Symbian Hardware Abstraction Layer (HAL) APIs

The Symbian C++ Hardware Abstraction Layer APIs can be used to query system capabilities, share state data with other applications and register for notification of changes to system states.

These APIs are implemented in two main components:

- The Hardware Abstraction Layer (HAL)
- The System Agent Components

HAL provides a very simple to use APIs, which can query the attributes of the device, for example:

- Manufacturer: manufacturer of the device, hardware and software version number.
- Processor: processor architecture, ABI and processor speed.
- Memory: total RAM, free RAM and total ROM.
- Screen: screen dimension and display color.
- Keyboard information.

The class `HAL` (`hal.h`) derives from `HALData` (`hal_data.h`), which contains the definition of the enumeration `TAttribute`.

Header Required:

```
#include <hal.h>
```

Library needed:

```
LIBRARY hal.lib
```

The API itself is very small, with only three functions, all provided by the class `HAL`:

```
static TInt Get(TAttribute anAttribute, TInt& aValue);
static TInt Set(TAttribute anAttribute, TInt& aValue);
static TInt GetAll(TInt& aNumEntries, SEntry*& aData);
```

Some examples for getting the device attributes

1. Getting all the device attributes

```
TInt numHalEntries;

HAL::SEntry* halEntries;

HAL::GetAll(numHalEntries, halEntries);
```

2. Getting one specific attribute, in this case total RAM in bytes.

```
TInt ram;

HAL::Get(HAL::EMemoryRAM, ram);
```

Some examples for setting the device attributes

```
TInt err = HAL::Set(HAL::ECaseSwitchDisplayOn, 1);
```

If a field is not settable, then the return value from the call will be one of the standard system error codes, typically `KErrNotSupported`.

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.