

UNISYS

OS 1100

DATA Processor and ELT
Processor

**Operations
Reference Manual**

Copyright © 1988 Unisys Corporation.
All rights reserved.
Unisys is a registered trademark of Unisys Corporation.

Release SB3

November 1988

Priced Item

Printed in U S America
UP-13402

NO WARRANTIES OF ANY NATURE ARE EXTENDED BY THE DOCUMENT. Any product and related material disclosed herein are only furnished pursuant and subject to the terms and conditions of a duly executed Program Product License or Agreement to purchase or lease equipment. The only warranties made by Unisys, if any, with respect to the products described in this document are set forth in such License or Agreement. Unisys cannot accept any financial or other responsibility that may be the result of your use of the information in this document or software material, including direct, indirect, special, or consequential damages.

You should be very careful to ensure that the use of this information and/or software material complies with the laws, rules, and regulations of the jurisdictions with respect to which it is used.

The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

Correspondence regarding this publication should be forwarded using the Business Reply Mail form in this document, or remarks may be addressed directly to Unisys Corporation, PI Response Card, P.O. Box 64942, St. Paul, Minnesota, 55164-9749, U.S.A.

Contents

About This Manual	vii
Section 1. Description of the DATA and ELT Processors	
DATA Processor	1-1
ELT Processor	1-1
Section 2. Format and Examples for the DATA Processor	
DATA Processor Call Format and Options	2-1
Input Termination Sentinel (@END)	2-2
DATA Examples	2-3
Section 3. Format and Examples for the ELT Processor	
ELT Processor Call Format and Options	3-1
Input Termination Sentinel (@END)	3-3
ELT Examples	3-3
Appendix A. Symbolic Input/Output Routine (SIR\$) Options	
Bibliography	1

About This Manual

Purpose

This manual shows you how to use the DATA and ELT processors to create, update, and list system data format (SDF) files and elements from within the runstream.

Scope

The manual describes the DATA and ELT processors and provides examples using each processor.

Audience

This manual is written for programmers who use OS 1100.

Prerequisites

You must know how to use OS 1100 to use DATA and ELT.

How to Use This Manual

Read Section 2 to familiarize yourself with the format and options for the DATA processor. Read Section 3 to familiarize yourself with the format options for the ELT processor.

Organization

This manual is organized as follows:

Section 1. Description of the DATA and ELT Processors

This section briefly describes the DATA and ELT processors, and how they operate on the OS 1100.

Section 2. Format and Examples for the DATA Processor

This section describes the command format for the DATA processor, and provides examples for using the format.

Section 3. Format and Examples for the ELT Processor

This section describes the command format for the ELT processor, and provides examples for using the format.

Appendix A. Symbolic Input/Output Routine (SIR\$) Options

This appendix describes the symbolic input/output routine (SIR\$), the standard processor interface routine that OS 1100 processors use to manage symbolic input and output.

This appendix also lists the SIR\$ options that you can use with either the DATA or ELT processor commands.

Related Product Information

OS 1100 ELT Processor Installation Guide (UP-10704)

OS 1100 COMUS Operations Reference Manual (UP-9197)

OS 1100 System Service Routines Library Reference (UP-8728)

OS 1100 Exec System Software Executive Requests Programming Reference Manual (UP-4144)

OS 1100 Exec System Software Executive Control Language (ECL) Operations and Programming Reference Manual (UP-11563)

Copies of these documents are available through your Unisys representative. Check the *Series 1100 and 2200/200 Systems Product Documentation Library Directory* (UP-7893) for the version of the document that corresponds to the software level in use at your site.

Section 1

Description of the DATA and ELT Processors

DATA Processor

The DATA processor enables you to create (initially insert), modify (make changes), and list system data format (SDF) files.

ELT Processor

The ELT processor enables you to create (initially insert), modify (make changes), and list program file elements.

Section 2

Format and Examples for the DATA Processor

DATA Processor Call Format and Options

The @DATA control statement is used to call the DATA processor. It must precede the data or correction images. The DATA processor is terminated by the @END or @FTN control statement. For more information on these control statements, see the *Exec ECL End Use and Programming Reference Manual*.

If the @DATA control statement contains a sentinel, the terminating @END control statement must also contain a matching sentinel.

The DATA processor treats statements appearing between the @DATA and matching @END control statements as data. The processor, however, does not treat the @FIN, @ADD,D, and @PASSWD control statements as data. For more information on these control statements, see the *Exec ECL End Use and Programming Reference Manual*.

The only parameter required on the call is *filename-1*. The other parameters for the @DATA control statement are optional.

Format

@DATA, *options* *filename-1*, *filename-2*, *sentinel*

where:

options

are as defined in the following list. With the exception of the SIR\$ U option, source input/output routine (SIR\$) options work as described in Appendix A.

If you omit the I option, the processor interprets the data images following the control statement as corrections to *filename-1*. If you specify *filename-2*, the processor generates an updated file.

Note: *The tape operations of the DATA processor are no longer supported.*

E

If a slash (/) is encountered in column one of an image, the image is printed on a new page. If the L option is not present, this option is ignored.

L

Generates a complete listing of the file. This includes sequential item numbers used when making corrections to the file and identification of added and deleted images. If this option and *filename-1* are the only parameters you specified, *filename-1* is listed.

U

Used only to update from relative F-cycle zero to relative F-cycle +1. You can specify only *filename-1*. Relative F-cycle +1 of the file must be assigned prior to referencing it implicitly through the U option on the @DATA statement. See the *Exec ECL End Use and Programming Reference Manual* for information on making line corrections.

V

Lists both original and updated line numbers. The processor ignores this option if you do not specify the L option. Inserted and replacement lines have a plus sign (+) following the line number.

X

Takes error exit (ERRS) upon occurrence of an error. See the *Exec Executive Requests Programming Reference Manual* for information on ERRS.

filename-1

specifies the file to which the data images and correction images in the runstream apply. You must catalog or assign the file to the run.

filename-2

specifies the updated file to be generated. You must catalog or assign the file to the run.

sentinel

specifies a character code of 1 to 6 characters used for comparison purposes in determining the proper terminating @END statement for data mode.

Input Termination Sentinel (@END)

Use the @END control statement to terminate the DATA processor. The control statement follows the data images inserted by the @DATA control statement. If the @DATA statement contains a sentinel, the terminating @END control statement must contain a matching sentinel.

Format

```
@END sentinel
```

where *sentinel* is a 1- to 6-character code corresponding to the sentinel contained in the @DATA statement introducing the data images. The end of input to the DATA processor is determined when this parameter matches the sentinel parameter specified in the associated @DATA statement.

Example

```
@END FINISH
```

If you called the DATA processor with a sentinel of FINISH, then this @END control statement marks the end of the images read by the DATA processor. If you called the DATA processor with a different sentinel or without a sentinel, then the DATA processor treats this @END control statement as an input image, and the processor continues reading images from the runstream.

DATA Examples

The *Exec ECL End Use and Programming Reference Manual* contains procedures for making the line corrections in examples 2, 3, and 4.

- The images following this @DATA control statement are inserted into FILEA until terminated by the @END statement. The images are listed because of the L option.

Your Input

```
@DATA, IL FILEA.  
This information is inserted into file (FILEA)  
to demonstrate  
the input capability of the DATA Processor.  
@END
```

Processor Response

```
DATA 9R1 SL74T9 03/25/88 09:59:00 (->1)
      1.      THIS INFORMATION IS INSERTED INTO FILE (FILEA)
      2.      TO DEMONSTRATE
      3.      THE INPUT CAPABILITY OF THE DATA PROCESSOR.
END DATA.  ERRORS: NONE.  TIME:  0:450 SEC.  IMAGE COUNT: 3
```

- The change images following this @DATA control statement are applied as corrections to FILEA. The updated version of this file is stored into the newly created file (FILEB). FILEB is listed with changes applied. The input line numbers and updated line numbers are also listed because of the V option. FILEA remains unchanged.

Your Input

```
@DATA, LV   FILEA, FILEB.
-3, 3
the Update capability of the DATA Processor
to create a new file (FILEB).
@END
```

Processor Response

```
DATA 9R1 SL74T9 03/25/88 09:59:01 (1->1)
      1.      1 THIS INFORMATION IS INSERTED INTO FILE (FILEA)
      2.      2 TO DEMONSTRATE
-3, 3
      3.      1+THE UPDATE CAPABILITY OF THE DATA PROCESSOR
      4.      2+TO CREATE A NEW FILE (FILEB).
END DATA.  ERRORS: NONE.  TIME:  0.899 SEC.  IMAGE COUNT: 4
```

- The following example shows images following this @DATA control statement that are applied as corrections to FILEB. FILEB is listed, with changes applied; however, FILEB remains unchanged. The input line numbers are also listed because of the V option.

Your Input

```
@DATA, LV   FILEB.
-3, 4
the Update capability of the DATA Processor
to list the updated version but
not create a new file or destroy the old one.
@END
```

Processor Response

```
DATA 9R1 SL74T9 03/25/88 09:59:01 (1->1)
      1.      1 THIS INFORMATION IS INSERTED INTO FILE (FILEA)
      2.      2 TO DEMONSTRATE
-3, 4
      3.      1+THE UPDATE CAPABILITY OF THE DATA PROCESSOR
      4.      2+TO LIST THE UPDATED VERSION BUT
      5.      3+NOT CREATE A NEW FILE OR DESTROY THE OLD ONE.
END DATA.  ERRORS: NONE.  TIME: 0:452 SEC.  IMAGE COUNT: 5
```

- The changes following the @DATA control statement are applied to FILEA to create relative F-cycle +1 of file FILEA. The changes are listed because of the W option.

Your Input

```
@ASG,C FILEA(+1)
@DATA,UWLV FILEA.
-3, 3
```

This creates a new F-cycle of file (FILEA) with these corrections.
@END

Processor Response

```
DATA 9R1 SL74T9 03/25/88 09:59:02 (1->2)
-3, 3

THIS CREATES A NEW F-CYCLE OF FILE (FILEA)
WITH THESE CORRECTIONS.
      1. 1 THIS INFORMATION IS INSERTED INTO FILE (FILEA)
      2. 2 TO DEMONSTRATE
-3, 3
      3. 1+
      4. 2+THIS CREATES A NEW F-CYCLE OF FILE (FILEA)
      5. 3+WITH THESE CORRECTIONS.
END DATA.  ERRORS: NONE.  TIME: 0.966 SEC.  IMAGE COUNT: 5
```


Section 3

Format and Examples for the ELT Processor

ELT Processor Call Format and Options

Use the control statement @ELT to call the ELT processor. It must precede the element or correction images. Terminate the ELT processor by using any nontransparent control statement. The *Exec ECL End Use and Programming Reference Manual* contains information on control statements.

The *elname-1* parameter is required. The other parameters for the @ELT control statement are optional.

Format

@ELT, *options el tname-1, el tname-2, senti nel , , subtype*

where:

options

are defined in the following list. All source input/output routine (SIRS) options can be used (see Appendix A). The O and S options identify the element type, while the other options are image handling options. Use the D option to insert control statements into a symbolic element. Those elements identified as type S are symbolic elements. You can alter them by using the change statements.

The S options is assumed when the element type options are not specified.

Note: *Absolute (A) and Relocatable (R) element type options of the ELT processor are no longer supported.*

Element type options:

O

Inserts images following the @ELT statement into an omnibus element as they appear in the runstream. The element is not formatted by ELT. Use only with the I option.

S

Identifies element as a symbolic element. The ELT processor assumes this options whenever you do not specify an element type option. If O is not specified, S is the default. The S option is assumed if an element type option is omitted.

Image handling options:

D

Indicates that the symbolic input images following the @ELT statement can include control statements that are to be transferred as element data. For more information on the D option, see the subsection headed "Input Termination Sentinel."

E

Ejects a page whenever a slash (/) appears in column one.

L

Generates a listing of the complete symbolic element. The listing provides line numbers, cycle information, and identification of the newly added and deleted images.

V

Prints, for symbolic elements only, both the input and updated line numbers with the correction lines. Inserted or replaced lines will have a plus sign (+) following the inserted line number.

X

Takes error exit (ERRS) upon occurrence of an error. See the *Exec Executive Requests Programming Reference Manual* for information on ERRS.

eltname-1

specifies the input element. Use this parameter with the I option to specify the element you want to insert into the program file. Use this parameter to specify both the symbolic input and output elements if the U option is present and *eltname-1* is not present.

eltname-2

specifies the new output element you want to generate. Use the U option to retain all element cycle information (through cycle specified in *eltname-1*) in a new source output (SO) element.

sentinel

specifies with the D option the character code that terminates the flow of data images into the element you are creating. This parameter can consist of 1 to 6 characters and must agree exactly with the sentinel parameter that appears on the @END statement used to terminate the ELT processor. This parameter is ignored if you omit the D option.

., (blank field)

indicates a field reserved for future use; therefore, the two commas are required.

subtype

specifies the element subtype. The subtype can be either the octal value or the mnemonic. See the *SYSLIB Programming Reference Manual*.

Input Termination Sentinel (@END)

When you use the D option with the ELT control statement (@ELT,D), you must use an @END control statement to terminate the ELT processor. If the @ELT,D control statement contains a sentinel, the terminating @END control statement must contain a matching sentinel. Except for the @FIN, @ADD,D, and @PASSWD control statements, the ELT processor treats any statement appearing between the @ELT,D and the matching @END control statement as data. See the *Exec ECL End Use and Programming Reference Manual* for more information.

Format

```
@END sentinel
```

where *sentinel* is a 1- to 6-character code corresponding to the sentinel contained in the @ELT,D statement introducing the data images. The end of input to the ELT processor is determined when this parameter matches the sentinel parameter specified in the associated @ELT statement.

Example

```
@END STOP
```

If you called the ELT processor with a sentinel of STOP, then this @END control statement marks the end of the images read by the ELT processor. If you called the ELT processor with a different sentinel or without a sentinel, then the ELT processor treats this @END control statement as an input image, and the processor continues reading images from the runstream.

ELT Examples

The input in the following examples is terminated with @EOF. You can, however, use any nontransparent control statement.

The *Exec ECL End Use and Programming Reference Manual* includes procedures for making the line corrections in examples 2, 3, and 5.

- The images following this @ELT control statement are inserted as a new symbolic element (ELEMENT1) in program file PF1. The image transfer is terminated by the first nontransparent control statement encountered. Because of the L option, the element is listed. The S option is the default option; you can omit it.

Your Input

```
@ELT, ILS PF1. ELEMENT1
This information is inserted into element (ELEMENT1)
of Program File (PF1) and is used to demonstrate
element input using the ELT Processor.
@eof
```

Processor Response

```
ELT  8R2B (8871020 0900: 40) 1988 Mar 25 Fri 0958: 17
  1.          00  This information is inserted into element (ELEMENT1)
  2.          00  of Program File (PF1) and is used to demonstrate
  3.          00  element input using the ELT Processor.
END ELT.  ERRORS: NONE TIME: 0:149 SEC.  IMAGE COUNT: 3
```

- The change images following this @ELT control statement update ELEMENT1 of program file PF1. Because the element type is not specified, the processor assumes the S option and treats the element as a symbolic.

Your Input

```
@ELT, UL PF1. ELEMENT1
3, 3
the Update capability of the ELT Processor.
to create an updated cycle of element (ELEMENT1)
from Program File (PF1).
@eof
```

Processor Response

```
ELT 8R2B (8871020 0900:40) 1988 Mar 25 Fri 0958:17
  1.      00  This information is inserted into element (ELEMENT1)
  2.      00  of the Program File (PF1) and is used to demonstrate
  3.  NEW  01  the Update capability of the ELT Processor
  4.  NEW  01  to create an updated cyc of element (ELEMENT1)
  5.  NEW  01  from Program File (PF1).
      -1
END ELT. ERRORS: NONE TIME: 0.283 SEC. IMAGE COUNT: 5
```

- The change images following the @ELT control statement are applied to ELEMENT1 of program file (PF1) to produce a new symbolic element (ELEMENT2) in program file (PF2). ELEMENT1 of program file (PF1) remains unchanged.

Your Input

```
@ELT, LV PF1. ELEMENT1, PF2. ELEMENT2
3,4
the Update capability of the ELT Processor
to create a new element (ELEMENT2)
in Program File (PF2).
@eof
```

Processor Response

```
ELT 8R2B (8871020 0900:40) 1988 Mar 25 Fri 0958:18
  1.      1  00  This information is inserted into element (ELEMENT1)
  2.      2  00  of Program File (PF1) and is used to demonstrate
3,4
  3.     1+  00  the Update capability of the ELT Processor
  4.     2+  00  to create a new element (ELEMENT2)
  5.     3+  00  in Program File (PF2).
  6.     5  00  from Program File (PF2).
END ELT. ERRORS: NONE TIME: 0.300 SEC. IMAGE COUNT: 6
```

Format and Examples for the ELT Processor

- The images following this @ELT control statement, including other control statements, are inserted as a new symbolic element (ELEMENT2) in program file (PF1). The data stream is terminated when as @END statement having the matching sentinel STOP is encountered. Because the L option is omitted, the resulting element is not listed.

Your Input

```
@ELT, ID PF1. ELEMENT2, , STOP
This option allows the insertion of JCL into element (ELEMENT2)
of Program File (PF1) and is used to demonstrate the
@ELT Processor handling JCL until sentinel (STOP) occurs.
@ASG, A PF1.
@FREE PF1.
@END STOP
```

Processor Response

```
ELT      8R2B (8871020 0900:40) 1988 Mar 25 Fri 0958:18
END ELT.  ERRORS:  NONE TIME:  0.179 SEC. IMAGE COUNT: 5
```

- The change images following this @ELT control statement are applied to ELEMENT1 in program file (PF1) to produce the new symbolic element (ELEMENTB). ELEMENT1 remains unchanged. Previous element cycles are retained in ELEMENTB. ELEMENTB is given the subtype of MAP(06).

Your Input

```
@ELT, ULW PF1. ELEMENT1, PF1. ELEMENTB, , , 06
-4, 1
ELEMENT1 is updated to form element (ELEMENTB)
in Program File (PF1).
ELEMENTB is given the subtype of MAP.
@eof
```

Processor Response

```
ELT      8R2B (8871020 0900:40) 1988 Mar 25 Fri 0958:18
-4, 1
ELEMENT1 is updated to form element (ELEMENTB)
in Program File (PF1).
Element B is given the subtype of MAP.
  1.   1  00  This information is inserted into element (ELEMENT1)
  2.   2  00  of Program File (PF1) and is used to demonstrate
  3.   3  01  the Update capability of the ELT Processor.
-4, 1
  4.  1+  02  ELEMENT1 is updated to form element (ELEMENTB)
  5.  2+  02  in Program File (PF1).
  6.  3+  02  Element B is given the subtype of MAP.
END ELT.  ERRORS:  NONE TIME:  0.408 SEC. IMAGE COUNT: 6
```

- The images following this @ELT control statement are inserted as a new omnibus element (OMNI) in program file (PF1). The images are listed as they appear in the runstream.

Your Input

```
@ELT, IOL PF1. OMNI
This creates an element (OMNI)
in Program File (PF1).
1234567890
ABCDEFGHI J
abcdefghij
@eof
```

Processor Response

```
ELT      8R2B (8871020 0900:40) 1988 Mar 25 Fri 0958:19
This creates an element (OMNI)
in Program File (PF1).
1234567890
ABCDEFGHI J
abcdefghij
END ELT.      ERRORS:  NONE  TIME:  0.211 SEC.  IMAGE COUNT:  5
```


Appendix A

Symbolic Input/Output Routine (SIR\$)

Options

The symbolic input/output routine (SIR\$) is a standard processor interface routine that OS 1100 processors use to manage symbolic input and output. SIR\$ is a component of the System Service Library (SYSLIB), and is documented in the *SYSLIB Programming Reference Manual*. OS 1100 processors use SIR\$ to obtain the symbolic images from the runstream following the control statement, from a symbolic element in a program file, or from an SDF file. Depending on the options selected, the routine automatically applies changes, lists changes, and produces an updated SDF file or a symbolic element that is inserted into a program file.

You can use SIR\$ options with the DATA and ELT processors. The following list explains the options you can use on the processor call statement:

Symbolic Input/Output Routine (SIR\$) Options

G

Input is compressed symbolic in columns 1 through 80 of the symbolic image.
Applies with the I option only.

H

Input contains sequence numbers in columns 73 through 80 of the symbolic images.
Applies with the I option.

I

Reads images from the runstream and inserts them into a new symbolic.

J

Input contains compressed symbolic images in columns 1 through 72 of the images and sequence numbers in columns 73 through 80. These sequence numbers are not checked by the K option. Applies with the I option only.

K

Checks sequence numbers in columns 73 through 80. Valid only with the H and I options.

P

Outputs symbolic in Fielddata. ASCII images are converted to Fielddata. See Q option.

Q

Outputs symbolic in ASCII. Fielddata images are converted to ASCII. If you specify both P and Q options, the output is native mode, and no conversion takes place. If you specify neither the P nor Q options, the output is native mode and correction images, if any, are converted to the current source input type.

U

Reads change images from the runstream, applies them to the input symbolic element, and produces a new cycle of the input symbolic.

W

Lists change images.

Bibliography

OS 1100 COMUS Operations Reference Manual (UP-9197). Unisys Corporation.

OS 1100 ELT Processor Installation Guide (UP-10704). Unisys Corporation.

OS 1100 Exec System Software Executive Control Language (ECL) Operations and Programming Reference Manual (UP-11563). Unisys Corporation.

OS 1100 Exec System Software Executive Requests Programming Reference Manual (UP-4144). Unisys Corporation.

OS 1100 System Service Routines Library (SYSLIB) Reference (UP-8727). Unisys Corporation

Series 1100 and 2200/200 Systems Product Documentation Library Directory (UP-7893). Unisys Corporation.