

# CA Gen

## Transaction Enabler User Guide

Release 8.5



This Documentation, which includes embedded help systems and electronically distributed materials (hereinafter referred to as the "Documentation"), is for your informational purposes only and is subject to change or withdrawal by CA at any time.

This Documentation may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA. This Documentation is confidential and proprietary information of CA and may not be disclosed by you or used for any purpose other than as may be permitted in (i) a separate agreement between you and CA governing your use of the CA software to which the Documentation relates; or (ii) a separate confidentiality agreement between you and CA.

Notwithstanding the foregoing, if you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with "Restricted Rights." Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © 2013 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

## CA Technologies Product References

This document references the following CA Technologies products:

- CA Gen

## Contact CA Technologies

### Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At <http://ca.com/support>, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

### Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to [techpubs@ca.com](mailto:techpubs@ca.com).

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at <http://ca.com/docs>.



# Contents

---

Chapter 1: Transaction Enabler	7
Prerequisite Tasks .....	8
Required Tasks .....	9
Chapter 2: AEF Asynchronous Daemon	11
Description .....	11
Invoking the AEFAD .....	12
Setting Environment Variables.....	12
Starting the AEFAD .....	12
AEFAD Options .....	14
Features of the AEFAD .....	21
Trace Logging .....	21
Statistics Logging.....	22
Chapter 3: AEF User Funnel	23
Description .....	23
Invoking the AEFUF .....	23
Setting Environment Variables.....	24
Starting the AEFUF .....	24
AEFUF Options .....	25
AEFUF Command Example .....	26
Features of the AEFUF .....	27
Trace Logging .....	28
Chapter 4: AEF Client	29
Description .....	29
Invoking the AEFC.....	30
Setting Environment Variables.....	30
Starting the AEFC .....	31
AEFC Options.....	31
AEFC Command Example .....	34
Features of the AEFC .....	34
Record/Playback .....	34
Environment Variable Copy .....	35
Controlling the AEFAD Server .....	35

---

Realtime Monitoring .....	40
 Chapter 5: Using the AEFAD and AEFUF as Windows Network Services	 45
Installation.....	45
Operation .....	46
regTE.cmd Parameters .....	46
regTE.cmd Example .....	47
unregTE.cmd Parameters .....	48
unregTE.cmd Example.....	48
Recommended Steps .....	48
 Chapter 6: Transaction Enabler Environment Variable Summary	 49
Transaction Enabler Environment Variables .....	49
 Index	 53

# Chapter 1: Transaction Enabler

---

The Transaction Enabler (TE) is a CA Gen version of a transaction processing monitoring system. The TE is used in conjunction with client/server applications to manage the server application. The TE is offered as a companion to the Implementation Toolset (IT) product suite.

The TE comprises of the following components:

- AEF Asynchronous Daemon (AEFAD)

Provides a production environment for transaction processing. It loads application load modules and keeps them resident in memory, connected to the database, and shareable (one user transaction at a time). The AEFAD also controls the unloading of applications as needed.

- AEF User Funnel (AEFUF)

Lets more than one user share an AEFAD TCP/IP session. It also lets multiple GUI clients connect to a single AEFAD environment.

**Note:** In Windows, the AEFAD and AEFUF can be started and stopped as services. For more information on AEFAD and AEFUF, see the appendix “[Using the AEFAD and AEFUF as Windows Network Services](#) (see page 45)”.

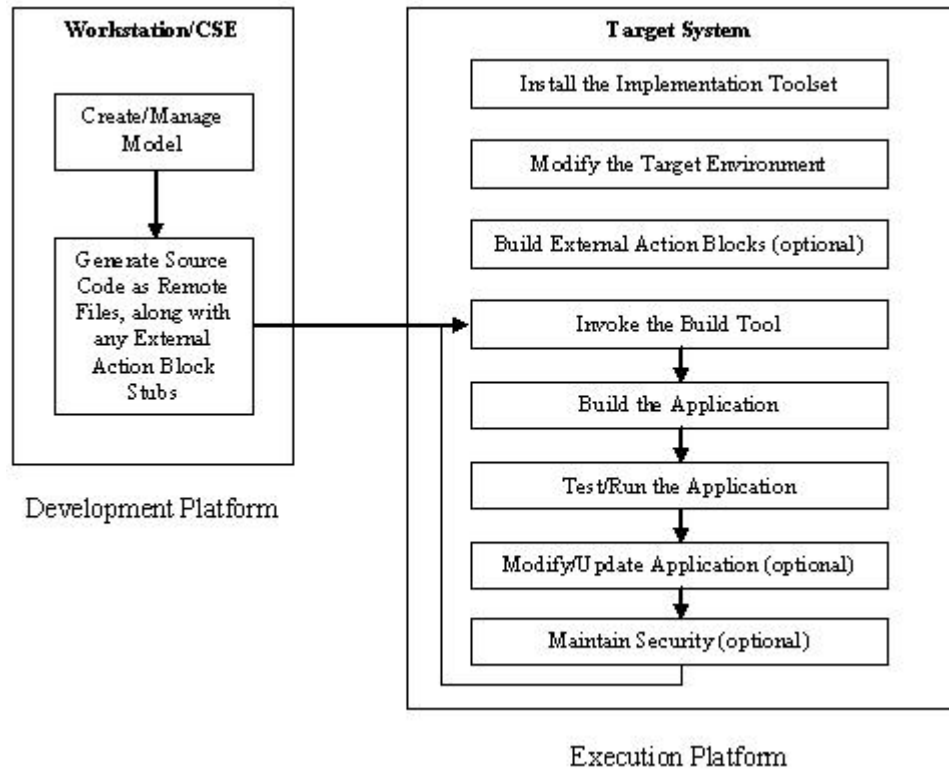
- AEF Client (AEFC)

Interface through which the user interacts with the AEFAD and AEFUF. The AEFC communicates directly with the AEFUF and establishes a connection to the AEFAD for the user. Each AEFC controls one user terminal.

**Note:** The TE was developed to provide a base set of capabilities for deploying generated server manager and block mode applications in an actively managed environment. The TE was not designed with the sort of redundancy and failover protection of TP monitors like Tuxedo and CICS. While we did not make any design choices that would not make TE able to operate 24x7, it would be unreasonable to assume that the TE was capable of providing a hands off service level similar to that demonstrated by the aforementioned TP monitors. No performance or throughput specifications are claimed for the TE and none must be assumed.

## Prerequisite Tasks

Prior to placing an application into production, you must complete the certain tasks. The following illustration explains these tasks:

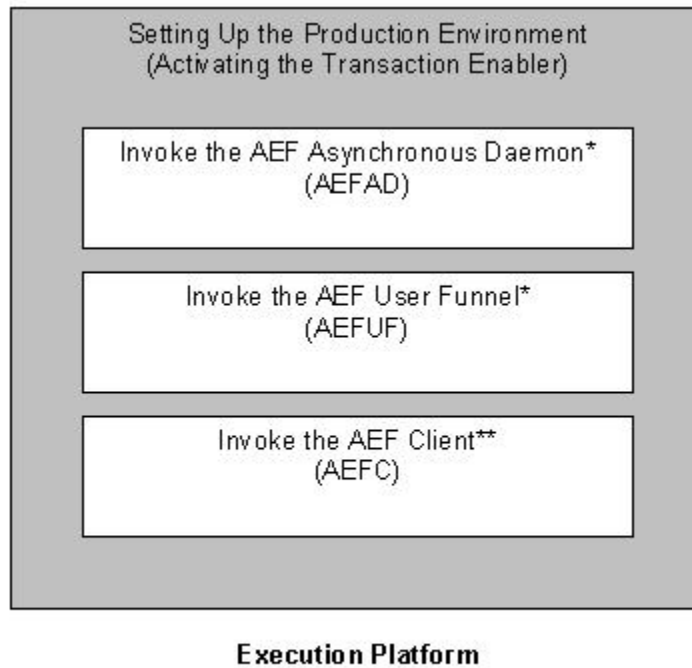


Development platform tasks are documented in the various workstation and CSE design guides. For more information about the Execution platform tasks, see the UNIX and Linux Implementation User Guide and Windows Implementation User Guide.



## Required Tasks

A generated application is ready for production when each of the tasks in the previous section has been completed. The following illustration shows the tasks that are then required to set up the production environment. In addition to having at least one generated application tested and ready for production, be sure that TCP/IP is available for communications.



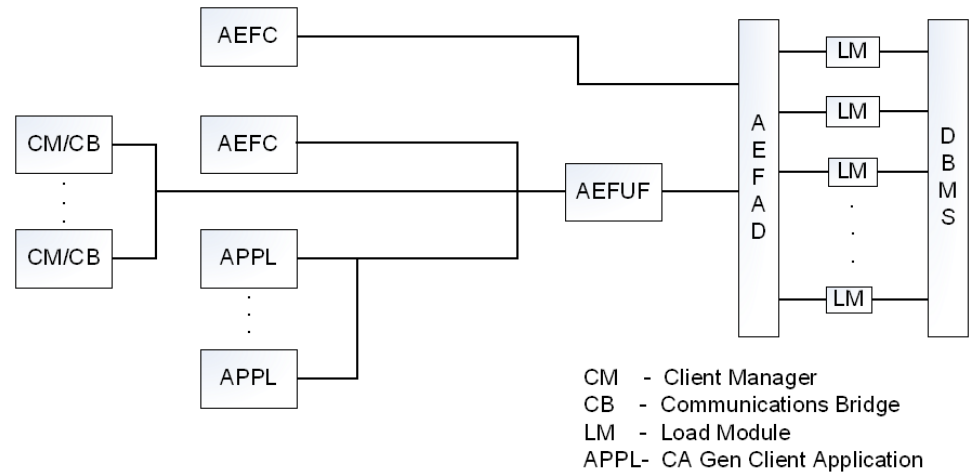
\* May be started as Network Services under Windows

\*\* Optional – AEFC is not required for Distribution Process Applications

## Structure of the Production Environment

The structure of the production environment depends on the requirements of the application used in the environment. A production environment can contain various numbers of Client Managers (CM) and/or Communication Bridges (CB), AEFCs, AEFUFs, and AEFADs. The balance of these components is configured to maximize performance and throughput.

The following illustration describes the inter-relationship of the various components of the transaction enabler, each of which is explained in detail in subsequent chapters of this guide:



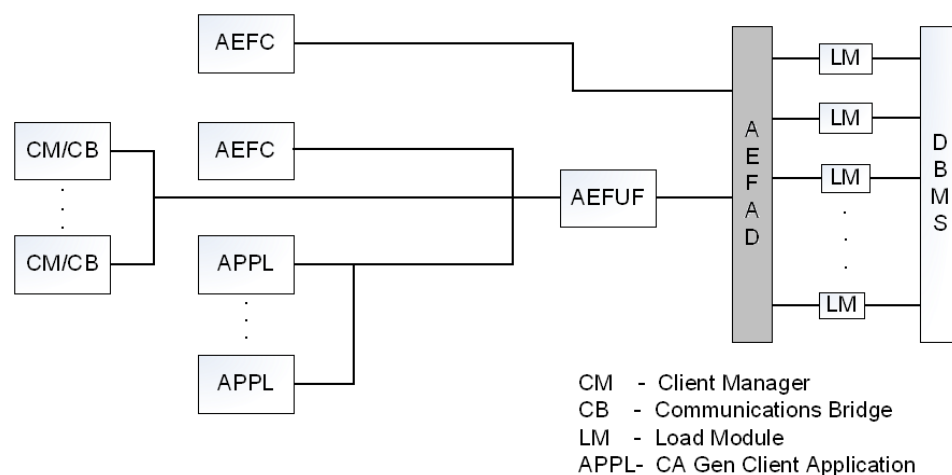
## Installing the Transaction Enabler

Each of the components of the TE is automatically loaded onto your target system when the IT is installed.

**Note:** For more information about choosing a particular platform for the implementation toolset, see the *Distributed Systems Installation Guide*.

## Chapter 2: AEF Asynchronous Daemon

To establish the production environment, you must first invoke the AEF Asynchronous Daemon (AEFAD) component of the Transaction Enabler, as explained in the following illustration.



This chapter describes the purpose and function of the AEFAD and the procedure required to invoke it.

**Note:** In Windows, the AEFAD can be started and stopped as a Network Service. For more information about AEFAD, see the appendix [“Using the AEFAD and AEFUF as Windows Network Services”](#) (see page 45)."

### Description

The AEFAD is a server program that contains message routing and CA Gen application program loading logic. It provides a production environment for transaction processing by loading application load modules and keeping them resident in memory. While in memory, the load modules remain connected to the database and shareable (one user transaction at a time). The load modules and their associated transaction codes must be known to the AEFAD. The application load modules must be located in an inqload subdirectory, and the trancode/load module mapping information must be defined in the aeenv file.

The AEFAD is started on the server (host) system where the generated application(s) are located. If required, the AEFAD can be located on a system other than the one on which the AEFUF and AEFC clients are located. Multiple AEFAD environments may exist on a single platform. The AEFAD communicates with the AEFUF and AEFC via TCP/IP sockets.

After the AEFAD is started, some of its operational parameters can be controlled via commands entered from the AEFC. For more information about AEF Client, see the chapter “[AEF Client](#) (see page 29).”

## Invoking the AEFAD

The following sections detail the procedures required to set up and invoke the AEFAD.

### Setting Environment Variables

**Note:** See your system documentation for the procedure required to set or change environment variables for your particular platform.

Before the AEFAD can be run, certain environment variables must be set. The list of these environment variables is as follows:

- IEFH
- AEFAD\_STATS
- AEPATH
- PATH (Windows only)
- SHLIB\_PATH (HP Itanium)
- LD\_LIBRARY\_PATH (Linux, Solaris)
- LIBPATH (AIX)

For more information about these environment variables for Windows, UNIX and Linux systems, see the appendix “[Transaction Enabler Environment Variable Summary](#) (see page 49).”

When the AEFAD is started, it uses IEFH to locate required software components. It then finds the transaction routing tables by searching the paths defined in AEPATH. Finally, the AEFAD searches the PATH specification on Windows to locate DBMS information and other executables that are used by the load modules that will be loaded.

### Starting the AEFAD

The method for starting the AEFAD depends on the type of platform you use. The following sections explain the procedure for starting AEFAD on UNIX and Linux and Windows.

## UNIX and Linux

From a system prompt, enter the following:

```
aefad -option 1 -option 2 ... -option n <Enter>
```

## Windows

To run the AEFAD in its default mode, you can simply double-click the program name in the File Manager listing provided. Otherwise, from a command prompt, enter the following:

```
%GENxx%\Gen\VSabc\AEFAD.EXE -option 1 -option 2... -option n
```

or

```
%GENxx%\Gen\VSabc\amd64\AEFAD.EXE -option 1 -option 2... -option n
```

The AEFAD will start and no further operator action is required.

**Note:** VSabc refers to the supported version of Visual Studio. Replace VSabc with VS100 for Visual Studio 2010 and VS110 for Visual Studio 2012. xx refers to the current release of CA Gen. For the current release number, see the *Release Notes*.

An icon can also be created with these options already defined and used to start the AEFAD.

## AEFAD Options

The following table define the options you can modify at startup. You can specify any number of command options in the command line, in any order.

### **-a [number]**

Specifies a value for the maximum number of CA Gen application load modules that this AEFAD server will simultaneously load into memory. When this value is reached, an additional load will delete the least recently used inactive load module. This option's value is limited by the available number of processes on the system. Raising this value above the default will lower the number of user connections possible. For more information, see [-u Option Details](#) (see page 19).

Default value

64

#### **Example:**

```
aefad -a 40
```

**Note:** For Windows, the maximum value for this parameter is 127. If it is set to a higher value, the aefad will start, but will lower the value to 127 and create a log file with a message that it has detected a higher value and has lowered it.

### **-d [number]**

Specifies a value for the time (in seconds), which will be used to determine which active transactions, the DISPLAY program will display. Transactions active longer than this value will be displayed. Active transactions are marked as type DACT.

**Default value**

5

#### **Example**

```
aefad -d 1
```

### **-e [number]**

Specifies a value for the number of seconds to wait for a TCP/IP send or receive request to complete before the assumption is made that the CA Gen application load module or network node is down. After this timeout is reached, the application or access to the network connection is automatically canceled and restarted.

**Default value**

5

#### **Example**

```
aefad -e 3
```

**-f [number]**

Defines the length of time (in milliseconds) to wait for a TCP/IP send or receive request to be logged in the AEFAD trace file (See option -t). The entries in this log file are used by CA Gen support personnel for troubleshooting.

**Default value**

1000

**Example**

```
aefad -f 3000
```

**-h [host\_name]**

(UNIX, Linux) Specifies a value that identifies the TCP/IP network interface hostname to which this AEFAD will connect with the application load modules. This is only needed in the rare cases where multiple Network Interface Cards are present and the default value *localhost* is assigned to a different NIC than needed. Using the default *localhost* has an advantage in performance.

(Windows) Is not used. A warning message is output and the AEFAD continues normally.

**Default value**

localhost

**Example**

```
aefad -h devbox
```

Where [host\_name] can be in IPv4 or IPv6 format.

**-i [port]**

Specifies the port to which the AEFAD server being started will listen. The CA Gen user activates the AEFAD or AEFUF program with a corresponding port number to connect to this AEFAD server.

**Default value**

2009

**Example**

```
aefad -i 2005
```

**-l**

Specifies that each client/server request should be verified by the security exit and that AEFC users will be required to enter a valid user ID to gain access to this AEFAD. If this switch is specified, the USERID, PASSWORD, and TRANSACTION NAME from each client/server request will be passed to a security process for validation.

**Default value**

No security checking

**Example**

aefad -l (to activate security checking)

**Note:** This processing applies only to buffers not containing enhanced security data. For buffers that contain enhanced security data, it is assumed that the DPSs provide their own security using the security validation exit (tirsecv) invoked by the server runtime.

**-m [number]**

Specifies the maximum number of copies of a particular application load module that are allowed to be loaded into memory. When this number of copies of an application has been loaded, additional users will queue to wait for an available load module.

**Default value**

64

**Example**

aefad -m 4

**-p [number]**

Specifies the length of time (in minutes) that the AEFAD server waits before it writes the transaction summary and load module statistics into the aestats-<procid> file (where <procid> is the process number of the AEFAD). For Windows, this file is created in the %USERPROFILE%\AppData\Local\CA\Genxx\logs\server directory. For UNIX and Linux, this file is created in the directory from which AEFAD was invoked.

**Default value**

15

**Example**

aefad -p 10



**-q [number]**

Specifies the number of user requests that must queue to execute the same application load module before an additional load module is loaded by the AEFAD server into memory. An additional load will occur up to the limit of application load modules (specified by the -m option). When the limit of applications is reached, an additional load will cause the least recently used inactive load module to be deleted. The depth of the user request queue is affected by the number of active copies of the load module. So the greater the number of active load modules, the actual depth of the queue is greater than the value of -q.

**Default value**

2

**Example**

```
aefad -q 4
```

**-s [number]**

Specifies a value for options that need to be logged to the aestats-<procid> file (where <procid> is the process number of the AEFAD). For Windows, this file is created in the %USERPROFILE%\AppData\Local\CA\Gen xx\logs\server directory. For UNIX and Linux, this file is created in the directory from which the AEFAD was invoked.

This number is the sum of the following values which activate particular loggings:

8

AEFAD CPU utilization statistics

4

AEFAD send/receive statistics

2

APPL statistics

1

USER statistics

If the -s flag is defaulted to 0, then no statistics are written periodically to the aestats-<pid> file.

**Default value**

0 (no statistics logging)

**Examples**

```
aefad -s 5 (USER and AEFAD send/receive statistics will be logged)
```

```
aefad -s 11 (USER, APPL, and AEFAD CPU utilization statistics will be logged)
```

**-t [number]**

Specifies a value between 0 and 31 that determines the level of information logged to an internal AEFAD program trace file. The log file is named lg-aefad-<procid>.log (where <procid> is the process number of the AEFAD) containing the trace data. For Windows, this file is created in the %USERPROFILE%\AppData\Local\CA\Genxx\logs\server directory. For UNIX and Linux, this file is created in the directory from which the AEFAD was invoked. The least amount of information will be provided with a value of 0, and the most information will be provided with a value of 31. However, as this value is increased, so too will the size of the file created, to the extent that at -t 31, an enormous amount of disk space will be consumed.

**Default value**

0 (no logging)

**Example**

```
aefad -t 15
```

**-u [number]**

Defines the maximum number of users (AEFC clients) that can be connected to this AEFAD server. For more information about the -u option, see the -u Option. Raising this value above the default will lower the number of application connections possible. See the -a option for more information.

**Default value**

512

**Example**

```
aefad -u 256
```

**-w [number]**

**UNIX and Linux**

Specifies the time (in seconds) of how often active transactions (DACT) checking will occur (looking for transactions that exceed the value of the -d parameter).

**Windows**

Specifies the length of time (in milliseconds) to wait on TCP/IP I/O (select timeout).

**Default value**

5 (UNIX and Linux), and 100 (Windows)

**Example**

```
aefad -w 2
```

**Note:** When the display program is used to monitor AEFAD activities, currently active transactions are marked as type DACT.

**-b[number]**

Specifies the TCPIP Send/Receive buffer size in 65,536 byte blocks.

**Default value**

1

**Maximum value**

100

**Example**

aefad -b5

**-n[number]**

Turns the webservice feature of the Asynchronous Daemon on or off.

**Values**

1=on (Default)

0=off

**Example**

aefad -n0

**-help**

Displays the usage information.

## -u Option Details

The maximum value of the -u option has been removed. You can set any value that the platform supports. The default value is 512. When setting this value, consider the impact on the memory required to support the value. The memory needed by the aefad is as follows:

- Each user actively connected requires 32 KB.
- The user in this case is a user funnel, a client manager connected to a user funnel, or a block mode display client-aefc.
- Each loaded application module requires 64 KB.
- Each user transaction that is active requires 32 KB.
- Each potential user requires 400 bytes.

### Example

Assume that the -u option has the -u 1000 setting.

The immediate memory requirement would be 400,000 bytes for aefad to initialize. As users connect, an additional 32 KB for each user would be required.

If there were:

- 800 connected users
- 100 application modules loaded
- 50 transactions active at a point in time

The memory required at the point in time would be the following:

$$(800 \times 32 \text{ KB}) + (100 \times 64 \text{ KB}) + (400 \times 1000) + (50 \times 32 \text{ KB}) =$$

$$26,214,400 + 3,276,800 + 400,000 + 1,638,400 =$$

$$31,529,600 \text{ bytes for the aefad process}$$

The server load module memory requirements are not included in this total.

**Note:** This parameter setting should include user connection and load module connections.

### AEFAD Command Example

The following example describes a typical UNIX or Linux command line sequence invoking the AEFAD, assuming the environment variables have not been previously set:

```
setenv IEFH /usr/ief/runtime
setenv AEPATH '$IEFH;/usr/appl/prod'
setenv PATH $IEFH/bin: $PATH
aefad -t 15 -i 4001 -a 8
```

First, the necessary environment variables have been set to allow the AEFAD to find the information it will need when invoked. Then, the command line has been entered to start the AEFAD, with certain options changed from their default values:

- Level of information logged by the program trace facility has been increased (-t 15).
- Application port number has been changed from 2009 to 4001 (-i 4001).
- Maximum number of applications this AEFAD server instance will load and share with connected users has been decreased to 8 from 64 (-a 8).

## Features of the AEFAD

This section provides information about the features of the AEF asynchronous daemon.

### Trace Logging

The log file lg-aefad-<procid>.log (where <procid> is the process number of the AEFAD) provides USER and APPL session activity entries and related error messages and is activated using the AEFAD command option -t number. For Windows, the lg-aefad-<procid>.log file is created in the %USERPROFILE%\AppData\Local\CA\Genxx\logs\server directory. For UNIX and Linux, this file is created in the directory from which the AEFAD was invoked. This file can be used for problem analysis but only by Technical Support. The following is a sample lg-aefad-<procid>.log file:

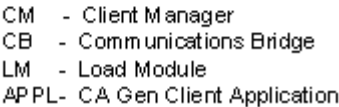
```
13:00:19.656====>aefad: allocating 226184 bytes memory for 577 pgm structures
13:00:19.666====>aefad: Server (ver 07000) started
13:00:19.666====>aefad: MAX funnels/direct users=123
13:00:19.666====>aefad: Number of pgm structures=577
13:00:19.666====>aefad: PARMS: -i2009 -u512 -a64 -q2 -m64 -d5 -e5 -f1000
13:00:19.666====>aefad: PARMS: -p15 -s3 -w100 -t1
13:00:19.666====>aefad: fd=28 socket = 28, listen on port 2009
13:00:19.666====>aefad: Bind code=0
13:00:19.666====>aefad: Listen code=0
13:00:31.843====>aefad: USER: user48, Sock = 48, sock48, connected
13:00:31.843====>rcvfhdr: FUNN: Sock = 48, user1, started
13:00:40.966====>aefad: USER: user32, Sock = 32, sock32, connected
13:00:40.966====>aefad: USER: Sock = 32, localhost, user1, USER START
13:00:53.464====>stopaef: /stopaef entered
13:00:53.464====>myclose: USER: Sock = 28, , , USER END
```

## Statistics Logging

The aestats-<procid> file (where <procid> is the process number of the AEFAD), provides a breakdown of the response times for the user's session. This file is written to by the AEFAD as dictated by the options provided when started. For Windows, this file is created in the %USERPROFILE%\AppData\Local\CA\Gen xx\logs\server directory. For UNIX and Linux, this file is created in the directory from which the AEFAD was invoked. A sample of the file follows:

```
PID=029114: 01/17/02 14:13:52 AEFAD PGM STARTED
PID=029114: 01/17/02 14:15:00 AEFAD PERIOD ENDED
PID=029114: period= 1 trans= 33, max= 0.850, aver= 0.443, 78%
PID=029114: period= 2 trans= 6, max= 1.350, aver= 1.241, 14%
PID=029114: period= 4 trans= 1, max= 2.300, aver= 2.300, 2%
PID=029114: period= 6 trans= 0, max= 0.000, aver= 0.000, 0%
PID=029114: period= 10 trans= 0, max= 0.000, aver= 0.000, 0%
PID=029114: period=200 trans= 2, max=11.450, aver=11.000, 4%
PID=029114: total transactions=42, average response= 1.104
PID=029114: 01/17/02 14:23:01 AEFAD PGM ENDED
PID=029114: period= 1 trans=331, max= 0.950, aver= 0.430, 81%
PID=029114: period= 2 trans= 73, max= 1.950, aver= 1.270, 17%
PID=029114: period= 4 trans= 2, max= 2.300, aver= 2.175, 0%
PID=029114: period= 6 trans= 0, max= 0.000, aver= 0.000, 0%
PID=029114: period= 10 trans= 0, max= 0.000, aver= 0.000, 0%
PID=029114: period=200 trans= 0, max=11.450, aver= 0.000, 0%
PID=029114: total transactions=406, average response= 0.590
```

---



## Setting Environment Variables

The AEFUF does not require that any environment variables be set prior to being invoked.

## Starting the AEFUF

The method for starting the AEFUF depends on the type of platform you use. The following sections explain the procedure for starting AEFAD on UNIX and Linux and Windows.

### UNIX and Linux

From the command prompt, enter the following:

```
aeuf -option 1 -option 2 ... -option n <Enter>
```

### Windows

To run the AEFUF in its default mode, you can simply double-click the program name in the File Manager listing provided. Otherwise, from a command prompt, enter the following:

```
%GENxx%\Gen\VSabc\AEFUF.EXE -option 1 -option 2... -option n
```

or

```
%GENxx%\Gen\VSabc\amd64\AEFUF.EXE -option 1 -option 2... -option n
```

The AEFUF will start and no further operator action is required.

**Note:** VSabc refers to the supported version of Visual Studio. Replace VSabc with VS100 for Visual Studio 2010 and VS110 for Visual Studio 2012. xx refers to the current release of CA Gen. For the current release number, see the *Release Notes*.

You can also create an icon with these already defined options and used to start the AEFUF.



## AEFUF Options

The following table defines the options you can be modify at startup. Any number of command options can be specified on the command line, in any order.

**-c [port]**

Specifies the application port number of the AEFAD server to which the AEFUF server is connected.

**Default value**

2009

**Example**

```
aeufuf -c 2009
```

**-h [host\_name]**

Is obsolete. A warning message is output and the value is ignored. The AEFUF continues normally.

**-i [port]**

Specifies the application port number to which the AEFUF server being started will listen. The AEFC program will be started with a corresponding port number to connect to this AEFUF server. The Client Manager and/or the Communications Bridge also connects to this port number.

**Default value**

2008

**Example**

```
aeufuf -i 2008
```

**-s [host\_name]**

Specifies the TCP/IP host name of the AEFAD to which this AEFUF will connect. The default parameter of localhost specifies connection to an AEFAD server on the local system.

**Default value**

localhost

**Example**

```
aeufuf -s devbox
```

Where can be in IPv4 or IPv6 format.

### **-t [number]**

Specifies a value between 0 and 31 that determines the level of information logged to an internal AEFUF program trace file. The log file is named lg-aefuf-<procid>.log (where <procid> is the process number of the AEFUF) containing the trace data. For Windows, this file is created in the %USERPROFILE%\AppData\Local\CA\Genxx\logs\server directory. For UNIX and Linux, this file is created in the directory from which the AEFUF was invoked. The least amount of information will be provided with a value of 0, and the most information will be provided with a value of 31. However, as this value is increased, the size of the file created will also increase, to the extent that at -t 31, an enormous amount of disk space will be consumed.

**Note:** xx refers to the current release of CA Gen. For the current release number, see the *Release Notes*.

#### **Default value**

0 (no logging)

### **Example**

```
aefuf -t 15
```

### **-b[number]**

Specifies the TCPIP Send/Receive buffer size in 65,536 byte blocks.

#### **Default Value**

1

#### **Maximum Value**

100

### **Example**

```
aefad -b5
```

## AEFUF Command Example

An example follows of a typical UNIX or Linux command line sequence invoking the AEFUF on a system other than the one on which the AEFAD is running:

```
env  
aefuf -h trump -i 2050 -s trumpad -c 2051 -t 15
```

The environment variables are checked to make sure they have been set. The variables are set if they appear on the list displayed.

The AEFUF executable has been specified with its default option settings modified. The modifications are as follows:

- TCP/IP system name where this AEFUF will be started (-h trump)
- Default port number to which this AEFUF will listen has been changed from 2008 to 2050 (-i 2050)
- Name of the AEFAD that this AEFUF will connect to has been identified (-s trumpad)
- Application port number of the AEFAD that this AEFUF will connect to has been identified (-c 2051)
- Program trace logging level has been increased from 1 to 15 (-t 15)

## Features of the AEFUF

This section provides information about the features of the AEF user funnel.

## Trace Logging

The log file lg-aefuf-<procid>.log (where <procid> is the process number of the AEFUF) provides USER and APPL session activity entries and related error messages. To activate activity logging, use the trace option -t when starting the AEFUF server program. For Windows, the lg-aefuf-<procid>.log file is created in the %USERPROFILE%\AppData\Local\CA\Gen xx\logs\server directory. For UNIX and Linux, this file is created in the directory from which the AEFUF was invoked. This file can be used for problem analysis but only by Technical Support. A sample lg-aefuf-<procid>.log file is shown next.

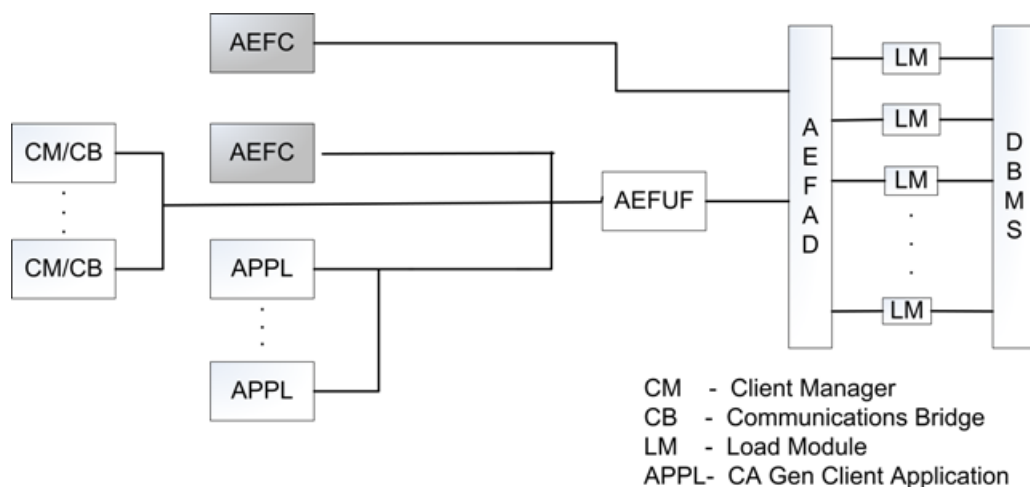
**Note:** xx refers to the current release of CA Gen. For the current release number, see the *Release Notes*.

```
14:13:04.662====>aefuf: Server (ver 8.5) started, users=251, appls=1
14:13:04.672====>aefuf: Server fd = 72, socket = 72, listen port 2008
14:13:04.672====>aefuf: Bind code=0
14:13:04.672====>aefuf: Listen code=0
14:13:04.672====>aefuf: Socket = 20 (port 3534) connected to localhost (2009)
14:13:04.672====>aefuf: FUNN: UF fd = 20, Sock = 20, , WAS 8192 send, 8192 recv
14:13:04.672====>aefuf: FUNN: UF fd = 20, Sock = 20, , NOW 32768 send, 32768 recv
14:13:10.150====>aefuf: USER: UF fd = 132, Sock = 132, user1, sock132, connected
14:13:10.160====>sndpgm: FUNN: UF fd = 20, Sock = 20, user1, sock20, GUI data
14:13:49.657====>rcvpqm: FUNN: UF fd = 20, Sock = 20, user1, sock20, No error
14:13:49.657====>myclose: USER: UF fd = 72, Sock = 72,
14:13:51.159====>aefuf: Connection to localhost (2009) failed, Connection refused
14:13:51.159====>myclose: USER: Sock =
```

## Chapter 4: AEF Client

---

The last task required to establish the production environment is to invoke the AEF Client (AEFC) component of the transaction enabler, as described in the following illustration:



This chapter describes the purpose and function of the AEFC and the procedure required to invoke it.

### Description

The AEFC is a single user application that communicates with the AEFUF, which establishes a connection for that user to the AEFAD. The AEFC can connect directly to the AEFAD and controls one user terminal.

The AEFC is a block mode client program containing terminal screen handling logic. It builds input message blocks and processes output message blocks. A user on a local system starts this program. AEFC connects the user to an AEFUF server on a local or remote system or directly to the AEFAD. TCP/IP is required. The following is an illustration of the main AEFC screen.

```
-----  
transcode clear-screen-options
```

```
CA Gen
```

```
CA Gen Application Execution Facility (IEFAE)
```

```
Transaction Processing Mode
```

```
pf3 = Terminate IEFAE
```

```
-----  
THE AEFC has many of the same functions that the AEF program has. The AEF is  
discussed in the UNIX and Linux or Windows IT user guides.
```

The AEFC is stored in the CA Gen home directory (%IEFH% for Windows systems, or \$IEFH/bin for UNIX and Linux systems).

## Invoking the AEFC

The following sections detail the procedures required to set up and invoke the AEFC.

### Setting Environment Variables

**Note:** See your system documentation for the procedure required to set or change environment variables for your particular platform.

Before the AEFC can be run, You must set the following environment variables:

- IEFH
- PATH
- PTHOME
- PTOPT
- TERM

For more information about these environment variables for both Windows and UNIX/Linux systems, see the appendix "[Transaction Enabler Environment Variable Summary](#) (see page 49)."

## Starting the AEFC

The method for starting the AEFC depends on the type of platform used. The following sections describe the procedure for starting the AEFC on UNIX and Linux and Windows.

### UNIX and Linux

From the command prompt, enter the following:

```
ae fc -option 1 -option 2 ... -option n <Enter>
```

### Windows

To run the AEFC in its default mode, you can simply double-click the program name in the File Manager listing provided. Otherwise, from a command prompt, enter the following:

```
%GENxx%\Gen\VSabc\AEFCN.EXE -option 1 -option 2... -option n
```

or

```
%GENxx%\Gen\VSabc\amd64\AEFCN.EXE -option 1 -option 2... -option n
```

The AEFCN will start and no further operator action is required.

**Note:** VSabc refers to the supported version of Visual Studio. Replace VSabc with VS100 for Visual Studio 2010 and VS110 for Visual Studio 2012. xx refers to the current release of CA Gen. For the current release number, see the *Release Notes*.

You can also create an icon with these already defined options and used to start the AEFC.

## AEFC Options

The following table define the options you can modify at startup. You can specify any number of command options in the command line, in any order.

#### **-a**

Specifies that application security is controlled by application menus and that no CA Gen transaction screen is displayed. When conditions occur that cause the CA Gen transaction screen to appear, normal terminal operation terminates and playback is resumed. This option is used in conjunction with the -p option.

#### **Default value**

None

#### **Example**

```
ae fc -p billing_1 -a
```

**-b**

Specifies that a blank screen should be displayed during playback. This option is used in conjunction with the -p option.

**Default value**

None

**Example**

```
ae fc -p billing_1 -b
```

**-d [number]**

Defines the length of the delay (in seconds) between input messages when replaying a script file. For example, -d 2 activates replay and with 2 seconds delay between message inputs. This parameter is required to activate the replay feature.

This option is used in conjunction with the -p option.

**Default value**

0

**Example**

```
ae fc -p billing_1 -d 2
```

**-h [host\_name]**

Identifies the TCP/IP system name of the local computer system where the AEFAD or AEFUF server is located.

**Default value**

localhost

**Example**

```
ae fc -h devbox
```

Where can be in IPv4 or IPv6 format.

**-i [port]**

Specifies the application port number of the AEFUF server to which this AEFC is connected (or AEFAD if a direct connection).

**Default value**

2008

**Example**

```
ae fc -i 2019
```



**-p [filename]**

Specifies the name of a script file containing input activity that was saved during a previous session.

**Default value**

No playback activity selected

**Example**

```
ae fc -p billing_1 -d 2
```

**-s [filename]**

Specifies the switch that records the current session in a script file with the designated name for replay at a later time using the -p option.

**Default value**

Do not record current session

**Example**

```
ae fc -s billing_1
```

**-t [number]**

Specifies a value between 0 and 31 that determines the level of information logged to an internal AEFC program trace file. The log file is named lg-ae fc-<procid>.log (where <procid> is the process number of the AEFC) containing the trace data. For Windows, this file is created in the %USERPROFILE%\AppData\Local\CA\Gen xx\logs\server directory. For UNIX and Linux, this file is created in the directory from which the AEFC was invoked. The least amount of information will be provided with a value of 0, and the most information will be provided with a value of 31. However, as this value is increased, the size of the file created will also increase, to the extent that at -t 31, an enormous amount of disk space will be consumed.

**Note:** xx refers to the current release of CA Gen. For the current release number, see the *Release Notes*.

**Default value**

0 (no logging)

**Example**

```
ae fc -t 15
```

### **/flshstat**

Causes the log file (defined to aefad by the AEFAD\_STATS environment variable) to be closed and reopened. The administrator can *mv* the log file to a new name, issue the /flshstat command, and then process the renamed log file.

### **-help**

Displays the usage information.

### **Example**

```
aefc -help
```

## AEFC Command Example

An example follows of a typical command line invoking the AEFC:

```
env  
aefc -h trumpuf -i 2050
```

The environment variables are checked to make sure they have been set. The variables are set if they appear on the list displayed.

The AEFC executable has been specified with its default option settings modified. The modifications are as follows:

- TCP/IP host name of the AEFUF (-h trumpuf) to which this AEFC will connect
- Port number of the AEFUF this AEFC will connect to has been identified (-i 2050)

## Features of the AEFC

This section provides information about the features of the AEF client.

### Record/Playback

The AEFC has the ability to record and playback a user's interaction with an application. The recording, or script, may be played back at a later time, as long as the database associated with the particular application has not been changed.

**Note:** For more information about the record/playback feature, see the *UNIX and Linux Implementation Toolset User Guide* or the *Windows Implementation Toolset User Guide*.

## Environment Variable Copy

The environment variable copy feature is used to override the values of the environment variables for a user connected to a particular AEFAD server when it was started. Values are changed without stopping and restarting the AEFAD server.

To override the original variable values, an aeflogin file must exist in the user's current working directory when the AEFCD program is initiated. The new values are read from the file when the AEFCD starts up. The file can be named either .aeflogin or aeflogin.

The environment variables are defined in the aeflogin file with the following syntax. You can use any one of the following:

- `setenv server_environment_variable value`
- `setenv server_environment_variable $name`

## Controlling the AEFAD Server

You can start, stop, or modify certain operational parameters of the AEFAD server to which this client is connected. These commands are entered onto the Application Execution Facility screen by any user connected to the AEFAD using the AEFCD program. The AEFAD-related commands that can be entered from the AEFCD are as follows:

### **`/lock [number] [load_module]`**

Locks the designated number of copies of a given application load module into memory and prevents them from being deleted for this AEFAD server.

#### **Example**

```
/lock 3 maincust
```

### **`/unlock [number] [load_module]`**

Unlocks the designated number of copies of a given application load module in memory, allowing them to be deleted for this AEFAD server.

#### **Example**

```
/unlock 2 maincust
```

### **`/load [load_module]`**

Loads a given application load module into memory and locks it for this AEFAD server. If the value of AEFAD parameter -a is reached, then the last least recently used load module will be removed from memory.

#### **Example**

```
/load maincust
```

**/delete [number] [load\_module]**

Deletes the designated number of copies of a given application load module from memory for this AEFAD server.

**Example**

```
/delete 5 maincust
```

**/reset [number] [load\_module]**

Resets the use counter (the number of times each load module is accessed) for a given application load module residing in memory for this AEFAD server. If the name of the load module is ALL, the use counters for all load modules are reset. If the name of the load module is TRANS, the total transaction counter (the number of times the AEFAD has been accessed) for the AEFAD server is reset.

**Example**

```
/reset all
```

**/dump [sock\_number]**

Dumps the internal program structure of the process associated with *sock\_number*; individual users (USER), individual user funnels (FUNN) and all associated users, or individual applications (APPL) for this AEFAD server. The socket number can be found using the *DISPLAY* transaction.

**Example**

```
/dump 7
```

**/kill [sock\_number]**

Cancels whatever process is associated with *sock\_number*; individual users (USER), individual user funnels (FUNN) and all associated users, or individual applications (APPL) from memory for this AEFAD server. This command is normally only used to clean up hung sessions due to TCP/IP network problems. The socket number can be found using the *DISPLAY* transaction.

**Example**

```
/kill 2050
```

**/aenv**

Updates the aenv file the AEFAD uses in memory. This is useful if additional applications are available after the AEFAD has been started.

**Example**

```
/aenv
```

**/stopall**

Stops and deletes from memory all inactive and unlocked applications for this AEFAD server.

**Example**

```
/stopall
```

**/stopaef**

Stops this AEFAD server, deletes all applications from memory, and close all connections to existing AEF C clients and AEFUF daemons.

**Example**

```
/stopaef
```

**/env**

Displays the environment variables that are set in the aeflogin or .aeflogin file.

**Example**

```
/env
```

**/-d[number]**

Specifies the time (in seconds) that will be used to determine which active transactions the DISPLAY program will display. Transactions active longer than this value will be displayed. Active transactions are marked as type DACT.

**Example**

```
/-d1
```

**/-e[number]**

Specifies the number of seconds to wait for a TCP/IP send or receive request to complete before the assumption is made that the CA Gen application load module or network node is down. After this timeout is reached, the application or access to the network connection is automatically canceled and restarted.

**Example**

```
/-e3
```

### **`/-f[number]`**

Defines the length of time (in milliseconds) to wait for a TCP/IP send or receive request to be logged in the AEFAD trace file (See command `/-t`). The entries in this log file are used by Technical Support for troubleshooting.

#### **Example**

```
/ - f3000
```

### **`/-m[number]`**

Specifies the maximum number of copies of a particular application load module that are allowed to be loaded into memory. When this number of copies of an application has been loaded, additional users will queue to wait for an available load module.

#### **Example**

```
/ - m4
```

### **`/-q[number]`**

Specifies the number of user requests that must queue to execute the same application load module before an additional load module is loaded by the AEFAD server into memory. An additional load will occur up to the limit of application load modules (specified by the `-m` option). When the limit of applications is reached, an additional load will cause the least recently used inactive load module to be deleted. The depth of the user request queue is affected by the number of active copies of the load module. So the greater the number of active load modules, the actual depth of the queue is greater than the value of `-q`.

#### **Example**

```
/ - q4
```

### **`/-w[number]`**

#### **UNIX and Linux**

Specifies the time (in seconds) of how often active transactions (DACT) checking will occur (looking for transactions that exceed the value of the `/-d` option).

#### **Windows**

Specifies the length of time (in milliseconds) to wait on TCP/IP I/O (select timeout).

#### **Default Value**

5 (UNIX and Linux)

100 (Windows)

#### **Example**

```
/ - w2
```

**`/-s[number]`**

Specifies which options are to be logged to the aestats-<procid> file (where <procid> is the process number of the AEFAD). For Windows, this file is created in the %USERPROFILE%\AppData\Local\CA\Gen xx\logs\server directory. For UNIX and Linux, this file is created in the directory from which the AEFAD was invoked. This number is the sum of the following values which activate particular loggings:

**Note:** xx refers to the current release of CA Gen. For the current release number, see the *Release Notes*.

8

AEFAD CPU utilization statistics

4

AEFAD send/receive statistics

2

APPL statistics

1

USER statistics

Providing a value of 0 disables statistics logging.

**Examples**

`/-s5` (USER and AEFAD send/receive statistics will be logged)

`/-s11` (USER, APPL, and AEFAD CPU utilization statistics will be logged)

**`/-t [number]`**

Specifies a value between 0 and 31 that determines the level of information logged to an internal AEFAD program trace file. The log file is named lg-aefad-<procid>.log (where <procid> is the process number of the AEFAD) containing the trace data. For Windows, this file is created in the %USERPROFILE%\AppData\Local\CA\Gen xx\logs\server directory. For UNIX and Linux, this file is created in the directory from which the AEFAD was invoked. The least amount of information will be provided with a value of 0, and the most information will be provided with a value of 31. However, as this value is increased, the size of the file created will also increase, to the extent that at -t 31, an enormous amount of disk space will be consumed.

**Example**

`/-t 15`

**`/flshstat`**

Causes the log file (defined to aefad by the AEFAD\_STATS environment variable) to be closed and reopened. The administrator can *mv* the log file to a new name, issue the `/flshstat` command, and then process the renamed log file.

## Realtime Monitoring

The DISPLAY transaction is used to monitor system activity in real time and can be started by any user connected to the AEFAD server. Monitoring AEFAD server activity is useful for system and application performance tuning and for problem analysis.

### Starting the DISPLAY Transaction

To start AEFAD monitoring, enter the DISPLAY trancode onto the AEF client screen and press Enter. When DISPLAY's main screen appears, type *all* in the **SELECT==>** entry field to display the current activity, as shown in the following illustration:

```

DISPLAY -----CA Gen ----- ROW (1/24)
COMMAND==> LOG==> N SCROLL==> PAGE
          USERS Connected/Active: 2/1          APPLs Loaded/Active: 3/1
PARMS: -i2009 -q2 -m128 -d5 -w5          75 USER (0.4), 23 APPL (0.4)

TYPE Sock Name Lock Use Aver Last Load Connect AEUSER Active
USER 700 user1 1 13.950 13.950 6.200 0.001 User1
FUNN 7 dskll 12 0.000 0.000 0.000 16.384
USER 701 user4 1 14.100 14.100 6.050 0.050 User4
USER 702 user2 1 14.750 14.750 5.500 0.001 User2
USER 703 user3 1 14.650 14.650 4.700 0.050 User3
USER 704 user5 1 14.350 14.350 3.650 0.001 User5 8
USER 705 user6 1 13.600 13.600 4.750 0.050 User6
USER 706 user7 1 14.500 14.500 1.700 0.050 User7 12
USER 707 user8 1 14.300 14.300 5.000 0.001 User8
USER 708 user9 1 14.850 14.850 2.700 0.050 User9 5
USER 709 user11 1 14.600 14.600 1.450 0.001 User11 10
USER 710 user10 1 16.200 16.200 1.650 0.001 User10
USER 711 user12 1 15.800 15.800 1.550 0.001 User12
USER 14 user13 0.000 0.000 0.850 0.050 User13 6
APPL 6 DISPLAY 1 0.850 0.850 1.500 0.600 14
Enter=refresh, clear=End, pf7=prev, pf8=next V.R: 1.0.3
SELECT==> all
[4] BLOCALHOST

```

The description of the columns shown in the screen are as follows:

- Display Header

#### USERS Connected/Active

Displays the number of users that are connected to this AEFAD and how many of those are active.

#### APPLs Loaded/Active

Displays the number of load modules that are loaded into memory and how many of those are active.



AEFAD information can be any two of the following:

**PARMS**

Displays the parameters and their corresponding values that are set for this AEFAD server.

**STATS**

Displays the total number of transactions requested by the connected users and the ratio of transaction requests per second (for example 75 USER (0.4)), along with the total number of transactions requests per application and the ratio of transaction requests per application per second (for example 23 APPL (0.4))

**CPU**

Displays the current CPU utilization (ex. **0% USR, 0% SYS, 99% IDLE**)

The default AEFAD Info displayed is PARMS & STATS, and can be changed using the **SELECT===>** entry field.

■ **TYPE**

Type of information being displayed. The values are as follows:

**USER**

Connected users.

**APPL**

Active CA Gen application load modules.

**FUNN**

Connected user funnels.

**TRAN**

Active CA Gen application procedures.

**LQUE**

CA Gen application load modules queued. For remainder of columns, LQUE is illustrated just as APPL.

- Sock

The socket number that the AEFAD server uses to communicate to the USER, APPL, or FUNN.

**FUNN rows**

Value represents the socket number associated with this User Funnel.

**USER rows**

Value represents the socket number associated with this user. If the USER communications through a particular User Funnel, then the socket numbers reflect the User Funnel. For example, if FUNN is using socket 7, then all USERS associated with the FUNN will be issued socket numbers beginning with 700.

**APPL rows**

Value represents the socket number associated with this load module.

**TRAN rows**

Value represents the socket number of the related APPL.

- Name

**FUNN rows**

Value represents the host name where the User Funnel is running.

**USER rows**

Value represents either the logon ID or the value of the \$IEF\_USER\_ID environment variable that was set using the environment variable copy feature.

**APPL rows**

Value represents the load module.

**TRAN rows**

Value represents the trancode name of the CA Gen procedure.

- Lock

The color of the load module name is yellow when the application is locked in memory. Set using the /lock [number] [load\_module] command.

- Use

**USER rows**

Value represents the number of transactions entered by the user.

**FUNN rows**

Value represents the number of transactions passing through the funnel.

**APPL rows**

Value represents the number of transactions processed for the load module.

**TRAN rows**

Value represents the number of times the procedure was executed.

## ■ Aver

**USER rows**

Value represents the user's average response time (in milliseconds) within the system, excluding any network time.

**APPL rows**

Value represents the load module's average response time (in milliseconds) when processing transactions within the system.

**TRAN rows**

Value represents the transaction's average response time (in milliseconds) within the system.

## ■ Last

**USER rows**

Value represents the response time (in seconds and milliseconds) of the last transaction entered by that user.

**APPL rows**

Value represents the response time (in seconds and milliseconds) of the last transactions processed by this load module.

**TRAN rows**

Value represents the response time (in seconds and milliseconds) of the last transaction processed.

## ■ Load

**USER rows**

Value represents the connect time (in seconds and milliseconds) for that user.

**APPL rows**

Value represents the length of time it took to load the application load module.

**TRAN rows**

Value represents the length of time application procedures have been active.

- Connect

- APPL rows**

- Value represents the amount of DBMS connect time (in seconds and milliseconds)

- USER rows**

- Value represents the total amount of connect time (in seconds and milliseconds)

- AEUSER

- This is the value of the \$AEUSER environment variable set using the environment variable copy feature.

- Active

- This is the sock number of the related active USER or APPL.

A SELECT field at the bottom of the Display screen allows you to display one or more specific rows. Enter a column name followed by one or more expected column values. The valid SELECT field values are as follows:

**all**

Selects all rows

**type <val1> <val2>**

Selects rows where the type column contains values val1 or val2. Values are either USER, APPL, FUNN, TRAN, or LQUE.

**name <val1> <val2>**

Selects rows where the Name column contains val1 or val2.

**<val1> <val2>**

Alters the AEFAD info displayed along the 4th line of the display. Choices to display are PARMS, CPU, or STATS. Val1 is displayed to the left, while val2 is displayed to the right. The default is PARMS to the left and STATS to the right.

**sock <val1>**

Selects rows where the Sock column contains value val1. The value represents an active socket number.

**<F3>**

Exits Display

# Chapter 5: Using the AEFAD and AEFUF as Windows Network Services

---

This appendix provides the information necessary to implement the AEFAD and AEFUF as network services on the Windows workstation. If you are not familiar with the services feature of Windows, see the System Guide shipped with your workstation software.

The Transaction Enabler (TE) is primarily comprised of two programs: the AEF Asynchronous Daemon (AEFAD) and the AEF User Funnel (AEFUF). These programs can be installed and controlled as network services on a Windows workstation using the Services option of the Control Panel. This option provides you with a convenient method to start, stop, pause, and continue operation of these programs without having to return to the AEF Client (AEFC) to explicitly enter the necessary command lines and associated parameters.

## Installation

Included with the Windows version of the transaction enabler are two programs that serve as the default mechanism for registering (regTE.cmd) and unregistering (unregTE.cmd) the TE as Windows services.

- regTE.cmd  
Updates the registry and installs (or registers) the AEFAD and AEFUF as services
- unregTE.cmd  
Updates the registry and removes (or unregisters) the AEFAD and AEFUF as services

If required, the AEFAD and AEFUF may be configured to start automatically whenever the Windows workstation is started.

**Note:** The user (de) installing the AEFAD and AEFUF as services must have Administrator privileges on Windows. All DLL's needed by the AEF(AD/UF) and the server modules must be in the system path. Any trace files produced by the AEF(AD/UF) or the server modules will be placed in the %TEMP% directory.

## Operation

The Services control panel is used to set up and control network services. Using this control panel, multiple instances of the AEFAD and AEFUF may be installed by specifying different TCP/IP port addresses and different instance numbers. The TE services are given the following names by regTE.cmd:

```
TE-AD-{version}-{instance}  
TE-UF-{version}-{instance}
```

AD represents the AEFAD, and UF represents the AEFUF. For example, to run two instances of the AEFAD/AEFUF, the names would be similar to the following:

```
TE-AD-8.5-1  
TE-UF-8.5-1  
TE-AD-8.5-2  
TE-UF-8.5-2
```

## regTE.cmd Parameters

The regTE.cmd script takes the following parameters:

**1**

Required. The current version of CA Gen being used.

**2**

Required. A sequential number used to identify or distinguish between currently active instances of the AEFAD and AEFUF.

**3**

Optional. The full path specification of the directory that contains the AEFAD and AEFUF services executables (aefadsvr.exe and aefufsvr.exe, respectively). This parameter defaults to the current value of %GENxx%Gen\VSabc.

**Note:** VSabc refers to the supported version of Visual Studio. Replace VSabc with VS100 for Visual Studio 2010 and VS110 for Visual Studio 2012. xx refers to the current release of CA Gen. For the current release number, see the *Release Notes*.

**4**

Optional. The number of the port the AEFAD listens to for connection requests. The AEFUF will automatically be configured to connect to this port. The default is 2009.

5

Optional. The number of the port the AEFUF listens to for connection requests from clients (such as Communications Bridges and Client Managers). The default is 2008.

6

Optional. The full path specification of CA Gen installation. This parameter defaults to the current value of the GENxx environment variable.

**Note:** xx refers to the current release of CA Gen. For the current release number, see the *Release Notes*.

7

Optional. The full path specification of the directory that contains the aeenv files and associated inqload directories. This parameter defaults to the current value of the AEPATH environment variable.

8

Optional. The trace level for the AEFAD, AEFUF, and all servers started through the AEFAD. When tracing is enabled (using a value greater than 0) the trace files (lg-<procname>-<procid>.log) will reside in the directory %USERPROFILE%\AppData\Local\CA\Gen xx\logs\server. The default is 0 (no tracing).

**Note:** xx refers to the current release of CA Gen. For the current release number, see the *Release Notes*.

## regTE.cmd Example

Following code is an example of a typical command line invoking the regTE.cmd script to install the AEFAD and AEFUF as services:

```
set AEPATH %GENxx%\Gen\VSabc;C:\application1;C:\application2
regTE.cmd 8.5 1 "%GENxx%\Gen\VSabc" 2050 2051
```

This code starts 1 instance of both the AEFAD and AEFUF identified as TE-AD-8.5-1 and TE-UF-8.5-1. The AEFAD and AEFUF service executables are in %GENxx%\Gen\VSabc, and use port numbers 2050 and 2051 respectively. Because the remaining parameters were not provided, the environment variables GENxx and AEPATH are used for locating images, translation files, and applications.

**Note:** VSabc refers to the supported version of Visual Studio. Replace VSabc with VS100 for Visual Studio 2010 and VS110 for Visual Studio 2012. xx refers to the current release of CA Gen. For the current release number, see the *Release Notes*.

## unregTE.cmd Parameters

The unregTE.cmd script takes the following parameters:

**1**

Required. The current version of CA Gen being used.

**2**

Required. A sequential number used to identify or distinguish between currently active instances of the AEFUF.

These parameters will uniquely identify the service names of the AEFAD and AEFUF that are to be removed.

## unregTE.cmd Example

Following is an example of a typical command line invoking the unregTE.cmd script to remove an installed set of AEFAD and AEFUF services:

```
unregTE.cmd 8.5 1
```

This will remove the installed set of AEFAD and AEFUF services identified as TE-AD-8.5-1 and TE-UF-8.5-1.

## Recommended Steps

The use of regTE.cmd and unregTE.cmd should be executed in the following order when being used interactively:

**Follow these steps:**

1. Execute the command, regTE.cmd, to register the AEFAD and AEFUF.
2. Start the AEFAD and AEFUF from the Start menu: Start / Settings / Control Panel / Administrative Tools / Services.
3. Execute your application(s).
4. Stop the AEFAD and AEFUF from the Start menu: Start / Settings / Control Panel / Administrative Tools / Services.
5. Un-register the AEFAD and AEFUF with the command, unregTe.cmd.

**Note:** You can execute steps 2, 3, and 4 repeatedly while the AEFAD and AEFUF are registered.



# Chapter 6: Transaction Enabler Environment Variable Summary

---

This appendix discusses the transaction enabler environment variables.

## Transaction Enabler Environment Variables

The following table lists the transaction enabler environment variables that are applicable to the AEFAD, AEFUF, and/or AEFC for both the Windows and UNIX and Linux operation systems:

### **AEFAD\_STATS**

This environment variable defines the name of a file to receive transaction execution records. Each transaction execution record has the following fields: date, time, userid, tranid, millisecond-response, cpu-usage. User-written programs may be written to analyze the information in the transaction log file to provide information about the transaction usage, response time, and system usage. The default is no transaction records written. The file is controlled using the /flshstat option with the AEFC.

### **AEPATH**

This environment variable defines a set of directories containing transaction routing tables (aeenv files), and load modules (stored in the subdirectory inqload). These set of directories specify the search order for finding routing tables and load modules. This allows both test and production directories, for example, to exist and be accessed in the proper order.

The IEFH environment variable should also be included as one of the directories, so that files such as the p3270key (Windows) or p3270keys (UNIX and Linux) can be found and used by the AEFAD.

### **AEUSER**

This environment variable is used to allow multiple users with the same logon ID to distinguish themselves to the AEFAD. One possible value of AEUSER is the terminal ID.

### **IEFGXTP**

(UNIX and Linux only) This environment variable represents the directory that contains the codepage translation files.

#### **IEFH**

This environment variable represents the home directory for the implementation toolset, which contains the CA Gen libraries and binaries. This variable cannot be overridden by an individual user connected to an AEFAD production server.

#### **IEF\_USER\_ID**

This environment variable is used to set a unique value for the CA Gen logon ID for the current user session. A user should set this value when accessing the same application on one system in multiple user sessions. Used with diagram trace.

#### **LD\_LIBRARY\_PATH**

(Solaris and Linux only). This environment variable specifies the shared library path(s) for Solaris systems. This path is needed so that all shared libraries can be found at the time of application execution (i.e., when you are running the application, not building the application). Possible library paths to include are the DBMS library path, and the CA Gen block mode runtime library path.

#### **LIBPATH**

(AIX only). This environment variable specifies the shared library path(s) for AIX systems. This path is needed so that all shared libraries can be found at the time of application execution (i.e., when you are running the application, not building the application). Possible library paths to include are the DBMS library path, and the CA Gen block mode runtime library path.

#### **PATH**

This environment variable defines a search path that enables DBMS and other necessary executables to be located without the need to specify the full directory path. The variable entry must include the string \$IEFH/bin for UNIX and Linux systems, or %IEFH% for Windows systems.

#### **PTHOME**

This environment variable specifies the directory where profile table files are to be created or where previously created files are located.

#### **PTOPT**

This environment variable specifies the type of profiling for this user. The following options are available:

##### **Yes**

Specifies that Profile Table files are to be used for this user.

##### **Yy**

Specifies that each user's profile stack is to be stored in a file with data compression.

**SHLIB\_PATH**

(HP Itanium only) This environment variable specifies the shared library path(s) for HPUX systems. This path is needed so that all shared libraries can be found at the time of application execution (i.e., when you are running the application, not building the application). Possible library paths to include are the DBMS library path, and the CA Gen Block Mode runtime library path.

**TERM**

(UNIX and Linux only) This environment variable specifies the terminal type. For example, when the variable is set to vt100, it specifies that the user is interfacing with the system using a vt100 terminal.

**TRANSLAT**

This environment variable is used to set the translation details flag, which provides translation duration information in the log files. The default is NO.



# Index

---

## A

### AEF asynchronous daemon

- application production environment figure • 11
- command example • 20
- controlling server • 35
- description description
  - AEF asynchronous daemon • 11
- example • 20
- invoking • 12
- sample lgxxxxxx file • 21
- server commands entered using AEF client • 35
- setting environment variable • 12
- starting • 12
- statistics logging • 22
- u option • 19
- UNIX and Linux • 13
- Windows NT • 13

### AEF client

- AEFAD server commands, using AEFC-table • 35
- application production environment, figure • 29
- command example • 34
- command options table • 31
- controlling AEFAD server • 35
- description • 29
- display screen column definitions, table • 40
- display screen select field values, table • 40
- environment variables table • 30
- features • 34
- invoking • 30
- options • 31
- realtime monitoring • 40
- sample display screen • 40
- setting environment variable • 30
- starting • 31
- starting, DISPLAY transaction • 40
- UNIX and Linux • 31
- Windows NT • 31

### AEF client feature

- environment variable copy • 35
- record/playback • 34

### AEF user funnel

- AEFUF • 23
- CA Gen production environment • 23
- command option list table • 24

- invoking • 23
- setting environment variable • 24
- starting • 24
- UNIX and Linux • 24
- windows NT • 24

### AEFAD and AEFUF as Windows NT network services

- installation • 45
- operation • 46
- regTE.cmd command parameters table • 46
- regTE.cmd parameters • 46
- unregTE.cmd command parameter table • 48
- unregTE.cmd parameter • 48

### AEFAD, -u option details • 19

### AEFC • 29

### AEFUF

- command example • 26
- description • 23
- options • 25

### application production environment

- AEFC figure • 29

## C

### CA Gen production environment

- AEFAD figure • 11
- AEFC figure • 29
- AEFUF • 23
- required tasks • 9

- command example, AEFAD • 20
- controlling AEFAD server • 35

## D

### describing AEFUF • 23

### description

- AEF Client • 29

### details, -u option • 19

### DISPLAY transaction

- column definitions, table • 40
- realtime monitoring • 40
- sample screen • 40
- select field values, table • 40

## E

### environment variable

- AEF client transaction enabler • 30

---

- transaction enabler, AEFAD • 12
- transaction enabler, AEFUF • 24
- transaction enabler, NT • 49
- environment variable copy , AEF client feature • 35
- example
  - u option • 20
  - AEF asynchronous daemon • 20
  - AEF client command • 34
  - AEFAD command • 20
  - AEFUF command • 26
  - regTE.cmd • 47
  - unregTE.cmd • 48
- execution environment
  - application production environment • 8
  - installing transaction enabler • 10
  - required tasks • 9
  - structure of production environment • 9

## F

- features, AEF client • 34

## I

- installing transaction enabler
  - installing AEFAD and AEFUF as NT network services • 45
- invoking
  - AEFAD • 12
  - AEFC • 30
  - AEFUF • 23

## L

- logging trace • 21

## O

- operation, regTE.cmd parameter • 46
- options
  - AEF client • 31
  - AEFUF • 25

## P

- production environment, structure • 9
- production execution environment
  - installing transaction enabler • 10
  - prerequisite tasks • 8
  - required tasks • 9
  - structure • 9
  - tasks required • 9

## R

- realtime monitoring, AEF client • 40
- record/ playback, AEF client feature • 34
- regTE.cmd parameter
  - installation • 45
  - table • 46
- regTE.cmd, example • 47
- required tasks
  - production execution environment • 9

## S

- sample • 21
  - AEFAD lgxxxxxx file • 21
  - display screen - AEFC • 40
- setting environment variable
  - AEFAD • 12
  - AEFClient • 30
  - AEFUF • 24
  - transaction enabler-NT • 49
- starting
  - AEF client • 31
  - AEF client on NT • 31
  - AEF client on UNIX and Linux • 31
  - AEFUF • 24
  - AEFUF, UNIX and Linux • 24
  - DISPLAY transaction • 40
  - DISPLAY transaction on NT • 31
  - UNIX and Linux DISPLAY transaction • 31
- starting AEFAD
  - NT • 13
  - UNIX and Linux • 13
- statistics logging, AEFAD • 22
- structure of production environment • 9
- summary, transaction enabler environment variable • 49

## T

- trace logging • 21
- transaction enabler
  - environment variables, NT • 49
  - structure of production environment • 9

## U

- u option
  - details • 19
  - example • 20
- unregTE.cmd

---

- example • 48
- unregTE.cmd parameter
  - installation • 45
  - table • 48
- using AEFAD and AEFUF as NT network services
  - installation • 45
  - operation • 46
  - regTE.cmd command parameter table • 46
  - regTE.cmd parameter • 46
  - unregTE.cmd command parameter table • 48
  - unregTE.cmd parameter • 48