
Customization and Usage of ACE

1	Overview	2
1.1	REFERENCES	2
2	reorganization of material in this doc. ...	Error! Bookmark not defined.
2.1	ACE STRUCTURES	ERROR! BOOKMARK NOT DEFINED.
2.1.1	aceProtocolInfo {}	Error! Bookmark not defined.
2.1.2	aceConfigInfo {}	Error! Bookmark not defined.
2.1.3	aceConfigInterfaceInfo {}	Error! Bookmark not defined.
2.1.4	aceInterfaceInfo {}	Error! Bookmark not defined.
2.1.5	aceProtoControlInfo {}	Error! Bookmark not defined.
2.1.6	devBoardParamsType	Error! Bookmark not defined.
2.1.7	configAceStaticInfo, configAceXXXXInfo	Error! Bookmark not defined.
2.2	ACE API	ERROR! BOOKMARK NOT DEFINED.
2.2.1	aceAddressEvent()	Error! Bookmark not defined.
2.2.2	aceInitialize()	Error! Bookmark not defined.
2.2.3	aceStart()	Error! Bookmark not defined.
2.2.4	aceStop()	Error! Bookmark not defined.
2.3	ACE CALLBACKS	ERROR! BOOKMARK NOT DEFINED.
3	ACE Start at Boot Time	Error! Bookmark not defined.
3.1	CUSTOMIZESTARTACE()	ERROR! BOOKMARK NOT DEFINED.
3.1.1	aceStart()	Error! Bookmark not defined.
4	Event Handling: aceTask().....	Error! Bookmark not defined.
4.1	ACE_STATE_START STATE	ERROR! BOOKMARK NOT DEFINED.
4.2	ACE_STATE_RUNNING STATE.....	ERROR! BOOKMARK NOT DEFINED.
5	Configuring ACE	2
5.1	SETTING STATIC IP CONFIGURATION	2
5.1.1	startInfo structure	3
5.2	SETTING DHCP CONFIGURATION	3
5.3	SETTING AUTOIP CONFIGURATION	4
6	Stopping ACE	4
7	Sample Code	5
7.1	USER INTERACTION	5
7.2	IP RECONFIGURATION.....	5
7.2.1	Restrictions, Anomalies, Limitations associated with this app.	Error! Bookmark not defined.

1 OVERVIEW

This document describes how to program changes for ACE. It is not intended to provide knowledge of the Address Resolution Protocols, merely how some of these protocols may be managed in the ACE configuration.

Detailed descriptions of functions and structures can be found in the **NET+OS API Reference**.

A sample application is provided, showing how to stop, change the configuration and restart ACE.

2 REFERENCES

Refer to the NET+OS API Reference for ACE.

3 CONFIGURING ACE

To add or remove a protocol from ACE, the NVRAM parameters (ACE Configuration) must be changed.

These functions store the protocol –specific configuration to NVRAM for the interface identified in the call. (**customizeAceSetInterfaceConfig()** writes all of the protocol-specific ACE configurations, or the ACE configuration for an interface and **customizeAceSetConfig()** writes the entire ACE configuration into NVRAM.)

The APIs presented to applications for this purpose are:

- **customizeAceSetConfig()**
- **customizeAceSetInterfaceConfig()**
- **customizeAceSetStaticConfig()**
- **customizeAceSetRarpConfig()**
- **customizeAceSetDhcpConfig()**
- **customizeAceSetBootpConfig()**
- **customizeAceSetAutoipConfig()**

After setting up the desired configuration info and saving to NVRAM, ACE can be restarted. At this time, ACE will read the configuration parameters from NVRAM.

3.1 Setting Static IP Configuration

To change the static IP configuration, the values in a structure of type **configAceStaticInfo** will be set.

The members: **isConfigValid** and **isEnabled** when set to TRUE, will enable this configuration to be processed by ACE. (Turns static IP on).

Following is a brief description of the parameters in the Static IP Configuration.

auto_assign - when true, causes this configuration to take precedence over other protocols. When false, static IP will be invoked only after other protocols fail.

ip_address, **subnet_mask** and **gateway** are required parameters and are not explained in this document.

name_server_address - may be specified. This is an ip address expressed as a 32-bit value.

3.1.1 startInfo structure

This member of the **configAceStaticInfo** structure contains the following (required) parameters:

protocol – a number defined in `ace_params.h` identifying the protocol (ACE_PROT_STATIC in this instance).

priority – non-negative number. Priority granted to the protocol is inversely proportional to this number (0 is highest priority).

delay_before_start – number of seconds to delay starting this protocol.

shutdown_type – three choices are supported: ACE_ALWAYS_SHUTDOWN, ACE_CONT_IF_GOT_ADDRESS and ACE_NEVER_SHUTDOWN.

3.2 Setting DHCP Configuration

(note: in the following, all ip address parameters are 32 bit words in network byte order.)

To change the DHCP configuration, the values in a structure of type **configAceDhcpInfo** will be set.

The members: **isConfigValid** and **isEnabled** when set to TRUE, will enable this configuration to be processed by ACE. (Turns DHCP on).

Following is a brief description of the parameters in the DHCP Configuration
suggested_ip_address – optionally provided.

server_ip_address – for ACE_RESTART_DHCP_REUSE

gateway – default gateway address

suggested_lease_time – time_t structure.

number_of_retries - -1 is infinite.

lease_start_time – time recorded at start of lease.

dhcp_restart_type – DHCP restart type

need_bcast_response – sets broadcast flag in DHCP message

do_init_delay – initial delay before sending discover message

arp_reply_timeout – reply timeout for ARP probe

desired_params – array of bytes to be used for optional params

num_desired_params – depth of the array above (desired_params)

startInfo – same structure as above (see Static IP section)

3.3 Setting AUTOIP Configuration

To change the AutoIP configuration, the values in a structure of type **configAceAutoipInfo()** will be set.

The members: **isConfigValid** and **isEnabled** when set to TRUE, will enable this configuration to be processed by ACE. (Turns AUTOIP on).

Following is a brief description of the parameters in the AUTOIP Configuration

autoip_local_addr - IP address which AutoIP should initially use when trying to configure an address

startInfo – same structure as above (see Static IP section)

4 STOPPING ACE

Stopping the service is necessary to make changes in the protocols which ACE uses to manage address events.

ACE is stopped by calling **aceStop()**. The only parameter passed in this call is the interface name (e.g.: “eth0”).

5 SAMPLE CODE

5.1 User Interaction

aceTestDHCP.c

The aceTest is invoked through the web page. *(The web page used by this application is directly lifted from another sample. The text and buttons do not identify their real functionality in this instance.)*

To invoke the ACE configuration manager app, open the browser to the page at the ip address of the dev board and follow the active link. The username and password are displayed in plain text. Select the submit button then focus on the serial terminal window.

You will be prompted:

```
“aceTest() - select ace mode by typing one of the following:”  
“autoip static dhcp none”  
“What is your selection?”
```

Choose your protocol (only adding or modifying a protocol is supported) and The first section of code deals with the user. A routine called **getNVParams()** retrieves the NVRAM ACE configuration and displays some IP parameters. Next, the selected mode determines which block of code is executed.

5.2 IP Reconfiguration

In the static IP mode, the **configAceStaticInfo** structure **staticIPInfo** is written to enable and validate the structure. The ip address entered by the user is also written along with a predetermined gateway and subnet mask.

Once this structure is initialized, **customizeAceSetStaticConfig()** is called to store the new configuration info into NVRAM.

After the configuration structure is stored, **customizeStartAce()** is called to restart ACE with the new IP address and other information.

5.3 Sample Application Files

The following files and directories comprise the sample application for ACE.

```
<ROOT_DIR>\_formreply.c  
<ROOT_DIR>\_netarm2.c  
<ROOT_DIR>\aceTestDHCP.c  
<ROOT_DIR>\appconf.h  
<ROOT_DIR>\gen_utils.c  
<ROOT_DIR>\gen_utils.h  
<ROOT_DIR>\project.bld
```

```
<ROOT_DIR>\root.c
<ROOT_DIR>\html2c\_formreply.c
<ROOT_DIR>\html2c\_netarm1.c
<ROOT_DIR>\html2c\_netarm2.c
<ROOT_DIR>\html2c\_netarm3.c
<ROOT_DIR>\html2c\bindata.c
<ROOT_DIR>\html2c\url.c
<ROOT_DIR>\html2c\webpages.web
<ROOT_DIR>\html2c\html\formreply.htm
ROOT_DIR>\html2c\html\netarm1.htm
ROOT_DIR>\html2c\html\netarm2.htm
ROOT_DIR>\html2c\html\netarm3.htm
ROOT_DIR>\html2c\html\Netsilicon.gif
```

5.4 Source ZipFile

The entire project for Green Hills 361 and NET+OS 6.0 is packaged in a file called nahttp_ace.zip