JAVA.LANG.RUNTIME.HALT*INTSTATUS* METHOD

Description

The **java.lang.Runtime.halt*intstatus*** method forcibly terminates the currently running Java virtual machine. This method never returns normally. This method should be used with extreme caution. Unlike the exit method, this method does not cause shutdown hooks to be started and does not run uninvoked finalizers if finalization-on-exit has been enabled. If the shutdown sequence has already been initiated then this method does not wait for any running shutdown hooks or finalizers to finish their work.

Declaration

Following is the declaration for **java.lang.Runtime.halt** method

```
public void halt(int status)
```

Parameters

- **status** -- Termination status. By convention, a nonzero status code indicates abnormal termination. If the exit *equivalently*, *System. exit* method has already been invoked then this status code will override the status code passed to that method.

Return Value

This method does not return a value.

Exception

- **SecurityException** -- If a security manager is present and its checkExit method does not permit an exit with the specified status

Example

The following example shows the usage of lang.Runtime.halt method.

```
package com.tutorialspoint;

public class RuntimeDemo {

   public static void main(String[] args) {

   // print when the program starts
   System.out.println("Program starting...");

   // halt this process
   Runtime.getRuntime().halt(0);

   // print a string, just to see if it process is halted
   System.out.println("Process is still running.");
   }
}
```

Let us compile and run the above program, this will produce the following result:

```
Program starting...
```