Vendor-Defined Hostid Types

If you are a C programmer and experienced with FLEX*lm*, you can use the FLEXible API to define your own hostid type. If you would like to discuss whether or not vendor-defined hostids are feasible for your application, you can contact GLOBEtrotter technical support.

In the FLEX*lm* kit, we provide a sample C source file, `examples\vendor_hostid\vendor_hostid.c`, in which a fixed vendor-defined hostid is set up. In this section, you can use this file to run through a procedure for setting up a vendor-defined hostid. In a real situation, you would not use a fixed vendor-defined hostid, but would define and call a function that returns the hostid that you want to use.

A vendor-defined hostid can be used on a SERVER or FEATURE line of a license file.

Editing Source Files

You must define your hostid type (for this example, we are using `vendor_hostid.c`), then make sure that the vendor daemon, FLEX*lm* license generators, and your client application can recognize and use your hostid type. Only `lmcrypt` and `makekey` can generate licenses with vendor-defined hostids; on Windows, you cannot use `genlic`.

1. Make a copy of your FLEX*lm* production kit. Follow these instructions using the files in the duplicate kit.
2. Copy `examples\vendor_hostid\vendor_hostid.c` to the `machind` directory.
3. View the file and find the `#define` statements. See `lmclient.h` for `HOSTID` and `LM_VENDOR_HOSTID` definitions.

```
#include "lmclient.h"
#include "lm_attr.h"
#include "string.h"

extern LM_HANDLE *lm_job; /* This must be the current job! */

/* This example returns only 1 hostid */
#define VENDEF_ID_TYPE HOSTID_VENDOR+1
#define VENDEF_ID_LABEL "VDH"
#define VENDEF_ID "12345678"

/*
 * x_flexlm_gethostid() - Callback to get vendor-defined hostid.
 *(Sorry about all the windows types for this function...)
 */
```

```
HOSTID *
#ifdef PC
LM_CALLBACK_TYPE
#endif /* PC */
/*
 * IMPORTANT NOTE:  This function MUST call l_new_hostid() for
 *                  a hostid struct on each call.
 *                  If more than one hostid of a type is
 *                  found, then call l_new_hostid for each
 *                  and make into a list using the ‘next’ field.
 */

x_flexlm_gethostid(idtype)
short idtype;
{
      HOSTID *h = l_new_hostid();

      memset(h, 0, sizeof(HOSTID));
        if (idtype == VENDEF_ID_TYPE)
        {
                h->type = VENDEF_ID_TYPE;

                strncpy(h->id.vendor, VENDEF_ID, MAX_HOSTID_LEN);
            h->id.vendor[MAX_HOSTID_LEN] = 0;
                return(h);
        }
        return((HOSTID *) NULL);
}



void
x_flexlm_newid(id)

HOSTID *id;

{
  LM_VENDOR_HOSTID h;

       memset(&h, 0, sizeof (h));
       h.label = VENDEF_ID_LABEL;
       h.hostid_num = VENDEF_ID_TYPE;
       h.case_sensitive = 0;
       h.get_vendor_id = x_flexlm_gethostid;
       if (lc_set_attr(lm_job, LM_A_VENDOR_ID_DECLARE,
              (LM_A_VAL_TYPE) &h))
              lc_perror(lm_job, "LM_A_VENDOR_ID_DECLARE FAILED");
}
```

The `VENDEF_ID` assignment would not be needed in a real situation in which you had a function that returned your vendor-defined hostid. Close `vendor_hostid.c`.

4. Open `machind\lsvendor.c` in a text editor. At the beginning of the vendor initialization routine section, add a line defining x_flexlm_newid() and modify the initial value of ls_user_init1() from 0 to x_flexlm_newid.

```
/* Vendor initialization routines */

void x_flexlm_newid();
void (*ls_user_init1)() = x_flexlm_newid;
```

5. Open `machind\lmcrypt.c` in a text editor. After the lc_init() call, add the following line:

```
x_flexlm_newid();
```

   That section of the code should resemble:

```
if (lc_init((LM_HANDLE *)0, VENDOR_NAME, &site_code, &lm_job))
{
      lc_perror(lm_job, "lc_init failed");
      exit(-1);
}

x_flexlm_newid();
```

6. Open `machind\makekey.c` in a text editor. After the lc_init() call, add the following line:

```
x_flexlm_newid();
```

   That section of the code should resemble:

```
if (lc_init((LM_HANDLE *)0,
      VENDOR_NAME, &site_code, (LM_HANDLE **) &lm_job) )
{
      lc_perror(lm_job, "lc_init failed");
      exit(1);
}

x_flexlm_newid();
```

7. Open your client application source file in a text editor. In this example, we are using `machind\lmflex.c`.
   - Make the `lm_job` variable global by moving it before main().

     ```
     VENDORCODE code;
     LM_HANDLE *lm_job;

     void
     main()
     ```
   - After the lc_new_job() call, add the following line:

     ```
     x_flexlm_newid();
     ```

     That section should resemble:

     ```
        if (lc_new_job(0, lc_new_job_arg2, &code, &lm_job))
        {
            lc_perror(lm_job, "lc_new_job failed");
            exit(lc_get_errno(lm_job));
        }
     x_flexlm_newid();
     ```
8. Open *`platform`*`\makefile` in a text editor. This example uses a Windows makefile.
   - Add your client application to the list of EXECS. For this example, add `lmflex.exe`.
   - After the `$(DAEMON)` section, add a section to build `vendor_hostid.obj`. For example:

     ```
     vendor_hostid.obj :   $(SRCDIR)/vendor_hostid.c
             $(CC) $(CFLAGS) -I../h $(SRCDIR)\vendor_hostid.c
     ```
   - Add `vendor_hostid.obj` to the link line for `$(DAEMON)`, `makekey`, `lmcrypt`, and `lmflex`. For example, for `lmflex.exe`:

     ```
     lmflex.exe:   $(SRCDIR)/lmflex.c  $(LMNEW_OBJ) \
                               $(CLIENTLIB) lmstrip.exe
           $(CC) $(CFLAGS) $(SRCDIR)/lmflex.c
           $(LD) /out:lmflex.exe lmflex.obj vendor_hostid.obj \
                               $(LMNEW_OBJ) $(CLIENTLIB) $(XTRALIB)
           if exist lmflex.obj del lmflex.obj
     ```
9. Rebuild your duplicate FLEX*lm* kit.

Test the Vendor-Defined Hostid

You will use the vendor daemon, license generator, and client application you just built to test a vendor-defined hostid.

1. Create a license file that contains a VENDOR line with the vendor daemon you just built. Change the hostid on the SERVER line to:

   `VDH=12345678`

2. Run this license file through the newly built `lmcrypt`.
3. Start your license server pointing to this license file.
4. Run `lmflex`. You should be able to check out “f1.”
5. Exit `lmflex` and stop the license server.

Additional Steps for Production Use of a Vendor-Defined Hostid Type

To implement a real vendor-defined hostid type, you must write a function that can find the hostid that you want to use, then use that function’s return value instead of the fixed value `VENDEF_ID` in strncpy() in `vendor_hostid.c`:

```
if (idtype == VENDEF_ID_TYPE)
{
        h->type = VENDEF_ID_TYPE;

        strncpy(h->id.vendor, VENDEF_ID, MAX_HOSTID_LEN);
        h->id.vendor[MAX_HOSTID_LEN] = 0;
        return(h);
}
```