[OpenVMS] Bugcheck, Quick Reference Component Article for OpenVMS
Last Technical Review: 16-FEB-2000

OVERVIEW:

When OpenVMS code detects an internal inconsistency, such as a corrupted data structure or an unexpected exception, it generates a bugcheck. If the inconsistency is not severe enough to prevent continued system operation, the bugcheck is considered NONFATAL and merely results in process deletion and an error log entry.

> Note:
> OpenVMS system crashes may be caused by either hardware or software.

If the error is serious enough to jeopardize system operation and data integrity, OpenVMS code generates a fatal bugcheck. This generally results in aborting normal system operation, recording the contents of memory to a dumpfile for later analysis, and rebooting the system.

This sequence of events can only occur if the system is configured correctly. If so, files containing information on the state of the system at the time of the crash, i.e.; dumpfile, errorlog file, and clue files, may help determine the cause.

Compaq provides tools to assist in the analysis of this Bugcheck information which are described in detail later in the article. Following is a brief description of these tools:

CCAT - "Compaq Crash Analysis Tool" is a repository of known and unknown crash patterns. (CCAT is the CANASTA replacement)

CCAT web GUI - Web based tool used to submit collected bugcheck data to CCAT for analysis and possible resolution.

AutoCLUE - Electronic Service Tool installed on customer systems which automatically transfers bugcheck information to CCAT for analysis and possible resolution.

CADC - "Crash Analysis Data Collector" is the replacement tool for AutoCLUE. CADC collects rule data from a system crash and attempts to match the data against a rule database on the customer system.

This article provides basic guidance in collecting crash data and assistance in acquiring the basic analysis of a bugcheck, if it exists. It also provides pointers to articles with more complex analysis techniques targeted to the more experienced.

This article includes the following sections:

- Overview
- Frequently Reported Problems
- Frequently Used Article List
- Tools Relating To Component
  CCAT
  CCAT WEB GUI
  AUTOCLUE
  CADC
- Performing Additional Research
- References

FREQUENTLY REPORTED PROBLEMS:

The majority of fatal bugchecks are noticed in one of the following ways:

- The system crashed with one of various FATAL BUGCHECK messages
- The system REBOOTED for no apparent reason
- The system received a MACHINE CHECK
- The system HALTED and the console prompt appears
- The system crashed and no dumpfile was produced

FREQUENTLY USED ARTICLE LIST:

Following is a list of articles believed to be the most beneficial in addressing bugcheck issues:

> Note:
> The database also contains many articles describing "How to force crash" a particular hardware platform or system which can be found by using FORCE_CRASH as a query.

[Canasta] Crashdump ANAlySis Troubleshooting Assistant - overview

[OpenVMS] Managing Dumpfiles on VAX & Alpha Systems or Clusters

[OpenVMS] CLUE for OpenVMS VAX - Logicals, Symbols and Syntax

[OpenVMS] CLUE for OpenVMS Alpha - Logicals and Syntax

[OpenVMS] How To Run CLUE After A MINIMUM Or UPGRADE System Startup

[AutoCLUE] AutoCLUE for OpenVMS User's Guide

[AutoCLUE] Paving the Way for Automatic Crash Analysis

[OpenVMS] SDA-E-DUMPEMPTY And NOTSAVED Errors From ANALYZE/CRASH

[OpenVMS] Sending Crash Dump Information to the CSC Using DSNlink or DIA

TOOLS RELATING TO COMPONENT:

CCAT:

The Compaq Crash Analysis Tool (CCAT) will take crash dump information, known as a CASE, and attempt to match the case data to a database of known crashes. CCAT can be accessed via the Internet using the address:

    http://CXOHOME.CXO.DEC.COM/ccat/

> Note:
> Compaq Crash Analysis Tool (CCAT) was formerly known as CANASTA.

CCAT stores important footprint information about a system crash in the CCAT CASE database. A CCAT CASE contains the unique crash dump parameters such as Bugcheck name, Process name, Module name, Module offset....

Case data is extracted from a crash dumpfile and sent to CCAT for processing and to determine if the crash matches an existing CCAT RULE.

> Note:
> See the detailed description on collecting case data in the next section of this article.

RULES contain specific information for identifying a unique crash pattern, including Problem description, Confirmation techniques, and Solution information.

```
  Example:
    RULE ID: 009A8D99-BE2EECA0-1C03D6 or for newer rules CCAT-V-A-1898
    Rule Parameters:
        Bugcheck = INVEXCEPTN
        Version = 5.5-2
        Module = SHDRIVER
        Offset = 3927
        Instruction = BBC

        Problem Description:
           OpenVMS VAX System crashes with INVEXCEPTN in SHDRIVER+3927

        Technique Text:
           This would be example data from CLUE file or SDA commands
           to be used to verify whether this rule really matches the
           problem.

        Solution Text:
           Please obtain and install patch VAXSHAD09_U2055.
```

CCAT WEB GUI:

The CCAT Web GUI (formerly called "CANASTA Mail Server") is used to send a CASE to CCAT. The Web GUI saves the CASE information in the CCAT database and attempts to match it to a rule. The CCAT Web GUI can be accessed via the Internet using the address:

```
     http://caddoweb.cxo.dec.com/CGI-BIN/CCATWEB/CCATWEB.EXE
```

the Following steps describe how to use the CCAT Web GUI:

Step 1:

Obtaining a CLUE or crash-data file:

If you have access to the crashed system, you will find the CLUE or crash-data files in the following locations:

```
     CLUE$OUTPUT:CLUE$LAST_node.LIS          ! For OpenVMS VAX
     CLUE$COLLECT:CLUE$node_ddmmyy_hhmm.LIS  ! For OpenVMS Alpha
```

The preferred data to send to the CCAT Web GUI for analysis is the CLUE data file.

If you have access to the system dump file, you can extract CLUE information with the following commands:

```
     OpenVMS VAX V6.0 or higher:
     ---------------------------
       $ CLUE :== $CLUE
       $ CLUE/OUT=<disk>:[<directory>]CRASH.DAT <dumpfile>

     OpenVMS Alpha V6.1 or higher:
     -----------------------------
       $ ANALYZE/CRASH <dumpfile>
       SDA> READ/EXEC
       SDA> SET OUT CRASH.DAT
       SDA> CLUE CRASH
       SDA> CLUE STACK
       SDA> CLUE CONFIG
       SDA> EXIT
```

The commands produce a CRASH.DAT file which contains sufficient information for the CCAT Web GUI to process.

Step 2:

Copy the CRASH.DAT file from the customer's system to your EWPC.

  Note:
    See the section "Moving CLUE files" for some suggestions
    on copying the data to your EWPC.

Step 3:

Connect to the CCAT Web GUI web page at:

```
     http://caddoweb.cxo.dec.com/CGI-BIN/CCATWEB/CCATWEB.EXE
```

Step 4:

Log in to the tool using your badge and a log number prefaced by a site code.

```
  Example:
    Log number prefaced by a site code.

      CXO-C991217-15
       ^  \        /
       |   ---------- Log number
       +------------- Site code
```

```
If there are several crashes for the same log number add a unique number
to the end. ----------+
                      |
  Example:            |
    CXO-C991217-15-1  |
                   ^  |
                   +--+
```

Step 5:

Select the correct operating system type tab.

Step 6:

Use the BROWSE option at the bottom of the screen to select the CRASH.DAT file, then select LOAD.

Moving CLUE Files From The Customers System:

This section offers possible options for getting the CLUE data file (CRASH.DAT) from the customers system to your EWPC.

- Internet Mail:

  Send the file over the Internet with the following command:

```
$ MAIL <filename>  <yourname>@compaq.com
```

- Capture Incoming Information:

  Use the following steps when a terminal emulator to dial into the customer system from your EWPC, e.g.; KEA session:

  1. Select "Capture Incoming Information" from the TOOLS menu.
  2. Define the characteristics of the file you want to create, i.e.; name, type, location.
  3. Type the CRASH.DAT file you created
  4. Select "End Capture" from the TOOLS menu.
  5. Open the text file of captured data and remove any captured prompt lines or unwanted text.

- Cut and Paste:

  Cut and Paste the CLUE information from the customers system to the CRASH.DAT file. This will provide the quickest answers but does not save enough information for later analysis in CCAT. Unless time is an issue, the other methods are recommended:

  1. On an OpenVMS Alpha system, Cut and Paste the crash parameters from a CLUE CRASH command in SDA.
  2. On an OpenVMS VAX system, Cut and Paste the crash parameters from the first 2 screens of the CRASH.DAT file on the remote system into a local CRASH.DAT file.

- FTP

  If customer has FTP access, use this method to move the CLUE file to a Compaq system for elevation, documentation, etc.

  Be aware that the following FTP commands are "CASE SENSITIVE":

  From The Remote System:

```
     FTP> open xfer-cxo.service.digital.com
                       - OR -
     FTP> open xfer-alf.service.digital.com

         Note:
           The IP address may also be used.

               xfer-cxo.service.digital.com = 192.208.35.20
               xfer-alf.service.digital.com = 192.208.34.20

     FTP> Username: anonymous
     FTP> Password: <email address>
     FTP> cd to_digital
     FTP> put <unique filename>
```

From The Compaq System:

```
     FTP> open xfer-cxo.service.digital.com
                     - OR -
     FTP> open xfer-alf.service.digital.com
```

Note:
The IP address may also be used.

```
          xfer-cxo.service.digital.com = 192.208.35.20
          xfer-alf.service.digital.com = 192.208.34.20
```

```
     FTP> Username: digital
     FTP> Password: DIGXFER
     FTP> cd to_digital
     FTP> get <same unique filename>
```

- DSNlink
  If the customer has DSNlink access, you can copy the file using:

```
     $ DSN COPY <filename> <sequence number>
```

- RFTS
  If the customer has a current version of DECevent installed on their Alpha system, i.e.; version 2.8 or higher, RFTS (Remote File Transfer Server) will be installed in the following directory by default:

```
     SYS$SYSDEVICE:[SYS0.DIA$TOOLS]
```

If you're dialed in via the HANC systems:

```
     $ SET DEFAULT SYS$SYSDEVICE:[SYS0.DIA$TOOLS]
     CTRL-A
     VTERMINAL> RFT
```

AUTOCLUE:

AutoCLUE is an Electronic Service Tool and can be installed on customer systems which have a DSNlink connection to Compaq. AutoCLUE has been replaced by CADC, which is described in the next section. If installed, AutoCLUE automatically send crash dump information via DSNlink to the Compaq CSC for CCAT crash dump analysis. AutoCLUE can be accessed via the Internet using the address:

http://www.service.digital.com/dsnlink/autoclue.htm

For in-house systems, AutoCLUE uses DECnet or SMTP mail and does not require DSNlink.

Customers using DSNlink for OpenVMS can use DSNlink EKD (Electronic Kit Delivery) to request the AutoCLUE kit. The kit can be requested by sending mail to DSN%AUTOCLUE-OVMS-KIT. The AutoCLUE kit will be automatically copied to the DSN$COPY_DIRECTORY: of the customer's DCN (DSNlink Connector Node). Once the kit has been successfully copied. the customer will receive mail containing installation instructions

Note:
Use the CSC32::AUTOCLUE notes file for answers to questions, and to report problems relating to AutoCLUE

When AutoCLUE sends crash data to the CSC via DSNlink, the crash dump information is added to the first work unit of an SRQ and the call is logged to the appropriate CHAMPS team queue.

The information is also automatically saved in the CCAT database as a CASE, and an attempt is made to match the CASE to known crash dump patterns (RULES). If a match is made, the results of the CCAT analysis are added as work unit two of the original AutoCLUE SRQ.

If an exact match is made which contains a solution that can be sent to the customer, then the results of the CCAT analysis will be added as work unit two, the solution automatically delivered, and the call will be placed in the appropriate CHAMPS queue for review.

> Note:
> The delivery of a solution may be prevented if the AutoCLUE call is pulled from the queue before DSNlink has sent the solution to the customer. When reviewing these calls ensure that the solution was delivered.

Anything other then exact match with a deliverable solution causes the call to be placed in the appropriate CHAMPS queue for additional analysis.

Installing AutoCLUE:

If the customer would like to install AutoCLUE, but doesn't have DSNlink, they can go to the DSNlink homepage for information on downloading and installing DSNlink.

```
     http://www.service.digital.com/dsnlink/
```

Once DSNlink is installed, you can obtain the installation kit for AutoCLUE on OpenVMS by sending mail to:

```
     DSN%AUTOCLUE-OVMS-KIT
```

See the following article for more information on installing AutoCLUE:

    [AutoCLUE] Paving the Way for Automatic Crash Analysis

CADC:

The Crash Analysis Data Collector (CADC) is part of CCAT and was developed as the replacement tool for AutoCLUE. CADC collects rule data from a system crash and attempts to match the data against a rule database on the customer system. The tool then sends the information to the CSC in a similar manner to AutoCLUE. Use the following Internet address for more information on CADC:

```
     http://pinkft.cxo.dec.com/svctools/webes/cadc.html
```

PERFORMING ADDITIONAL RESEARCH:

Using CCAT

Use CCAT to look for the same or similar crash pattern from other customers (CASE search), or to look for similar rules (RULE search):

Note:

A hint for both CASE and RULE search (see Step 4), it's usually good to enter the module but leave off the offset.

Step 1:

Note the crash parameters and connect to CCAT Rule Writer

http://cxohome.cxo.dec.com/cgi-shl/ccatdbi/CCATDBI.EXE

Step 2:

Log in. If you do not have a valid password enter anything and log in as guest.

Step 3:

Select the correct operating system.

Step 4:

Select CASE SEARCH or RULE SEARCH and enter only enough parameters to match your crash.

Using CLUE History

The crash history information on the customers system can be reviewed to determine:

1. If the current crash pattern matches any previous system crash
2. If the system is crashing with different bugchecks
3 The systems crash frequency

- CLUE History for OpenVMS VAX Systems:

CLUE history data is maintained in CLUE$OUTPUT:CLUE$HISTORY.DATA whose contents can be viewed by issuing the following commands:

Note:

Some fields in the following displays have been compressed for display purposes.

```
$ CLUE :== $SYS$SYSTEM:CLUE
$ CLUE/DISPLAY
   # Node   Time         Type        Proc    Module  Offset
   1 TSTSR  17-JUL 18:19 OPERATOR    _OPA0:  UNKNOWN      0
   2 TSTSR   2-JUN 17:36 OPERATOR    BILES   UNKNOWN      0
           .                  .             .
           .                  .             .
  20 TSTSR   7-JUN 13:33 ASYNCWRTER  USR_1   SYSLOA    79FA
CLUE_DISPLAY>
```

Canasta parameter data can be viewed using following command:

```
CLUE_DISPLAY> SHOW CANASTA 20
 #20  ASYNCWRTER crash on TSTSR at  7-JUN 13:33:32.40
 CANASTA Parameter               Value
 VMS VERSION                     6.1
 BUGCHECK TYPE                   ASYNCWRTER
 PROCESS NAME                    USR_1
 IMAGE NAME                      PSDC$DC_V5
 CPU TYPE                        7000-620
 SID                             17000202
 SIGNAL ARRAY COUNT              00000000
 EXCEPTION REG 1                 FFFFFFFF
 EXCEPTION REG 2                 FFFFFFFF
 EXCEPTION REG 3                 FFFFFFFF
 EXCEPTION PC                    85F0A33A
 EXCEPTION PSL                   041F0000
 FAILING INSTRUCTION             BUGW
 FAILING MODULE                  SYSLOA
 OFFSET                          79FA
```

The SHOW ALL command can also be used.

Data can also be extracted into a CRASH.DAT file:

```
CLUE_DISPLAY > EXTRACT/OUT=CRASH.DAT 20
```

Note:
Normal system shutdowns are also logged for VAX systems so avoid extracting them (Type = OPERATOR).

- CLUE History for OpenVMS Alpha Systems:
  CLUE$SDA will run automatically during system startup and create a crash history entry and a full CLUE listing file, when the system automatically reboots after a system crash. Unlike VAX systems, OPERATOR shutdowns are not listed. The crash history file can be viewed with the following commands:

```
$ TYPE CLUE$HISTORY
Date          Version   System/CPU           Node    Bugcheck
Process     PC        Module                   Offset
------------  --------  -------------------  ------  ----------
----------  --------  -----------------------  --------
20-OCT 16:47  V6.1      DEC 2000 Model 300S  HAOAX1  INVEXCEPTN
NULL        800386F4  SYS$CPU_ROUTINES_0602    000006F4
25-SEP 15:03  V6.2      DEC 2000 Model 300   HAOAX1  OPERATOR
SYSTEM      000305B0  OPCCRASH                 000305B0
```

Note:
The output from the above command is 132 columns. Set the terminal width to 132 columns to justify the display.

```
$ SET TERM/WIDTH=132
```