

ALL-IN-1 Information Update
August 1991

Part Number AD-ND93B-B4

ALL-IN-1 Information Update

Published by
Corporate User Information Products (MRO1-3/T4)
Digital Equipment Corporation
P. O. Box 1001
Marlborough, MA 01752-9840

The *ALL-IN-1 Information Update* is a quarterly publication reporting on important ALL-IN-1 Software Performance Reports (SPRs), management, and tuning. It also describes workarounds and provides customers with information that may help them avoid problems. Much of the material is developed from SPR answers significant to the general audience and is printed here to supplement the maintenance updates.

Distribution

The *ALL-IN-1 Information Update* is directed to one software contact for each software product. No mailing will be made to addresses without a software contact name. **Address change requests should be sent to the nearest Digital field office. Include the new address and mailing label from the most recently received publication.**

The prerequisite to using Digital Software is the appropriate license. The standard Terms and Conditions and Digital Business Agreement (DBA) contain the license terms for all software other than DECsystem-10.

Susan A. Thompson, Editor

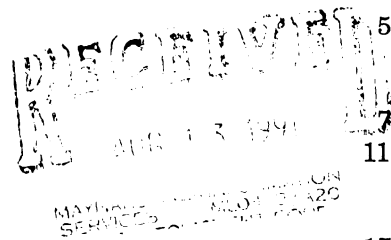
Copyright © Digital Equipment Corporation 1991. All Rights Reserved.

The material in this document is for informational purposes only. Digital believes the information in this publication is accurate as of its publication date; such information is subject to change without notice. Digital is not responsible for any inadvertent errors. Comments on the contents of this publication should be directed to your local Digital field office.

The following are trademarks of Digital Equipment Corporation: ALL-IN-1, CDA, DATATRIEVE, DDIF, DEClaser, DECnet, DECsystem, DECtrace, DECwindows, DECUS, DECUS logo, DTIF, FMS, ULTRIX, VAX, VAX Notes, VAXcluster, VMS, VT, WPS-PLUS, and the DIGITAL logo.

CONTENTS

	Page
INTRODUCTION	1
DOCUMENTATION	
Bookreader Versions of ALL-IN-1 Manuals	3
USING ALL-IN-1	
Hints for Users	5
APPLICATION PROGRAMMING	
Programming Hints: Working with Symbols	11
Programming Hints: Date and Time Fields	11
SYSTEM MANAGEMENT	
Testing and Repairing Users' File Cabinets	17
Cumulative Index	23
Attention U.S. Software Contract Customers Submitting Software Problems	33
Digital Equipment Computer Users Society (DECUS)	35
DECUS Subscription Service	39
Reader's Comments	41



INTRODUCTION

Welcome to the August 1991 issue of the *ALL-IN-1 Information Update*. The *ALL-IN-1 Information Update* is a quarterly publication that aims to keep you up-to-date with ALL-IN-1 and contains useful information for system managers, application programmers, and ALL-IN-1 users.

Some of the articles published in the *ALL-IN-1 Information Update* are developed from reports of problems that we have received from customers and are included to help you to avoid these problems. Other articles contain practical information about ALL-IN-1 features of which users might not be aware.

If you have any questions about ALL-IN-1 Versions 2.3 or 2.4 and are covered by a service contract, contact your local Digital Customer Support Center.

We are eager to provide you with helpful information. Therefore, please take the time to complete the Reader's Comments page at the back of this issue. This will help us get a better idea about the type of articles you would like to see in future issues.

DOCUMENTATION

Bookreader Versions of ALL-IN-1 Manuals

Most ALL-IN-1 manuals are now available in DECwindows Bookreader format and are distributed on compact disc read-only memory (CDROM) as part of the VMS Online Documentation Library. There are library *shelves* for ALL-IN-1 Management, Application Programming, and User Documentation. On the ALL-IN-1 Management shelf, a *Differences Between ALL-IN-1 V2.3 and ALL-IN-1 V2.4* provides a quick, easy reference for sites planning their upgrade or wishing to read up on the changes after their upgrade.

The following ALL-IN-1 Version 2.4 books are available in Bookreader form:

- ALL-IN-1 User's Reference*
- ALL-IN-1 Administration Guide*
- ALL-IN-1 Management Guide*
- ALL-IN-1 Mail Management Guide*
- ALL-IN-1 Application Programming: Getting Started*
- ALL-IN-1 Application Programming: Guide*
- ALL-IN-1 Application Programming: Reference—Volumes 1 and 2*
- ALL-IN-1 Application Programming: User Interface Standards*
- ALL-IN-1 Group Conferencing: User's Guide*
- ALL-IN-1 Group Conferencing: Installation and Management Guide*
- ALL-IN-1 Group Conferencing: Application Programming Reference*
- Differences Between ALL-IN-1 V2.3 and ALL-IN-1 V2.4*

ALL-IN-1 keyboard cards (ALL-IN-1 and WPS-PLUS key locations on VT100 and VT200 series keyboards) are also included.

Bookreader versions include updates and corrections that are not available in hard copy.

The *Disk User's Guide* in the Online Documentation Library tells you how to configure the CDROM and control access to it. It also lists the contents of the library. The entire library is accessible, even the documentation for software that the site does not have.

USING ALL-IN-1

Hints for Users

WPS-PLUS: Making GOLD GET Work More Quickly

When you incorporate text from one document into another document that you are writing, you use GOLD GET. By default, the lines of the text appear on the screen as they are incorporated. To merge the documents more quickly, you can switch the display feature off in the following way:

1. When you are writing or editing a WPS-PLUS document, press GOLD MENU.
2. When the Editor menu is displayed, enter EA. The Editor Attributes form appears.
3. Tab down to the GET DOGMT screen on field.
4. Enter N, since you do not want to see the document scroll as it is copied.
5. Press RETURN.
6. Press EXIT to return to your document.

The next time you use GOLD GET, the document will not be scrolled on the screen while it is being incorporated.

When you switch off the scrolling feature for GOLD GET in one document, it remains switched off in all documents until you change the setting again.

Reading Mail: Skipping Long Distribution Lists

When you read on the screen a mail message with a long distribution list, press GOLD DOWN ARROW once to move to the beginning of the message text.

Printing Mail: Omitting the Distribution Lists

When you want to print a message without the distribution list, use the File Text (FT) option to file the message without the mail header and then print the filed document.

If the message was created as an attachment (identified by a preceding dotted line and the word ATTACHMENT), use the File Attachment (FA) option to file the message only and then print the filed document.

Clearing Out Old Mail

Read and Outbox Folders

Documents in the Read and Outbox folders take up valuable disk space. Unlike the Wastebasket, the Read and Outbox folders are not emptied by the ALL-IN-1 housekeeping procedure. You should, therefore, regularly place unwanted Read messages in the Wastebasket by the Delete option and take the following steps to control the size of your Outbox folder:

- Check your Outbox regularly to find how many mail messages you created and the date you created them. To do this, enter IO from the Electronic Messaging (EM) menu to

select the index of your Outbox. Press GOLD BOTTOM to give the name and date of the oldest message at the bottom of the list.

- To delete all the messages in the Outbox, enter ALL and press RETURN to select all the messages. An X before each message shows the message is selected. Enter XD and all the selected messages are deleted.
- You may not want to delete all the messages. To keep the ones you want:
 1. Select all the messages using ALL.
 2. Use the UP or DOWN ARROW to move the cursor to a message you want to keep.
 3. Press GOLD SELECT. This removes the X prefix. Do this with all the messages you want to keep.
 4. Enter XD.
 5. Press RETURN.
- All the messages will be deleted except the ones you want to keep.

Other Folders

If, after reading mail, you regularly use the Refile document (RFD) option to file the Read messages into other folders, remember the messages remain on the disk, continuing to take up valuable space. Many of these refiled messages may be out-of-date or redundant. To find out where and when you refiled old mail messages, do the following:

- From the Electronic Messaging (EM) menu, enter Index (I).
- Move the cursor to the previous field and delete the name of the folder.
- Press TAB to move the cursor to the TO: field *after* the CREATION field.
- Enter a date in the TO: field, for example 10-JAN-90, for a list of all mail refiled before that date.
- Press RETURN.

The list gives you the date of each message and the name of the folder where it is refiled.

To delete any unwanted documents, use the procedure described above for the Read and Outbox folders.

Saving Paper When Printing

When you print a document, you can avoid extra flag pages and burst pages by altering your user setup. To do this, from the Main menu enter More menus (M). Then enter User setup (US) and Print options (PO) from the subsequent menus. Check your settings on your user setup. Make sure all the settings are set to N. This eliminates a number of unnecessary pages.

APPLICATION PROGRAMMING

Programming Hints: Working with Symbols

This article describes methods for doing the following:

- Adding leading zeros to numbers stored as symbols
- Right-justifying the contents of a symbol in a field or template

The methods described here use the techniques for specifying symbol substrings, described in the chapter on symbols in *Application Programming: Reference Volume 1*.

Adding Leading Zeros

You may want to add leading zeros to a number that is stored as a symbol, for example, a data set whose key contains a numerical value. This is because, if the numbers are of different lengths, ALL-IN-1 does not sort them in numerical order. Instead, it treats the numbers as strings and compares them character by character, starting from the left.

To ensure that symbols containing numbers to be loaded in a field of a form are of equal length, you can use one of the methods described here. Where a field is always entered by a user, you can use the FMS field attributes Right Justify and Zero Fill to ensure that the numbers contain the correct number of characters.

Method 1

This method involves the following steps:

1. Add a large number, such as a power of 10, to the number in the symbol. The number added should end with the same number of zeros as the number of characters that you want the symbol to contain. For example, if the symbol is to fill a 6-character field, add 1000000 to the number in the symbol.
2. Specify the length of the symbol substring to match the number of characters you want the symbol to contain and an offset starting position to remove the leading digit that you added in step 1.

For example, if the number in the symbol #NUMBER must fill a field that is six characters long, add 1000000, specify the length of the symbol substring as 6, and specify a starting position of 1 to remove the leading 1, as in the following script:

```
DECIMAL I
COMPUTE #TEMP = #NUMBER + 1000000
GET #NUMBER = #TEMP:6:1
```

You can modify this method to increment or decrement the value of the number held in the symbol. The following example increments the value of #NUMBER by 1 for each iteration of the script:

```
DECIMAL I
COMPUTE #TEMP = #NUMBER + 1000001
GET #NUMBER = #TEMP:6:1
```

APPLICATION PROGRAMMING

The following example decrements the value of #NUMBER by 1 for each iteration of the script:

```
DECIMAL I
COMPUTE #TEMP = #NUMBER + 999999
GET #NUMBER = #TEMP:6:1
```

Method 2

This method involves the following steps:

1. Reverse the number in the symbol. To do this, specify H as the symbol substring starting position.
2. Add a string of zeros to the end of the reversed symbol. Add at least as many zeros as the number of characters in the field. For example, if the number field is six characters, add at least six zeros.
3. Truncate the result of step 2 to the required length (six characters in the example).
4. Reverse the result of step 3.

The following script shows how to do this to add leading zeros to the number in #NUMBER so that it fills a 6-character field:

```
GET #TEMP = #NUMBER:H "000000"
GET #TEMP = #TEMP:6
GET #NUMBER = #TEMP:H
```

Right-Justifying the Contents of a Symbol

To ensure that the contents of a symbol are justified to the right of a field or a template, use one of the methods described here.

Method 1

To add leading spaces to a symbol, so that when you load a field or a template with the value of the symbol, the value is justified to the right of the field, you can use a method similar to method 2 for adding leading zeros.

For example, assume you have a field that is 10 characters long and you want to load it with the contents of a symbol #TEXT so that the output is justified to the right of the field. The following script adds an appropriate number of spaces to the left of the text:

```
GET #TEMP = #TEXT:H "      "
GET #TEMP = #TEMP:10
GET #TEXT = #TEMP:H
```

You can then use the WRITE function to enter the value of #TEXT in the field of the data set.

Method 2

An alternative method of right-justifying the contents of a symbol is to specify the field length as the length, and Z as the starting position of the symbol substring.

For example, assume you have a template in which you want to display a column of three figures whose values are held in symbols #A, #B, and #C. If the maximum value of each of these figures is 999999, specify a substring length of 6, as in the following example template:

```
Line 1: <#A:6:Z>  
Line 2: <#B:6:Z>  
Line 3: <#C:6:Z>
```

If #A has a value of 35199, #B a value of 23, and #C a value of 123785, the output of the template will be as follows:

```
Line 1: 35199  
Line 2: 23  
Line 3: 123785
```

Programming Hints: Date and Time Fields

This article provides information on how to program date and time fields within ALL-IN-1. In particular, it describes how to do the following:

- Create date and time fields
- Validate date and time fields
- Convert dates between various formats
- Compare date fields

Creating Date Fields

ALL-IN-1 allows you to enter dates in several different formats. Users specify their preferred date format using the Set Working Conditions form; this information is held in the user profile. Wherever possible, ALL-IN-1 displays dates in an alphanumeric format, such as 15-Aug-1991 or Aug-15-1991. To create a date field for an alphanumeric date, use the FMS editor to create an 11-character field.

NOTE

ALL-IN-1 provides its own date validation facilities as described in the next section. Therefore, do not use the facility provided by FMS to create predefined date fields, since ALL-IN-1 does not process these fields correctly.

Validating Date Fields

ALL-IN-1 provides several special symbols and data set access blocks (DSABs) that you can use to validate dates. These symbols and DSABs are described in *Application Programming: Reference Volume 1*. The following named data shows you how to use special symbols and validation DSABs to validate a date field, called BIRTH_DATE:

```
;;BIRTH_DATE;;  
/VALID=CAL$SET_DATE①:"BIRTH_DATE"②
```

- ① CAL\$SET_DATE is a validation DSAB that ensures a valid date was entered and sets the current calendar date.

The CAL\$SET_DATE DSAB allows you to enter dates in several different formats, depending on the preferred date format specified in the user profile. These formats include the following:

- DD/MMM/YYYY — 15-May-1991
- DD/MM/YY — 15/05/91
- MM/DD/YY — 05/15/91
- A keyword, such as TODAY, TOMORROW, and TODAY+1W

CAL\$SET_DATE also lets you compute dates by units of days, weeks, or months. The following example shows you how to compute the date of 7 days ago and put the result into a symbol #RESULT:

```
GET CAL$SET_DATE:"#RESULT".%WHOLE["TODAY -7D"]
```

APPLICATION PROGRAMMING

- ② This updates the field BIRTH_DATE with the validated date in the format specified in the user's working conditions.

Converting Date Formats

ALL-IN-1 allows you to convert dates from one format to another, by using the DATE_CONVERT function. DATE_CONVERT has the following syntax:

```
DATE_CONVERT dest-sym, source-sym, format key
```

For example, to convert the value of the current date (contained in the special symbol OA\$DATE) to month format and store the output value in the symbol #DATE, enter the following:

```
<DATE_CONVERT #DATE, OA$DATE, 6
```

If the value of OA\$DATE is 24/08/91, the value of #DATE is August.

You can also use keywords with DATE_CONVERT. The following example shows you how to compute the date of 7 days ago in National Bureau of Standards (NBS) format:

```
<DATE_CONVERT #RESULT, "TODAY -7D", 7
```

For a list of the valid parameters for DATE_CONVERT, refer to *Application Programming: Reference Volume 2*.

Comparing Dates

It is often useful to be able to compare dates. For example, in an employee registration application that stores the name (NAME), employee number (NUMBER), and joining date (START_DATE) of all employees, you may want to compare certain dates so you can create an index of all the people who joined the company between the specified search dates.

The obvious way to create an index of records is to do a direct comparison between the contents of the field. With date fields, however, a direct comparison does not produce chronologically accurate results. This inaccuracy is caused by the way ALL-IN-1 processes dates. When ALL-IN-1 compares dates, it does a string comparison, which means that it compares the dates character by character, starting from the left.

As an example, assume that the application contains the following records:

Number	Name	Start Date
113457	John Smith	21-Jan-90
113514	Maria Braun	15-Jun-90
113645	Bill Harvey	10-Apr-91

A direct comparison of the contents of START_DATE would be as follows:

```
;;BIND;;  
BIND *STAFF TO STAFF ENTRY WITH  
.START_DATE GES #SAVED_FROM_DATE AND  
.START_DATE LES #SAVED_TO_DATE
```

If the symbols #SAVED_FROM_DATE and #SAVED_TO_DATE contain the dates 10-Jan-90 and 20-May-90, respectively, ALL-IN-1 returns only the record whose joining date is 15-Jun-90. Clearly, this is chronologically inaccurate, but as a string value it falls between the specified dates.

To compare dates chronologically, you must first convert the dates into NBS format, using the DATE_CONVERT function.

The NBS format is a purely numeric representation of a date, starting with the year, then the month, day, hour, minute, second, and hundredth of a second. For example, the date 30-Jan-1991 13:55 is 1991013013550000 in NBS format.

By comparing dates in NBS format, ALL-IN-1 produces chronologically accurate data.

Therefore, to produce an index of records from a comparison of dates (using the staff registration application as an example), do the following:

1. Create a hidden field on the entry form to store the NBS format of the date entered in START_DATE.

To create a hidden field, use the FMS editor to create a 16-character field on the entry form. Give this field the FMS attributes No Echo and Display Only (so the user cannot modify the contents of the field), and assign a name to it, for example START_DATE_NBS.

2. Use DATE_CONVERT to convert the date from the START_DATE field to NBS format and store it in START_DATE_NBS.

Perform the date conversion when postprocessing the entry form. To do this, enter the following named data for the entry form:

```
;;.TYPE;;
ENTRY/MODE=UPDATE/HARD = .....
/POST='IFEXIT'①.IF START_DATE NES "" THEN
  DATE_CONVERT START_DATE_NBS,START_DATE,7 ELSE
  GET START_DATE_NBS = "0000000000000000"②
;;START_DATE;;
/VALID=CALL$SET_DATE;GET START_DATE=OA$TM_DATE
/OPTIONAL
```

- ① Perform the date conversion when postprocessing the form. You can use the DATE_CONVERT function as a post function to the /VALID field qualifier, but this has the following disadvantages:

- If a user exits the field using BACKSPACE or UP ARROW, ALL-IN-1 does not perform the post function and, thus, does not update the NBS field.
- If a user changes an existing date (by using ERASE LINE to clear the field and then entering a new date), ALL-IN-1 does not update the value already loaded into the field containing the NBS format.

- ② An empty NBS field has a string value of " ". This value does not fall within the range "0000000000000000" to "9999999999999999". Therefore, you must specify that a blank NBS date field has the value "0000000000000000", otherwise the related record does not appear in the index search.

APPLICATION PROGRAMMING

3. Create an index form that compares dates stored in NBS format.

To compare dates from the START_DATE_NBS field, include named data like the following:

```
;;BIND;;  
XOP "~~SET_UP_NBS_DATES~~"  
BIND *STAFF_ENTRY TO STAFF_ENTRY WITH❶  
.START_DATE_NBS GES #FROM_NBS AND  
.START_DATE_NBS LES #TO_NBS  
;;~~SET_UP_NBS_DATES~~;;❷  
.IF #SAVED_FROM_DATE NES "" THEN  
DATE_CONVERT #FROM_NBS,#SAVED_FROM_DATE,7  
ELSE GET #FROM_NBS = "0000000000000000"\  
.IF #SAVED_TO_DATE NES "" THEN  
DATE_CONVERT #TO_NBS,#SAVED_TO_DATE,7  
ELSE GET #TO_NBS = "9999999999999999"
```

- ❶ The BIND record selection expression (RSE) uses the dates stored in the NBS date field to create the phantom data set. Prior to executing the BIND function, set up symbols to contain the NBS equivalents of the two search dates.
- ❷ The XOP ~~SET_UP_NBS_DATES~~ converts the dates to NBS format.

Creating Time Fields

ALL-IN-1 allows you to enter times in several different formats, including keywords such as NOON and MIDNIGHT. The user profile specifies whether times are displayed in 12- or 24-hour clock format. To create a time field for ALL-IN-1, use the FMS editor to create a 7-character field and assign it a name, for example MEAL_TIME.

NOTE

ALL-IN-1 provides its own time validation facilities, as described in the next section. Therefore, do not use the facility provided by FMS to create predefined time fields, since ALL-IN-1 does not process these correctly.

Validating Time Fields

ALL-IN-1 provides several special symbols and DSABs that you can use to validate times. These symbols and DSABs are described in *Application Programming: Reference Volume 1*. The following named data shows you how to use special symbols and validation DSABs to validate a field called START_TIME:

```
;;START_TIME;;  
/VALID=CAL$TIME_CHECK❶:"START_TIME"❷
```

- ❶ CAL\$TIME_CHECK is a validation DSAB that validates a time. ALL-IN-1 checks the time against the user profile work start and end hours. So, if users work from 8:00 a.m. to 5:00 p.m., they can enter 3 to mean 3:00 p.m.
- ❷ This symbol contains the validated time in the format specified in the user profile.

Validating Start and End Times

If you have two time fields on a form, for example, to schedule the start and end times of a meeting, you might want to validate that the end time is later than the start time. To validate time fields in this way, add the following named data to the form:

```
;;MEETING_TIME_BEGIN;;
/VALID=CAL$TIME_CHECK:"MEETING_TIME_BEGIN"
;;MEETING_TIME_END;;
/VALID=CAL$END_TIME_CHECK❶:"MEETING_TIME_BEGIN,MEETING_TIME_END"❷
```

- ❶ CAL\$END_TIME_CHECK validates an end time against a start time.
- ❷ The key of reference consists of two symbols, the first to contain the start time and the second to contain the validated time in the format specified in the user profile.

SYSTEM MANAGEMENT

Testing and Repairing Users' File Cabinets

This article supplements information on testing and repairing users' File Cabinets in Chapter 9 of the *ALL-IN-1 Management Guide*. The article describes:

- How the Preliminary Test of File Cabinet (PT), Test and Repair User Areas (TRU), and Test and Repair Mail Areas (TRM) housekeeping procedures work
- How to configure your system so that the housekeeping procedures work efficiently
- The error reports that are produced when you run the TRU and TRM procedures

Configuring for the Test and Repair Procedures

The performance of the test and repair procedures can be considerably improved by:

- Creating a number of execution batch queues that are assigned to a clusterwide generic batch queue.

By default, both the test and repair procedures use the batch queue, SYS\$BATCH. However, Digital recommends that you create a dedicated generic batch queue for these procedures to prevent SYS\$BATCH from becoming overloaded.

To balance the use of processing resources in a VAXcluster system, set up a queue on each node in the VAXcluster that is running ALL-IN-1.

- Ensuring that users' accounts are distributed across more than one disk.

When a test and repair procedure is submitted to a generic batch queue, it relies on a master-and-slave relationship between the multiple processes: a master process controls the operations, and a number of slave processes perform the tasks delegated to them by the master process. A slave process is activated for each disk containing users' accounts.

By setting up the File Cabinet test and repair procedures to run in this way, the procedures can process users' File Cabinets in parallel — this speeds up the procedures by quite a considerable amount. The procedure submits a batch process for each disk storing users' accounts; the separate batch processes run at the same time. If the number of disks to be processed is greater than the number of execution queues available, the remainder of the disks are processed in turn when an execution queue becomes free.

If the procedures are not submitted to a generic queue, the processing is performed by a single process.

To optimize the performance of the test and repair procedures in a VAXcluster, the master process should be submitted to the batch queue on the most powerful machine or node. Do this as follows:

1. Set up each of the execution batch queues, for example:

```
$ INITIALIZE/QUEUE/BATCH/ON=node_name:: queue_1
```

where:

queue_1 runs on the most powerful node in the VAXcluster system.

2. Set up the generic queue, specifying the execution queue that runs on the most powerful node first, for example:

```
$ INITIALIZE/QUEUE/BATCH/GENERIC=(queue_1,queue_2,queue_3)/START generic_queue
```

The master process is submitted to the execution queue with the name *queue_1*.

The test and repair procedures can benefit on a single-node system, just as on a VAXcluster, from the use of a generic queue and associated execution queues. One process can be performing the disk input/output (I/O) operations, while another process performs central processing unit (CPU) operations.

For information about creating generic and execution queues, refer to the guide to maintaining a VMS system. See the *VMS VAXcluster Manual* for information on setting up queues in a clustered environment.

How the Test and Repair Procedures Work

Both the test and repair procedures use the same File Cabinet Verify and Repair (FCVR) program to test and repair the File Cabinet. The FCVR program, OA\$SM_FCVR.EXE, performs the processing required by both the master and slave processes. The FCVR program tests and repairs the File Cabinet in the following three phases:

- Phase 1

Phase 1 verifies and repairs users' private documents in their File Cabinets; repairs to shared documents are made during phases 2 and 3.

If the FCVR program is invoked by the TRU procedure, the program stops when phase 1 is complete.

When a generic batch queue is available, the FCVR program invokes a number of slave processes, one for each disk holding users' accounts. The slave processes perform phase 1 tasks only.

If the FCVR program is invoked by the TRM procedure, any references to shared documents are stored in temporary files for examination during phase 3. The temporary files are stored in the directory defined by the system logical name OA\$FCVR_FILES.

- Phase 2

Phase 2 searches the Shared Document Attributes Files (SDAFs) and the PENDING.DAT file for references to shared documents. These references are either for attachments to other shared documents (found by examining the SDAF files) or for undelivered mail messages (found by examining the PENDING.DAT file).

When a generic batch queue is available, phase 2 operations are carried out by the master process at the same time as the slave processes are performing phase 1 operations.

- Phase 3

Phase 3 starts when phases 1 and 2 are complete and verifies and repairs users' shared documents. This phase examines all SDAF references stored in the temporary files created during phases 1 and 2.

Preliminary Test of File Cabinet (PT) Procedure

Before scheduling either of the test and repair procedures, Digital recommends that you schedule the PT procedure.

The PT procedure checks for conditions that may prevent the user area and mail area test and repair procedures from running successfully. This procedure works quickly, and you can save a lot of time by using it to identify and correct these conditions before you schedule the TRU and TRM procedures. This particularly applies to the TRM procedure, which requires ALL-IN-1 to be shut down when it is running.

Scheduling the Test and Repair User Areas (TRU) Procedure

If possible, schedule the TRU procedure to run at a time when no one is likely to be using ALL-IN-1. Accounts that are being accessed when the TRU procedure tries to access them are omitted from the procedure.

If you scheduled the PT procedure to run before the TRU procedure, check the log file from the PT procedure to ensure that there are no conditions that will prevent the TRU procedure from running successfully.

Error Reporting for the Test and Repair User Areas Procedure

When the TRU procedure finishes, use the Show information (SI) option on the Administration menu to see whether the procedure ran successfully. For a detailed list of the events that took place during the procedure, refer to the log file OA\$LOG:OA\$SM_FCVR_USER_AREA.LOGNBS_time.

If the procedure did not run successfully, ALL-IN-1 sends an exception report to the person who is designated to receive exception reports for the TRU procedure — usually the ALL-IN-1 manager. The exception report refers you to the log file OA\$SM_FCVR_USER_AREA.LOGNBS_time, which indicates what occurred to prevent the procedure from completing successfully.

If the procedure did run successfully, the log file contains one or more of the following lists:

- A list of accounts that were repaired and accounts that have errors, if you specified that you wanted to repair errors found in File Cabinets (the default setting).
If, when you check the log file, you find that some documents were recovered and put into users' Recovered Documents folders, tell the users that they will find the documents in this folder.
- A list of the repairs that need to be done, if you did not specify the default setting.
- A list of errors that will stop the TRM procedure from completing.

If you set up a generic queue and associated execution queues, log files are created for the master and slave processes, in addition to the housekeeping procedure log file in OA\$LOG. These files are:

- The subprocess log files for the master and slave processes.

These are held in the ALL-IN-1 directory of the account from which the procedure is scheduled and are all called A1SUB.LOG.

SYSTEM MANAGEMENT

These log files contain information about the accounts that were processed by slave processes and list the SDAF and PENDING.DAT files that were scanned and repaired by the master process.

The files from the most recent run of the procedure are deleted when the procedure is next run.

- The batch job log files for the slave processes.

These files are located in the top-level directory of the account from which the procedure is scheduled. The file names are of the format FCVRSLAVEPROCESSnnnnn.LOG, where *nnnnn* is the identity number of the process. The information in the log files is about the batch job and is only likely to be of use after the TRU procedure fails, when they may yield some extra information.

The files from the most recent run of the procedure are deleted when the procedure is next run.

Scheduling the Test and Repair Mail Areas (TRM) Procedure

The TRM procedure shuts down and logs users out of ALL-IN-1, so schedule it to run when users are not logged in to ALL-IN-1, for example, in the evening. If users log in when this procedure is running (using the /OVERRIDE=SHUTDOWN qualifier, for example), the procedure fails.

If you scheduled the PT procedure to run before the TRM procedure, check the log file from the PT procedure to ensure that there are no conditions that will prevent the TRM procedure from running successfully.

Error Reporting for the Test and Repair Mail Areas Procedure

When the TRM procedure finishes, use the Show information (SI) option on the Administration menu to see whether the procedure ran successfully. For a detailed list of the events that took place during the procedure, refer to the log file OA\$LOG:OA\$SM_FCVR_MAIL_AREA.LOGNBS_time.

If the procedure did not run successfully, ALL-IN-1 sends an exception report to the person who is designated to receive exception reports for the TRM procedure — usually the ALL-IN-1 manager. The exception report refers you to the log file OA\$SM_FCVR_MAIL_AREA.LOGNBS_time, which indicates what occurred to prevent the procedure from completing successfully.

If the procedure did run successfully, the log file contains one or more of the following lists:

- A list of accounts that have errors and accounts that were repaired in mail areas, if you specified that you wanted to repair mail areas, but not File Cabinets (the default setting).
- A list of accounts that have errors and that were repaired in user areas, as well as mail areas, if you specified that you wanted to repair all File Cabinets, as well as mail areas.
- A list of accounts that were tested and the repairs that need to be done, if you specified that you did not want to repair errors found in mail areas.

When you check the log file, if you specified that shared documents should be recovered, you may find that documents were put into the Recovered Shared Documents folder. Use the Show message status (SH) option on the Electronic Messaging menu to see who the sender and recipients of the message are. You can then inform users that you recovered the message.

If you set up a generic queue and associated execution queues, log files are created for the master and slave processes, in addition to the housekeeping procedure log file in OA\$LOG. These files are:

- The subprocess log files for the master and slave processes.

These are held in the ALL-IN-1 directory of the account from which the procedure is scheduled and are all called A1SUB.LOG.

These log files contain information about the accounts that were processed by slave processes and list the SDAF and PENDING.DAT files that were scanned and repaired by the master process.

The files from the most recent run of the procedure are deleted when the procedure is next run.

- The batch job log files for the slave processes.

These files are located in the top-level directory of the account from which the procedure is scheduled. The file names are of the format FCVRSLAVEPROCESSnnnnn.LOG, where *nnnnn* is the identity number of the process. The information in the log files is about the batch job and is only likely to be of use after the TRM procedure fails, when they may yield some extra information.

The files from the most recent run of the procedure are deleted when the procedure is next run.

The following guidelines will help you determine how long the TRM procedure takes to run:

- When running the TRM procedure without repairs, the time taken depends mainly on the size of the SDAFs. Allow approximately 1 minute for every 1500 to 2000 records. The time of the run is increased if other processes are running on the same node as the master TRM process.
- The time taken by the repair process depends on the number of repairs to be performed. The most time-consuming repair is the recovery of shared documents, performed if you enter an account name into the Recovered messages account field on the Schedule Test and Repair of Mail Areas form. Allow approximately 1 hour for every 1000 recovered documents.

If you run the TRM process regularly, the repair phase takes less time.

These guidelines are based on performance tests running ALL-IN-1 on a VAX 8800 system.

Cumulative Index

The index on the following pages contains entries from the August and November 1990 and February and May 1991 issues of the *ALL-IN-1 Information Update*, as well as from this issue. For each entry, the issue date and page number are shown.

INDEX

8-bit characters, *August 1990*, 11
in user names, *August 1990*, 11

A

A1PAT509, *November 1990*, 5
A1PAT510, *November 1990*, 7
A1PAT511, *November 1990*, 8
A1PAT512FOREIGN, *February 1991*, 5
A1SUB.LOG log file
 See Log file
A1TRACE.LOG, *November 1990*, 13
 contents of, *November 1990*, 14
 viewing, *November 1990*, 14
Access violation, *November 1990*, 5, 6, 8, 9
Addresses, long
 not matching full names, *November 1990*, 6
 reading messages with, *November 1990*, 8
 showing message status, *November 1990*, 8
Address validation, *November 1990*, 6
Alias file, *May 1991*, 20
ALL-IN-1 subprocess, *November 1990*, 23
AND operator
 See Operators
Applications
 compute-intensive
 managing, *May 1991*, 16
 problems, *May 1991*, 16
 solutions to problems, *May 1991*, 16
ASCII documents
 reading, *November 1990*, 7
ATTACH function
 See Functions
Auto forward, *November 1990*, 5

B

Balance mail areas, *November 1990*, 5
Batch
 VAXcluster queues, *August 1990*, 16
Batch queue
 specifying for File Cabinet test and repair
 procedures, *August 1991*, 17
BEGINNING operator
 See Operators
BIND function, *November 1990*, 19
BINDW function, *November 1990*, 19
Boolean expression, *May 1991*, 12

Break, *August 1990*, 43
Broadcast handling, *November 1990*, 7

C

CAL\$CHECK_DATE DSAB, *August 1990*, 65
CAL\$SET_DATE DSAB, *August 1991*, 11
CAL\$TIME_CHECK DSAB, *August 1991*, 15
CALLBACK, *November 1990*, 7
CDA support, *November 1990*, 11
CDA_CONVERT function, *February 1991*, 21
Common-environment VAXcluster system
 running ALL-IN-1, *August 1990*, 15
Communications Control, *August 1990*, 43
Compile and install CM text library (CCM)
 menu option
 See Menu options
Compound documents (CDA)
 defined, *February 1991*, 19
 storage format, *February 1991*, 19
 STORED SEMANTICS attribute, *February 1991*, 19
 support in Version 2.4, *August 1990*, 3
Compute-intensive applications
 See Applications
CONTAINING operator
 See Operators
COPY function
 See Functions
CTRL/T, *November 1990*, 23
CTRL/Y, *November 1990*, 23
Customization Management, *August 1990*, 45
 extending, *August 1990*, 45
Customizations
 after installing patches, *November 1990*, 7
CXP subfunctions, *August 1990*, 43

D

Data
 accessing, *May 1991*, 11
Data buckets, *May 1991*, 17
 optimum size, *May 1991*, 17
Data sets, phantom, *November 1990*, 19
DATATRIEVE, *August 1990*, 36
DATA_FILE subfunctions
 See Subfunctions

Date field
 comparing, *August 1991*, 12
 creating, *August 1991*, 11
 NBS format, *August 1991*, 13
 validating, *August 1991*, 11
 Date format
 NBS, *August 1991*, 12
 user's preferred, *August 1990*, 65
 Date validation, *August 1990*, 65; *August 1991*, 11
 form field, *August 1990*, 65
 DATE_CONVERT function, *August 1990*, 65; *August 1991*, 12
 DDIF, *February 1991*, 19
 DECLARE_METER function
 See Functions
 DEClaser features, *February 1991*, 13
 DEClaser printer
 Adding support for, *February 1991*, 9
 Duplex attributes, *February 1991*, 11
 Manual feed option attribute, *February 1991*, 13
 Tray selection attributes, *February 1991*, 13
 Using, *February 1991*, 9
 DECnet, *August 1990*, 24, 43
 DECTrace, *May 1991*, 10
 Deferred messages, *August 1990*, 23
 retry count, *November 1990*, 8
 Define Communications Lines, *August 1990*, 44
 Define Computer Systems, *August 1990*, 44
 Delete all (XD) option, *August 1991*, 6
 DELETE_FILE function
 See Functions
 Development area, *August 1990*, 45
 Distribution lists, *November 1990*, 6
 DOWN ARROW, *November 1990*, 6
 omitting when printing mail, *August 1991*, 5
 skipping long lists when reading mail, *August 1991*, 5
 Documentation
 Bookreader versions, *August 1991*, 3
 Document formatting queue
 in a VAXcluster system, *August 1990*, 17
 Documents
 compound, *February 1991*, 19
 selecting, *August 1990*, 13
 Document transfers, *November 1990*, 7
 DO function
 See Functions
 DO scripts, *August 1990*, 61; *May 1991*, 7, 8, 16
 See also Scripts
 DOTS, *February 1991*, 19
 DSAB (data set access block), *May 1991*, 13

DSAB (data set access block) (Cont.)
 validation, *August 1991*, 11
 DSR (Data Set Reference), *May 1991*, 11
 using, *May 1991*, 11
 DTIF, *February 1991*, 19

E

EDIT function
 See Functions
 Elapsed time
 calculating, *May 1991*, 10
 Electronic Messaging, *November 1990*, 5
 address validation, *November 1990*, 6
 deferred messages, *November 1990*, 8
 FIND, *November 1990*, 6
 long addresses, *November 1990*, 6, 8
 multiple delivery, *November 1990*, 8
 nondelivery, *November 1990*, 8
 paper mail, *November 1990*, 8
 sent date, *November 1990*, 8
 ULTRIX mail gateway, *November 1990*, 6
 Electronic Messaging (EM) menu, *August 1991*, 5
 Empty Wastebasket failure, *November 1990*, 8
 ENS operator
 See Operators
 EQS operator
 See Operators
 Escape sequence, *August 1990*, 43
 ESM (Element Speed Measurement) menu
 option, *May 1991*, 9
 EXECUTE function
 See Functions
 External reference, *February 1991*, 19

F

FCVR (File Cabinet Verify and Repair)
 phase 1 overview, *August 1991*, 18
 phase 2 overview, *August 1991*, 18
 phase 3 overview, *August 1991*, 18
 program, *August 1991*, 18
 FCVRSLAVEPROCESSnnnn.LOG log file
 See Log file
 Fetcher, *August 1990*, 24; *November 1990*, 7
 Field validation, *May 1991*, 13
 File Attachment (FA) menu option
 See Menu options
 File Cabinet
 preliminary testing for repair procedures, *August 1991*, 19
 reducing run time for test and repair procedures, *August 1991*, 17
 repairing, *August 1991*, 17
 specifying batch queue for test and repair procedures, *August 1991*, 17
 testing, *August 1991*, 17

- Files
 - backing up, *February 1991*, 23
 - File definition language (FDL), *May 1991*, 17
- File Text (FT) menu option
 - See Menu options
- Fixes
 - issuing, *February 1991*, 3
- FMS
 - field attributes
 - Right Justify, *August 1991*, 7
 - Zero Fill, *August 1991*, 7
 - formatting features
 - examples, *May 1991*, 6
 - overhead costs, *May 1991*, 6
- FMS form driver, *May 1991*, 15
- FMS forms, *August 1990*, 57
- Folders
 - Outbox, *August 1991*, 5, 6
 - Read, *August 1991*, 5, 6
- FOREIGN DSAB, *August 1990*, 63
- FOR function
 - See Functions
- Format
 - of compound documents, *February 1991*, 19
 - converting, *February 1991*, 21
- Formatter queue
 - in a VAXcluster system, *August 1990*, 17
- Form directory, *May 1991*, 15
- Form driver
 - FMS, *May 1991*, 15
- FORM function
 - See Functions
- Form library, *August 1990*, 49; *May 1991*, 15
 - compiling, *May 1991*, 3
 - development (DEVELOP.FLB), *May 1991*, 15
 - installing in a VAXcluster system, *August 1990*, 19
- Forms
 - compiling
 - with unresolved special symbols, *May 1991*, 3
 - designing, *May 1991*, 6
 - Editor Attributes (EA) form, *August 1991*, 5
 - FMS, *May 1991*, 5
 - FORM function, *May 1991*, 6
 - libraries, *May 1991*, 5, 15
 - number limit, *May 1991*, 6
 - optimum value, *May 1991*, 6
 - performing an operation without calling, *May 1991*, 6
 - size
 - examples, *May 1991*, 5
 - with .MORE directive, *May 1991*, 3
- Forms compiler

- Forms compiler (Cont.)
 - reporting an error, *May 1991*, 3
- Functions
 - ATTACH, *May 1991*, 9
 - BIND, *November 1990*, 19
 - BINDW, *November 1990*, 19
 - BOTTOM, *August 1991*, 5
 - CDA_CONVERT, *February 1991*, 21
 - COPY, *May 1991*, 9
 - DATE_CONVERT, *August 1990*, 65; *August 1991*, 12
 - DECLARE_METER, *May 1991*, 9
 - DELETE_FILE, *May 1991*, 9
 - DO, *May 1991*, 8
 - EDIT, *May 1991*, 9
 - EXECUTE, *May 1991*, 9
 - FOR, *May 1991*, 7
 - FORM, *May 1991*, 6
 - GET, *May 1991*, 7
 - making GOLD GET work more quickly, *August 1991*, 5
 - GET_BY_KEY, *May 1991*, 13
 - GET_NEXT, *May 1991*, 13
 - GET_RMS_SEMANTIC_TAG, *February 1991*, 20
 - INSTALL, *May 1991*, 9
 - LOCATE, *May 1991*, 13
 - MERGE, *May 1991*, 8, 16
 - OA\$LOG_TO_FILE, *May 1991*, 10
 - OA\$STAT_GET_STATISTICS, *May 1991*, 10
 - PURGE_FILE, *May 1991*, 9
 - QUEUE_BATCH, *May 1991*, 9
 - RENAME, *May 1991*, 9
 - SCRIPT, *May 1991*, 8
 - SELECT, *August 1991*, 6
 - SPAWN, *May 1991*, 9
 - XOP, *May 1991*, 7
- .FUNCTION script directive, *February 1991*, 15
- .FX script directive, *February 1991*, 15

G

- Generic batch queue
 - See Batch queue
- GET DOCMT screen on field, *August 1991*, 5
- GET function
 - See Functions
- GET_BY_KEY function
 - See Functions
- GET_NEXT function
 - See Functions
- GET_RMS_SEMANTIC_TAG function, *February 1991*, 20
- Global buffers, *May 1991*, 17
- GOLD MESSAGE, *November 1990*, 7

GOLD TRACE, *November 1990*, 14

H

ALL-IN-1 image
 privileges needed, *August 1990*, 9
Help, *August 1990*, 55
 customizing, *August 1990*, 55
 keysize, *August 1990*, 55
 keyword, *August 1990*, 55
 library, *August 1990*, 55
Heterogeneous VAXcluster system
 See Multiple-environment VAXcluster system
HLPSOURCE.HLB, *August 1990*, 56
Homogeneous VAXcluster system
 See Common-environment VAXcluster system
Housekeeping procedures
 Preliminary Test of File Cabinet, *August 1991*, 17
 scheduling, *August 1991*, 19
 Test and Repair Mail Areas, *August 1991*, 17, 20
 time taken to run, *August 1991*, 21
 Test and Repair User Areas, *August 1991*, 17, 19

I

Image file
 installing in a VAXcluster system, *August 1990*, 18
INCREMENT_METER function, *November 1990*, 7
Indexed search, *May 1991*, 13
 for records, *May 1991*, 11
Indexes
 search phrases, *November 1990*, 8
Index forms
 using phantom data sets, *November 1990*, 19
INSTALL function, *May 1991*, 9
 See also Functions
 with CALLBACK, *November 1990*, 7
Installing form library
 in a VAXcluster system, *August 1990*, 19
Installing image file
 in a VAXcluster system, *August 1990*, 18
Installing text library
 in a VAXcluster system, *August 1990*, 19
IVP (Installation Verification Procedure), *May 1991*, 3

K

Key of reference, *May 1991*, 11

L

LIB\$FREE_VM error, *November 1990*, 8
Live area, *August 1990*, 45
Live link, *February 1991*, 19
LOCATE function
 See Functions
Lock element (LE), *August 1990*, 52
Log file
 A1SUB.LOG, *August 1991*, 19, 21
 FCVRSLAVEPROCESSnnnn.LOG, *August 1991*, 20, 21
 for File Cabinet test and repair procedures, *August 1991*, 19, 20, 21
Long addresses
 not matching full names, *November 1990*, 6
 reading messages with, *November 1990*, 8
 showing message status, *November 1990*, 8

M

Mail
 clearing out old messages, *August 1991*, 5
 printing
 omitting long distribution lists, *August 1991*, 5
 skipping long distribution lists, *August 1991*, 5
Mail area
 description of testing and repairing, *August 1991*, 20
 time taken to test and repair, *August 1991*, 21
Mail directory, *August 1990*, 21
Mail Directory, *November 1990*, 6
Meeting notices, *November 1990*, 7
Menu options
 Compile and install CM text library (CCM), *May 1991*, 16
 File Attachment (FA), *August 1991*, 5
 File Text (FT), *August 1991*, 5
 Precompile a form library (PFC), *May 1991*, 15
 Print options (PO), *August 1991*, 6
 Refile document (RFD), *August 1991*, 6
 Search mail directory (SMD), *May 1991*, 19
 User setup (US), *August 1991*, 6
MERGE function
 See Functions
Message
 priority for, *August 1990*, 35
Message Router Directory Service, *May 1991*, 19
 setting up entries, *May 1991*, 20

Metering
 using, *May 1991*, 10
 Modem
 autodial, *August 1990*, 43
 Move to live area (MLA), *August 1990*, 46
 MRP (Message Router/P Gateway), *May 1991*, 19
 MRS (Message Router/S Gateway), *May 1991*, 19
 MR-Telex (Message Router Telex Gateway), *May 1991*, 19
 MTA (Message Transfer Agent), *May 1991*, 20
 Multiple-environment VAXcluster system
 running ALL-IN-1, *August 1990*, 15

N

Named data, *May 1991*, 7
 EM menu, *May 1991*, 6
 NETWORK.DAT, *February 1991*, 23
 Nonstandard files, *August 1990*, 63

O

OA\$DAF_y.DAT, *February 1991*, 23
 OA\$DATA:NETWORK.DAT, *February 1991*, 23
 OA\$DATA:PENDING.DAT, *February 1991*, 23
 OA\$DATA:PROFILE.DAT, *February 1991*, 23
 OA\$DATA_LLIV, *May 1991*, 17
 OA\$DATA_SHARE, *May 1991*, 17
 OA\$DATA_SHARE:CM_SDC_ELEMENTS.DAT, *November 1990*, 11
 OA\$FILE_SEARCH_ORDER, *August 1990*, 50; *May 1991*, 8
 OA\$LIB_SHARE, *May 1991*, 17
 OA\$MTI_ERR, *August 1990*, 35, 36
 OA\$MTI_LOG, *August 1990*, 35
 OA\$SHARy:OA\$DAF_y.DAT, *February 1991*, 23
 OA\$SM_FCVR.EXE, *August 1991*, 18
 OA\$STATUS
 checking value, *May 1991*, 7
 OA\$TABLE_DSAB, *November 1990*, 25
 specifying numbers in, *November 1990*, 25
 using hyphens in, *November 1990*, 25
 OA\$TEXT_DSAB, *August 1990*, 63
 OA\$TRA_SET IO, *May 1991*, 13
 OA\$TXL_SEARCH_LAST symbol, *August 1990*, 61
 OA\$TXL_SEARCH_ORDER symbol, *August 1990*, 61; *May 1991*, 8
 OAHELP.HLB, *August 1990*, 56
 OAMTISEND.COM, *August 1990*, 26
 Operators
 AND, *May 1991*, 12
 BEGINNING, *May 1991*, 12
 CONTAINING, *May 1991*, 12

Operators (Cont.)

ENS, *May 1991*, 12
 EQS, *May 1991*, 12

P

Paper mail
 formatting, *November 1990*, 8
 Patches, *November 1990*, 5
 A1PAT509, *November 1990*, 5
 A1PAT510, *November 1990*, 7
 A1PAT511, *November 1990*, 8
 A1PAT512FOREIGN, *February 1991*, 5
 available for ALL-IN-1 Version 2.3, *February 1991*, 5
 customizations after installing, *November 1990*, 7
 process for Version 2.3, *February 1991*, 4
 process for Version 2.4, *February 1991*, 3
 SUS, *February 1991*, 3
 PENDING.DAT, *February 1991*, 23
 Performance
 cost, *May 1991*, 5
 designing an application for, *May 1991*, 5
 maintaining files for, *May 1991*, 16
 managing an application for, *May 1991*, 15
 perceived, *May 1991*, 5
 programming an application for, *May 1991*, 5
 speed of, *May 1991*, 15
 testing applications for, *May 1991*, 9
 Permanent symbol table file (.PST), *May 1991*, 9
 Phantom data sets, *November 1990*, 19
 Phone, *August 1990*, 43
 Precompile a form library (PCF) menu option
 See Menu options
 Preliminary Test of File Cabinet (PT)
 See Housekeeping procedures
 Printing
 saving paper, *August 1991*, 6
 Print options (PO)
 See Menu options
 Privileges, *August 1990*, 9
 Profile
 large entry in Department field, *November 1990*, 5
 PROFILE.DAT, *February 1991*, 23
 Programming
 for optimal field access, *May 1991*, 11
 Protect an element from changes (PFC), *August 1990*, 52
 PURGE_FILE function
 See Functions

Q

Qualifiers

/RSE_VALID, *May 1991*, 13

/VALID, *May 1991*, 13

QUEUE_BATCH function

See Functions

R

Reading ASCII documents, *November 1990*, 7

Receive a document from VMS (RV), *August 1990*, 63

Receive without conversion (RVC), *August 1990*, 63

Recognition

using OA\$TABLE, *November 1990*, 25

Record session (RS), *August 1990*, 43

Refile document (RFD) menu option

See Menu options

RENAME function

See Functions

Retirement of support

Version 2.2, *November 1990*, 3

Retry count

deferred messages, *November 1990*, 8

RMS, *May 1991*, 11

checking data files for errors, *February 1991*, 23

RSE (record selection expression)

rules, *May 1991*, 12

using, *May 1991*, 12

with the FOR function, *May 1991*, 11

writing, *May 1991*, 12

S

Script directives

.GOTO, *May 1991*, 7

.LABEL, *May 1991*, 7

SCRIPT function

See Functions

Scripts

DO mode, *February 1991*, 15; *May 1991*, 7, 16

nesting, *February 1991*, 17

SCRIPT, *May 1991*, 16

writing guidelines, *May 1991*, 7

SCRIPT scripts, *August 1990*, 57, 61; *May 1991*, 8, 16

SCRIPTS Script, *February 1991*, 15

Scrolling feature

switching off during GOLD GET, *August 1991*, 5

SDAF

See shared document attributes file, February 1991, 23

Search mail directory (SMD)

See Menu options

Search order, *August 1990*, 62

Search phrases

with multilingual characters, *November 1990*, 8

Semantic tag

defined, *February 1991*, 19

SENDCHECK.EXE, *August 1990*, 26

Sender, *August 1990*, 23

batch job, *August 1990*, 35

log, *August 1990*, 23

message processing time, *August 1990*, 28

queue, *August 1990*, 23

run time, *August 1990*, 28

tuning, *August 1990*, 23

Sent date on messages, *November 1990*, 8

Sequential search, *May 1991*, 13

for records, *May 1991*, 11

Set date option, *November 1990*, 6

Set Working Conditions (SWC)

date format, *August 1990*, 65

Shared document attributes file (SDAF)

checking for RMS errors, *February 1991*, 23

effect of size on time taken to test and repair mail areas, *August 1991*, 21

reorganizing, *February 1991*, 23

SPAWN, *November 1990*, 23

See also Functions

Special fields

%CLOSE, *May 1991*, 11, 12

%KEY, *May 1991*, 12

%OPEN, *May 1991*, 11, 12

Special symbols, *May 1991*, 9

unresolved, *May 1991*, 3

Starting ALL-IN-1

in a VAXcluster system, *August 1990*, 18

Stopping ALL-IN-1

in a VAXcluster system, *August 1990*, 18

STORED SEMANTICS attribute, *February 1991*, 19

Subfunctions

DATA_FILE, *May 1991*, 11, 13

Subprocess

avoiding use of, *May 1991*, 9

Subprocess, ALL-IN-1, *November 1990*, 23

Support Update Software (SUS), *February 1991*, 3

Switch log files (SL), *August 1990*, 36

Symbols

adding leading zeros to, *August 1991*, 7

comparing, *May 1991*, 9

OA\$DATE_NBS, *May 1991*, 10

OA\$DA_FORM_CACHE_SIZE, *May 1991*, 6

OA\$FILE_SEARCH_ORDER, *May 1991*, 8

Symbols (Cont.)

OA\$MENU_REMAINDER, *May 1991*, 6
OA\$PROFIL_DIRECT, *May 1991*, 9
OA\$TXL_SEARCH_LAST, *August 1990*, 61
OA\$TXL_SEARCH_ORDER, *August 1990*,
61; *May 1991*, 8
right-justifying, *August 1991*, 7
specifying substrings, *August 1991*, 7
using, *May 1991*, 8
SYS\$CLUSTER_NODE, *August 1990*, 41

T

Templates, *August 1990*, 61
Terminal session disconnection, *November 1990*, 8
Test and Repair Mail Areas (TRM)
 See Housekeeping procedures
Test and Repair User Areas (TRU)
 See Housekeeping procedures
Text elements, *August 1990*, 50
Text library, *August 1990*, 50, 61; *May 1991*,
15
 CM, *August 1990*, 61; *May 1991*, 8
 location of files, *May 1991*, 8
 compiling, *May 1991*, 16
 installing, *May 1991*, 16
 installing in a VAXcluster system, *August 1990*, 19
 managing, *May 1991*, 15
 noncompiled, *May 1991*, 15
 precompiled, *May 1991*, 15
 source locations, *August 1990*, 51
Time field, *August 1991*, 11, 14
 creating, *August 1991*, 14
 validating, *August 1991*, 14
Time Management, *November 1990*, 6
 meeting notices, *November 1990*, 7
Trace facility, *November 1990*, 13
 to create UDPs, *November 1990*, 13
 turning off, *November 1990*, 14
 turning on, *November 1990*, 14
 use to trace records, *May 1991*, 10
Trace log file
 listing without exiting, *May 1991*, 3
Tracing
 See Trace facility
Training
 modifying, *August 1990*, 57
Tuning ALL-IN-1
 in a VAXcluster system, *August 1990*, 16
TXL, *August 1990*, 50, 61

U

UAF (user authorization file), *August 1990*, 22
UDP
 See User-defined processes
ULTRIX Mail Connection (UMC), *May 1991*,
20
ULTRIX mail gateway, *November 1990*, 6
UMC, *November 1990*, 6
User accounts
 creating, *August 1990*, 21
User area
 testing and repairing
 description, *August 1991*, 19
User-defined processes(UDP), *November 1990*, 13 to 17
 example of, *November 1990*, 15
 script directives in, *November 1990*, 17
 UDP names in, *November 1990*, 13
 using UDPs to automatically create,
 November 1990, 16
 writing, *November 1990*, 13
User hints, *August 1991*, 5
User names with apostrophes, *November 1990*, 5
User setup (US) menu option
 See Menu options

V

Validation
 date, *August 1990*, 65
 using OA\$TABLE, *November 1990*, 25
VAXcluster system
 alias, *August 1990*, 16
 batch queues, *August 1990*, 16
 common-environment
 running ALL-IN-1, *August 1990*, 15
 formatter queue, *August 1990*, 17
 installing form library, *August 1990*, 19
 installing image file, *August 1990*, 18
 installing text library, *August 1990*, 19
 multiple-environment
 running ALL-IN-1, *August 1990*, 15
 specifying batch queue for File Cabinet test
 and repair procedures, *August 1991*, 17
 starting ALL-IN-1, *August 1990*, 18
 stopping ALL-IN-1, *August 1990*, 18
 tasks, *August 1990*, 16
 tuning, *August 1990*, 16
Version 2.2
 retirement of support, *November 1990*, 3
Version 2.3 patch process, *February 1991*, 4
Version 2.4 patch process, *February 1991*, 3
VMS logical names, *August 1990*, 41
VMSmail import, *November 1990*, 7

VMS Online Documentation Library, *August 1991*, 3

W

Wastebasket, *August 1991*, 5
WPS-PLUS

incorporating text from one document into another, *August 1991*, 5

X

X.400
 problem in addresses with multiple
 organizational units, *May 1991*, 19
XOP function
 See Functions

Attention U.S. Software Contract Customers Submitting Software Problems

As part of Digital's continuing effort to improve customer satisfaction, U.S. Software Product Services (SPS) has changed the process for U.S. customers to submit software problems. These customers should report their software problems by calling their Customer Support Center (CSC) or by accessing either the Digital Software Information Network (DSIN) or DSNlink.

The CSCs have the technical expertise to quickly respond to many of our customers' problems without requiring Engineering intervention. Because many software problems are duplicates of problems that have already been addressed by Engineering, the CSCs can immediately respond to many inquiries.

If you do not have access to a CSC, submit your software problem to:

**Digital Equipment Corporation
P.O. Box 1167
Alpharetta, Georgia 30239-1167**

This address replaces the one currently found on the instruction sheet of the Software Performance Report (SPR) form.

Digital Equipment Computer Users Society

Benefits of Belonging

The Digital Equipment Computer Users Society (DECUS) is one of the largest and most respected users groups in the computer industry today. Membership in DECUS, which is free and voluntary, affords the individual user information and services not found anywhere else.

DECUS provides an environment where users of Digital Equipment Corporation products can share information with other users and with Digital. Members can find out the latest news on Digital's hardware, software, and educational products. The feedback exchange with Digital allows the users of Digital's products a voice in the direction the company takes in the future.

Founded in 1961, DECUS has three autonomous chapters worldwide—DECUS U.S., DECUS Europe, and DECUS General International Area (GIA). DECUS services and activities are shared between these chapters through mutual agreements.

All DECUS services promote the exchange of information in a commercial-free environment. These services include:

Special Interest Groups (SIGs)

These groups, formed around an area of common interest, exist for a variety of hardware, operating systems, languages, applications, and marketing areas. Participation in these groups allows contact among fellow users to exchange information and share technical expertise in the area of most interest to the user.

Local Users Groups (LUGs) and National Users Groups (NUGs)

LUGs and NUGs are licensed groups of individuals who gather to share information with other users on a periodic basis. Not only do these people share professional interest, but they also share geographic and cultural ties. Often, Digital representatives attend these meetings to discuss new products and services and supply updates on existing policies and procedures.

Symposia

DECUS holds symposia each year in the different chapters, two per year in the U.S. These meetings provide unique opportunities for users with a wide spectrum of experience to meet for up to five intensive days of technical exchange. Included in symposia activities are workshops, clinics, panels, tutorials, and formal paper presentations. Digital participates in symposia by sending product group managers and developers to discuss strategies, products, problems, and solutions.

Communications

The flow of information among users, as well as between users and Digital, is the primary goal of DECUS. Various publications are published by DECUS to support this communication. They include chapter newsletters and *The Proceedings*, a technical volume published after each symposium.

DECUS also publishes the Special Interest Groups *SIGs Newsletter* on a monthly basis. The newsletter is divided into Special Interest Group sections and provides specialized information pertaining to specific Digital products. The monthly newsletter and *The Proceedings* are available through the DECUS Subscription Service.

Seminars

DECUS recognizes the need for ongoing training opportunities that help members keep pace with the changing technologies of the computer industry. To that end, DECUS sponsors 1-day technical seminars throughout the year in various locations. These seminars are developed and presented by DECUS members who are experienced industry professionals.

Program Library

The DECUS Program Library is the main vehicle for the exchange of public domain software among users of all Digital systems. The library contains over 1000 software programs written and voluntarily submitted by users. These programs include compilers, editors, utilities, numerical and statistical functions, as well as games and graphic routines. The library publishes a software catalog on an annual basis that lists and describes all the DECUS programs.

DECUServe

DECUServe is an electronic conferencing system based on VAX Notes. The network is available to all U.S. members 24 hours a day, 7 days a week for an annual subscription fee.

Joining DECUS

You are cordially invited to join with over 100,000 other users of Digital products around the world and begin to share your experiences, both the successes and problems.

For more information, contact the appropriate DECUS Chapter Office.

DECUS Chapter Offices—Worldwide

DECUS U.S.

DECUS U.S.
333 South Street, (SHR1-4/D30)
Shrewsbury, Massachusetts 01545-4112
Tel. (508) 841-3389

DECUS Europe

DECUS Europe Headquarters
12 Avenue des Morgines
Case Postale 176
CH-1213 Petit-Lancy 1
Switzerland

DECUS BELUX
Luchtschipstraat 1
1, Rue de l'Aeronef
B-1140 Evere (Brussels)
Belgium

DECUS Finland
P.O. Box 6
SF-02201 Espoo
Finland

DECUS Holland
Europalaan 44
P.O. Box 9212
3506 GE Utrecht
The Netherlands

DECUS Ireland
c/o Digital Equipment Ireland Ltd.
Park House, North Circular Road
IRL-Dublin 7
Ireland

DECUS Italia
Viale Monza 338
I-20128 Milano
Italy

DECUS At-Large Chapter
12 Avenue des Morgines
Case Postale 176
CH-1213 Petit-Lancy 1/GE
Switzerland

DECUS Denmark
c/o Digital Equipment Corporation A/S
Aadalsvej 99
DK-2970 Hoersholm
Denmark

DECUS France
Parc du Bois Briard
9/13 avenue du Lac, BP 235
F-91007 Evry Cedex
France

DECUS Iceland
c/o DECUS Denmark

DECUS Israel
c/o Digital Equipment
Acadia Junction
Herzlia 46 733
Israel

DECUS Muenchen
Freischuetzstrasse 91, Postfach 810247
D-8000 Muenchen 81
Federal Republic of Germany

DECUS Chapter Offices—Worldwide (Continued)

DECUS Norway
Ammerudveien 22
N-0958 Oslo 9
Norway

DECUS Portugal
c/o Digital Equipment Portugal Lda.
Empreendimento Torres das Amoreiras
Av. Eng. Duarte Pacheco, Torre 1-9 andar
P-1000 Lisbon
Portugal

DECUS Spain
c/o Digital Equipment Corporation SA
Cerro del Castanar 72, Mirasierra
E-28034 Madrid
Spain

DECUS Sweden
Box 5194
S-402 26 Gothenburg
Sweden

DECUS Switzerland
DECpark
Uberlandstrasse 1 Postfach
CH-8600 Dubendorf 1
Switzerland

DECUS U.K., Ireland, and Middle East
Queen's House
Forbury Road
GB-Reading, RG 1 3JJ
U.K.

DECUS GIA (General International Area)

DECUS GIA Headquarters
100 Nagog Park AKO1-1/B11
Acton, Massachusetts 01720
U.S.A.

DECUS South Pacific Chapter
Australia Office
Locked Bag 16, 410 Concord Road
Rhodes, New South Wales 2138
Australia

DECUS
New Zealand Office
P.O. Box 8610
Symonds Street
Auckland 3, New Zealand

DECUS Canada
505 University Avenue, 15th Floor
Toronto, Ontario M5G 1X4
Canada

DECUS Far East
Digital Equipment Corporation
19-21 Fleet House
38 Gloucester Road
Wanchai, Hong Kong

DECUS Japan
Nihon Digital Equipment KK
Sunshine 60, P.O. Box 1135
1-1. Higashi Ikebukuro 3 - chome
Toshima-ku Tokyo
Japan 170

DECUS South America
Digital Equipment do Brasil Ltda.
Av. Presidente Wilson
231 26th Floor
20030 Rio de Janeiro, RJ
Brasil

DECUS Latin America/Carribean Chapter
DECUS LACC
800 Fairway Drive
Suite 400
Deerfield Beach, FL 33441
U.S.A.

DECUS SUBSCRIPTION SERVICE
SIGs NEWSLETTER
U.S. Chapter Proceedings
(U.S. Chapter Members Only)

As a member of the DECUS U.S. Chapter, you are entitled to contribute and subscribe to the DECUS monthly publication, *SIGs Newsletter*. You also have the opportunity to subscribe to the Symposia Proceedings which are a compendium of the reports from various speakers at the U.S. National DECUS Symposia.

- The order form below must be used as an invoice
- All checks must be made payable to DECUS
- All orders **must** be paid in full
- \$25.00 minimum for credit card orders
- No refunds will be made
- The address provided below will be used for all DECUS mailings, e.g., membership, subscription service, and symposia
- SIGs Newsletter price is for a one-year subscription beginning the month following receipt of payment

.....

Name _____ DECUS Member No. _____

Company _____

Address _____

City _____ State _____ Zip _____

Phone _____

Subscription Service Offering	Qty	Unit Price	Total
SIGs Newsletter	_____	\$40.00	_____
Spring '90 Proceedings (SP0)	_____	15.00	_____
Fall '90 Proceedings (FA0)	_____	15.00	_____
Spring '91 Proceedings (SP1)	_____	15.00	_____
Fall '91 Proceedings (FA1)	_____	15.00	_____
Total Cost of Subscription			\$ _____

☐ MasterCard® ☐ VISA® ☐ DINERS CLUB® ☐ CARTE BLANCHE® ☐ American Express®

Card No. _____ Exp. Date _____

I understand that there will be no refunds, even if I decide to cancel my subscription.

Signature _____

For Digital Employee Use Only

For DECUS Office Only

Badge No. _____ CC: _____

Check No. _____

CC Mgr Name _____

Bank No. _____

CC Mgr Signature _____

Amount \$ _____

Send to: Subscription Service, DECUS, 333 South Street (SHR1-4/D30), Shrewsbury, MA 01545-4112, (508) 841-3389.

Note: DECUS Europe and DECUS GIA members should contact their local DECUS office.



SD891205
MR-4829-RA

Reader's Comments

We welcome your comments and suggestions regarding the *ALL-IN-1 Information Update*. They will help us in our continuing effort to improve the quality and usefulness of the Update. Please take a few minutes to complete this form and return it to us.

What do you think of the Update content?

- ☐ Too technical
- ☐ Not technical enough
- ☐ About right

How often do you refer to the Update?

- ☐ Occasionally
- ☐ Regularly
- ☐ Only to solve a problem

Is the Update of general use to you? (explain)

- ☐ Yes _____
- ☐ No _____

What other information would you like to see in the Update? _____

Please outline any errors you have found in the Update or any suggestions for improvement. _____

(Attach additional pages, if necessary)

Additional comments: _____

How many years experience do you have in the following?

- _____ ALL-IN-1
- _____ EDT and/or WPS editor
- _____ WPS-PLUS editor
- _____ Other office automation systems
- _____ Other word processing systems

I am/am not* willing to have you contact me by mail/telephone* regarding my comments and suggestions.

Name: _____

Title: _____

Company: _____

Street: _____

City: _____ State: _____ Zip Code: _____

Country: _____ Telephone: _____

*Delete as applicable.

Do Not Tear — Fold Here and Tape

digital™



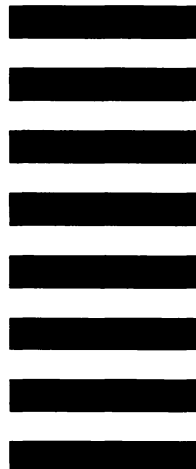
No Postage
Necessary
if Mailed in the
United States

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 33 MAYNARD, MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

Susan A. Thompson
Editor, ALL-IN-1 Information Update
Digital Equipment Corporation
Corporate User Information Products (MR01-3/T4)
P. O. Box 1001
Marlborough, MA 01752-9840



Do Not Tear — Fold Here and Tape