1
NDATMRET DS    0AL2            6   MAIN RETURN CODE
NDATMR2  DS    AL1             6   MAIN RETURN CODE 2
NDATMR1  DS    AL1             7   MAIN RETURN CODE 1
NDATFHL  EQU   8               8   GENERAL OPERAND LIST HEADER LENGTH
*
*   main return codes
NDATNOE  EQU   0                         No error
NDATIDI  EQU   1032                      Invalid value of DEVID
NDATITO  EQU   1044                      TIMEOUT = *OFF not allowed for
*                                        device
NDATSYE  EQU   3072                      System error
*
*
NDATINP  DS    0XL8                      --- Input Parameters -----
NDATDEVI DS    F                         Device-ID
NDATTVAL DS    F                         Timeout period
*
NDAT#    EQU   *-NDATHDR
```

# FWFSG
# Wait for next device signal

Macro type: S (action macro)

### Macro description

This macro enables you to wait for the arrival of the next device signal (attention interrupt). You must previously have used the macro to request ADAM to receive the device signal by means of the FRECSG macro.
If you specify the WTIME operand, you indicate the maximum length of time you are prepared to wait for the device signal. If a device signal arrives within the specified time period, termination processing for the receive signal job is initiated.
Control is returned to you if no device signal arrives before the waiting time expires. This is indicated by an appropriate return code. You terminate the receive signal job by means of the FWFSG or FCEKSG macro. Otherwise, ADAM will not accept any further receive signal jobs.

The interrupt data is stored in the device information block (DIB). You must therefore initialize the DIB by means of the FDEVIB macro before you issue the FWFSG macro. You reference the device information block in the DIB operand.

### Macro call format

| Macro name | Operands |
|---|---|
| `FWFSG` | ` MF = C / D / E / L / M`<br>`,DEVID = <var: int:4> / (<reg: int:4>)`<br>`,WTIME = <integer 0..21599> / <var: int:4> / (<reg: int:4>)`<br>`,DIB = <var: pointer> / (<reg: pointer>)` |

### Operand description

**MF = C / D / E / L / M**
See page 19 for the meanings of the operand values.

**DEVID =**
Indicates the device whose next signal you are waiting for. The device ID is returned when you open device control by means of the FOPNDV macro.

**<var: int:4>**
Specification of a symbolic field name; the field contains the device ID as an integer (4 bytes).

**(<reg: int:4>)**
Specification of a register; the register contains the device ID as an integer (4 bytes).

**WTIME =**
Indicates the maximum amount of time you are prepared to wait for the arrival of a device signal.
If the value specified exceeds 21599, the maximum permitted value is transferred to the parameter area. If you specify zero, it is replaced by the value 1.

**<integer 0..21599>**
Direct specification of the waiting time as an integer (in decimal form).
The value is expressed in seconds.

**<var: int:4>**
Specification of a symbolic field name; the field name contains the waiting time as an integer (4 bytes). The value is expressed in seconds.

**(<reg: int:4>)**
Specification of a register; the register contains the waiting time as an integer (4 bytes).

**DIB =**
Indicates the device information block (DIB). Here you must specify the address of the DIB that you generated by means of the FDEVIB macro. Interrupt data is stored in the DIB.

**<var: pointer>**
There are two ways of specifying the address of the DIB:

– Specification of a symbolic field name: the field contains the address.
This specification is only possible in conjunction with MF=M. In addition, an operand list must be generated with MF=C or MF=D.
– Direct specification in the form A(field).

**(<reg: pointer>)**
Specification of a register containing the address of the DIB.

### Return codes

| SC2 | SC1 | Maincode | Meaning |
|---|---|---|---|
| 00 | 00 | 0000 | No error; device signal arrived. |
| 00 | 01 | 0408 | Invalid device ID (DEVID operand). |
| 00 | 01 | 0418 | Invalid DIB address (DIB operand) |
| xx | 20 | 0C00 | System error:<br>Subcode 2 (SC2) contains a system error code intrinsic to ADAM |
| 00 | 40 | 0010 | Signal did not arrive within the specified time period. |
| 00 | 40 | 0820 | No receive signal job requested. |

## Assembler DSECT

```
         FWFSG MF=D
         MFTST MF=D,PREFIX=N,MACID=DAW,ALIGN=F,                        C
               DMACID=DAW,SUPPORT=(E,D,C,M,L),DNAME=DAWWFSG
NDAWWFSG DSECT ,
               *,##### PREFIX=N, MACID=DAW ####
*   **** ===== FWFSG PARAMETER AREA =====
NDAWHDR  FHDR  MF=(C,NDAW),EQUATES=NO        --- FHDR -----------------
NDAWHDR  DS    0A
NDAWFHE  DS    0XL8            0   GENERAL PARAMETER AREA HEADER
*
NDAWIFID DS    0A              0   INTERFACE IDENTIFIER
NDAWFCTU DS    AL2             0   FUNCTION UNIT NUMBER
*                                  BIT 15    HEADER FLAG BIT,
*                                  MUST BE RESET UNTIL FURTHER NOTICE
*                                  BIT 14-12 UNUSED, MUST BE RESET
*                                  BIT 11-0  REAL FUNCTION UNIT NUMBER
NDAWFCT  DS    AL1             2   FUNCTION NUMBER
NDAWFCTV DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
*
NDAWRET  DS    0A              4   GENERAL RETURN CODE
NDAWSRET DS    0AL2            4   SUB RETURN CODE
NDAWSR2  DS    AL1             4   SUB RETURN CODE 2
NDAWSR1  DS    AL1             5   SUB RETURN CODE 1
NDAWMRET DS    0AL2            6   MAIN RETURN CODE
NDAWMR2  DS    AL1             6   MAIN RETURN CODE 2
NDAWMR1  DS    AL1             7   MAIN RETURN CODE 1
NDAWFHL  EQU   8               8   GENERAL OPERAND LIST HEADER LENGTH
*
*   main return codes
NDAWNOE  EQU   0                         No error; Signal received
NDAWIDI  EQU   1032                      Invalid value of DEVID
NDAWIDB  EQU   1048                      Invalid address of DIB
NDAWSYE  EQU   3072                      System error
NDAWSNA  EQU   16                        Signal not arrived within
*                                        WTIME
NDAWNSR  EQU   2080                      No receive signal job
*                                        requested
NDAWWNA  EQU   2084                      Function WFSG not allowed with
*                                        EVNTMSG= *FITC
*
*
NDAWINP  DS    0XL24                     --- Input Parameters -----
NDAWDEVI DS    F                         Device-ID
NDAWFLG1 DS    XL1                       *** reserved ***
NDAWFLG2 DS    XL1                       *** reserved ***
NDAWFLG3 DS    XL1                       *** reserved ***
NDAWFLG4 DS    XL1                       *** reserved ***
NDAWWTIM DS    F                         Waiting time
NDAWDIB@ DS    A                         Address of DIB
NDAWRSV1 DS    XL4                       *** reserved ***
NDAWRSV2 DS    XL4                       *** reserved ***
*
NDAW#    EQU   *-NDAWHDR
```

## FWFTIO
## Wait for termination of asynchronous I/O operation

Macro type: S (action macro)

### Macro description

This macro enables you to wait for termination of an asynchronous I/O operation.
The task is placed in a wait state until the I/O operation is terminated. ADAM then performs termination processing.
Information on the result of the I/O operation is stored in the device information block (DIB). You must initialize the DIB by means of the FDEVIB macro before calling the FWFTIO macro. You reference the device information block using the DIB operand of the FWFTIO macro.

If the FWFTIO macro references an I/O operation that ADAM has already terminated, it will be rejected.

### Macro call format

| Macro name | Operands |
|---|---|
| `FWFTIO` | `MF = C / D / E / L / M`<br>`,DEVID = <var: int:4> / (<reg: int:4>)`<br>`,IOID = <var: int:4> / (<reg: int:4>)`<br>`,DIB = <var: pointer> / (<reg: pointer>)` |

### Operand description

**MF = C / D / E / L / M**
See page 19 for the meanings of the operand values.

**DEVID =**
Indicates the device on which the asynchronous I/O operation whose termination you are waiting for is running. The device ID is returned when you open device control by means of the FOPNDV macro.

**<var: int:4>**
Specification of a symbolic field name; the field contains the device ID as an integer (4 bytes).

**(<reg: int:4>)**
Specification of a register; the register contains the device ID as an integer (4 bytes).

**IOID =**
Indicates a specific I/O operation whose termination you are waiting for. The I/O identifier is returned if you started the asynchronous I/O operation by means of the FEXCIO macro. Once the I/O operation has been terminated, ADAM performs termination processing. The I/O operation has then been completed in the normal manner.

**<var: int:4>**
Specification of a symbolic field name; the field contains the I/O identifier as an integer (4 bytes).

**(<reg: int:4>)**
Specification of a register; the register contains the I/O identifier as an integer (4 bytes).

**DIB =**
Indicates the device information block (DIB). Here you must specify the address of the DIB that you generated by means of the FDEVIB macro. If the I/O operation is terminated with an error, supplementary information is stored in the DIB.

**<var: pointer>**
There are two ways of specifying the address of the DIB:

- Specification of a symbolic field name: the field contains the address. This specification is only possible in conjunction with MF=M. In addition, an operand list must be generated with MF=C or MF=D.
- Direct specification in the form A(field).

**(<reg: pointer>)**
Specification of a register containing the address of the DIB.

### Return codes

| SC2 | SC1 | Maincode | Meaning |
|---|---|---|---|
| 00 | 00 | 0000 | No error; I/O terminated without errors. |
| 00 | 01 | 0400 | Invalid I/O identifier (IOID operand). |
| 00 | 01 | 0408 | Invalid device ID (DEVID operand). |
| 00 | 01 | 0418 | Invalid DIB address (DIB operand). |
| xx | 20 | 0C00 | System error:<br>Subcode 2 (SC2) contains a system error code intrinsic to ADAM. |
| 00 | 40 | 0014 | I/O terminated with error. |
| 00 | 40 | 0018 | I/O terminated with the FHLTIO macro. |
| 00 | 40 | 0820 | I/O identifier references an I/O operation for which ADAM has already performed termination processing. |
| 00 | 40 | 084C | I/O operation could not be started; wait and try again. |

### Assembler DSECT

```
         FWFTIO MF=D
         MFTST MF=D,PREFIX=N,MACID=DA6,ALIGN=F,                        C
               DMACID=DA6,SUPPORT=(E,D,C,M,L),DNAME=DA6WTIO
NDA6WTIO DSECT ,
               *,##### PREFIX=N, MACID=DA6 ####
*   **** ===== FWFTIO PARAMETER AREA =====
NDA6HDR  FHDR  MF=(C,NDA6),EQUATES=NO       ---- FHDR -----------------
NDA6HDR  DS    0A
NDA6FHE  DS    0XL8            0   GENERAL PARAMETER AREA HEADER
*
NDA6IFID DS    0A              0   INTERFACE IDENTIFIER
NDA6FCTU DS    AL2             0   FUNCTION UNIT NUMBER
*                                  BIT 15    HEADER FLAG BIT,
*                                  MUST BE RESET UNTIL FURTHER NOTICE
*                                  BIT 14-12 UNUSED, MUST BE RESET
*                                  BIT 11-0  REAL FUNCTION UNIT NUMBER
NDA6FCT  DS    AL1             2   FUNCTION NUMBER
NDA6FCTV DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
*
NDA6RET  DS    0A              4   GENERAL RETURN CODE
NDA6SRET DS    0AL2            4   SUB RETURN CODE
NDA6SR2  DS    AL1             4   SUB RETURN CODE 2
NDA6SR1  DS    AL1             5   SUB RETURN CODE 1
NDA6MRET DS    0AL2            6   MAIN RETURN CODE
NDA6MR2  DS    AL1             6   MAIN RETURN CODE 2
NDA6MR1  DS    AL1             7   MAIN RETURN CODE 1
NDA6FHL  EQU   8               8   GENERAL OPERAND LIST HEADER LENGTH
*
*   main return codes
NDA6NOE  EQU   0                         No error
NDA6III  EQU   1024                      Invalid value of IOID
NDA6IDI  EQU   1032                      Invalid value of DEVID
NDA6IDB  EQU   1048                      Invalid address of DIB
NDA6SYE  EQU   3072                      System error
NDA6IOE  EQU   20                        IO terminated with error
NDA6IOH  EQU   24                        IO halted
NDA6INR  EQU   2080                      IO identifier refers not to an
*                                        incomplete IO
NDA6IWF  EQU   2084                      IO started with EVNTMSG= *FITC
*                                        - WFTIO call not allowed
NDA6INS  EQU   2124                      IO could not be started - wait
*                                        and retry
*
*
NDA6INP  DS    0XL24                     ---- Input Parameters ------
NDA6DEVI DS    F                         Device-ID
NDA6FLG1 DS    XL1                       *** reserved ***
NDA6FLG2 DS    XL1                       *** reserved ***
NDA6FLG3 DS    XL1                       *** reserved ***
NDA6FLG4 DS    XL1                       *** reserved ***
NDA6IOID DS    F                         IO identifier
NDA6DIB@ DS    A                         Address of DIB
NDA6CIB@ DS    A                         Address of CIB
NDA6RSV1 DS    XL4                       *** reserved ***
*
NDA6#    EQU   *-NDA6HDR
```

# 4 ADAM subsystem

ADAM is a subsystem that is managed by DSSM (Dynamic SubSystem Management). Its features are defined in a subsystem declaration file which forms part of the ADAM selectable unit.

## 4.1 ADAM selectable unit

The ADAM selectable unit comprises:

| **Component** | **Meaning** |
|---|---|
| SIPLIB.ADAM.<version> | System internal interface library<br>(necessary for the customer in special cases). |
| SKMLNK.ADAM.<version> | Module library for SQ servers |
| SPMLNK.ADAM.<version> | Module library for SX servers |
| SYSLIB.ADAM.<version> | User interface library (ASSEMBLER macros) |
| SYSLNK.ADAM.<version> | Module library for SX servers |
| SYSNRF.ADAM.<version> | NOREF file (Control file for REP processing) |
| SYSRME.ADAM.<version>.D/E | Readme file (german/english) |
| SYSRMS.ADAM.<version> | RMS depot |
| SYSSII.ADAM.<version> | Strukturinformation für IMON |
| SYSSSC.ADAM.<version> | Subsystem-Definition |

Table 6: ADAM components

## 4.2 Subsystem installation

In BS2000/OSD ADAM is installed with the IMON installation monitor.

## 4.3 Loading and unloading the subsystem

The ADAM subsystem is normally loaded by the first ADAM call and not unloaded again until system shutdown. It is, however, possible to load and unload ADAM explicitly. Unloading is possible only if no more tasks are running in the subsystem. To display all the tasks running on ADAM, you can request the status of the ADAM subsystem. It is not possible to control the availability of the ADAM subsystem by means of the HOLD-SUBSYSTEM and RESUME-SUBSYSTEM commands.

### Load subsystem

```
/START-SUBSYSTEM SUBSYSTEM-NAME=ADAM
```

### Unload subsystem

```
/STOP-SUBSYSTEM SUBSYSTEM-NAME=ADAM
```

### Display subsystem status

```
/SHOW-SUBSYSTEM-STATUS SUBSYSTEM-NAME=ADAM
```

# 5 System generation

All devices supported as standard by BS2000, i.e. devices for which a logical access method is provided, have a fixed device name and device type code (other than X‘71‘...X‘7F‘) within the system.

For non-standard devices, i.e. all devices which are only supported by ADAM, first an entry for the device type name must be generated in the system, and some device attributes may need to be defined. This is done during hardware generation by means of the IOGEN utility routine and during system startup by means of the startup parameter service.

ADAM can operate a device only if it has been generated and a device module exists for the device in the ADAM module library. This device module must have been created by the ADAM developers.

### Hardware generation by means of IOGEN

Each ADAM device on an S or SX server must be generated like any other using a DVC statement. In addition, an ADT statement and possibly an EVA statement is required for each ADAM device type. The IOGEN statements and the predefined ADAM device type names are described in the “System Installation” manual [1].

*ADT (Abstract Device Type) statement*

The ADT statement assigns exactly one device type name from the set X’71’...X’7F’ to one device type code. Consequently at most 15 device types are possible per generation.

The assignment between device type code and device type name must be unambiguous. A device type code may not be assigned two different device type names and vice versa.

For the ADAM devices which have already been released, the device attributes required by IOGEN are already defined in a table within IOGEN which is known as the basic table. For unbundled ADAM devices, this table must always be expanded using the EVA statement.

Only the ADAM device type names contained in the basic table (which may have been expanded) are permitted as device type names.

*EVA (Evaluate ADAM devices) statement*

The EVA statement must be used to define the device type name and the operating mode of the new devices in the case of unbundled ADAM devices. In the case of ADAM device types which have already been entered, it can be used to change the standard operating mode.

IOGEN does not check whether the connection to the controller or to the channel makes sense.

*Example of an IOGEN statement sequence:*

```
EVA NEWDEV,0,0,0,1
ADT (71,NEWDEV)
DVC P0,71,A,00,(00),AT=CHN
DVC P1,71,A,00,(01),AT=CHN
```

In this example a new ADAM device type with the device type name NEWDEV is defined, and the internal device type code X'71' is assigned to it. Two of these devices with the mnemonic device names P0 and P1 are connected to the system. They are available to the system (attached) at system startup. P0 is connected to the channel with the channel-path-id 00, P1 to the channel with the channel-path-id 01. Each device has the device address 0 at its port.

The following operating parameters are set:

ta=0: The signal sequences between the channel and the device are to be subjected to the channel’s timeout monitoring (Timeout active = yes).

cu=0 : The controller can control more than one input/output simultaneously.

cp =0 : The channel protocol type used is the interlock protocol (a response to a data request is required).

sv=1 : Status Verification Facility is enabled.

### ADAM parameter record in the startup parameter file

The information of the ADT statements (IOGEN) must also be defined in the parameter service.

To do this, the system administrator must either generate an ADAM parameter file with an ADAM parameter block or enter an ADAM parameter block in the standard parameter file (SYSPAR.BS2.<version>).

The keyword for defining the ADAM parameter block is ADAM.

Up to 16 ADAM parameter records can be specified in the ADAM parameter block.

An ADAM parameter record has the same format as the IOGEN statement ADT, except that only a single device (type,name) may be specified per ADT parameter record. The definition in the parameter record must match that of the IOGEN generation. No automatic check takes place.

Parameter records with incorrect syntax can be corrected interactively during startup.

The parameter service is described in detail in the “Introductory Guide to Systems Support”.

*Format of an ADAM parameter record:*

ADT (type,name)

type :	Device type code

name:	Symbolic name of the ADAM device

*Example 1:*

ADAM parameter block in the BS2000 parameter file*

```
********************************************************
*  PARAMETER SERVICE : ADAM                            *
********************************************************
*  MAXIMUM OF PARAMETER RECORDS:  16                   *
/BEGIN ADAM
ADT (71,NEWDEV)
/EOF
*
```

*Example 2:*

ADAM parameter block in a separate ADAM parameter file PARAMS.ADAM

```
/BS2000 PARAMS
/BEGIN ADAM
ADT (71,NEWDEV)
/EOF ADAM
/END-PARAMS
```

A reference to the ADAM parameter file must then be entered in the BS2000 parameter file SYSPAR.BS2.<version>:

```
*
********************************************************
*  PARAMETER SERVICE : ADAM                            *
********************************************************
/ADD SYSPAR.ADAM
*
```

# 6 Device modules

A device module contains the device attributes that you have declared in collaboration with the ADAM developers.

You specify the device instruction code (in decimal form; CMD operand of the FDEFIO macro) in your device job. This is generally a value taken directly from the device's instruction set.
However, instruction code conversion can also be agreed on. The user then enters a neutral ADAM instruction code which ADAM converts to the required instruction code from the device's instruction set. This enables the user always to specify the same ADAM instruction code for the same types of operation on different devices, even when the device instruction codes differ.

In the case of devices for which instruction code conversion has been defined, the device description includes an additional table describing the instruction and any special conditions pertaining to each valid ADAM instruction code.

ADAM jobs whose device instruction code is not supported by the device normally result in I/O errors.
The instruction code X'FF' (255 in decimal form) is not permitted for reasons intrinsic to ADAM.

The following attributes can be defined for a device:

| | | |
|---|---|---|
| Timeout monitoring | : | can be disabled / cannot be disabled |
| Maximum timeout value | : | in seconds |
| Standard timeout value | : | in seconds |
| Data chaining | : | permitted / not permitted |
| Device job loop | : | permitted / not permitted |
| Error mask | : | Hex-value, 3 byte<br>The bits of the mask correspond to the sense bytes 00 to 02 |
| Privilege | : | Privilege-ID / not required |

Example of an instruction code conversion table when instruction code conversion is required:

| ADAM instruction code (decimal) | Description | Data length | Device instruction code (hex) |
|---|---|---|---|
| 1 | Write | | 0A |
| 2 | Read | | 08 |
| 3 | Transmit check bytes | 5 | 1C |
| 4 | Dummy write instruction | 0 | 1A |
| 5 | Dummy read instruction | 0 | 18 |

Special conditions

Data length <> 0

If the data length specified for an operation is not equal to zero, a corresponding device job can only be placed with this data length (LENGTH operand of the FDEFIO macro).

Data length = 0

No data transfer to or from the user buffer. Any transfer data required is generated internally by ADAM.

# 7 Sample program

The sample program below illustrates how the different forms of the macros can be used to program an ADAM application.

```
*****************************************************************************
* ADAM Application example
*****************************************************************************

FADAM01 @ENTR  TYP=M,AMODE=ANY,RMODE=ANY
*
* —— Address variables ——————————————————————————————————————————————————————
        @DATA  CLASS=A,BASE=R9,DSECT=VAR
*
*
*————————————————————————————————————————————————————————————————————————————
* Open device control
*————————————————————————————————————————————————————————————————————————————
*
* —— Address parameter area —————————————————————————————————————————————————
        LA     R2,PFOPNDV
        @DATA  CLASS=B,BASE=R2,DSECT=FOPNDV
*
* —— Initialize parameter area ——————————————————————————————————————————————
        MVC    FOPNDV(NDA1#),LFOPNDV
*
* —— Modify parameter area ——————————————————————————————————————————————————
        LA     R3,DEVID            Address of the field for the device ID
        FOPNDV MF=M,DEVIDRE=(R3)
*
* —— Execute function ———————————————————————————————————————————————————————
        FOPNDV MF=E,PARAM=(R2)
*
* —— Check return code ——————————————————————————————————————————————————————
        @IF    NE
        CLI    NDA1SR1,NDA1NOE
        @THEN
        ...                        Error
        @BEND
```

```
*
*__________________________________________________________________________
* Generate and store channel program
*__________________________________________________________________________
*
*    ______________________________________________________________________
*      First device job
*    ______________________________________________________________________
*
* -- Address device job ---------------------------------------------------
        LA     R2,IOJOBWR
        @DATA  CLASS=B,BASE=R2,DSECT=FDEFIO
*
* -- Initialize device job ------------------------------------------------
        MVC    FDEFIO(NDAI#),IOPRGWR
*
* -- Modify device job ----------------------------------------------------
        LA     R3,DATAWR            Address of data buffer
        LA     R4,IOJOBRD           Address of next device job
        FDEFIO MF=M,DATA=(R3),CHAIN=(R4)
*
        MVC    DATAWR(4),=C'NFID'   Output data
*
*    ______________________________________________________________________
*      Next device job
*    ______________________________________________________________________
*
* -- Address device job ---------------------------------------------------
        LA     R2,IOJOBRD
        @DATA  CLASS=B,BASE=R2,DSECT=FDEFIO
*
* -- Initialize device job ------------------------------------------------
        MVC    FDEFIO(NDAI#),IOPRGRD
*
* -- Modify device job ----------------------------------------------------
        LA     R3,DATARD            Address of data buffer
        FDEFIO MF=M,DATA=(R3)
*
*    ______________________________________________________________________
*      Predefine channel program
*    ______________________________________________________________________
*
* -- Address parameter area
{--------------------------------------------------
        LA     R2,PFDECCP
        @DATA  CLASS=B,BASE=R2,DSECT=FDECCP
*
* -- Initialize parameter area --------------------------------------------
        MVC    FDECCP(NDA8#),LFDECCP
*
* -- Modify parameter area ------------------------------------------------
        LA     R3,IOJOBWR           Address of first device job
        LA     R4,CPID              Address of field for channel program ID
        FDECCP MF=M,DEVID=DEVID,IOPRG=(R3),CPIDRET=(R4)
*
* -- Execute function -----------------------------------------------------
        FDECCP MF=E,PARAM=(R2)
*
```

```
* —— Check return code ————————————————————————————————————————————————
        @IF    NE
        CLI    NDA8SR1,NDA8NOE
        @THEN
        ...                        Error
        @BEND
*_____________________________________________________________________
* Start asynchronous I/O
* (with predefined channel program)
*_____________________________________________________________________
*
* —— Address parameter area ————————————————————————————————————————————
        LA     R2,PFEXCIO
        @DATA  CLASS=B,BASE=R2,DSECT=FEXCIO
*
* —— Initialize parameter area —————————————————————————————————————————
        MVC    FEXCIO(NDA4#),LFEXCIO
*
* —— Modify parameter area —————————————————————————————————————————————
        LA     R3,IOID             Address of field for I/O identifier
        FEXCIO MF=M,DEVID=DEVID,CPID=CPID,IOIDRET=(R3)
*
* —— Execute function ——————————————————————————————————————————————————
        FEXCIO MF=E,PARAM=(R2)
*
* —— Check return code  ————————————————————————————————————————————————
        @IF    NE
        CLI    NDA4SR1,NDA4NOE
        @THEN
        ...                        Error
        @BEND
*
*_____________________________________________________________________
* Check I/O status
*_____________________________________________________________________
*
*    _________________________________________________________________
*     Prepare device information block (DIB)
*    _________________________________________________________________
        MVC    DIB(NDAJ#),LFDEVIB
*
*    _________________________________________________________________
*     Check status
*    _________________________________________________________________
*
* —— Address parameter area ————————————————————————————————————————————
        LA     R2,PFCEKIO
        @DATA  CLASS=B,BASE=R2,DSECT=FCEKIO
*
* —— Initialize parameter area ———————————————————————————————————————————————
        MVC    FCEKIO(NDA5#),LFCEKIO
*
* —— Modify parameter area ———————————————————————————————————————————————————
        LA     R3,DIB               Address of the DIB
        FCEKIO MF=M,DEVID=DEVID,IOID=IOID,DIB=(R3)
*

* —— Execute function ————————————————————————————————————————————————————————
        FCEKIO MF=E,PARAM=(R2)
*
* —— Check return code ———————————————————————————————————————————————————————
        @IF    EQ
        CLI    NDA5SR1,NDA5NOE      I/O terminated without error
        @THEN
        B      MFDELCP              Delete channel program
        @BEND
*
        @IF    EQ
        CLI    NDA5MR1,NDA5INT      I/O not yet terminated
        @THEN
        B      MFWFTIO              Wait for end of I/O operation
        @ELSE
        ...                         Error
        @BEND

*
*_____________________________________________________________________________
* Wait for end of I/O operation
*_____________________________________________________________________________
*
MFWFTIO EQU    *
*
* —— Address parameter area —————————————————————————————————————————————————
        LA     R2,PFWFTIO
        @DATA  CLASS=B,BASE=R2,DSECT=FWFTIO
*
* —— Initialize parameter area ———————————————————————————————————————————————
        MVC    FWFTIO(NDA6#),LFWFTIO
*
* —— Modify parameter area ———————————————————————————————————————————————————
        LA     R3,DIB               Address of the DIB
        FWFTIO MF=M,DEVID=DEVID,IOID=IOID,DIB=(R3)
*
* —— Execute function ————————————————————————————————————————————————————————
        FWFTIO MF=E,PARAM=(R2)
```

```
*
* —— Check return code ————————————————————————————————————————————————
        @IF    NE
        CLI    NDA6SR1,NDA6NOE
        @THEN
        ...                         Error
        @BEND
*_____________________________________________________________________
* Delete channel program
*_____________________________________________________________________
*
MFDELCP EQU    *
*
* —— Address parameter area ———————————————————————————————————————————
        LA     R2,PFDELCP
        @DATA  CLASS=B,BASE=R2,DSECT=FDELCP
*
* —— Initialize parameter area ————————————————————————————————————————
        MVC    FDELCP(NDA9#),LFDELCP
*
* —— Modify parameter area ————————————————————————————————————————————
        FDELCP MF=M,DEVID=DEVID
*
* —— Execute function —————————————————————————————————————————————————
        FDELCP MF=E,PARAM=(R2)
*
* —— Check return code ————————————————————————————————————————————————
        @IF    NE
        CLI    NDA9SR1,NDA9NOE
        @THEN
        ...                         Error
        @BEND
*
*_____________________________________________________________________
* Start synchronous I/O
* (channel program generated dynamically (implicitly) by ADAM)
*_____________________________________________________________________
*
*    _________________________________________________________________
*     Prepare device information block (DIB)
*    _________________________________________________________________
        MVC    DIB(NDAJ#),LFDEVIB
```

```
*    _________________________________________________________________________
*     Define device job
*    _________________________________________________________________________
*
* —— Address device job ——————————————————————————————————————————————————————
        LA     R2,IOJOBRD
        @DATA  CLASS=B,BASE=R2,DSECT=FDEFIO
*
* —— Initialize device job ———————————————————————————————————————————————————
        MVC    FDEFIO(NDAI#),IOPRGSEN
*
* —— Modify device job ———————————————————————————————————————————————————————
        LA     R3,DATARD              Address of data buffer
        FDEFIO MF=M,DATA=(R3)
*
*
*    _________________________________________________________________________
*     Start I/O operation
*    _________________________________________________________________________
*
* —— Address parameter area —————————————————————————————————————————————————
        LA     R2,PFEXIOW
        @DATA  CLASS=B,BASE=R2,DSECT=FEXIOW
*
* —— Initialize parameter area ——————————————————————————————————————————————
        MVC    FEXIOW(NDA3#),LFEXIOW
*
* —— Modify parameter area ——————————————————————————————————————————————————
        LA     R3,DIB
        LA     R4,IOJOBRD
        FEXIOW MF=M,DEVID=DEVID,DIB=(R3),IOPRG=(R4)
*
* —— Execute function ———————————————————————————————————————————————————————
        FEXIOW MF=E,PARAM=(R2)
*
* —— Check return code ——————————————————————————————————————————————————————
        @IF    NE
        CLI    NDA3SR1,NDA3NOE
        @THEN
        ...                           Error
        @BEND
*
*
*_____________________________________________________________________________
* Close device control
*_____________________________________________________________________________
*
* —— Address parameter area —————————————————————————————————————————————————
        LA     R2,PFCLSDV
        @DATA  CLASS=B,BASE=R2,DSECT=FCLSDV
*
* –– Initialize parameter area ––––––––––––––––––––––––––––––––––––––––––––
        MVC    FCLSDV(NDA2#),LFCLSDV
*
* –– Modify parameter area ––––––––––––––––––––––––––––––––––––––––––––––––
        FCLSDV MF=M,DEVID=DEVID
*
* –– Execute function –––––––––––––––––––––––––––––––––––––––––––––––––––––
        FCLSDV MF=E,PARAM=(R2)
*
* –– Check return code ––––––––––––––––––––––––––––––––––––––––––––––––––––
        @IF    NE
        CLI    NDA2SR1,NDA2NOE
        @THEN
        ...                         Error
        @BEND
*
*__________________________________________________________________________
* Dynamic end of the main procedure
*__________________________________________________________________________
*
        @EXIT
*
*
*__________________________________________________________________________
* Constants
*__________________________________________________________________________
*
*    _____________________________________________________________________
*     Initialization data for parameter area
*    _____________________________________________________________________
*
LFOPNDV FOPNDV    MF=L,SYMBTYP='EXOT'
LFCLSDV FCLSDV    MF=L
LFDECCP FDECCP    MF=L
LFDELCP FDELCP    MF=L,CPID=*ALL
LFEXCIO FEXCIO    MF=L
LFCEKIO FCEKIO    MF=L
LFWFTIO FWFTIO    MF=L
LFEXIOW FEXIOW    MF=L
*
LFDEVIB FDEVIB    MF=L
*
*    _____________________________________________________________________
*     Device jobs
*    _____________________________________________________________________
*
IOPRGWR  FDEFIO   MF=L,CMD=1,DATA=A(IOPRGWR),LENGTH=4,              -
                  CHAIN=(IOPRGRD)
*
IOPRGRD  FDEFIO   MF=L,CMD=2,DATA=A(IOPRGRD),LENGTH=80,FLAG=*SLI
*
IOPRGSEN FDEFIO   MF=L,CMD=3,LENGTH=3
*
*__________________________________________________________________________
* End of main procedure
```

```
*____________________________________________________________________________
*
         @END
*____________________________________________________________________________
* DSECT of the variable area created dynamically
*____________________________________________________________________________
*
VAR      @PAR   D=YES
PAR1     DS     F
PAR2     DS     F
*
DEVID    DS     F                Device ID
CPID     DS     F                Channel program ID
IOID     DS     F                I/O identifier
DIB      DS     0D,XL(NDAJ#)
IOJOBWR  DS     0D,XL(NDAI#)
IOJOBRD  DS     0D,XL(NDAI#)
PFOPNDV  DS     0D,XL(NDA1#)
PFCLSDV  DS     0D,XL(NDA2#)
PFEXIOW  DS     0D,XL(NDA3#)
PFEXCIO  DS     0D,XL(NDA4#)
PFCEKIO  DS     0D,XL(NDA5#)
PFWFTIO  DS     0D,XL(NDA6#)
PFDECCP  DS     0D,XL(NDA8#)
PFDELCP  DS     0D,XL(NDA9#)
*
DATAWR   DS     CL4
DATARD   DS     CL80
VAR      @PAR   LEND=YES
*
*    ________________________________________________________________________
*     ADAM DSECTs
*    ________________________________________________________________________
*
FOPNDV   FOPNDV  MF=D
FCLSDV   FCLSDV  MF=D
FDEFIO   FDEFIO  MF=D
FDEVIB   FDEVIB  MF=D
FEXCIO   FEXCIO  MF=D
FCEKIO   FCEKIO  MF=D
FWFTIO   FWFTIO  MF=D
FEXIOW   FEXIOW  MF=D
FDECCP   FDECCP  MF=D
FDELCP   FDELCP  MF=D
*
         END     FADAM01
```

# 8 Appendix: old macros

## 8.1 General information

The appendix contains the macro calls that are available for programming purposes up to and including ADAM V11.0A.
These macros continue to be supported for reasons of compatibility. You should not, however, use them in new user programs.

You will find a brief description of the macros and operands on the following pages. The macro call formats, as well as the return codes and error flags, are included in their entirety.

## 8.2 Metacharacters

Explanation of the metacharacters used in this manual:

| **Formal notation** | **Explanation** | **Example** |
|---|---|---|
| UPPERCASE LETTERS | Uppercase letters denote constants. | DEVICE=DISKETTE |
| lowercase letters | Lowercase letters denote variables. | SECTORS=number |
| { } | Braces enclose alternatives. | TAPE=$\left\{ \begin{matrix} Y \\ E \end{matrix} \right\}$ |
| \| | A vertical line separates alternative entries (i.e. it performs the same function as braces). | TAPE=YES\|NO |
| [ ] | Square brackets enclose optional entries. | filename [,ERASE] |
| NO | Underscoring indicates the default value. | TAPE=$\left\{ \begin{matrix} Y \\ E \end{matrix} \right\}$ |
| ...... | Periods denote repetition. | (filename, ....) |
| ␣ | This character represents a blank (X'40') | STD␣<br>Write 'STD ' |

Table 7: Metasymbols used in macros (selected entries)

## 8.3 Macro types

Depending on the way in which operands are passed, the macros are either type S (storage) or type O (other). The macro type is indicated in the heading of each macro description.

**S-type** macros allow three forms of macro expansion, which can be specified by means of the MF operand.

$$\text{MF=}\left\{\begin{array}{l}\text{L}\\ \text{(E,address)}\\ \text{(E,(1))}\end{array}\right\}$$

If the MF operand is not specified, the standard form applies, i.e. the operand list and the instruction code are generated.

L Only the operand list is generated.

E Only the instruction code is generated.

address Address of the operand list as a symbolic name.

(1) Register 1 must be loaded with the address of the operand list before the macro is issued.

**Type-O** macros are definition macros. They use either no registers at all, or only those declared as working registers by the macro.

## 8.4 Macros

# BLDIO
# Define device job dynamically (type O)

### Macro description

This macro constructs a new device job or updates an existing device job dynamically (i.e. while the program is running). The memory area in which the specified values are stored must have been reserved by the user. It may be located in any program segment or in a dynamically requested memory area. When a new device job is generated, all operands except WREG1 and WREG2 must be specified. Prior to calling BLDIO to generate a new device job the memory area provided for it should be filled with binary zeroes so that any reserved fields in the operand list of the device job are deleted.

### Call format and operand description

| Name | Operation | Operands |
|---|---|---|
| [name] | BLDIO | ADDR=$\left\{\begin{array}{l}\text{address}\\\text{(r)}\end{array}\right\}$<br>[,CMD=$\left\{\begin{array}{l}\text{operation}\\\text{(r)}\end{array}\right\}$]<br>[,DATA=$\left\{\begin{array}{l}\text{address}\\\text{(r)}\end{array}\right\}$]<br>[,LENGTH=$\left\{\begin{array}{l}\text{length}\\\text{(r)}\end{array}\right\}$]<br>[,FLAG=$\left\{\begin{array}{l}\text{option}\\\text{(r)}\end{array}\right\}$]<br>[,REPEAT=$\left\{\begin{array}{l}\text{number}\\\text{(r)}\end{array}\right\}$]<br>[,CHAIN=$\left\{\begin{array}{l}\text{address}\\\text{NO}\\\text{NEXT}\\\text{(r)}\end{array}\right\}$]<br>[,WREG1=$\left\{\begin{array}{l}\underline{0}\\\text{(r)}\end{array}\right\}$,WREG2=$\left\{\begin{array}{l}\underline{1}\\\text{(r)}\end{array}\right\}$] |

ADDR

Address of the device job or a register containing the address of the device job.

CMD Number of the operation to be executed or a register containing the operation number.

DATA Address of the I/O buffer for transfer data or a register containing the address of the I/O buffer.

LENGTH Length of the transfer data in bytes or a register containing the data length.

FLAG Device job option.

REPEAT The number of times a specific job is to be repeated or a register containing the number of repetitions.

CHAIN Indicates whether and how the device job chain is to continue.

WREG1 Register to be used as the first working register.

WREG2 Register to be used as the second working register.

## CEKIO<br>Check I/O status (type S)

### Macro description

This macro allow you to check the status of a specific I/O operation and to initiate termination processing by ADAM if the I/O operation has been completed.

### Call format and operand description

| Name | Operation | Operands |
|---|---|---|
| `[name]` | `CEKIO` | `DCB={address`<br>`(r)},IOCB=(r)` |

MF operand (see page 98) permitted.

DCB Address of the device control block (DCB) or a register containing the address of the DCB.

IOCB Address of the input/output control block (IOCB) or a register containing the address of the IOCB.

### Return codes and error flags

Register 0: Address of the IOCB associated with the I/O operation.

Register 15: Return code (RC).

General section of the DCB: subcodes 1 and 2 (SC1 and SC2).

| Flags | | | Reference | Meaning | Notes |
|---|---|---|---|---|---|
| RC | SC1 | SC2 | | | |
| 00 | 00 | 00 | IOCB | I/O terminated without error | a) |
| 04 | -- | -- | ----- | Job could not be interpreted | |
| 08 | 00 | 00 | OPNDV | Job not executed owing to no devoce control open | |
| 08 | 04 | 00<br>20 | IOCB<br>IOCB | Job not executed due to:<br>Invalid IOCB address<br>Invalid I/O buffer address | |
| 08 | 08 | 20 | IOCB | Job not executed because IOCB not assigned to any I/O operation | |
| 08 | 0C | 00 | ----- | Job not executed due to failure of a system function | |
| 10 | 00 | 00 | IOCB | I/O not yet terminated | |
| 14 | 00 | 00 | IOCB | I/O terminated with error | a) |

a) I/O terminated, result data in the IOCB; the address of the IOCB associated with the I/O is stored in register 0.

## CEKSG
## Check device signal status (type S)

### Macro description

This macro enables you to check whether a device signal (attention interrupt) has arrived. If so, termination processing for this signal is initiated. You must submit a request to ADAM to receive a device signal (RECSG macro) before calling this macro.

### Call format and operand description

| Name | Operation | Operands |
|---|---|---|
| [name] | CEKSG | DCB= $\left\{\begin{array}{l}\text{address}\\ \text{(r)}\end{array}\right\}$ |

MF operand (see page 98) permitted.

DCB Address of the device control block (DCB) or a register containing the address of the DCB.

### Return codes and error flags

Register 15: Return code (RC).

General section of the DCB: subcodes 1 and 2 (SC1 and SC2).

| Flags | | | Reference | Meaning | Notes |
|---|---|---|---|---|---|
| RC | SC1 | SC2 | | | |
| 00 | 00 | 00 | DCB | Device signal has arrived | a) |
| 04 | -- | -- | ----- | Job could not be interpreted | |
| 08 | 00 | 00 | OPNDV | Job not executed owing to no device conntrol open | |
| 08 | | | | Job not executed due to: | |
| | 08 | 10 | DCB | No device signal processing specified at device control open | |
| | | 20 | DCB | No receive signal job requested | |
| 08 | 0C | xx | ----- | Job not executed owing to failure of a system function. In this case, subcode 2 (SC2) contains a system error code intrinsic to ADAM | |
| 10 | 00 | 00 | DCB | Signal not yet arrived | |

a) Receive signal job completed, result data in signal control block.

# CLSDV
# Close device control (type S)

## Macro description

This macro relinquishes control of a device reserved by means of an OPNDV macro.

## Call format and operand description

| Name | Operation | Operands |
|---|---|---|
| [name] | CLSDV | DCB=$\left\{\begin{array}{l}\texttt{address}\\\texttt{(r)}\end{array}\right\}$ |

MF operand (see page 98) permitted.

DCB Address of the device control block (DCB) or a register containing the address of the DCB.

## Return codes and error flags

Register 15: Return code (RC).

General section of the DCB: subcodes 1 and 2 (SC1 and SC2).

| Flags | | | Reference | Meaning | Notes |
|---|---|---|---|---|---|
| RC | SC1 | SC2 | | | |
| 00 | 00 | 00 | DCB | Device control closed | a) |
| 04 | -- | -- | ----- | Job could not be interpreted | |
| 08 | 00 | 00 | OPNDV | Job not executed owing to no device control open | |
| 08 | 04 | 00 | DCB | Job not executed because IOCB/SGCB not assigned | |

a) The result data of implicitly terminated I/O operations is stored in the associated IOCBs; the interrupt data of an implicitly terminated receive signal job is stored in the signal control block.

# DECCP
# Generate channel program (type S)

### Macro description

This macro makes a device job chain known to ADAM.
The chain is defined by the BLDIO or DEFIO macro. ADAM generates a channel program from the device job chain.

### Call format and operand description

| Name | Operation | Operands |
|---|---|---|
| [name] | DECCP | $\texttt{DCB=}\left\{\begin{array}{l}\texttt{address}\\ \texttt{(r)}\end{array}\right\}\texttt{,IOPRG=}\left\{\begin{array}{l}\texttt{address}\\ \texttt{(r)}\end{array}\right\}$ |

MF operand (see page 98) permitted.

DCB Address of the device control block (DCB) or a register containing the address of the DCB.

IOPRG Address of the device job chain or a register containing the address of the device job chain to be processed.

### Return codes and error flags

Register 0: Channel program ID (only if return code is 00).

Register 15: Return code (RC).

General section of the DCB: Subcodes 1 and 2 (SC1 and SC2).

| Flags | | | Reference | Meaning | Notes |
|---|---|---|---|---|---|
| RC | SC1 | SC2 | | | |
| 00 | 00 | 00 | IOPRG | Channel program generated | a) |
| 04 | -- | -- | ----- | Job could not be interpreted | |
| 08 | 00 | 00 | OPNDV | Job not executed owing to no device control open | |
| 08 | | | | Job not executed due to: | |
| | 04 | 04 | IOPRG | Invalid device job address | |
| | | 20 | IOPRG | Inavlid I/O buffer address | |
| | | 24 | IOPRG | Invalid data length | |
| | | 28 | IOPRG | Invalid operation number | |
| | | 2C | IOPRG | Invalid job option | |
| | | 30 | IOPRG | Invalid repetition factor | |
| | | 34 | IOPRG | Invalid continuation address in device job | |
| | | 38 | IOPRG | Device job option DC not permitted for this device | |
| | | 3C | IOPRG | Device job option DC not permitted for this operation | |
| | | 40 | IOPRG | Device job option DC not permitted for repetition factor 0 | |
| | | 44 | IOPRG | Device job option DC incompatible with continuation address | |
| 08 | 08 | 34 | DCB | Job not executed due to maximum number of channel programs being reached | |
| 08 | 0C | XX | ----- | Job not executed owing to failure of a system function. In this case, subcode 2 (SC2) contains a system error code intrinsic to ADAM | |

a) Channel program ID in register 0.

# DEFIO
# Define device job statically (type O)

### Macro description

This macro is used to define a device job at assembly time, i.e. before the program is started (statically).

### Call format and operand description

| Name | Operation | Operands |
|---|---|---|
| [name] | DEFIO | {CMD=operation<br>[,DATA=address]<br>[,LENGTH=length]<br>[,FLAG=option]<br>[,REPEAT=number]<br>[,CHAIN={address<br>NO<br>NEXT}]<br>[D],{prefix<br>*}]} |

CMD Number of the operation to be executed.

DATA Address of the I/O buffer for transfer data.

LENGTH Length of the transfer data.

FLAG Device job option.

REPEAT Repetition factor for a specific job.

CHAIN Indicates how the device job chain is to continue. With this operand it is possible to form a device job chain from a number of different subchains.

D Indicates whether is DSECT is to be generated.

## DELCP
## Delete channel program (type S)

### Macro description

This macro deletes channel programs created by means of the DECCP macro.

### Call format and operand description

| Name | Operation | Operands |
|---|---|---|
| [name] | DELCP | { DCB={address<br>(r)} [,CPRG=(r)]<br>ALL=YES } |

MF operand (see page 98) permitted.

DCB Address of the device control block (DCB) or a register containing the address of the DCB.

CPRG Register containing the channel program ID.

ALL Deletes all channel programs associated with the task.

### Return codes and error flags

Register 15: Return code (RC).
General section of the DCB (not if ALL = YES): subcodes 1 and 2 (SC1 and SC2)

| Flags | | | Reference | Meaning |
|---|---|---|---|---|
| RC | SC1 | SC2 | | |
| 00 | 00 | 00 | IOCB | Channel program deleted |
| 04 | -- | -- | ----- | Job could not be interpreted |
| 08 | 00 | 00 | OPNDV | Job not executed owing to no device control open |
| 08 | 04 | 10 | CRPG | Job not executed due to incorrect channel program ID |
| 08 | 08 | 20 | DCB,ALL | Job not executed because channel program does not exist |
| 0C | 00 | 00 | DCB,<br>CPRG<br>ALL | Job not, or only partially executed, because one or more channel programs are still active |

# DEVCB
# Create device control block (type O)

### Macro description

This macro creates the device control block (DCB) necessary for controlling device access.

### Call format and operand description

| Name | Operation | Operands |
|---|---|---|
| [name] | DEVCB | MAXREQ#=number<br>NO<br>[,SGCB=<br>YES ]<br>MNEMON=mnemonic<br>[,<br>SYMBTYP=typ ]<br>prefix<br>[D],<br>* |

MAXREQ# Maximum number of I/O control blocks (IOCBs) that are to be created in the DCB.

SGCB Indicates whether a signal control block (SGCB) is to be set up in the DCB.

MNEMON Mnemonic device name of a specific device.

SYMBTYP Symbolic device type. This device type must have been defined when the system was generated and must be supported by ADAM.

D A DSECT must be generated. Default: no DSECT is generated.

prefix Not more than 3 characters, alphanumeric, first character must be a letter. All the names generated are given this prefix.

* All the names generated are given the default prefix ADM.

# EXCIO
# Initiate I/O operation (type S)

### Macro description

This macro initiates the execution of an I/O operation. Control is returned to the user immediately after the I/O operation is initiated.

### Call format and operand description

| Name | Operation | Operands |
|---|---|---|
| [name] | EXCIO | $\text{DCB=}\left\{\begin{array}{l}\text{address}\\ \text{(r)}\end{array}\right\}$<br>$\left\{\begin{array}{l}\text{,IOPRG=}\left\{\begin{array}{l}\text{address}\\ \text{(r)}\end{array}\right\}\\ \text{,CPRG=(r)}\end{array}\right\}$<br>$\text{[,IOTERM=}\left\{\begin{array}{l}\underline{\text{DEVEND}}\\ \text{CHNEND}\end{array}\right\}\text{]}$<br>$\text{[,DER=}\left\{\begin{array}{l}\underline{\text{NO}}\\ \text{YES}\end{array}\right\}\text{]}$ |

MF operand (see page 98) permitted.

DCB
Address of the device control block (DCB) or a register containing the address of the DCB.

IOPRG
Address of the device job chain for which the channel program is to be generated, or a register containing the address of the device job chain to be processed.

CPRG
Register containing the channel program ID returned by ADAM in register 0 following the DECCP call.

IOTERM
Indicates whether I/O is to terminated when the end-of-message signal is received from the controller or only when the end-of-message signal is received from the device.

DER
Device error recovery routine to be used in the event of an I/O error.

### Return codes and error flags

Register 0: Address of the IOCB associated with the I/O.
Register 15: Return code (RC).
General section of the DCB: subcodes 1 and 2 (SC1 and SC2).

| Flags | | | Reference | Meaning | Notes |
|---|---|---|---|---|---|
| RC | SC1 | SC2 | | | |
| 00 | 00 | 00 | IOPRG,<br>CPRG | I/O initiated | a) |
| 04 | -- | -- | ----- | Job could not be interpreted | |
| 08 | | | | Job not executed owing to no device control | |
| | 00 | 00 | OPNDV | open | |
| 08 | | | | Job not executed due to: | |
| | 04 | 04 | IOPRG | Invalid device job address | |
| | | 10 | CPRG | Incorrect channel program IF | |
| | | 20 | IOPRG,<br>CPRG | Invalid I/O buffer address | |
| | | 24 | IOPRG | Invalid data length | |
| | | 28 | IOPRG | Invalid operation number | |
| | | 2C | IOPRG | Invalid device job option | |
| | | 30 | IOPRG | Invalid repretition factor | |
| | | 34 | IOPRG | Invalid continuation address in device job | |
| | | 38 | IOPRG | Device job option DC not permitted for this device | |
| | | 3C | IOPRG | Device job option DC not permitted for this operation | |
| | | 40 | IOPRG | Device job option DC not permitted for repetition factor 0 | |
| | | 44 | IOPRG | Device job option DC incompatible with continuation address | |
| 08 | | | | Job not executed due to: | |
| | 08 | 30 | DCB | No free IOCB available in the DCB | |
| | | 34 | DCB | Maximum nujmber of parallel I/Os reached | |
| | | 44 | DCB | Invalid event item ID | |
| | | 48 | | Resources temporarily not available | |
| 08 | | | | Job not executed owing to failure of a system | |
| | 0C | XX | ----- | function. In this case, subcode 2 (SC2) contains a system error code intrinsic to ADAM | |
| 0C | 00 | 00 | CPRG | I/O not initiated because channel program is still in use | |

a) The address of the IOCB used to handle the I/O operation is stored in register 0.

# EXIOW
# Initiate I/O and wait for termination (type S)

## Macro description

This macro initiates an I/O operation and waits until it has ended and termination processing has been completed.

## Call format and operand description

| Name | Operation | Operands |
|---|---|---|
| [name] | EXIOW | DCB=$\left\{\begin{array}{l}\text{address}\\\text{(r)}\end{array}\right\}$<br>$\left\{\begin{array}{l}\text{,IOPRG=}\left\{\begin{array}{l}\text{address}\\\text{(r)}\end{array}\right\}\\\text{,CPRG=(r)}\end{array}\right\}$<br>[,IOTERM=$\left\{\begin{array}{l}\underline{\text{DEVEND}}\\\text{CHNEND}\end{array}\right\}$ ]<br>[,DER=$\left\{\begin{array}{l}\underline{\text{NO}}\\\text{YES}\end{array}\right\}$] |

MF operand (see page 98) permitted.

DCB
Address of the device control block (DCB) or a register containing the address of the DCB.

IOPRG
Address of the device job chain for which the channel program is to be generated, or a register containing the address of the device job chain to be processed.

CPRG
Register containing the channel program ID returned by ADAM in register 0 following the DECCP call.

IOTERM
Indicates whether I/O is to terminated when the end-of-message signal is received from the controller or only when the end-of-message signal is received from the device. This operand is only relevant for devices connected to type-2 channels.

DER
Device error recovery routine to be used in the event of an I/O error.

### Return codes and error flags

Register 0: Address of the IOCB associated with the I/O operation.
Register 15: Return code (RC).
General section of the DCB: subcodes 1 and 2 (SC1 and SC2).

| Flags | | | Reference | Meaning | Notes |
|---|---|---|---|---|---|
| RC | SC1 | SC2 | | | |
| 00 | 00 | 00 | IOPRG,CPRG | I/O terminated without error | a) |
| 04 | -- | -- | ----- | Job could not be interpreted | |
| 08 | 00 | 00 | OPNDV | Job not executed owing to no device control open | |
| 08 | | | | Job not executed due to: | |
| | 04 | 04 | IOPRG | Invalid device job address | |
| | | 10 | CPRG | Incorrect channel program ID | |
| | | 20 | IOPRG,CPRG | Invalid I/O buffer address | |
| | | 24 | IOPRG | Invalid data length | |
| | | 28 | IOPRG | Invalid operation number | |
| | | 2C | IOPRG | Invalid device job option | |
| | | 30 | IOPRG | Invalid repetition factor | |
| | | 34 | IOPRG | Invalid continuation address in device job | |
| | | 38 | IOPRG | Device job option DC not permitted for this device | |
| | | 3C | IOPRG | Device job option DC not permitted for this operation | |
| | | 40 | IOPRG | Device job option DC not permitted for repetition factor 0 | |
| | | 44 | IOPRG | Device job option DC incompatible with continuation address | |
| 08 | | | | Job not executed due to: | |
| | 08 | 30 | DCB | No free IOCB in the DCB | |
| | | 34 | DCB | Maximum number of parallel I/Os reached | |
| | | 38 | DCB | An event item ID has been entered in the DCB (ADMDEIID field). This entry is only permissible with EXCIO. | |
| 08 | 0C | XX | ----- | Job not executed owing to failure of a system function. In this case, subcode 2 (SC2) contains a system error code intrinsic to ADAM | |
| 0C | 00 | 00 | CPRG | I/O not initiated because channel program is still in use | |
| 14 | 00 | 00 | IOPRG,CPRG | I/O terminated with error | a) |

a) The address of the IOCB used to handle the I/O is contained in register 0; I/O terminated, result data in IOCB.

# HLTIO
# Force termination of I/O operation (type S)

### Macro description

This macro can be used to force the termination of all I/O operations of a device that were not terminated up to the time of issuing the macro.

### Call format and operand description

| Name | Operation | Operands |
|---|---|---|
| [name] | HLTIO | DCB={ address<br>(r) } |

MF operand (see page 98) permitted.

DCB Address of the device control block (DCB) or a register containing the address of the DCB.

### Return codes and error flags

Register 0: Address of the IOCB associated with a forcibly terminated I/O operation.
Register 15: Return code (RC).

General section of the DCB: subcodes 1 and 2 (SC1 and SC2).

| Flags | | | Reference | Meaning | Notes |
|---|---|---|---|---|---|
| RC | SC1 | SC2 | | | |
| 00 | 00 | 00 | DCB | Waiting I/O operations terminated. I/O abort initiated | a) |
| 04 | -- | -- | ----- | Job could not be interpreted | |
| 08 | 00 | 00 | OPNDV | Job not executed owing to no device control open | |
| 08 | 08 | 20 | DCB | Job not executed because no I/O was initiated | |
| 14 | 00 | 00 | DCB | I/O operations already terminated | |

a) After the HLTIO macro, the address of the IOCB for one of the forcibly terminated I/O operations is stored in register 0.

## OPNDV
## Open device control (type S)

### Macro description

This macro initiates device control and grants the program right of access to the desired device. The macro can be used to change any of the static values in the device control block.

### Call format and operand description

| Name | Operation | Operands |
|---|---|---|
| [name] | OPNDV | DCB=$\left\{\begin{array}{l}\texttt{address}\\\texttt{(r)}\end{array}\right\}$<br>[,$\left\{\begin{array}{l}\texttt{MNEMON=}\left\{\begin{array}{l}\texttt{mnemonic}\\\texttt{(r)}\end{array}\right\}\\\texttt{SYMBTYP=}\left\{\begin{array}{l}\texttt{typ}\\\texttt{(r)}\end{array}\right\}\end{array}\right\}$]<br>[,DER=SYSTEM]<br>[,ADAMDER=$\left\{\begin{array}{l}\texttt{NODER}\\\texttt{option}\\\texttt{(option,option, ...)}\end{array}\right\}$]<br>[,MAXREQ#=$\left\{\begin{array}{l}\texttt{number}\\\texttt{(r)}\end{array}\right\}$]<br>[,SIGNALS=$\left\{\begin{array}{l}\underline{\texttt{NO}}\\\texttt{YES}\end{array}\right\}$] |

MF operand (see page 98) permitted.

DCB
Address of the device control block (DCB) or a register containing the address of the DCB.

MNEMON
Mnemonic device name or a register containing the address of the mnemonic device name (2 bytes).

SYMBTYP
Symbolic type designation of the device or a register containing the address of the symbolic device type (8 bytes).

DER=SYSTEM Indicates that a device-type-specific error recovery routine (SYSTEM-DER) is to be provided for the specified device.

ADAMDER Permits restrictions to error recovery using ADAM-DER.

MAXREQ# Maximum number of parallel I/O operations possible during a device control function, or a register containing the maximum number of parallel I/O operations.

SIGNALS Indicates whether processing of device signals (attention interrupts) is also to be possible during the device control function.

### Return codes and error flags

Register 15: Return code (RC).

General section of the DCB: subcodes 1 and 2 (SC1 and SC2).

| Flags | | | Reference | Meaning |
|---|---|---|---|---|
| RC | SC1 | SC2 | | |
| 00 | 00 | 00 | DCB | Device control opened |
| 04 | -- | -- | ----- | Job could not be interpreted |
| 08 | | | | Job not executed due to: |
| | 04 | 00 | DCB | IOCB not allocated |
| | | 04 | DCB | SGCB not allocated |
| | | 50 | MNEMON, SYMBTYP | Device not assigned or already reserved or missing system privilege |
| | | 60 | MAXREQ# | Invalid value for #MAXREQ |
| | | 64 | MNEMON | Invalid device name |
| | | 68 | SYMBTYP | Invalid device type |
| | | 6C | SIGNALS | No signal control block in DCB |
| 08 | | | | Job not executed due to: |
| | 08 | 50 | MNEMON, SYMBTYP | Device not reserved or not generated |
| | | 54 | MNEMON, SYMBTYP | Device-specific ADAM module cannot be loaded:<br>- not in ADAM library<br>- library is not available<br>- incorrect module version<br>- incorrect channel type |
| 08 | | | | Job not executed owing to failure of a system |
| | 0C | XX | ----- | function. In this case, subcode 2 (SC2) contains a system error code intrinsic to ADAM |
| FF | -- | -- | ------ | ADAM not called in caller's system environment |

# RECSG
# Request device signal (type S)

## Macro description

This macro is used to instruct ADAM to receive the next device signal (attention interrupt).

## Call format and operand description

| Name | Operation | Operands |
|---|---|---|
| [name] | RECSG | DCB={address<br>(r)} |

DCB Address of the device control block (DCB) or a register containing the address of the DCB.

## Return codes and error flags

Register 15: Return code (RC).

General section of the DCB: subcodes 1 and 2 (SC1 and SC2).

| Flags | | | Reference | Meaning |
|---|---|---|---|---|
| RC | SC1 | SC2 | | |
| 00 | 00 | 00 | DCB | Receive signal job accepted |
| 04 | -- | -- | ----- | Job could not be interpreted |
| 08 | 00 | 00 | OPNDV | Job not executed owing to no device control open |
| 08 | | | | Job not executed due to: |
| | 08 | 10 | DCB | No device signal processing specified at device control open |
| | | 30 | DCB | Preceding receive signal job not yet completed |
| | | 48 | | Resources temporarily not available |
| 08 | 0C | XX | ----- | Job not executed owing to failure of a system function. In this case, subcode 2 (SC2) contains a system error code intrinsic to ADAM |

# TIMOT
# Define timeout interval (type S)

## Macro description

This macro controls the timeout period for I/O operations.

## Call format and operand description

| Name | Operation | Operands |
|---|---|---|
| [name] | TIMOT | $\text{DCB=}\left\{\begin{array}{l}\text{address}\\ \text{(r)}\end{array}\right\}\text{, TIMOUT=}\left\{\begin{array}{l}\text{number}\\ \text{*OFF}\\ \text{(r)}\end{array}\right\}$ |

MF operand (see page 98) permitted.

DCB Address of the device control block (DCB) or a register containing the address of the DCB.

TIMEOUT Indicates the timeout value (in seconds) or a register containing the timeout value. In addition, you can also specify whether timeout monitoring is to be disabled.

## Return codes and error flags

Register 15: Return code (RC).

General section of the DCB: subcodes 1 and 2 (SC1 and SC2).

| Flags | | | Reference | Meaning |
|---|---|---|---|---|
| RC | SC1 | SC2 | | |
| 00 | 00 | 00 | TIMEOUT | Timeout value changed |
| 04 | -- | -- | ----- | Job could not be interpreted |
| 08 | 00 | 00 | OPNDV | Job not executed owing to no device control open |
| 08 | 04 | 14 | TIMEOUT | Job not executed because TIMEOUT=*OFF not permitted for device |

# WFSG
# Wait for device signal (type S)

### Macro description

This macro allows you to wait for the arrival of a device signal (attention interrupt). You must request ADAM to receive a device signal (RECSG macro) before you call this macro.

### Call format and operand description

| Name | Operation | Operands |
|---|---|---|
| [name] | WFSG | $\text{DCB=}\left\{\begin{array}{l}\text{address}\\ \text{(r)}\end{array}\right\}\text{,WTIME=}\left\{\begin{array}{l}\text{time}\\ \text{(r)}\end{array}\right\}$ |

DCB Address of the device control block (DCB) or a register containing the address of the DCB.

WTIME Maximum waiting time (in seconds) or a register containing the time value.

### Return codes and error flags

Register 15: Return code (RC).
General section of the DCB: Subcodes 1 and 2 (SC1 and SC2).

| Flags | | | Reference | Meaning | Notes |
|---|---|---|---|---|---|
| RC | SC1 | SC2 | | | |
| 00 | 00 | 00 | DCB | Device signal has arrived | a) |
| 04 | -- | -- | ----- | Job could not be interpreted | |
| 08 | 00 | 00 | OPNDV | Job not executed owing to no device control open | |
| 08 | 08 | 10<br>20 | DCB<br>DCB | Job not executed due to:<br>No device signal processing specified at device control open<br>No preceding receive signal job | |
| 08 | 0C | XX | ----- | Job not executed owing to failure of a system function. In this case, subcode 2 (SC2) contains a system error code intrinsic to ADAM. | |
| 10 | 00 | 00 | WTIME | Signal not arrived before expiration of waiting time | |

a) Receive signal job completed, result data in signal control block.

## WFTIO
## Wait for termination of I/O operation (type S)

### Macro description

This macro places the task in a wait state until the I/O operation is terminated and termination processing has been completed.

### Call format and operand description

| Name | Operation | Operands |
|---|---|---|
| [name] | WFTIO | $\text{DCB=}\left\{\begin{array}{l}\text{address}\\ \text{(r)}\end{array}\right\}\text{, IOCB=(r)}$ |

MF operand (see page 98) permitted.

DCB  Address of the device control block (DCB) or a register containing the address of the DCB.

IOCB  Address of the IOCB used at initiation of the I/O operation or a register containing the address of the IOCB.

### Return codes and error flags

Register 0:  Address of the IOCB associated with the I/O operation.
Register 15:  Return code (RC).
General section of the DCB: Subcodes 1 and 2 (SC1 and SC2).

| Flags | | | Reference | Meaning | Notes |
|---|---|---|---|---|---|
| RC | SC1 | SC2 | | | |
| 00 | 00 | 00 | IOCB | I/O terminated without errors | a) |
| 04 | -- | -- | ----- | Job could not be interpreted | |
| 08 | 00 | 00 | OPNDV | Job not executed owing to no device control open | |
| 08 | 04 | 00<br>20 | IOCB<br>IOCB | Job not executed due to:<br>Invalid IOCB address<br>Invalid I/O buffer address | |
| 08 | 08 | 20 | IOCB | Job not executed because IOCB is not assigned to any I/O operation | |
| 08 | 0C | 00 | ----- | Job not executed due to failure of a system function | |
| 14 | 00 | 00 | IOCB | I/O terminated with error | a) |

a) I/O terminated, result data in IOCB; the address of the IOCB associated with the I/O is stored in register 0.

# Related publications

The manuals are available as online manuals, see *http://manuals.ts.fujitsu.com*, or in printed form which must be paid and ordered separately at *http://manualshop.ts.fujitsu.com*.

[1] **BS2000/OSD-BC**
System Installation
User Guide

[2] **BS2000/OSD-BC**
Executive Macros
User Guide

[3] **SECOS** (BS2000/OSD)
Security Control System
User Guides

# Index