



**Australian Government**

**Department of Education, Employment and Workplace Relations**

# **ICANWK409A Create scripts for networking**

**Release: 1**

## ICANWK409A Create scripts for networking

### Modification History

Release	Comments
Release 1	This Unit first released with <i>ICAI1 Information and Communications Technology Training Package version 1.0</i>

### Unit Descriptor

This unit describes the performance outcomes, skills and knowledge required to undertake scripted programming tasks for networking-related activities.

### Application of the Unit

This unit applies to those employed in network or systems administration roles.

### 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

<b>Element</b>	<b>Performance Criteria</b>
<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

<p>1. Develop algorithms to represent solutions to a given problem</p>	<p>1.1 Consult with <i>client</i> and key <i>stakeholders</i> to identify the problem and associated script requirements</p> <p>1.2 Employ abbreviated software development cycle to script creation</p> <p>1.3 Develop an algorithm to solve the problem and meet client requirements</p> <p>1.4 Develop an algorithm which takes account of expected possible situations</p> <p>1.5 Develop an algorithm which is guaranteed to end</p> <p>1.6 Demonstrate use of structure, sequence, selection and iteration</p>
<p>2. Create code</p>	<p>2.1 Select appropriate <i>scripting language</i></p> <p>2.2 Demonstrate understanding and application of basic language syntax rules and best practices</p> <p>2.3 Select and use language data types, operators and expressions to create clear and concise code</p> <p>2.4 Use techniques of selection, iteration and sequence to control script execution flow</p> <p>2.5 Use techniques for sequential file input and output to retrieve and store information</p> <p>2.6 Obtain and use user input to affect the operation of the script</p> <p>2.7 Apply internal document principles to created code</p> <p>2.8 Follow <i>organisational guidelines</i> for developing maintainable code when creating scripts</p> <p>2.9 Adhere to <i>coding standards</i> when creating scripts</p>
<p>3. Use operating system tools</p>	<p>3.1 Use searching and sorting tools to select information from the logging output of <i>operating system</i> (OS)</p> <p>3.2 Implement controls to ensure that where significant events occur, script creates and maintains a log of operations via operating system logging mechanism</p> <p>3.3 Register and run scripts with OS scheduling facility</p>
<p>4. Test and debug code</p>	<p>4.1 Engineer, document and conduct simple tests to confirm code meets design specification</p> <p>4.2 Identify areas that are not covered or are covered incorrectly in the script</p>

	<p>4.3 Take action to ensure that code complies with security policy</p> <p>4.4 Take action to ensure that code operates with proper permissions</p> <p>4.5 Use script debugging techniques suitable for use with scripting language to detect and resolve errors of syntactical, logical and design origin</p>
5. Document script	<p>5.1 Create technical-level documentation</p> <p>5.2 Create user-level documentation</p>

## Required Skills and Knowledge

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

### Required skills

- communication skills to liaise with external and external personnel to discuss problems and requirements
- literacy skills to:
  - produce and evaluate technical documents
  - produce user and peer documentation
- problem-solving and contingency-management skills to:
  - debug syntax and semantic errors in the program during testing
  - develop algorithmic solutions to a given problem
- technical skills to:
  - create scripts to automate operating system tasks
  - execute scripts
  - interact with user via script
  - manipulate and extract information contained in files
  - use inbuilt scripting options for a variety of scenarios.

### Required knowledge

- overview knowledge:
  - algorithm design
  - operating system components, such as command line interface, log files, program scheduling utilities, development methodologies, tools and utilities and testing methods
- debugging for a variety of scripting scenarios
- programming structured control constructs: sequence, selection, iteration
- scripting techniques and language syntax.

## 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.*

<b>Overview of assessment</b>	
<b>Critical aspects for assessment and evidence required to demonstrate competency in this unit</b>	<p>Evidence of the ability to:</p> <ul style="list-style-type: none"> <li>• develop an algorithmic statement of a solution for a set process</li> <li>• design, document, construct and test a small single-purpose OS utility application in response to a problem description</li> <li>• create scripted programs that access information stored in files on the system and use system utility programs to sort or find information within these files</li> <li>• validate and record script results.</li> </ul>
<b>Context of and specific resources for assessment</b>	<p>Assessment must ensure access to:</p> <ul style="list-style-type: none"> <li>• technical requirements</li> <li>• software development environment</li> <li>• software testing environment</li> <li>• appropriate learning and assessment support when required.</li> </ul> <p>Where applicable, physical resources should include equipment modified for people with special needs.</p>
<b>Method of assessment</b>	<p>A range of assessment methods should be used to assess practical skills and knowledge. The following examples are appropriate for this unit:</p> <ul style="list-style-type: none"> <li>• verbal or written questioning of required skills and knowledge, such as: algorithm design, structured control constructs, basic scripting development methodologies and their application</li> <li>• evaluation of algorithm design</li> <li>• evaluation of a small single-purpose OS utility application</li> <li>• evaluation of completed test plan and documentation.</li> </ul>
<b>Guidance information for assessment</b>	<p>Holistic assessment with other units relevant to the industry sector, workplace and job role is recommended, where appropriate.</p> <p>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.</p> <p>Indigenous people and other people from a non-English speaking</p>

	<p>background may need additional support.</p> <p>In cases where practical assessment is used it should be combined with targeted questioning to assess required knowledge.</p>
--	---



## 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.*

<b><i>Client</i></b> may include:	<ul style="list-style-type: none"> <li>• external organisations</li> <li>• information and communications technology (ICT) company</li> <li>• individuals</li> <li>• internal departments</li> <li>• internal employees</li> <li>• service industry.</li> </ul>
<b><i>Stakeholders</i></b> may include:	<ul style="list-style-type: none"> <li>• development team</li> <li>• information technology (IT) manager or representative</li> <li>• network engineer</li> <li>• project team</li> <li>• systems administrator</li> <li>• user.</li> </ul>
<b><i>Algorithm</i></b> may include:	<ul style="list-style-type: none"> <li>• flow chart</li> <li>• pseudocode</li> <li>• structured English.</li> </ul>
<b><i>Scripting language</i></b> may include:	<ul style="list-style-type: none"> <li>• C, C++</li> <li>• JavaScript</li> <li>• Linux shell scripts</li> <li>• Perl</li> <li>• Python</li> <li>• VBScript</li> <li>• VB.Net</li> <li>• Windows PowerShell.</li> </ul>
<b><i>Organisational guidelines</i></b> may include:	<ul style="list-style-type: none"> <li>• communication methods</li> <li>• content of emails</li> <li>• dispute resolution</li> <li>• document procedures and templates</li> <li>• downloading information and accessing particular websites</li> <li>• financial control mechanisms</li> <li>• opening mail with attachments</li> <li>• personal use of emails and internet access</li> <li>• virus risk.</li> </ul>
<b><i>Coding standards</i></b> may	<ul style="list-style-type: none"> <li>• GNU coding standard</li> </ul>

include:	<ul style="list-style-type: none"><li>• Java coding standard</li><li>• organisation standards.</li></ul>
<i>Operating system</i> may include:	<ul style="list-style-type: none"><li>• Linux</li><li>• Mac</li><li>• Novell</li><li>• Windows.</li></ul>

## Unit Sector(s)

Networking