



# Guardian API Programming for HPE NonStop Systems U4152S

<b>HPE course number</b>	U4152S
<b>Course length</b>	5 days
<b>Delivery mode</b>	ILT
<b>View schedule, local pricing, and register</b>	<a href="#">View now</a>
<b>View related courses</b>	<a href="#">View now</a>

## Why HPE Education Services?

- IDC MarketScape leader 4 years running for IT education and training\*
- Recognized by IDC for leading with global coverage, unmatched technical expertise, and targeted education consulting services\*
- Key partnerships with industry leaders OpenStack®, VMware®, Linux®, Microsoft®, ITIL, PMI, CSA, and (ISC)²
- Complete continuum of training delivery options—self-paced eLearning, custom education consulting, traditional classroom, video on-demand instruction, live virtual instructor-led with hands-on lab, dedicated onsite training
- Simplified purchase option with HPE Training Credits

This course is for C, TAL, and pTAL developers. Learn how to use the Guardian Application Program Interface (API) to obtain operating system services including interprocess communication, process creation, Enscribe file access, and more. This course is 60 percent lecture and 40 percent hands-on labs using HPE NonStop servers.

## Audience

- Systems programmers
- Application programmers
- Programmer analysts

## Prerequisites

- Concepts and Facilities for NonStop Systems (U4147S) and
- TAL/pTAL Programming (U4198S) or C Programming in the Guardian Environment (U4143S)

## Course objectives

At the conclusion of this course, you should be able to:

- Code for interprocess communication
- Develop applications that access Enscribe files
- Handle terminal input and output
- Identify different types of system messages

- Create and manage new processes
- Available timing facilities
- Know how nowait I/O can be used to improve application performance

## Benefits to you

- Learn the functionality of Guardian procedures and Guardian API

## Detailed course outline

---

### Module 0: Course Overview

---

#### Module 1: Introduction to the Guardian API

- Major functions of the NonStop operating system
  - Components of an executing process
  - The required function prototype or declaration file for any Guardian procedure
  - The NonStop operating system's message-based architecture
  - Lab: determine the appropriate procedure to use for a given function, identify the parameter types used by specific procedures and if the procedures return a value or set the condition code
- 

#### Module 2: File-System Concepts

- Concept of a file, purpose of an I/O process, and different file name types
  - Coding the procedure calls that are common to all file accesses
  - Using the DEFINE and ASSIGN functions to equate physical and logical file names
  - Coding the procedure calls to perform file-name conversion
  - Lab: coding Guardian procedures using the DEFINE capability and the Inspect debugger
- 

#### Module 3: Interprocess Communication

- Requester-server model
  - Coding the procedures for interprocess communication
  - Differences between system and application messages
  - TACL startup protocol
  - Lab: use \$RECEIVE for message handling and the INITIALIZER or get\_startup\_msg to handle the TACL startup protocol messages
- 

#### Module 4: Enscribe Database Access

- Enscribe file types and coding the procedures to create Enscribe files
  - Coding the procedures to access and manage access of any Enscribe file
  - Appropriate error checking
  - Lab: access an Enscribe key-sequenced data file
  - Lab: code programs to perform Enscribe file accessing including adding, updating, and deleting records
- 

#### Module 5: Terminal Input and Output

- Determining the home terminal of a process
  - Procedures used to write to and read from a terminal in conversation mode
  - Errors that can occur during I/O
  - Numeric conversions from ASCII to binary and vice versa
  - Using the FIXSTRING procedure to support command alteration
  - Lab: code a program to determine the process home terminal, perform I/O operations from the home terminal, and send application messages to server processes
- 

#### Module 6: Timing Facilities

- Coding the procedures to determine the date and time
  - Coding the procedures to schedule events for future execution
  - Coding the procedures to measure time intervals
- 

#### Module 7: Nowait I/O Processing

- Coding the procedures for nowait I/O operations
  - One function of the SETMODE procedure
  - Internal and external request management
-

## Course data sheet

---

### Module 8: Process Management

- Calling the procedures to start and manage a process
- Calling the PROCESS\_GETINFOLIST\_ procedure to obtain information about processes
- HIGHPIN and HIGHREQUESTERS object file flags
- Lab: code a procedure to execute a new process
- Lab (optional): modify a given program to perform nowait I/O on selected files

---

### Module 9: TS/MP Programming

- Major components of the HPE NonStop TS/MP environment
- Pathsend environment and capabilities of a Pathsend requester
- Code calls to the Pathsend procedures
- Code nowait Pathsend operations
- Pathsend error handling
- Design considerations for creating applications using Pathsend
- Design considerations of HPE NonStop TS/MP servers

---

### Module 10: Data Integrity and Application Availability

- Guardian procedure calls to invoke the HPE NonStop TMF subsystem
  - Using HPE NonStop TS/MP subsystem to provide for continuous process availability
  - Lab (optional): using Pathsend
- 

Learn more at  
[hpe.com/ww/learnnonstop](http://hpe.com/ww/learnnonstop)

#### Follow us:



---

© Copyright 2015–2016 Hewlett Packard Enterprise Development LP. The information contained herein is subject to change without notice. The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein.

Microsoft is either a registered trademark or trademark of Microsoft Corporation in the United States and/or other countries. The OpenStack Word Mark is either a registered trademark/service mark or trademark/service mark of the OpenStack Foundation, in the United States and other countries and is used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation or the OpenStack community. Pivotal and Cloud Foundry are trademarks and/or registered trademarks of Pivotal Software, Inc. in the United States and/or other countries. Linux is the registered trademark of Linus Torvalds in the U.S. and other countries. VMware is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions. All other third-party trademark(s) is/are property of their respective owner(s).

c04593598, November 2016, Rev. 2