



Australian Government

Department of Education, Employment and Workplace Relations

ICAPRG503A Debug and monitor applications

Release: 1

ICAPRG503A Debug and monitor applications

Modification History

Release	Comments
Release 1	This Unit first released with <i>ICA11 Information and Communications Technology Training Package version 1.0</i>

Unit Descriptor

This unit describes the performance outcomes, skills and knowledge required to debug and monitor a software application.

Application of the Unit

This unit addresses the knowledge and processes necessary for debugging and monitoring software applications.

This unit is relevant to those who work as developers, testers and support engineers who use logging and tracing techniques to identify software problems and monitor systems.

Licensing/Regulatory Information

No licensing, legislative, regulatory or certification requirements apply to this unit at the time of endorsement but users should confirm requirements with the relevant federal, state or territory authority.

Pre-Requisites

Not applicable.

Employability Skills Information

This unit contains employability skills.

Elements and Performance Criteria Pre-Content

Element	Performance Criteria
<i>Elements describe the essential outcomes of a unit of competency.</i>	<i>Performance criteria describe the performance needed to demonstrate achievement of the element. Where bold italicised text is used, further information is detailed in the required skills and knowledge section and the range statement. Assessment of performance is to be consistent with the evidence guide.</i>

Elements and Performance Criteria

1. Implement a framework for logging and error handling	1.1 Determine the logging framework for writing text messages at a certain level or priority to log files or sending data to monitoring applications 1.2 Create a custom event log for an application 1.3 Analyse the logs to check the state of the running application
2. Debug and trace an application	2.1 Apply basic debugging techniques such as breakpoints and stepping through and over code and stack trace 2.2 Identify and use tools to debug software applications 2.3 Write code for debugging (print, assert and stop statements)
3. Monitor the application performance	3.1 Identify and use profiling tools to verify the parts of the system that consume the most resources, such as random access memory (RAM), central processing unit (CPU) and time 3.2 Analyse performance issues and apply changes to improve the performance of the application

Required Skills and Knowledge

This section describes the skills and knowledge required for this unit.

Required skills

- analytical skills to:
 - evaluate and compare results from different profiling tools
 - evaluate log file information and develop solutions as a result of analysis
- communication skills to recommend an approach to debugging and performance issues
- initiative and enterprise skills to identify potential improvements to the application performance
- literacy skills to:
 - read log files
 - write log messages
- numeracy skills to read and interpret the monitoring applications data
- research skills to keep up-to-date with monitoring and debugging technologies and tools
- technical skills to:
 - use software development tools
 - write code to create custom event log.

Required knowledge

- basic knowledge of:
 - computer hardware, networking and components
 - database-management systems
 - object-oriented programming
 - open-source development tools
- maths at a basic level
- procedures for developing small-size applications
- software development life cycle (SDLC).

Evidence Guide

The evidence guide provides advice on assessment and must be read in conjunction with the performance criteria, required skills and knowledge, range statement and the Assessment Guidelines for the Training Package.

Overview of assessment	
Critical aspects for assessment and evidence required to demonstrate competency in this unit	Evidence of the ability to: • create error logging framework to persist error logs, debug and trace an application, and plan a performance-monitoring strategy • use debugging techniques and tools, and profiling tools and analyse performance issues.
Context of and specific resources for assessment	Assessment must ensure access to: • client and server system • sufficient privileges to access and use performance monitoring tools in client and server systems • debugging tools • specific tools and licences, depending on particular platform • appropriate learning and assessment support when required • modified equipment for people with special needs.
Method of assessment	A range of assessment methods should be used to assess practical skills and knowledge. The following examples are appropriate for this unit: • written or verbal questioning of debugging, diagnostics and monitoring strategy • evaluation of candidate's logging framework and custom event log for an application • direct observation of candidate using an integrated development environment (IDE) for code development and debugging and the monitoring applications.
Guidance information for assessment	Holistic assessment with other units relevant to the industry sector, workplace and job role is recommended, where appropriate. Assessment processes and techniques must be culturally appropriate, and suitable to the communication skill level, language, literacy and numeracy capacity of the candidate and the work being performed. Indigenous people and other people from a non-English speaking background may need additional support. In cases where practical assessment is used it should be

	combined with targeted questioning to assess required knowledge.
--	--

Range Statement

The range statement relates to the unit of competency as a whole. It allows for different work environments and situations that may affect performance. Bold italicised wording, if used in the performance criteria, is detailed below. Essential operating conditions that may be present with training and assessment (depending on the work situation, needs of the candidate, accessibility of the item, and local industry and regional contexts) may also be included.

Logging framework may include:	• Java logging framework: • Apache Commons Logging • Java Logging application programming interface (API) • Log4J • SLF4J • .NET logging and tracing tools: • C# Logger • DebugWriter • Log4net • LogThis • Logview4net • Microsoft's Enterprise Library • NetTrace • TrafficMonitor.
Level may include:	• debug • error • fatal • information • trace • warning.
Tools may include:	• Eclipse debugger • Firebug (JavaScript debugger) • Java debugger • NetBeans debugger • Visual Studio debugger • WinDBG, NTSD, and CDB to debug .NET applications.
Profiling tools may include:	• CLR profiler • Microsoft operations manager • open-source profilers in Java: • Cougaar Memory Profiler • Extensible Java Profiler • JMP

	• NetBeans Profiler • TomcatProbe • structured query language (SQL) profiler • system monitor • Visual Studio profiler.
--	---

Unit Sector(s)

Programming and software development