12 Environment Variables Module (env.hhf)

The env module contains a couple of functions that fetch environment strings from the operating system. These functions can be used to test global values set by the user in the environment space inherited by your processes.

12.1 The Env Module

To use the environment functions in your application, you will need to include one of the following statements at the beginning of your HLA application:

```
#include( "env.hhf" )
or
#include( "stdlib.hhf" )
```

12.2 Retrieving Environment Strings

The env module contains two functions for retrieving environment string data. To each of these functions you must pass the name of an environment variable (in the envVar string parameter). These functions will locate the specified environment variable in the system (if it is present) and return the associated string value.

procedure env.get(envVar:string; dest:string);

env.get copies the environment string data to the (preallocated) string variable you specify via the dest parameter. Note that the dest parameter must contain the address of a value HLA string object or this function may fail (or raise an exception). If the allocated storage isn't large enough to hold the string data, or the environment variable's string is greater than 4095 characters long, this function will raise an exception. This function returns true in EAX if it locates the environment variable in the environment space and successfully returns the environment variable's data. This function returns false in EAX if it cannot locate the environment variable in the environment space.

```
HLA high-level calling sequence examples:

  env.get( envVarName, envData );
  if( eax ) then

     // process the environment data in envData

  endif;

HLA low-level calling sequence examples:

  push( envVarName );// Assumption: envVarName is a string variable
  call env.get;
```

procedure env.a_get(envVar:string);

env.a_get also returns the value of an environment variable; however, it allocates storage for the result and returns a pointer to the string result (on the heap) in the EAX register, if such an environment variable exists. This function returns NULL in EAX if the environment variable does not exist. This function raises an exception if the environment variable's data is greater than 4095 characters long. The caller is responsible for freeing the storage allocated on the heap by this function.

```
HLA high-level calling sequence examples:

env.a_get( envVarName );
mov( eax, envData );
if( eax <> NULL ) then

  // process the environment data pointed at by envData

endif;
str.free( envData );

HLA low-level calling sequence examples:

  push( envVarName );// Assumption: envVarName is a string variable
  call env.get;
  mov( eax, envData );
```