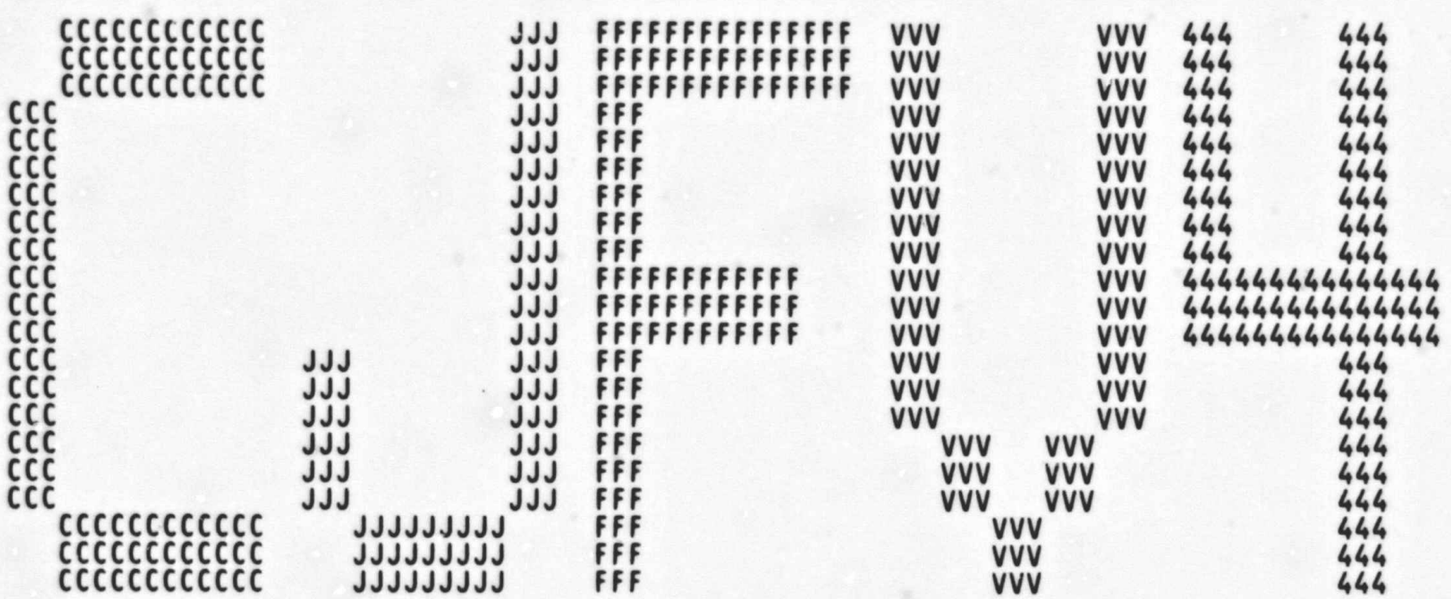
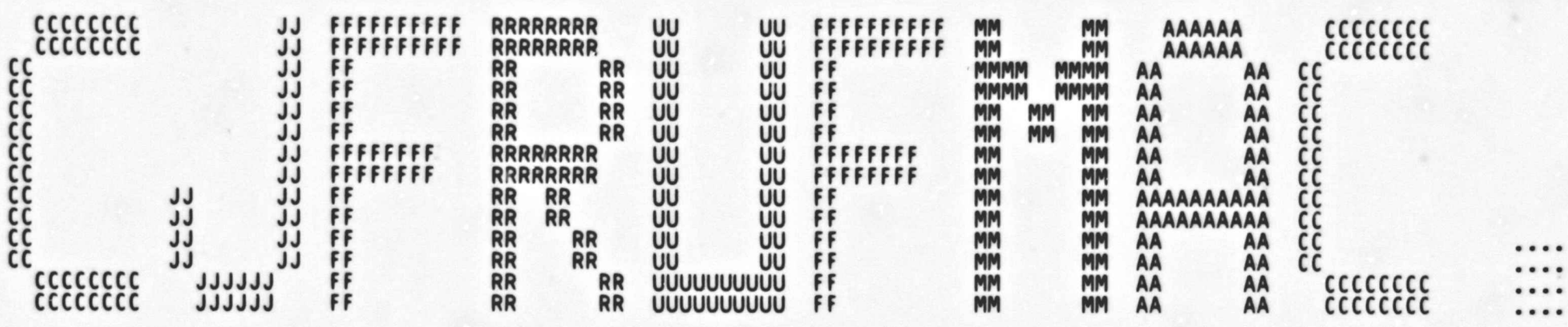
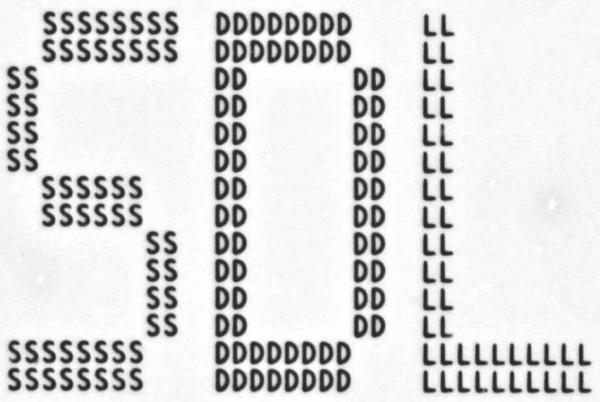

CCCCCCCCCCC JJJ FFFFFFFFFFFFFFF VVV VVV 444 444
CCCCCCCCCCC JJJ FFFFFFFFFFFFFFF VVV VVV 444 444
CCCCCCCCCCC JJJ FFFFFFFFFFFFFFF VVV VVV 444 444
CCC JJJ FFF VVV VVV 444 444
CCC JJJ FFF VVV VVV 444 444
CCC JJJ FFF VVV VVV 444 444
CCC JJJ FFF VVV VVV 444 444
CCC JJJ FFF VVV VVV 444 444
CCC JJJ FFFFFFFFFFFFF VVV VVV 444444444444444
CCC JJJ FFFFFFFFFFFFF VVV VVV 444444444444444
CCC JJJ FFFFFFFFFFFFF VVV VVV 444444444444444
CCC JJJ JJJ FFF VVV VVV 444
CCC JJJ JJJ FFF VVV VVV 444
CCC JJJ JJJ FFF VVV VVV 444
CCC JJJ JJJ FFF VVV VVV 444
CCC JJJ JJJ FFF VVV VVV 444
CCC JJJ JJJ FFF VVV VVV 444
CCCCCCCCCCCC JJJJJJJJJ FFF VVV 444
CCCCCCCCCCCC JJJJJJJJJ FFF VVV 444
CCCCCCCCCCCC JJJJJJJJJ FFF VVV 444

FILEID**CJFRUFMAC





```
{       .TITLE  CJFRUFMAC - VAX-11 SDL MACROS FOR CJF AND RUF "SYSTEM SERVICE" CALLS
{       .IDENT  'V04-000'       /* Please read the comment about the TYPE key-
{                               /* word before adding services to this file!
{                               /* The comment follows the modification history.
{
{*****************************************************************************
{*                                                                           *
{*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
{*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
{*  ALL RIGHTS RESERVED.                                                     *
{*                                                                           *
{*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
{*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
{*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
{*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
{*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
{*  TRANSFERRED.                                                             *
{*                                                                           *
{*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
{*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
{*  CORPORATION.                                                             *
{*                                                                           *
{*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
{*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
{*                                                                           *
{*****************************************************************************

{++
{ FACILITY: CJF Macros

{ ABSTRACT:

{       This module contains the VAX-11 SDL macros for calling CJF and RUF
{       services.   These macros allow the user to specify arguments with
{       keywords,  and to omit arguments which have default values.

{       NOTE:   This module is modeled after STARLET.SDL

{ ENVIRONMENT:

{ AUTHOR: Joost Verhofstad      , CREATION DATE: 29-MAR-1983

{ MODIFIED BY:

{       V03-007 RAS0220         Ron Schaefer            8-Dec-1983
{               Change module name back to STARLET since STARLET hack is
{               NOT gone.

{       V03-006 CWH3006         CW Hobbs                6-Dec-1983
{               Change items in $GETRUI to be optional, change module
{               name since STARLET hack is gone.

{       V03-005 MKL0209         Mary Kay Lyons          30-Nov-1983
{               Change ITMLST to LSTADR in CRENWV call.
{
```

```
{       V03-004 PRB0253         Paul Beck       15-Sep-1983  14:28
{       Change names of RUF macros to include RU designation.
{
{       V03-003 MKL0166         Mary Kay Lyons          18-AUG-1983
{       Add $GETCJI
{
{       V03-002 JSV0247         Joost Verhofstad        09-MAY-1983
{       Change $DCNJNLF
{
{       V03-001 CWH0001         CW Hobbs                27-Apr-1983
{       Add the TYPE keyword to every parameter so that automatic
{       documentation generators will have enough information to
{       produce useful results.  These are added as comments until
{       the SDL group adds the type keyword.
{--


{+++
{       Each of the parameter declarations must have a TYPE entity which
{       is used to supply additional information for the documentation group.
{       A special SDL backend takes this information and automatically
{       produces data type information for the system services manual.
{
{       Please try to find the entity in this table which most accurately
{       describes the datatype of a parameter for a service.  If you have
{       questions about these datatypes, please direct them to CW Hobbs
{       (DELPHI::HOBBS) or Mike Fallet (GALAXY::FALLET).
{
{       SDL currently does not support the TYPE keyword, so these entities
{       are added as local comments.  Note that each parameter line is
{       followed by exactly:
{
{               <tab>{/*<space>TYPE(xxx)?
{
{       (where ? is a comma, semicolon, or null depending on how the
{       original line was terminated) so that the comments can be quickly
{       converted to actual lines when the TYPE keyword is added to SDL.
{       This format is also required by some temporary tools used by the
{       documentation group (temporary until TYPE is a formal keyword).
{
{       Please try to use EXACTLY this format for your TYPE lines.


{ACMODE   - Access mode
{       Hardware access mode, as in User, Supervisor, Executive and
{       Kernel
{
{ADDRESS  - Memory address
{       Address of a location in memory, of either data or code.  Not the
{       address of an entry mask for code.
{
{ARGLIST  - Procedure argument list
{       Structure is a counted argument list, as for the VAX CALL
{       instructions.
```

ASTADR - Address of AST routine
Address of the entry mask of routine which will be called at AST level, which includes RMS Error and Success routines.

BOOLEAN - Boolean truth value flag
(Some parameters which only allow 0 and 1 are not really boolean - see if saying that parm=true makes sense. If not, call it an NUMBER. For an example, look at the REGION parameter on $EXPREG. REGION=TRUE sounds quite silly even though 0 and 1 are the only possibilities.)

CHANNEL - I/O channel
Address of an I/O channel

CHARDESC - Character string
Character string descriptor. Note that several common character strings have their own types, e.g. DEVNAME, LOGNAME, SECTNAME.

CNTRLBLK - Control block
A structure which is interpreted by the service. The elements of the structure are heterogeneous. Note that several common structures have their own types, e.g. FAB, RAB, EXHBLOCK. (Contrast with LIST and VECTOR).

CONDVALU - Condition value
A return status or system condition code, as is returned by a procedure in R0.

CONTEXT - Context
A piece of information used by the service to maintain position over an iterative sequence of calls. Probably initialized by the user to start the sequence, but thereafter manipulated by the service.

DEVNAME - Device name
A character string describing a device name. This can be a logical name, but it must translate to a name which is valid for a device name.

EFCLUSTR - Event flag cluster name
A character string describing an event flag cluster name. This can be a logical name, but it must translate to a name which is valid for an event flag cluster.

EFNUM - Event flag number
An integer representing the number of an event flag.

ENTRYADR - Procedure entry address
The address of a procedure entry mask. This procedure will not called at AST level (use ASTADR type for that).

EXHBLOCK - Exit handler control block
A control block describing an exit handler

FAB - File access block
An RMS File Access Block

FILEPROT - File protection mask
A 16-bit mask describing the file protection for system,owner, group and world

FUNCCODE - Function code
A function code, as for a QIO service. This is a combination of a NUMBER and a MASK. If a function code is a simple enumeration of values, it should be type NUMBER.

HOLDER - Access rights holder
The holder entity for the access rights services.

IOSB - I/O status block
A 64-bit structure which describes the results of an I/O or similar (i.e. $GETxxx) operation

ITEMLIST - Item list
An item list, consisting of groups of 3 longword items which is terminated by a longword 0

LIST - List
An array of elements, terminated by an element (or partial element) with a particular value. The quota list for $CREPRC is an example.

LOCKID - Lock identifier
A number identifying a particular lock, assigned by the system when the lock was granted

LOCKSTAT - Lock status block
Receives status of a lock request, contains the LOCKID and an optional LOCKVALU block

LOCKVALU - Lock value block
A 16-byte block to contain a lock value

LOGNAM - Logical name string
A 1 to 63 character string for a logical or equivalence name. Some types are passed by logical names, but ultimately must translate to a string more restricted than a logical name, for example a DEVNAME. If this is the case, use the restricted name.

MASK - Mask
A group of flags or bitmasks, interpreted by the individual service

NULLARG - Null argument
A place holding argument

NUMBER - An integer count
A signed or unsigned quantity which is a count of something, such as a number of pages or the length of a character string. Also used for an indicator with a number of unique values, such as the TBLFLG argument to $CRELOG, where 0->system, 1->group, and

```
        2-> process name table.

PAGEPROT - Hardware page protection
        The 4-bit memory management page protection recognized by
        the VAX hardware

PRIVMASK - Privilege mask
        A 64-bit mask of process privileges

PROCID   - Process ID
        A longword number identifying a process, assigned by the system
        when the process is created

PROCNAME - Process name
        A character string describing the name of a process

QUADID   - Quadword identifier
        An access rights entity consisting of an id (USERID) and the
        attributes mask associated with the id.

QUADTIME - Quadword system time
        A time value in the 64-bit system time format

RAB      - Record access block
        An RMS Record Access Block

RETID    - Returned id
        An identifier which is created by the system and assigned to
        an object, and used on subsequent references to that object.

SECTID   - Section version and validation
        A quadword specifying the version for a global section and
        specifying the criteria for matching that version

SECTNAME - Section name
        A character string describing a process or global section name.
        This can be a logical name, but it must translate to a name
        which is valid for a section.

SYSTEMID - System access id
        An identifier which is used to define a access rights system.

TIMEDESC - Time descriptor
        A character string describing a time value in the standard
        system format.

USERID   - User identifier
        A user identification code, e.g. UIC

USERPARM - User interpreted argument
        A longword quantity interpreted at the discretion of the user,
        for example the ASTPRM parameter for AST services or the
        REQIDT associated with timer services.

VARANGE  - Virtual address range
        A pair of longwords specifying the beginning and ending virtual
```

```
{           address of a range of memory, as used by memory management
{           services
{
{   VARIES   - Varying parameter
{           A parameter which can take multiple types depending on other
{           arguments in the call.  Examples are $FAO parameters and the
{           P1-P6 for QIO calls.
{
{   VECTOR   - Homogeneous array
{           An array of identical items, length either implied or described
{           by one of the parameters.  (A vector terminated by an item with
{           a special value is described as a LIST.)
{---

MODULE STARLET;
/*
/* CJF AND RUF SERVICE MACRO DEFINITIONS
/*

/*
/* $ALLJDR
/*
/* ALLOCATE A DISK OR TAPE DRIVE FOR JOURNALING
/*
/*      DEVNAM    - ADDRESS OF A CHARACTER STRING DESCRIPTOR FOR THE DEVICE
/*                  NAME STRING.
/*     [RSLLEN]   - ADDRESS OF A WORD TO RECEIVE THE LENGTH OF THE ALLOCATED
/*                  DEVICE NAME STRING.
/*     [RSLBUF]   - ADDRESS OF A CHARACTER STRING DESCRIPTOR FOR RECEIVING THE
/*                  PHYSICAL DEVICE NAME STRING.
/*     [JNLACP]   - ADDRESS OF A DESCRIPTOR FOR A JOURNAL ACP PROCESS NAME.
/*

ENTRY CJF$ALLJDR ALIAS $ALLJDR PARAMETER (
        CHARACTER DESCRIPTOR NAMED DEVNAM IN,    {/* TYPE(DEVNAME),
        WORD UNSIGNED REFERENCE NAMED RSLLEN OUT DEFAULT 0,      {/* TYPE(NUMBER),
        CHARACTER DESCRIPTOR NAMED RSLBUF OUT DEFAULT 0,         {/* TYPE(CHARDESC),
        CHARACTER DESCRIPTOR NAMED JNLACP IN DEFAULT 0  {/* TYPE(PROCNAME)
        ) RETURNS LONGWORD;      {/* TYPE(CONDVALU);



/*
/* $ASSJNL
/*
/* ASSIGN A CHANNEL TO A JOURNAL
/*
/*      CHAN      - ADDRESS OF A WORD TO RECEIVE THE CHANNEL NUMBER ASSIGNED.
/*      JNLTYP    - TYPE OF JOURNAL.
/*     [JNLNAM]   - ADDRESS OF A CHARACTER STRING DESCRIPTOR FOR THE JOURNAL
/*                  NAME STRING.
/*     [ACMODE]   - ADDRESS OF A BYTE CONTAINING THE ACCESS MODE TO BE ASSOCIATED
/*                  WITH CHANNEL.
/*     [PROT]     - ADDRESS OF A WORD CONTAINING THE PROTECTION MASK GIVEN TO
```

```
/*                  ENTRIES WRITTEN OVER CHANNEL.
/*     [FACCOD]  - ADDRESS OF A WORD CONTAINING THE FACILITY CODE VALUE GIVEN TO
/*                  ENTRIES.
/*     [FLAGS]   - FLAGS FOR CHANNEL.
/*     [DEVNAM]  - ADDRESS OF A CHARACTER STRING DESCRIPTOR FOR THE DEVICE
/*                  NAME STRING.
/*     [OBJUIC]  - ADDRESS OF A LONGWORD CONTAINING OBJECTS UIC
/*     [SESSID]  - ADDRESS OF A STRING DESCRIPTOR FOR SESSION ID
/*

ENTRY CJF$ASSJNL ALIAS $ASSJNL PARAMETER (
        WORD UNSIGNED REFERENCE NAMED CHAN OUT, {/* TYPE(CHANNEL),
        BYTE UNSIGNED VALUE NAMED JNLTYP IN,    {/* TYPE(NUMBER),
        CHARACTER DESCRIPTOR NAMED JNLNAM IN DEFAULT 0, {/* TYPE(CHARDESC),
        BYTE UNSIGNED REFERENCE NAMED ACMODE IN DEFAULT 0,      {/* TYPE(ACMODE),
        WORD UNSIGNED REFERENCE NAMED PROT IN DEFAULT 0,        {/* TYPE(FILEPROT),
        WORD UNSIGNED REFERENCE NAMED FACCOD IN DEFAULT 0,      {/* TYPE(NUMBER),
        LONGWORD UNSIGNED VALUE NAMED FLAGS IN DEFAULT 0,       {/* TYPE(MASK),
        CHARACTER DESCRIPTOR NAMED DEVNAM IN DEFAULT 0, {/* TYPE(DEVNAME),
        LONGWORD UNSIGNED REFERENCE NAMED OBJUIC IN DEFAULT 0,  {/* TYPE(USERID),
        CHARACTER DESCRIPTOR NAMED SESSID IN DEFAULT 0  {/* TYPE(USERPARM)
        ) RETURNS LONGWORD;     {/* TYPE(CONDVALU);



/*
/* $CREJNL
/*
/* CREATE A JOURNAL
/*
/*      CHAN      - ADDRESS OF A WORD TO RECEIVE THE CHANNEL NUMBER ASSIGNED.
/*      JSB       - ADDRESS OF THE JOURNAL SPECIFICATION BLOCK.
/*     [ACMODE]   - ADDRESS OF A BYTE CONTAINING THE ACCESS MODE TO BE ASSOCIATED
/*                  WITH CHANNEL.
/*     [PROT]     - ADDRESS OF A WORD CONTAINING THE PROTECTION MASK GIVEN TO
/*                  ENTRIES WRITTEN OVER CHANNEL.
/*     [FACCOD]   - ADDRESS OF A WORD CONTAINING THE FACILITY CODE VALUE GIVEN TO
/*                  ENTRIES.
/*     [FLAGS]    - FLAGS FOR CHANNEL.
/*     [OBJUIC]   - ADDRESS OF A LONGWORD CONTAINING OBJECTS UIC
/*     [SESSID]   - ADDRESS OF A STRING DESCRIPTOR FOR SESSION ID
/*     [IOSB]     - ADDRESS OF A IOSB
/*

ENTRY CJF$CREJNL ALIAS $CREJNL PARAMETER (
        WORD UNSIGNED REFERENCE NAMED CHAN OUT, {/* TYPE(CHANNEL),
        ANY NAMED JSB,  {/* TYPE(CNTRLBLK),
        BYTE UNSIGNED REFERENCE NAMED ACMODE IN DEFAULT 0,      {/* TYPE(ACMODE),
        WORD UNSIGNED REFERENCE NAMED PROT IN DEFAULT 0,        {/* TYPE(FILEPROT),
        WORD UNSIGNED REFERENCE NAMED FACCOD IN DEFAULT 0,      {/* TYPE(NUMBER),
        LONGWORD UNSIGNED VALUE NAMED FLAGS IN DEFAULT 0,       {/* TYPE(MASK),
        LONGWORD UNSIGNED REFERENCE NAMED OBJUIC IN DEFAULT 0,  {/* TYPE(USERID),
        CHARACTER DESCRIPTOR NAMED SESSID IN DEFAULT 0, {/* TYPE(USERPARM),
        QUADWORD UNSIGNED REFERENCE NAMED IOSB OUT DEFAULT 0    {/* TYPE(IOSB)
        ) RETURNS LONGWORD;     {/* TYPE(CONDVALU);
```

```
/*
/* $DEASJNL
/*
/* DEASSIGN A CHANNEL TO A JOURNAL
/*
/*    CHAN      - NUMBER OF THE I/O CHANNEL TO BE DEASSIGNED.
/*    IOSB      - ADDRESS OF AN IOSB
/*

ENTRY CJF$DEASJNL ALIAS $DEASJNL PARAMETER (
        WORD UNSIGNED VALUE NAMED CHAN IN,      {/* TYPE(CHANNEL),
        QUADWORD UNSIGNED REFERENCE NAMED IOSB OUT DEFAULT 0    {/* TYPE(IOSB)
        ) RETURNS LONGWORD;     {/* TYPE(CONDVALU);


/*
/* $DEALJDR
/*
/* DEALLOCATE A JOURNAL DISK OR TAPE DRIVE.
/*
/*    DEVNAM    - ADDRESS OF A CHARACTER STRING DESCRIPTOR FOR THE DEVICE
/*                NAME STRING.
/*

ENTRY CJF$DEALJDR ALIAS $DEALJDR PARAMETER (
        CHARACTER DESCRIPTOR NAMED DEVNAM IN    {/* TYPE(DEVNAME)
        ) RETURNS LONGWORD;     {/* TYPE(CONDVALU);


/*
/* $DELJNL
/*
/* DELETE A JOURNAL
/*
/*     CHAN     - NUMBER OF THE I/O CHANNEL TO THE JOURNAL TO BE DELETED.
/*    [FLAGS]   - FLAGS FOR CHANNEL.
/*

ENTRY CJF$DELJNL ALIAS $DELJNL PARAMETER (
        WORD UNSIGNED VALUE NAMED CHAN IN,      {/* TYPE(CHANNEL),
        LONGWORD UNSIGNED VALUE NAMED FLAGS IN DEFAULT 0        {/* TYPE(MASK)
        ) RETURNS LONGWORD;     {/* TYPE(CONDVALU);



/*
/* $DMTJMD AND $DMTJMDW
/*
/* DISMOUNT A JOURNAL MEDIUM
/*
/*    [DEVNAM]  - ADDRESS OF A CHARACTER STRING DESCRIPTOR FOR THE DEVICE
/*                NAME STRING.
/*    [GRPNAM]  - ADDRESS OF A CHARACTER STRING DESCRIPTOR FOR THE TAPE
```

```
/*                  SHADOW GROUP.
/*     [FLAGS]   - 32 BIT VALUE OF FLAGS.
/*     [EFN]     - NUMBER OF EVENT FLAG TO BE SET UPON COMPLETION.
/*     [IOSB]    - ADDRESS OF QUADWORD I/O STATUS BLOCK THAT IS TO RECEIVE
/*                 FINAL COMPLETION STATUS.
/*     [ASTADR]  - ADDRESS OF THE ENTRY MASK OF AN AST SERVICE ROUTINE TO BE
/*                 EXECUTED WHEN THE DISMOUNT COMPLETES.
/*     [ASTPRM]  - AST PARAMETER LONGWORD TO BE PASSED TO THE AST ROUTINE.
/*

ENTRY CJF$DMTJMD ALIAS $DMTJMD PARAMETER (
        CHARACTER DESCRIPTOR NAMED DEVNAM IN DEFAULT 0, {/* TYPE(DEVNAME),
        CHARACTER DESCRIPTOR NAMED GRPNAM IN DEFAULT 0, {/* TYPE(CHARDESC),
        LONGWORD UNSIGNED REFERENCE NAMED FLAGS IN DEFAULT 0,   {/* TYPE(MASK),
        LONGWORD UNSIGNED VALUE NAMED EFN IN DEFAULT 0, {/* TYPE(EFNUM),
        QUADWORD UNSIGNED REFERENCE NAMED IOSB IN DEFAULT 0,    {/* TYPE(IOSB),
        ADDRESS(ENTRY) NAMED ASTADR DEFAULT 0,  {/* TYPE(ASTADR),
        LONGWORD UNSIGNED VALUE NAMED ASTPRM IN DEFAULT 0       {/* TYPE(USERPARM)
        ) RETURNS LONGWORD;     {/* TYPE(CONDVALU);

ENTRY CJF$DMTJMDW ALIAS $DMTJMDW PARAMETER (
        CHARACTER DESCRIPTOR NAMED DEVNAM IN DEFAULT 0, {/* TYPE(DEVNAME),
        CHARACTER DESCRIPTOR NAMED GRPNAM IN DEFAULT 0, {/* TYPE(CHARDESC),
        LONGWORD UNSIGNED REFERENCE NAMED FLAGS IN DEFAULT 0,   {/* TYPE(MASK),
        LONGWORD UNSIGNED VALUE NAMED EFN IN DEFAULT 0, {/* TYPE(EFNUM),
        QUADWORD UNSIGNED REFERENCE NAMED IOSB IN DEFAULT 0,    {/* TYPE(IOSB),
        ADDRESS(ENTRY) NAMED ASTADR DEFAULT 0,  {/* TYPE(ASTADR),
        LONGWORD UNSIGNED VALUE NAMED ASTPRM IN DEFAULT 0       {/* TYPE(USERPARM)
        ) RETURNS LONGWORD;     {/* TYPE(CONDVALU);



/*
/* $FORCEJNL AND $FORCEJNLW
/*
/* FORCE THE JOURNAL'S CURRENT BUFFERS OUT TO SECONDARY STORAGE
/*
/*      CHAN      - NUMBER OF A CHANNEL TO THE JOURNAL.
/*     [SEQNO]    - HIGHEST SEQUENCE NUMBER TO FORCE TO.
/*     [EFN]      - NUMBER OF EVENT FLAG TO BE SET UPON COMPLETION.
/*     [IOSB]     - ADDRESS OF QUADWORD I/O STATUS BLOCK THAT IS TO RECEIVE
/*                  FINAL COMPLETION STATUS.
/*     [ASTADR]   - ADDRESS OF THE ENTRY MASK OF AN AST SERVICE ROUTINE TO BE
/*                  EXECUTED UPON COMPLETION.
/*     [ASTPRM]   - AST PARAMETER LONGWORD TO BE PASSED TO THE AST ROUTINE.
/*

ENTRY CJF$FORCEJNL ALIAS $FORCEJNL PARAMETER (
        WORD UNSIGNED VALUE NAMED CHAN IN,      {/* TYPE(CHANNEL),
        LONGWORD UNSIGNED VALUE NAMED SEQNO IN DEFAULT 0,       {/* TYPE(NUMBER),
        LONGWORD UNSIGNED VALUE NAMED EFN IN DEFAULT 0, {/* TYPE(EFNUM),
        QUADWORD UNSIGNED REFERENCE NAMED IOSB IN DEFAULT 0,    {/* TYPE(IOSB),
        ADDRESS(ENTRY) NAMED ASTADR DEFAULT 0,  {/* TYPE(ASTADR),
        LONGWORD UNSIGNED VALUE NAMED ASTPRM IN DEFAULT 0       {/* TYPE(USERPARM)
```

```
        ) RETURNS LONGWORD;     {/* TYPE(CONDVALU);


ENTRY CJF$FORCEJNLW ALIAS $FORCEJNLW PARAMETER (
        WORD UNSIGNED VALUE NAMED CHAN IN,      {/* TYPE(CHANNEL),
        LONGWORD UNSIGNED VALUE NAMED SEQNO IN DEFAULT 0,       {/* TYPE(NUMBER),
        LONGWORD UNSIGNED VALUE NAMED EFN IN DEFAULT 0, {/* TYPE(EFNUM),
        QUADWORD UNSIGNED REFERENCE NAMED IOSB IN DEFAULT 0,    {/* TYPE(IOSB),
        ADDRESS(ENTRY) NAMED ASTADR DEFAULT 0,  {/* TYPE(ASTADR),
        LONGWORD UNSIGNED VALUE NAMED ASTPRM IN DEFAULT 0       {/* TYPE(USERPARM)
        ) RETURNS LONGWORD;     {/* TYPE(CONDVALU);



/*
/* $GETJNL
/*
/* GET THE DEFAULT JOURNAL NAME FOR A DEVICE-UIC PAIR
/*
/*      DEVNAM    - ADDRESS OF A CHARACTER STRING DESCRIPTOR FOR THE DEVICE
/*                  NAME STRING.
/*      UIC       - UIC OF THE OWNER OF THE OBJECT TO BE JOURNALED.
/*      JNLTYP    - TYPE OF JOURNAL.
/*      JNLNAM    - ADDRESS OF DESRCIPTOR TO RECEIVE THE JOURNAL NAME.
/*     [RSLLEN]   - ADDRESS OF A WORD TO RECEIVE THE LENGTH OF THE RESULTANT
/*                  NAME STRING.
/*

ENTRY CJF$GETJNL ALIAS $GETJNL PARAMETER (
        CHARACTER DESCRIPTOR NAMED DEVNAM IN,   {/* TYPE(DEVNAME),
        LONGWORD UNSIGNED VALUE NAMED UIC IN,   {/* TYPE(USERID),
        BYTE UNSIGNED VALUE NAMED JNLTYP IN,    {/* TYPE(NUMBER),
        CHARACTER DESCRIPTOR NAMED JNLNAM IN,   {/* TYPE(CHARDESC),
        WORD UNSIGNED REFERENCE NAMED RSLLEN IN DEFAULT 0       {/* TYPE(NUMBER)
        ) RETURNS LONGWORD;     {/* TYPE(CONDVALU);


/*
/* $GETRUI
/*
/* GET RECOVERY UNIT INFORMATION
/*                 Note: One of [RUID,DEVNAM] must be specified, but for SDL's
/*                       purposes they must both be considered optional
/*     [RUID]     - ADDRESS OF AN OCTAWORD RECOVERY UNIT ID.
/*     [DEVNAM]   - ADDRESS OF A CHARACTER STRING DESCRIPTOR FOR THE DEVICE
/*                  NAME STRING.
/*      RSLBUF    - ADDRESS OF A CHARACTER STRING DESCRIPTOR FOR THE BUFFER
/*                  TO RECEIVE THE RU STRUCTURES (RUS's).
/*     [RSLCNT]   - ADDRESS OF A WORD TO RECEIVE THE NUMBER OF RUS's RETURNED.
/*     [FLAGS]    - 32 BIT VALUE OF FLAGS.
/*

ENTRY CJF$GETRUI ALIAS $GETRUI PARAMETER (
        ANY NAMED RUID IN DEFAULT 0,    {/* TYPE(CNTRLBLK),
        CHARACTER DESCRIPTOR NAMED DEVNAM IN DEFAULT 0, {/* TYPE(DEVNAME),
```

```
        CHARACTER DESCRIPTOR NAMED RSLBUF OUT,  {/* TYPE(CHARDESC),
        WORD UNSIGNED REFERENCE NAMED RSLCNT OUT DEFAULT 0,    {/* TYPE(NUMBER),
        LONGWORD UNSIGNED REFERENCE NAMED FLAGS IN DEFAULT 0   {/* TYPE(MASK)
        ) RETURNS LONGWORD;     {/* TYPE(CONDVALU);



/*
/* $MNTJMD
/*
/* MOUNT A JOURNAL DISK OR TAPE MEDIUM
/*
/*      ITMLST    - ADDRESS OF A ZERO-TERMINATED LIST OF INPUT PARAMETERS.
/*     [JNLACP]   - ADDRESS OF A DESCRIPTOR FOR A JOURNAL ACP PROCESS NAME.
/*     [SGB]      - ADDRESS OF A ZERO-TERMINATED SHADOW GROUP BLOCK.
/*

ENTRY CJF$MNTJMD ALIAS $MNTJMD PARAMETER (
        ANY NAMED ITMLST IN,    {/* TYPE(ITEMLIST),
        CHARACTER DESCRIPTOR NAMED JNLACP IN DEFAULT 0, {/* TYPE(PROCNAME),
        ANY NAMED SGB IN DEFAULT 0      {/* TYPE(ITEMLIST)
        ) RETURNS LONGWORD;     {/* TYPE(CONDVALU);


/*
/* $MODFLT AND $MODFLTW
/*
/* MODIFY A FILTER
/*
/*      CHAN      - NUMBER OF A CHANNEL TO THE JOURNAL.
/*      FLTLST    - ADDRESS OF A ZERO-TERMINATED LIST OF POINTERS TO FILTERS.
/*      FLAGS     - 32 BIT VALUE OF FLAGS.
/*     [EFN]      - NUMBER OF EVENT FLAG TO BE SET UPON COMPLETION.
/*     [IOSB]     - ADDRESS OF QUADWORD I/O STATUS BLOCK THAT IS TO RECEIVE
/*                  FINAL COMPLETION STATUS.
/*     [ASTADR]   - ADDRESS OF THE ENTRY MASK OF AN AST SERVICE ROUTINE TO BE
/*                  EXECUTED UPON COMPLETION.
/*     [ASTPRM]   - AST PARAMETER LONGWORD TO BE PASSED TO THE AST ROUTINE.
/*

ENTRY CJF$MODFLT ALIAS $MODFLT PARAMETER (
        WORD UNSIGNED VALUE NAMED CHAN IN,      {/* TYPE(CHANNEL),
        ANY NAMED FLTLST IN,    {/* TYPE(LIST),
        LONGWORD UNSIGNED VALUE NAMED FLAGS IN DEFAULT 0,       {/* TYPE(MASK),
        LONGWORD UNSIGNED VALUE NAMED EFN IN DEFAULT 0, {/* TYPE(EFNUM),
        QUADWORD UNSIGNED REFERENCE NAMED IOSB IN DEFAULT 0,    {/* TYPE(IOSB),
        ADDRESS(ENTRY) NAMED ASTADR DEFAULT 0,  {/* TYPE(ASTADR),
        LONGWORD UNSIGNED VALUE NAMED ASTPRM IN DEFAULT 0       {/* TYPE(USERPARM)
        ) RETURNS LONGWORD;     {/* TYPE(CONDVALU);

ENTRY CJF$MODFLTW ALIAS $MODFLTW PARAMETER (
        WORD UNSIGNED VALUE NAMED CHAN IN,      {/* TYPE(CHANNEL),
        ANY NAMED FLTLST IN,    {/* TYPE(LIST),
        LONGWORD UNSIGNED VALUE NAMED FLAGS IN DEFAULT 0,       {/* TYPE(MASK),
        LONGWORD UNSIGNED VALUE NAMED EFN IN DEFAULT 0, {/* TYPE(EFNUM),
```

```
        QUADWORD UNSIGNED REFERENCE NAMED IOSB IN DEFAULT 0,     {/* TYPE(IOSB),
        ADDRESS(ENTRY) NAMED ASTADR DEFAULT 0,  {/* TYPE(ASTADR),
        LONGWORD UNSIGNED VALUE NAMED ASTPRM IN DEFAULT 0        {/* TYPE(USERPARM)
        ) RETURNS LONGWORD;     {/* TYPE(CONDVALU);


/*
/* $POSJNL AND $POSJNLW
/*
/* POSITION A FILTER PRIOR TO READING
/*
/*      CHAN      - NUMBER OF A CHANNEL TO THE JOURNAL.
/*      FLTLST    - ADDRESS OF A ZERO-TERMINATED LIST OF POINTERS TO FILTERS.
/*     [FLAGS]    - 32 BIT VALUE OF FLAGS.
/*     [EFN]      - NUMBER OF EVENT FLAG TO BE SET UPON COMPLETION.
/*     [IOSB]     - ADDRESS OF QUADWORD I/O STATUS BLOCK THAT IS TO RECEIVE
/*                  FINAL COMPLETION STATUS.
/*     [ASTADR]   - ADDRESS OF THE ENTRY MASK OF AN AST SERVICE ROUTINE TO BE
/*                  EXECUTED UPON COMPLETION.
/*     [ASTPRM]   - AST PARAMETER LONGWORD TO BE PASSED TO THE AST ROUTINE.
/*

ENTRY CJF$POSJNL ALIAS $POSJNL PARAMETER (
        WORD UNSIGNED VALUE NAMED CHAN IN,       {/* TYPE(CHANNEL),
        ANY NAMED FLTLST IN,     {/* TYPE(LIST),
        LONGWORD UNSIGNED VALUE NAMED FLAGS IN DEFAULT 0,        {/* TYPE(MASK),
        LONGWORD UNSIGNED VALUE NAMED EFN IN DEFAULT 0, {/* TYPE(EFNUM),
        QUADWORD UNSIGNED REFERENCE NAMED IOSB IN DEFAULT 0,     {/* TYPE(IOSB),
        ADDRESS(ENTRY) NAMED ASTADR DEFAULT 0,  {/* TYPE(ASTADR),
        LONGWORD UNSIGNED VALUE NAMED ASTPRM IN DEFAULT 0        {/* TYPE(USERPARM)
        ) RETURNS LONGWORD;     {/* TYPE(CONDVALU);

ENTRY CJF$POSJNLW ALIAS $POSJNLW PARAMETER (
        WORD UNSIGNED VALUE NAMED CHAN IN,       {/* TYPE(CHANNEL),
        ANY NAMED FLTLST IN,     {/* TYPE(LIST),
        LONGWORD UNSIGNED VALUE NAMED FLAGS IN DEFAULT 0,        {/* TYPE(MASK),
        LONGWORD UNSIGNED VALUE NAMED EFN IN DEFAULT 0, {/* TYPE(EFNUM),
        QUADWORD UNSIGNED REFERENCE NAMED IOSB IN DEFAULT 0,     {/* TYPE(IOSB),
        ADDRESS(ENTRY) NAMED ASTADR DEFAULT 0,  {/* TYPE(ASTADR),
        LONGWORD UNSIGNED VALUE NAMED ASTPRM IN DEFAULT 0        {/* TYPE(USERPARM)
        ) RETURNS LONGWORD;     {/* TYPE(CONDVALU);


/*
/* $READJNL AND $READJNLW
/*
/* READ THE NEXT ENTRY FROM A JOURNAL
/*
/*      CHAN      - NUMBER OF A CHANNEL TO THE JOURNAL.
/*     [RSLBUF]   - ADDRESS OF DESCRIPTOR OF RESULT BUFFER.
/*     [ATRLST]   - ADDRESS OF A ZERO-TERMINATED LIST OF ATTRIBUTE DESCRIPTORS.
/*     [EFN]      - NUMBER OF EVENT FLAG TO BE SET UPON COMPLETION.
/*     [IOSB]     - ADDRESS OF QUADWORD I/O STATUS BLOCK THAT IS TO RECEIVE
/*                  FINAL COMPLETION STATUS.
```

```
/*     [ASTADR]  - ADDRESS OF THE ENTRY MASK OF AN AST SERVICE ROUTINE TO BE
/*                 EXECUTED UPON COMPLETION.
/*     [ASTPRM]  - AST PARAMETER LONGWORD TO BE PASSED TO THE AST ROUTINE.
/*

ENTRY CJF$READJNL ALIAS $READJNL PARAMETER (
        WORD UNSIGNED VALUE NAMED CHAN IN,      {/* TYPE(CHANNEL),
        CHARACTER DESCRIPTOR NAMED RSLBUF IN DEFAULT 0, {/* TYPE(CHARDESC),
        ANY NAMED ATRLST IN DEFAULT 0,  {/* TYPE(LIST),
        LONGWORD UNSIGNED VALUE NAMED EFN IN DEFAULT 0, {/* TYPE(EFNUM),
        QUADWORD UNSIGNED REFERENCE NAMED IOSB IN DEFAULT 0,    {/* TYPE(IOSB),
        ADDRESS(ENTRY) NAMED ASTADR DEFAULT 0,  {/* TYPE(ASTADR),
        LONGWORD UNSIGNED VALUE NAMED ASTPRM IN DEFAULT 0       {/* TYPE(USERPARM)
        ) RETURNS LONGWORD;     {/* TYPE(CONDVALU);


ENTRY CJF$READJNLW ALIAS $READJNLW PARAMETER (
        WORD UNSIGNED VALUE NAMED CHAN IN,      {/* TYPE(CHANNEL),
        CHARACTER DESCRIPTOR NAMED RSLBUF IN DEFAULT 0, {/* TYPE(CHARDESC),
        ANY NAMED ATRLST IN DEFAULT 0,  {/* TYPE(LIST),
        LONGWORD UNSIGNED VALUE NAMED EFN IN DEFAULT 0, {/* TYPE(EFNUM),
        QUADWORD UNSIGNED REFERENCE NAMED IOSB IN DEFAULT 0,    {/* TYPE(IOSB),
        ADDRESS(ENTRY) NAMED ASTADR DEFAULT 0,  {/* TYPE(ASTADR),
        LONGWORD UNSIGNED VALUE NAMED ASTPRM IN DEFAULT 0       {/* TYPE(USERPARM)
        ) RETURNS LONGWORD;     {/* TYPE(CONDVALU);


/*
/* $RECOVER AND $RECOVERW
/*
/* PERFORM A GENERAL ROLLBACK OR ROLLFORWARD RECOVERY OPERATION
/*
/*      FUNC      - TYPE OF RECOVERY TO PERFORM.
/*      OBJECT    - ADDRESS OF A ZERO-TERMINATED LIST OF RECOVERY OBJECT DESCRIPTORS.
/*     [FLTLST]   - ADDRESS OF A ZERO-TERMINATED LIST OF POINTERS TO FILTERS.
/*     [ACMODE]   - ADDRESS OF A BYTE CONTAINING THE ACMODE TO BE USED.
/*     [EFN]      - NUMBER OF EVENT FLAG TO BE SET UPON COMPLETION.
/*     [IOSB]     - ADDRESS OF QUADWORD I/O STATUS BLOCK THAT IS TO RECEIVE
/*                  FINAL COMPLETION STATUS.
/*     [ASTADR]   - ADDRESS OF THE ENTRY MASK OF AN AST SERVICE ROUTINE TO BE
/*                  EXECUTED UPON COMPLETION.
/*     [ASTPRM]   - AST PARAMETER LONGWORD TO BE PASSED TO THE AST ROUTINE.
/*     [LOGMBX]   - ADDRESS OF A STRING DESCRIPTOR POINTING TO THE LOGICAL
/*                  NAME OF THE MAILBOX TO RECEIVE LOG MESSAGES FROM THE RCP
/*

ENTRY CJF$RECOVER ALIAS $RECOVER PARAMETER (
        WORD UNSIGNED VALUE NAMED FUNC IN,      {/* TYPE(FUNCCODE),
        ANY NAMED OBJECT IN,    {/* TYPE(LIST),
        ANY NAMED FLTLST IN DEFAULT 0,  {/* TYPE(LIST),
        BYTE UNSIGNED REFERENCE NAMED ACMODE IN DEFAULT 0,      {/* TYPE(ACMODE),
        LONGWORD UNSIGNED VALUE NAMED EFN IN DEFAULT 0, {/* TYPE(EFNUM),
        QUADWORD UNSIGNED REFERENCE NAMED IOSB IN DEFAULT 0,    {/* TYPE(IOSB),
        ADDRESS(ENTRY) NAMED ASTADR DEFAULT 0,  {/* TYPE(ASTADR),
        LONGWORD UNSIGNED VALUE NAMED ASTPRM IN DEFAULT 0,      {/* TYPE(USERPARM),
        CHARACTER DESCRIPTOR NAMED LOGMBX IN DEFAULT 0  {/* TYPE(LOGNAME)
```

```
        ) RETURNS LONGWORD;     {/* TYPE(CONDVALU);


ENTRY CJF$RECOVERW ALIAS $RECOVERW PARAMETER (
        WORD UNSIGNED VALUE NAMED FUNC IN,      {/* TYPE(FUNCCODE),
        ANY NAMED OBJECT IN,    {/* TYPE(LIST),
        ANY NAMED FLTLST IN DEFAULT 0,  {/* TYPE(LIST),
        BYTE UNSIGNED REFERENCE NAMED ACMODE IN DEFAULT 0,      {/* TYPE(ACMODE),
        LONGWORD UNSIGNED VALUE NAMED EFN IN DEFAULT 0, {/* TYPE(EFNUM),
        QUADWORD UNSIGNED REFERENCE NAMED IOSB IN DEFAULT 0,    {/* TYPE(IOSB),
        ADDRESS(ENTRY) NAMED ASTADR DEFAULT 0,  {/* TYPE(ASTADR),
        LONGWORD UNSIGNED VALUE NAMED ASTPRM IN DEFAULT 0,      {/* TYPE(USERPARM),
        CHARACTER DESCRIPTOR NAMED LOGMBX IN DEFAULT 0  {/* TYPE(LOGNAME)
        ) RETURNS LONGWORD;     {/* TYPE(CONDVALU);


/*
/* $WRITEJNL AND $WRITEJNLW
/*
/* WRITE AN ENTRY TO THE JOURNAL
/*
/*      CHAN      - NUMBER OF A CHANNEL TO THE JOURNAL.
/*      WRTBUF    - ADDRESS OF DESCRIPTOR OF THE BUFFER HOLDING THE ENTRY TO BE
/*                  WRITTEN.
/*     [MODIF]    - MODIFIERS AFFECTING THE WRITE OPERATION.
/*     [ENTMSK]   - KEY WHICH PERMITS WRITING THIS ENTRY ONLY IF CORRESPONDING
/*                  BITS ARE SET IN THE JOURNAL'S MASK.
/*     [ENTATR]   - VALUE WRITTEN TO THE JOURNAL AS AN ENTRY ATTRIBUTE.
/*     [EFN]      - NUMBER OF EVENT FLAG TO BE SET UPON COMPLETION.
/*     [IOSB]     - ADDRESS OF QUADWORD I/O STATUS BLOCK THAT IS TO RECEIVE
/*                  FINAL COMPLETION STATUS.
/*     [ASTADR]   - ADDRESS OF THE ENTRY MASK OF AN AST SERVICE ROUTINE TO BE
/*                  EXECUTED UPON COMPLETION.
/*     [ASTPRM]   - AST PARAMETER LONGWORD TO BE PASSED TO THE AST ROUTINE.
/*

ENTRY CJF$WRITEJNL ALIAS $WRITEJNL PARAMETER (
        WORD UNSIGNED VALUE NAMED CHAN IN,      {/* TYPE(CHANNEL),
        CHARACTER DESCRIPTOR NAMED WRTBUF IN,   {/* TYPE(CHARDESC),
        LONGWORD UNSIGNED VALUE NAMED MODIF IN DEFAULT 0,       {/* TYPE(MASK),
        LONGWORD UNSIGNED VALUE NAMED ENTMSK IN DEFAULT 0,      {/* TYPE(MASK),
        LONGWORD UNSIGNED VALUE NAMED ENTATR IN DEFAULT 0,      {/* TYPE(MASK),
        LONGWORD UNSIGNED VALUE NAMED EFN IN DEFAULT 0, {/* TYPE(EFNUM),
        QUADWORD UNSIGNED REFERENCE NAMED IOSB IN DEFAULT 0,    {/* TYPE(IOSB),
        ADDRESS(ENTRY) NAMED ASTADR DEFAULT 0,  {/* TYPE(ASTADR),
        LONGWORD UNSIGNED VALUE NAMED ASTPRM IN DEFAULT 0       {/* TYPE(USERPARM)
        ) RETURNS LONGWORD;     {/* TYPE(CONDVALU);


ENTRY CJF$WRITEJNLW ALIAS $WRITEJNLW PARAMETER (
        WORD UNSIGNED VALUE NAMED CHAN IN,      {/* TYPE(CHANNEL),
        CHARACTER DESCRIPTOR NAMED WRTBUF IN,   {/* TYPE(CHARDESC),
        LONGWORD UNSIGNED VALUE NAMED MODIF IN DEFAULT 0,       {/* TYPE(MASK),
```

```
        LONGWORD UNSIGNED VALUE NAMED ENTMSK IN DEFAULT 0,      {/* TYPE(MASK),
        LONGWORD UNSIGNED VALUE NAMED ENTATR IN DEFAULT 0,      {/* TYPE(MASK),
        LONGWORD UNSIGNED VALUE NAMED EFN IN DEFAULT 0, {/* TYPE(EFNUM),
        QUADWORD UNSIGNED REFERENCE NAMED IOSB IN DEFAULT 0,    {/* TYPE(IOSB),
        ADDRESS(ENTRY) NAMED ASTADR DEFAULT 0,  {/* TYPE(ASTADR),
        LONGWORD UNSIGNED VALUE NAMED ASTPRM IN DEFAULT 0       {/* TYPE(USERPARM)
        ) RETURNS LONGWORD;     {/* TYPE(CONDVALU);


/*
/* $CRENWV
/*
/* CREATE A NEW JOURNAL FILE VERSION
/*
/*      CHAN      - ADDRESS OF A WORD TO RECEIVE THE CHANNEL NUMBER ASSIGNED.
/*      LSTADR    - ADDRESS OF A LIST OF ITEM LIST ADDRESSES.
/*     [WRTBUF]   - ADDRESS OF DESCRIPTOR FOR A BUFFER CONTAINING THE LAST
/*                  ENTRY TO BE WRITTEN TO THE CURRENT VERSION(S)
/*     [IOSB]     - ADDRESS OF AN IOSB FOR THE WRITE OF THE LAST ENTRY
/*

ENTRY CJF$CRENWV ALIAS $CRENWV PARAMETER (
        WORD UNSIGNED VALUE NAMED CHAN IN,      {/* TYPE(CHANNEL),
        ANY NAMED LSTADR IN,    {/* TYPE(LIST),
        CHARACTER DESCRIPTOR NAMED WRTBUF IN DEFAULT 0, {/* TYPE(CHARDESC),
        QUADWORD UNSIGNED REFERENCE NAMED IOSB IN DEFAULT 0     {/* TYPE(IOSB)
        ) RETURNS LONGWORD;     {/* TYPE(CONDVALU);


/*
/* $CONJNLF
/*
/* CONNECT AN OLD VERSION OF THE JOURNAL FILE
/*
/*      CHAN      - ADDRESS OF A WORD TO RECEIVE THE CHANNEL NUMBER ASSIGNED.
/*      ITMLST    - ADDRESS OF THE ITEM LIST SPECIFYING THE OLD JOURNAL FILE VERSION
/*

ENTRY CJF$CONJNLF ALIAS $CONJNLF PARAMETER (
        WORD UNSIGNED VALUE NAMED CHAN IN,      {/* TYPE(CHANNEL),
        ANY NAMED ITMLST IN     {/* TYPE(ITEMLIST)
        ) RETURNS LONGWORD;     {/* TYPE(CONDVALU);


/*
/* $DCNJNLF
/*
/* DISCONNECT A (SET OF) OLD JOURNAL FILE VERSIONS
/*
/*      CHAN      - ADDRESS OF A WORD TO RECEIVE THE CHANNEL NUMBER ASSIGNED.
/*     [ITMLST]   - ADDRESS OF THE JOURNAL SPECIFICATION BLOCK.
/*
```

```
ENTRY CJF$DCNJNLF ALIAS $DCNJNLF PARAMETER (
        WORD UNSIGNED VALUE NAMED CHAN IN,      {/* TYPE(CHANNEL),
        ANY NAMED ITMLST IN DEFAULT 0   {/* TYPE(ITEMLIST),
        ) RETURNS LONGWORD;     {/* TYPE(CONDVALU);




/*
/* $GETCJI
/*
/* GET COMMON JOURNALING INFORMATION
/*
/*      CHAN     - NUMBER OF A CHANNEL TO THE JOURNAL.
/*      ITMLST   - ADDRESS OF A ZERO TERMINTATED ITEM LIST
/*

ENTRY CJF$GETCJI ALIAS $GETCJI PARAMETER (
        WORD UNSIGNED VALUE NAMED CHAN IN,      {/* TYPE(CHANNEL),
        ANY NAMED ITMLST IN     {/* TYPE(ITEMLIST)
        ) RETURNS LONGWORD;     {/* TYPE(CONDVALU);




/*
/* $CANCELRU
/*
/* CANCEL A RECOVERY UNIT
/*
/*

ENTRY RUF$CANCELRU ALIAS $CANCELRU
        RETURNS LONGWORD;       {/* TYPE(CONDVALU);


/*
/* $CANRUH
/*
/* CANCEL RECOVERY UNIT HANDLER
/*
/* HID  - HANDLER IDENTIFIER (AS GIVEN BY $DCLRUH)
/*

ENTRY RUF$CANRUH ALIAS $CANRUH PARAMETER (
        LONGWORD UNSIGNED VALUE NAMED HID IN    {/* TYPE(RETID)
        ) RETURNS LONGWORD;     {/* TYPE(CONDVALU);



/*
```

```
/* $DCLRUH
/*
/* DECLARE RECOVERY UNIT HANDLER
/*
/*      ADDR      - ADDRESS OF RECOVERY UNIT HANDLER
/*    [PARAM]     - VALUE OF THE PARAMETER PASSED TO THE HANDLER WHEN CALLED
/*    [HID]       - ADDRESS TO RECEIVE HANDLER ID ASSIGNED BY RUF
/*

ENTRY RUF$DCLRUH ALIAS $DCLRUH PARAMETER (
        ADDRESS(ENTRY) NAMED ADDR,      {/* TYPE(ENTRYADR),
        LONGWORD UNSIGNED VALUE NAMED PARAM IN DEFAULT 0,       {/* TYPE(USERPARM),
        LONGWORD UNSIGNED REFERENCE NAMED HID IN DEFAULT 0      {/* TYPE(RETID)
        ) RETURNS LONGWORD;     {/* TYPE(CONDVALU);



/*
/* $ENDRU
/*
/* SUCCESSFULLY COMPLETE A RECOVERY UNIT
/*
/*

ENTRY RUF$ENDRU ALIAS $ENDRU
        RETURNS LONGWORD;       {/* TYPE(CONDVALU);


/*
/* $MARKPOINTRU
/*
/* ESTABLISH INTERMEDIATE MARKPOINT FOR RECOVERY UNIT
/*
/*    [MARKID]  - ADDRESS OF LONGWORD TO RECEIVE MARK IDENTIFIER
/*

ENTRY RUF$MARKPOINTRU ALIAS $MARKPOINTRU PARAMETER (
        LONGWORD UNSIGNED REFERENCE NAMED MARKID IN DEFAULT 0   {/* TYPE(RETID)
        ) RETURNS LONGWORD;     {/* TYPE(CONDVALU);


/*
/* $PHASE1
/*
/* SUCCESSFULLY COMPLETE PHASE1 FOR A RECOVERY UNIT
/*
/*

ENTRY RUF$PHASE1 ALIAS $PHASE1
        RETURNS LONGWORD;       {/* TYPE(CONDVALU);


/*
```

```
/* $PHASE2
/*
/* SUCCESSFULLY COMPLETE PHASE1 FOR A RECOVERY UNIT
/*
/*

ENTRY RUF$PHASE2 ALIAS $PHASE2
        RETURNS LONGWORD;       {/* TYPE(CONDVALU);



/*
/* $RESETRU
/*
/* RESTORE RECOVERY UNIT STATE TO THE STATE OF GIVEN MARKPOINT
/*
/*     MARKID   - ADDRESS OF LONGWORD TO RECEIVE MARK IDENTIFIER
/*

ENTRY RUF$RESETRU ALIAS $RESETRU PARAMETER (
        LONGWORD UNSIGNED REFERENCE NAMED MARKID IN     {/* TYPE(RETID)
        ) RETURNS LONGWORD;     {/* TYPE(CONDVALU);



/*
/* $STARTRU
/*
/* START A RECOVERY UNIT
/*
/*     [RUFID]   - ADDRESS OF OCTAWORD TO RECEIVE RECOVERY UNIT IDENTIFIER
/*

ENTRY RUF$STARTRU ALIAS $STARTRU PARAMETER (
        LONGWORD UNSIGNED REFERENCE NAMED RUFID IN DEFAULT 0    {/* TYPE(RETID)
        ) RETURNS LONGWORD;     {/* TYPE(CONDVALU);



/*
/* $RUSTATUS
/*
/* GET THE STATUS OF A RECOVERY UNIT
/*
/*     [RUFID]   - ADDRESS OF OCTAWORD TO RECEIVE RECOVERY UNIT IDENTIFIER
/*

ENTRY RUF$RUSTATUS ALIAS $RUSTATUS PARAMETER (
        LONGWORD UNSIGNED REFERENCE NAMED RUFID IN DEFAULT 0    {/* TYPE(RETID)
        ) RETURNS LONGWORD;     {/* TYPE(CONDVALU);
```

```
END_MODULE;
```

JNLBUSR
R32

JNLDEFINT
SDL

CJFV4.

JNLPREFIX
R32

CJFRUFMAC
SDL

RUFUSR
SDL

UPGRADE
LIS

JNLFILE
SDL

BOPTIONS
R32

JNLDEF
SDL