



Amduus Information Works, Inc.
<http://amduus.com>

Table of Contents

Introduction.....	3
The Web Based Viewer.....	4
Inserting Services Into The System.....	4
Observing Zeno's Run State.....	6
Observing Job Processing.....	8
Drilling Into More Information On The Job	10
Turning Off Zeno.....	12

Introduction

This document should be read by programmers, administrators, and management.

The Zeno processor is a framework of scripts and code that allows the user to spawn off on or more processes on one or more machines to handle requests.

Be sure to see the Ezine also included in the /doc directory for programmer oriented documentation.

Note the web pages created are very basic on purpose. This is so you can easily update the look and feel for your company.

The Web Based Viewer

Zeno comes with a set of web pages that can be run on Webspeed (or one of the Webspeed alternative tools) to show what is happening within the various processes. It also will cover the jobs that are waiting in queue to be processed and information about them.

Note the web pages created are very basic on purpose. This is so you can easily update the look and feel for your company.

I will explain the screens with code and commands to show what activities beget what information on the various pages.

Inserting Services Into The System

The systems purpose is to act as a framework to request work from a Zeno client to a Zeno server process. In order for a Zeno server process to perform some work on the programmer's behalf, the programmer needs to develop that code and insert it into the system with the APIs available:

```
/* Test program to populate the services available in the system */  
  
DEFINE VARIABLE hjobclient AS HANDLE NO-UNDO.
```

Zeno Processor

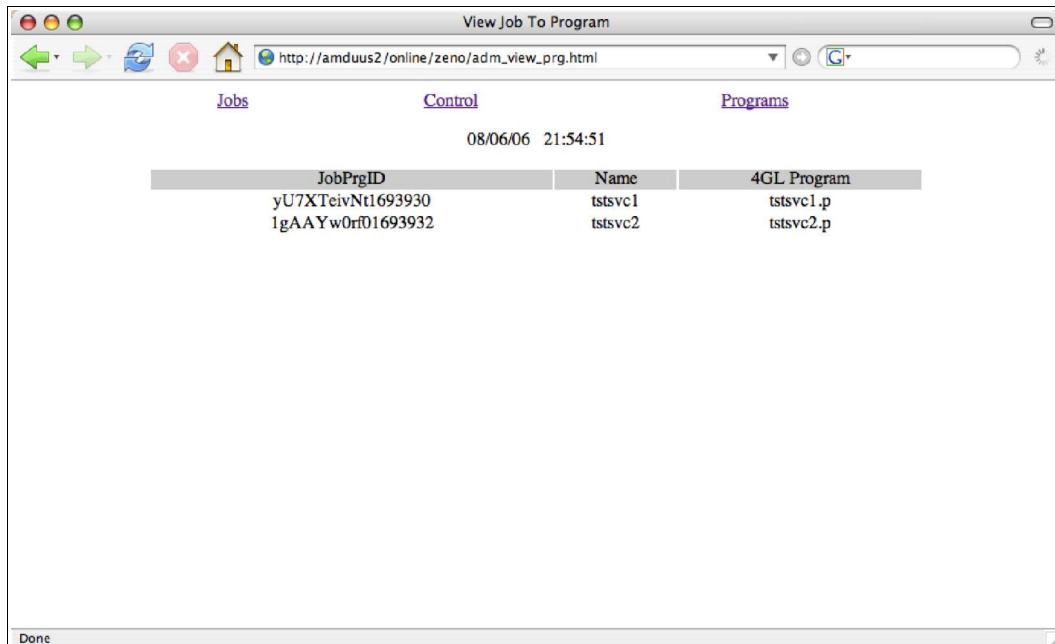
```
RUN jobclient.p PERSISTENT SET hjobclient.
```

```
RUN DefineService IN hjobclient (INPUT "tstsvc1", INPUT "tstsvc1.p").
```

```
RUN DefineService IN hjobclient (INPUT "tstsvc2", INPUT "tstsvc2.p").
```

```
DELETE OBJECT hjobclient.
```

After executing such a program, the “Programs” tab will appear as this:



The screenshot shows a web browser window with the address bar containing `http://amduus2/online/zeno/adm_view_prg.html`. The page has three tabs: "Jobs", "Control", and "Programs". The "Programs" tab is active, displaying a table with the following data:

JobPrgID	Name	4GL Program
yU7XTeivNtl693930	tstsvc1	tstsvc1.p
1gAAYw0rf01693932	tstsvc2	tstsvc2.p

The status bar at the bottom of the browser window shows "Done".

This identifies the name of the service used with APIs and which program will be used to process that service. The JobPrgID is of no particular interest and generated randomly upon creation of the record. Its purpose is for future web manipulation of the records as well as a means of uniquely identifying the record in the table.

Observing Zeno's Run State

In order for the Zeno system to actually do anything, a set of processes need to be started. Each service is a process within the system. You can have more than one process handling a given service on the same machine. Of course, you can have more than one process across multiple machines to handle a service for heavy loads or services that are not so quick (think credit card processing.)

The screenshot shows a web browser window with the address bar containing http://amduus2/online/zeno/adm_view_control.html. The page content includes a navigation menu with 'Jobs', 'Control', and 'Programs'. Below the menu, the date and time '08/06/06 22:06:48' are displayed. A table with the following data is shown:

ActionAlert	JobCtrlID	JobID	Operation	PrcHostName	PrcHostPID	PrcName
STOP	1luYFhlVH274		Stopped	amduus2	5727	tstsvc2
STOP	6hGZ5WDaf1203		Stopped	amduus2	5728	tstsvc3
STOP	wtYWQzBvLW421		Stopped	amduus2	5726	tstsvc1

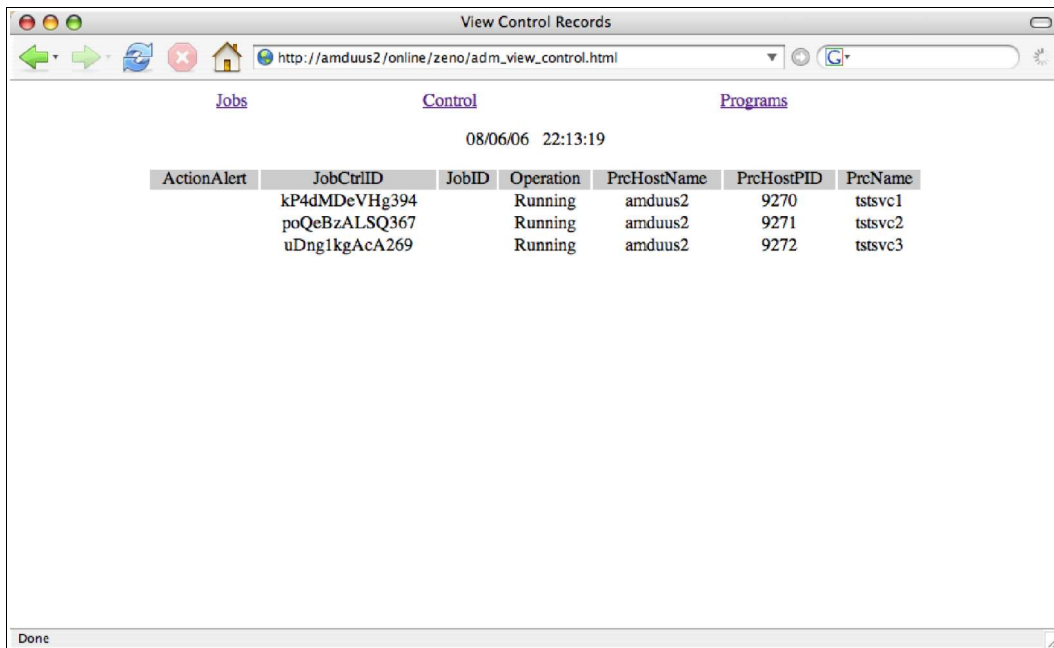
When the system is turned off, you should see something like the above. This information describes which records have received an Action Alert – which basically is a means of telling the actual server process something (usually starting and stopping.) The Operation field describes what state the server is running in. The PrcHostName describes which machine the process is running on. As a means of identifying the process, the PID of it is also provided. Finally, we see what service the server process is meant to work with under PrcName.

Zeno Processor

Zeno is UNIX oriented – so we have a set of scripts set up where the administrator can start up a server process with a command similar to the following:

```
mngprc.bash -start -prcname tstsvc1 &  
mngprc.bash -start -prcname tstsvc2 &  
mngprc.bash -start -prcname tstsvc3 &
```

After starting up these processes, the screen will change to something like the following:



The screenshot shows a web browser window with the title "View Control Records". The address bar contains the URL "http://amduus2/online/zeno/adm_view_control.html". The page content includes three tabs: "Jobs", "Control", and "Programs". The "Control" tab is active, displaying a timestamp "08/06/06 22:13:19" and a table of running processes. The table has seven columns: ActionAlert, JobCtrlID, JobID, Operation, PrcHostName, PrcHostPID, and PrcName. Three rows of data are shown, all with "Running" operations.

ActionAlert	JobCtrlID	JobID	Operation	PrcHostName	PrcHostPID	PrcName
	kP4dMDeVHg394		Running	amduus2	9270	tstsvc1
	poQeBzALSQ367		Running	amduus2	9271	tstsvc2
	uDng1kgAcA269		Running	amduus2	9272	tstsvc3

This page identifies three processes running and what services they will provide to the Zeno clients using the APIs to write jobs to the system. Lets take a look at 9270:

```
# ps -fp 9270 --width=200
UID          PID  PPID  C  STIME TTY          TIME CMD
root         9270   9267  17  22:13 pts/0        00:00:30 /usr/dlc/bin/_progres
-b -pf /home/sauge/code/progress/zeno/db/zeno.pf -p jobsvr.p
```

Be sure to see the Ezine on how to handle system crashes that leave the state records in the wrong operation.

Observing Job Processing

Now that there are processes out there waiting for something to do, we will use this example program to use the Client APIs to submit jobs to the system:

```
/* Useful for populating the jobs quickly */

DEFINE VARIABLE hjobclient AS HANDLE NO-UNDO.
DEFINE VARIABLE cJobID      AS CHARACTER NO-UNDO.

RUN jobclient.p PERSISTENT SET hjobclient.

RUN SubmitJob IN hjobclient (INPUT "tstsvc3", INPUT 1, INPUT "Job1",
OUTPUT cJobID). /* 101 */
RUN SubmitJob IN hjobclient (INPUT "tstsvc4", INPUT 1, INPUT "Job1",
OUTPUT cJobID). /* 100 */
RUN SubmitJob IN hjobclient (INPUT "tstsvc1", INPUT 1, INPUT "Job1",
OUTPUT cJobID).
RUN SubmitJob IN hjobclient (INPUT "tstsvc2", INPUT 1, INPUT "Job1",
OUTPUT cJobID).

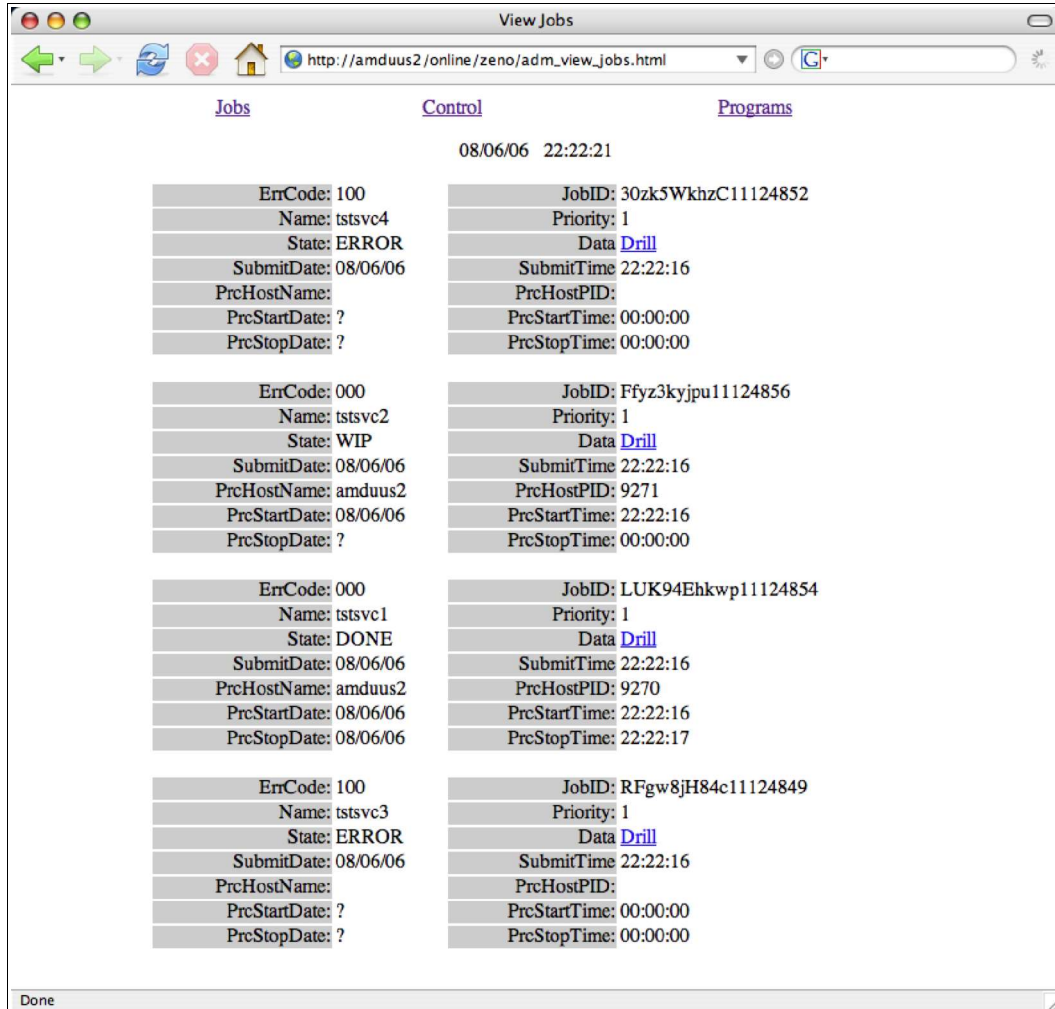
DELETE OBJECT hjobclient.
```

This code is actually some test code where errors will be generated. See the E-

Zeno Processor

See [Zine](#) for more information about what the error numbers mean.

Running this program submits jobs into the system and the various processes will look for jobs of the same service name it is started with.



The screenshot shows a web browser window titled "View Jobs" with the URL http://amduus2/online/zeno/adm_view_jobs.html. The page has three tabs: "Jobs", "Control", and "Programs". The "Jobs" tab is active, displaying a list of jobs. At the top right of the job list, the date and time "08/06/06 22:22:21" are shown. Each job is represented by two columns of information: job details on the left and process details on the right. The jobs listed are:

JobID	Name	State	ErrCode	Priority	SubmitTime	PrcHostPID	PrcStartTime	PrcStopTime
30zk5WkhzC11124852	tstsvc4	ERROR	100	1	22:22:16		00:00:00	00:00:00
Ffyz3kyjpu11124856	tstsvc2	WIP	000	1	22:22:16	9271	22:22:16	00:00:00
LUK94Ehkwp11124854	tstsvc1	DONE	000	1	22:22:16	9270	22:22:16	22:22:17
RFgw8jH84c11124849	tstsvc3	ERROR	100	1	22:22:16		00:00:00	00:00:00

The "Done" status is visible at the bottom left of the browser window.

There is a lot of information available about each job on this page. Each job is given a unique JobID which is passed back to the program submitting the job. This is the “name” of the job as far as the APIs are concerned. The State field

describes if the job is waiting to be processed, done, in process, or errored out for some reason. It names the service expected to process the job as well as the priority given by the client program asking Zeno to process it. Once the job request has been picked up, it will contain the host name and the PID of the process that performed the processing. The submit date and time denote when the job was added to the system by the client API. The process start time and date denote when a server process picked up the record and started performing the service the record requested of it. And of course, the ProcStopDate and time denote when the server process finished with the record.

The date and time information can help you determine the load and how fast job requests are being processed by the processor.

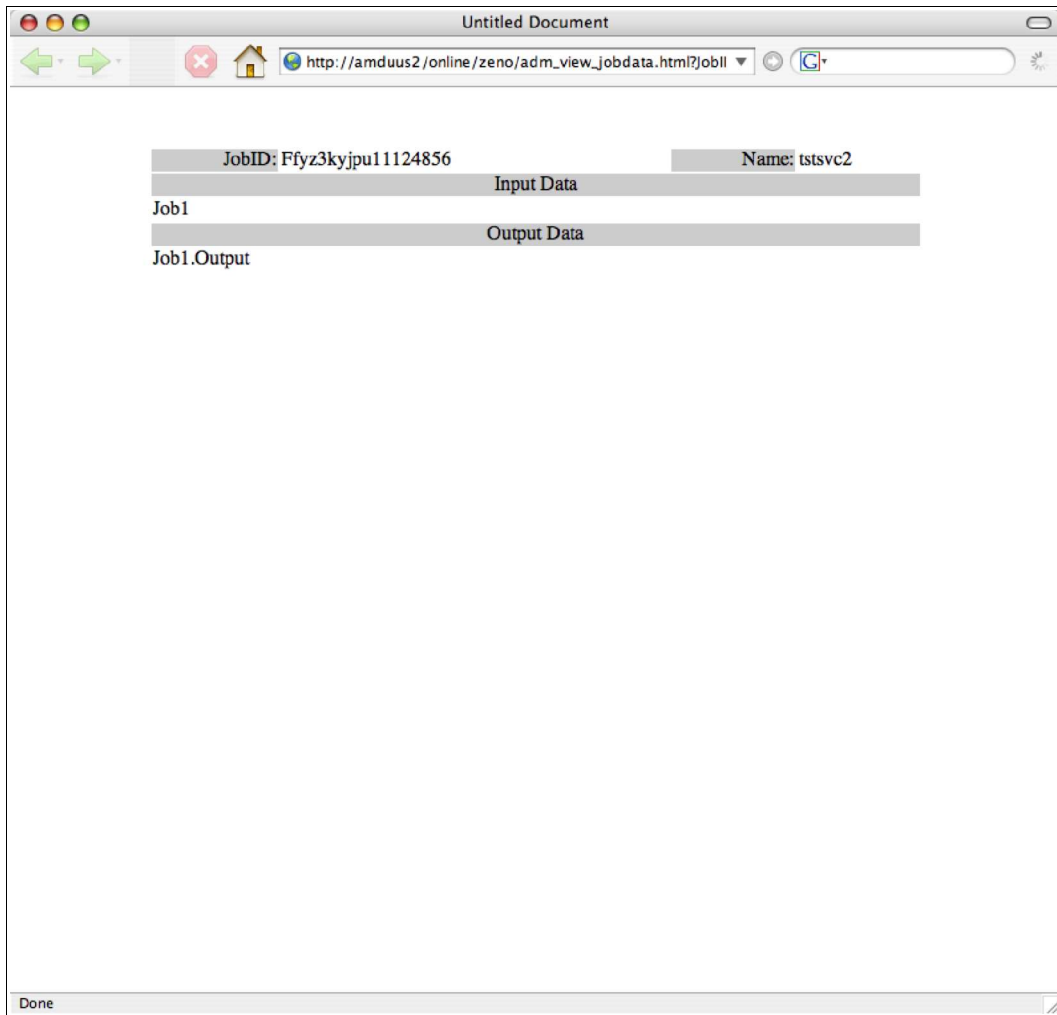
Drilling Into More Information On The Job

Often, the Program In data and the Program out data may be lengthy, so a “drill” link is made available to the browser to help the administrator, problem solver, or programmer can use to examine the data the server ended up working with and returned.

Below we can see the client API sent in the data “Job1” and the program generated the data “Job1 Output” in the data output field¹.

¹ As the E-Zine notes, the output field might actually contain a key into other records or a disk file for large result sets (such as generated PDFs.)

Zeno Processor



Turning Off Zeno

Zeno is controlled by a command line – it is generally designed to run as a UNIX process. Here are some example commands:

```
mngprc.bash -stop -hostname amduus2 -prcname tstsvc1
mngprc.bash -stop -hostname amduus2 -prcname tstsvc2
mngprc.bash -stop -hostname amduus2 -prcname tstsvc3
```

These commands stop the processes on the given machine for the given service. The results are shown below:

```
[/home/sauge/code/progress/zeno/script]# ./stoptstsvc.bash
This host is amduus2
This PID is 10677
Stopping all processes named tstsvc1 on amduus2
Stopping 9270 on amduus2
This host is amduus2
This PID is 9270
Starting the processor...
Looking for JobCtrl records to delete...
Deleting Record For PID 5726 on Host amduus2
Deleting Record For PID 5728 on Host amduus2
Deleting Record For PID 5727 on Host amduus2
Done looking for JobCtrl records to delete...
08/06/0622:22:16Running tstsvc1.p!
08/06/0622:22:17Done Running tstsvc1.p!
Stop issued for this process
This host is amduus2
```

Zeno Processor

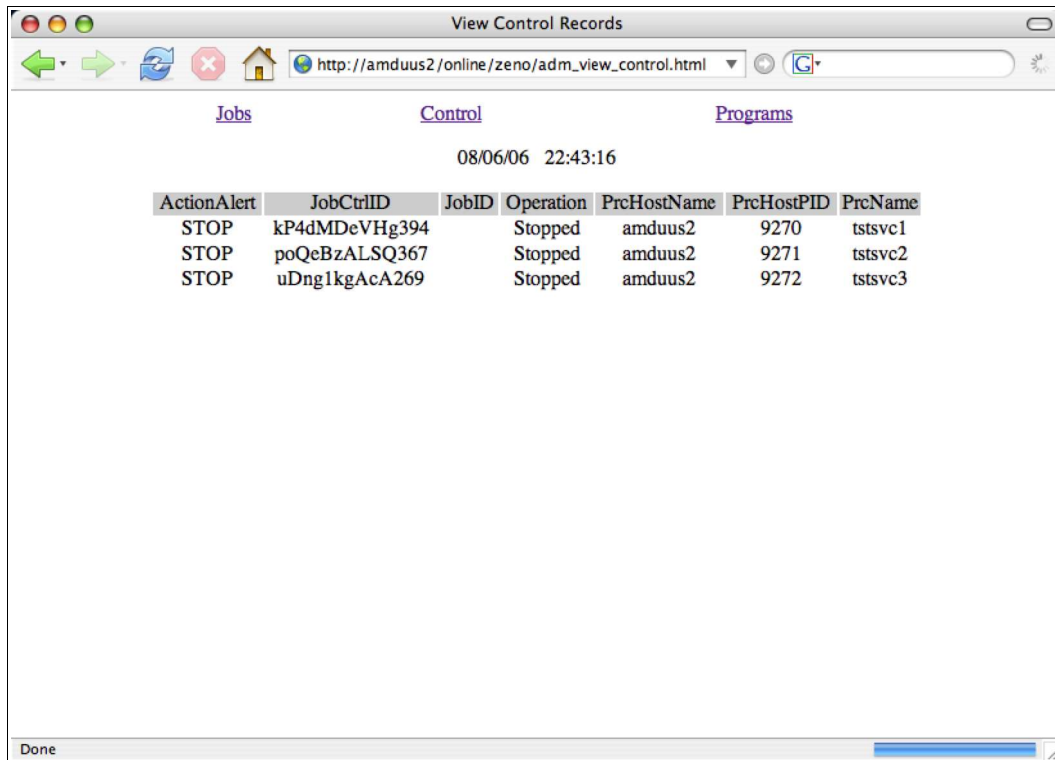
```
This PID is 10682
Stopping all processes named tstsvc2 on amduus2
Stopping 9271 on amduus2
This host is amduus2
This PID is 9271
Starting the processor...
Looking for JobCtrl records to delete...
Done looking for JobCtrl records to delete...
08/06/0622:22:16Running tstsvc2.p!
08/06/0622:22:46Done Running tstsvc2.p!
Stop issued for this process
This host is amduus2
This PID is 10685
Stopping all processes named tstsvc3 on amduus2
Stopping 9272 on amduus2
This host is amduus2
This PID is 9272
Starting the processor...
Looking for JobCtrl records to delete...
Done looking for JobCtrl records to delete...
Stop issued for this process
```

The actual shutdown is a service so there are messages identifying the process and then it identifies which records and processes it is working on.

The ending of the processes (or at least the ones stopped²) should yield a screen like this:

² You do not need to stop all the processes on all the machines to stop a particular Zeno service. This can have the benefit of having Zeno running a service on two machines and allowing one of them to be taken offline and not see an interruption in service.

Zeno Processor



View Control Records

http://amduus2/online/zeno/adm_view_control.html

[Jobs](#) [Control](#) [Programs](#)

08/06/06 22:43:16

ActionAlert	JobCtrlID	JobID	Operation	PrcHostName	PrcHostPID	PrcName
STOP	kP4dMDeVHg394		Stopped	amduus2	9270	tstsvc1
STOP	poQeBzALSQ367		Stopped	amduus2	9271	tstsvc2
STOP	uDng1kgAcA269		Stopped	amduus2	9272	tstsvc3

Done

If you see an Action Alert of STOP and an operation not “Stopped” and there are no jobs in a “WIP” state on the service for that machine – something bad probably happened (disconnected from the database, a programming error in the service, etc.)