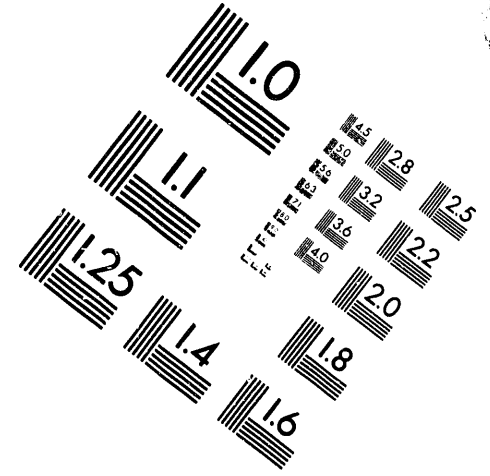
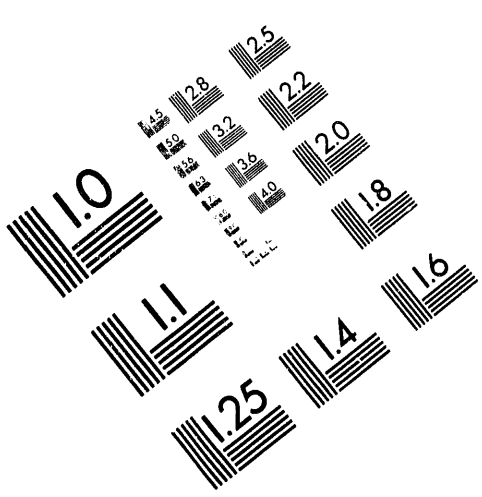




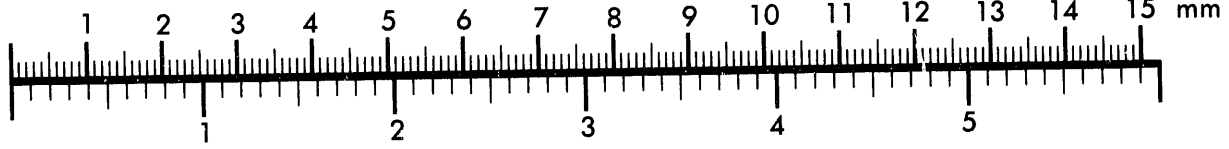
AIM

Association for Information and Image Management

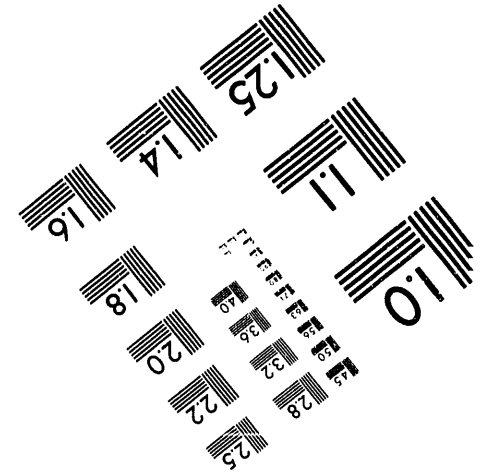
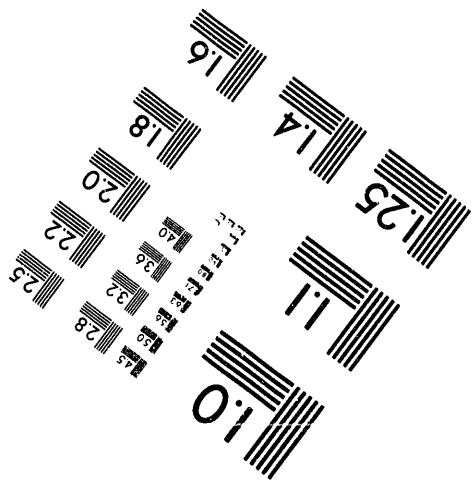
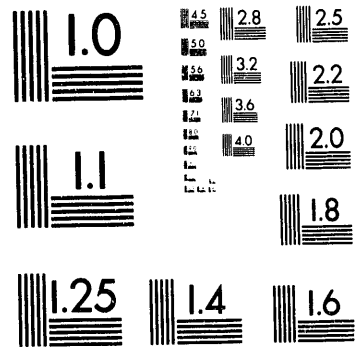
1100 Wayne Avenue, Suite 1100
Silver Spring, Maryland 20910
301/587-8202



Centimeter



Inches



MANUFACTURED TO AIM STANDARDS
BY APPLIED IMAGE, INC.

1 of 1

UCRL-JC-110007
PREPRINT

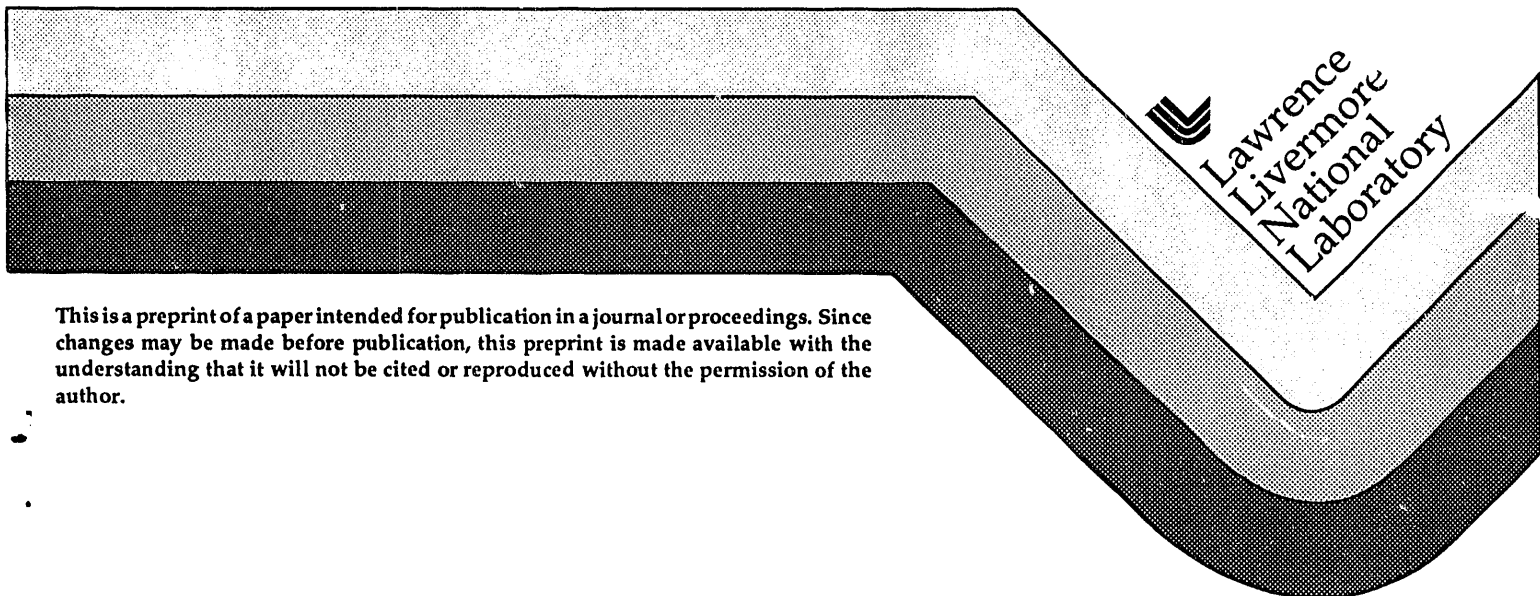
Training End-users An On-line Help Package

Marlene Waite

This paper was prepared for submittal to the
Ingres World '92
North American Ingres User Association
Tarpon Springs, Florida
May 17-21, 1992

March 1992

RECEIVED
JUN 11 1993
OSTI



This is a preprint of a paper intended for publication in a journal or proceedings. Since changes may be made before publication, this preprint is made available with the understanding that it will not be cited or reproduced without the permission of the author.

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

Abstract

No matter how wonderful your INGRES application, if the end users don't know how to use the application it will not be utilized. This paper discusses an on-line help package implemented in our manufacturing database application in order to help our user community use the application.

The Materials Fabrication Division (MFD) of Mechanical Engineering is a series of machining shops that do production jobs for the Laboratory. The Fabrication Control System (FCS) is the user interface to an INGRES database. FCS tracks jobs as they traverse the machine shops and FCS maintains the employee payroll. This interface consists of VIFRED forms, menus and EQUQL Fortran code.

The on-line help package is an interactive system that responds to the users "single key stroke" request for information. The information retrieved is based upon which help key is used and the form or menu from which the request is generated.

Introduction

The importance of the user community is sometimes lost in the day to day efforts to develop applications and fight brush fires. This presentation discusses an on-line help application used at Lawrence Livermore National Laboratory in the Materials Fabrication Division (MFD). The philosophies of user training and help keys are presented in this paper. There will also be code and screen examples to guide you in your own development of a help key application.

MFD Background

The Materials Fabrication Division (MFD) of Mechanical Engineering is a series of machining shops that do production jobs for the Laboratory programs. MFD has existed since the beginning of the Laboratory (30+ years ago). Originally, all records of jobs were kept on paper. The next generation of information storage and organization was on a Singer computer and in 1985 three Vax 785s were purchased and the Ingres database and interface that is used today was born.

MFD performs machining jobs that cannot be sent outside the Laboratory due to schedule, classification, hazard, etc. They are self supporting, depending upon the jobs in the shops to cover operating expenses.

*This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

FCS Background

The Fabrication Control System (FCS) is the user interface to an INGRES database. FCS tracks the production jobs as they traverse the machine shops and FCS maintains the employee payroll. This interface consists of VIFRED forms, menus and EQUOL Fortran code.

The Fabrication Control System (FCS) was developed in 1986 with Ingres 5 on a VAX/VMS environment. FCS was developed on the MFD business rules of 1986. Incorporated into this structure was the employee effort, which is based upon charges to jobs in the shops. Since all the employee effort is contained in the database, the payroll process was also coded into FCS and a weekly payroll process is run.

FCS also provides audit information on job performance that is reviewed by outside auditors. Finally, FCS contains a variety of reports used by the program, shop managers, and shop customers.

Computing Environment

FCS was created with Ingres5/04, VMS 4.7 on a cluster of three VAX 785s which contained a development and production installation.

Today we are at Ingres6.3/03, VMS5.4. We just recently upgraded to a cluster on two VAX 4300s for production and development.

Training End Users Philosophy

The user is often resistant to using the computer. This dictates the need to make operating information as easily obtainable as possible. We do have an introductory document for new users but the Help Key technique seems to be a painless method of instruction. They may not even realize that you are trying to teach them anything.

The best philosophy is to keep it consistent. We have rules and templates for everything from code format, screen layout and report headers, to maintain consistency throughout the facility. This helps the user to make an intelligent guess as to what to try in a new situation.

What's the next step? If you have a user community that enjoys working at the computer, an on-line problem reporting mechanism could be a useful communication and organization tool. I would certainly be interested in any information that you may have on such an installation in your base of operations.

Help Key Philosophy

The best approach to any type of help facility is to "keep it simple" and to "be consistent". Our user interface is made up of forms containing fields that correspond to our table elements or menus with brief explanations, with command options at the bottom of the screen.

Two help keys are used to explain all of the above. One key is the 'Field Help' and the other is the 'Screen Help'. Two keys are all the keys that the users need to remember. These same two keys can be used from all screens. That is keeping it simple and consistent.

Screen Help

Menugen Utility

Production Database

The MENUGEN Utility menu displays the list of available options that can be selected from this screen. To select an option, type the appropriate initials and press [RETURN] or press the indicated key on the numeric keypad. Each option is explained in further detail below.

- | | |
|----------|---|
| MACHSERV | - This option, when selected, will display a menu of all the Machine Tool Services Data Facilities. |
| TOOLCRIB | - Select this option when you wish to see the Tool Crib menu. All available Tool Crib options will be displayed after you select this option. |
| FCS | - This covers the FCS support utilities required to maintain the underlining FCS structure. |

Press [HELP SCREEN] to return

Field Help

Single Item - Single Route

Work Order: Initiate

WO Number: Status: **Initiate Date:** Req Due Date: Category:

Cust Id: Name: Phone Mailstop: Org.: WBS #: Ref #: Acct# - 100% Acct#: - % Acct# - % Acct#: - %

Overtime Authorization: Max Overtime Hours: Customer Group:

Rep #: Name: Ext: Lead Shop:

Item #: Drawing #: Quantity: Priority Class: Mat Hazard: Mat Source: Estimate:

Item Description: Item Comments: Task Description: Task Comments:

Date work order is initiated (online) by MFD.

VMS Screen Management (SMG)

The original FCS programming team developed the Integrated Data Dictionary and the On-line Help Facility (IDD) in 1986 under Ingres 5. The package does not incorporate Ingres 6 tools because it has not been updated since origination (more on this later). The original IDD developer used the VMS Screen Management (SMG) run-time libraries to maintain and manage the help screens. The SMG\$ libraries build the help screens. The libraries position the help information on the help screens and they place informational prompts on the help screens.

The Ingres "activate keyname" command from EQUOL Fortran code defines the help keys. The 'Field Help' key retrieves the form name and field name of the displayed screen upon help request. The help text is based upon the form and field name. If there is no entry in the database table for that form a generic text description is retrieved on the field alone with a blank form entry in the database table retrieve statement. If there is no matching entry in the

database table a message is displayed to the users that no 'Field Help' information is available.

Help Text Retrieve

```
CALL S_DisplayInformationMessage (26) !Processing request for HELP
C...
C Get Help Text -- Form Specific
C...
## RETRIEVE (f_fldtxt = FLDIDD.fld_text) WHERE FLDIDD.form=
## f_form AND FLDIDD.fld = f_fld)

## INQUIRE_INGRES (rwcnt = 'rowcount')
C...
C If no Form Specific Text exists get GENERIC TEXT DESCRIPTION
C...
IF (rwcnt .EQ. 0) THEN
## RETRIEVE (f_fldtxt = FLDIDD.fld_text) WHERE (FLDIDD.form = ''
## AND FLDIDD.fld = f_fld)

## INQUIRE_INGRES (rwcnt = 'rowcount')
C...
C If no Help Text was found give a 'sorry' message.
C...
IF (rwcnt .EQ. 0) THEN
CALL S_DisplaySleepingMessage (28) ! no [HELP FIELD] avail.
RETURN
ENDIF
ENDIF
```

The following is a code excerpt of SMG\$ library commands that write the 'Field Help' text to screen.

```
CALL S_Virtual_Keyboard (istat, 'field')
istat = SMG$HOME_CURSOR (fldhlpid,)
istat = SMG$PUT_LINE (fldhlpid,f_fldtxt(1:80),,,,,,)
istat = SMG$PUT_LINE (fldhlpid,f_fldtxt(81:160),,,,,,)
istat = SMG$PUT_LINE (fldhlpid,f_fldtxt(161:240),0,,,,,)
istat = SMG$PASTE_VIRTUAL_DISPLAY (fldhlpid,boardid,22,1,)
C ...
C Wait for the 'Field Help' key to be pressed again to exit help
C...
yesno = -1
DO WHILE (yesno .NE. endtype .AND. yesno .NE. help_code)
istat = SMG$READ_KEYSTROKE (keypad,yesno,,,,,)
C...
C If user hits the HELP KEYPAD then keypad help is displayed and the
C 'Field Help' utility is exited.
```

```

C...
      IF (yesno .EQ. help_code) THEN
          istat = SMG$UNPASTE_VIRTUAL_DISPLAY (fldhlpid,boardid)
          CALL Help_Keypad (ddic,endtype)
          CALL S_DelVirtualKeyboard (istat,'field')
          RETURN
C...
C Tell the user to press [HELP FIELD] and then unpaste the error
C message from the pasteboard.
C...
      ELSE IF (yesno .NE. endtype) THEN
          istat = SMG$UNPASTE_VIRTUAL_DISPLAY (errid,boardid,24,1,)
##          SLEEP 2
          istat = SMG$UNPASTE_VIRTUAL_DISPLAY (errid,boardid)
      ENDIF
ENDDO

```

Database Help Tables

There are four Ingres tables in the database that contain the help text. These four tables are: SCRIDD, FLDIDD, MENUIDD, and OPTIDD.

SCRIDD

This table contains help text for every form in the INGRES forms system. There is no limit to the number of pages that may be used to describe any particular form.

```

##      CREATE SCRIDD (
##          form          = c12 NOT NULL WITH DEFAULT,
##          seq_number   = i2  NOT NULL WITH DEFAULT,
##          scr_text     = text(1840) NOT NULL WITH DEFAULT
##      )

```

FLDIDD

This table contains the help text for every field displayed on any application screen. This help is limited to three lines of eighty characters.

```

##      CREATE FLDIDD (
##          form      = c12 NOT NULL WITH DEFAULT,
##          fld       = c12 NOT NULL WITH DEFAULT,
##          fld_text  = text(240) NOT NULL WITH DEFAULT
##      )

```

MENUIDD

This table contains six lines of help text for each menu in FCS. This gets combined at run time with the necessary help text for each menu option to create a page of help for the menu screen being displayed.

```

## CREATE MENUIDD (
##     menu_name = c10 NOT NULL WITH DEFAULT,
##     menu_id   = i2  NOT NULL WITH DEFAULT,
##     menu_text  = text(480) NOT NULL WITH DEFAULT
## )

```

OPTIDD

This table contains three lines of help for each option entry in FCS. The text from MENUIDD combines, at runtime, to create a page of menu screen help for the menus dynamically generated in FCS. This must be dynamic because the menu options displayed in FCS are dependent upon the VMS login account.

```

## CREATE OPTIDD (
##     menu_name  = c10 NOT NULL WITH DEFAULT,
##     menu_id    = i2  NOT NULL WITH DEFAULT,
##     option_name = c10 NOT NULL WITH DEFAULT,
##     option_text = text(240) NOT NULL WITH DEFAULT
## )

```

Maintaining the IDD Tables

A single utility (IDD_Utility) maintains the contents of the IDD tables. With this utility the DBA can update or append new information to the help tables.

VIFRED forms and an EQUOL Fortran code build this utility. The design is exactly like our FCS utility, starting with a MainMenu that allows the DBA to choose which table to update through a MainMenu selection.

The DBA selects a table from the MainMenu. A selection form is displayed. The table field on the selection form contains information from the system catalog ii_objects. The information is retrieved from the ii_objects table with the Ingres LOADTABLE command. The UNLOADTABLE command updates the table field "help available" column. This column has a value of 'Y' if there is a corresponding entry in the database help table.

The DBA then selects the item to be updated. The update form is displayed with the appropriate information. If there is no help available the help table is appended otherwise the help table information is replaced. The same process is used in the utility to maintain all the IDD help tables.

Ingres 6 Possibilities

The Knowledge Package and Rules in Ingres 6 seem like natural tools to implement help keys. Help keys could be mapped to call Ingres procedures that, once identifying current cursor location and screen identification would call

the help text information from the database. All this information could probably be displayed on popup screens.

Rules could be defined such that upon failure of a field verification a popup screen is triggered to explain what is needed in the specific field.

Conclusions

The user community must be trained in order for them to realize the value of the interface application that you may have just spent several months of person hours to produce. They need to know what this new tool can do for them. This is very important but perhaps easier at the initial release stage.

A help key mechanism is not the totality of end-user training but rather a reinforcing part of it. What is needed to maintain the viability of the existing tool is a procedure to train new users . If you do not consider these new users eventually no one will know how to use the tool and it will be set aside for something new.

Include one-on-one introductory sessions with new users and a new user document. If your application is for a business process, offer a business rule document as it is likely that your application will support these rules. Form an information group available to the users for quick response to requests and questions. To assist the information group, develop an on-line problem or request reporting mechanism.

It is necessary to have close communication with the user community. All of the above suggestions strive to that end. With open communication lines the user community will use your interface application to its limits.

**DATE
FILMED**

8 / 17 / 93

END