CRISP Automation Systems Documentation

Manual Number: MAN-DEBUG-INST-031

Subject: CRISP DEBUG 3.1 Installation and Use

Date Created: 08/25/2002 Revision 1.0 Date Revised: N/A

Effective Date: 08/25/2002

Supercedes: N/A

Revision History

Revision	Date Issued	Description	Author	Approved
1.0	08/25/2002	Original Issue	S. Quayle	K. Wild

Reference Documents:

N/A

Installation and Use

General

This manual provides a step-by-step installation procedure for CRISP DEBUG, Version 3.1 and later. This procedure can be used for a new installation or for updating an existing CRISP DEBUG system.

This manual also provides the necessary information to use CRISP DEBUG.

This manual is broken down into the following sections.

Section	Description
On-Line Release Notes	Defines the location of the release notes and how to access the notes.
Requirements	Defines the system requirements before installation of CRISP DEBUG.
Using VMSINSTAL	Defines the VMSINSTAL command procedure.
Installing CRISP DEBUG	Defines the procedures to install the CRISP DEBUG product.
Using CRISP DEBUG	Defines the procedures to use the CRISP DEBUG product.

NOTE

Before performing this procedure, verify that the system hardware and software has been properly installed per the appropriate OpenVMS documentation.

Refer to the Software Product Description (SPD) document for details on required OpenVMS software.

On-Line Release Notes

General

On-line release notes are provided with each distribution of CRISP DEBUG. When VMSINSTAL is invoked with the `OPTIONS N` parameter, a prompt to print or display the release notes appears during the installation.

After CRISP DEBUG has been installed, the release notes may be printed or typed by accessing the file `SYS$HELP:CRISPDEBUG031.RELEASE_NOTES`.

The release notes for each release of CRISP DEBUG are distributed for those users who are upgrading by several versions at once.

Requirements

General

Depending on the media and the system configuration, the installation of CRISP DEBUG takes several minutes. The following privileges and resources must be available before CRISP DEBUG can be installed.

- SETPRV privileges, or CMKRNL, WORLD, and SYSPRV privileges
- Free disk space as follows:
    - 35,000 blocks on the system disk
    - 35,000 blocks on the CRISP disk.
- An additional amount of disk space equal to the total of the previous numbers is required on the system disk during installation. This space will be released when the installation completes or otherwise terminates.
- A minimum of 32 MB (VAX) or 64 MB (Alpha) of physical memory.

CRISP/32 must be installed first before the CRISP DEBUG can be installed. The version of CRISP DEBUG must exactly match the version of CRISP/32. For example, CRISP DEBUG version B3.1-17 cannot be installed on a CRISP/32 version V3.1-17.

Using VMSINSTALL

Description

VMSINSTAL is the command procedure invoked to install CRISP DEBUG. Before invoking this command procedure, log into the system management account, username SYSTEM. Execution of the VMSINSTAL command procedure is shown in steps 1 and 2 of the Installing CRISP DEBUG sections of this manual.

Format

The VMSINSTAL command procedure has the following syntax.

```
$ @SYS$UPDATE:VMSINSTAL DEBUG031 ddcu: [OPTIONS N]
```

ddcu: Represents a device name where the distribution volumes are mounted for the CRISP DEBUG installation media, where `dd` is the device code, `c` is the controller code, and `u` is the unit number. The device name used in examples in this manual is `MUA0:`.

OPTIONS N Is an optional parameter used if you want to be prompted to display or print the CRISP DEBUG release notes. If omitted, you are not prompted to display the release notes.

How It Works

If you do not supply the first two parameters, VMSINSTAL prompts you for the product and device names. The syntax in this case would be as follows.

```
$ @SYS$UPDATE:VMSINSTAL
```

When VMSINSTAL is invoked, the following are checked.

- Whether you are logged in to a privileged account. It is recommended that you install

software from the system manager's account.

- Whether you have adequate quotas for installation. VMSINSTAL checks for the following quota values.
  - ASTLM = 24
  - BIOLM = 18
  - BYTLM = 18000
  - DIOLM = 18
  - ENQLM = 30
  - FILLM = 20

At this point in the installation, VMSINSTAL checks whether any user processes are running. If VMSINSTAL detects these processes, , you are prompted as to whether you want to continue the installation. Enter `YES` to continue or `<return>` to stop the installation.

Installing CRISP DEBUG

General

This section contains explains the installation procedure. Defaults, where available, appear in brackets throughout the installation procedure.

The installation procedure may be aborted at any time by entering CRTL/Y. If CRTL/Y is pressed, the installation procedure deletes all files it has created up to that point and returns you to the VMSINSTAL prompt.

Requirements Before Installation

Before CRISP DEBUG can be installed, the license product authorization key (PAK) called "`CRISP-SOURCE`" must be installed and loaded. If the license is not installed, the installation procedure will terminate.

The installation procedure determines whether or not CRISP/32 is installed on the system by the presence of system logical name `CRISP$`, and the presence of file `CRISP$DOC:CRISPREL.DAT`. These must be present for proper operation of the installation.

The version string in `CRISP$DOC:CRISPREL.DAT` must exactly match the version string of the CRISP DEBUG product. If they do not match, the installation procedure will terminate.

Warning

Installing CRISP DEBUG is not supported when installed on a production system. Even if the debugging options are not used, debugging code will be included in any logics that are created after installation.

To reverse the installation of CRISP DEBUG, do the following:

- `$ DELETE CRISP$LIB:*.%LB;*`
- Re-install CRISP/32.

Installation Procedure

Perform the following steps to install CRISP DEBUG.

Step 1. Log in to a privileged account.

```
<Ret>
Username: SYSTEM<Ret>
Password: <Ret>
$ SET DEFAULT SYS$UPDATE <Ret>
```

Step 2. Invoke VMSINSTAL

```
$ @VMSINSTAL DEBUG031 MUA0: OPTIONS N <Ret>

OpenVMS VAX Software Product Installation Procedure V7.3

It is 16-AUG-2002 at 19:30

Enter a question mark (?) at any time for help.

Are you satisfied with the backup of you system disk [YES]? <Ret>
```

Press the Return key if you are satisfied with the backup of your system disk; otherwise, enter NO to discontinue the installation procedure.

Step 3. Mount the first installation kit volume

```
Please mount the first volume of the set on MUA0:.
Are you ready? YES <Ret>

The following products will be processed:
```

```
CRISP DEBUG V3.1

Beginning installation of CRISP DEBUG V3.1 at 19:35

%VMSINSTAL-I-RESTORE, Restoring saveset A...
```

To continue the installation, mount volume 1 and enter YES.

Step 4. Select a release notes option

This step applies only if OPTIONS N was specified in step 2.

```
Release Notes Options:

    1. Display release notes
    2. Print release notes
    3. Both 1 and 2
    4. None of the above

Select option [2]: <Ret>
```

If option 1 is selected, VMSINSTAL displays the release notes. Terminate the display by pressing CRTL/C. At this time, VMSINSTAL copies the release notes to SYS$HELP.

If option 2 is selected, VMSINSTAL prompts you for a queue name (use this option only if a print queue has been setup on your system). Either enter a queue name or press the Return key to send the file to the default output print device. After printing the release notes, VMSINSTAL copies the release notes to SYS$HELP.

```
Queue name [SYS$PRINT]: <Ret>
```

If option 3 is selected, VMSINSTAL displays the release notes. Terminate the display by pressing CRTL/C. VMSINSTAL then prompts you for a queue name. Either enter a queue name or press the Return key to send the file to the default output print device. After printing the release notes, VMSINSTAL copies the release notes to SYS$HELP.

If option 4 is selected, VMSINSTAL does not display, print, or copy the release notes. It prompts you to decide whether to continue the installation.

```
Do you want to continue the installation [N]: Y <Ret>
```

Enter YES to continue the installation. If you enter NO, VMSINSTAL discontinues the installation.

NOTE

The version of the CRISP DEBUG release notes file installed by VMSINSTAL is labeled with the current product name and version number (e.g., CRISPDEBUG031.RELEASE_NOTES). Do not delete release notes for previous versions of CRISP DEBUG.

Step 5. Installation questions

This section shows the questions that are asked during installation.

```
Do you want to purge files replaced by this installation [YES]?
```

New versions of library files are provided during installation, but the old files will not be purged unless you answer YES to this question. The source files themselves are always replaced and purged, regardless of the answer to this question.

```
Continue with installation:
```

This question is asked after a warning is issued about not installing on a production system. There is no default value. If "No" is entered, the installation ends.

Step 6. End the installation procedure

When the installation procedure completes, the following message is displayed.

```
VMSINSTAL procedure is done at hh:mm
```

```
$
```

The distribution media (tape) may now be removed from the drive.

Using CRISP DEBUG

Logic Structure

The CRISP Compiler translates a user's CRISP/32 code (*.C32) into VAX Macro code. The macro code is then assembled (on VAX) or compiled (on non-VAX). The resulting object is usually linked against run-time libraries (RTL's), including the CRISP RTL (`CRISPRTL.EXE`) and the CRISP Logic RTL (`CRISPLOGICRTL.EXE`).

The only part of a CRISP logic that is not in the RTL's is the Logic Shell, which is a module named LGS$SHELL. When the logic is linked, that module is extracted from object library `CRISP$LIB:CLELIB.OLB`. The Logic Shell is the main program of the logic.

This is the reason for the requirement that logics be re-built for all CRISP/32 upgrades. If the CRISP Logic RTL changes, but the Logic Shell does not, the executable will not work correctly.

The CRISP DEBUG product allows a user to not only step through his logic (the macro code), but also step into the internals of CRISP/32 itself. This is why a license for the CRISP/32 source code is required.

The CRISP DEBUG product provides object libraries for all the necessary parts of CRISP/32, compiled with optimization disabled and full debugging capability. Because the Logic Shell provided by the CRISP DEBUG product has optimization disabled, installation of CRISP DEBUG on a production system is not supported.

Building Logics

When building a logic for debug, specify "`/LIB/DEBUG`" to the "Enter special Macro and link qualifiers" question. On non-VAX systems, specifying "`/DEBUG`" disables optimization in the macro assembler step.

Specifying just "`/LIB`" will cause the logic to link against the object libraries, but will not start the debugger.

Specifying just "`/DEBUG`" will cause the debugger to come up. However, the logic will be

linked against the RTL's. This will prevent stepping into the internals of the CRISP logic.

Preparing to Debug

If running on a system from a DECterm, and the character-cell debugger is desired, instead of the DECwindows debugger, execute the following command.

```
$ DEFINE/SYS DBG$DECW$DISPLAY " "
```

Create a terminal session with which the debugger will communicate. In that session, do the following:

```
$ SHOW TERMINAL
$ DEFINE/SYS DBG$INPUT x:
$ DEFINE/SYS DBG$OUTPUT x:
$ RUN CRISP$UTL:TERMHIBER
```

Where "x" is the device name of the session, as reported by the `SHOW TERMINAL` command. There will be no command prompt after the `TERMHIBER` command.

Starting a Logic with the Debugger

Install the database as usual (`$ DBINS INSTALL X`).

Start the logic as usual (`$ LGINS START X`). The debugger should come up on the debugging session. A message about "sharing the user process" is normal.

Viewing Source

To see the CRISP/32 source code, enter a command of "`$ SET SOURCE CRISP$SRC:`". When stepping into the macro code of the logic itself, enter a `SET SOURCE` command, specifying the directory holding the Macro code.

Cleanup

When done debugging, re-define system logical names `DBG$INPUT` and `DBG$OUTPUT` back to "`SYS$INPUT:`" and "`SYS$OUTPUT:`", respectively.

Special Notes

Unlike debugging normal processes, pressing Ctrl-Y will not cause the process to stop and return to the debugger. Instead, the `TERMHIBER` program will exit, and the command prompt and debugger will be interlaced on the screen. Run `TERMHIBER` to suppress the command prompt should this happen.

Because there is no way to break the logic's execution, make sure that at least one useful breakpoint is declared before entering the "go" command.

If the debugger does not start on the desired terminal, make sure that the CRISP username has the "SHARE" privilege as a default privilege. SHARE is needed to allow another process (the logic) to access the terminal.