

UNISYS

**ALLY®
Software
Development
Environment**

**ALLY U Series
Installation
Guide**

Copyright © 1987 Unisys Corporation.

All rights reserved.

Unisys is a trademark of Unisys Corporation.

ALLY is a registered trademark of
Foundation Computer Systems, Inc.

Foundation Computer Systems is
a wholly owned subsidiary of Unisys Corporation.

June 1988

Priced Item

Printed in U S America
UP-14221

Notice

Foundation Computer Systems (Foundation) has written this manual for use by Foundation customers. The information contained in this manual shall not be reproduced in whole or in part without Foundation's prior written approval.

Foundation reserves the right to make changes in specifications and other information contained in this manual without prior notice. The reader should, in all cases, consult Foundation to determine whether any such changes have been made.

ALLY is a registered trademark of Foundation Computer Systems, Inc.

C-ISAM is a trademark of Informix Corporation.

ORACLE is a registered trademark of Oracle Corporation.

SVT-1220 is a trademark of Unisys Corporation.

UNIFY is a trademark of Unify Corporation.

UNIX is a trademark of AT&T Bell Laboratories.

Foundation Computer Systems is a wholly-owned subsidiary of
Unisys Corporation.

© Copyright 1988 by Foundation Computer Systems, Inc.
All rights reserved

Preface

This manual describes ALLY release 2.0.

This guide provides information about how to install ALLY on Unisys System V UNIX operating systems for U Series computers.

This guide has five appendixes. The first appendix contains the correct set-up values for the SVT-1220 and UVT-1224 terminals. The remaining four appendixes contain information about configuring specific systems.

Audience

This guide is written for the person who installs ALLY on the computer system, such as a systems analyst, site administrator, or data processing manager. The installer should understand the UNIX operating system and other software applications already installed on the system.

Prerequisites

We assume that you have a thorough knowledge of UNIX commands and syntax. If you are installing the ALLY Development System, you can refer to the *ALLY System Manager's Guide* (UP-13765) and the access method developer notes. Refer to other documentation for your computer system and terminals as needed.

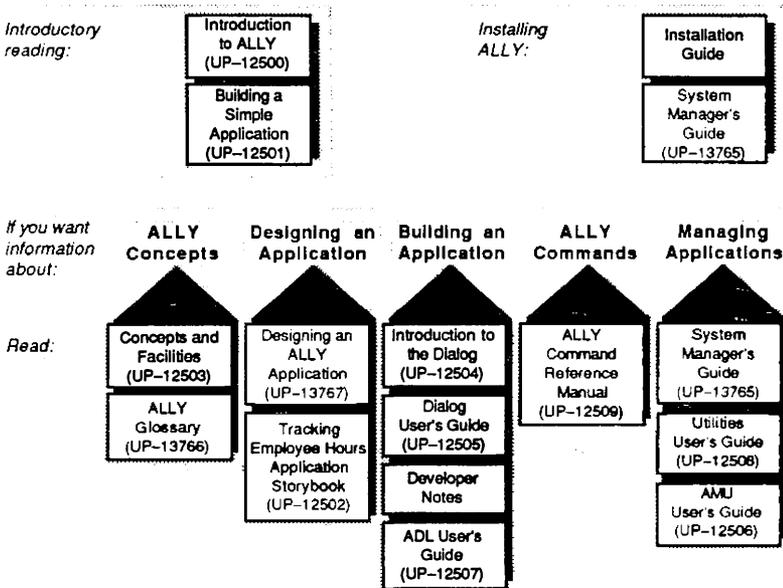
The ALLY Documentation—What to Read

The following illustration shows you how the ALLY documentation is organized.

If you want to develop ALLY applications, we suggest that you start by reading the ALLY system's introductory brochure, *Introduction to ALLY*. Then, you can build the application in *Building a Simple Application*.

If you want to install ALLY, you should read the installation guide for your system.

Note that the documentation for the runtime system of ALLY includes only the installation guide for your system, *ALLY Command Reference Manual*, and *AMU User's Guide*.



F002-0814-00

Conventions

You should read carefully the description of documentation conventions before reading this manual.

We use the following conventions in this manual:

- Single quotes (' ') Identify command names.
- Boldface type (bold)** Highlights text you are to enter. Boldface is also used within command syntax statements.
- Double quotes (" ") Identify text strings within text sections. These strings are typically located in examples or as part of the prompts that ALLY sends to your display.
- Sometimes the exact content of a text string is affected by the traditional rules of punctuation. In these cases, we place the closing quotation mark at the end of the text string. For example, instead of:
- You see the prompt "Macro number:."
- We say:
- You see the prompt "Macro number:".
- Brackets (<x>) Enclose a specific key (x = key) to be typed. Angle brackets are used in command syntax and key assignment lines. For example:
- | | |
|----------|---------------------------------------|
| <,> | the "comma" key on the numeric keypad |
| <Return> | the "Return" key |
| <Do> | the "Do" key |
| <F18> | the "function" key 18 |
| <2> | the "2" on the numeric keypad |
- Empty brackets (<>) Represent a form that requires input in menu path sequences, e.g., 1 2 <> 3 4.

- Space (<Space>) Represents a blank to be entered. A <Space> request asks you to type the space bar once.
- Square brackets ([]) Enclose an argument for the command-line invocation of a utility. For example, **newprint** *[printer name] [output file]*.

The key-definition files that are supplied with ALLY assign keys to several commands. For convenience, we assume that the following ALLY commands are assigned as shown.

<u>Command</u>	<u>Key Assignment</u>
'add new line'	< Return >
'up'	< ↑ >
'down'	< ↓ >
'right'	< → >
'left'	< ← >
'define macro'	(See template)

The < ↑ >, < ↓ >, < → >, and < ← > key assignments typically use a terminal's cursor-control keys.

End of Preface

Contents

ALLY Installation Guide

Installation Overview	2
Release Requirements	3
Disk Space Requirements	3
Memory Allocation	3
Step 1: Installing ALLY	6
Step 2: Establishing the ALLY Environment	8
File-Locking Table	8
ALLY Environment Variables	10
Access-Method Dependencies	13
ORACLE Environment Variables	13
UNIFY Environment Variables	14
Search Path	15
Step 3: Verifying the Installation	16
Verifying on the Runtime System	16
Verifying on the Development System	17
Trouble Shooting	18
Terminal Configuration Problems	18
Installation Verification Problems	19
Other ALLY Installation-Related Problems	19
ORACLE Problems	20

Appendix A. Setting Your Terminal

Terminal Configuration for ALLY	A-1
---------------------------------------	-----

Appendix B. Configuring the U7000 Series

Changing the File-Locking Table Size	B-1
--	-----

Appendix C. Configuring the U6000/50

Changing the File-Locking Table Size	C-1
--	-----

Appendix D. Configuring the U5000/85,90,95

Changing the File-Locking Table Size D-1

Appendix E. Configuring the U5000/30,35,50,55

Increasing the Process Memory Space and
Changing the File-Locking Table Size E-1

Appendix F. Files: .cshrc and .profile

Figures

1	Memory Allocation	4
F-1	“.cshrc” File Contents	F-1
F-2	“.profile” File Contents	F-1

Tables

1	Supported Access-Method Release Levels	1
2	ALLY Memory Management Buffer Allocation	6
3	Default File-Locking Table Size	9
4	ALLY Environment Variables	10
A-1	SVT-1220 and UVT-1224 Terminal Settings	A-1

ALLY Installation Guide

This document contains instructions for installing ALLY and establishing ALLY's working environment on Unisys System V UNIX operating systems for the following U Series computers:

- U7000/30,40,50,51,52
- U6000/50
- U5000/85,90,95
- U5000/30,35,50,55

Your tape is labeled with the system type.

ALLY supports several different access methods, including C-ISAM, ORACLE, and UNIFY. Table 1 lists the access-method release levels that are valid for use with ALLY 2.0.

Table 1. Supported Access-Method Release Levels

<u>Access Method</u>	<u>Release Level</u>
C-ISAM	3.1
ORACLE	5.1.17.4
UNIFY	4.0

Installation Overview

The installation process is divided into three major steps:

- 1) Installing ALLY with the Unisys Menu System. Refer to the *Menu System Guide* (UP-11763) for information about the menu system.
- 2) Establishing the ALLY working environment, which includes:
 - Checking the default maximum size of the file-locking table. Information about how to increase the maximum size of the file-locking table is system-specific and is provided in Appendixes B, C, D, and E.
 - Checking ALLY environment variable values and setting them if necessary.
For more information about environment variables, see the *ALLY System Manager's Guide* (UP-13765), or call your ALLY distributor.
 - Checking access-method dependencies and setting their associated variables if necessary. For further information about access-method dependencies, see the developer notes for your access method, or call your ALLY distributor.
 - Checking that the search path is correctly set.
- 3) Verifying that the ALLY installation is successful.

The next sections list requirements for installing ALLY and describe the installation procedure. The last section in this document is a trouble-shooting guide to help you determine why an installation does not verify, or why an application does not execute successfully.

We recommend that you read all of the instructions that are applicable to your site before you begin the installation procedure.

Release Requirements

Before starting the ALLY installation procedure, you must have:

- a unique group for the ALLY account with the group name “ally,” in lowercase letters.
- a unique user account for ALLY. The user login name must be “ally,” in lowercase letters.
- enough space in the ALLY account file system. (The disk space requirements are listed in the following section.)
- a terminal that communicates with ALLY. Appendix A contains information about how to set the SVT-1220 and UVT-1224 terminals.

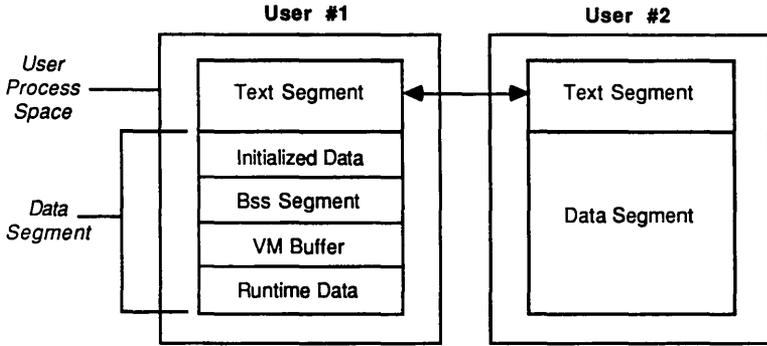
If this is your first ALLY release, you need to set up the ALLY group and user account. If you have installed a previous ALLY release, you already have an ALLY group and user account.

Disk Space Requirements

ALLY 2.0 requires approximately nine megabytes of disk space for the development system and three megabytes for the runtime system.

Memory Allocation

The amount of memory required to load an ALLY executable image is the sum of the bytes in the shared *text* segment and in the nonshared *data* segments, as shown in Figure 1. The combined text and data segment for each user is the *user process space*. The text segment contains the executable machine instructions. Data segments contain the initialized and uninitialized data for the executable image.



F002-0585-00

Figure 1. Memory Allocation

Once ALLY's text segment is loaded into memory, it is shared by all users. This means that all users can execute ALLY with only one copy of each executable image in memory.

The data segment is local for each user (i.e., not shared by other users). A part of the data segment is the *bss* segment, which contains uninitialized data for the executable image.

NOTE: To run ALLY on the U5000/30,35 or U5000/50,55 requires a user process space of at least two megabytes and preferably four megabytes.

See Appendix E, "Configuring the U5000/30,35,50,55," for information about how to increase the maximum process-memory space.

Minimum Memory

The minimum memory required to run ALLY is the executable image's total size plus ALLY's memory management buffers. The formula to calculate the executable image's total size is:

$$\begin{aligned} & \textit{text segment} \\ & + \textit{data segment (initialized)} \\ & + \textit{bss segment (uninitialized data)} \end{aligned}$$

For one ALLY user to invoke the Application Developer's Dialog requires one megabyte of memory. Each additional ALLY user requires about one fifth of a megabyte to invoke the Dialog.

ALLY's Memory Management Buffers

Each ALLY memory buffer is the equivalent of 512 bytes of memory. The number of memory buffers assigned to ALLY affects the read and write operations for AFILES, external symbol table files, FX files, and ALLY temporary work files. In general, the more ALLY memory buffers assigned, the less often data has to be written to and read from disks.

The AFILE defines the initial memory request, which allocates the number of ALLY internal memory management buffers, called virtual memory buffers. Subsequent memory requests made during the application's execution expand an ALLY user's data segment.

Memory and ALLY's Memory Buffers

To calculate the exact amount of physical memory required to run an ALLY application, add the executable image's total size to the number of memory management buffers multiplied by the number of bytes per buffer (number of buffers \times 512). The formula to calculate the executable image's total size is given in the "Minimum Memory" section.

Table 2 shows the number of ALLY memory management buffers assigned to run the listed AFILES.

Table 2. ALLY Memory Management Buffer Allocation

To Run	Plan On
The Dialog	300 memory buffers
The AMU	300 memory buffers
Application AFILES	300 memory buffers

Step 1: Installing ALLY

This section describes how to install ALLY with the Unisys Menu System. Commands that you are to type on your keyboard are shown in boldface print. Type the text exactly as shown and terminate commands by typing <Return>. Do not type the angle brackets—they simply indicate a key.

- 1) **Mount the ALLY installation tape onto the tape drive.** (Mount the tape onto drive zero if you have more than one drive.)
- 2) **Log-in to the Unisys menu system and follow the menu system's software installation instructions.** (You will select these choices: System Management, Install, and Software Product.)

The menu system prints several status messages on your terminal. Then, the ALLY installation command file verifies that the proper account and group exist.

If the ALLY installation command file cannot find the proper account or group, it:

- displays an error message
- terminates the installation process
- returns you to the menu system

If the account and group are appropriate, the following messages are displayed on your terminal:

```
Verifying ally account
OK

Verifying ally group
OK

ALLY will be installed in "directory path"

Type "Y<Return>" to continue the installation process or "N<Return>"
to terminate the installation process and return to the menu system. __
```

Comment: The words "directory path" are replaced by the path assigned to the ALLY login directory.

- 3) If you have previously installed any version of ALLY, the installation procedure displays the following:

Response: Do you want to remove ally_{x_}yz?

Type: **Y<Return>** to remove the other ALLY release, or

N<Return> to preserve an earlier ALLY release.

Comment: Each release is stored in a subdirectory named "ally_{x_}yz," where _{x_}yz is the release number. You do not need to remove the directory containing the other ALLY release.

Response: Removing ally_{x_}yz. Please wait...

- 4) When the installation process is complete, the following message is displayed:

ALLY installation is complete.
Before you use ALLY, you must establish the appropriate UNIX search path and environment variables as described in the "Step 2: Establishing the ALLY Environment" section of the ALLY Installation Guide.

You are then returned to the menu system, from which you can log-out.

- 5) Read the "Step 2: Establishing the ALLY Environment" section that follows for information about how to set the appropriate search path and environment variables.

Step 2: Establishing the ALLY Environment

This section briefly describes the following aspects of establishing the ALLY environment on UNIX systems:

- maximum size of the file-locking table
- ALLY environment variables
- access-method dependencies
- search path to the ALLY directory

File-Locking Table

ALLY locks application AFILES and at least six ALLY system files for each user. In addition, some access methods (e.g., C-ISAM) protect their file system with file locks. On some systems, the file-locking table may not be large enough to support multiple ALLY users. Depending on the number of users and the type of system at your site, you may need to increase the default maximum size of the UNIX file-locking table.

Each file lock is an entry in the file-locking table. The number of table entries is the “size” of the file-locking table. As shown in Table 3, the default maximum size of the file-locking table varies by machine.

Table 3. Default File-Locking Table Size

Unisys Computer	Operating System Release	Default Table Size
U7000/30,40,50,51,52	1R1, 1R2, 2.20.01	100
U6000/50	2.00.10.10	100
U5000/85,90,95	1R1, 2R1, 3R1, 4.0	30
U5000/30,35,50,55	1R1, 1R1A, 2R1, 1.03.xx	200

Calculating the File-Locking Table Size

To determine the appropriate maximum file-locking table size for your site, assume that each user needs to be able to lock twenty-five files. This is more than ample and provides a margin of safety. Multiply the number of file locks per user by the number of anticipated users to obtain the maximum number of file locks needed. If the number of file locks needed exceeds the default maximum table size (Table 3), you need to increase the table size.

See the appropriate appendix for your system for information about how to increase the default maximum size of the file-locking table.

- Appendix B For the U7000 series
- Appendix C For the U6000/50
- Appendix D For the U5000/85,90,95
- Appendix E For the U5000/30,35 and U5000/50,55

ALLY Environment Variables

Environment variables specify runtime information, such as paths to files. When an application executes, ALLY looks for the values of environment variables in the operating system's list of environment variable definitions.

Table 4 describes four operating-system environment variables that ALLY uses.

Table 4. ALLY Environment Variables

Variable	Value	Required?
ally	The path to the top-level ALLY directory	Yes
path	The path to the subdirectory that contains ALLY's command files	Yes
TERM	The name of a terminal type (e.g., svt1220 for the SVT-1220)	Yes
allyprinter	Overrides the default printer spooling device or queue name for ALLY output	No

You can use the following command to determine whether an environment variable is defined:

echo \$variable_name

Required environment variables must be set for each ALLY-user process.

“ally” Environment Variable

The value of the “ally” environment variable is the path to the ALLY release subdirectory in the ALLY login directory. The “ally” environment variable is used by:

- ALLY to construct file names for library AFILES and utility files (e.g., the terminal definition file, printer definition file, and Format File)
- users to invoke ALLY from their working directories

C shell users can use this command to set the “ally” variable:

```
setenv ally ~ally/allyx_ym
```

Bourne shell users can use these commands to set the “ally” variable:

```
ally=home directory/allyx_ym  
export ally
```

“path” Environment Variable

The value of the “path” environment variable is the sequence of directories to search for ALLY command files that invoke ALLY and its utilities. C shell users can use this command to set the “path” variable:

```
set path=($path $ally/bin)
```

Bourne shell users can use these commands to set the “path” variable:

```
PATH=:$PATH:$ally/bin:  
export PATH
```

“TERM” Environment Variable

ALLY needs a terminal description file for each type of terminal used with interactive ALLY sessions. The Terminal Definer utility produces the terminal description file, which tells ALLY what action to take when a key is typed. ALLY command files append a “c” to the value of the “TERM” variable and look for a terminal description file with this name.

You can override the “TERM” variable setting by specifying a different terminal description file as an argument to an ALLY invocation command. ALLY cannot run if it cannot find the terminal description file.

C shell users can use this command to set the “TERM” variable:

```
setenv TERM terminal_type
```

Bourne shell users can use these commands to set the “TERM” variable:

```
TERM=terminal_type  
export TERM
```

“allyprinter” Environment Variable

The environment variable “allyprinter” is not required to be set. However, it can be set to either “copy” or “none.”

C shell users can use either of these two commands to set the “allyprinter” variable:

```
setenv allyprinter copy  
or  
setenv allyprinter none
```

Bourne shell users can use either of these pairs of commands to set the “allyprinter” variable:

```
allyprinter=copy  
export allyprinter
```

or

```
allyprinter=none  
export allyprinter
```

See the *System Manager's Guide* for more information about ALLY environment variables.

Access-Method Dependencies

In addition to the environment variables described in the previous section, some access methods require you to set environment variables. Dependencies for ORACLE and UNIFY are described in the next two sections. If you are using another access method, you can skip these sections and continue with the “Search Path” section.

ORACLE Environment Variables

There are two ORACLE environment variables that must be set for ALLY to access ORACLE tables: ORACLE_HOME and ORACLE_SID.

ORACLE_HOME

ORACLE_HOME identifies the home directory containing the ORACLE product files. If you do not assign a home directory, ORACLE is assumed to be installed in the default directory named “/usr/oracle.” C shell users can use the following command to set this environment variable:

```
setenv ORACLE_HOME oracle_directory_name
```

Bourne shell users can use the following commands to set ORACLE_HOME:

```
ORACLE_HOME=oracle_directory_name  
export ORACLE_HOME
```

ORACLE_SID

ORACLE_SID is the ORACLE System Identifier that identifies a database. If you are using ORACLE's default production system, the ORACLE_SID number should be set to null. Use two double quotation marks (""") to indicate a null value (for example, ORACLE_SID="""). Each additional database system should have a unique letter for its ORACLE_SID.

C shell users can use the following command to set this environment variable to a null value:

```
setenv ORACLE_SID ""
```

Bourne shell users can use the following commands to set ORACLE_SID to null:

```
ORACLE_SID= ""  
export ORACLE_SID
```

If you have many ALLY users that access ORACLE tables, you may receive error messages indicating insufficient enqueues or open cursors. In this case, you may need to increase two parameter settings when the ORACLE database is initialized. See the "Trouble Shooting" section for more information.

UNIFY Environment Variables

There are two UNIFY environment variables that you may need to set. You need to set the "DBPATH" environment variable if your UNIFY files are not in the directory from which you execute ALLY. The value of "DBPATH" is the directory that contains your UNIFY files. You need to set the "DBNAME" environment variable if your UNIFY database file has a name that is *not* the default name (file.db).

These two UNIFY environment variables, DBPATH and DBNAME, can be set from your operating system shell.

If you are using the C shell, use the following syntax when setting these environment variables:

```
setenv [variable_name] [value]
```

If you are using the Bourne shell, use the following syntax:

```
variable_name=value  
export variable_name
```

See the *UNIFY Developer Notes* (UP-12969) for more information about accessing a UNIFY database with ALLY.

Search Path

The ALLY installation procedure creates “.cshrc” and “.profile” files in the ALLY account’s home directory. These files contain the commands that set the environment variables and the search path for invoking ALLY. The search path points to the directory that contains UNIX command files (scripts) that invoke ALLY and the ALLY utilities. See Appendix F for the contents of the “.cshrc” and “.profile” files.

Each ALLY-user process must use these environment variables and search path. C shell users can add this statement to their own “.cshrc” file:

```
source ~ally/.cshrc
```

A C shell user’s “.login” file may execute after the “.cshrc” file. Therefore, the “.login” file should not contain path definitions that conflict with those in the “.cshrc” file.

Bourne shell users can add the following statement to their own “.profile” file:

```
. ally_login_directory_path/.profile
```

Step 3: Verifying the Installation

The next two sections describe how to verify that the ALLY installation is successful, and that users can access the error and help library AFILES. Verification is accomplished by running an AFILE on both the Runtime System and the Development System. Follow the testing procedure for both systems if you are installing the Development System.

Verifying on the Runtime System

The command shown below runs an application named "HOURS.A" on the Runtime System. (The "HOURS.A" AFILE is included on the installation tape.)

ally \$ally/afiles/storybook/HOURS.A

- 1) From the application's main menu, select choice number 6 (a nonexistent choice). The path to the error AFILE has been established if the following message is displayed on the bottom of the menu:

You selected a choice that is not valid for this menu.

- 2) Next, while the error message is displayed, type the "Do" key to access the ALLY Command Menus. Select choice 1 and then press the key assigned to "help." The path to the help AFILE has been established if help information for task and action control commands is displayed on your screen.
- 3) Type <Return> three times to return to the Dialog's main menu (once to exit from the help facility, once to exit from the Command Menus, and once to exit from the error message). Now select the exit choice from the main menu to return to the operating system.

If the appropriate error and help messages are not displayed, read the "Trouble Shooting" section at the end of this guide.

Verifying on the Development System

Type the following command to invoke the Application Developer's Dialog:

dialog

Once you have invoked the Dialog, follow these steps:

- 1) When prompted, type the name of any new AFILE (e.g., TEST.A). This creates a new AFILE "skeleton," called a trunk AFILE, which you can delete after you verify the installation.
- 2) From the Dialog's main menu, select choice number 10 (a nonexistent choice). The path to the error AFILE has been established if the following message is displayed on the bottom of the menu:

You selected a choice that is not valid for this menu.

- 3) Next, while the error message is displayed, type the "Do" key to access the ALLY Command Menus. Select choice 1 and then press the key assigned to "help." The path to the help AFILE has been established if help information for task and action control commands is displayed on your screen.
- 4) Type <Return> three times to return to the Dialog's main menu (once to exit from the help facility, once to exit from the Command Menus, and once to exit from the error message). Now select the exit choice from the Dialog's main menu to return to the operating system.

If the appropriate error and help messages are not displayed, read the "Trouble Shooting" section that follows.

Trouble Shooting

This section describes some conditions that can prevent an installation from verifying or an application from executing as expected.

Terminal Configuration Problems

ALLY command files append a "c" to the value of the "TERM" variable and look for a terminal description file with this name. You receive an error message indicating that ALLY cannot open the terminal description file if either the "TERM" variable is not set or the specified file cannot be found:

You have specified the name of a file that is not a valid terminal description file.

If your terminal does not behave as expected, that is, if the commands you want do not execute when you type the keys, you do not have a terminal description file that matches your terminal type.

Type: <Esc> <t> <a> <p>

Comment: If your keyboard does not have an "Escape" key (<Esc>) type the "Control" key (<Ctrl>) and the "left square-bracket" (<[>) key to send the octal code for "escape." In this case, you would type the following key sequence:

<Ctrl><[><t><a><p>

Response: Exit from ALLY.

Once you have exited from ALLY, use the following command to see whether the value of the "TERM" variable matches your terminal type:

echo \$TERM

Set the "TERM" variable to your terminal type if it is not already set or is set incorrectly. If the "TERM" variable is correctly set, and you still experience problems with your terminal, Appendix A shows the proper Terminal Set-Up values for SVT-1220 and UVT-1224 terminals.

Installation Verification Problems

If you receive an error message stating that ALLY cannot access the help AFILE or the error AFILE, check that the "ally" environment variable is set to the ALLY account home directory and release subdirectory, as shown below:

ally/ally2_00

If you cannot execute ALLY command files (you receive an error message that says ALLY cannot access the specified file or directory), check that the "path" environment variable includes the following ALLY directories:

ally/ally2_00/bin

Other ALLY Installation-Related Problems

Depending on your system and the number of ALLY users at your site, you may need to increase the default maximum file-locking table size. If the number of file locks available is not sufficient, you will receive an error message stating that the specified file cannot be opened. The file could be a terminal file or any AFILE, including help and error message AFILES.

ORACLE Problems

If you have many ALLY users accessing ORACLE tables, you may receive error messages indicating an insufficient number of enqueues and/or open cursors. In this case, you need to increase the settings of two parameters when you initialize the ORACLE database:

- enqueues** specifies the maximum number of total enqueue entries. The default value is 125, which is five times the number of default ORACLE processes. Each database cursor requires one enqueue for each table accessed or updated, and each row locked.
- open_cursors** specifies the maximum number of cursors that each user is allowed to open. An open cursor is needed for each insert, update, delete, and select statement from a DSD. The value of **open_cursors** can be any number from 10 to 255; the default value is 50.

Refer to your ORACLE documentation, particularly the *ORACLE Database Administrator's Guide*, for more information about these two parameters.

End of ALLY Installation Guide

Appendix A

Setting Your Terminal

Terminal Configuration for ALLY

This appendix contains information for configuring SVT-1220 and UVT-1224 terminals to communicate with ALLY. (The SVT-1210 terminal does not need to be set specifically to run with ALLY.) Only settings that apply specifically to ALLY are shown. See the documentation supplied with your terminal for more information.

Table A-1. SVT-1220 and UVT-1224 Terminal Settings

Category	Setting
Display	80 Columns Interpret Controls 60 Hz.
General	VT200 Mode, 7-Bit Controls User-Defined Keys Unlocked User Features Unlocked Clear User-Defined Keys Numeric Keypad Normal Cursor Keys No New Line Multinational

Appendix A

Category	Setting
Communications	Transmit=9600 Receive=Transmit XOFF at 64 8 Bits, No Parity 1 Stop Bit No Local Echo Main Port, Data Leads Only Disconnect, 2 s Delay Unlimited Transmit
Enhancements	Main Received XOFF=Implement Shift User-Defined F-Keys VT100 F-Keys=ESC/BS/LF Only Space Suppression On Transmit Area=Screen Immediate Transmit

End of Appendix A

Appendix B

Configuring the U7000 Series

Changing the File-Locking Table Size

NOTE: This process requires you to shut down the operating system. Make sure that no one is using the system when you do this.

Follow the steps below to change the default maximum file-locking table size on U7000 series systems.

- 1) Log-in to the system as the root (super user).
Type: **root<Return>**
- 2) Access the system configuration subdirectory.
Type: **cd /usr/src/uts/machine/cf<Return>**
- 3) An option in the system configuration file allows you to override the default maximum table size. To use this option, you first need to change the file "space.c."

Locate the following lines in the file named "space.c."

```
/* file and record locking */
#define   FLOCKREC   100   /* # lock structures */

struct flockinfo flockinfo={
    FLOCKREC,
```

Edit the “space.c” file and change the lines shown above to the following:

```
/* file and record locking */
#ifdef FLCKREC
#define FLCKREC 100 /* # lock structures */
#endif
struct flckinfo flckinfo={
    FLCKREC,
```

- 4) Enter the following command.

Type:

```
echo "options FLCKREC=table_size" >> NCF<Return>
```

Comment: Enter the estimated maximum number of file locks needed at your site for *table_size*.

- 5) Make a new UNIX kernel by entering the following series of commands.

Type:

```
cat CF >> NCF<Return>
```

```
config NCF<Return>
```

```
make -f cf.mk<Return>
```

```
cd ..<Return>
```

```
cp unixX /<Return>
```

Comment: Your new kernel may be called “unixX,” where *X* is one of the following: “unisys,” “cf,” “sperry,” or *site_name*.

- 6) Shut down the system to single-user mode with the following command.

Type: **/etc/shutdown**

Comment: You are prompted for verification when the system shuts down.

- 7) Log-in to the system as the root (super user).
Type: **root<Return>**
- 8) Save the old UNIX kernel (in the "unix" file) by copying it to another file ("unix.old"). Then create a new UNIX kernel by copying your new kernel to the "unix" file.
Type: **mv unix unix.old<Return>**
 mv unixX unix<Return>
- 9) Type the 'sync' command three times to flush the file system buffers to the disk.
Type: **sync; sync; sync;<Return>**
- 10) Reboot the system.

End of Appendix B

Appendix C

Configuring the U6000/50

Changing the File-Locking Table Size

NOTE: This process requires you to shut down the operating system. Make sure that no one is using the system when you do this.

Follow the steps below to change the default maximum file-locking table size on the U6000/50 system.

- 1) Log-in to the system as the root (super user).

Type: **root**<Return>

- 2) Access the system configuration subdirectory.

Type: **cd /usr/sys/build/SPC200**<Return>

etc/master

- 3) Make a copy of the original system configuration file (dfile). The following statement copies the system configuration file to a file named "dfile.orig."

Type: **cp dfile dfile.orig**<Return>

- 4) Enter the following command.

Type: **echo "nfllocks table_size" >> dfile**<Return>

Comment: Enter the estimated maximum number of file locks needed at your site for *table_size*.

- 5) Make a new UNIX kernel.

Type: **make -f unix.mk**<Return>

Response: The new kernel is placed in the file named "unix."

- 6) Shut down the system to single-user mode.
Type: **/etc/shutdown<Return>**
Comment: You are prompted for verification when the system shuts down.
- 7) Log-in to the system as the root (super user).
Type: **root<Return>**
- 8) Access the system configuration subdirectory.
Type: **cd /usr/sys/build/SPC200<Return>**
- 9) Enter the following command to copy the new kernel to the root directory.
Type: **cp unix /unix.new<Return>**
Response: The new kernel is copied to the root directory in the file named "unix.new."
- 10) Save the old UNIX kernel (in the "unix" file) by copying it to another file (unix.old). Then create a new UNIX kernel by copying "unix.new" to the "unix" file.
Type: **mv unix unix.old<Return>**
 cp unix.new unix<Return>
- 11) Type the 'sync' command three times to flush the file system buffers to the disk.
Type: **sync; sync; sync;<Return>**
- 12) Reboot the system.

End of Appendix C

Appendix D

Configuring the U5000/85,90,95

Changing the File-Locking Table Size

NOTE: This process requires you to shut down the operating system. Make sure that no one is using the system when you do this.

Follow the steps below to change the default maximum file-locking table size on U5000/85,90,95 systems.

- 1) Log-in to the system as the root (super user).

Type: **root<Return>**

- 2) Access the system configuration directory and enter the following series of commands.

Type: **cd /usr/sys/cf<Return>**
cp system system.old<Return>
echo "locks table_size" >>
system<Return>

Comment: Enter the estimated maximum number of file locks needed at your site for *table_size*.

- 3) Enter the following commands to make a new UNIX kernel.

Type: **cd ..<Return>**
make -f local.mk NODE=site_name
<Return>
cp site_name.MMDD /<Return>

Comment: *MM* is a two-digit number that specifies the month. *DD* is a two-digit number that specifies the day of the month.

- 4) Shut down the system to single-user mode.
Type: **/etc/shutdown<Return>**
Comment: You are prompted for verification when the system shuts down.
- 5) Log-in to the system as the root (super user).
Type: **root<Return>**
- 6) Enter the following command.
Type: **In site_name.MMDD syst<Return>**
Comment: The first letter in "In" is lowercase L. MM is a two-digit number that specifies the month. DD is a two-digit number that specifies the day of the month.
- 7) Type the 'sync' command three times to flush the file system buffers to the disk.
Type: **sync; sync; sync;<Return>**
- 8) Reboot the system.

End of Appendix D

Appendix E

Configuring the U5000/30,35,50,55

You must increase the maximum process memory space to run ALLY on U5000/30,35 and U5000/50,55 systems. ALLY requires a process memory space of at least two megabytes and preferably four megabytes. Depending on the number of users at your site, you may also need to increase the default maximum size of the file-locking table.

Increasing the Process Memory Space and Changing the File-Locking Table Size

Follow the steps outlined below to increase the maximum process memory space and reconfigure your system. These instructions include an optional step if you need to increase the maximum size of the file-locking table. See the *Administrator Reference Manual* (UP-11761) for further details.

NOTE: This process requires you to shut down the operating system. Make sure that no one is using the system when you do this.

- 1) Log-in to the system as the root (super user).
Type: **root<Return>**
- 2) Access the configuration subdirectory.
Type: **cd /kernel/sperry/cf<Return>**
- 3) Make a copy of the original configuration file. This statement copies the "5.2.cf" file to a file named "5.2.cf.orig."
Type: **cp 5.2.cf 5.2.cf.orig<Return>**

- 4) Modify the configuration file. This statement edits the "5.2.cf" file and adds the line "maxspace 4194304" in part 2.

Type: **echo "maxspace 4194304" >>
5.2.cf<Return>**

- 5) Do this step only if you need to increase the maximum size of the file-locking table. (If you do not need to change the file-locking table size, skip this step and proceed with step 6.)

Modify the file named "/usr/include.20/sys/inode.h" by changing the value of "NFLOCKS" from 200 to the estimated maximum number of file locks needed at your site.

- 6) Make a new UNIX file.

Type: **cd /kernel<Return>
make unix<Return>**

Response: The new UNIX file is placed in "kernel/sperry/unix5.2."

- 7) Enter the following command to copy the file to the root directory.

Type: **cp unix5.2 /**

- 8) Shut down the system to single-user mode.

Type: **/etc/shutdown<Return>**

Comment: You are prompted for verification when the system shuts down.

- 9) Log-in to the system as the root (super user).

Type: **root<Return>**

- 10) Make a backup copy of the old UNIX file and then overwrite the old file with the UNIX file just created.

Type: **mv unix unix.old
mv unix5.2 unix**

- 11) Type the 'sync' command three times to flush the file system buffers to the disk.

Type: **sync; sync; sync;<Return>**

- 12) Reboot the system.

End of Appendix E

Appendix F

Files: .cshrc and .profile

Figure F-1 shows the contents of the “.cshrc” file.

```
#
# Establish csh environment for ALLY.
# Set up search path to scan ALLY's bin directory,
# set ALLY's environment variable ->ally<- so ALLY knows where things live.
#
#
echo $path | grep -s /g/customer/ally/ally2_00 >& /dev/null
if ( $status = 1 ) then
#
# Define ALLY's environment variable.
#
    setenv ally /a/ally2_00
    set path = ( $path $ally/bin )
endif
#
# Define shorter alias for allyedit.
#
alias ae allyedit
```

Figure F-1. “.cshrc” File Contents

The “.profile” file is shown in Figure F-2.

```
ally=/g/customer/ally/ally2_00
export ally
PATH=./g/customer/ally/ally2_00/bin:$PATH:
export PATH
```

Figure F-2. “.profile” File Contents

End of Appendix F

Index

- Access methods, 1
- ALLY memory buffers, 5
- “ally” environment variable, 10
- “allyprinter” environment variable, 10
- Bss segments, 4
- “.cshrc” file, F-1
- Data segments, 3, 4
- DBNAME, 14
- DBPATH, 14
- Disk space requirements, 3
- Enqueues, 20
- Environment variables, 10
 - “ally”, 10
 - “allyprinter”, 10
 - DBNAME, 14
 - DBPATH, 14
 - ORACLE_HOME, 13
 - ORACLE_SID, 14
 - “path”, 10
 - TERM, 11
 - UNIFY, 14
- File lock, 9
- File-locking table, 8
 - calculating size of, 9
 - changing size of, B-1, C-1, D-1, E-1
 - default maximum size, 9
- Installation
 - verification problems, 19
 - verifying on development system, 17
 - verifying on runtime system, 16
- Installation-related problems, 19
- Memory buffers
 - ALLY, 5
 - virtual, 5
- Open_cursors, 20
- ORACLE environment variables, 13
 - ORACLE_HOME, 13
 - ORACLE_SID, 14
- ORACLE parameters
 - enqueues, 20
 - open_cursors, 20
- ORACLE problems, 20
- ORACLE_HOME, 13
- ORACLE_SID, 14
- Parameters
 - enqueues, 20
 - open_cursors, 20
- “path” environment variable, 10
- Process memory space, increasing, E-1
- “.profile” file, F-1
- Requirements, 3
 - disk space, 3
 - release, 3
- Search path, 15
- Segments
 - bss, 4
 - data, 3, 4
 - text, 3, 4
- Table, file-locking, 8
- TERM environment variable, 10
- Terminal configuration, A-1
 - problems with, 18
- Text segments, 3, 4
- UNIFY environment variables, 14
 - DBNAME, 14
 - DBPATH, 14
- Variables, environment, 10
- Virtual memory buffers, 5

End of Index

