invensys® rail group
WESTINGHOUSE®
rail systems australia

invensys® rail group
WESTINGHOUSE®
rail systems india

WESTRACE
Graphical Configuration Sub-System (GCSS)

User Manual

We deliver. We go further.

Westinghouse Rail Systems Australia
ABN 78 000 102 483
179–185 Normanby Rd
(Locked Bag 66)
South Melbourne
Victoria 3205
Australia
P +61 3 9233 8888
F +61 3 9233 8702
E wrsa@wrsa.com.au
W http://www.wrsa.com.au

Invensys Rail Systems India Pvt Ltd
No. 112 & 114 Raheja Chambers
12 Museum Road
Bangalore 560 001
India
P +91 80 3058 8763/64
F +91 80 3058 8765
E rail.enquiries@invensysrail.com
W http://www.wrsi.in

WESTRACE
Graphical Configuration Sub-System (GCSS)

User Manual

Document CI: **WRTOGCSS**

Issue: **9.0**

Date of Issue: **11 February 2009**

Change History:

Issue	Date	Comment	Changed	Checked	Approved
1.0	2/11/95	First Release			
2.0	10/9/96	Revised for CR292, 295, 299, 308	DC		
3.0	24/9/96	Revised for CR327	DC		
4.0	16/10/02	Revised to Invensys style, and for CR411, 428, 509, 513, 515, 542 Updated for GCSS 6.0 (Release 5)	PGB		
5.0	29/04/04	Revised for CR815, 829	PGB	EV	
6.0	29/06/05	Bug 2962, 3015, 3030, 3048, 3061, 3073, 3074, 3084, 3090, 3114, 3115, 3127, 3154 CR328, 372, 378 Revised for GCSS 7.0.0	ML	LB, GW, HW	
7.0	7/2/06	CR344 (amended)	ML	SR	WMcD
8.0	28/5/08	Revised for GCSS 7.1.x Added BeeProg information Corrections and improvements	ML	DC	WMcD
9.0	11/2/09	Updated branding	ML	ML	WMcD

Controlled Copy Number:

Controlled Copies are identified by a Controlled Copy Number that must be inserted in red handwriting. Any copy not meeting this criterion is considered to be uncontrolled and may not be the current version.

Copyright
This document is protected by Copyright and all information contained therein is confidential. The contents of this document must not be copied and the information therein must not be used, or disclosed except with the written permission of and in the manner permitted by Westinghouse Rail Systems Australia Pty Ltd.

Copyright © 2009 Westinghouse Rail Systems Australia Pty Ltd ABN 78 000 102 483

PREFACE

This is the user manual for the WESTRACE Graphical Configuration Sub-System (GCSS) version 7.1.x.

The GCSS may be applied to:

- WESTRACE Vital Signalling Systems;
- WESTECT Encoders;
- combined WESTRACE interlocking WESTECT Encoders.

SAFETY

WARNING

WESTRACE is a safety system. Although the GCSS is not a safety system, it is used to design WESTRACE.

To achieve and maintain the intended level of safety, WESTRACE systems must be designed, maintained and operated in accordance with the WESTRACE Application Manual (WRTOAPPM) and the WESTRACE First-Line Maintenance Manual (WRTOFLMM).

Table of Contents

6 Ladder Logic

7 Consistency, Compiling, Checking and Approval of Installations

8 Data Download, Upload and Compare

Appendix A: GCSS Interface

LIST OF TABLES

List of Figures

1. INTRODUCTION

The Graphical Configuration Sub-System (GCSS) is a component of the WESTRACE system.

It is used to:

- specify the modules for a WESTRACE application;
- specify the configuration logic for a WESTRACE application;
- program vital logic module PROMs with application data, and;
- download data to non-vital logic modules.

Note: ***This issue of the WESTRACE Graphical Configuration Sub-System (GCSS) User Manual (WRTOGCSS) applies to GCSS version 7.1.x.***

1.1 Changes in GCSS Version 7.1.x

Version 7.1.x of GCSS contains the following changes:

Enhancements

- Consistency checks added for NCDM installation to range-check network configuration data entered as comment data, and to ensure interface file is present when module is configured to operate with a VLM6.
- Vital approval reports enhanced to provide details of all associated interface files.
- Non-vital installation reports enhanced to provide S2 switch bank settings.
- Archiving of installations with associated interface files reworked to provide more robust operation.
- Installation check reports available for reference installation OR uploaded installation.
- Difference reporting of uploaded non-vital configurations aligned with objects actually decompiled.
- Guards added to prevent unapproved non-vital installations being downloaded via ICS.
- Default states for non-assigned outputs enforced.
- Corrected VLM6 compiler for case when no DPRAM modules are configured.
- Dependency on PC timezone for installation approval and checking reworked to increase portability.
- PROM programming enhanced by the addition of Intel Hex file download and upload capability.

Removals

- NCDM configuration of non-vital WESTRACE sessions removed.
- Contiguous configuration restriction of protocols on the network port of the NCDM removed.
- Changes to mnemonics for NCDM installations extracted from archives removed.

1.2 Purpose of This Manual

The purpose of this manual is to:

- provide an overview of basic WESTRACE application data concepts;
- introduce the operating concepts of the GCSS;
- describe how to use the GCSS for:
  - defining WESTRACE system components, application logic and communications settings;
  - programming PROMs and downloading data to non-vital modules;
- provide:
  - installation and setup instructions;
  - assistance in getting started;
  - guidance in version control;
  - assistance with error messages;
  - supporting information for GCSS operators and system supervisors.

1.3 Scope of This Manual

This manual is for Signal Engineers who need to design railway signal logic for WESTRACE Vital Signalling Equipment. It describes how to use the WESTRACE GCSS. It does not describe detailed design of a WESTRACE application—for this information see reference [APPM]. However, it does provide some system design information to assist in the development of WESTRACE applications when using the GCSS

This manual assumes existing knowledge of railway signalling principles applicable to the target railway.

This manual does not cover the use of the GCS Templates Tool but is applicable to using logic ladders developed using the GCS Templates Tool.

1.4 Safety

The GCSS supports both vital (HVLM128, HVLM128a, VLM5 or VLM6) and non-vital (NCDM or NVC/DM) installations.

SAFETY
WARNING

The Signal Engineer using the GCSS must ensure the:

- vital installation includes all vital functionality;
- non-vital installation does not include any vital functionality.

The Signal Engineer checking installation data prior to approval must explicitly check that no vital functionality is included in a non-vital installation.

See also WESTRACE GCS Templates Tool User Manual [GTT].

1.5 References

This manual refers to the following WESTRACE manuals. All are available from Westinghouse Rail Systems Australia.

[SOM] WESTRACE System Overview Manual, WRTOOVER: describes the WESTRACE system, what it is used for and what it consists of.

[APPM] WESTRACE Application Manual, WRTOAPPM: describes the application design of a WESTRACE signalling system. It is the “how to” document that must be followed by system designers and installers. The GCSS requires issue 4.0 or later.

[FLM] WESTRACE First-Line Maintenance Manual, WRTFLMM: describes maintenance procedures to quickly and accurately diagnose system problems so that correct operation is restored (including WESTECT trackside equipment).

[ICS] WESTRACE Installation Check System User Manual, WRTO_ICS: describes the use and application of the ICS to check that the application logic is installed correctly in an HVLM (and later) based WESTRACE system.

[GTT] WESTRACE GCS Templates Tool User Manual, WRTO_GTT: describes the use and application of the GTT to create installation templates for later use with the GCSS.

The GCSS Release Notes that came with the software should also be read and understood. These notes contain:

- the latest changes not reflected in this manual;
- any known problems with matching recommended work- around;
- any other information that may affect successful operation of the GCSS.

1.6 Relationship with Other WESTRACE Manuals

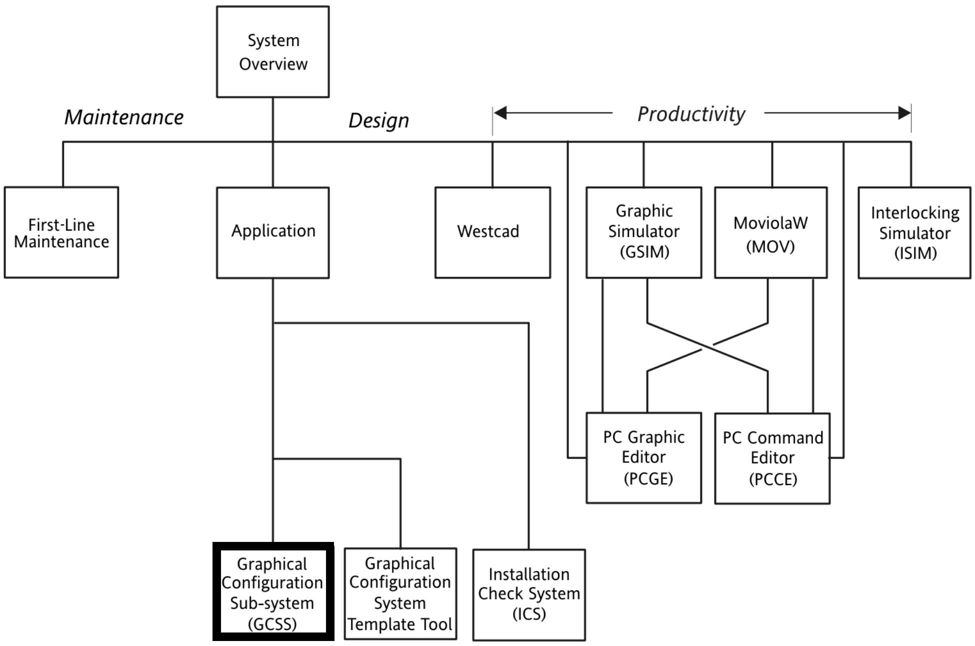


Figure 1.1 WESTRACE and related manuals

1.7 Organisation of This Manual

Chapter 1	*Introduction*—introduces and describes this user manual. Provides an overview of the GCSS design cycle.
Chapter 2	*WESTRACE and the GCSS*—discusses the design of Installations and the application of the GCSS.
Chapter 3	*Starting*—discusses starting the GCSS and basic operating procedures.
Chapter 4	*Housings and Modules*—describes how to use the housing editor to place modules in a VLM installation. See section 1.8.1 for definitions of “VLM” and “installation”.
Chapter 5	*Modules and Interface Files*—describes how to use the module and port editors. Describes selecting or creating and editing an Interface file. See section 1.8.1 for definitions of interface files.
Chapter 6	*Ladder Logic*—describes how to use the ladder logic editors and the Rung Viewer.
Chapter 7	*Consistency, Compiling, Checking and Approval of Installations*—describes the procedures for checking an installation and interface files for consistency, compiling the installation, manually checking interface files and the installation for compliance with railway requirements and then approving interface files and the installation for production of PROMs or downloading to a non-vital logic module.
Chapter 8	*Data Download, Upload and Compare*—describes the procedures for programming PROMs or downloading Non-Vital Configuration data.
Chapter 9	*Find and Replace Mnemonics*—the general procedure for finding and replacing mnemonics within the editors.
Chapter 10	*GCSS Installation and Setup*—explains how to install and set up the GCSS software.
Appendix A	*GCSS Interface*—provides additional information about the GCSS user interface. Space is provided for you to add your own notes on GCSS data entry fields.
Appendix B	*Reports*—provides an overview and samples of all GCSS reports.
Appendix C	*Compiler Error Messages*—lists the Compiler error messages with suggested remedies.
Appendix D	*Decompiler Error Messages*—lists the Decompiler error messages with suggested remedies.

1.8 Definitions and Conventions

1.8.1 Important Definitions

VLM	An acronym representing the HVLM128, HVLM128a, VLM5 or VLM6 modules.
NVLM	An acronym representing non-vital logic modules such as the NCDM or NVC/DM.
WESTRACE application	Comprises WESTRACE modules (hardware) installed in one to four housings plus application data to control the WESTRACE modules.
Application Data	All of the data for a particular WESTRACE application, comprising all module placements in WESTRACE housings, application logic and communication settings.
Vital PROM Data	The compiled data that is downloaded to PROMs for insertion in the vital logic module.
Non-Vital Configuration	The compiled data that is downloaded to the non-vital logic module.
Installation	Denotes part of a complete set of WESTRACE Application Data. Often preceded by VLM or NVLM. The complete WESTRACE Application Data can comprise: • a VLM installation alone; • a VLM installation, an NVLM installation plus Interface file(s).
VLM-NVLM Interface file	Contains information passed between a VLM installation and an NVLM installation: • Vital Logic Module type and hot-standby configuration; • NCDM or NVC/DM Module Address within the VLM installation; • Inputs and Outputs shared between the VLM and NVLM installations; • Configuration history of the interface file.
VLM6-VLM6 Interface file	Contains information passed between two VLM6 installations: • Inputs and Outputs shared between the two VLM6 installations; • Configuration history of the interface file.
Server	Also known as Slave or Field
Client	Also known as Master or Office

1.8.2 Conventions

1.8.2.1 Keystroke Instructions

Nomenclature	Example	Action
Space between characters	**Alt s f**	Press the keys consecutively
+ sign between characters	**Ctrl+Alt+p**	Press all the keys simultaneously

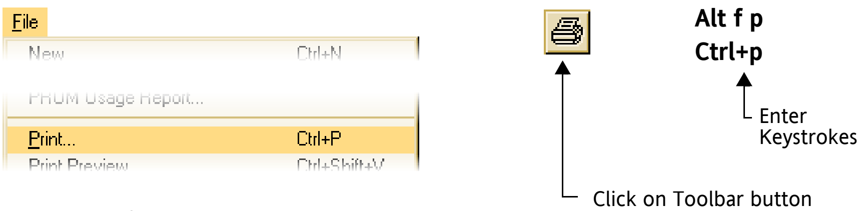
1.8.2.2 The Mouse

- **Click**—means “select”.
- **Double-click**—means “select and enter”.
- **Drag**—select the required display element, hold down the mouse button as you move the mouse.

See Appendix A, section A.1.1. for further assistance.

1.8.2.3 Commands

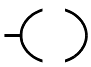
The alternative ways of implementing commands are depicted like this example.

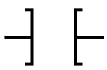


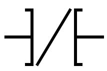
Scrap view of menu highlighting relevant menu items. Click on desired item.

1.8.2.4 Ladder Logic Symbols

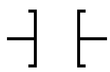
The ladder logic symbols used in this manual and by the GCSS are:

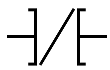

Relay Coil


Normally open contact with relay coil in the energised state. A “front” contact in relay terminology.


Normally closed contact with relay closed in the de-energised state. A “back” contact in relay terminology.

The GCSS provides alternative representations of these contacts. To change to the alternative representations, choose the appropriate option from the View menu or use the accelerator keys **Ctrl+V** and **Ctrl+]**.

 for 

 for 

1.8.2.5 Logic States

This manual uses the following terminology for logic states:

- Internal State= Logic State;
- Energised= Logic 1(also known as “True” or “Set” or “High”);
- De-energised= Logic 0(also known as “False” or “Clear” or “Low”).

1.8.2.6 Important Information

This manual highlights important information as follows:

“Action Point”—identifies a task or a requirement for performing a task.

Note: ***“Note”—highlights important information.***

Caution: ***“Caution”—highlights the possibility of damage to equipment, but not necessarily danger to personnel when handling, operating or maintaining equipment.***

SAFETY WARNING ***“Safety Warning”—highlights information relating to safety hazards. Failure to follow these warnings may lead directly or indirectly to serious equipment damage, or serious injury or death of personnel.***

1.9 Operator Prerequisites

The prerequisite knowledge for operating the GCSS package is a user’s working knowledge of Windows.

Note: ***Appropriate signal experience and authority is required for design and approval of Railway Signal Interlockings.***

1.10 How to Use This Manual—The Design Cycle

1.10.1 Summary of Tasks

This is a brief overview of the tasks performed to produce, verify and validate the design, Vital PROM Data and the Non-Vital Configuration for a WESTRACE application. Section 1.10.2 provides a more detailed overview of the design process performed by using the GCSS.

A Preparation, Specification and Design

Follow the design rules and guidelines laid down in the WESTRACE Application Manual (Reference [APPM]).

B Data Entry

a) Use the GCSS to create the data for the WESTRACE application.

The GCSS stores the application data in installation and interface "Source Files" that also record the source file checksums and the data version.

b) Check this data by:

i. Interactive use of the GCSS.

ii. The print reports produced by the GCSS.

iii. Simulation, using the simulator provided with the GCSS.

c) Produce a set of GCSS reports when the data is deemed to be correct.

These reports are the Approval Reports required for the Approval process.

C Approval Checking

The person authorised to approve the design will check the Approval Reports against the Track Plan and Control Tables. This may include further simulation.

D Approval

When the Approval Printouts are acceptable, the person authorised to approve the design will:

a) Use the GCSS to approve the "Source Files" for the WESTRACE application. The GCSS creates "Approved Vital and Non-Vital Source Files" and produces both Vital and Non-Vital Installation Approval Reports. The Approval Reports contain the version numbers and CRC values for both the "Source File" approved, and the "Approved Source File".

b) Ensure that the data version and CRC values for the "Source File" approved are the same as in the Approval Reports.

New Approval Reports MUST be prepared when the data version or CRC values are not the same. The Approval Reports MUST NOT be signed, and the Approver must ensure that the "Approved Source Files" are deleted.

c) Sign the Approval Reports as a record of the approval.

d) Store the “Approved Source Files”, Approval Reports and signed Approval Reports under a secure document control system to ensure traceability of the vital and non-vital installation data.

E Compilation, PROM Programming and Non-Vital Configuration Download

The Vital Installation data (Vital PROM Data) is programmed into PROMs. The Non-Vital Installation data (Non-Vital Configuration), if any, is downloaded to the non-vital logic module (NCDM or NVC/DM).

a) Use the GCSS to compile the “Approved Vital Source File” and program the PROMs.

   The GCSS will only allow PROMs to be produced from an “Approved Vital Source File”. Checks are performed during PROM programming.

   The person programming the PROMs must sign the PROM Download Report (figure B.6) as a record of the PROM programming.

b) Use the GCSS to compile the “Approved Non-Vital Source File” and download the “Non-Vital Configuration” to the non-vital logic module.

   The GCSS will only allow the “Non-Vital Configuration” to be produced from an “Approved Non-Vital Source File”.

   The person downloading the Non-Vital Configuration to the non-vital logic module must sign the “Non-Vital Configuration Download Report” as a record.

F Configuration Check and Manual Check of CRCs (Optional)

a) Use the GCSS to upload the installed data from the:

   - vital PROMs, via the PROM programmer;
   - directly from the Non-Vital Logic Module (NVLM).

   The GCSS decompiles the uploaded data to recreate a vital source file.

b) Use the GCSS to compare the recreated source file against the original “Approved Source File”.

   The GCSS generates a “Difference Report” which contains recalculated CRC values for both source files plus other comparative information.

c) Manually check the report to ensure both source files are identical.

G Installation Check and Manual Check of CRCs

This check procedure is performed on the actual WESTRACE hardware either in the factory or when the hardware is installed on-site.

The procedure is described in the Installation Check System (ICS) manual. See [ICS].

1.10.2 Design Process

The table below:

- lists each task in the GCSS design process (in the usual order).
- nominates the appropriate GCSS menu option(s).
- provides a reference to the relevant section(s) in this manual.

The described process assumes a new design. Some tasks may be omitted when revising an existing design.

The Check, Compile, and Approve functions are mandatory for all formal installations.

To interpret the table, consider the following example:

2.	Create or open a vital logic (VLM) installation file	New VLM installation Open existing VLM installation	3.5.1 3.5.2

The row means:

- To create a new installation file, see section 3.5.1 in this manual.
- To open an installation file, see section 3.5.2 in this manual.

Most commands can be executed using Toolbar Buttons or accelerator keys. Details are provided in the body of the manual.

Task		Comment	See Section
1.	Start GCSS.	Use one of the usual Microsoft Windows methods. Enter your name and password	3.2 3.2.2
2	Create or open a vital logic (VLM) Installation file. The GTT can be used to create these files. See section 2.1.2.4.	New VLM installation. Open existing VLM installation. (You will need to create or open and edit a matching NVLM installation (step **17**) and an associated VLM-NVLM interface file (step **8**) when the WESTRACE application uses a non-vital logic module such as NCDM or NVC/DM.)	3.5.1 3.5.2
3	Edit VLM installation header data.	Through the VLM Installation window.	3.6
4	Print initial VLM installation report.	Through the VLM Installation window.	3.6.1
5	Allocate modules to VLM installation.	Use the Housing Editor. Add Design records as required.	4.2 2.10.3
6	Print Housing Editor report.	Through the Housing Editor window.	4.2.6.1

Task		Comment	See Section

7	Assign, enter or edit as appropriate for each module in the VLM installation (including the VLM module):
	• sequence numbers;
	• fault parameters, and;
	• mnemonics for:
	  • latches;
	  • timers;
	  • inputs, and;
	  • outputs.
		Use the Module Editor.	5.2

	(You will need to create or open and edit a matching NVLM installation (step **17**) and an associated VLM-NVLM interface file (step **8**) when the WESTRACE application uses a non-vital logic module such as NCDM or NVC/DM.)	Add Design records as required.	2.10.3 5.2.3
8	Create a VLM-NVLM interface file through the NVLM module when the VLM installation includes a non-vital logic module such as an NCDM or NVC/DM.	Use the Module Editor.	5.2.8.1
	Enter mnemonics for inputs and outputs as required.		5.2.8.2
		Add Design records as required.	2.10.3
	Perform consistency check on the interface file		5.2.8.3
9	When the VLM is a VLM6 that is required to communicate with another VLM6:		
	• Correctly configure the VLM installation.	Through the VLM Installation window. Repeat for all communicating VLM installations now or later as is convenient.	5.2.7.1
	• Create or select a VLM6-VLM6 interface file through the VLM6 module of one of the communicating VLM installations.	Use the Module Editor. Select the interface file if already created through the other communicating VLM6. There must be a separate single interface file for every instance of a VLM6 communicating with another VLM6.	5.2.7.1
	• Enter or edit interface file inputs and output mnemonics as required.	Use the Module Editor. Repeat for each interface file. Add Design Records as required.	5.2.2.2 2.10.3
	• Perform a consistency check on the interface file.	Repeat for each interface file	5.2.7.2
	• Check and approve the VLM6-VLM6 interface file.	Add Checked record. Add Approved record. Repeat for each interface file.	5.2.7.2 5.2.7.2
10	Print Module reports for the VLM installation.	Print reports individually through the Module Editor, or print ALL module reports through the Housing Editor.	5.2.5 4.2.6.2
11	Print mnemonic usage report for the VLM installation.	Through the VLM Installation window. Use as a reference when adding and editing rungs in the logic ladder.	3.7
12	Add rungs to logic ladder for the VLM installation.	Use the Ladder Editor.	6.2
		Add Design records as required. Ladder Editor must be closed first.	2.10.3

Task		Comment	See Section
13	Edit the logic content of each ladder rung in VLM logic ladder.	Use the Rung Editor.	6.3
		Add Design records as required. Rung and Ladder editors must be closed first.	2.10.3
14	View and print the VLM logic ladder logic content.	Use the Rung Viewer.	6.4
15	Add Design Record when logic ladder editing is complete.	At least one Design record should be added now particularly when no records were added during logic ladder editing.	2.10.3
16	Print mnemonic usage report for the VLM installation.	Through the VLM Installation window. Use to confirm all mnemonics are correctly used in the ladder logic.	3.7
17	Create or select an NVLM installation when the VLM installation includes a non-vital logic module such as NCDM or NVC/DM.	New NVLM installation. Open existing NVLM installation. Open an installation created using GTT.	3.5.1 3.5.2 2.1.2.4
18	Edit NVLM installation header data.	Through the NVLM Installation window.	3.6
19	Print initial NVLM installation report.	Through the NVLM Installation window.	3.6.1
20	Select and assign the VLM-NVLM interface file created in step **8** to the NVLM installation.	Use the Module Editor.	5.3.9.1
	Edit the interface file inputs and outputs if required.	Add design records as required.	5.3.9.2 2.10.3
21	Make the VLM installation active	Click on it with the mouse.	
22	Display the housings. Double-click on the NVLM module to activate the Module Editor.	Use the Housing Editor.	4.2
	Perform a consistency check on the VLM-NVLM interface file.	Add checked record.	5.2.8.3 5.2.8.3
	Approve the VLM-NVLM Interface file	Add approved record.	5.2.8.3
23	Perform consistency check on the VLM installation. Repeat steps **7**, and steps **10** to **16** until the vital logic is correct and the VLM Installation passes the consistency check.	Through the VLM Installation window.	7.2
24	Compile the VLM installation.	Through the VLM Installation window.	7.3
25	Make the NVLM installation active.	Click on it with the mouse.	
26	Setup inputs and outputs for all remaining ports.	Use the Port Editor.	5.3.8
27	Close Port Editor.	Add design record as required.	2.10.3
28	Assign and edit Latch and Timer mnemonics for the NVLM.	Use the Module Editor.	5.3.3
		Add design records as required.	2.10.3
29	Print mnemonic usage report for the NVLM installation.	Through the NVLM Installation window. Use as a reference when adding and editing rungs in the logic ladder.	3.7

Task		Comment	See Section
30	Add rungs to logic ladder for the NVLM installation.	Use the Ladder Editor.	6.2
		Add design records as required. Ladder Editor must be closed first.	2.10.3
31	Edit the logic content of each ladder rung in NVLM logic ladder.	Use the Rung Editor.	6.3
		Add design records as required. Ladder and Rung editors must be closed first.	2.10.3
32	View and print the ladder logic content.	Use the Rung Viewer.	6.4
33	Add Design Record when logic ladder editing is complete.	At least one Design record should be added now particularly when no records were added during logic ladder editing.	2.10.3
34	Print mnemonic usage report for NVLM installation.	Through the NVLM Installation window. Use to confirm all mnemonics are correctly used in the ladder logic	3.7
35	Perform consistency check on the NVLM installation.	Through the NVLM Installation window.	7.2
	Repeat steps **26** to **34** until the non-vital logic is correct and the NVLM installation passes the consistency check.		
36	Compile the NVLM installation.	Through the NVLM Installation window.	7.3
37	Manual check of all aspects of the design by an independent authorised Signalling Engineer.		7.4
	Add a Check record for the VLM installation.	Through the VLM Installation window.	2.10.3
	Add a Check record for the NVLM installation (if used).	Through the NVLM Installation window.	2.10.3
38	Approval of installation(s) by an independent authorised Signalling Engineer.		7.5
	Approve the VLM installation.	Through the VLM Installation window.	2.10.3
	Approve the NVLM installation (if used).	Through the NVLM Installation window.	2.10.3
39	Make PROMs.	Through the VLM Installation window. The GCSS automatically recompiles the VLM installation.	8.1
	Download to PROMs and print PROM Download Report	Download through the VLM installation.	
	Download a second time to confirm the accuracy of the first download	Download through the VLM installation.	
40	Download Non-Vital Configuration and print Non-Vital Configuration Download Report.	Download through the NVLM Installation.	8.2
41	Optional: Upload installed Vital PROM Data and Non-Vital Configuration.	Upload through the installation window(s) and decompile.	8.3.1 8.3.2
42	Optional: Check installed PROM or Non-Vital Configuration by comparing uploaded and decompiled data with the original source data.	Through the installation window(s).	8.3.3

1.11 Backwards Compatibility

1.11.1 Installation Files

GCSS version 7.1.x is backwards-compatible for installation files (.ins, .ncd and .nvc) with GCSS versions 3.1, 5.0, 5.2.1, 6.0, 6.0.2, 6.1.0, 6.2.0, 7.0.0 and later.

When opening files created with earlier versions, GCSS 7.1.x reminds you to carry out a consistency check (see section 2.7) and sets the file's status to Modified as shown in figure 1.2.

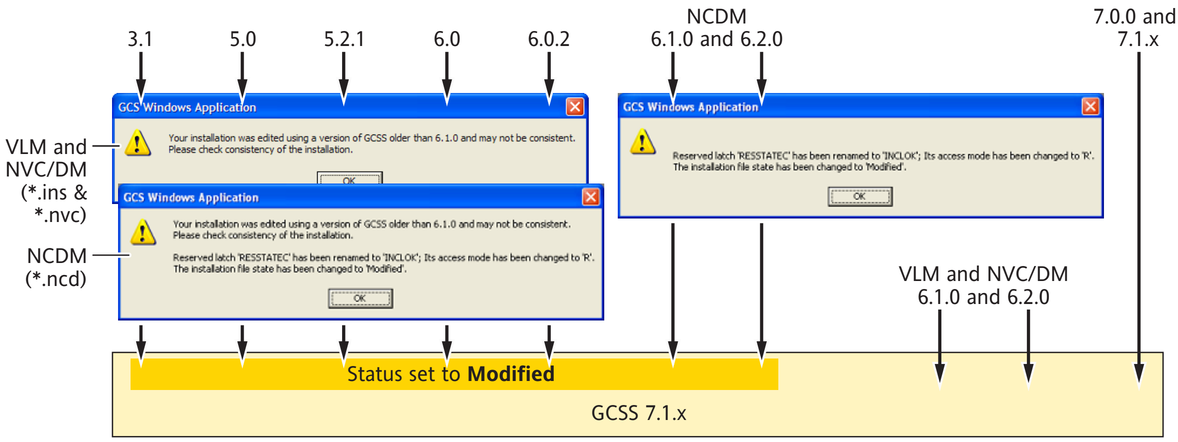


Figure 1.2 Opening installation files in GCSS 7.1.x

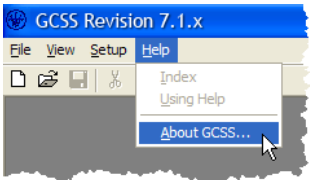
Only GCSS 7.0.0 and later, and ICS 7.0.0 and later support locked (template) rungs. See section 6.1.2.

1.11.2 Installation Image Data

GCSS version 7.1.x is backwards-compatible for installation data download, upload and compare (see Chapter 8) with installation image data created using GCSS versions 6.0.2, 6.1.0, 6.2.0, 7.0.0 and later.

1.12 About GCSS

Select **About GCSS** from the **Help** menu to open the About GCSS dialog box.



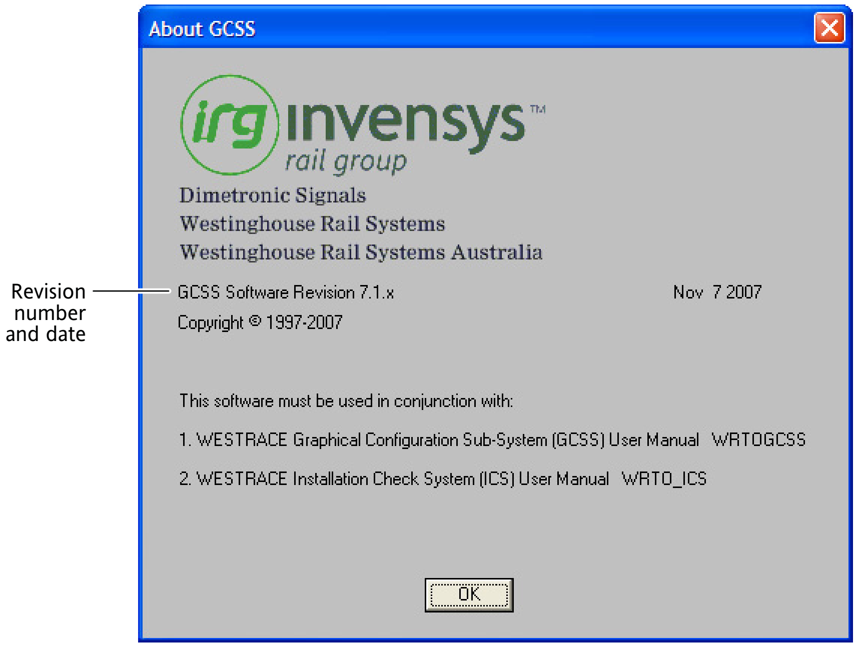


Figure 1.3 About GCSS dialog box

2. WESTRACE and the GCSS

The GCSS is computer software used by Signal Engineers to design Railway Signal logic for WESTRACE Vital Signalling Equipment. It operates on a personal computer in either a networked or stand-alone environment.

2.1 Concepts

2.1.1 WESTRACE applications

A WESTRACE application comprises WESTRACE modules (hardware) installed in one to four housings plus application data to control the WESTRACE modules. The modules receive input from and send output to signalling equipment such as signal lamps, point machines, relays and track circuits.

The application data for every system will contain:

- header information stored in PROM as text that identifies the application data;
- details of the included WESTRACE modules:
  - a single vital logic module with latches and timers;
  - a set of input and output modules;
  - a single diagnostic module or non-vital logic and diagnostic module.
- logic definition for the vital logic module;
- independent logic definition for the non-vital logic and diagnostic module (if present).

Figure 2.1 is a representation of a typical WESTRACE application.

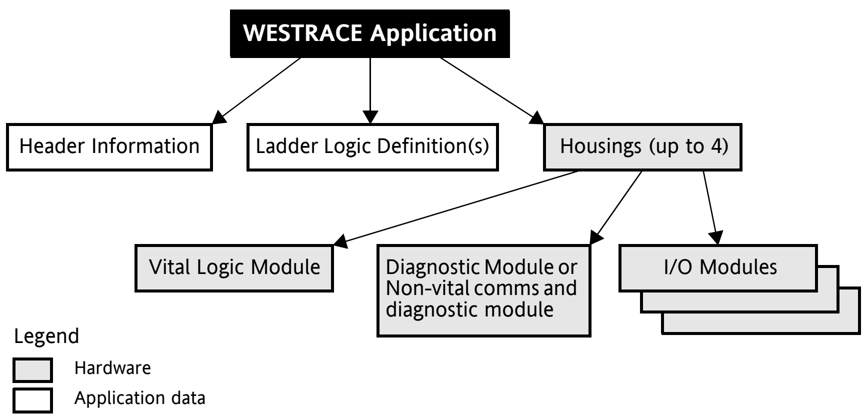


Figure 2.1 WESTRACE application—typical structure

For further information on WESTRACE refer to:

- WESTRACE Application Manual [APPM];
- WESTRACE System Overview Manual [SOM].

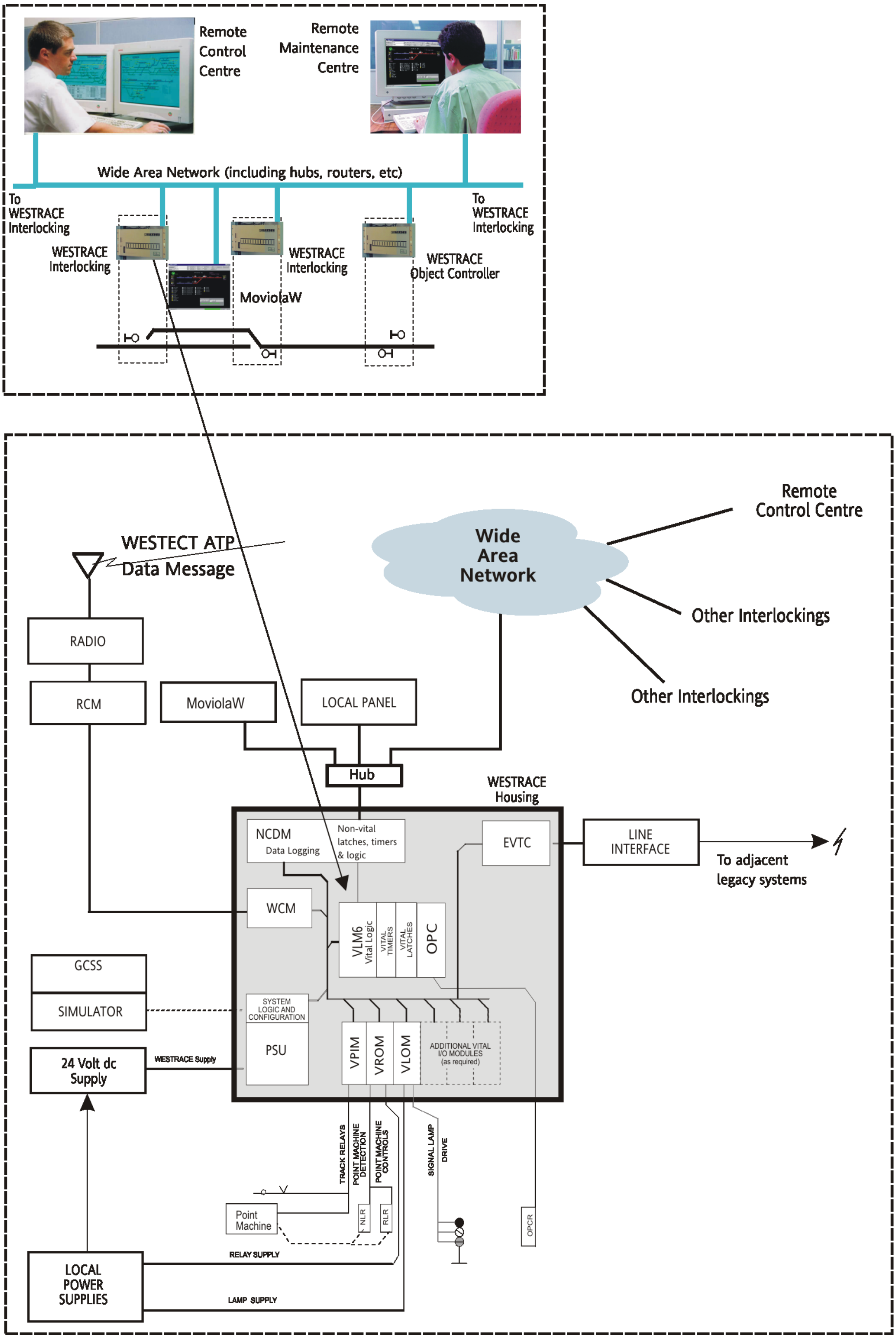


Figure 2.2 WESTRACE interface—external

2.1.2 GCSS

2.1.2.1 Overview

The GCSS graphically depicts the WESTRACE modules, communication port configuration, external interfaces and the logical and temporal relationships between these inputs and outputs.

It enables Railway Signal Engineers to:
- create or modify railway signalling applications;
- load data into PROM devices for VLM modules and download data to NVLM modules;
- produce reports to document the process.

It has facilities for defining:
- WESTRACE modules in the WESTRACE application;
- module input and output assignments;
- timers;
- ladder logic;
- communication port configuration.

The GCSS defines the logical relationship between the inputs and outputs of the WESTRACE application. It creates relay equivalent (ladder) logic circuits and timers. The GCSS checks for syntactic and semantic errors but the resulting logic must be functionally tested using GSIM or actual hardware by the Signal Engineer.

Logic that has passed the internal check and functional tests can be approved and programmed into PROMs for the VLM or downloaded to the NVLM.

2.1.2.2 GCSS Representation of WESTRACE Application Data

The GCSS divides the application data for a typical WESTRACE application into parts:
- **Vital Logic (VLM) installation**—defines the structure of the WESTRACE application.
  - Defines the modules (including possible non-vital logic or communication modules) and their physical placement in the WESTRACE housings;
  - Contains all the necessary application data excepting non-vital data used by any non-vital logic module that may be present;
  - Comprises the entire WESTRACE application data if a non-vital logic module is not present;
  - Tailored VLM installations can be created to use HVLM128, HVLM128a, VLM5 or VLM6 vital logic modules;
  - Contains its own configuration history.
- **Non-Vital Logic (NVLM) installation**—defines the structure that contains the necessary non-vital logic, plus timers, latches and port settings required by the non-vital logic module.
  - Only required when a non-vital logic module (such as NCDM or NVC/DM) is included in the VLM installation;
  - Contains its own configuration history.

- **VLM-NVLM Interface file**—defines mnemonic states passed between the VLM and NVLM installations for the same WESTRACE application.

  It contains:
  - the VLM type and the hot-standby or stand-alone configuration;
  - the NVLM module address within the VLM installation;
  - the NVLM module sequence number within the VLM installation;
  - input and output mnemonics shared between the VLM and NVLM installations;
  - its own configuration history.

  It:
  - is only required when an NVLM (such as an NCDM or NVC/DM) is part of the VLM installation;
  - is created for a particular VLM-NVLM installation pair.
- **VLM6-VLM6 Interface file**—defines mnemonic states passed between two VLM6 installations for different WESTRACE applications.

  It:
  - is only required when the VLM6 module is required to communicate with another VLM6 module;
  - A separate interface file is required for each instance of a VLM6 communicating with another VLM6;
  - contains its own configuration history.

Note: ***A VLM6 module can communicate with 0 to 16 other VLM6 modules.***

2.1.2.3 Building WESTRACE Application Data

The GCSS combines and compiles data from installation files and interface files to produce the application data required by a WESTRACE application.

There are three basic cases:

- A simple WESTRACE application based upon a VLM alone (figure 2.3). The VLM may be a HVLM128, HVLM128a, VLM5 or VLM6.

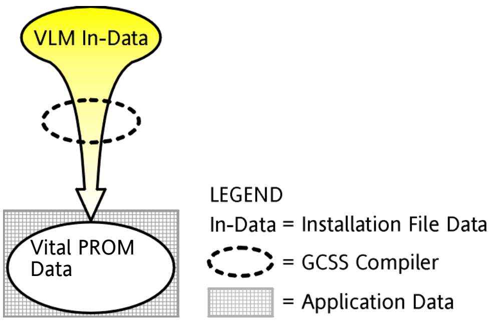


Figure 2.3 Application Data—VLM Installation only

- A WESTRACE application based upon a VLM and NVLM (figure 2.4). The VLM can be a HVLM128, HVLM128a, VLM5 or VLM6 and the NVLM an NCDM or NVC/DM. See table 4.1 on page 4-4 for valid combinations of VLM and NVLM modules.

Note
VLM-NVLM interface files are normally unique to a specific WESTRACE application

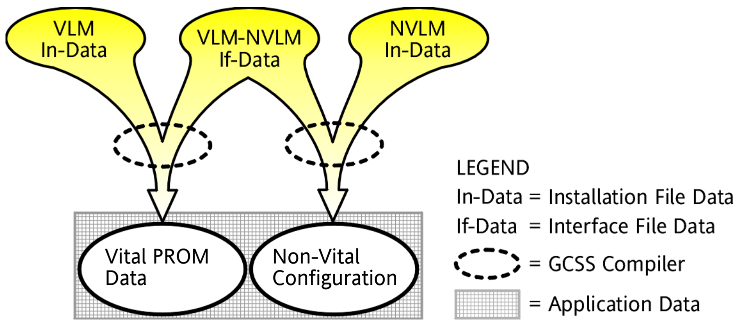
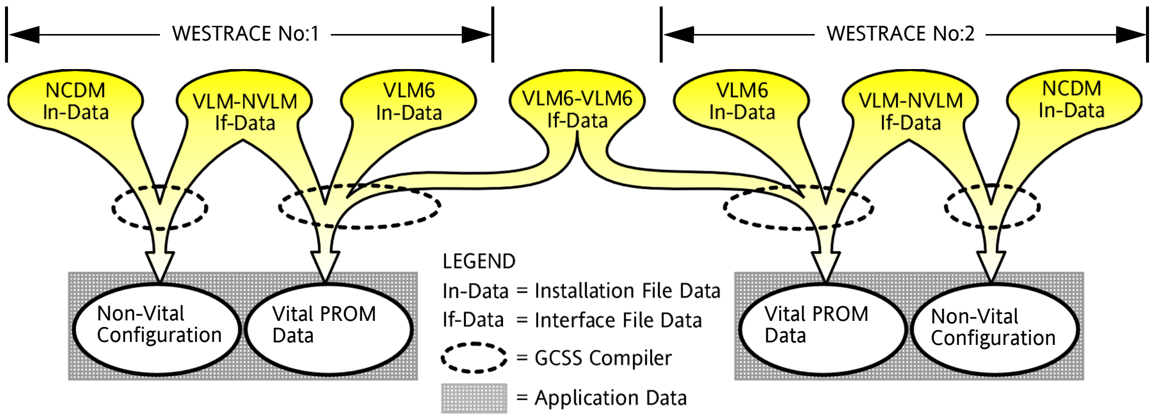


Figure 2.4 Application Data—VLM and NVLM Installations

- Two VLM6-based WESTRACE applications that are required to communicate with each other (figure 2.5). Any VLM6 can communicate with up to 16 other VLM6-based WESTRACE applications.



Note
1. VLM-NVLM interface files are normally unique to a specific WESTRACE application.
2. Every instance of a VLM6 communicating with another VLM6 requires a unique VLM6-VLM6 interface file

Figure 2.5 Application Data—Two interconnected VLM6-based WESTRACE applications

2.1.2.4 The GCS Templates Tool (GTT)

The GCS Templates Tool is an alternative to the GCSS Ladder Logic Editors for preparing ladder logic. It is used to create templates (standard logic ladders) for various signalling objects, and then to export each template to either a new or an existing GCSS installation (figure 2.6).

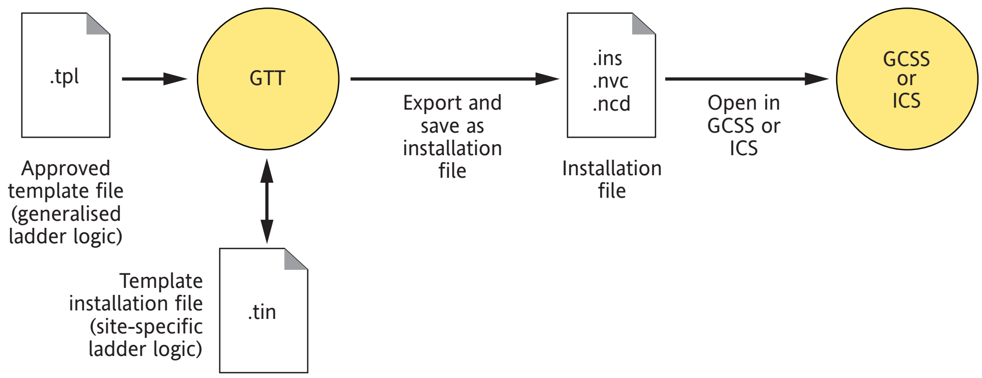


Figure 2.6 GTT file concepts

Templates:

- can use optional contacts in rungs to provide flexibility in application;
- use mnemonics that have a fixed portion and a changeable portion (the changeable portion takes a location or similar reference at the time of insertion).

These template-derived installation files can be opened in GCSS and modified if necessary. See “Locked (Template) Rungs” on page 6-1.

2.2 Modules and GCSS Editors

2.2.1 Modules

- The Logic module(s) contain the latches and timers for the application.
- The Input and Output (I/O) modules provide the physical inputs and outputs.

2.2.2 GCSS Editors

Figure 2.7 illustrates the relationship between the GCSS editors and the WESTRACE application data. The “item numbers” referenced in the explanation below refer to the labels in figure 2.7.

Housing Editor—(item ➀)

Provides a means for selecting modules and allocating them to specific WESTRACE application housing slots;

Activation:

- from the Installation window toolbar for a VLM installation;
- not used by an NVLM installation.

Module Editor—(items ➁ and ➂)

Configures the inputs, outputs, timers, and latches of all modules including the HVLM128, HVLM128a and VLM5. The VLM6 and NVLM modules are special cases. See below.

Activated from:

- the Housing Editor window toolbar for a VLM installation;
- the Installation window toolbar for an NVLM installation.

Editing an NVLM Module Within a VLM Installation—(item ➁)

- Select or create a VLM-NVLM Interface file[1];
- Edit the inputs and outputs in the interface file from the point of view of the VLM in the VLM installation[1].

Editing a VLM6 Module Within a VLM Installation—(item ➁)

- Edit timers and latches for the VLM6;
- Select or create a VLM6-VLM6 Interface file[1];
- Edit the inputs and outputs in the interface file from the point of view of the parent VLM installation[1].

[1] Chapter 5 explains this in more detail.

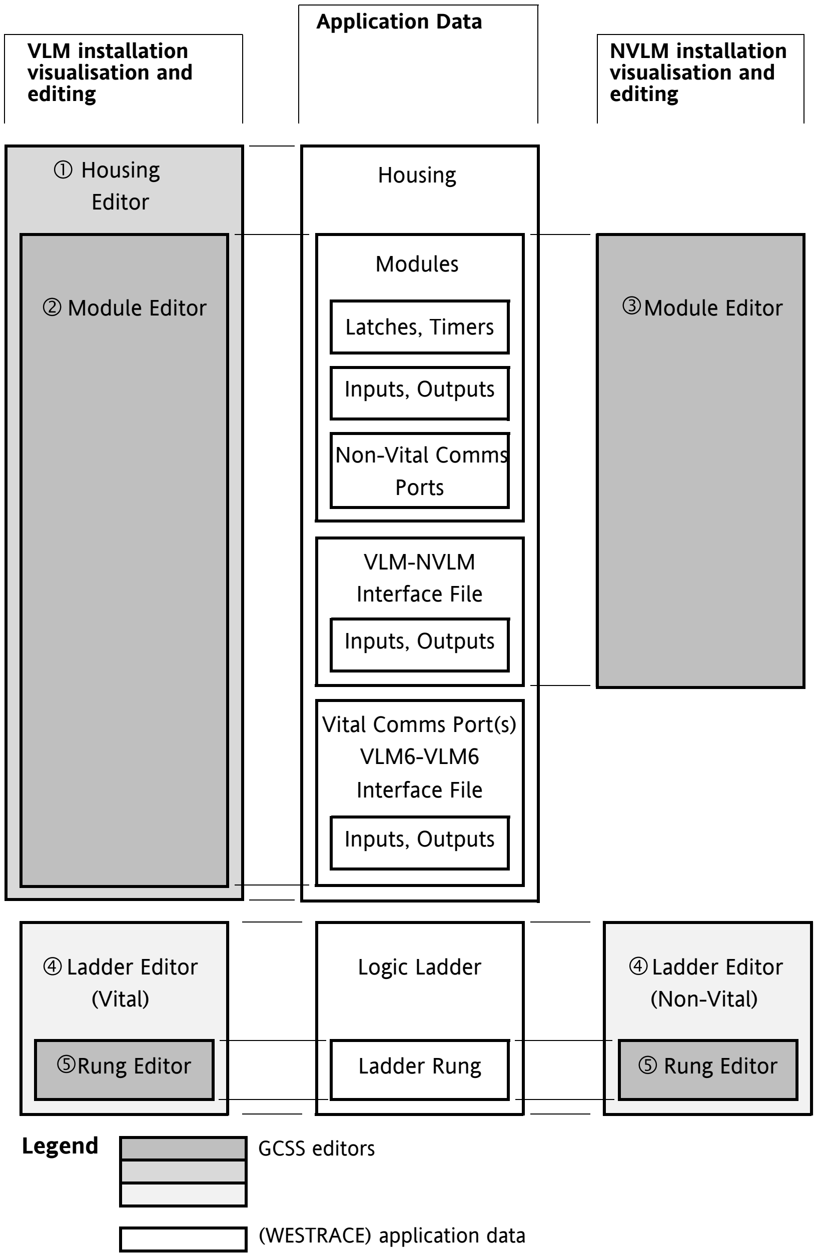


Figure 2.7 Application Data—applying GCSS editors

Editing an NVLM Module Within an NVLM Installation—(item ③)

- Edit timers and latches for the NVLM module;
- Select a port and apply a port type to it;
- Select or create a VLM-NVLM Interface file[2];
- Edit the inputs and outputs in the interface file from the point of view of the NVLM installation[2].

[2] Chapter 5 explains this in more detail.

Ladder Editor—(item ④)

Used to create or edit relay logic in the form of ladder rungs for both the VLM and NVLM installations. Each rung represents a relay or coil and is labelled with the name allocated to the relay or coil.

Activated from the Installation window toolbar for both VLM and NVLM installations;

- Add or remove rungs in the logic ladder.

Rung Editor—(item ⑤)

Used to create boolean logic based on interconnected relay equivalent contacts to energise a relay coil.

Activated for a selected ladder rung from within the Ladder Editor for both VLM and NVLM installations;

- Add or remove rungs in the logic ladder;
- Add and interconnect or remove relay contacts on the ladder rungs

2.3 Module Types, Sizes and Housing Slot Occupancy

A module can take up more than one housing slot; it can occupy slots either side of the addressed slot. The addressed slot is the slot selected when manually allocating modules to housings so if allowance is not made for any required extra slot space, the Housing Editor will decline the assignment.

For example, a VLOMFT110 module allocated to slot 6, actually occupies slots 5, 6, and 7.

Table 2.1 lists the type, size and overlap directions for all modules. Not all modules are available in every country.

Module	Description	Total Slots Occupied	Slots Occupied to the Left of the Addressed Slot	Slots Occupied to the Right of the Addressed Slot
VLM6	Vital Logic	2	1	0 or 1[1]
VLM5	Vital Logic	2	1	0 or 1[1]
HVLM128	Hot-Standby Vital Logic	2	1	0 or 1[1]
HVLM128a	Hot-Standby Vital Logic	2	1	0 or 1[1]
VPIM12	12 × 12 Volt, Vital Parallel Input	2	1	0
VPIM24	12 × 24 Volt, Vital Parallel Input	2	1	0
VPIM50	12 × 50 Volt, Vital Parallel Input	2	1	0
VROM12	8 × 12 Volt, Vital Relay Output	2	1	0
VROM24	8 × 24 Volt, Vital Relay Output	2	1	0
VROM50	8 × 50 Volt, Vital Relay Output	2	1	0

Table 2.1 Module type, size and slot overlap

Module	Description	Total Slots Occupied	Slots Occupied to the Left of the Addressed Slot	Slots Occupied to the Right of the Addressed Slot
VLOMFS12	6 × 12 Volt, Vital (Flashing) Lamp Output	2	1	0
VLOMFS24	6 × 24 Volt, Vital (Flashing) Lamp Output	2	1	0
VLOMFS110	6 × 110 Volt, Vital (Flashing) Lamp Output	2	1	0
VLOMSS12	6 × 12 Volt, Vital (Steady) Lamp Output	2	1	0
VLOMSS24	6 × 24 Volt, Vital (Steady) Lamp Output	2	1	0
VLOMSS110	6 × 110 Volt, Vital (Steady) Lamp Output	2	1	0
VLOMFT12	12 × 12 Volt, Vital (Flashing) Lamp Output	3	1	1
VLOMFT24	12 × 24 Volt, Vital (Flashing) Lamp Output	3	1	1
VLOMFT110	12 × 110 Volt, Vital (Flashing) Lamp Output	3	1	1
VLOMST12	12 × 12 Volt, Vital (Steady) Lamp Output	3	1	1
VLOMST24	12 × 24 Volt, Vital (Steady) Lamp Output	3	1	1
VLOMST110	12 × 110 Volt, Vital (Steady) Lamp Output	3	1	1
VTC232	RS232 Vital Telemetry Continuous	1	0	0
EVTC	RS232 Enhanced Vital Telemetry Continuous	1	0	0
NVC232	RS232 Non-Vital Communications	1	0	0
NVC422	RS422 Non-Vital Communications	1	0	0
NCDM	Non-Vital logic, network comms and diagnostic	1	0	0
NVC/DM	Non-Vital logic, serial comms and diagnostic	2	1	0
CNVC	Configurable Non-Vital Communications	1	0	0
DM, DIAG	Diagnostic	1	0	0
DM128	Diagnostic (used with HVLM128)	1	0	0
DM128a	Diagnostic (used with HVLM128a)	1	0	0
GPOM110L	General Purpose Output	1	0	0
GPOM50LL	General Purpose Output	1	0	0
GPOMFSU	General Purpose Output Switching Unit	1	0	0
TCOM	Track Code Output	1	0	0
CCM	Configurable Communications	1	0	0
WCM	WESTECT Communications	1	0	0

Table 2.1 Module type, size and slot overlap *(Continued)*

1 Determined by the Compatibility Index. See table 2.2.

Housings and Slots

Each housing has 15 slots that can be used for mounting modules. There can be up to 4 housings in an application with the housings and slots numbered as follows:

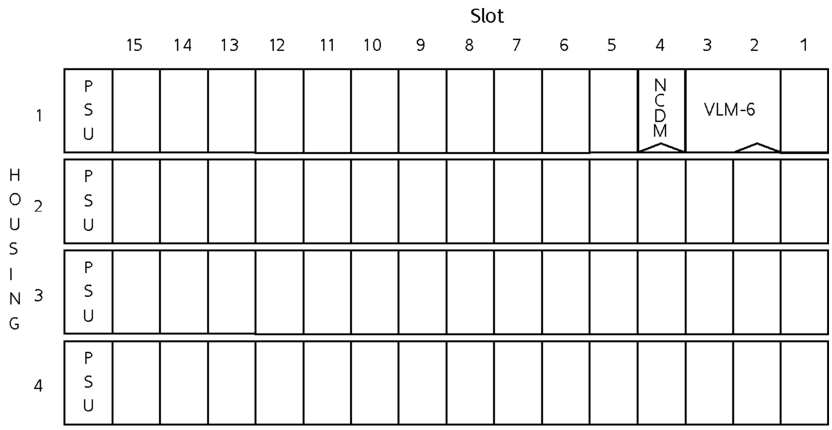


Figure 2.8 Typical housing—WRSA installation

Please note:

- The housings and slots are viewed from the front of the housings.
- Automatic placement of logic modules by the GCSS into a WESTRACE housing is determined by the Compatibility Index to suit the local WESTRACE housing.
- The Hot-Standby Vital Logic Module (HVLM), consisting of two plug-in cards and its own discrete lower backplane, is automatically placed by the GCSS into the housing slots as shown in table 2.2.
- The VLM6 consisting of two plug-in cards and its companion NCDM are treated as a single entity by the GCSS and are automatically placed by the GCSS into the housing slots as shown in table 2.2.

  The NCDM module (when used) **must** be immediately adjacent to the VLM6 in housing 1.

  The NCDM module is not mandatory and may be deleted from the WESTRACE housing if required however the system will then not have a diagnostic module.
- There must be at least three vital modules per system (including the VLM).
- By convention, the Diagnostic Module (DM128) (when used) always occupies slot 15 in housing 1 even though it can be placed anywhere in the housing.
- NVC/DM module (when used) should be immediately adjacent to the VLM in housing 1.
- Tables 4.1 and 4.2, commencing on page 4-4, show the compatibility and maximum number permitted of each module type that can be used in a WESTRACE application.

Logic Module	Company and Compatibility Index Range	
	WRSA (3–31)	Dimetronic (32–63) WRSL (64–95) Safetran (96–127)
	Housing Slots	Housing Slots
HVLM128	2–3	1–2
HVLM128A	2–3	1–2
VLM5	2–3	1–2
VLM6	2–3	1–2
NVC/DM	4–5	3–4
NCDM	4	3

Table 2.2 Logic module—placement in WESTRACE housing

2.4 Mnemonics

Mnemonic names of up to fifteen alphanumeric characters are used to represent internal states of an application including internal states reserved for special uses.

There are two categories of mnemonics:

- USER
- RESERVED

Section 2.4.3 covers RESERVED mnemonics in detail.

As a general rule, railway organisations have their own preferences for defining and organising mnemonics.

2.4.1 Initial States

Many WESTRACE modules require the initial state of a mnemonic to be specified. The specified initial state is recorded in the VLM Vital PROM Data or the NVLM Non-Vital Configuration and is applied immediately on WESTRACE startup.

Often you will be required to select between “low” or “high” for the initial state of a mnemonic. In this context:

- low is equivalent to “logic 0”, and;
- high is equivalent to “logic 1”.

2.4.2 Timers

Each timer is represented by a pair of mnemonics:

- the start condition logic state, and;
- the end condition logic state.

It is strongly recommended that both these mnemonics be used together in the logic to ensure the timer is reset immediately the timer start condition becomes false.

	Condition	Logic State of Timer Mnemonics: Start Condition (Input)	Logic State of Timer Mnemonics: End Condition (Output)
Delay To Pick	Initial state	0	0
	Timer started	1	0
	Timer's prescribed duration has elapsed since the start condition became logic 1	0 or 1	1
	Reset condition ready for a a new start	0 (Must be held for at least one VLM cycle)	0
Delay To Drop	Initial state	1	1
	Timer started	0	1
	Timer's prescribed duration has elapsed since the start condition became logic 0	0 or 1	0
	Reset condition ready for a a new start	1 (Must be held for at least one VLM cycle)	1

Table 2.3 Timer logic states

A timer can be reset whilst it is running (after the start condition has been set to logic 1 and before the end condition becomes logic 1).

Figure 2.9 applies to both vital and non-vital delay to pick timers where only the duration (time delay) is configured. Figure 2.10 applies to non-vital delay to drop timers (ie NCDM or NVC/DM).

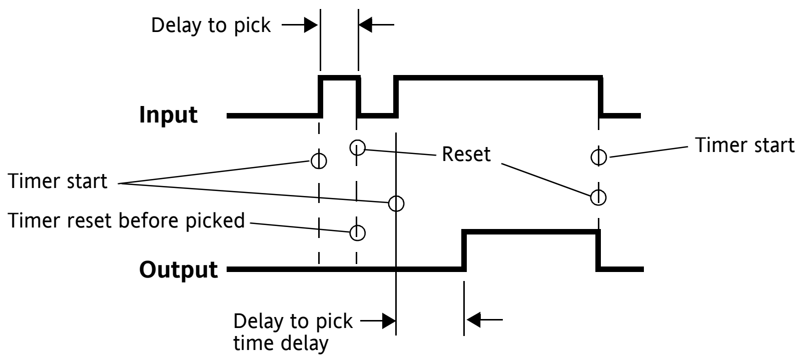


Figure 2.9 Timer—Delay To Pick—vital and non-vital—operation

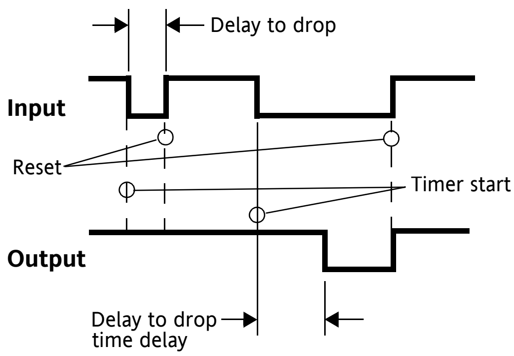


Figure 2.10 Timer—Delay to Drop—non-vital NCDM & NVC/DM only—operation

See reference [APPM], chapters 3 and 4 for application of timers.

2.4.3 Declaring, Defining and Referencing User Mnemonics

The logic state of mnemonics must be *defined* by being:

- declared as an input on an I/O module, or;
- defined as a coil in the application logic, or;
- declared to be a timer output, or;
- declared as a transfer state in a VLM-NVLM or VLM6-VLM6 interface file.

A logic state being an input on both members of an NVC or CNVC pair will result in a single definition.

The logic state of mnemonics are *referenced* by being:

- declared as an output on an I/O module;
- referenced in the application logic (a contact);
- declared to be a timer input;
- declared as a transfer state in a VLM-NVLM or VLM6-VLM6 interface file.

When a mnemonic is defined in the application logic and referenced in the application logic only, it must also be declared as a latch on the Latches Data Entry screen.

Rules for all *user* specified mnemonics:

- all user mnemonics must be defined once;
- all user mnemonics must be referenced at least once;
- mnemonics used on timers must be unique within the timers list.

Naming User Mnemonics

Names given to user mnemonics can be any combination of characters selected from:

- a to z
- A to Z
- 0 to 9

and perhaps some special characters provided they do not clash with any associated MoviolaW configuration.

Do not use:

- the IPCSM Prefix Separator character(s) used by associated MoviolaW(s)—selected from `` !"#$%&'()*+,-/:;<=>?`{|}~@_ ``
- the space character because they are C programming language string separators
- escape codes used by the C programming language such as `\n` `\o` `\t` `\r` `\b`
- the period character "`.`" in the first character position of a mnemonic name because it is used in special mnemonics (such as `.LINK`).

2.4.4 Mnemonic and Timer Limits

2.4.4.1 VLM Modules

Table 2.4 list the limits to the configuration parameters applicable to VLM modules.

	HVLM128	HVLM128A	VLM5	VLM6
Maximum Number of Mnemonics (Logic States)	2,500	2,000	4,000	4,000
Maximum Mnemonic Length	15	15	15	15
Number of Available User Timers	193	143	293	293

Table 2.4 Configuration parameters—limits—VLM

2.4.4.2 NVLM Modules

Table 2.5 list the limits to the configuration parameters applicable to NVLM modules.

	NVC/DM	NCDM
Maximum Number of Mnemonics (Logic States)	40,000	40,000
Maximum Mnemonic Length	15 char.	15 char.
Number of Available User Timers	3,000	3,000
Number of Time of Day Timers	10	10
Number of Available User Latches	15,000	15,000
Number of Set Reset (SR) Latches[1]	2,000	2,000

Table 2.5 Configuration parameters—limits—NVLM

1 SR latches reset when the system shuts down.

2.4.4.3 Maximum Timer Duration

The maximum value allowed is 59.6 h (or 3576 m).

2.4.5 Logic Module Reserved Mnemonics

Reserved mnemonics are listed according to the operations that can be performed on them. Table 2.6 lists the operations.

Category	Definition
R	Read only. The mnemonic can be referenced zero or more times.
W	Write only. The mnemonic can be defined zero times or one time.
R/W	Read, or write. The mnemonic can be referenced zero or more times. The mnemonic can be defined zero times or one time.
NONE	The mnemonic must not be defined or referenced.

Table 2.6 Mnemonics—reserved categories

Table 2.7 defines the rules for using reserved mnemonics. Refer to [APPM] for details of how and where to use these mnemonics.

2.4.5.1 VLM Reserved Latch Mnemonics

Mnemonic	Category	Usage	Initial Value	Definition
#LOC	R	Logic VLM6 only	0	The LOC (Loss of Communication) mnemonic is set to logic 0 whenever a vital communication session over a network has not received a valid message during the configurable LOC timeout period. # represents the vital communication sessions between 8 – 23.
#LOIC[1]	R	Logic	0	The LOIC (Loss of Input channel) mnemonic. See sections 2.4.7.5 and 2.4.7.6.
#LOOC[1]	R	Logic	0	The LOOC (Loss of Output Channel) mnemonic. See sections 2.4.7.5 and 2.4.7.6.
CONFLT[2]	R	Internal VLM6 only	1	Logic 1: Indicates the VLM6 - NCDM interface is operating correctly. Logic 0: Indicates a fault in the interface.
EXOPCR	R/W	Logic	0	Set by the application logic (to logic 1) to initiate a test of the OPCR.
MYMODE	R	Internal Hot Standby	0	Logic 1: Online mode Logic 0: Standby mode
NOPCR	W	Logic	0	Used to shut down the WESTRACE application and drop the OPCR Relay when set to logic 1. It is recommended that all mutually exclusive inputs functions (for example, the ON and OFF indications for Point Detection) should be proved as inputs to this latch.
NVCSEL	R/W		1	Logic 1: Selects the primary NVC or CNCV pair for input. Logic 0: Selects the secondary NVC or CNCV pair for input.
OPCRFLT	R	Logic	1	Set to logic 0 whenever a fault is detected during an OPCR test.

Table 2.7 Mnemonics—reserved latch—VLM

Mnemonic	Category	Usage	Initial Value	Definition
PIOFLT	R/W	Logic	1	Set to logic 0 whenever there is or has been a fault in any parallel I/O module. Typically, graceful degradation will cause this fault. The mnemonic should be in conjunction with user defined individual module fault mnemonics.
RCOVER	R/W	Internal Hot Standby	0	Set to logic 1 when there has been a request to change over.
RESLATA	NONE	-		Reserved for future use.
RESLATB	NONE	-		Reserved for future use.
SBYSTAT	R	Internal Hot Standby	0	Logic 1: Indicates that the connected WESTRACE is active. Logic 0: Indicates that the connected WESTRACE is dormant.
SSWITCH	R	Internal Hot Standby	0	Logic 1: Selector switch is either "Auto" or "Request" Logic 0: Selector switch is Inhibited.

Table 2.7 Mnemonics—reserved latch—VLM *(Continued)*

1 The "Loss Of Input Channel" (LOIC) state is normally used to latch any set states being received when the link was lost. "Loss Of Output Channel" (LOOC) is not used. Contrary to its name, LOOC indicates the health of the HDLC protocol on the CCM link. LOIC includes HDLC health, so LOOC is not normally used. However, LOOC may be useful during setting to work. For example, if the link installation addresses do not match, HDLC level communication will be successful but application level communication will not succeed. LOOC will then be set but LOIC will be unset.

2 The VLM6 in the Online WESTRACE application maintains this reserved mnemonic. It can be used by the application engineer to initiate a remote change over (RCOVER).

2.4.5.2 NCDM Reserved Latch Mnemonics

Mnemonic	Category	Usage	Initial Value	Definition
MYMODE	R	Internal Hot Standby	0	Logic 1: Online Logic 0: Standby
NCDMOK	R	Logic	1	Logic 1: Indicates the NCDM is operating correctly. Logic 0: Indicates the existence of a non-fatal internal NCDM fault condition.
NCONFLT	R	Logic Hot Standby	1	This is a non-vital copy of the VLM6 CONFLT reserved state. Logic 1: Indicates the VLM6 – NCDM interface is operating correctly. Logic 0: Indicates a faulty VLM6 – NCDM interface.
NETPORTOK	R	Logic	1	Logic 1: Indicates the network port is operating correctly. This does not imply that any equipment connected to the port is operating correctly. Logic 0: Indicates the network port is disconnected or the network is not functioning.

Table 2.8 Mnemonics—reserved latch states—NCDM

Mnemonic	Category	Usage	Initial Value	Definition
INCLOK	R	Logic Hot Standby	0	Logic 1: Indicates the INCL (Inter-NCDM Communications Link) is operating correctly, that is, the NCDM received a valid message over the INCL during the last cycle. Logic 0: Indicates the NCDM did not receive a valid message over the INCL during the last cycle.
RESSTATED	NONE	Logic	0	Reserved for future use.
SBMODE	R	Logic	0	A copy of the MYMODE reserved state from the adjacent NCDM in a WESTRACE hot-standby system. Only used in a hot-standby system.
SBNCDMOK	R	Logic	0	A copy of the NCDMOK reserved state from the adjacent NCDM in a WESTRACE hot-standby system. Only used in a hot-standby system.
SBNCONFLT	R	Logic Hot Standby	0	A copy of the NCONFLT reserved state from the adjacent NCDM in a WESTRACE hot-standby system. Only used in a hot-standby system.
SBNETPORTOK	R	Logic	0	A copy of the NETPORTOK reserved state from the adjacent NCDM in a WESTRACE hot-standby system. Only used in a hot-standby system.
SBSERPORT1OK	R	Logic	0	A copy of the SERPORT1OK reserved state from the adjacent NCDM in a WESTRACE hot-standby system. Only used in a hot-standby system.
SBSERPORT2OK	R	Logic	0	A copy of the SERPORT2OK reserved state from the adjacent NCDM in a WESTRACE hot-standby system. Only used in a hot-standby system.
SBVLMPORTOK	R	Logic	0	A copy of the VLMPORTOK reserved state from the adjacent NCDM in a WESTRACE hot-standby system. Only used in a hot-standby system.
SERPORT1OK	R	Logic	1	Logic 1 indicates NCDM serial port 1 is operating correctly. This does not imply that any equipment connected to the port is operating correctly.
SERPORT2OK	R	Logic	1	Logic 1 indicates NCDM serial port 2 is operating correctly. This does not imply that any equipment connected to the port is operating correctly.
TIMESYNCIN	W	Logic	0	Not used at present. Logic 1 causes the server to adjust the system time to the nearest hour (forward or back in time). Used as a server input state (from a connected client system) to synchronise the server clock with the clock of the client system.
TIMESYNCOUT	R	Logic	0	Not used at present. Set to logic 1 for one cycle every hour on the hour. Used as a client system output for the purpose of synchronising the clock(s) of the connected server(s).
VLMPORTOK	R	Logic	1	Logic 1: Indicates the configured VLM port is operating correctly. Logic 0: Indicates a valid message was not received when expected.

Table 2.8 Mnemonics—reserved latch states—NCDM *(Continued)*

NCDM Reserved States—Online

The NCDM in the online WESTRACE application maintains the reserved state mnemonics associated with itself. These mnemonics may indicate a fault with the NCDM and may require attention.

The reserved *Online* state mnemonics:

- NCDMOK
- VLMPORTOK
- NETPORTOK
- NCONFLT

may be used by the Application Engineer to initiate a remote change over (RCOVER), or to advise the Control Centre of a problem with the Online installation. The reserved *Standby* state mnemonics listed below can be used in conjunction with the *Online* state mnemonics above to ensure that a change over is only attempted when the Hot Standby WESTRACE application is healthy.

NCDM Reserved States—Standby

The NCDM in the Online WESTRACE application maintains the reserved mnemonics associated with the NCDM in the hot-standby WESTRACE application.

The reserved *Standby* state mnemonics:

- SBNCDMOK
- SBSERPORT1OK
- SBSERPORT2OK
- SBVLMPORTOK
- SBNETPORTOK
- SBNCONFLTOK
- INCLOK

may be used by the Application Engineer to inhibit a remote change-over (RCOVER), or to advise the Control Centre of a problem with the Hot Standby installation.

2.4.5.3 NVC/DM Reserved Latch Mnemonics

Mnemonic	Category	Usage	Initial Value	Definition
NVCDMOK	R	Logic	1	Logic 1: Indicates the NVC/DM is operating correctly. Logic 0: Indicates a non-fatal internal NVC/DM fault condition exists; eg low battery. This does not include port faults.
PORTSOK	R	Logic	1	Logic 1: Indicates that all configured NVC/DM ports are operating correctly. This does not necessarily imply any equipment connected to the ports is operating correctly.
RESSTATEA	NONE	NONE		Reserved for future use.
RESSTATEB	NONE	NONE		Reserved for future use.
TIMESYNCIN	W	Logic	0	Not used at present. Logic 1 causes the server to adjust the system time to the nearest hour (forward or back in time). Used as a server input state (from a connected client system) to synchronise the server clock with the clock of the client system.

Table 2.9 Mnemonics—reserved latch—NVC/DM

Mnemonic	Category	Usage	Initial Value	Definition
TIMESYNCOUT	R	Logic	0	Not used at present. Set to logic 1 for one cycle every hour on the hour. Used as a client system output for the purpose of synchronising the clock(s) of the connected server(s).
RESSTATEC	NONE	NONE		Reserved for future use.
RESSTATED	NONE	NONE		Reserved for future use.

Table 2.9 Mnemonics—reserved latch—NVC/DM *(Continued)*

2.4.6 Logic Module Reserved Timer Mnemonics

The reserved timer mnemonics follow the convention:

- the output (expiry condition) has 1 as the last character, and;
- the input (start condition) does not.

For example: for start condition mnemonic TIM, the expiry condition mnemonic is TIM1.

Refer to [APPM] for details on how to use these reserved timer mnemonics.

2.4.6.1 VLM Reserved Timer Mnemonics

Mnemonic	Category	Usage	Initial State	Definition
APPDEL	NONE	Internal	Default 120 seconds	The APPDEL timer is used to time the user defined Application Delay. The output of this timer is APPDEL1. The duration of this timer is set in the Header Data Entry.
APPDEL1	R	Logic	0	Output of the APPDEL timer and is set when the application delay expires. Normally used in the application logic to inhibit outputs and set initialisation states.
STRDEL	NONE	Internal	3 seconds	The STRDEL timer is used to time the startup delay. The output of this timer is STRDEL1. This timer is set to 3 s and cannot be altered.
STRDEL1	R	Logic	0	Output of STRDEL timer. All outputs are held at the initialisation values until this timer expires to provide time for all inputs to be processed. The application delay begins when set.
INTDET	NONE	Internal	28 seconds	The INTDET timer is used to delay energising the OPCR. This timer is set to 28 s and cannot be altered.
INTDET1	NONE	Internal	0	Output of the INTDET. System negation is initialised if set before the OPCR can energise.
RESTMA	NONE	Internal		Reserved for future use.
RESTMA1	NONE	Internal		Reserved for future use.
RESTMB	NONE	Internal		Reserved for future use.
RESTMB1	NONE	Internal		Reserved for future use.
RESTMC	NONE	Internal		Reserved for future use.

Table 2.10 Mnemonics—reserved timer—VLM

Mnemonic	Category	Usage	Initial State	Definition
RESTMC1	NONE	Internal		Reserved for future use.
RESTMD	NONE	Internal		Reserved for future use.
RESTMD1	NONE	Internal		Reserved for future use.

Table 2.10 Mnemonics—reserved timer—VLM *(Continued)*

2.4.6.2 NCDM Reserved Timer Mnemonics

Mnemonic	Category	Usage	Initial State	Definition
NAPPDEL1	R	Logic	0	Output state for application delay timer. Set when the configured application delay (since the first logic cycle) has elapsed.
NSTRDEL1	R	Logic	0	Output state for startup delay timer. Set when the fixed startup delay (since the first logic cycle) has elapsed.
RESTIMERA1	NONE	NONE		Reserved for future use.
RESTIMERB1	NONE	NONE		Reserved for future use.

Table 2.11 Mnemonics—reserved timer—NCDM

2.4.6.3 NVC/DM Reserved Timer Mnemonics

Reserved timers are those that have a predefined function. A reserved timer uses reserved input and output states, and has fixed or configurable pick and drop delays. The NVC/DM has four reserved timers.

Mnemonic	Category	Usage	Initial Value	Definition
NAPPDEL1	R	Logic	0	Output state for application delay timer. Set to logic 1 when the configured application delay (since the first logic cycle) has elapsed.
NSTRDEL1	R	Logic	0	Output state for startup delay timer. Set to logic 1 when the fixed startup delay (since the first logic cycle) has elapsed.
RESTIMERA1	NONE	NONE		Reserved for future use.
RESTIMERB1	NONE	NONE		Reserved for future use.

Table 2.12 Mnemonics—reserved timer—NVC/DM

2.4.7 I/O Module Mnemonics

This section covers module mnemonics other than inputs and outputs.

2.4.7.1 Vital Parallel Input Modules

Mnemonic Type	Logic 0	Logic 1
Fault	Indicates VPIM has detected an internal error and is operating in a degraded mode with one or more inputs ignored.	Indicates VPIM is operating with no detected faults.
Bit *n*	Indicates that input *n* is de-energised.	Indicates that input *n* is energised.

Table 2.13 Vital Parallel Input Modules

2.4.7.2 Lamp Driver Modules

Mnemonic Type	Logic 0	Logic 1
Fault	Indicates VLOM has detected an internal error and is operating in a degraded mode.	Indicates VLOM is operating with no detected errors.
Filament Proving Name (Filament Proving)	Indicates the filament has failed. ***Note:*** The state of the filament is filtered such that changes in the detected state take 1.5 seconds before being available to the VLM.	Indicates that the filament is being detected as intact.
Output Name (Lamp Control)	The lamp output is off.	The lamp output is on.

Table 2.14 Lamp Driver Modules

2.4.7.3 Flashing Lamp Driver Modules

Mnemonic Type	Logic 0	Logic 1
Fault	Indicates VLOM has detected an internal error and is operating in a degraded mode.	Indicates VLOM is operating with no detected errors.
Filament Proving Name (Filament Proving)	Indicates the filament has failed. ***Note:*** The state of the filament is filtered such that changes in the detected state take 1.5 seconds before being available to the VLM.	Indicates that the filament is being detected as intact.

Table 2.15 Flashing Lamp Driver Modules

Mnemonic Type	Logic 0	Logic 1
Output Name (Lamp Control)	The lamp output is off.	The lamp output is on.
Flash Control Name (Flash Control)	Steady illumination.	Flashing illumination.
Flash Proving Name (Flash Proving)	Indicates a fault exists that prevents the module from flashing that output. When the module is unable to flash outputs, selection of flashing in the logic results in no output.	Indicates the flashing output VLOM's ability to flash its outputs when requested.

Table 2.15 Flashing Lamp Driver Modules *(Continued)*

2.4.7.4 Vital Relay Driver Modules

Mnemonic Type	Logic 0	Logic 1
Fault	Indicates that a fault has been found by the VROM and that it is operating in a degraded mode. For example, an output has been disabled by blowing a fuse.	Indicates there are no faults detected.
Bit *n*	The relay output *n* is turned off.	The relay output *n* is turned on.

Table 2.16 Vital Relay Output Modules

2.4.7.5 Vital Telemetry Continuous Modules (VTC and EVTC)

Mnemonic Type	Logic 0	Logic 1
#LOIC (# is the port address of the VTC (0–7))	The module has not received valid input data for at least the last 2 seconds.	The module has received good input data within the last 2 seconds.
#LOOC (# is the port address of the VTC (0–7))	Indicates a loss of channel in either direction.	No loss of channel in either direction.

Table 2.17 Vital Telemetry Continuous Modules

2.4.7.6 Configurable Communications Modules (CCM)

Mnemonic Type	Logic 0	Logic 1
#LOIC (# is the port address of the CCM (0-7))	The module has not received valid input data for at least the last 2 seconds.	The module has received good input data within the last 2 seconds.
#LOOC (# is the port address of the CCM (0-7))	Indicates a loss of channel in either direction.	No loss of channel in either direction.
HDLC Status	Refer to your nearest Invensys Rail office.	Refer to your nearest Invensys Rail office.
Link Status	”	”
Message Status	”	”
Sequence Status	”	”
Header Status	”	”

Table 2.18 Configurable Communications Modules

2.4.7.7 Track Code Output Modules (TCOM)

Mnemonic Type	Logic 0	Logic 1
Change Over #	Refer to your nearest Invensys Rail office.	Refer to your nearest Invensys Rail office.
Control Code #	”	”
Code Enable	”	”
Channel Fault	”	”

Table 2.19 Track Code Output Modules

2.4.7.8 General Purpose Output Modules (GPOM)

Mnemonic Type	Logic 0	Logic 1
Fault Input Name	Refer to your nearest Invensys Rail office.	Refer to your nearest Invensys Rail office.
Dimming Control Name	”	”
Dim Status Name	”	”
Output Name (Lamp Control)	”	”
Output Status Name	”	”
Flash Control Name (Flash Control)	”	”

Table 2.20 General Purpose Output Modules

2.4.7.9 WESTECT Communications Modules (WCM)

Mnemonic Type	Logic 0	Logic 1
Output 1	Down Direction Route	Up Direction Route

Logic State for Mnemonic Type			Route Entrance Speed km/h
Output 2	Output 3	Output 4	
0	0	0	0
0	0	1	25
0	1	0	35
0	1	1	50
1	0	0	65
1	0	1	80
1	1	0	100
1	1	1	120

Table 2.21 WESTECT Communications Modules

2.5 Ladder Logic

The operation of the interlocking is defined by relay equivalent circuits, analogous to the circuits used in conventional relay interlockings. These circuits are arranged as rungs in a *ladder*.

The maximum number of rungs for any module is usually limited by the logic execution time rather than the “Maximum Allowable” shown in table 2.22. Typical maximums are also shown in table 2.22 but these can vary significantly with the complexity of the logic and the amount of logic that must be executed for each rung (see section 2.5.5).

Logic Module	Maximum Allowable Number of Logic Rungs	Typical Maximum Number of Logic Rungs[1]
VLM6	4,000	1800[2]
VLM5	4,000	1800[2]
HVLM128a	2,000	750[2]
HVLM128	2,000	750[2]
HVLM64	2,000	300[2]
VLM1	2,000	300[2]
NCDM	15,000	15,000
NVC/DM	15,000	15,000

Table 2.22 Logic modules—limits to number of logic rungs

1 Depends upon the logic complexity.

2 Less 30% for hot standby.

Each *rung* of the ladder is analogous to the circuit to operate a single *relay* or *coil* with a set of front and back *contacts* connecting the *power supply* on the left side to the *coil* and *power supply return* on the right. Each *rung* of the *ladder* is arranged and interconnected as a matrix of up to 31 columns and 16 rows of *contacts*.

The *contacts* are the state of *coils* in other rungs in the ladder. *Contacts* may also be system inputs from an I/O module. When the states of the contacts in the rung are such that the path between the power supply and the relay coil is closed, the relay will be considered *picked* and energised.

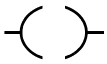
Rung representation is similar to a wired relay circuit with the relay coil located in the upper right corner of the logic representation. Each rung represents an equation in which the value of a coil is dependent on the states of the contacts in the rung and their arrangement in the rung (1 = energised and 0 = de-energised). Although the logic representation resembles relays, the ladder logic is evaluated sequentially in WESTRACE.

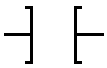
SAFETY WARNING

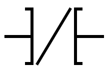
There are some significant differences in logic between WESTRACE ladder logic and the same circuits implemented in relays. Read and understand this section before designing circuits.

All logic must be thoroughly simulated or tested before use in service.

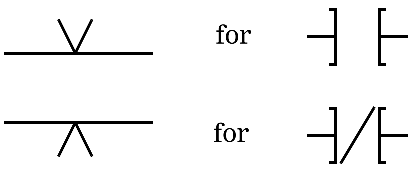
The ladder logic symbols are:


Relay Coil.


Normally open contact with relay coil in the energised state. A "front" contact in relay terminology.


Normally closed contact with relay closed in the de-energised state. A "back" contact in relay terminology.

The GCSS provides alternative representations of these contacts. To change to the alternative representations, choose the appropriate option from the View menu or use the accelerator keys **Ctrl+W** and **Ctrl+]**.



Each coil and contact must have a mnemonic assigned to it. The mnemonic should have 1 to 15 characters.

In GCSS:

- a coil can represent a latch, a timer, an output, or a user-write-able reserved mnemonic;
- a contact can represent the state of an input or output, a latch, or a reserved mnemonic in the rung logic for a coil;
- a coil that is not an output, a timer, or a reserved mnemonic must be declared as a latch.

For the purpose of illustration, figure 2.11 shows a ladder logic screen filled to capacity with a series of 31 columns of 16 contacts in parallel. However the GCSS cannot compile rungs with some complex structures. Large logic networks must be broken into smaller ones.

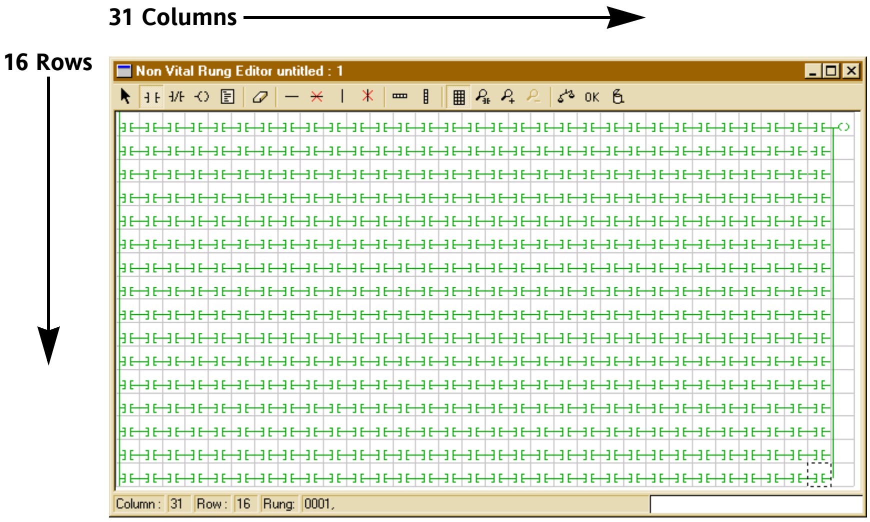


Figure 2.11 Ladder Logic screen—filled to capacity

2.5.1 Sequential Processing

SAFETY WARNING

The WESTRACE application logic, although similar in representation to relay logic, may produce different results (than a relay system) if relay logic is converted directly into rungs of a ladder.

Many relays in an all-relay interlocking can operate independently of other relays in the interlocking. In other words, more than one logic evaluation can occur at the same time. This is called parallel processing.

WESTRACE logic evaluation occurs one rung at a time beginning at rung 1. This is called sequential processing. The evaluation of all the rungs just once is called a cycle.

For each cycle, WESTRACE:

- reads all inputs;
- processes logic;
- sets outputs.

The implications of sequential processing in comparison to parallel processing are:

- Logic that produces a safe result when implemented in relays may produce an unsafe or different result when processed as ladder logic.
- More than 1 cycle of the VLM may be required to process a change of state of outputs in response to the change of state of an input. This can result in variable system response time, and in intermediate outputs after each cycle.
- Poor Application Logic design can put the VLM into an endless loop where alternating changes of state are given in response to the change of state of an input. For example, if a non-vital request to clear a signal is received, the logic should be such that no other conflicting non-vital requests can stop the original request being performed. However, this logic should not prevent a request to restore the same signal to stop.

- Intermediate outputs can be generated, ie an output may be energised when the inputs are such that the logic equation is satisfied for that output, but when the inputs are reprocessed on the next cycle, the logic equation may no longer be satisfied and the output is subsequently de-energised.

  The Application Engineer and Checking Engineer should take care to ensure that intermediate outputs cannot be generated.

SAFETY WARNING

The order of rungs in the ladder logic is critical to the correct operation of the system. Ensure that all required processing is complete prior to a change of state of outputs.

The back or down proving traditionally provided in relay interlocking circuitry, which on first thought may not seem necessary in this type of logic, is the typical means by which the problem can be mitigated.

2.5.2 Railway Signalling Examples

In railway signalling applications, it is not always possible to order the rungs for optimum processing as demonstrated by the following examples. However, the Signal Engineer should always take the processing sequence into consideration.

The typical order of operation to clear a signal is:

a) Non-vital request.

b) Route locking.

c) Route proving.

d) Approach locking.

e) Clear signal.

The rungs of logic will be ordered this way and thus the request will be performed in one cycle. The typical order of operation to restore a signal and release the locking is:

a) Cancel route proving (by track occupation).

b) Restore signal.

c) Release approach locking.

d) Normalise route locking.

If the order of logic is as described for clearing a signal, then for the restoration:

- First cycle events—cancel route proving and restore signal;
- Second cycle event—release approach locking;
- Third cycle event—normalise route locking.

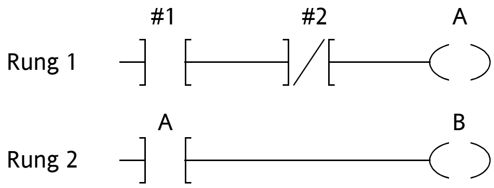
Thus it can be seen that although a signal can be cleared in one cycle, its restoration may take three cycles.

The arrangement of application logic in GCSS is as important as the care taken in the use of relay repeaters in an all-relay interlocking.

Application Logic must be arranged so that outputs cannot change until all the individual equations allowing output change are solved in proper sequence and are proved to still be correct in the output.

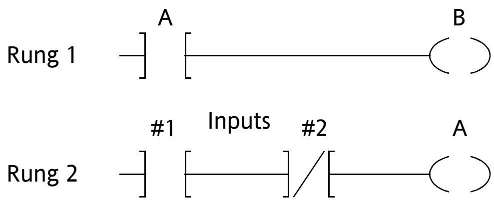
Example 1 (for One Cycle)

The following is a simple example of two sets of rungs and the number of cycles taken to energise Coil B.



	Input #1	Input #2	A	B
If Initial:	0	0	0	0
Cycle 1:	1	0	1	1

Example 1 (for Two Cycles)



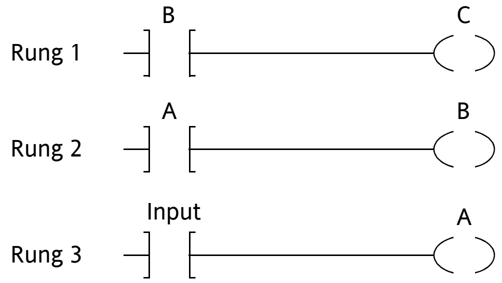
	Input #1	Input #2	A	B
If Initial:	0	0	0	0
Cycle 1:	1	0	1	0
Cycle 2:	1	0	1	1

The state of Coil B is not in step with the state of inputs until the second cycle providing the inputs have remained unchanged.

This example can be used when a slow-release function is required.

Example 2

The following more complex example shows it can take three cycles for an output to get in step with an input and that the inputs must remain unchanged for three cycles to make this possible.



	Input	A	B	C
If Initial:	0	0	0	0
End of Cycle 1:	1	1	0	0
End of Cycle 2:	1	1	1	0
End of Cycle 3:	1	1	1	1

In Example 2, if the state of the input changes to 0 before the start of cycle 2, the following sequence will occur.

	Input	A	B	C
If Initial:	0	0	0	0
End of Cycle 1:	1	1	0	0
End of Cycle 2:	0	0	1	0
End of Cycle 3:	0	0	0	1
End of Cycle 4:	0	0	0	0

Coil C, which can be an output, will change to energised after the input has been de-energised for one full cycle. This is unlikely to be what the designer intended.

Example 3

This example shows how an unstable state that exists in parallel (relay) processing does not exist in sequential (WESTRACE Ladder Logic) processing.

Figure 2.12 shows a possible fault in the design of relay Interlockings, that is, an attempt to pick a relay over its own contacts. Figure 2.13 shows the same circuit represented in ladder logic.

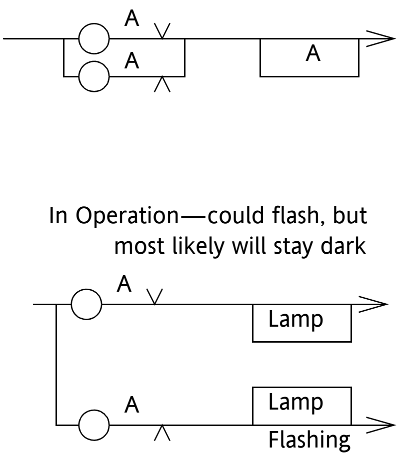


Figure 2.12 Relay Logic

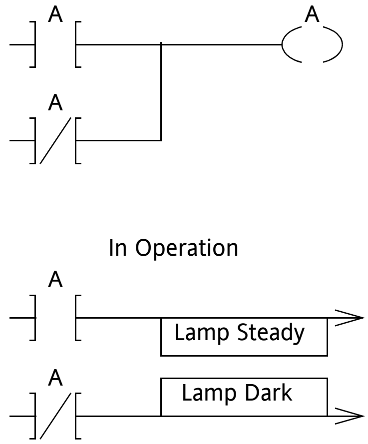


Figure 2.13 Ladder Logic

The relay logic results in:

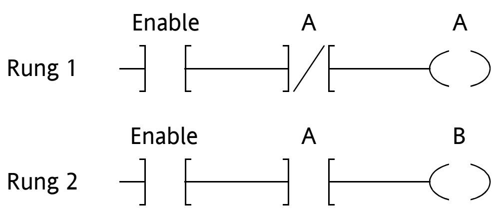
- the relay attempting to pick just enough to open its back contact but failing to close front and thereby hold its coil energised;
- the back opening removes power to the coil, causing the relay to de-energise, thus making the back contact;
- power is restored to the relay, starting another cycle.

The ladder logic operation results in the coil picking on the first cycle and staying picked.

This example illustrates there is no time at which a normally open contact and normally closed contact are both opened in ladder logic, as can be the case in relay logic.

Example 4

This example shows how an unstable ladder logic output, similar to the relay logic illustrated in Example 3, can occur.



	Enable	**A**	**B**
If Initial:	0	0	0
Cycle 1:	1	1	1
Cycle 2:	1	0	0
Cycle 3:	1	1	1
Even Cycles:	1	0	0
Odd Cycles:	1	1	1

Once enabled, A and B will continually alternate between on and off at a rate equivalent to the cycle time for the particular system. The outputs from A and B in the VROM or VLOM could be flashing or appear stable depending on cycle time.

2.5.3 Use of Latches

In addition to using latches like the neutral relays in railway signal logic, it may be necessary to use latches in the following ways:

- Dividing up large logic networks exceeding the capacity of a single rung (31 series or 16 parallel contact limits) into two or more rungs.
- Comparing two mutually exclusive input values for correspondence before use in logic, so that the relationship is established in one place in the ladder.
- Combining repeated parts of logic into one ladder rung so that this logic group can be replaced with a single contact in several other places in the logic network. This may either increase or decrease the speed of logic execution depending on whether the grouped logic would normally be solved under quiescent track conditions. The effect on cycle time should be considered.

2.5.4 Set Reset Latches

Set Reset latch relays have two input states (Set and Reset) and one output state. Table 2.23 shows the four possible logic states of the output resulting from the possible logic states of the Set and Reset inputs.

		"Set" Input Logic State	
		0	**1**
"Reset" Logic State	**0**	No change	1
	1	0	0

Table 2.23 Output states—Set Reset latch

2.5.5 Evaluation Order

The evaluation of a rung starts at the coil and moves from right to left (the coil always being at the extreme upper right of the logic structure). Minimise logic execution time by careful ordering of the contacts to limit the amount of logic processed.

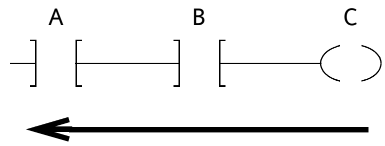
When the logic has multiple rows, the elements of each row in the same column are checked as a group as the evaluation progresses from right to left.

Earlier versions of the GCSS prohibited certain rung constructs and redrew them to a logical equivalent. However such equivalents may not have preserved the evaluation order. GCSS version 6 has reduced the amount of rung redraw and always retains the evaluation order.

In examples 1 and 2 below:

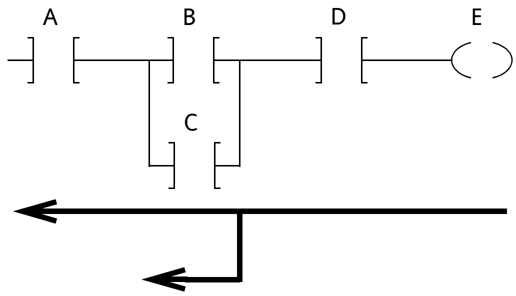
- True= Logic 1;
- False= Logic 0.

Example 1: a simple rung



- If B is true then A is checked and if true C is made true;
- If B is false then C is made false and no check is made of A.

Example 2: a more complex rung



- If D is true, then B and C are checked, if either is true then A is checked and if true then E is made true.
- If D is false then A, B and C are not checked and E is made false.

Note: ***Earlier versions of GCSS incorrectly evaluated ladder logic. WESTRACE systems compiled with GCSS version 6 may execute faster.***

2.5.6 Evaluation Time

A shorter ladder logic processing time equates to a faster overall response of the interlocking to change.

Knowing how the elements on a complete set of ladder logic rungs interact is necessary for reducing processing time and designing a system with the shortest possible cycle time.

WESTRACE has a time limit for evaluating the entire logic of any particular installation.

If the installation time exceeds the maximum time allowed, WESTRACE shuts down with an evaluation time error.

2.6 Aspects of System Design

2.6.1 WESTRACE System Initialisation

The VLM and all the system modules perform self-tests and health checks when powered up. The time taken by these tasks is approximately 28 seconds. Upon satisfactory completion of these checks, the VLM will set its serial and parallel vital outputs according to the initialisation data defined by the Application Engineer.

In the event that the initialisation tests fail, the Diagnostic Module will identify the stage at which the tests failed and the faulty module.

Satisfactory completion of the these functions results in the OPCR being energised and the initialisation outputs being sent to the controlled equipment.

Generally, in a railway signalling application, all outputs are set to 0 (de-energised) with the exception of red signal lamps which must be set to 1 (energised). The red signal lamps must be ON at initialisation because the red retaining applied by the OPCR being de-energised is lost before processed outputs are enabled.

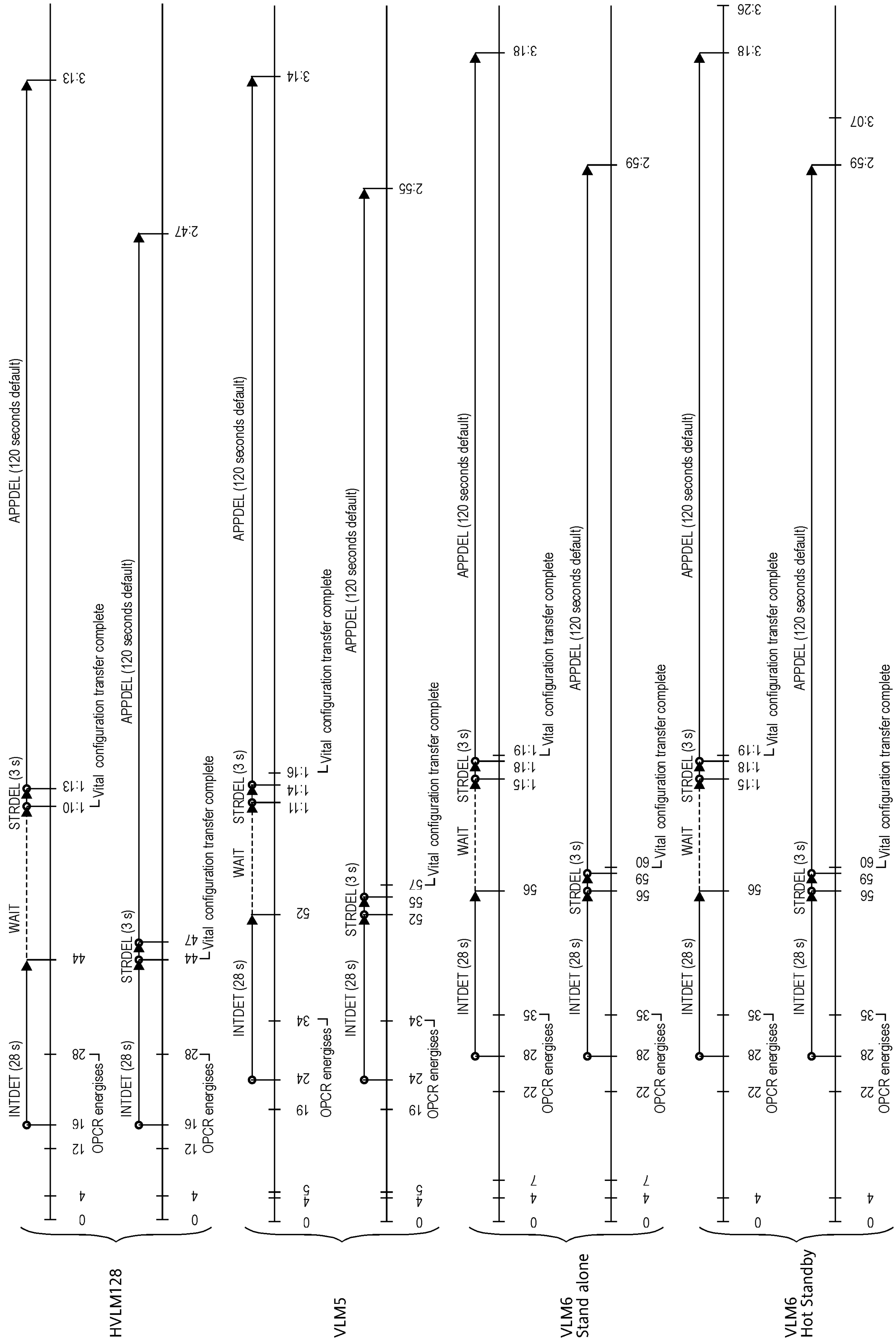


Figure 2.14 Typical WESTRACE startup events

Tables 2.24 to 2.27 summarize the startup sequence for the vital logic modules.

Time Since Power On	Event	Module LEDs	Action	Diagnostic Module Messages Enabled	Timer Running
0:00	Power On.	Display WESTRACE backplane address where the module is installed on diagnostic LEDs.	Self tests.	-	-
0:04	-	Display module software version on diagnostic LEDs.		-	-
0:12	-	-	Send initial states to diagnostic module.	Initialisation	-
0:16	INTDET starts.	HVLM128 green watchdog flashing.	Transfer vital configuration from VLM to diagnostic module. Send initial states to vital modules; read inputs. Send actual states to non-vital modules; read inputs. Evaluate timers (no application logic).		INTDET (28s).
0:28[1]	OPCR energises.				
0:44	INTDET expires.	HVLM128 green watchdog flashing.	Check that OPCR has energised. Continue actions listed above.		-
	WAIT begins.	Variable waiting period (0 to 3 minutes).			-
1:10[2]	STRDEL starts. Vital configuration transfer complete.	HVLM128 green and yellow watchdog flashing.	Send initial states to vital modules; read inputs. Send actual states to non-vital modules; read inputs. Send actual states to diagnostic module. Evaluate timers and application logic. Update hot-stand-by system if applicable.	Application delay.	STRDEL (3s).
1:13[2]	STRDEL expires. APPDEL starts.				APPDEL (Configurable, typically 2 min.)
3:13[3]	APPDEL expires.	HVLM128 green and yellow watchdog flashing.	Send actual states to vital modules; read inputs. Send actual states to non-vital modules; read inputs. Send actual states to diagnostic module. Evaluate timers and application logic. Update hot-stand-by system if applicable.	Normal	-

Table 2.24 WESTRACE startup sequence—HVLM128

1 Typical.

2 Typical; depends on WAIT.

3 Typical; depends on WAIT and APPDEL.

Time Since Power On	Event	Module LEDs	Action	Diagnostic Module Messages Enabled	Timer Running
0:00	Power On.	Display WESTRACE backplane address where the module is installed on diagnostic LEDs.	Self tests.	-	-
0:04	-	Display module software version on diagnostic LEDs.		-	-
0:05	-	NVC/DM green watchdog flashing.	NVC/DM available to transmit non-vital application messages.	-	-
0:19	-	-	Send initial states to diagnostic module.	Initialisation	-
0:24	INTDET starts.	VLM5 green watchdog flashing.	Transfer vital configuration from VLM to diagnostic module. Send initial states to vital modules; read inputs. Send actual states to non-vital modules; read inputs. Evaluate timers (no application logic).		INTDET (28s)
0:34[1]	OPCR energises.				
0:52	INTDET expires.	VLM5 green watchdog flashing.	Check that OPCR has energised. Continue actions listed above.		-
	WAIT begins.	Variable waiting period (0 to 3 minutes).			-
1:11[2]	STRDEL starts.	-		Application delay.	STRDEL (3s)
1:14[2]	STRDEL expires. APPDEL starts.	VLM5 green and yellow watchdog flashing.	Send initial states to vital modules; read inputs. Send actual states to non-vital modules; read inputs. Send actual states to diagnostic module. Evaluate timers and application logic. Update hot-stand-by system if applicable.		APPDEL (Configurable, typically 2 min.)
1:16[2]	Vital configuration transfer complete.				

Table 2.25 WESTRACE startup sequence—VLM5

Time Since Power On	Event	Module LEDs	Action	Diagnostic Module Messages Enabled	Timer Running
3:14[3]	APPDEL expires.	VLM5 green and yellow watchdog flashing.	Send actual states to vital modules; read inputs. Send actual states to non-vital modules; read inputs. Send actual states to diagnostic module. Evaluate timers and application logic. Update hot-stand-by system if applicable.	Normal	-

Table 2.25 WESTRACE startup sequence—VLM5 *(Continued)*

1 Typical.

2 Typical; depends on WAIT.

3 Typical; depends on WAIT and APPDEL.

Time Since Power On	Event	Module LEDs	Action	Diagnostic Module Messages Enabled	Timer Running
0:00	Power On.	Display WESTRACE backplane address where the module is installed on diagnostic LEDs.	Self tests.	-	-
0:04	-	Display module software version on diagnostic LEDs.		-	-
0:07	-	NCDM green watchdog flashing.	NCDM available to transmit vital and non-vital application messages.	-	-
0:22	-	-	Send initial states to diagnostic module.	Initialisation	-
0:28	INTDET starts.	VLM6 green watchdog flashing.	Transfer vital configuration from VLM to diagnostic module. Send initial states to vital modules; read inputs. Send actual states to non-vital modules; read inputs. Evaluate timers (no application logic).		INTDET (28s)
0:35[1]	OPCR energises.				
0:56	INTDET expires.	VLM6 green watchdog flashing.	Check that OPCR has energised. Continue actions listed above.		-
	WAIT begins.	Variable waiting period (0 to 3 minutes)			-

Table 2.26 WESTRACE startup sequence—VLM6—Stand-Alone

Time Since Power On	Event	Module LEDs	Action	Diagnostic Module Messages Enabled	Timer Running
1:15[2]	STRDEL starts	-	-	Application delay.	STRDEL (3s).
1:18[2]	STRDEL expires. APPDEL starts	VLM6 green and yellow watchdog flashing.	Send initial states to vital modules; read inputs. Send actual states to non-vital modules; read inputs. Send actual states to diagnostic module. Evaluate timers and application logic. Update hot-stand-by system if applicable.		APPDEL (Configurable, typically 2 min.)
1:19[2]	Vital configuration transfer complete.				
3:18[3]	APPDEL expires.	VLM6 green and yellow watchdog flashing.	Send actual states to vital modules; read inputs. Send actual states to non-vital modules; read inputs. Send actual states to diagnostic module. Evaluate timers and application logic. Update hot-stand-by system if applicable.	Normal	-

Table 2.26 WESTRACE startup sequence—VLM6—Stand-Alone *(Continued)*

1 Typical.

2 Typical; depends on WAIT.

3 Typical; depends on WAIT and APPDEL.

Time Since Power On	Event	Module LEDs	VLM Action	Diagnostic Module Messages Enabled	Timer Running
0:00	Power On.	Display WESTRACE backplane address where the module is installed on diagnostic LEDs.	Self tests.	-	-
0:04	-	Display module software version on diagnostic LEDs.	Self tests.	-	-
0:22	-	-	Send initial states to diagnostic module.	Initialisation	-
0:28	INTDET starts.	VLM6 green watchdog flashing.	Transfer vital configuration from VLM to diagnostic module. Send initial states to vital modules; read inputs. Send actual states to non-vital modules; read inputs. Evaluate timers (no application logic).		INTDET (28s)
0:35[1]	OPCR energises.				
0:56	INTDET expires.	VLM6 green watchdog flashing.	Check that OPCR has energised. Continue actions listed above.		-
	WAIT begins.	Variable waiting period (0 to 3 minutes).			-
1:15[2]	STRDEL starts			Application delay.	STRDEL (3s).
1:18[2]	STRDEL expires.	VLM6 green and yellow watchdog flashing.	Send initial states to vital modules; read inputs. Send actual states to non-vital modules; read inputs. Send actual states to diagnostic module. Evaluate timers and application logic. Update hot-stand-by system if applicable.		APPDEL (Configurable, typically 2 min.)
1:19[2]	Vital configuration transfer complete.				
3:18[3]	APPDEL expires.	VLM6 green and yellow watchdog flashing.	Send actual states to vital modules; read inputs. Send actual states to non-vital modules; read inputs. Send actual states to diagnostic module. Evaluate timers and application logic. Update hot-stand-by system if applicable.	Normal	-
3:26	-	NCDM green watchdog flashing.	NCDM available to transmit vital and non-vital application messages.	-	-

Table 2.27 WESTRACE startup sequence—VLM6—Online—Hot-Standby Installation

1 Typical.

2 Typical; depends on WAIT.

3 Typical; depends on WAIT and APPDEL.

Application Delay

WESTRACE will commence to read inputs at completion of the self tests. All outputs are held at their initialisation state for 3 seconds when the STRDEL timer expires. The OPCR is energised and VSEV is made available at this time.

It is necessary to hold some signalling outputs at the initialisation state for a longer time to ensure adequate time to bring any train movements to a halt when WESTRACE is started. This must by done through use of the APPDEL1 mnemonic in the Application Logic (APPDEL1 will be clear until the Application Delay expires).

Unlike relay interlockings with latched relays, WESTRACE has no facilities for storing states when it is not powered up. Typically, Point Controls and Ground Frames need this information to set and stabilise outputs. The APPDEL1 mnemonic can be used, in conjunction with other logic, to detect the current states of this equipment and set the internal logic appropriately during the application delay. Figure 2.15 shows an example.

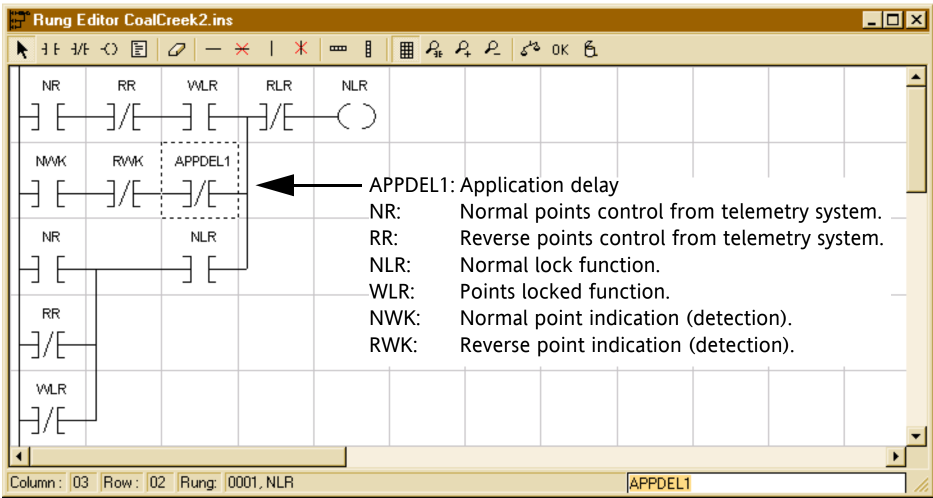


Figure 2.15 APPDEL1—using to enable latch logic resetting

In the example above, the detected lie of the points is allowed to set the latch function only in the initialisation period. In normal operation this facility is not available. If this facility were not provided, then the operator would be required to control each and every set of points every time power was applied to the system.

2.6.2 Exercising the OPCR

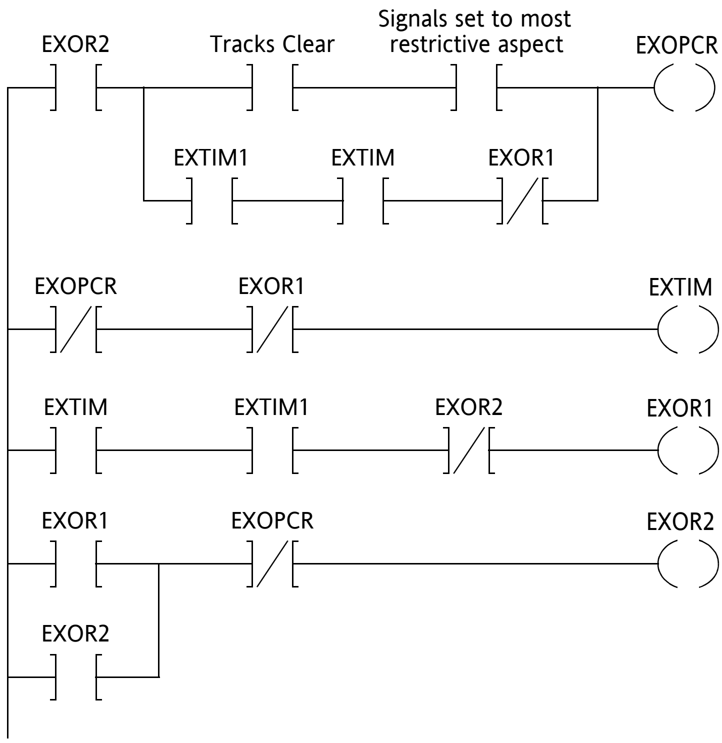
The WESTRACE Application Manual [APPM] states that the OPCR (Output Power Control Relays) must be tested regularly.

The following example uses a single timer for the timing of the exercising of the OPCR (occasional setting of EXOPCR). This fragment of application logic shows the preferred method of exercising the OPCR relay. Two consecutive periods of the timer EXTIM are used for this function.

Period 1: Sets the minimum period between exercise of OPCR.

Period 2: Allows exercise of OPCR when the other conditions are true (logic 1).

Exercise of OPCR is forced if this period expires. Period 2 is terminated and Period 1 restarted or reset and when the OPCR is exercised.



EXTIM is the mnemonic for the start conditions timer.
EXTIM1 is the mnemonic for the expired timer.
Note: The order of the rungs of logic is extremely important.

Figure 2.16 Timer—exercising the OPCR

You can extend the exercise time period by the inclusion of additional timers after the first in the order of logic, as detailed in figure 2.17, together with the replacement of the timer contacts in EXOPCR and EXOR1 with the final timer.

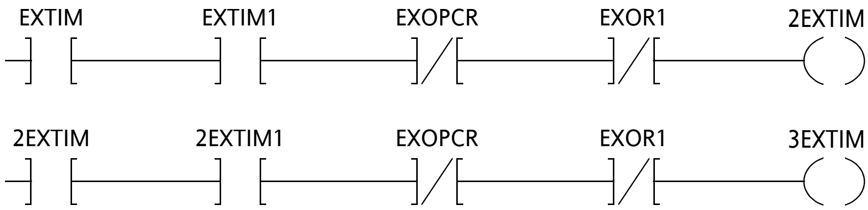


Figure 2.17 Timer—including additional

Notes on OPCR exercising:

- If the EXOPCR logic state changes to logic 1 within one HVLM cycle period of changing to logic 0, then the OPCR may not be exercised.
- If the EXOPCR logic state changes to logic 1 within 1 second of a previous transition to logic 1, then the OPCR may not be exercised.
- The OPCR is not exercised during the application delay period. If the EXOPCR logic state is set to logic 1 during the application delay period and remains set to logic 1, then the OPCR will be exercised when the HVLM enters normal operation.
- If the user defined application delay timer is specified to have a duration of less than 1 minute, then the HVLM may not be able to detect if the OPCR is not energised during WESTRACE startup. This condition will be detected if the OPCR is subsequently exercised.

2.6.3 Effect of Communication Failures on Object Controller Signal Aspects

You should carefully consider the effect on signal aspects of a communications failure between a WESTRACE interlocking and an object controller.

The data link is reversionary (ie the inputs to the object controller are assumed to be at logic zero if the link fails). You should consider how the red aspect will be controlled so that it will be illuminated if a link failure occurs.

A suitable method is to send a Logic 0 to illuminate the red (and logic 1 to turn it off). Proceed aspects might be sent as Logic 1 to illuminate. A link failure would be interpreted as Logic 0 and would therefore cause all proceed aspects to be turned off and the stop aspect to be illuminated.

The signal in rear control function (eg HR) may also need to be conditioned by the link condition so that it retains a correct aspect sequence. A failure of the object controller link should downgrade a signal in rear's proceed aspect to a caution aspect. This could be achieved by proving the data link operational (eg with LOC or LOOC) before allowing an unrestricted aspect for this function.

2.6.4 NCDM Non-Vital Configuration Settings

Note: ***All of the NCDM's internal states are returned to their default settings whenever it is re-powered. This occurs during normal WESTRACE startup, and if the NCDM self-restarts to overcome an error condition.***

2.6.4.1 Passwords

The NCDM diagnostic protocol provides basic security by requiring the external diagnostic system to login with a password. Two different passwords are used:

- read-only access
- read/write access

These passwords are stored in the Non-Vital Configuration. They can be changed by the external diagnostic system during operation. The new setting override those stored in the Non-Vital Configuration and are stored in the battery backed RAM on the NCDM.

Set these passwords through the GCSS. See section 5.3.3.

2.6.4.2 Telephone Numbers

The NCDM requires a telephone number for every diagnostic port having a modem dial out connection. See section 2.6.4.6.1.

These telephone numbers are stored in the Non-Vital Configuration. They can be changed by the external diagnostic system during operation. The new setting override those stored in the Non-Vital Configuration and are stored in the battery backed RAM on the NCDM.

Set these telephone numbers through the GCSS. See section 5.3.8.

2.6.4.3 Protocols

Terminology: Server = Slave or Field, Client = Master or Office

The following protocols can be set up for diagnostics or system control as serial connections or network sessions:

- Diagnostic—serial and network
- WSL/S2 server—serial
- WSA/S2 client—serial (ie the NCDM is an office)
- WSA/S2 server—serial (ie the NCDM is a field)
- WSL/S2 server—network
- WSA/S2 server—network (ie the NCDM is a field)
- WSA/S2 client—network (ie the NCDM is an office)
- WESTRACE vital protocol—network only—when NCDM is used in conjunction with VLM6

See:

- Section 2.6.4.4 and subsections for setting up Network Port and Sessions;
- Section 2.6.4.6 and subsections for setting up Serial Port Connections.

Note: ***We recommend that you design network and associated parameters in conjunction with a network expert before entering this data.***

2.6.4.4 Network Port and Sessions

Configure network ports and sessions as described in this section. Also see sections 5.3.1, 5.3.3 and 5.3.8.

Overview

The NCDM has a single Ethernet connection that can support:

- up to 16 vital sessions to other main and standby WESTRACEs
- up to 32 non-vital sessions to other main and standby WESTRACEs, control systems and maintenance systems (eg MoviolaW and NGetLog).

As figure 2.18 shows, each NCDM configuration can use:

- a **main** IP address for the main (on-line) WESTRACE;
- a **secondary** IP address for the standby (off-line) WESTRACE in a hot-standby pair;
- an optional **additional main** IP address for the main NCDM where a second external network may be used;
- an optional **additional secondary** IP address for the standby NCDM where a second external network may be used;
- a mandatory subnet mask (all of the NCDM's IP addresses and all the ones it needs to communicate with must be within the subnet mask range)
- an optional default gateway, plus up to 4 additional gateways.

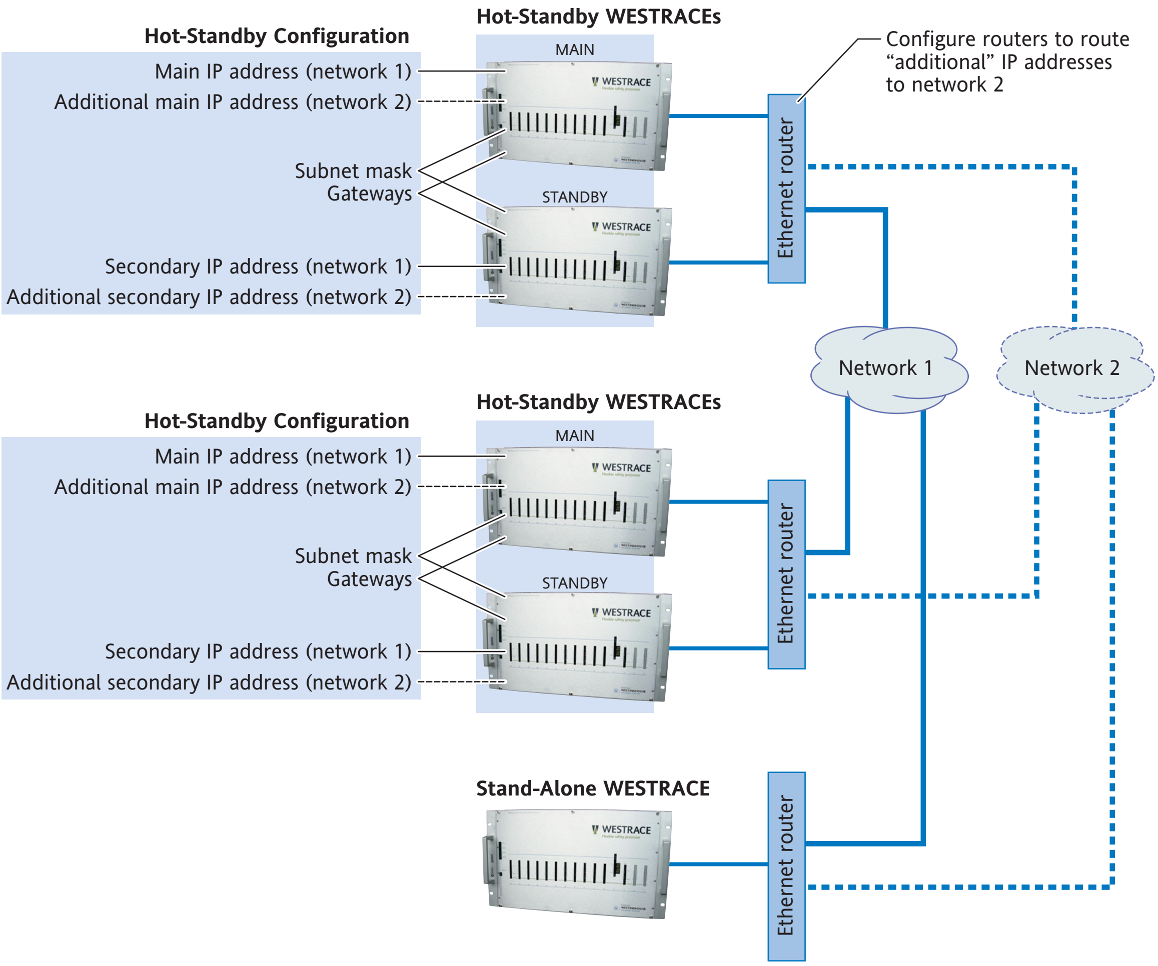


Figure 2.18 Simplified network diagram

This system of main, secondary and additional IP addresses allows any WESTRACE to communicate with the main WESTRACE of any hot-standby pair over either network 1 or network 2. Note the following:

- The network routers must be configured to route relevant addresses to the required network. This is a task for a network expert.
- For a hot-standby pair, a single configuration is loaded into both the main WESTRACE and the standby WESTRACE, and the actual IP addresses used are determined by the position of switch 3 of the NCDM's Maintenance switch bank S2 (see [APPM]).
- Only one network is necessary. The second network can be used where higher availability is required for the communications. WESTRACE will only use the data received first, and will discard data received second (from the other network).

Specifying Network Settings

The main and secondary IP addresses are entered in the NCDM General tab (figure 2.19). See also "WESTRACE Vital—Session" on page 5-47.

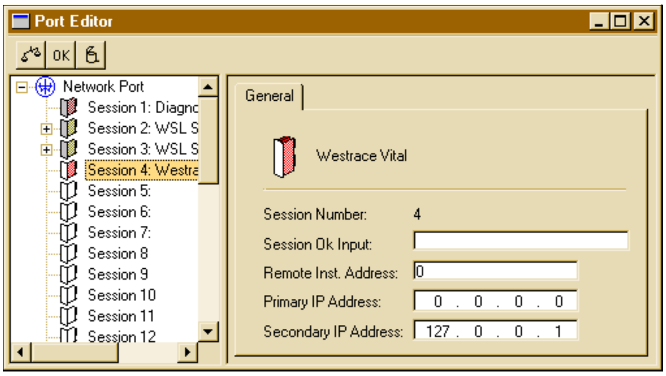


Figure 2.19 NCDM General tab

The other settings are entered in the Comments fields of the NCDM Installation (figure 2.20) as follows:.

Subnet Mask
`SUBNET MASK = xxxxxxxx`
where `xxxxxxxx` is 8 hex-decimal characters

Default Gateway
`DEF GATE = ggg.ggg.ggg.ggg`
where `ggg.ggg.ggg.ggg` is the gateway IP address

Additional Gateway
`GATEz = ddd.ddd.ddd.ddd,ggg.ggg.ggg.ggg`
where:
- `z` is the gate number between 1 to 5
- `ddd.ddd.ddd.ddd` is the IP address of the destination network
- `ggg.ggg.ggg.ggg` is the gateway IP address

Additional IP Addresses
`IP2PRI=aaa.aaa.aaa.aaa`
`IP2SEC=bbb.bbb.bbb.bbb`
where:
- `aaa.aaa.aaa.aaa` is the additional IP address for the main NCDM
- `bbb.bbb.bbb.bbb` is the additional IP address for the standby NCDM

You must enter the configuration information exactly as specified, including spaces and commas as delimiters. GCSS 7.1.x and later checks the information you enter and warns you—with a dialog box and in the Consistency Check Report (figure B.16)—of any inconsistency.

The subnet mask, default gateway and additional gateways must be separated by a space as shown in figure 2.20.

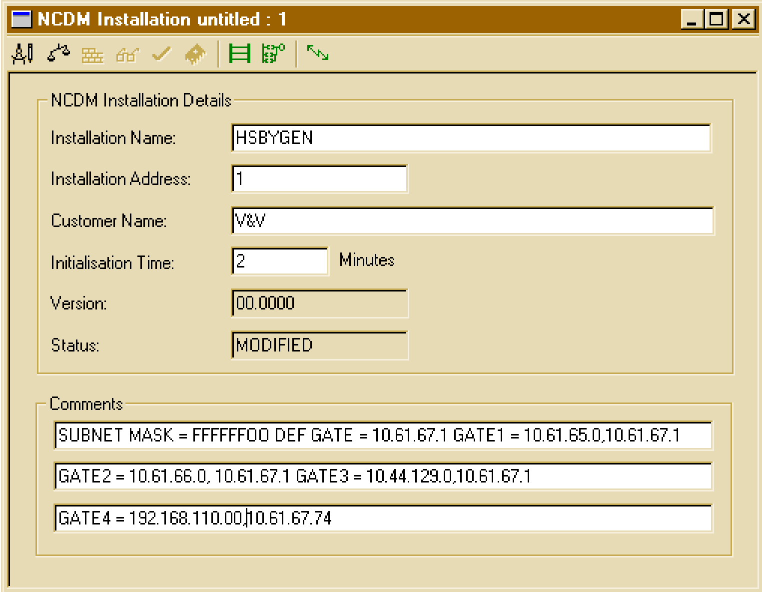


Figure 2.20 Settings—network port and sessions

The network may be configured with the following general parameters:

All sessions OK: Specifies an optional mnemonic that is energised if all sessions are operating correctly.

This mnemonic is initially set to 1 so that LOC timeouts are not reported until communication has been established.

LOC Time-out: Range: 0–20 s see section 2.6.4.5. (*Not used by NCDM).*

Session Type: Specifies which protocol shall be used by the session. Select from:

- Diagnostic;
- WSL/S2 Server;
- WSA/S2 Server;
- WSA/S2 Client;
- WESTRACE Vital
  Non-Vital Client
  Non-Vital Server **Reserved for future use**

Terminology: Server = Slave or Field, Client = Master or Office

See section 5.3.8.

2.6.4.4.1 Diagnostic Session

Session OK:	Specifies an optional mnemonic that is energised if the session is operating correctly. This mnemonic is initially set to 1 so that LOC timeouts are not reported until communication has been established.	
LOC Time-out:	Range: 1–20 s	see section 2.6.4.5.
Diagnostic Address:	Range: 1–127	each diagnostic session must have a diagnostic address different to all other sessions IDs and diagnostic addresses used in network sessions on the NCDM. Do not use address 127 for a hot-standby application as the supplementary address (255) is reserved for broadcast. Up to 5 diagnostic sessions can be configured.

See *Diagnostic—Session* on page 5-45

2.6.4.4.2 WSL/S2 Server Session

Session OK:	Specifies an optional mnemonic that is energised if the session is operating correctly. This mnemonic is initially set to 1 so that LOC timeouts are not reported until communication has been established.	
LOC Time-out:	Range: 1–20 s	see section 2.6.4.5
Session ID:	Range: 1–255	each session must have a session ID different to all other sessions IDs and diagnostic addresses used in network sessions on the NCDM
Broadcast Window:	Range: 5–100	*Not Used by NCDM for network sessions.*
Housing Address:	Range: 0–63 except for 13	Invensys Rail recommends that unique housing addresses be used for each WSL/S2 session.
I/O Slot:	Select from 16 available slots to allow configuration of 32 input, output and flashing input mnemonics	

See *WSL S2 Server—Session* on page 5-45

2.6.4.4.3 WSA/S2 Server Session

Session OK:	Specifies an optional mnemonic that is energised if the session is operating correctly. This mnemonic is initially set to 1 so that LOC timeouts are not reported until communication has been established.	
LOC Time-out:	Range: 1–20 s	see section 2.6.4.5
Session ID:	Range: 1–255	each session must have a session ID different to all other sessions IDs and diagnostic addresses used in network sessions on the NCDM
Housing Address:	Range: 1–62	Invensys Rail recommends that unique housing addresses be used for each WSA/S2 session.
Data Word Length:	32, 48, 64, 96, 128 or 256 bits.	
I/O Configuration:	Input, output and flashing input mnemonics.	

See *WSA S2 Server—Session* on page 5-46

2.6.4.4.4 WSA/S2 Client Session

Session OK:	Specifies an optional mnemonic that is energised if the session is operating correctly.	
LOC Time-out:	Range: 1–20 s	see section 2.6.4.5
Session ID:	Range: 1–255	each session must have a session ID different to all other sessions IDs and diagnostic addresses used in network sessions on the NCDM
Housing Address:	Range: 1–62	Invensys Rail recommends that unique housing addresses are used for each WSA/S2 session.
Data Word Length:	32, 48, 64, 96, 128 or 256 bits.	
I/O Configuration:	Input, output and flashing output mnemonics.	
Primary IP Address:	IP address of connected system.	
Secondary IP Address:	Use the Secondary IP address when the connected system is a hot-standby system.	

See *WSA S2 Client—Session* on page 5-46

2.6.4.4.5 WESTRACE Vital Session

Session OK: Specifies an optional mnemonic that is energised if the session is operating correctly. (*Not used by NCDM.*)

Remote Installation Address: WESTRACE Address of connected system.

Primary IP Address: IP address of connected system.

Secondary IP Address: Use the Secondary IP address when the connected system is a hot-standby system.

Set the Secondary IP address to the default value (127.0.0.10 when the connected system is a stand-alone system.

The Primary IP Address and the Secondary IP Address must be different.

See *WESTRACE Vital—Session* on page 5-47

2.6.4.5 Network Timeout

The timeout values for the network port and network sessions are configurable in the range 1 - 20 seconds. The recommended value is 5 seconds for all network sessions.

Note: ***The NCDM does not use the network port timeout value. It detects loss of network connection immediately based on hardware signal level detection.***

2.6.4.6 Serial Port Connections

The NCDM has two physical serial communication ports therefore only two serial communication protocols can be set up. The lowest numbered serial configured port on the GCSS user interface maps to physical serial port 2 on the NCDM NCD PFM. The other serial configured port (if any) maps to physical serial port 3 on the NCDM NCD PFM.

2.6.4.6.1 Diagnostic Serial Ports

Data transfer rate: 1200, 2400, 4800, 9600, 19200, 38400, 57600 or 64000 bps

Connection type:	Permanent	fault logged if port stops polling the NCDM.
	Temporary	No fault logged when an external diagnostic system stops polling the NCDM.
	Dial-in	modem answers incoming calls from an external diagnostic system.
	Dial-out	modem dials an external diagnostic system when a fault occurs. No fault logged when an external diagnostic system stops polling the NCDM.

	Dial in and out	combines the two previous options.
Handshake:	None	
	RTS/CTS	NCDM waits for CTS before transmitting.
Carrier Detect:	Yes	NCDM waits for DCD before transmitting.
	No	DCD signal not used (ignored).
Port Enabled (PEN):	Enabled	port output drivers (TxD, CLKOUT, DTR, RTS) always enabled (active). Suitable for point-to-point or modem links.
	Transmitting	output drivers disabled (tri-stated) except when the NCDM is transmitting; essential for multi-drop links.
Diagnostic Address:	Range 1–127	multiple NCDM slaves on a multi-drop link must have a unique address.
Modem Initialisation string:	E0V1	Use to enter any modem initialisation commands. Use the example given or substitute for your modem. E0 Echo Disabled. V1 Verbose responses. Add any other commands required by the modem.
Modem Auto-answer string:	S0=3	Use to set the number of rings before the modem on a dial-in port will answer. Example shows answer on 3rd ring. Parameter only required where the port must answer a call.
Default Telephone Numbers:	Specify up to five telephone numbers for a dial-out port. If a fault occurs, the numbers are tried in sequence, then repeated, until a successful connection is made. Use other characters supported by your modem, eg T Tone Dial , Delay 2 seconds	
Connected:	Specifies an optional mnemonic which is energised if an external diagnostic system is currently logged in using this port.	

See *Diagnostic (Serial)* on page 5-41

Minimum Modem Requirements

NCDM will operate with any modem that supports these few basic Hayes commands.

AT	The modem is expected to respond with **OK**.
+++	The modem is expected to switch to command mode.
ATH0Z	The modem is expected to hang up, reset, then respond with **OK**.

AT *(modem initialisation string)*	The modem is expected to respond with **OK**.
AT *(auto answer string)*	The modem is expected to enable auto-answer, then respond with **OK**.
ATD *(telephone number)*	The modem is expected to dial the specified number, then respond with **CONNECT** if the connection is successful.

NCDM will ignore all other responses.

You must ensure that modems:

- are compatible with the communications circuits;
- are compatible with the modem(s) at the remote end;
- use the command subset as shown.

The GCSS or NCDM will not validate any of the user supplied modem commands.

2.6.4.6.2 Control System Serial Ports in General

Most settings are common to all three control system protocols available for serial ports (WSL/S2 slave, WSA/S2 slave, and WSA/S2 master).

Data transfer rate:	1200, 2400, 4800, 9600, 19200, 38400, 57600 or 64000 bps	
Handshake:	None	
	RTS/CTS	NCDM waits for CTS before transmitting.
Carrier Detect:	Yes	NCDM waits for DCD before transmitting.
	No	DCD is ignored (not used).
Port Enabled:	Enabled	Port output drivers (TxD, CLKOUT, DTR, RTS) always enabled. Suitable for point-to-point or modem links.
	Transmitting	Output drivers disabled (tri-stated) except when the NCDM is transmitting; essential for multi-drop links.
I/O mappings:	S2 housing address (for each S2 address)	
	Input, output and flashing mnemonics.	
Module (Housing) OK:	Optional mnemonic which is energised if messages to or from the S2 address are operating correctly.	
All Modules OK:	Optional mnemonic which is energised if all S2 addresses handled by the port are working correctly.	

2.6.4.6.3 Control System Serial Port—WSL/S2 Server

The NCDM behaves as a WSL/S2 field station when this protocol is selected and can respond to up to 63 different S2 housing addresses.

For each configured housing address, the NCDM emulates up to 16 “cards”, each with 32 input bits *and* 32 output bits. This differs from a “real” S2 housing, where each card is either an input or an output card.

Primary Port: Two slave telemetry ports can be configured as a duplicated pair. This parameter is used by the secondary of such a pair to hold the number of the primary port. Both ports must use the same protocol, but they can use different data transfer rates.

Broadcast Window: Defines the length of the time window (in milliseconds) within which each slave must respond.

- Typically set to 25 ms.
- Only applicable if the WSL/S2 office supports the Broadcast message.

See *WSL S2 Server (Serial)* on page 5-42

2.6.4.6.4 Control System Serial Port—WSA/S2 Server

The NCDM behaves as a WSA/S2 field station when this protocol is selected and can respond to up to 62 different S2 addresses.

Tx Clock: Specifies whether the transmit clock is generated by the NCDM or the external device.

Rx Clock: Specifies whether the receive clock is generated by the NCDM, or the external device, or is reconstructed from the data.

Primary Port: Two slave telemetry ports can be configured as a duplicated pair. This parameter is used by the secondary of such a pair to hold the number of the primary port. Both ports must use the same protocol and data word length, but they can use different data transfer rates.

Data Word Length: 32, 48, 64, 96, 128 or 256 bits

See *WSA S2 Server (Serial)* on page 5-44

2.6.4.6.5 Control System Serial Port—WSA/S2 Client

The NCDM behaves as a WSA/S2 office when this protocol is selected and can control up to 62 different S2 addresses.

Tx Clock: Specifies whether the transmit clock is generated by the NCDM or the external device.

Rx Clock: Specifies whether the receive clock is generated by the NCDM, or the external device, or is reconstructed from the data.

Data Word Length: 32, 48, 64, 96, 128 or 256 bits

Inter-scan Delay: Defines the gap in milliseconds between the end of one message and the start of the next.

Should exceed the sum of:

- 10 ms;
- any RTS-CTS delay set on slave modems;
- and transmission delay over the data link (primarily due to regeneration of data).

Should be less than any timeout values.

A typical value is 20 ms.

See *WSA S2 Client (Serial)* on page 5-43

2.6.4.7 Control System Timeout

The control system port timeout is configurable in the range 1 - 60 seconds. The recommended value is 10 seconds. In this case, the office must ensure that it sends a message to the NCDM at least once every 10 seconds.

To help detect configuration errors, the NCDM has an “address timeout” which is the maximum permissible time between messages sent to or received from a particular S2 address.

The address (module) timeout is the port timeout multiplied by the number of addresses handled by the port. Thus if a port has to handle four S2 addresses, the maximum time between messages to a particular address is 40 seconds.

As an example, suppose port 2 on the NCDM is configured for addresses 1, 2, 7 and 12 but the office is (incorrectly) configured for addresses 1, 2, 8 and 12. The NCDM will ignore messages sent to address 8, and will not receive messages for address 7. After 40 seconds it will flag a “port 2, address 7 timeout” fault code (2079h) See Appendix A, page A-108.

The diagnostic port timeout is configurable in the range 1 - 60 seconds. The recommended value is 30 seconds. If no message is received from the master within this time, the NCDM will terminate the session and hang up the modem (if used).

2.6.5 Typical NCDM Network Port Configurations

NCDM modules may be connected using the 10 baseT Ethernet network compliant port. Figures 2.21 to 2.26 illustrate some typical WESTRACE NCDM configurations.

2.6.5.1 Local Control Panel (WESTCAD or SCC)

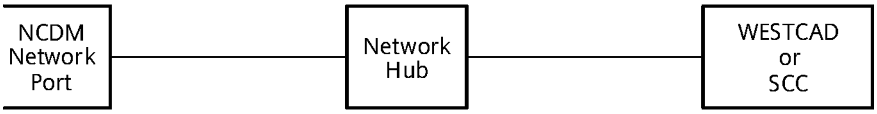


Figure 2.21 Network connection—local control panel—NCDM

The NCDM Port configuration is:

Port Type	:	Network
Protocol	:	WSA_S2_Slave or WSL_S2_Slave
Session LOC Timeout	:	To suit Office. (5 seconds recommended.)
Session ID	:	Unique session ID. (Must be different to all other session IDs and Diagnostic Addresses used in network sessions on the NCDM.)
Housing Address	:	To suit Office.

2.6.5.2 Centralised Train Control System (WESTCAD or SCC)

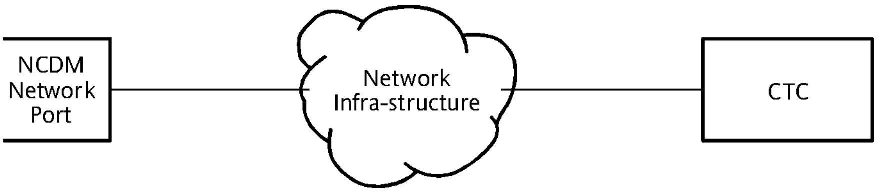


Figure 2.22 Network connection—Centralised Train Control system—NCDM

The NCDM Port configuration is:

Port Type	:	Network
Protocol	:	WSA_S2_Slave or WSL_S2_Slave
Session LOC Timeout	:	To suit Office. (5 seconds recommended.)
Session ID	:	Unique session ID. (Must be different to all other session IDs and Diagnostic Addresses used in network sessions on the NCDM.)
Housing Address	:	To suit Office.

2.6.5.3 Local MoviolaW Connection

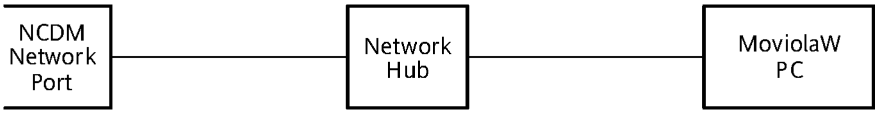


Figure 2.23 Network connection—local MoviolaW—NCDM

The NCDM Port configuration is:

Port Type	:	Network
Protocol	:	Diagnostic
Session LOC Timeout	:	To suit MoviolaW. (5 seconds recommended.)
Diagnostic Address	:	Unique Diagnostic Address. (Must be different to all other session IDs and Diagnostic Addresses used in network sessions on the NCDM.)

2.6.5.4 Remote MoviolaW Connection

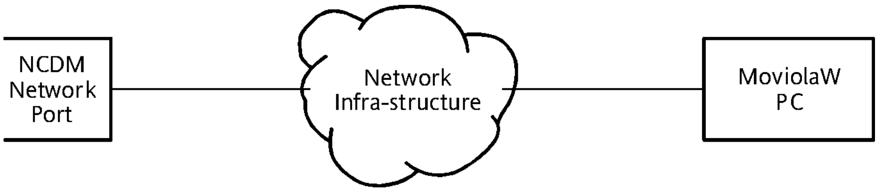


Figure 2.24 Network connection—remote MoviolaW—NCDM

The NCDM Port configuration is:

Port Type	:	Network
Protocol	:	Diagnostic
Session LOC Timeout	:	To suit MoviolaW. (5 seconds recommended.)
Diagnostic Address	:	Unique Diagnostic Address. (Must be different to all other session IDs and Diagnostic Addresses used in network sessions on the NCDM.)

2.6.5.5 WESTRACE to WESTRACE Vital Connection

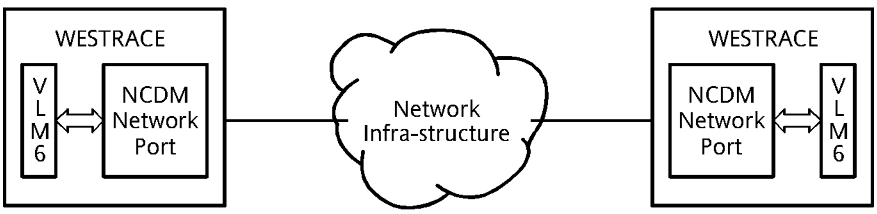


Figure 2.25 Network connection—WESTRACE to WESTRACE Vital—NCDM

The NCDM Port configuration is:

Port Type	:	Network

Protocol	:	Vital Protocol
Destination Address	:	WESTRACE Address for destination system.
IP Address	:	IP Address for destination system.

Note: ***The same IP Address must be used when more than one vital connection to the same WESTRACE Address is required, otherwise the NCDM will fail on startup.***

2.6.5.6 WESTRACE to WESTRACE Non-Vital Connection

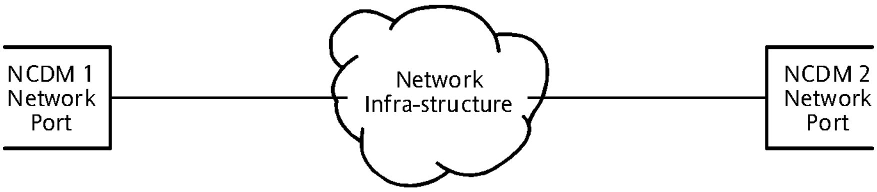


Figure 2.26 Network connection—WESTRACE to WESTRACE non-vital—NCDM

The NCDM Port configuration is:

		NCDM 1	**NCDM 2**
Port Type	:	Network	Network
Protocol	:	WSA_S2_Client	WSA_S2_Server
		(Terminology: Server = Slave or Field, Client = Master or Office)	
Session LOC Timeout	:	5 seconds recommended for both NCDM 1 and 2.	
Session ID	:	Unique session ID for both NCDM 1 and NCDM 2.	
		(Session IDs must be different to all other session IDs and Diagnostic Addresses used in network sessions on the NCDM.)	
Housing Address	:	To suit NCDM 2.	To suit NCDM 1.

2.6.6 Typical NCDM Serial Port Configurations

NCDM modules are connected using either RS232C- or RS485-level outputs (RS485 is recommended for multiple NCDM modules).

Figures 2.27 to 2.36 illustrate some typical WESTRACE NCDM serial port configurations.

2.6.6.1 Local Hard-Wired Panel

Figure 2.27 illustrates a single NCDM module communicating with an S2 field station and hard-wired panel. Separate clock signals are used in this example.

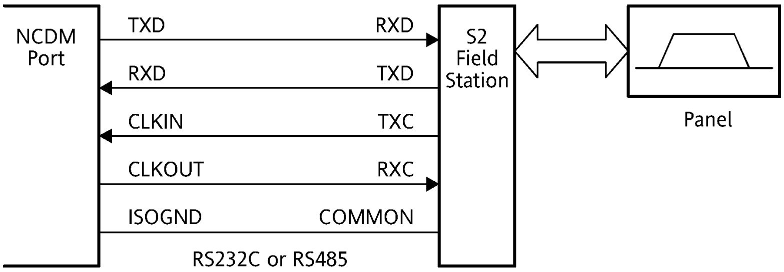
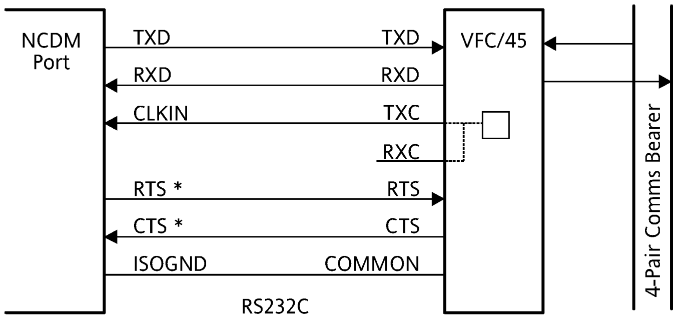


Figure 2.27 Local hard-wired panel—S2 field station—NCDM

The NCDM Port configuration is:

Port Type	:	Serial
Protocol	:	WSA_S2_Master
Handshaking	:	None
Carrier Detect	:	No
Port Enabled	:	Enabled
Data Transfer Rate	:	To suit field station
Tx Clock	:	Port
Rx Clock	:	External
Primary Port	:	None
Word Length	:	To suit field station
Housing Address(es)	:	To suit field station

2.6.6.2 Single NCDM Communicating via a Modem



* Required for multi-drop bearer only.

Figure 2.28 Communicating via modem—single NCDM

The NCDM Port configuration is:

Port Type	:	Serial
Protocol	:	WSA_S2_Slave
Handshaking	:	RTS/CTS
Carrier Detect	:	No
Port Enabled	:	Enabled
Data Transfer Rate	:	1200 bps
Tx Clock	:	External
Rx Clock	:	External
Electrical Characteristic	:	RS232
Primary Port	:	None
Word Length	:	To suit office
Housing Address(es)	:	To suit office

Note: ***RTS is dropped immediately the NCDM completes transmitting the data, which may prevent some modems from completing their transmission. (This may be changed in future releases of NCDM.)***

2.6.6.3 Multi-dropped NCDM with Clock Reconstruction

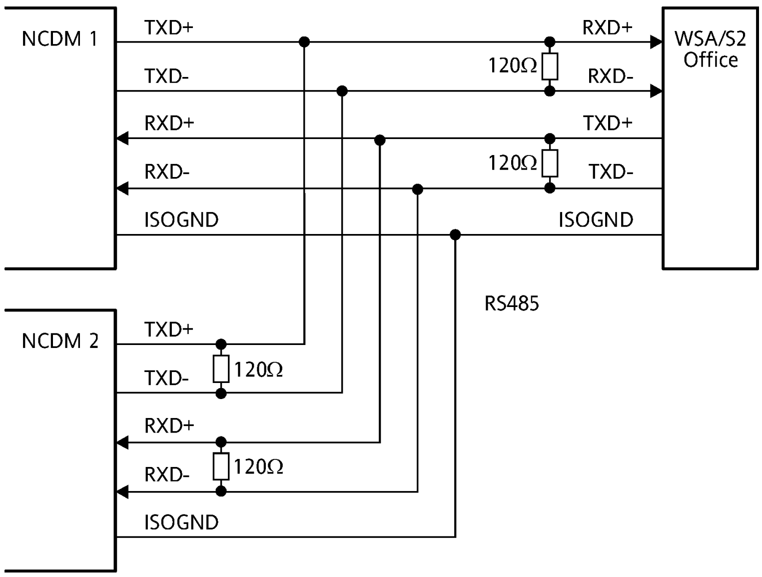


Figure 2.29 Multi-dropped with clock reconstruction—NCDM

The NCDM Port configuration is:

Port Type	:	Serial
Protocol	:	WSA_S2_Slave
Handshaking	:	None
Carrier Detect	:	No
Port Enabled	:	Transmitting
Data Transfer Rate	:	To suit office. Maximum 38400 bps
Tx Clock	:	Port
Rx Clock	:	Reconstructed
Electrical Characteristic	:	RS485
Primary Port	:	None
Word Length	:	To suit office
Housing Address(es)	:	To suit office

Note: ***120 Ω termination resistors are recommended at each most remote end of an RS485 link.***

2.6.6.4 Multi-Dropped NCDM

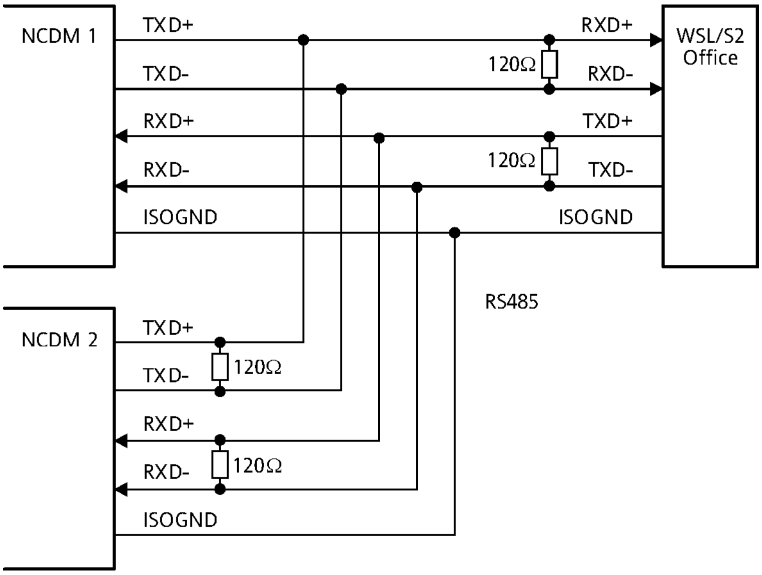


Figure 2.30 Multi-dropped—NCDM

The NCDM Port configuration is:

Port Type	:	Serial
Protocol	:	WSL_S2_Slave
Handshaking	:	None
Carrier Detect	:	No
Port Enabled	:	Transmitting
Data Transfer Rate	:	To suit office. Maximum 64000 bps
Electrical Characteristic	:	RS485
Primary Port	:	None
I/O Slots	:	To suit office
Broadcast Window	:	25
Housing Address(es)	:	To suit office

Note: ***120 Ω termination resistors are recommended at each most remote end of an RS485 link.***

2.6.6.5 NCDM Driving Multi-Dropped Field Stations

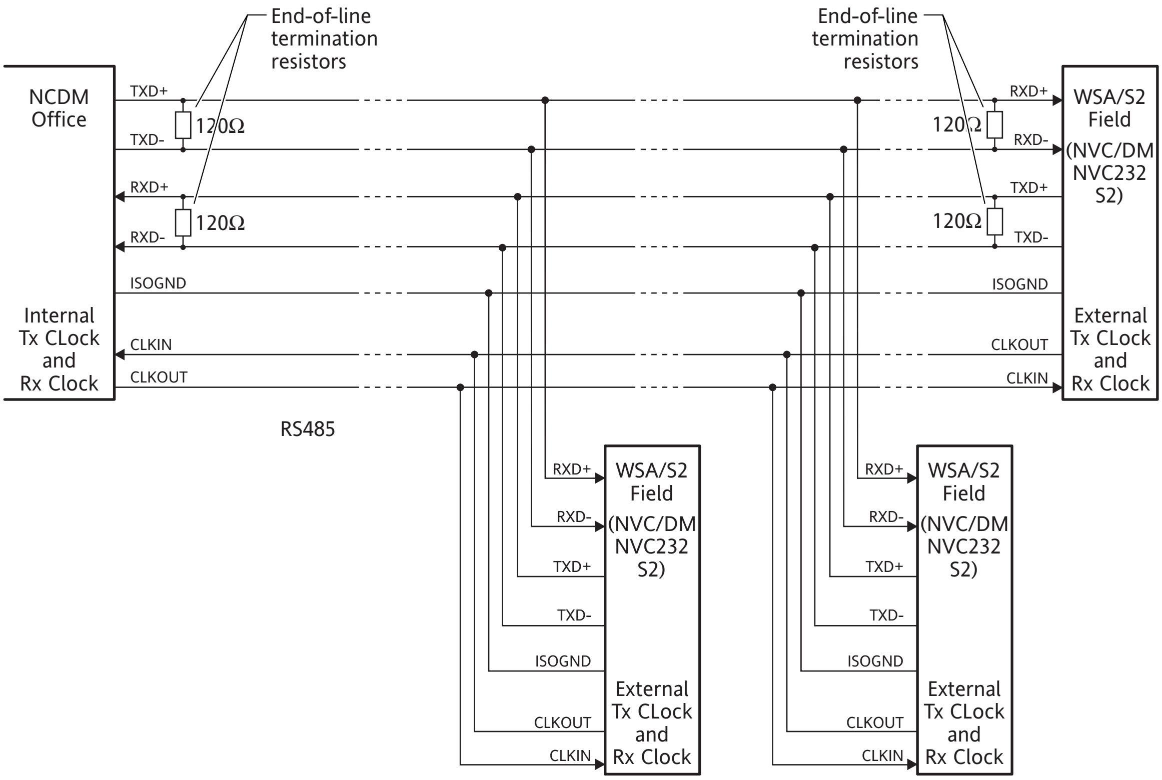


Figure 2.31 Multi-dropped field stations—NCDM

The NCDM Port configuration is:

Port Type	:	Serial
Protocol	:	WSA_S2_Master
Handshaking	:	None
Carrier Detect	:	No
Port Enabled	:	Transmitting
Data Transfer Rate	:	To suit office. Maximum 38400 bps
Tx Clock	:	Port
Rx Clock	:	Port
Electrical Characteristic	:	RS485
Primary Port	:	None
Word Length	:	To suit office
Housing Address(es)	:	To suit office

Note: ***120 Ω termination resistors are recommended at each most remote end of an RS485 link.***

2.6.6.6 WSL/S2 Office and Duplicated Bearers

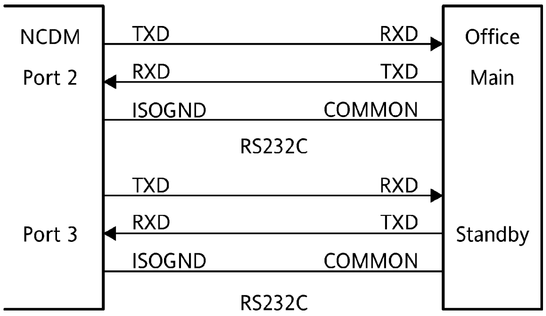


Figure 2.32 WSL/S2 office and duplicated bearers—NCDM

The NCDM Port configurations are:

		Port 2	**Port 3**
Port Type	:	Serial	Serial
Protocol	:	WSL_S2_Slave	WSL_S2_Slave
Handshaking	:	None	None
Carrier Detect	:	No	No
Port Enabled	:	Enabled	Enabled
Data Transfer Rate	:	To suit office main port.	To suit office standby port
Electrical Characteristic	:	RS232	RS232
Primary Port	:	None	2
Broadcast Window	:	25	25
Housing Address	:	To suit office	

2.6.6.7 WSA/S2 Office and Duplicated Bearers

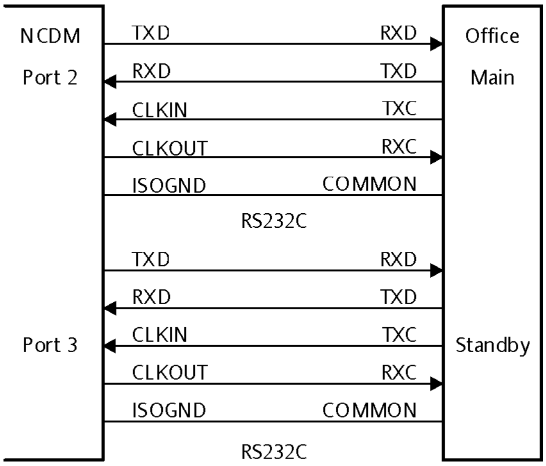


Figure 2.33 WSA/S2 office and duplicated bearers—NCDM

The NCDM Port configurations are:

		Port 2	**Port 3**
Port Type	:	Serial	Serial
Protocol	:	WSA_S2_Slave	WSA_S2_Slave
Handshaking	:	None	None
Carrier Detect	:	No	No
Port Enabled	:	Enabled	Enabled
Data Transfer Rate	:	To suit office main port.	To suit office standby port
Primary Port	:	None	2
Tx Clock	:	Port	Port
Rx Clock	:	External	External
Electrical Characteristic	:	RS232	RS232
Word Length	:	To suite office.	To suite office.
Housing Address	:	To suit office.	To suit office.

2.6.6.8 Temporary Local & Permanent Remote Moviolaw Connections

In this example, a local MoviolaW installations is connected directly to a diagnostic serial port of the NCDM, only when required.

In addition, a permanent remote MoviolaW installation is connected using a PCM interface.

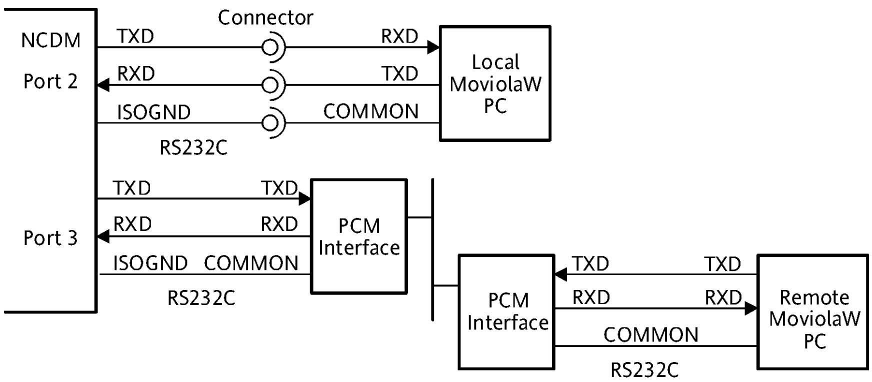


Figure 2.34 MoviolaW connections—temporary local and permanent remote—NCDM

The NCDM Port configurations are:

		Port 2	**Port 3**
Port Type	:	Serial	Serial
Protocol	:	Diagnostic	Diagnostic
Connection Type	:	Temporary	Permanent
Electrical Characteristic	:	RS232	RS232
Handshaking	:	None	None
Carrier Detect	:	No	No
Port Enabled	:	Enabled	Enabled
Data Transfer Rate	:	To suit local MoviolaW	To suit PCM interface and remote MoviolaW
Diagnostic Address	:	To suit local MoviolaW	To suit remote MoviolaW

2.6.6.9 Multi-dropped NCDM and MoviolaW Connection

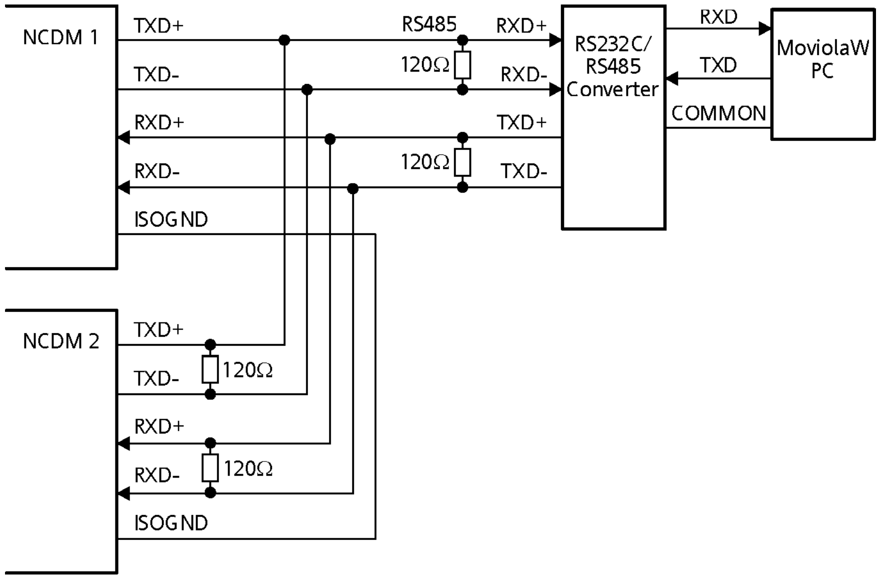


Figure 2.35 MoviolaW connection—remote—NCDM

The NCDM Port configuration is:

Port Type	:	Serial
Protocol	:	Diagnostic
Connection Type	:	Permanent
Handshaking	:	None
Carrier Detect	:	No
Port Enabled	:	Transmitting
Electrical Characteristic	:	RS485
Data Transfer Rate	:	To suit MoviolaW
Diagnostic Address	:	Must be different for each NCDM

2.6.6.10 MoviolaW Dial-up Connection

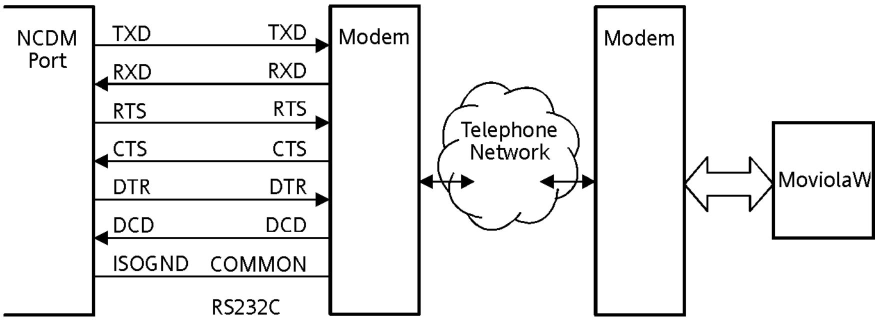


Figure 2.36 MoviolaW connection—dial-in dial-out—NCDM

The NCDM Port configuration is:

Port Type	:	Serial
Protocol	:	Diagnostic
Connection Type	:	Dial-in, dial-out or dial in and out
Handshaking	:	RTS/CTS
Carrier Detect	:	Yes
Port Enabled	:	Enabled
Electrical Characteristic	:	RS232
Data Transfer Rate	:	To suit MoviolaW (see below)
Diagnostic Address	:	To suit MoviolaW
Phone Numbers		To suit MoviolaW

To maximise data throughput, the NCDM port data transfer rate should normally be configured to be at least as fast as the maximum modem connect speed, but should not exceed the configured MoviolaW data transfer rate. For a 28–56 kbps modem, 57600 bps is an appropriate setting.

Note: ***A standard PC to modem cable should not be used to connect between the NCDM port and the modem. Doing so will connect the DSR output on the modem to the CLKOUT output on the NCDM.***

2.6.7 NVC/DM Non-Vital Configuration Settings

NVC/DM Non-Vital Configuration is entered using GCSS, compiled, then downloaded to the NVC/DM.

The Non-Vital Configuration contains:

- non-vital application logic;
- configuration parameters.

2.6.7.1 Passwords

The NVC/DM diagnostic protocol provides basic security by requiring the external diagnostic system to login with a password. Two different passwords:

- read-only access
- read/write access

These passwords are stored in the Non-Vital Configuration. They can be changed by the external diagnostic system during operation. The new setting override those stored in the Non-Vital Configuration and are stored in the battery backed RAM on the NVC/DM.

Set the these passwords though GCSS. See section 5.3.3.

2.6.7.2 Telephone Numbers

The NCDM requires a telephone number for every diagnostic port having a modem dial out connection.

These telephone numbers are stored in the Non-Vital Configuration. They can be changed by the external diagnostic system during operation. The new setting override those stored in the Non-Vital Configuration and are stored in the battery backed RAM on the NCDM.

Set these telephone numbers through the GCSS. See section 5.3.8.

2.6.7.3 Protocols

The Following protocols can be set up for diagnostics or telemetry through serial ports:

- Diagnostic (ports 2, 3 and 4 only)
- WSA/S2 slave (ie the NVC/DM is a field)
- WSA/S2 master (ie the NVC/DM is an office)
- WSL/S2 slave

See section 2.6.7.4 and subsections for setting up Serial Port Connections.

2.6.7.4 Serial Port Connections

Up to six serial communication ports can be configured on the NVC/DM. Ports 2, 3, 4, 5, 6, and 7 on the GCSS user interface map directly to the same numbered physical serial ports on the NVC/DM CIMPM.

2.6.7.4.1 Diagnostic Ports

Configure diagnostic ports.

Data transfer rate:	1200, 2400, 4800, 9600, 19200, 38400, 57600 or 64000 bps	
Connection type:	Permanent	fault logged if port stops polling the NVC/DM
	Temporary	
	Dial-in	modem answers incoming calls from an external diagnostic system
	Dial-out	modem dials an external diagnostic system when a fault occurs
	Dial in and out	combines the two previous options
Handshake:	None	
	RTS/CTS	NVC/DM waits for CTS before transmitting
Carrier Detect:	Yes	NVC/DM waits for DCD before transmitting
	No	DCD signal is ignored (not used).
Port Enabled:	Enabled	port output drivers (TxD, CLKOUT, DTR, RTS) always enabled. Suitable for point-to-point or modem links.
	Transmitting	output drivers disabled (tri-stated) except when the NVC/DM is transmitting; essential for multi-drop links
Port Address:	Range 1–127	multiple NVC/DM slaves on a multi-drop link must have a unique address
Modem Initialisation string:	E0V1	Use to enter any modem initialisation commands. Use the example given or substitute for your modem. E0 Echo Disabled V1 Verbose responses Add any other commands required by the modem
Modem Auto-answer string:	S0=3	Use to set the number of rings before the modem on a dial-in port will answer. Example shows answer on 3rd ring Parameter only required where the port must answer a call.
Default Telephone Numbers:	Specify up to five telephone numbers for a dial-out port. If a fault occurs, the numbers are tried in sequence, then repeated, until a successful connection is made. Use other characters supported by your modem, eg T Tone Dial , Delay 2 seconds	
Port OK:	Specifies an optional mnemonic which is energised if the port is operating correctly, de-energised if the port has failed.	
Connected:	Specifies an optional mnemonic which is energised if an external diagnostic system is currently logged in using this port	

See *Diagnostic (Serial)* on page 5-48.

Minimum Modem Requirements

NVC/DM will operate with any modem that supports these few basic Hayes commands.

AT	The modem is expected to respond with *OK*.
+++	The modem is expected to switch to command mode.
ATH0Z	The modem is expected to hang up, reset, then respond with *OK*.
AT *(modem initialisation string)*	The modem is expected to respond with *OK*.
AT *(auto answer string)*	The modem is expected to enable auto-answer, then respond with *OK*.
ATD *(telephone number)*	The modem is expected to dial the specified number, then respond with *CONNECT* if the connection is successful.

NVC/DM will ignore all other responses.

You must ensure that modems:

- are compatible with the communications circuits;
- are compatible with the modem(s) at the remote end;
- use the command subset as shown.

GCSS or NVC/DM will not validate any of the user supplied modem commands.

2.6.7.4.2 Telemetry Ports in General

Most settings are common to all three telemetry protocols available for serial ports (WSA/S2 slave, WSA/S2 master, and WSL/S2 slave).

Data transfer rate:	1200, 2400, 4800, 9600, 19200, 38400, 57600 or 64000 bps	
Handshake:	None	
	RTS/CTS	NVC/DM waits for CTS before transmitting
Carrier Detect:	Yes	NVC/DM waits for DCD before transmitting
	No	DCD signal is ignored (not used).
Port Enabled:	Enabled	Port output drivers (TxD, CLKOUT, DTR, RTS) always enabled. Suitable for point-to-point or modem links.
	Transmitting	Output drivers disabled (tri-stated) except when the NVC/DM is transmitting; essential for multi-drop links
Tx Clock:	Specifies whether the transmit clock is generated by the NVC/DM or the external device	
Rx Clock:	Specifies whether the receive clock is generated by the NVC/DM, or the external device, or reconstructed from the data	

Primary Port:	Two slave telemetry ports can be configured as a duplicated pair. Used by the secondary of such a pair to hold the number of the primary port. Both ports must use the same protocol, but they can use different data transfer rates. Not applicable for WSA/S2 Master.
Port OK:	Optional mnemonic which is energised if the port is operating correctly, de-energised if the port has failed.
I/O mappings: (for each S2 address)	S2 housing address
	Input, output and flashing mnemonics
Module OK:	Optional mnemonic which is energised if messages to or from the S2 address are operating correctly
All Modules OK:	Optional mnemonic which is energised if all S2 addresses handled by the port are working correctly

2.6.7.4.3 Telemetry Serial Port—WSA/S2 Slave

The NVC/DM behaves as a WSA/S2 field station when this protocol is selected and can respond to up to 62 different S2 addresses. On extra parameter may be required.

Data Word Length: 32, 48, 64, 96, 128 or 256 bits

See *WSA/S2 Slave (Serial)* on page 5-49

2.6.7.4.4 Telemetry Serial Port—WSA/S2 Master

The NVC/DM behaves as a WSA/S2 office when this protocol is selected and can control up to 62 different S2 addresses.

Data Word Length:	As for WSA/S2 Slave.
Inter-scan Delay:	Defines the gap in milliseconds between the end of one message and the start of the next. Should exceed the sum of: • 10 ms; • any RTS-CTS delay set on slave modems; • and transmission delay over the data link (primarily due to regeneration of data). Should be less than any timeout values. A typical value is 20 ms.

See *WSA/S2 Master (Serial)* on page 5-49

2.6.7.4.5 Telemetry Serial Port—WSL/S2 Slave

The NVC/DM behaves as a WSL/S2 field station when this protocol is selected and can respond to up to 63 different S2 housing addresses.

For each configured housing address, the NVC/DM emulates up to 16 “cards”, each with 32 input bits *and* 32 output bits. This differs from a “real” S2 housing, where each card is either an input or an output card.

Note: ***Do not configure more that 15 “cards” on the NVC/DM even the though the GCSS will allow configuration of 16 “cards”.***

Duplicated WSL/S2 links are supported. The NVC/DM supports both the “WSL method” where the Inhibit bit is used to select the off-line port and the “Dimetronic method”, where the off-line port is held in the “non-configured” state.

Broadcast Window: Defines the length of the time window (in milliseconds) within which each slave must respond.

- Typically set to 25 ms.
- Only applicable if the WSL/S2 office supports the Broadcast message.

See *WSL/S2 Slave (Serial)* on page 5-48

2.6.7.5 Telemetry Port Timeout

The telemetry port timeout is fixed at 10 seconds. The office must ensure that it sends a message to the NVC/DM at least once every 10 seconds.

To help detect configuration errors, the NVC/DM has an “address timeout” which is the maximum permissible time between messages sent to or received from a particular S2 address.

The address timeout is the port timeout multiplied by the number of addresses handled by the port. Thus if a port has to handle four S2 addresses, the maximum time between messages to a particular address is 40 seconds.

As an example, suppose port 5 on the NVC/DM is configured for addresses 1, 2, 7 and 12 but the office is (incorrectly) configured for addresses 1, 2, 8 and 12. The NVC/DM will ignore messages sent to address 8, and will not receive messages for address 7. After 40 seconds it will flag a “port 5, address 7 timeout” fault code (5079h) See Appendix A, page A-139.

The timeout is 30 seconds for a diagnostic port. If no message is received from the master within this time, the NVC/DM will terminate the session and hang up the modem (if used).

2.6.8 Typical NVC/DM Configurations

NVC/DM modules are connected using either RS232C- or RS485-level outputs (RS485 is recommended for multiple NVC/DM modules). Figures 2.37 to 2.43 illustrate some typical WESTRACE NVC/DM module configurations.

2.6.8.1 Local Hard Wired Panel

Figure 2.37 illustrates a single NVC/DM module communicating with an S2 field station and hard wired panel. Separate clock signals are used in this example.

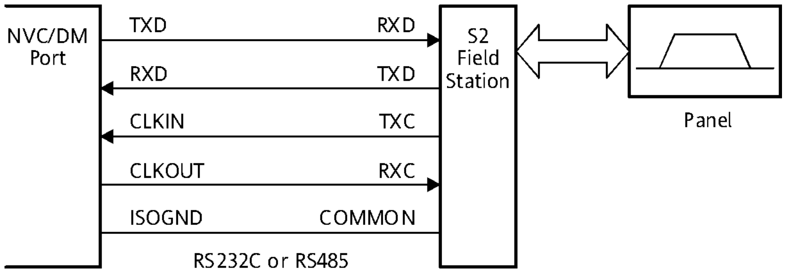


Figure 2.37 Local hard wired panel—S2 field station—NVC/DM

The NVC/DM Port configuration is:

Port Type	:	Serial
Protocol	:	WSA_S2_Master
Handshaking	:	None
Carrier Detect	:	No
Port Enabled	:	Enabled
Data Transfer Rate	:	To suit field station
Tx Clock	:	Port
Rx Clock	:	External
Primary Port	:	None
Word Length	:	To suit field station
Housing Address(es)	:	To suit field station

2.6.8.2 Single NVC/DM Communicating via a Modem

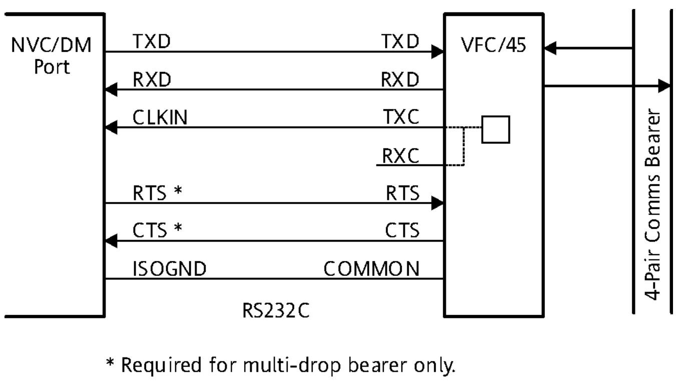


Figure 2.38 Communicating via modem—single NVC/DM

The NVC/DM Port configuration is:

Port Type	:	Serial
Protocol	:	WSA_S2_Slave
Handshaking	:	RTS/CTS
Carrier Detect	:	No
Port Enabled	:	Enabled
Data Transfer Rate	:	1200 bps
Tx Clock	:	External
Rx Clock	:	External
Primary Port	:	0
Word Length	:	To suit office
Housing Address(es)	:	To suit office

2.6.8.3 Multi-dropped NVC/DM with Clock Reconstruction

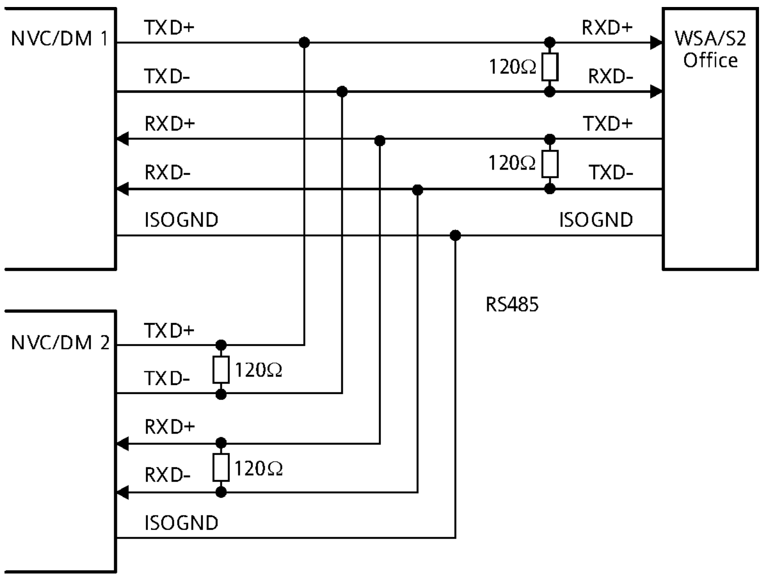


Figure 2.39 Multi-dropped with clock reconstruction—NVC/DM

The NVC/DM Port configuration is:

Port Type	:	Serial
Protocol	:	WSA_S2_Slave
Handshaking	:	None
Carrier Detect	:	No
Port Enabled	:	Transmitting
Data Transfer Rate	:	To suit office. Maximum 38400 bps
Tx Clock	:	Port

Rx Clock	:	Reconstructed
Primary Port	:	None
Word Length	:	To suit office
Housing Address(es)	:	To suit office

Note: ***120 Ω termination resistors are recommended at each most remote end of an RS485 link.***

2.6.8.4 WSL/S2 Office and Duplicated Bearers

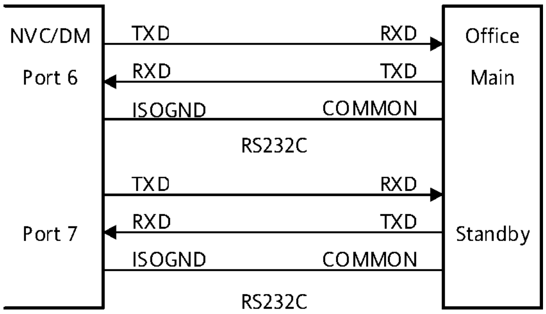


Figure 2.40 WSL/S2 office and duplicated bearers—NVC/DM

The NVC/DM Port configurations are:

		Port 6	Port 7
Port Type	:	Serial	Serial
Protocol	:	WSL_S2_Slave	WSL_S2_Slave
Handshaking	:	None	None
Carrier Detect	:	No	No
Port Enabled	:	Enabled	Enabled
Data Transfer Rate	:	To suit office main port.	To suit office standby port
Broadcast Wind	:	0	6
Housing Address	:	To suit office	

2.6.8.5 Temporary Local & Permanent Remote Moviolaw Connections

In this example, a local MoviolaW installations is connected directly to a diagnostic serial port of the NVC/DM, only when required.

In addition, a permanent remote MoviolaW installation is connected using a PCM interface.

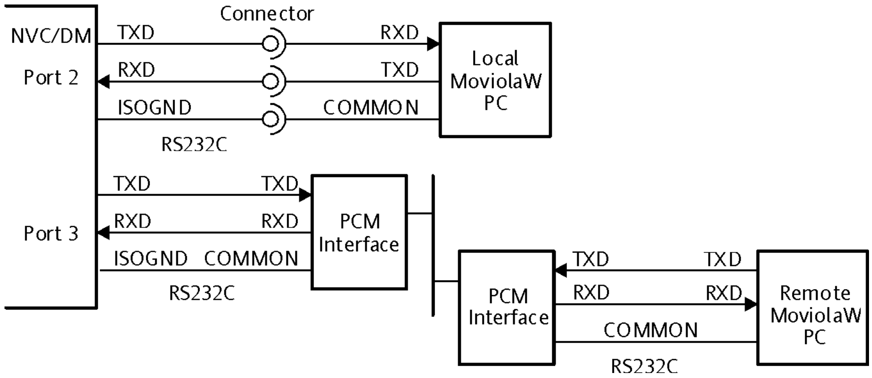


Figure 2.41 MoviolaW connections—temporary local and permanent remote—NVC/DM

The NVC/DM Port configurations are:

		Port 2	**Port 3**
Port Type	:	Serial	serial
Protocol	:	Diagnostic	Diagnostic
Connection Type	:	Temporary	Permanent
Handshaking	:	None	None
Carrier Detect	:	No	No
Port Enabled	:	Enabled	Enabled
Data Transfer Rate	:	To suit local MoviolaW	To suit PCM interface and remote MoviolaW
Port Address	:	To suit local MoviolaW	To suit remote MoviolaW

2.6.8.6 Multi-dropped NVC/DM and MoviolaW connection

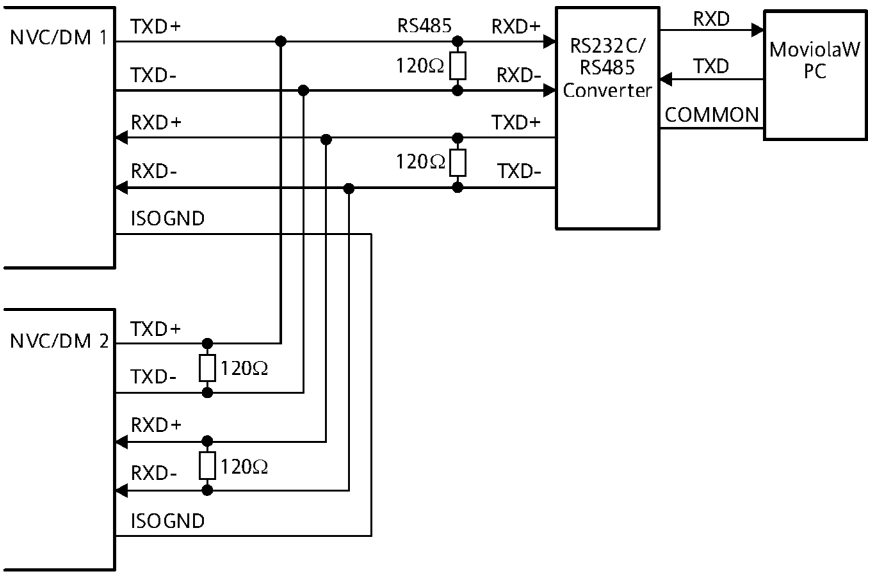


Figure 2.42 MoviolaW connection—remote—NVC/DM

The NVC/DM Port configuration is:

Port Type	:	Serial
Protocol	:	Diagnostic
Connection Type	:	Permanent
Handshaking	:	None
Carrier Detect	:	No
Port Enabled	:	Transmitting
Data Transfer Rate	:	To suit MoviolaW
Port Address	:	Must be different for each NVC/DM

2.6.8.7 MoviolaW Dial-up Connection

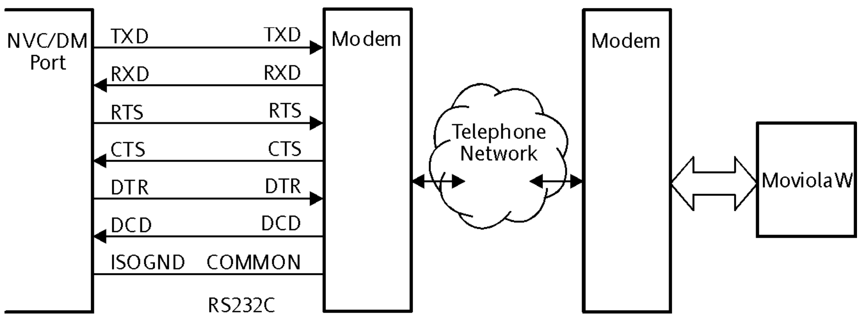


Figure 2.43 MoviolaW connection—dial-in dial-out—NVC/DM

The NVC/DM Port configuration is:

Port Type	:	Serial

Protocol	:	Diagnostic
Connection Type	:	Dial-in, dial-out or dial in and out
Handshaking	:	RTS/CTS
Carrier Detect	:	Yes
Port Enabled	:	Enabled
Data Transfer Rate	:	To suit MoviolaW (see below)
Port Address	:	To suit MoviolaW
Phone Numbers		To suit MoviolaW

To maximise data throughput, the NVC/DM port data transfer rate should normally be configured to be at least as fast as the maximum modem connect speed, but should not exceed the configured MoviolaW data transfer rate. For a 28–56 kbps modem, 57600 bps is an appropriate setting.

Note: ***A standard PC to modem cable should not be used to connect between the NVC/DM port and the modem. Doing so will connect the DSR output on the modem to the CLKOUT output on the NVC/DM.***

2.7 GCSS Checks and Constraints

The GCSS checks the consistency of data. Checks and cross-references are performed to ensure that:

- All entities defined in the modules are used in the ladder logic;
- All entities defined in the ladder logic have appropriate modules;
- The whole installation represents a consistent set of information that can be compiled.

Specific tests are performed on each type of installation.

Note: ***Installations with consistency check warnings will pass the consistency check and can be manually checked and approved. If warnings are not desirable, inconsistencies must be fixed prior to approval.***

2.7.1 VLM Consistency Checks

The GCSS performs the following consistency checks on a VLM installation and records any discrepancies in the Consistency Check Report.

Consistent names—all names (logic states) defined in the modules and ladder logic are cross referenced to ensure they represent a consistent set. Any names defined but not used or vice versa are reported.

Initial states—all directly mapped logic states must have the same initial state, ie if an input is directly mapped to an output, by assigning the same mnemonic name to both, then they both must have the same initial state.

Module sequence numbers—all module sequence numbers must be in the range 1 to 99. Duplicate module sequence numbers are not allowed.

Vital Modules—a VLM installation must contain at least two vital modules (eg VPIM, VROM etc) in addition to the VLM.

Module I/O data—each I/O module must have at least one input or one output defined (with the exception of a paired NVC or CNVC module).

NVC or CNVC pair:

- An installation can have only one set of paired NVC or CNVC modules.
- An installation cannot have an NVC pair and a CNVC pair.
- NVC and CNVC modules can only be paired with modules of the same type.
- The pair must occupy the two highest allocated Module Sequence Numbers in the installation.
- The paired module with the lower Module Sequence Number must not have any I/O data defined.

The GCSS checks to ensure the housings contain either zero or exactly two paired NVC or CNVC modules and if present, the one with the higher Module Sequence Number has defined I/O data and the other does not.

VLM6 and NCDM Pair—an installation using a VLM6 may or may not also have an NCDM. When an NCDM is present, the GCSS checks to ensure the NCDM is installed adjacent to the VLM6.

Allowed modules for logic module type—certain modules can only be used with certain logic modules. See section 4.2.3.2.

Total number of modules for logic module type—number of modules must not exceed 57. In addition, the number of modules in each module category must not exceed its allowed limit. See section 4.2.3.2.

Complete rungs—all ladder rungs must be complete and consistent.

Maximum number of defined names—(or logic states) that can be defined for the logic module. See table 2.4.

Header information—must define an installation name, installation address and customer name.

Port address—only 8 serial communications modules are allowed within an installation and each one must have a unique port address in the range 0 to 7.

Duplicate output names—modules must not have two or more outputs with the same name. *This check is classified as a warning*.

2.7.2 NVLM Consistency Checks

The GCSS performs the following consistency checks on an NVLM installation and records any discrepancies in the Consistency Check Report.

Consistent Names (Logic States)—all names defined within an installation must form a valid set of:

- inputs;
- outputs;
- latches;
- SR Latches;
- timers;
- Time-of-Day timers.

An input to an SR Latch must be one of:

- another SR Latch output;
- a coil;
- user timer output;
- read-only reserved timer output;
- read-only reserved latch;
- read-write reserved latch;
- port input.

An SR Latch output must be one of:
- an SR Latch input;
- contacts;
- port output;
- user timer trigger.

All names defined in the ladder logic are cross referenced to ensure they represent a consistent set. Any names defined but not used or vice versa are reported. For example: All coils in a ladder must be defined as a latch or an output with the NVLM module.

Maximum number of names (logic states)—must not exceed the value shown in table 2.5.

Complete rungs—all rungs in the ladder must be complete and consistent.

Initial states—all directly mapped logic states must have the same initial state, ie if an input is directly mapped to an output, by assigning the same mnemonic name to both, then they both must have the same initial state.

Duplicate output names—modules must not have two or more outputs with the same name. *This check is classified as a warning*.

2.8 GCSS Compilation and Decompilation Support

The GCSS compiles installation data into Vital PROM Data for vital logic modules and a Non-Vital Configuration for non-vital logic modules. Vital PROM Data are downloaded to a PROM Programmer. A Non-Vital Configuration is downloaded directly to the non-vital logic module.

The GCSS can also upload and decompile Vital PROM Data and a Non-Vital Configuration from the PROM Programmer (Vital PROM Data), or from the actual interlocking through the Diagnostic Module using the Installation Check System (ICS) application.

2.9 GCSS Checking and Approval Control

Figure 2.44 illustrates the basic checking and approval process within the GCSS. It also illustrates how the companion Installation Check System (ICS) should be used to verify that the actual WESTRACE equipment contains the intended data.

Note:

The GCSS can perform all of the uploading and comparison functions that the ICS can perform but they differ as follows:
- ***The GCSS uploads from the PROM programmer or from the NVLM module.***
- ***The ICS uploads from the diagnostic module while it is installed in the WESTRACE housing.***

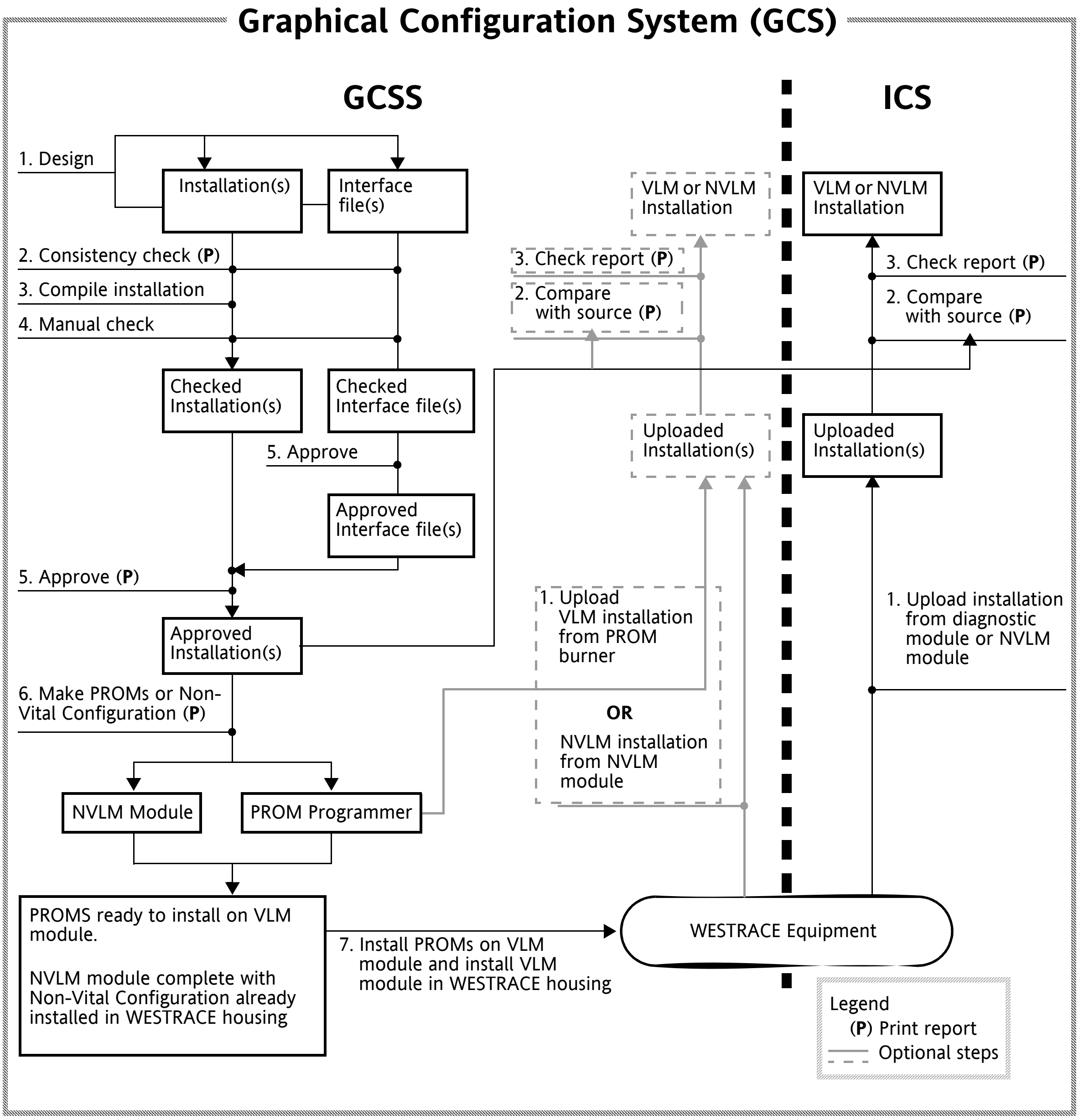


Figure 2.44 Checking and approval process

Note: ***Interface files must be approved before associated installation files can be approved.***

2.10 GCSS Version Control—Installations and Interface Files

Installation development typically involves a number of minor versions before the final version is checked and approved. The GCSS allows saving incremental versions of installation and automatic recording of status change.

2.10.1 Overview

All installations have an associated Log File, and may also have an associated interface file.

- **Log files**—are archive files created or updated whenever any command from the Configuration menu is used. They contain the version history plus all prior versions of the installation or interface file in a compressed format.
- **Installation files**—contain a single version of an installation plus version history. They are created or updated every time the GCSS **Save** or **Save As** commands are used.
- **VLM-NVLM Interface files**—contain information that is shared by VLM and NVLM installations. They contain a single version of the interface data plus version history.
- **VLM6-VLM6 Interface files**—contain information that is shared by the VLM6 modules of two separate WESTRACE applications. They contain a single version of the interface data plus version history.

Version history records are written to these files whenever configuration commands are used. The **Save** and **Save As** commands do not add version records.

Note: ***Log files are automatically saved to the same directory as the installation file.***

Identifying Installation, Log and Interface files.

	Filenames and Extensions	
Type of Installation	**Installation File**	**Log File**
VLM	*xxxx*.INS	*xxxx*.LOG
NVC/DM	*yyyy*.NVC	*yyyy*_ncvdm.LOG
NCDM	*zzzz*.NCD	*zzzz*_ncdm.LOG

Table 2.28 File naming conventions—Installation files

	Filenames and Extensions	
Type of Interface File	**Interface File**	**Log File**
VLM-NVLM	*aaaa*.IFC	*aaaa*.ILG
VLM6-VLM6	*bbbb*.IFC	*bbbb*.ILG

Table 2.29 File naming conventions—Interface files

2.10.2 Creating Versions

Adding a design-modification, check, or approval record to an installation changes the version of the installation file. The GCSS adds a compressed copy of the installation file to its log file and deletes the compiled Vital PROM Data or Non-Vital Configuration (if one exists). The GCSS also adds a design-modification record to all of an installation's interface files that are in the "MODIFIED" state.

This ensures:

- all prior versions can be recovered for future reference;
- the correct versions of associated interface files are recovered for a given installation file;
- Vital PROM Data or Non-Vital Configuration cannot be made without recompiling the installation.

Similarly, whenever a design-modification, check, or approval record is added to an interface file, the file is saved with the record stored within and a compressed copy of the interface file is stored in the associated log file.

The development of installation and interface files pass through these stages:

a) Add Design-modification Record (minor versions 0.n).

b) Add Checked Record (minor versions 0.n).

c) Add Approval Record (major version n.0).

As a minimum, there must be at least one of each record for a major version.

2.10.3 Modification, Checking, and Approval

Use the Configuration menu or toolbar buttons for:

- adding a design-modification record;
- adding a checked record;
- adding an approval.

These menu items and toolbar buttons open a dialog box similar to those shown in figure 2.45.

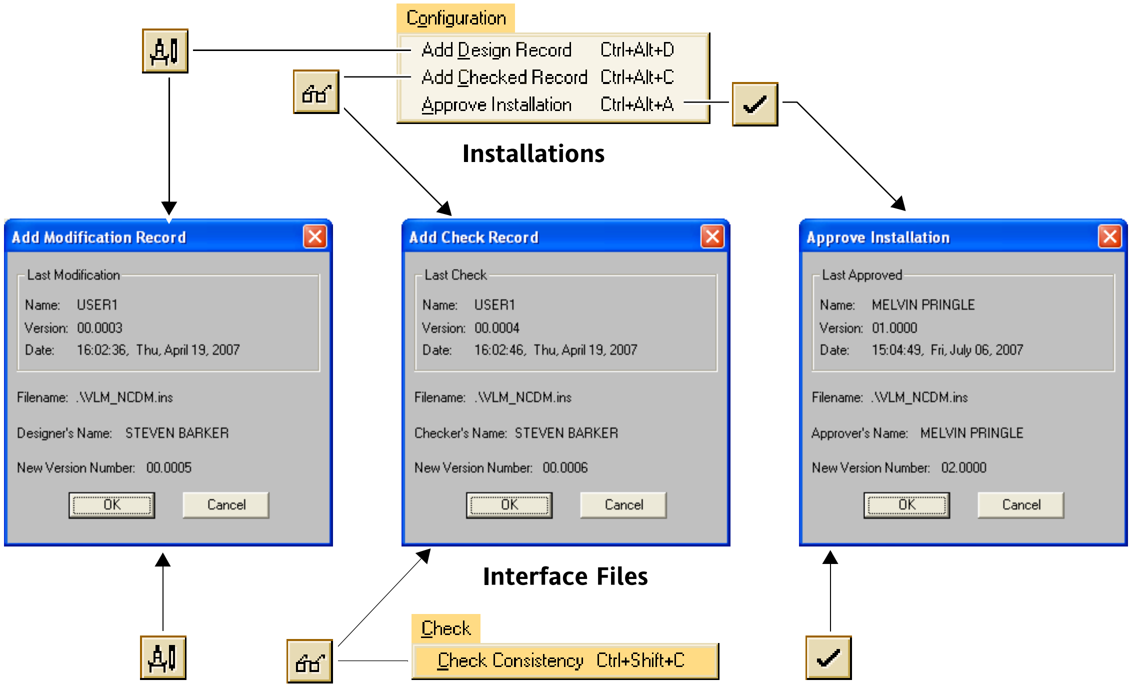


Figure 2.45 Configuration dialog boxes

The availability of these commands is determined by the current status of the installation or interface file. See table 3.2 on page 3-14.

Selecting the **OK** button closes the dialog box and automatically updates the installation file or interface file and the appropriate log file.

Note: ***The GCSS automatically creates a new log file when one is not present in the same directory as the installation or interface file.***

3. STARTING

This chapter gets you started with the GCSS and provides some guidance for the tasks to be performed when developing an installation.
Section 1.10.2 provides a detailed outline of all steps in the design process.

Note: ***The Release Notes that came with the GCSS software should also be read and understood.***

These notes contain:
- ***the latest changes not reflected in this manual;***
- ***any known problems with recommended work-arounds;***
- ***any other information that may affect successful operation of the GCSS.***

3.1 The GCSS User Interface

The GCSS has a Multi Document Interface (MDI) which means that more than one installation can be open at any one time.

The basic structure of the GCSS User Interface comprises:

- a main window;
- one or more child windows.

Figure 3.1 illustrates the GCSS main window with three child windows (one of which has been minimised) and a dialog box.

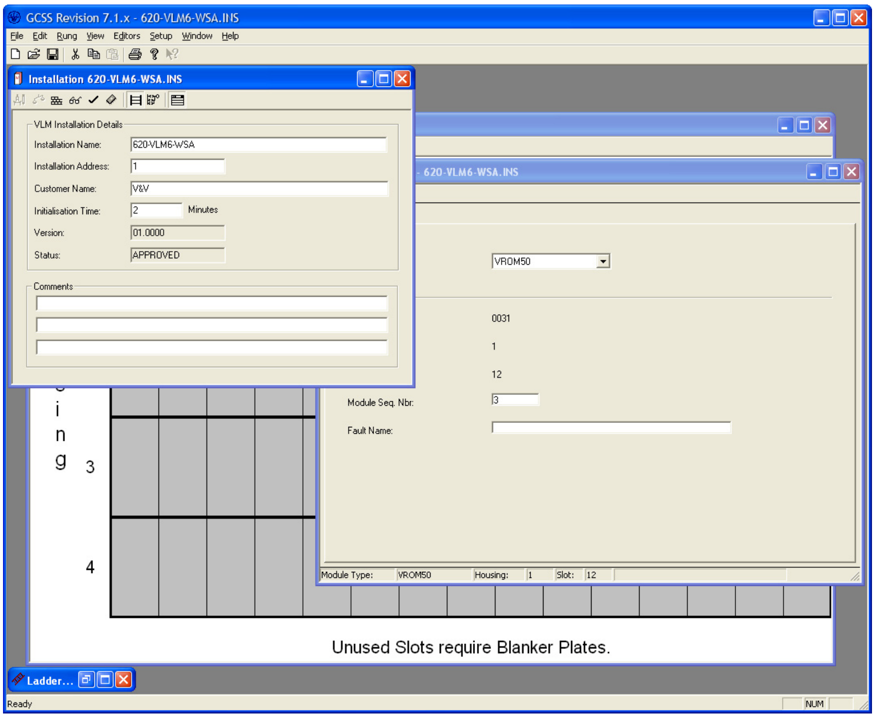


Figure 3.1 GCSS main window with child windows

3.1.1 GCSS Child Windows

Every open installation (there can be more than one) is displayed in an Installation Window. Other child windows can be opened from each Installation window including:

- Housing Editor—described in Chapter 4
- Module Editor—described in Chapter 5
- Ladder Editor—described in Chapter 6
- Rung Viewer—described in Chapter 6
- Rung Editor—described in Chapter 6
- Report Window—described in Appendix B

See section 1.8.2 and Appendix A for basic assistance with the GCSS User Interface.

Note: ***Only the Installation window has to be open for any particular installation.***

3.2 Starting the Software

Do one of the following:

- Select the GCSS icon on the computer desktop.
- Select GCSS from the Windows Start menu.
- Open the GCSS folder in Windows Explorer and select the GCSS.exe icon.

The main GCSS window opens to a full screen with the Copyright Panel in the centre.

Note: ***Disable screen savers before starting the GCSS.***

3.2.1 GCSS Self Check

The GCSS generates a checksum of the GCSS executable file at startup and compares it with the encrypted checksum stored in `GCSS.DAT`.

The GCSS will shutdown if:

- the checksums do not match;
- the `GCSS.DAT` file is corrupted;
- the `GCSS.DAT` file is not present.

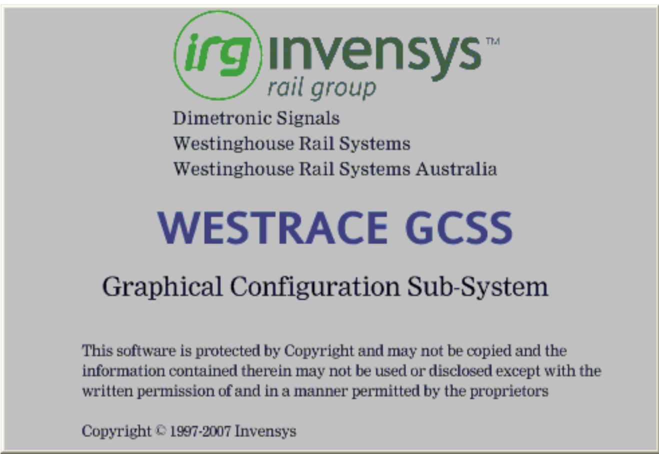


Figure 3.2 Copyright screen

This screen closes automatically after 15 seconds, or you can close it immediately by:

- pressing any key, or;
- clicking the left mouse button;

The Login dialog box then appears.

3.2.2 Login Dialog Box

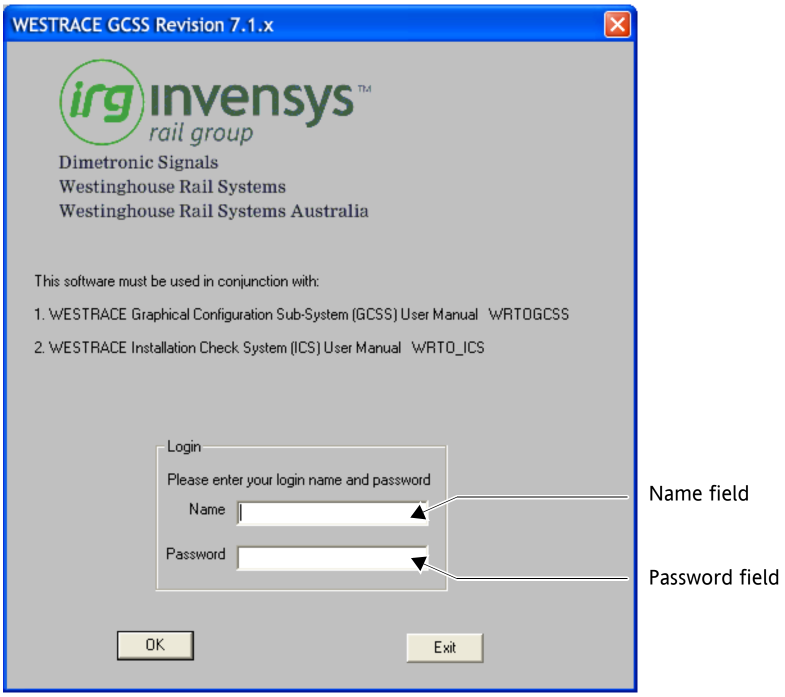


Figure 3.3 Login dialog box

a) Type your "user name" into the Name field.

b) Press the **Tab** key to move the insertion point to the Password field (or click on the Password entry field with the mouse).

c) Type your password and press the **Enter** key (or click the **OK** button).

An error in your user name or password is reported by the alert message box (figure 3.4).

Figure 3.4 Password file error warning

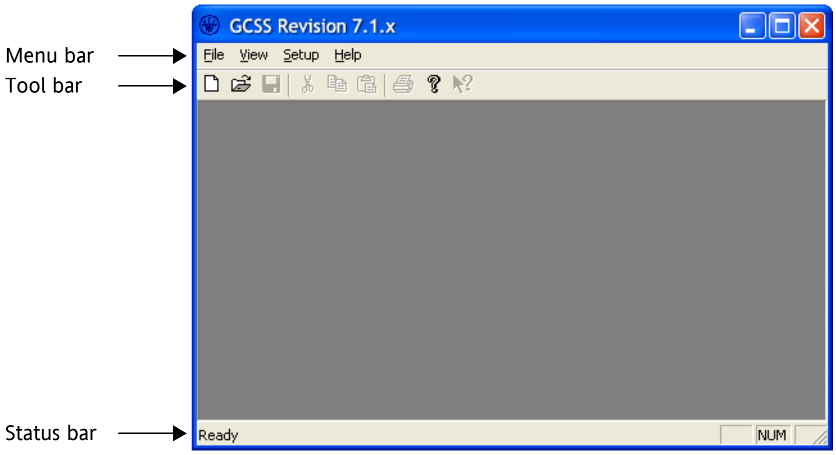
Note: ***Your user name and password must already be registered with the GCSS. Speak to your GCSS Supervisor if you cannot log in. See Chapter 10.***

The GCSS may ask you to enter a Compatibility Index—see section 3.4.

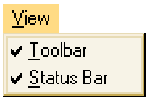
Exiting the GCSS Without Logging In

Click the **Exit** button.

3.2.3 GCSS Main Window



The tool and status bars can be toggled on or off through the View menu.



See Appendix A if you need more assistance with this window.

3.3 Changing Your Password

Select **Change Password** to open the Change GCSS User Password dialog box.

Alt s p
Ctl+Alt+p

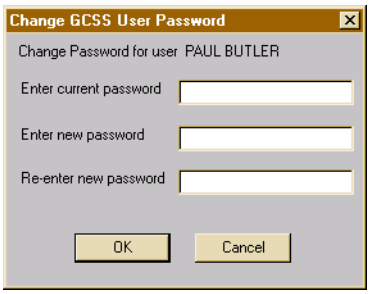


Figure 3.5 Change User Password dialog box

The labels on each field are self explanatory, but:

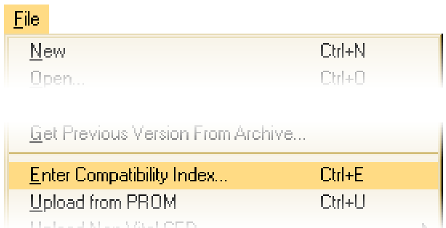
- what you type in a field will not be displayed;
- a password can contain numerals and uppercase and lowercase password length may be 1 to 15 characters.

3.4 Compatibility Index

The Compatibility Index number is used to ensure that each vital module and the application data are fully compatible with each other. You must use a Compatibility Index number in GCSS appropriate to the WESTRACE hardware. See table 3.1, or contact your Invensys Rail supplier for this number.

The GCSS must always have a compatibility index value stored in the GCSS *ini* file. The compatibility index dialog box is automatically displayed by the GCSS (when started) if this information is missing or incorrect. The **Cancel** button on the dialog box is disabled to enforce selection of a compatibility index value. This dialog box is also displayed when the GCSS is first started after installation.

a) Select **Enter Compatibility Index** to open the Enter Compatibility Index dialog box.



Alt f e
Ctrl+e

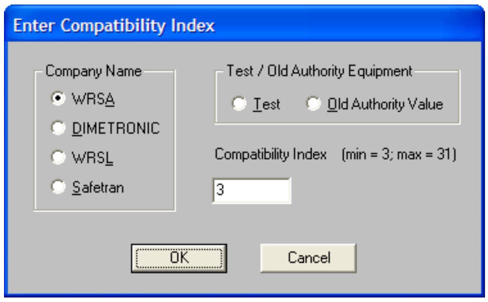


Figure 3.6 Enter Compatibility Index dialog box

b) Select:
- The **Test** button if the installation is intended for test equipment. This sets the compatibility index to zero.
- The **Old Authority Value** to set the compatibility index to 1.
- Your company, and enter the appropriate compatibility index value in the field provided (see table 3.1).

c) Click the **OK** button.

Company	CI Used (as at June 2005)	CI Range
WRSA	3	3–31
DIMETRONIC	32	32–63
WRSL	64	64–95
Safetran	96	96–127

Table 3.1 Compatibility Index (CI) values

SAFETY
WARNING

Never use Compatibility Index 0 for a system used in service. (0 is reserved for testing.)

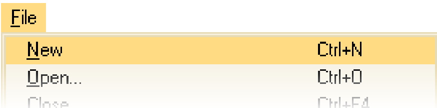
3.5 Accessing Installations

This section describes the options for accessing and saving installations. Additional references are provided as appropriate.

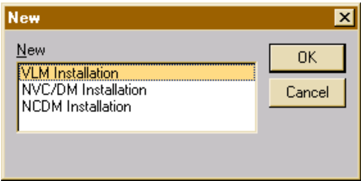
This manual uses the word “installation” to define the functionality associated with VLM (ie. HVLM128, HVLM128a, VLM5 or VLM6) and NVLM (ie. NCDM or NVC/DM) modules. Do not confuse this with a physical installation (called an “application” in this manual) that may contain both a VLM and one of the NVLM modules.

3.5.1 Creating New Installations

Select **New** and then select the type of installation to be created from the dialog box.



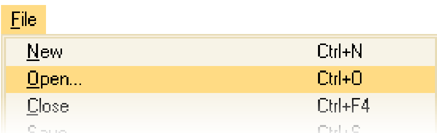
Alt f n
Ctrl+n



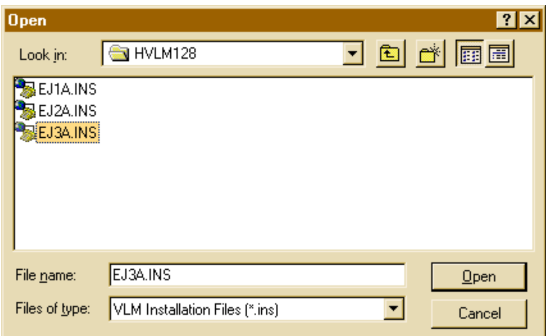
This action creates an empty, unnamed installation and opens the appropriate installation window. See section 3.6.

3.5.2 Opening Existing Installations

a) Select **Open** to display the Open dialog box.



Alt f o
Ctrl+o



b) Select the desired type of installation from the drop down list in the "Files of Type" field:
   - VLM Installation Files (*.ins) for VLM installations;
   - NVC/DM Installation Files (*.nvc) for NVC/DM installations;
   - NCDM Installation Files (*.ncd) for NCDM installations;
   - GCS Installation Files (*.ins, *.nvc, *.ncd) for all installation files.

c) Select the desired installation from the dialog box to display its installation window. See section 3.6.

   The GCSS may ask you to change the compatibility index. See section 3.4.

3.5.2.1 File Locking

This prevents two people simultaneously opening an installation. The File Locked dialog box shown in figure 3.7 is displayed when:

- the file is open on another GCSS workstation;
- GCSS has failed while the file was open.

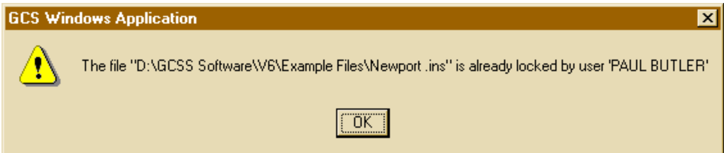


Figure 3.7 File Locked dialog box

Note: ***Section 10.5.5 explains how to unlock a file that is not being edited.***

3.5.2.2 Authentication Code

This is a checksum of all the information in the installation file and is embedded within the installation file. A new checksum is generated and compared to the embedded value when an installation is opened.

The GCSS will not allow the installation file to remain open if the two checksums do not match.

Note: ***GCSS data files can be viewed using an ASCII text editor but NEVER manually edit the file.***

3.5.2.3 Removing Unsupported GPOM Variants

The GCSS deletes unsupported GPOM variants when opening an installation. It displays an alert message for each deleted GPOM showing the unsupported GPOM type, its housing number and its slot number.

3.5.2.4 Installations Created by Early Versions of GCSS

See section 1.11, "Backwards Compatibility" on page 1-14.

Refer to the Release Notes or contact your Invensys Rail supplier for advice about converting old installation files.

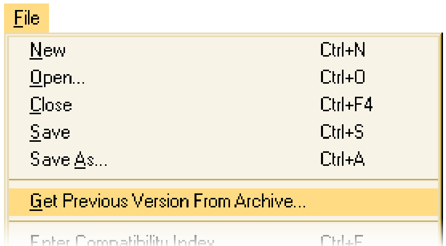
3.5.3 Previous Versions of an Installation

3.5.3.1 Recovering a Previous Version of an Installation

Previous versions of an installation can be recovered from its Log file. See section 2.10 about creating and archiving versions of an installation.

a) Open the current version of the desired installation. See section 3.5.2.

b) Select **Get Previous Version From Archive** to open the Previous Versions dialog box.



Alt f g

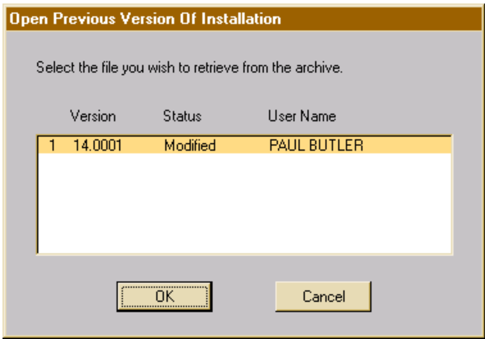


Figure 3.8 Previous Version dialog box

c) Select the required version from the box and select the **OK** button.

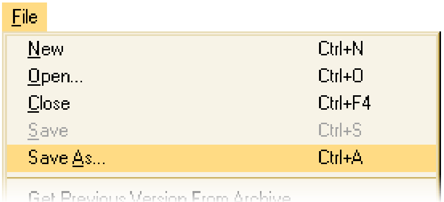
The selected file is opened as an installation with a title that indicates it is a earlier version by showing the version number.

The previous version can be viewed or printed but the following operations cannot be performed:

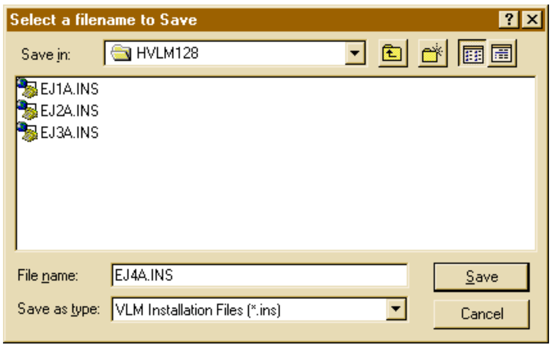
- Compile
- Make PROMs
- Save
- Add design-modification records
- Add check records
- Approve

3.5.3.2 Creating a New Installation From a Previous Version

a) Select **Save As** from the File menu to open the Windows Save As dialog box.



Alt f a
Ctrl+a



b) Save the file to a new installation file with a new name.

c) Modify as required.

3.6 Installation Window

The Installation window opened by the GCSS is appropriate to the type of installation, VLM or NVLM (see section 1.8.1 for a definition). The visual differences between the types of installation window are limited to the installation name and the colour of the toolbar buttons (VLM are black, HVLM are green).

The labels applied to the data fields are identical and accept the same type of information.

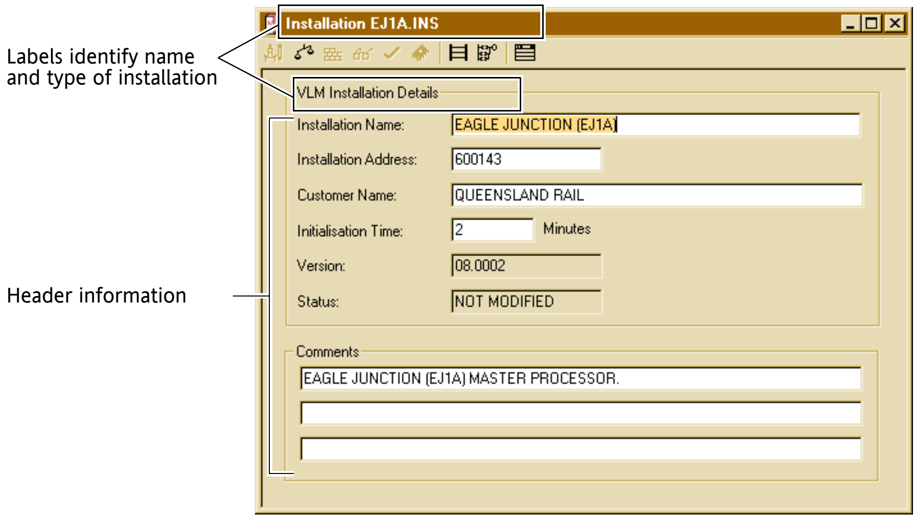


Figure 3.9 Installation window—VLM Installation

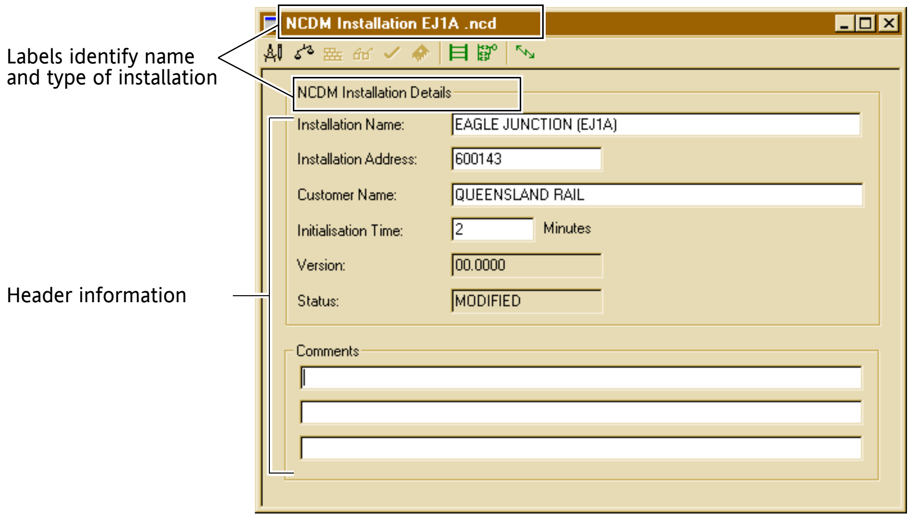


Figure 3.10 Installation window—HVLM Installation

a) **Installation Name**—a name that clearly and uniquely describes the project.

b) **Installation Address**—the unique installation address. Must be an integer. Must be obtained from Invensys Rail because the address must be unique world wide. Contact your nearest Invensys Rail office for an installation address allocation.

The same installation address **must** be used by associated VLM and NVLM installations within the same WESTRACE application. Different WESTRACE applications **must** use different addresses.

c) **Customer Name**—the name of the end user railway.

d) **Initialisation Time**—the required Application Delay in minutes (see section 2.6.1). Must be an integer.

A change in this value while the VLM module editor is open will not be reflected in the module editor until it is closed and opened again.

When a VLM is changed (see section 4.2.5), the duration for APPDEL in the VLM module editor will revert to the default value.

You must:
- close the VLM module editor (if open);
- change the Initialisation Time value to another, then change it back to the desired value;
- open the VLM module editor and confirm the APPDEL duration is as desired.

e) **Version**—shows the current version of the installation.

f) **Status**—shows the current position of the installation in the development cycle. See section 3.6.3.

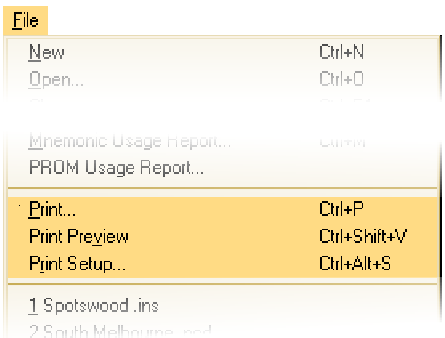
g) **Comments**—optional. Enter up to three lines of helpful information.

3.6.1 Installation Report

This single-page printed report contains the installation's header information, and address and configuration link jumper details.

a) Select **Print Setup** and change the paper orientation or select another printer if necessary.

b) Select **Print** or **Print Preview** to generate the report.



Alt f r
Ctrl+Alt+s

Alt f p
Ctrl+p

Alt f v
Ctrl+Shift+v

The report can be sent to the printer after previewing in on screen.

See Appendix B, section B.2 for further assistance. See section B.5.1 for an example.

3.6.2 Installation Window Toolbar

Every installation window has a button toolbar for quick activation of certain operations that are performed on the entire installation.

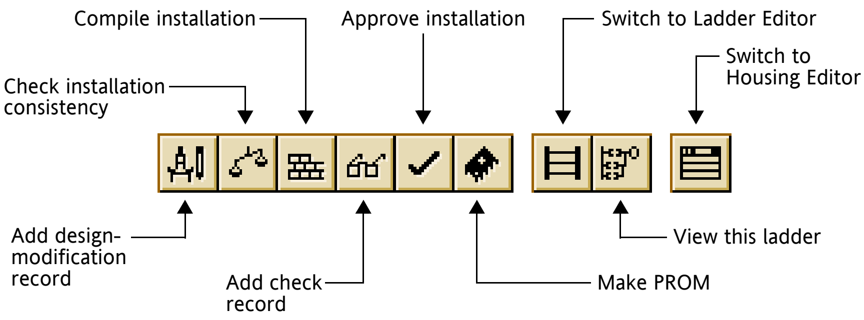


Figure 3.11 Toolbar—VLM Installations (black icons)

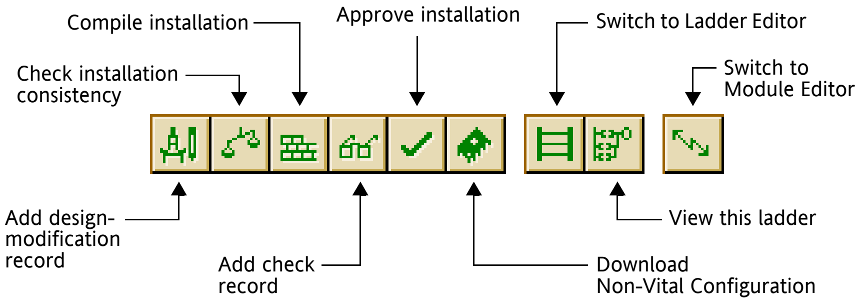


Figure 3.12 Toolbar—NVLM Installations (green icons)

Use these buttons to activate the denoted operations.

The operations can also be activated from the main menu (when the installation window is active).

Note: ***The wording of menu items may not exactly match the button labels in figures 3.11 and 3.12.***

3.6.3 State of Installations and Interface Files and Allowed GCSS Operations

Installations and interface files pass through a number of states as they go from creation (or modification) to approval. The state of the installation or interface file determines the operations that can be performed and therefore the enabled toolbar buttons and menu items. Unlisted operations can be performed at any time but will usually cause the status to revert to “modified”.

Installation Files

State of Installation	GCSS Operations					
	Add Design Modification Record	Check Installation Consistency	Compile Installation	Add Check Record	Add Approval Record	Make PROM or Download Non-Vital Configuration
New	Disabled[1]	Enabled	Disabled	Disabled	Disabled	Disabled
Modified	Enabled[2]	Enabled	Disabled	Disabled	Disabled	Disabled
Consistency Checked	[3]	Enabled	Enabled	Disabled	Disabled	Disabled
Compiled	[4]	Enabled	Enabled	Enabled	Disabled	Disabled
Checked	Disabled	Enabled	Enabled	Enabled	Enabled[5]	Disabled
Approved	Disabled	Enabled	Enabled	Enabled	Enabled	Enabled[6]

Table 3.2 State of an Installation and allowed GCSS Operations

1 A new installation by definition cannot be a modified version of a previously saved installation. Therefore a new installation cannot have a Modification Record added to it.

2 An installation opened after being previously saved in a modified state but without a corresponding Modification Record, is opened in the Modified state but has not yet been Modification Record marked.

3 A Modification Record is requested after a modified installation passes a full Consistency Check but has not yet been Modification Record marked.

4 A Modification Record can be added in this state if the installation is modified but a Modification Record was not added at the end of the successful full Consistency Check.

5 An installation cannot be approved until ALL (if any) attached interface files have been approved.

6 Adding any configuration control record changes the installation's version information. The GCSS deletes any existing compiled PROM data or Non-Vital Configuration to ensure the installation is recompiled before making a PROM or downloading the Non-Vital Configuration.

Interface Files

State of Interface File	GCSS Operations			
	Add Design Modification Record	Check Consistency	Add Check Record	Add Approval Record
New	Disabled[1]	Enabled	Disabled	Disabled
Modified	Enabled[2]	Enabled	Disabled	Disabled
Consistency Checked	Enabled	Enabled	Disabled	Disabled
Checked	Disabled	Enabled	Enabled	Enabled
Approved	Disabled	Enabled	Enabled	Enabled

Table 3.3 State of an interface file and allowed GCSS Operations

1 A new interface file by definition cannot be a modified version of a previously saved interface file. Therefore a new interface file cannot have a Modification Record added to it.

2 An interface file opened after being previously saved in a modified state but without a corresponding Modification Record, is opened in the Modified state but has not yet been Modification Record marked.

3.6.4 Who Can Perform GCSS Operations

The GCSS recognises users by Logon ID and password so that:

- only authorised users can access and operate the GCSS;
- manual check records and approval records for installations cannot be entered by the same user.
  The latter restriction does not apply to interface files.

Tables 3.4 and 3.5 summarize who can perform GCSS operations on installations and interface files.

Installations

Authorised Users (Application Engineers)	GCSS Operation						
	Develop Design	Add Design Modification Record	Perform Consistency Check	Compile Installation	Add Manual Check Record	Add Approval Record	Make PROM or Download to NVLM
Engineer 1	✓	✓	✓	✓	1		
Engineer 2					✓	1	
Engineer 3						✓	
Any valid user							✓

Table 3.4 Relationship between GCSS operations on Installations and authorised users

1 Engineer 2 can add the Approval Record provided Engineer 1 added the Manual Check Record. The same person cannot add both the Manual Check and Approval records.

Interface Files

Authorised Users (Application Engineers)	GCSS Operation				
	Develop Design	Add Design Modification Record	Perform Consistency Check	Add Manual Check Record	Add Approval Record
Engineer 1	✓	✓	✓	✓	✓

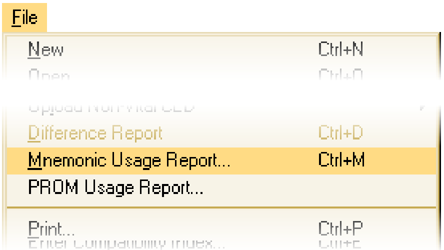
Table 3.5 Relationship between GCSS operations on interface files and authorised users

3.7 Mnemonic Usage

Large installations (both VLM and NVLM) will use many mnemonics. To keep track of the mnemonics and to ensure all required are present and properly utilised, generate the Mnemonic Usage Report. The report is written to the Report Window, where it can be viewed and printed.

a) Select the desired installation to make it active.

b) Select **Mnemonic Usage Report** to display the dialog box.



Alt f m
Ctrl+m

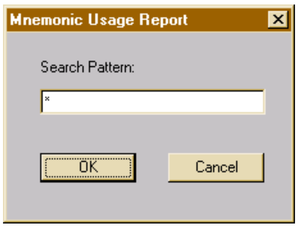


Figure 3.13 Mnemonic Usage Report dialog box

c) Define the mnemonics to be included in the report by entering a search pattern.

The Search Pattern field accepts an ASCII character string which can be the exact mnemonic name required or it can include the following wild card characters:

* Use alone to include all mnemonics in the report.

Insert specific characters before the * to find mnemonics that begin with those characters and have any number of characters from the * position to the end of the string. For example, ABC* will find any mnemonic that begins with ABC.

? Use to match any character in a particular position.

For example: A?B means find any mnemonic that begins with A, has any character in the second position, and ends with B.

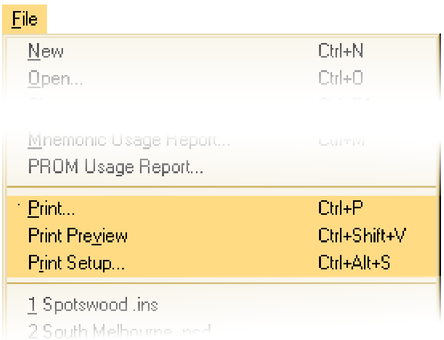
The report can be printed directly from the Report Window using the current printer settings.

Note: ***The search pattern is not case-sensitive.***

You can change the paper orientation or change the current printer.

a) Select **Print Setup** and change the paper orientation or select another printer if necessary.

b) Select **Print** or **Print Preview** to generate the report.



Alt f r
Ctrl+Alt+s

Alt f p
Ctrl+p

Alt f v
Ctrl+Shft+v

The report can be sent to the printer after previewing in on screen.

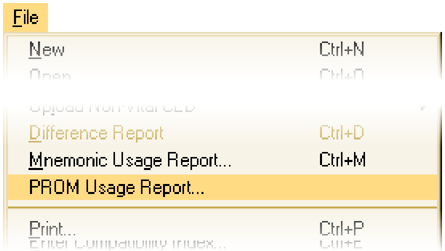
See Appendix B, section B.2 for further assistance. See section B.5.3 for an example.

3.8 PROM or Non-Vital CED Usage

Generate the PROM or Non-Vital CED Usage Report to get statistics on the Vital PROM Data or NVLM Non-Vital Configuration contents. These reports are written to the Report Window, where they can be viewed and printed.

a) Select the desired installation to make it active.

b) Select **PROM Usage Report** or **Non-Vital CED Usage Report** to display the report.



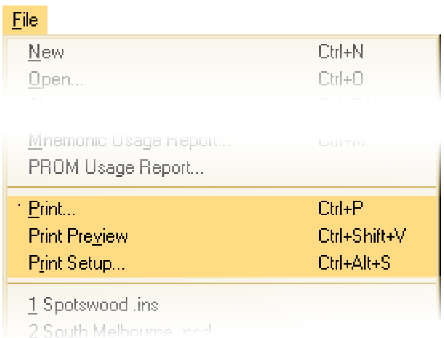
The report can be printed directly from the Report Window using the current printer settings.

You can change the paper orientation or change the current printer.

a) Select **Print Setup** to change the paper orientation or select another printer.

b) Select **Print** or **Print Preview** to generate the report.

Alt f r
Ctrl+Alt+s



Alt f v
Ctrl+p

Alt f v
Ctrl+Shft+v

The report can be sent to the printer after previewing in on screen.

See Appendix B, section B.2 for further assistance. See section B.5.4 for an example.

3.9 Comparing Installations (Difference Report)

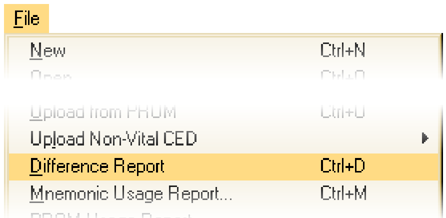
Typically, you use the GCSS Difference Report to:

- compare the current installation with a previous version after making changes check that:
  - all intended changes have been made;
  - no unintended changes have been made.
- compare an installation with another outwardly similar installation to determine the differences.

The Difference Report is displayed in the Report Window where it can be viewed or printed as required.

a) In the same GCSS session, open two installations, or two versions of the same installation.

b) Select **Difference Report** to open the Installation Difference Report dialog box. The currently active installation is the subject of comparison.



Alt f d
Ctrl+d

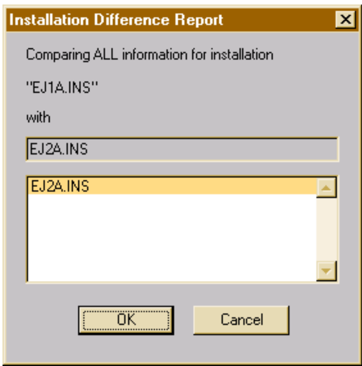


Figure 3.14 Installation Differences Report dialog box

c) Select the other installation for the comparison and click the **OK** button to display the report in the Report Window.

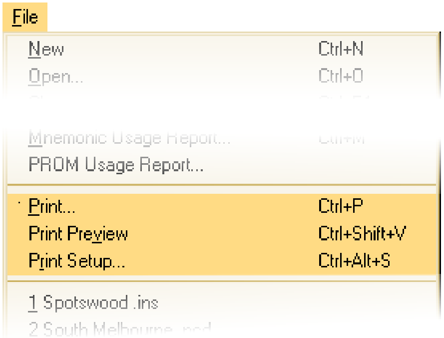
The report can be printed directly from the Report Window using the current printer settings.

Note: ***Only the first difference detected within rungs is reported.***

You can change the paper orientation or change the current printer.

a) Select **Print Setup** to change the paper orientation or select another printer.

b) Select **Print** or **Print Preview** to generate the report.



Alt f r
Ctrl+Alt+s

Alt f p
Ctrl+p

Alt f v
Ctrl+Shft+v

The report can be sent to the printer after previewing in on screen.

See Appendix B, section B.2 for further assistance. See section B.5.2 for an example.

3.10 Saving Installations

There are two possibilities:

- Saving a compressed copy in a log file for convenient archiving, version control and possible future recovery;
- Saving to normal file storage (eg hard disk).

3.10.1 Saving Archived Versions

A archived copy of an installation file, including a new version record, can be saved at any time. Any associated Interface file has a version record added but the file is not archived.

a) Select **Add Design Record** to open the Modification Record dialog box.

Alt o d
Ctrl+Alt+d

b) Select the **OK** button.

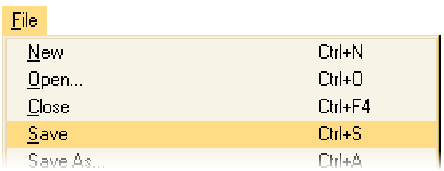
A design-modification record is added to the installation file (and all associated interface files) and the file(s) is saved. A compressed copy of the installation file is added to the Log file and the Log file is saved.

3.10.2 Normal File Saving

3.10.2.1 Same Name

A new or opened installation (not an opened archive) can be saved at any time.

Select **Save**.



Alt f s
Ctrl+s

The Save As dialog box is used for a new installation. See section A.2.2.2.

3.10.2.2 Another Name or Directory

An existing installation (including an archive) can be saved to the same directory with a different file name, or to another directory with the same or different file name.

Select **Save as** to display the Save As dialog box.

Alt f a
Ctrl+a

The status of the new installation will not be the same as the status of the original installation. It will drop to a lower status.

See section A.2.2.2 if you need assistance with the SaveAs dialog box.

3.11 Exiting the GCSS

Select **Exit**.

Alt f x
Alt+F4

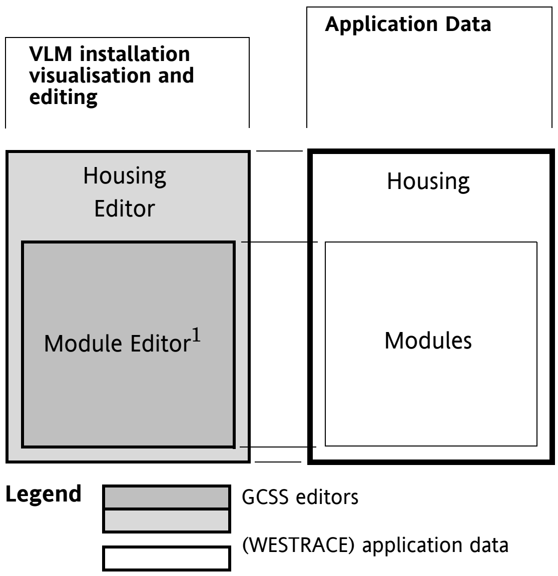
This is the preferred way to quit the GCSS program. It guarantees all files are properly closed before the program terminates.

4. Housings and Modules

This chapter describes the Housing Editor and how it is used to build a VLM installation.

4.1 Overview

Use the Housing Editor to allocate modules to the housings of a VLM installation. Once a module is in place, its parameters can be edited by invoking the Module Editor (see Chapter 5).



1. Module Editor is started from within the Housing editor (see Chapter 5).

Figure 4.1 Housing Editor—applied to Application Data

4.2 Housing Editor

4.2.1 Housing Editor Activation

a) Select the VLM Installation window to make it active.

b) Select **Housing Editor**.

Alt d h
Ctrl+h

4.2.2 Housing Editor Window

Figure 4.2 shows a typical Housing Editor window with modules added.

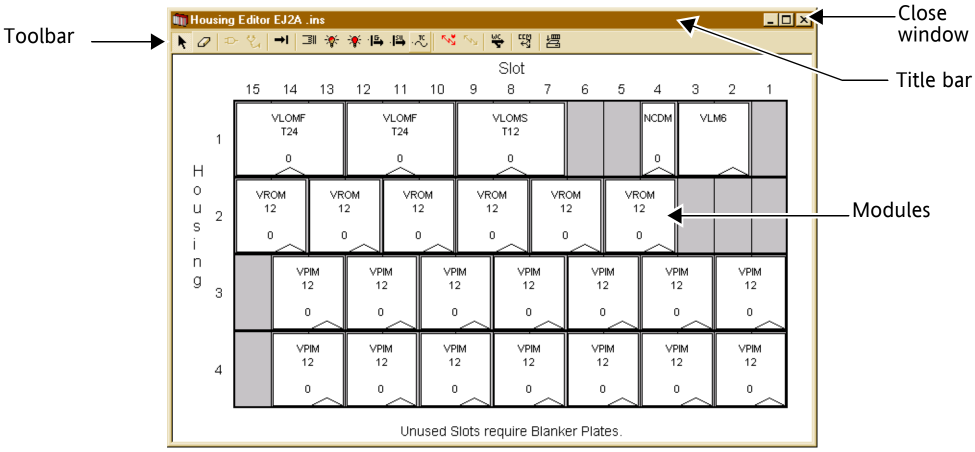


Figure 4.2 Housing Editor—window

4.2.2.1 Module Representation

Modules are depicted as rectangles occupying the appropriate slots in the housing. The rectangle contains the module's type, sequence number, and a 'chevron' to indicate its addressed slot (the slot that was clicked when the module was allocated to the housing).

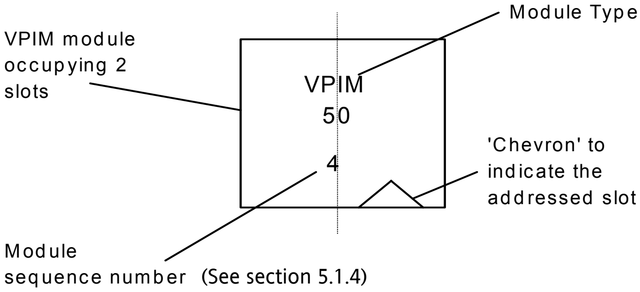


Figure 4.3 Housing Editor—module representation

Table 2.1, page 2-8 defines the number of slots occupied by a module plus the number of slots to the left or right of the addressed slot.

4.2.2.2 Supported Modules

Table 2.1 on page 2-8 lists modules supported by the GCSS.

4.2.2.3 Housing Editor Toolbar

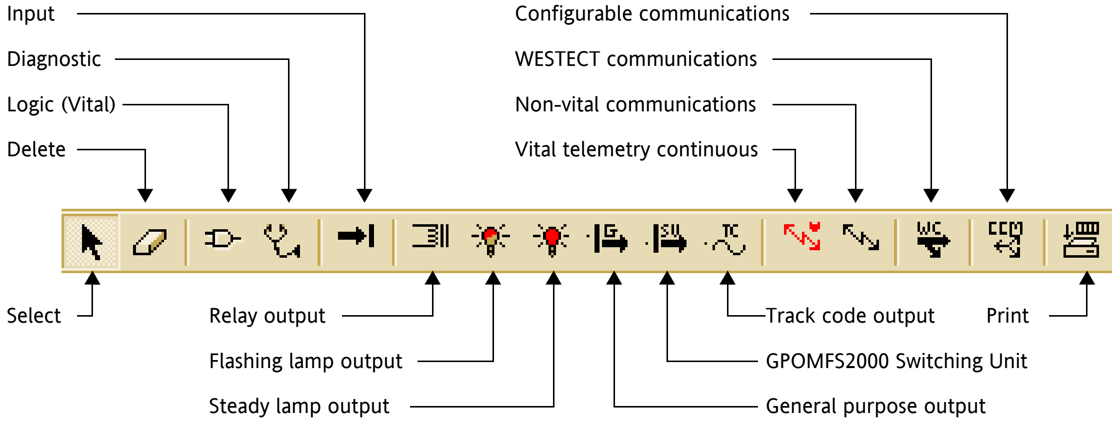


Figure 4.4 Housing Editor—toolbar

Use the buttons on this toolbar or options on the Modules menu when editing the WESTRACE application housings.

4.2.3 Configure the Modules Menu and Housing Editor Toolbar

The direct relationship between the Configure menu, the Modules menu and the Housing Editor toolbar, is shown by figure 4.5.

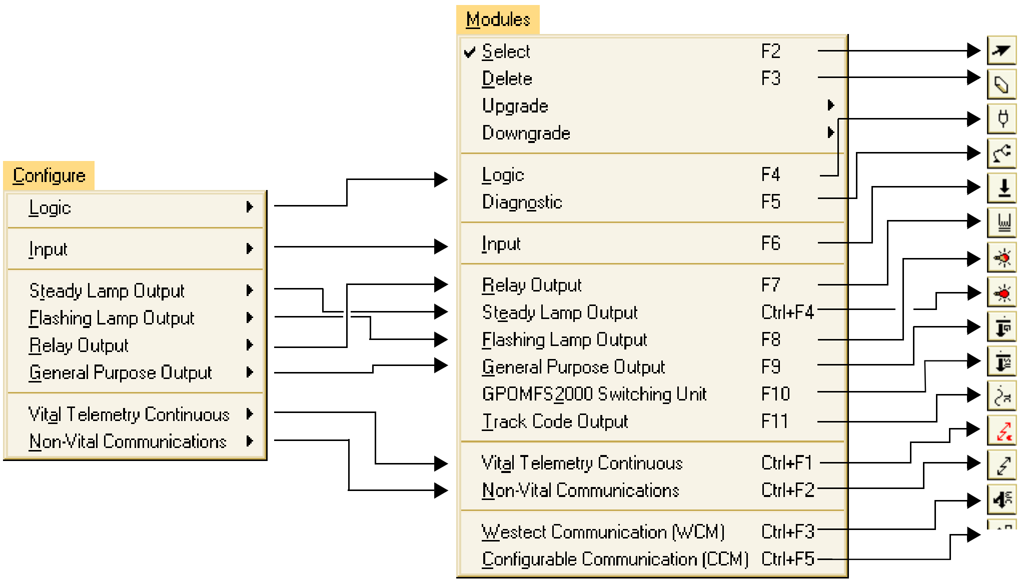


Figure 4.5 Configuration menu, Modules menu and Housing Editor toolbar—relationships

4.2.3.1 Configure Menu

Use this menu to select and link a module variant from each module group to the relevant item or button on the Modules menu and Housing Editor toolbar.

Note: ***Always select and link the Logic (Vital) module first. The choice can affect the availability of other modules.***

a) Select the appropriate menu group from the Configure menu.

b) Move the mouse pointer over the ▶ and select the required module from the sub menu. The selected module is indicated with a tick (✓).

The current configuration is stored in the GCSS "*.ini*" file and is immediately available next time the GCSS is started.

4.2.3.2 Compatible WESTRACE Modules

Tables 4.1 and 4.2 show the permissible combinations of WESTRACE modules that can be set through the Configure menu.

Diagnostic modules and the NCDM or NVC/DM module are mutually exclusive. An installation cannot contain both.

		Vital Logic Module			
WESTRACE Module	**Max. Permitted in application**	**HVLM128**	**HVLM128A**	**VLM5**	**VLM6**
DM128	1	✓	✗	✗	✗
DM128a	1	✗	✓	✗	✗
NVC/DM	1	✓	✗	✓	✗
NCDM	1	✗	✗	✗	✓

Table 4.1 Compatible combinations—VLM, NVLM and diagnostic modules

		Vital Logic Module			
WESTRACE Module	**Max. Permitted in application**	**HVLM128**	**HVLM128A**	**VLM5**	**VLM6**
VPIM12	26	✓	✓	✓	✓
VPIM24	26	✓	✓	✓	✓
VPIM50	26	✓	✓	✓	✓
VROM12	26	✓	✓	✓	✓
VROM24	26	✓	✓	✓	✓

Table 4.2 Compatible combinations—VLM, I/O and communications modules

		Vital Logic Module			
WESTRACE Module	**Max. Permitted in application**	**HVLM128**	**HVLM128A**	**VLM5**	**VLM6**
VROM50	26	✓	✓	✓	✓
VLOMFS12	18	✓	✓	✓	✓
VLOMFS24	18	✓	✓	✓	✓
VLOMFS110	18	✓	✓	✓	✓
VLOMSS12	18	✓	✓	✓	✓
VLOMSS24	18	✓	✓	✓	✓
VLOMSS110	18	✓	✓	✓	✓
VLOMFT12	18	✓	✓	✓	✓
VLOMFT24	18	✓	✓	✓	✓
VLOMFT110	18	✓	✓	✓	✓
VLOMST12	18	✓	✓	✓	✓
VLOMST24	18	✓	✓	✓	✓
VLOMST110	18	✓	✓	✓	✓
VTC232	8	✓	✓	✓	✓
EVTC	8	✓	✓	✓	✓
NVC232	10	✓	✓	✓	✓
NVC422	10	✓	✓	✓	✓
CNVC	1	✓	✓	✓	✓
GPOM110L	57	✓	✓	✓	✓
GPOM50LL	57	✓	✓	✓	✓
GPOMFSU	57	✓	✓	✓	✓
TCOM	57	✓	✓	✓	✓
CCM	57	✓	✓	✓	✓
WCM	57	✓	✓	✓	✓

Table 4.2 Compatible combinations—VLM, I/O and communications modules

4.2.4 Allocate Modules to Housings

Always allocate the Vital Logic module to the housing first.

The GCSS automatically adds the vital logic module to the appropriate slot as soon as the button or menu item is selected. Particular non-vital or diagnostic modules may also be automatically added with the vital logic module. The slots in housing 1 that automatically receive the vital logic module (and possibly the non-vital logic module) are determined by the compatibility index (see section 3.4). The relationship between these slot and the compatibility index are shown by table 2.2 on page 2-11.

4.2.4.1 Adding Modules

a) Select the appropriate toolbar button or menu item from the Modules menu

The mouse pointer will change to reflect the type of module selected.

b) Click on the housing slot where the module is to be placed. The selected slot becomes the "addressed" slot for the module.

An error message will appear if the selected slot is inappropriate. See *Placement Checks* below.

Prior selection or placement of modules may preclude or limit later selections or placements.

The GCSS will allow modules to be placed in almost any location, however you would normally group all modules of one type together and leave spare slots between groups for later expansion.

Placement Checks

The Housing Editor performs the following placement checks:

- There must be sufficient free slots for the added module to fit.

  Some modules requires a free slot to the left or right of the selected slot. See table 2.1 on page 2-8.
- Slots 1 to 3 or 4 are only occupied by logic modules as shown by table 2.2 on page 2-11.

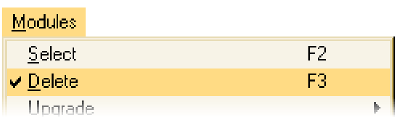
Diagnostic Modules

A DM module cannot be added to a housing with an NCDM or NVC/DM module, nor can an NCDM or NVC/DM module be added to a housing with a DM module. By convention DM modules are placed in slot 15 of housing 1.

4.2.4.2 Deleting Modules

Note: ***Deleting a module is irreversible. There is no "Undo" facility. The deleted module must be added again and configured as needed.***

a) Select **Delete**.



Alt m d
F3
Delete

b) Click on the module to be deleted. The deletion must be confirmed in a confirmation dialog box.

4.2.4.3 Cut, Copy and Paste of Whole Modules

Whole modules can be cut or copied from the Housing Editor to the Windows clipboard and then pasted from the clipboard into the Housing Editor for the same or a different installation. The entire content of the original module (eg. parameters and mnemonics) is carried over to the pasted module.

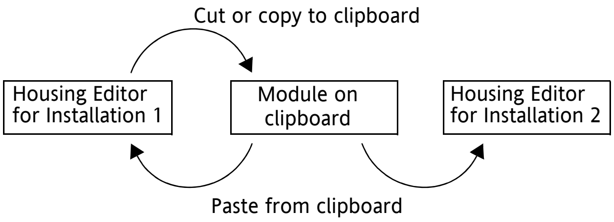


Figure 4.6 Copy, Cut and Paste using the clipboard

The clipboard stores the modules input, output, latch and timer assignments as ASCII text and it can accept only one module at a time. Each time a module is cut or copied, the prior content of the clipboard is replaced.

Cutting a Module

Select the desired module and then select **Cut**.

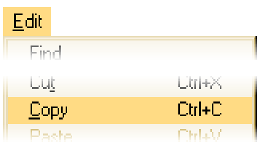
Alt e t
Ctrl+x
Shift+Delete

The selected module is removed from the Housing Editor and stored in the Windows clipboard.

Note: ***A module cannot be cut to the clipboard while it is being edited.***

Copying a Module

Select the desired module and then select **Copy**.



Alt e c
Ctrl+c
Ctrl+Insert

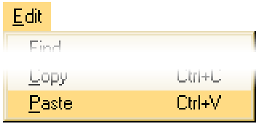
The selected module is copied to the Windows clipboard.

Note: ***A module cannot be copied to the clipboard while it is being edited.***

Pasting a Module

a) Activate the Housing Editor of the target installation.

b) Select **Paste**.



Alt e p
Ctrl+v
Shift+Insert

The mouse pointer will show a paste icon.

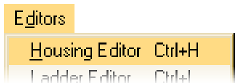
c) Click on the desired housing slot in the Housing Editor.

The same module can be pasted more than once by repeating the paste operation. Pasting of modules employs the same placement rules as adding modules.

4.2.5 Changing the VLM Type

The logic module of a VLM installation can be exchanged for another type.

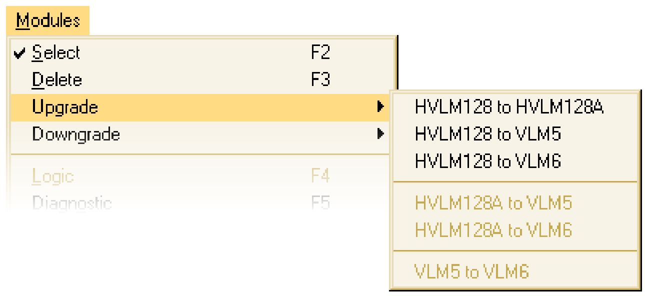
a) Select **Housing Editor** to open the Housing Editor.

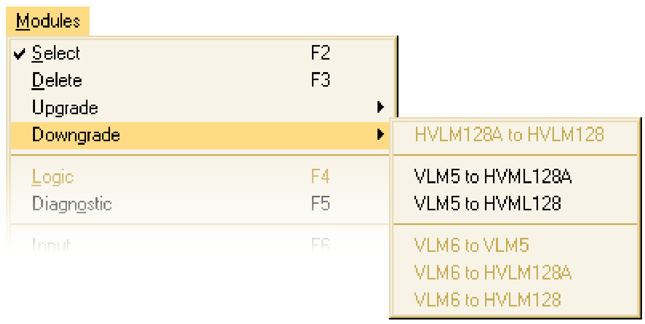


Alt d h
Ctrl+h

b) Select the vital logic module but do not double-click.
Double-clicking will initiate the Module Editor.

c) Select either **Upgrade** or **Downgrade** from the Modules menu.





Sub-menus show upgrades or downgrades appropriate for the selected vital logic module. Non-valid options are greyed out.

d) Select the desired option.

When an upgrade or downgrade is performed:

- The replacement vital logic module receives the same hot-standby or stand-alone setting as the original module.
- Timers and latches from the original vital logic module are moved to their corresponding place in the timers and latches lists of the replacement module.
  A user timer or latch in the current module is not moved and an appropriate message is displayed if the timer or latch is assigned to an unavailable number in the replacement module.
- The existing diagnostic module is not affected therefore it might be incompatible with the replacement vital logic module.

a) Check the compatibility of the diagnostic module with the replacement vital logic module. See table 4.1.

b) Check to ensure that the APPDEL duration is correctly set.

4.2.6 Reports

Two types of report are available through the Housing editor:

- Housing report
- All Modules report

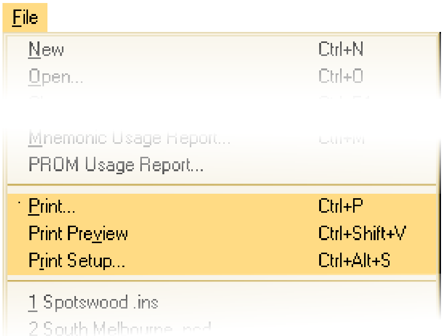
4.2.6.1 Housing Report

This report is a diagram of the WESTRACE housings showing the position of all modules. Print and keep a copy for guidance when assembling the physical modules into the housings.

With the Housing Editor active:

a) Select **Print Setup** and change the page orientation or select another printer if necessary.

b) Select **Print** or **Print Preview**.



Alt f r
Ctrl+Alt+s

Alt f p
Ctrl+p

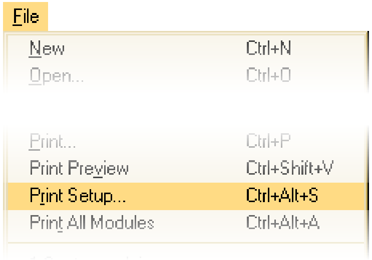
Alt f v
Ctrl+Shift+v

See section B.5.11 for a typical Housing report.

4.2.6.2 All Modules Report

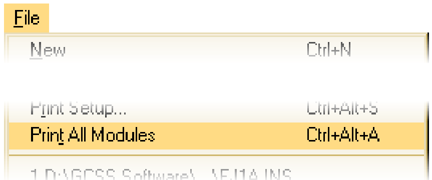
This is a multi-page report covering all modules in an installation. It is convenient way of getting the information that would be displayed if each module was opened in turn through the Module Editor. See section B.5.9 for a typical All Modules report.

a) Select **Print Setup** and change the page orientation or select another printer if necessary.



Alt f r
Ctrl+Alt+s

b) Select **Print All Modules** to display the dialog box.



Alt f t
Ctrl+Alt+a

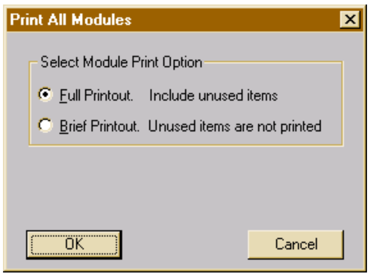


Figure 4.7 Print All Modules dialog box

c) Select the type of report required.

Full Printout—contains all list box fields in the Module Editor window regardless of whether they contain data or not.

Brief Printout—only contains those list box fields that contain data. Use this option unless you specifically need the blank fields included in the full printout.

d) Make sure the printer is switched on and select the **OK** button.

4.2.7 Selecting Modules for Editing

The Housing Editor must remain open while editing modules. Individual modules for a particular installation can be edited one at a time by invoking the Module Editor from within the Housing Editor for the installation.

The Module Editor for the installation must be closed (by selecting the **OK** or **Cancel** buttons from its toolbar) before another module from the same installation can be selected for editing.

Multiple installations can be open at any time and their associated Module Editors can all be open.

Module editing is covered in Chapter 5.

4.2.8 Finding and Replacing Mnemonics within Modules

The GCSS has the ability to find and replace mnemonics within a VLM installation when the Housing editor is active. See Chapter 9 for details.

4.2.9 Closing the Housing Editor

Closing the Housing Editor will also close the associated Module Editor.

Click the **Close** button at the top right corner of the Housing Editor Window.

5. Modules and Interface Files

This chapter describes the Module Editor and how it is used to:

- configure WESTRACE modules;
- create or select and edit interface files.

Every WESTRACE application always comprises a VLM installation and may contain an NVLM installation.

WESTRACE applications must contain an NVLM installation when the VLM installation includes an NVLM module such as an NCDM or NVC/DM. Refer to section 2.1.2.3.

Note: ***NVLM installations need not be part of a WESTRACE application. They can be used separately.***

VLM Installations

This chapter describes how to:

- set up parameters and mnemonics for all modules in the VLM installation except an NVLM (if included);
- create or select and edit a VLM-NVLM Interface File[1] that defines the data communicated between the NVLM and VLM modules when the VLM installation includes an NVLM module (NCDM or NVC/DM);
- select and setup VLM6 communication ports for communication with other VLM6-based WESTRACE applications;
- create or select and edit a VLM6-VLM6 Interface File[2] that defines the data communicated between VLM6-based WESTRACE applications. VLM6-VLM6 interface file are associated with VLM6 communication ports.

NVLM Installations

This chapter describes how to:

- set up parameters and mnemonics for the NVLM module;
- set up external communication ports;
- create or select and edit a VLM-NVLM Interface File[1] for communication between the NVLM and VLM modules.

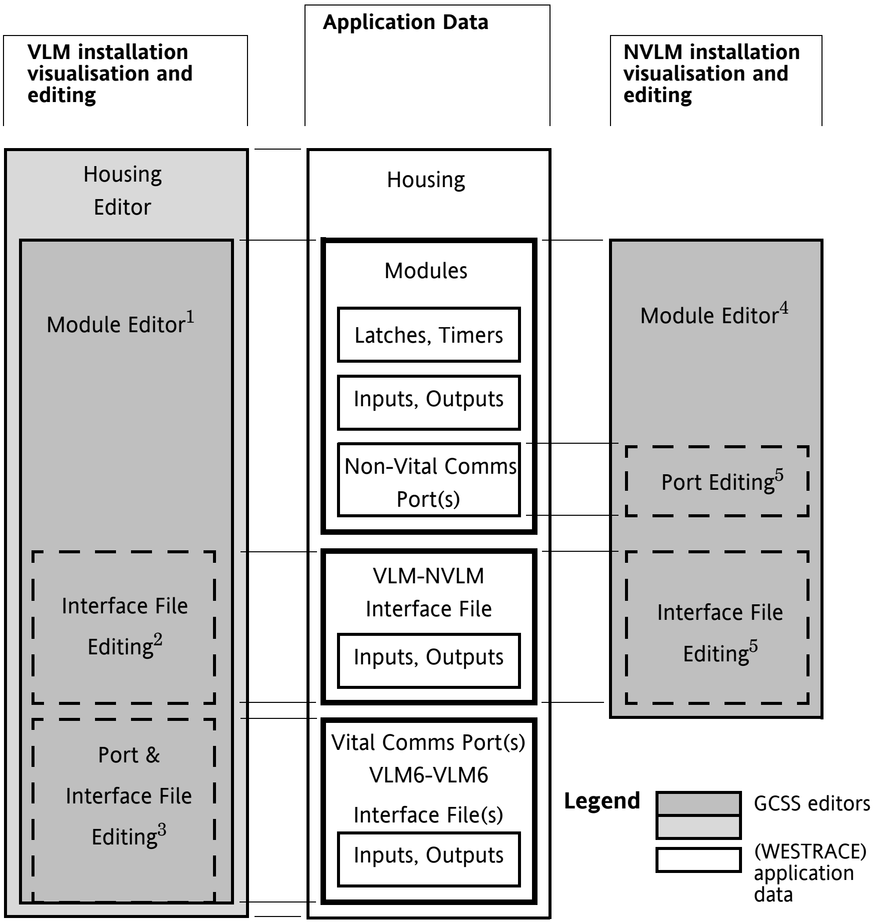
5.1 Module Editing Overview

Editing comprises the entry, modification or removal of mnemonics for inputs, outputs, latches and timers. It also involves the setting of other parameters including timer settings, communication protocols, module sequence numbers and passwords.

1 Invensys Rail recommends the VLM-NVLM interface file be created by editing the NVLM module in the VLM installation then selecting and attaching the interface file to the NVLM module when editing the NVLM installation.

2 Invensys Rail recommends the VLM6-VLM6 interface file be created by editing the VLM6 module in one VLM installation and then selecting and attaching the interface file to the VLM6 in the other VLM installation.

Figure 5.1 provides an overview of how the editors are applied to components of the WESTRACE application data.



1. Module Editor is a child window opened from within the Housing Editor window.
2. Interface file editing is via the (NCDM or NVC/DM) Module Editor.
3. Port and interface file editing is via a tab sheet within the (VLM6) Module Editor.
4. Module Editor is a child window opened from within the Installation window.
5. Port and interface file editing is via a child window opened from a tab sheet within the (NCDM or NVC/DM) Module Editor window.

Figure 5.1 Module Editor—applied to Application Data

The Module and Port editors work on a copy of the module data and only write over the original data when properly closed. See figure 5.2.

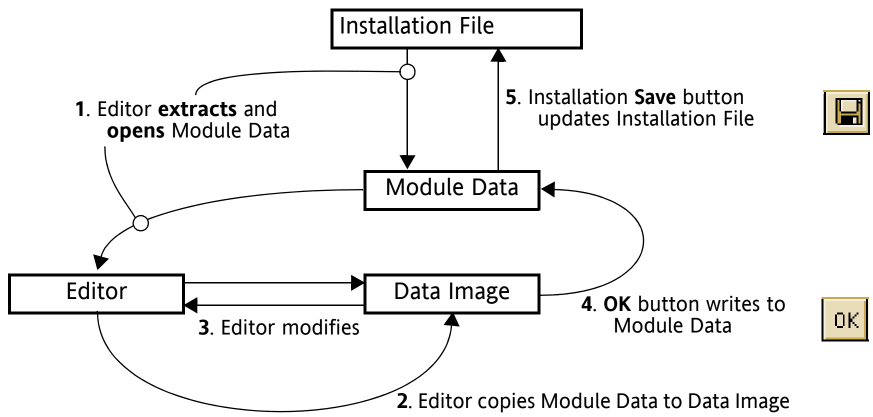


Figure 5.2 Editing data flow

5.1.1 Module Editing Window

The Module Editor window displays data and tools relevant to the module being edited. Figure 5.3 shows a general example.

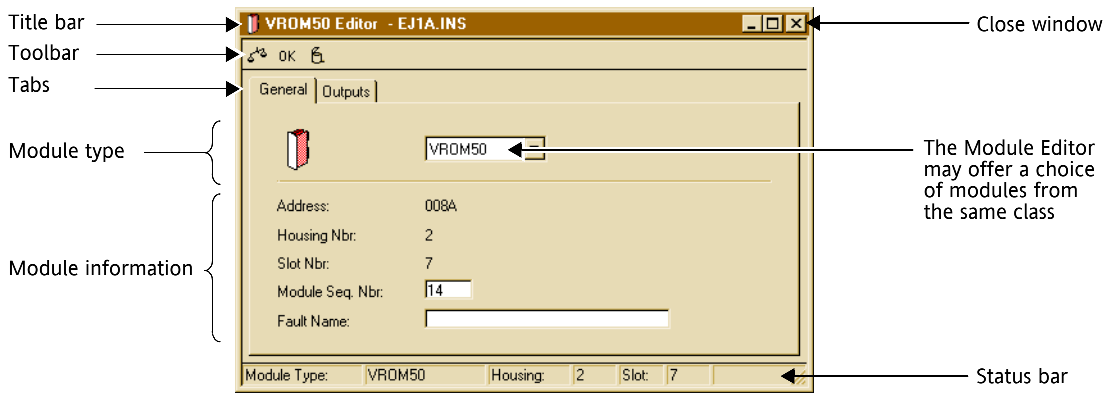


Figure 5.3 Module Editor window

- **Title bar**—shows the name of the module being edited. For example, when editing a VROM50 module, the title bar shows "VROM 50 Editor". When editing "port" configuration data, the title bar may show "Port editor".
- **Tool bar**—changes according to what is being edited. See section 5.1.2.
- **Tabs**—change according to the type of module being edited. Select these tabs to edit the underlying module parameters.
- **Module type**—tells you which type of module is being edited. Sometimes the editor allows you to select a sub-type.
- **Module information**—changes according to the type of module being edited:
  - See section A.3 for details of specific modules.
  - See section 5.1.4 for information about Module Sequence Numbers.
- **Status bar**—changes according to the type of module and the data being edited.

5.1.2 Module Editor Toolbars

The editor toolbar changes according to what is being edited. Figure 5.4 illustrates the two variants of the toolbar.

These buttons only appear or become active when an interface file is associated with a port. See section 5.1.3.

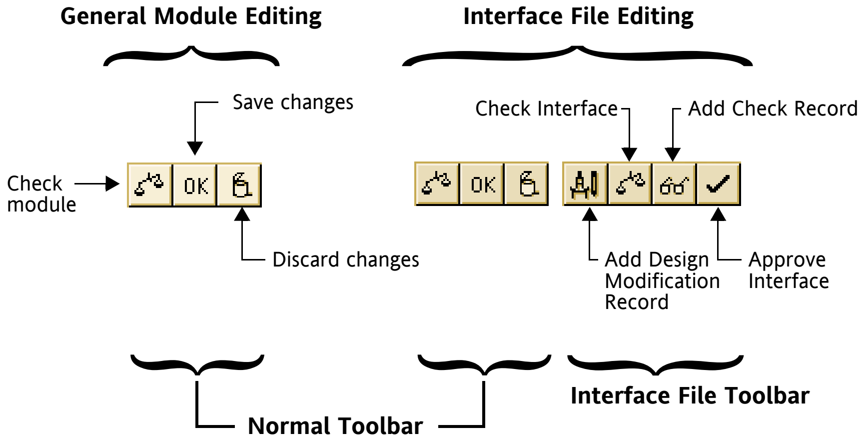


Figure 5.4 Module Editor—toolbars

Check module—select to perform consistency and data validity checks on the module after editing. See section 5.2.3.

Save changes—select to:

- perform validity checks on the edited information (see section 5.2.3), and;
- accept edited information (if valid) and write it to disk, and;
- close the Module Editor window.

Discard changes—select to abandon the edited information and close the Module Editor window. Abandonment must be confirmed in a confirmation dialog box to ensure the edited data is not inadvertently lost.

Add design-modification record—select to add a design record after creating or selecting or editing the interface file. The file is saved.

Check interface—select to perform consistency checks on the interface file.

Add check record—select to add a checked record after the consistency checks are passed. The interface file is saved.

Approve interface—select to add an approval record after a checked record has been added. The interface file is saved.

5.1.3 Ports

VLM6 and NVLM modules use ports for external communication.

VLM6 modules can communicate with other VLM6 modules (in other WESTRACE applications) via the network port. Every instance of a communication link between VLM6 modules must use an associated, unique VLM6-VLM6 interface file.

NVLM modules (NCDM or NVC/DM):

- communicate with the associated VLM module through a VLM interface port (there must be an associated VLM-NVLM interface file);
- have external communication ports for:
  - Diagnostic connections;
  - S2 connections.

5.1.4 Module Sequence Numbers

Every module supported by the GCSS, including non-vital logic modules (NCDM or NVC/DM), but excluding the vital logic module (HVLM128, HVLM128a, VLM5 or VLM6) and the diagnostic module (DM128 or DM128a), must have a module sequence number.

Module sequence numbers serve a dual purpose:

- They uniquely identify each I/O module within an installation.
- They define the order in which the modules are stored in the Vital PROM Data and hence the order in which modules are processed by the WESTRACE equipment.

  The exception to this rule are NVC and CNVC paired modules which are always processed last by WESTRACE.

Module sequence numbers must be in the range 1 to 99 and all sequence numbers within an installation must be unique. They are specified by entry in a field on the module editing window. See figure 5.5.

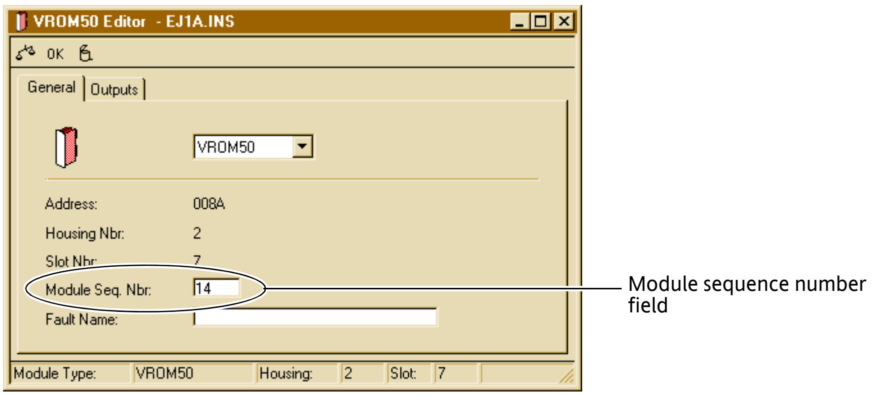


Figure 5.5 Module sequence number entry—example

Module sequence numbers are not stored in the Vital PROM Data and cannot be recreated in VLM installations uploaded from PROMs or via the diagnostic module. Therefore, sequence numbers can differ between uploaded and original installations.

Recommended Order for Allocating Sequence Numbers

Reading column 1 top to bottom, table 5.1 shows the recommended order for assigning sequence numbers.

	Category	WESTRACE Modules
1	Non-vital logic	NVC/DM, NCDM
	Vital Telemetry	VTC232, EVTC
	Non-vital communications	NVC232, NVC422, CNVC
	Other communications	CCM, WCM
	Outputs	VROM, VLOM, GPOM, TCOM
99	Inputs	VPIM

Table 5.1 Sequence numbers—recommended assignation

5.1.5 Fault Names

Fault names are pre-defined reserved mnemonics. See section 2.4.

5.1.6 Lamps, User Latches, User Timers, Inputs, Outputs and Channels

5.1.6.1 Names

Names for specific lamps, user latches, user timers, inputs, outputs, or channels are user defined mnemonics. A unique mnemonic for each of these items must be used as needed throughout the GCSS data entry system. Use familiar signalling mnemonics to help understand operation. See section 2.4.

5.1.6.2 Initial States

All directly mapped logic states must have the same initial state, ie if an input is directly mapped to an output, by assigning the same mnemonic name to both, then they both must have the same initial state.

5.1.6.3 Lists

Most modules use mnemonics for one or more of lamps, latches, timers, inputs, outputs, or channels and these mnemonics are displayed as lists on tab sheets. Figures 5.6 to 5.10 show examples.

Each row in a list corresponds to one lamp, latch, timer, input, output, or channel and the columns contain a parameter for each. Scroll bars appear when the list contains more information than can be displayed. The currently-selected field is highlighted.

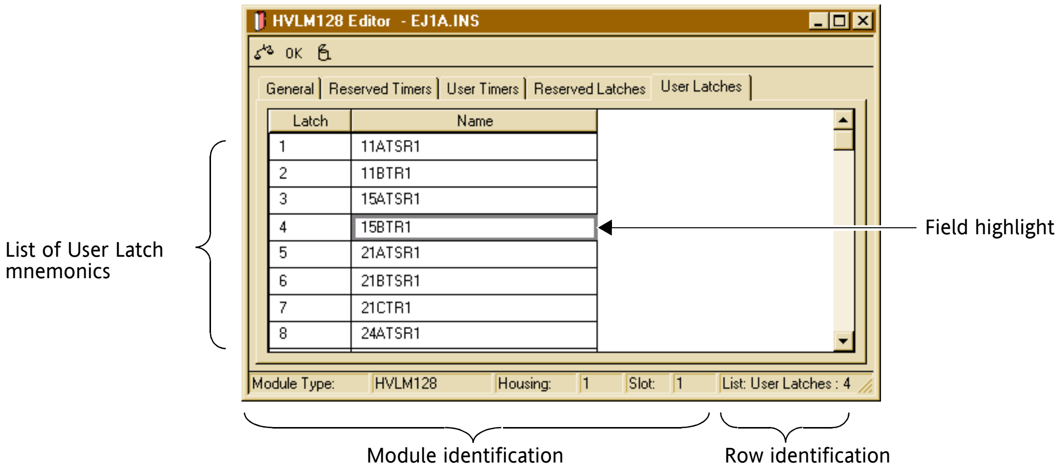


Figure 5.6 Lists—User Latch mnemonics

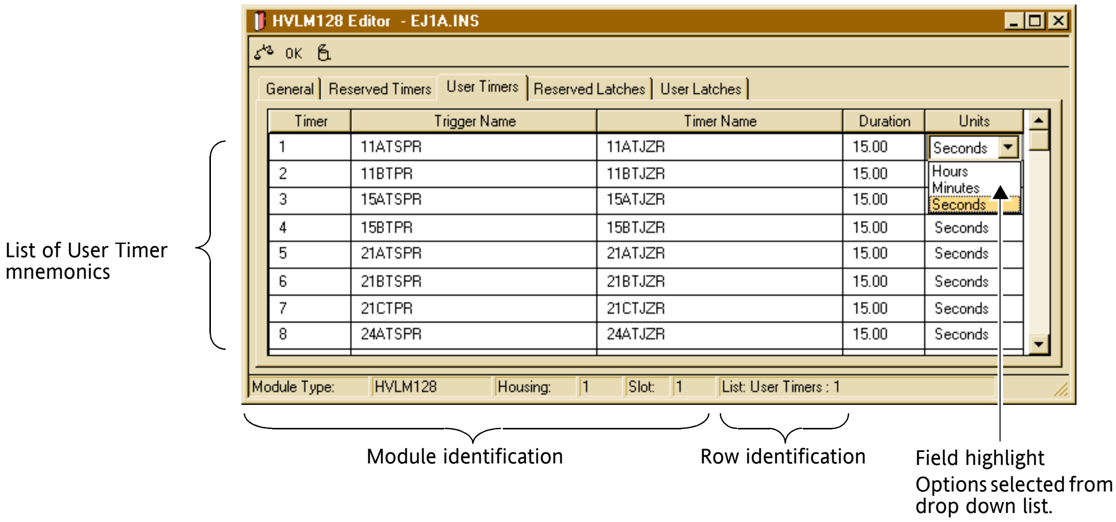


Figure 5.7 Lists—User Timer mnemonics

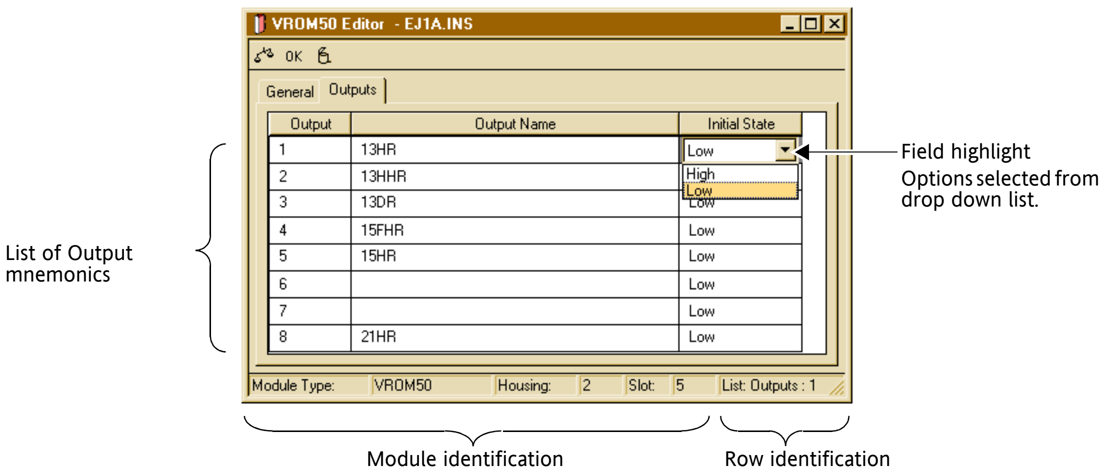


Figure 5.8 Lists—Output mnemonics

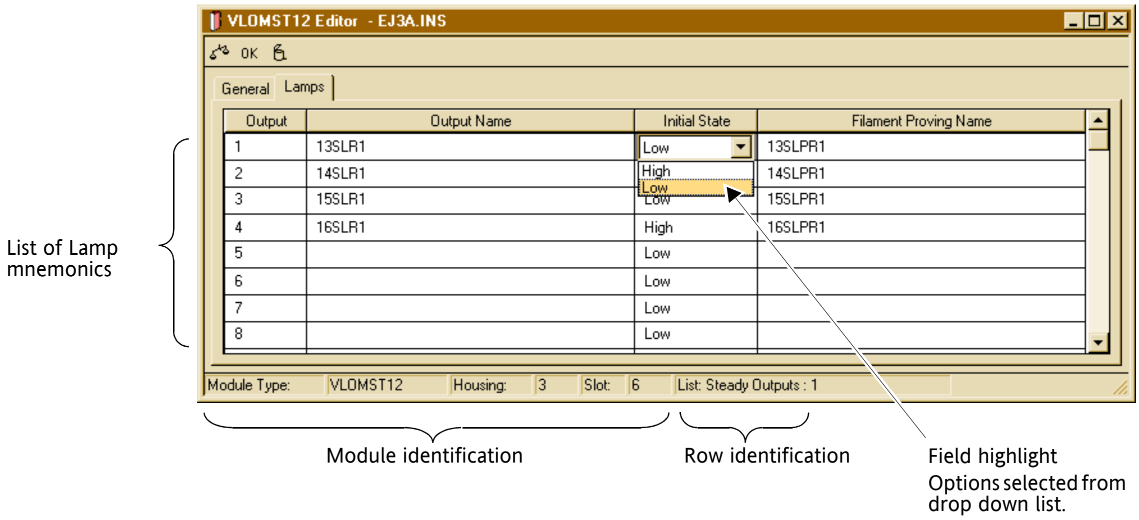


Figure 5.9 Lists—lamp mnemonics

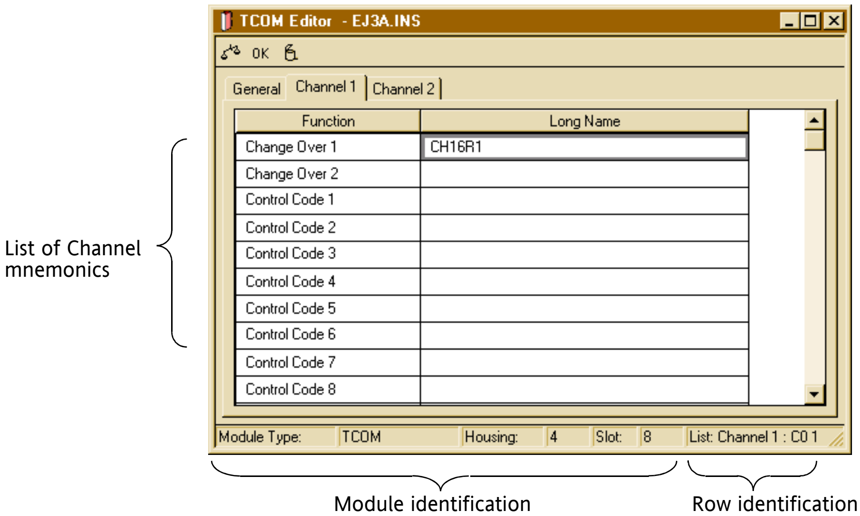


Figure 5.10 Lists—channel mnemonics

5.1.6.3.1 Finding and Replacing Mnemonics within Lists

The GCSS has the ability to find and replace mnemonics within module lists. See Chapter 9 for details.

5.1.6.3.2 List Field Types

Lists can have three types of fields:

- **Non editable fields**—contain fixed information that cannot be changed by the user, such as the input numbers or all the reserved timer and latch names. Although these fields can be highlighted, no editing is permitted.
- **Text fields**—contain alphanumeric mnemonics that can be edited when highlighted. Use uppercase for preference. See “Editing Mnemonic Text” on page 5-11.
- **Option fields**—contain a range of values. When highlighted, these fields display a drop down list of permitted values.

5.1.6.3.3 Inserting and Deleting Rows in Lists

For some modules, a blank row can be inserted or a row can be deleted.

Inserting a Blank Row

a) Ensure that the last mnemonic (“name”) field in the list is empty (see section 5.1.6.3.5, “Editing Mnemonic Text”).

The editor does not allow adding a row if the last field contains data because lower rows must be moved down.

b) Highlight a non-editable field or an option field, then right-click in the editor window.

A context menu opens (figure 5.11).

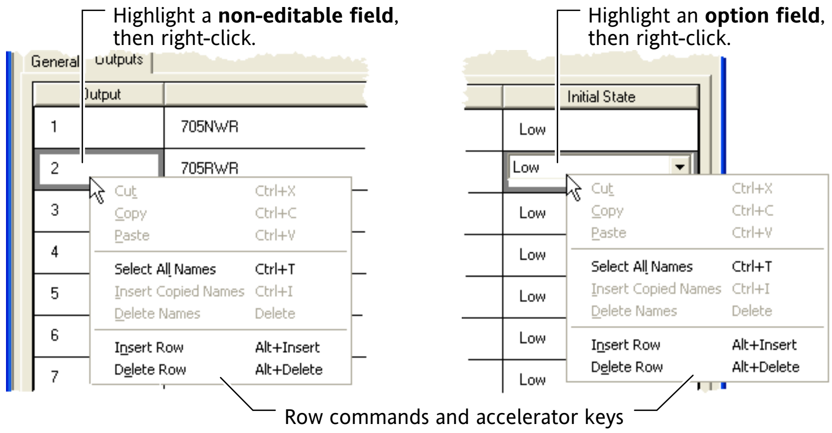


Figure 5.11 Accessing the Insert Row and Delete Row commands

c) Choose **Insert Row** from the context menu (or press **Alt+Insert**).

A blank row is inserted at the highlighted position. The highlighted row and all lower rows are shifted down by one.

Deleting a Row

a) Highlight a non-editable field or an option field in the row to be deleted, then right-click in the editor window.

b) Choose **Delete Row** from the context menu (or press **Alt+Delete**).

The row containing the highlighted field is deleted and all lower rows are shifted up by one.

Caution: ***Inserting and deleting may change the order of mnemonics in a list. Therefore be sure to keep the correct relationship between each input or output and its mnemonic when you insert or delete list rows.***

5.1.6.3.4 Working with Mnemonic Text and Fields in Editor Lists

Note: ***This section does NOT apply to timers and latches.***

In addition to inserting an empty row and deleting a row (section 5.1.6.3.3) you can:

- edit the mnemonic text within a "name" field—see section 5.1.6.3.5, "Editing Mnemonic Text";
- move one or more mnemonic fields—see section 5.1.6.3.6, "Moving Mnemonic Fields".

Figure 5.12 shows the commands available for these two different operations, and how they become available.

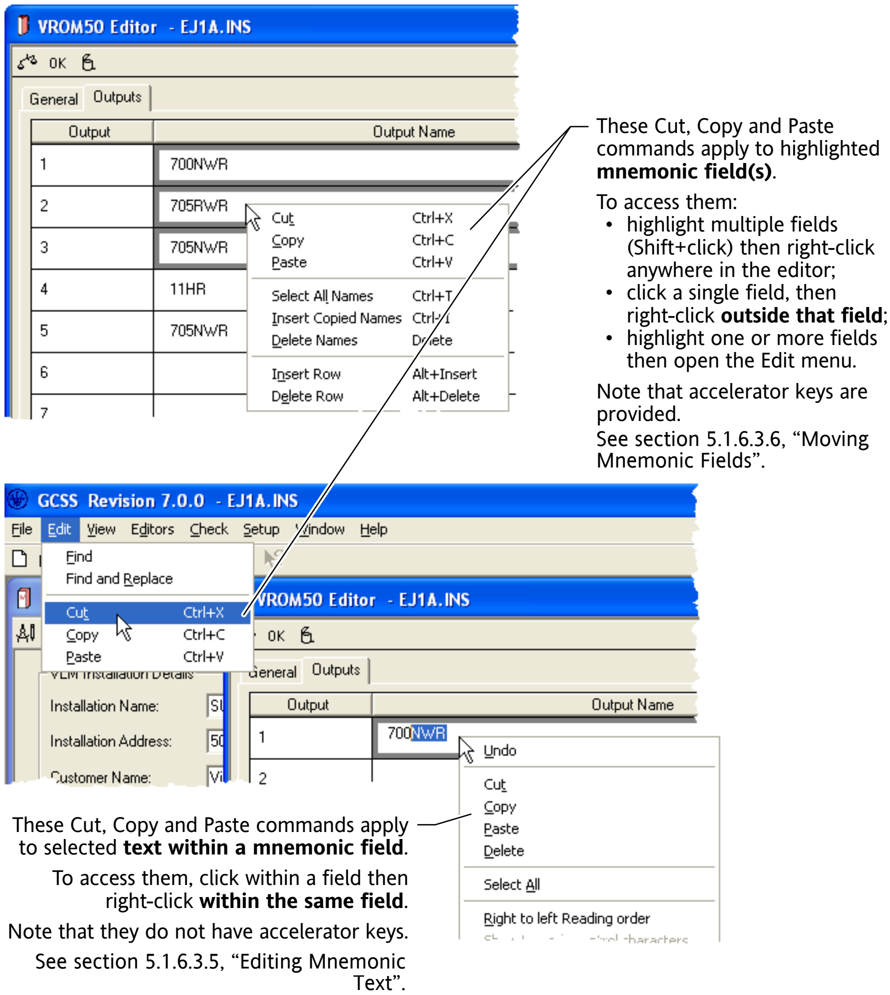
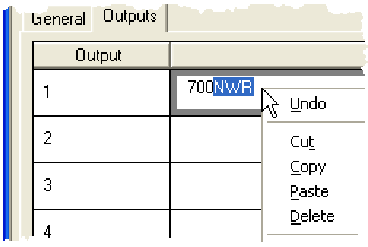


Figure 5.12 Working with field contents or entire fields—use the correct editing command

5.1.6.3.5 Editing Mnemonic Text

Note: ***This section does NOT apply to timers and latches.***



To edit text within a mnemonic field, click in the field then use standard text editing procedures such as:

- typing or pasting text at the insertion point;
- moving the insertion point using the left and right arrow keys;
- selecting text by dragging (or using Shift + arrow key) then:
  - deleting it using the Backspace (left delete) key;
  - cutting or copying it to the clipboard
  - replacing it by pasting.

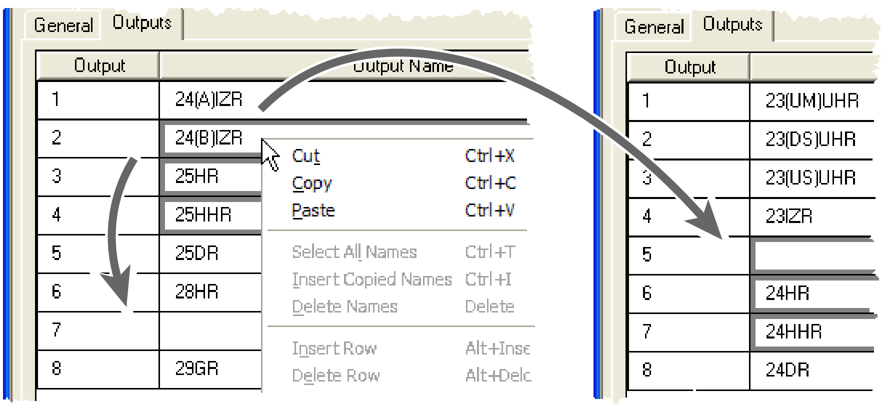
Note:

- The Cut, Copy and Paste commands you use to edit text within a mnemonic field are only available on the module editor's context (right-click) menu **when a single field is highlighted** and when you right-click **inside that field**. See figure 5.12.
- These commands have no accelerator keys. See figure 5.12.

  Any Cut, Copy or Paste command that is displayed with an accelerator key does not apply to text within a field, but to the entire highlighted field or fields (section 5.1.6.3.6).

5.1.6.3.6 Moving Mnemonic Fields

Note: ***This section does NOT apply to timers and latches.***



You can move mnemonic fields from one location to another within a list, or from one list to another, using Cut and Paste or Copy and Paste.

Figure 5.12 shows how you access the Cut, Copy and Paste commands that do this. You can Cut, Copy or Paste:

- a single mnemonic field, or;
- a contiguous group of mnemonic fields (click the first field of the group then Shift+click the last field of the group).

The status bar displays the list name and row number within the list.

Note: ***Pasting overwrites existing mnemonics. Therefore remember to move[3] any existing wanted fields to elsewhere in the list before pasting the new fields.***

Cutting

Highlight the mnemonic field(s), then choose **Cut**.

Alt e t
Ctrl+x

Alternatively, highlight the fields then right-click in the module editor (for a single field, right-click **outside** the field) and choose **Cut** from the context menu (figure 5.12).

The highlighted fields are removed and stored in the clipboard from where they can be pasted later. The source mnemonic fields are left empty, and the associated option fields are reset to their default values.

Caution: ***Cutting may change the order of mnemonics in a list. Therefore be sure to keep the correct relationship between each input or output and its mnemonic when you Cut fields from a list.***

Copying

Highlight the mnemonic field(s), then choose **Copy**.

Alt e c
Ctrl+c
Ctrl+Insert

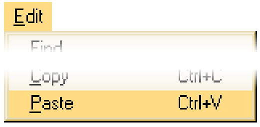
Alternatively, highlight the fields then right-click in the module editor (for a single field, right-click **outside** the field) and choose **Copy** from the context menu (figure 5.12).

The highlighted fields are copied to the clipboard from where they can be pasted later.

[3] Cut/Copy then Paste.

Pasting

Highlight the destination mnemonic field (or a contiguous group of destination fields), then choose **Paste**.



Alt e p
Ctrl+v
Shift+Insert

Alternatively, highlight the destination mnemonic field(s), then right-click in the module editor (for a single destination field, right-click **outside** the field) and choose **Paste** from the context menu.

The contents of the clipboard are pasted into the highlighted field(s). You will be alerted if:

- the destination fields are not empty (a dialog box asks if you want to overwrite the data);
- the number of fields in the clipboard is not the same as the number of highlighted destination fields.

Other fields in the destination rows are reset to their default values when fields are pasted.

The same fields can be pasted more than once by repeating the paste operation.

Caution: ***Pasting may change the order of mnemonics in a list. Therefore be sure to keep the correct relationship between each input or output and its mnemonic when you Paste fields into a list.***

Caution: ***Check that pasted data is what you intended and not an unwanted remnant from the clipboard.***

5.1.6.3.7 Other Mnemonic Commands

Note: ***This section does NOT apply to timers and latches.***

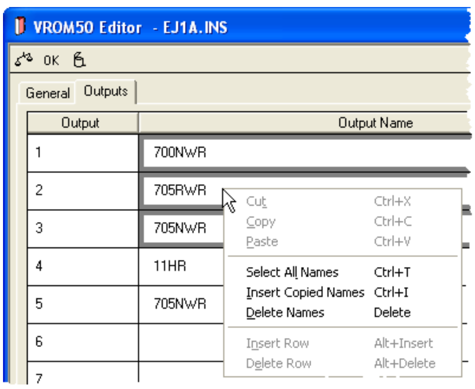


Figure 5.13 Select All Names, Insert Copied Names and Delete Names commands

Select All Names

Use **Select All Names** (accelerator keys: **Ctrl+t**) to highlight all mnemonic fields in a module list.

The **Select All Names** command is only available on the module editor's context menu (figure 5.13):

- With multiple fields highlighted, right-click anywhere in the module editor.
- With only one field highlighted, right-click outside that field.

Insert Copied Names

Use **Insert Copied Names** (accelerator keys: **Ctrl+i**) to insert copied fields into a module list at the highlighted position. The highlighted row and all lower rows are shifted down. There must be an appropriate number of empty mnemonic fields at the end of the list; GCSS warns you if this is not the case.

The associated option fields are reset to their default values

The **Insert Copied Names** command is only available on the module editor's context menu (figure 5.13):

- With multiple fields highlighted, right-click anywhere in the module editor.
- With only one field highlighted, right-click outside that field.

Delete Names

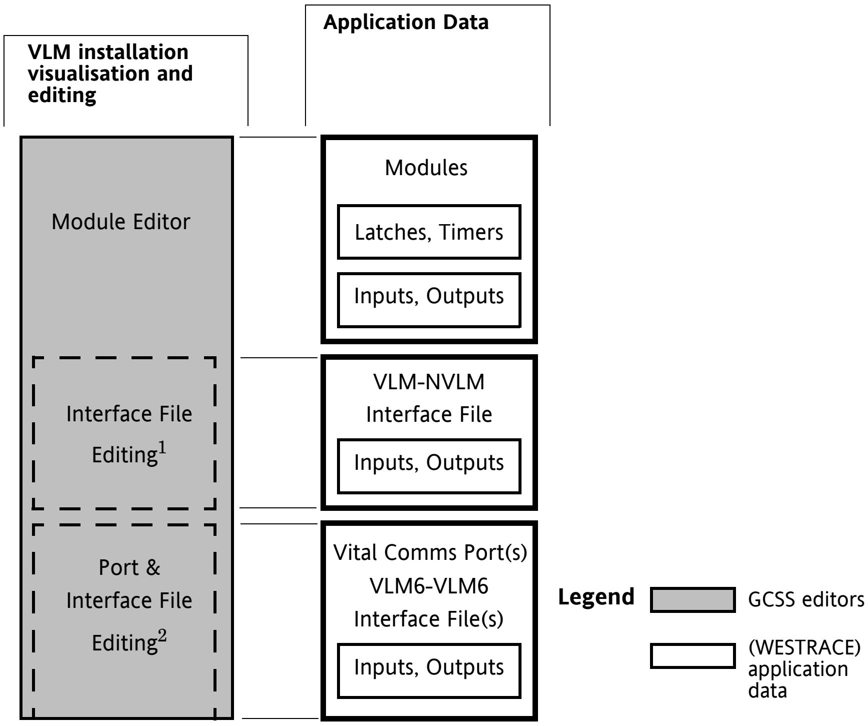
Use **Delete Names** (accelerator key: **Delete**) to delete the currently-highlighted mnemonics from a module list.

The mnemonic fields are left empty and the associated option fields are reset to their default values. Names deleted in this way are *not* copied to the clipboard.

The **Delete Names** command is only available on the module editor's context menu (figure 5.13):

- With multiple fields highlighted, right-click anywhere in the module editor.
- With only one field highlighted, right-click outside that field.

5.2 Module Editing—VLM Installations



1. VLM-NVLM interface file editing is via the (NVLM) Module Editor. See section 5.2.8.
2. Port and interface file editing is via a tab sheet within the (VLM6) Module Editor. See section 5.2.7.

Figure 5.14 Module Editing—VLM Installation

Summary of Module Editing

General editing of modules in a VLM installation comprises the following basic steps:

a) Activate the module editor. See section 5.2.1.

b) Set general parameters, values, input and output mnemonics as required. See section 5.2.2.

c) Check module data validity. See section 5.2.3.

d) Save the module data. See section 5.2.4.

e) Print the Module Editor Report. See section 5.2.5.

f) Close the module editor. See section 5.2.6.

g) Save the VLM installation periodically and when all module editing is complete. See section 3.10.

Port and Interface File Editing

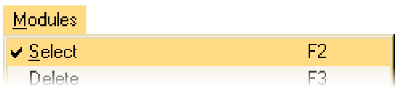
There are two likely situations:

- The VLM installation contains an NVLM module, therefore a VLM-NVLM interface file is required. See section 5.2.2.3 plus section 5.2.8 and its subsections.
- The VLM installation contains a VLM6 and an NCDM module and the VLM6 is required to communicate with a VLM6 in another WESTRACE application. Therefore:
  - a VLM-NVLM interface file is required (see section 5.2.2.3 plus section 5.2.8 and its subsections), and;
  - a VLM6-VLM6 interface file is required (see section 5.2.2.2 plus section 5.2.7 and its subsections).

5.2.1 Module Editor Activation

a) Click the VLM installation Housing Editor window to make it active.

b) Choose **Select** to change the mouse pointer to the selection arrow.



Alt m s
F2

c) Double-click on the required module with the left mouse button to open the Module Editor window for the selected module.

The labelling of the selected module changes to red to indicate it is being edited.

The Module Editor must be closed before another module from the same installation can be selected for editing.

5.2.2 General Module Editing—VLM Installation

The GCSS Module Editor represents all WESTRACE modules in a manner similar to the example shown in figure 5.15. Different aspects of the module configuration are accessed through tab sheets. All modules have a “General” tab sheet and most have other tab sheets for such things as inputs, outputs, latches, timers etc.

Appendix A section A.3 provides some general assistance with the tab sheets for all WESTRACE modules and the data fields that appear on those tab sheets. Space is provided for you to add your own notes.

The following sections in Chapter 2 provide more specific assistance when configuring modules:

- all of section 2.4, *Mnemonics*
- section 2.5.3, *Use of Latches*
- all of section 2.6, *Aspects of System Design*

Tabs

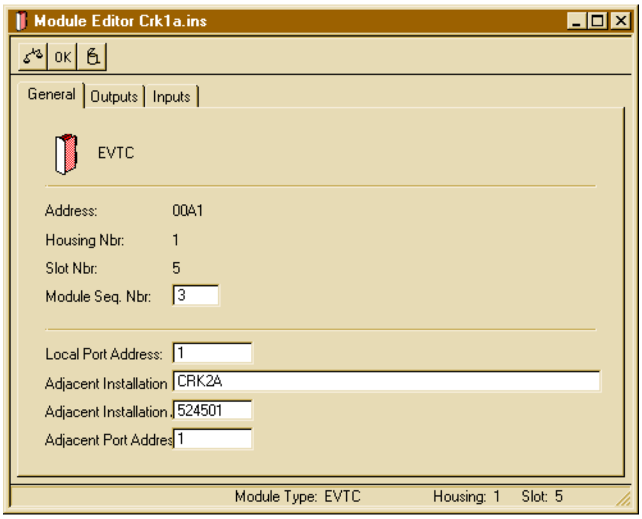


Figure 5.15 Module Editor window—typical

Select each tab on the Module Editor window in turn and:

- set values and other parameters as appropriate for the module;
- add, edit or remove mnemonics as appropriate for the module. See sections 2.4 and 5.1.6.3.5.

5.2.2.1 GPOM Modules

The GPOM modules are particularly complex with a large number of interrelated data fields. To help conform with the consistency rules, some data fields are enabled, disabled, or defaulted automatically depending on the state or values of the other data fields. This behaviour assists in the entry of consistent data.

5.2.2.2 VLM Modules

All VLM modules are set up in a similar way in that they have:

- reserved latches, and;
- reserved timers;

that cannot be edited but can be used in vital logic design.

The functions of reserved latches and timers are described in sections 2.4.5.1 and 2.4.6.1.

They and also have:

- user latches;
- user timers;

for which mnemonics must be entered and edited.

See section 5.1.6.3 for general assistance with user latch and timer mnemonics and other parameters and section 5.1.6.3.5 for editing them.

VLM6 Modules

The VLM6 module is different to other VLM modules in that it has network ports for communication with other VLM6 modules in addition to the timers and latches listed above. When a VLM6 is required to communicate with a VLM6 module in another WESTRACE application, a network port must be setup and a VLM6-VLM6 interface file must be created or selected and assigned to the port.

VLM6-VLM6 Interface Files

VLM6-VLM6 interface files have tab sheets for input and output mnemonics. These mnemonics are entered and edited in the general way described in section 5.1.6.3.5.

Specific assistance in creating and attaching VLM6-VLM6 interface files is provided in section 5.2.7.

5.2.2.3 NVLM Modules

These modules (NCDM or NVC/DM), when part of a VLM installation, must be edited to assign a VLM-NVLM interface file to the VLM installation. See section 5.2.8.

An NVLM installation must also be created to:

- Assign the VLM-NVLM interface file to the NVLM module;
- Set up communication ports as required;
- Enter and edit mnemonics for timers and latches.

Creating and editing an NVLM installation is covered in section 5.3.

5.2.3 Module Data Consistency Check

This procedure should be performed when editing is complete and before the module editor is closed. It could also be advantageous to check module data validity at selected stages while entering and editing module mnemonics.

This procedure:

- performs data validity checks on the module information set;
- checks the ranges of all numerical fields;
- ensures the module information forms a consistent set.

Any inconsistencies are reported in an alert message box.

Select **Check Consistency**.

Check
Check Consistency Ctrl+Shift+C

Alt c c
Ctrl+Shift+c

Note: ***This action will not find duplicated sequence numbers. They are found by the Installation Consistency Check. See section 7.2.***

5.2.4 Saving Module Data

The Module Editor works on a copy or image of the module data and only writes over the original module data when properly closed. See figure 5.2.

Select the **OK** button on the Module Editor toolbar before saving the installation.

An appropriate warning message is displayed if the Module Editor is open when the installation is saved or closed with unsaved module edits.

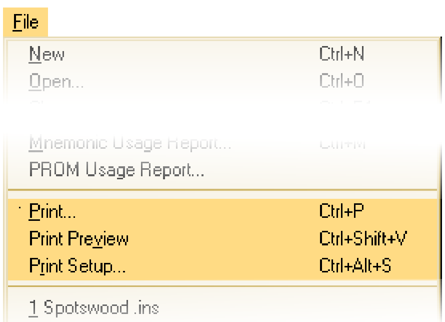
5.2.5 Module Editor Report

This printed report replicates the layout of the Module Editor window. If any list box is too wide to fit on a single page, it is continued on subsequent pages. In these circumstances, the column containing the row number is reproduced on each page. Page orientation can be changed if needed.

a) Select **Print Setup** and change the page orientation or select another printer if necessary.
   See Appendix B, section B.2 for further assistance.

b) Select **Print** or **Print Preview** to display the Print Module dialog box.

Alt f r
Ctrl+Alt+s



Alt f p
Ctrl+p

Alt f v
Ctrl+Shift+V

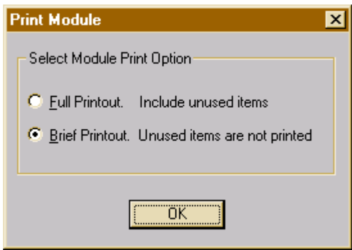
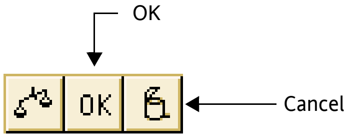


Figure 5.16 Print Module dialog box

c) Select the type of report required:
   Full Printout—shows all list box fields in the Module Editor window regardless of whether they contain data or not;
   Brief Printout—only shows those list box fields that contain data. Use this option unless you specifically need the blank fields included in the full printout.

The resulting report will be similar to the example in figure B.14 on page B-18. The examples in figures B.14 and B.15 were generated for the same NVC/DM module in matching VLM and NVLM installations.

5.2.6 Closing the Module Editor



Close the editor window by clicking the:

- **OK** button to:
  - perform data validity checks;
  - save the data and close the editor if there are no inconsistencies; Otherwise inconsistencies are reported and the editor remains open.
- **Cancel** button (or the **Window Close** button) to:
  - abandon any changes made to the data;
  - close the editor after confirmation.

5.2.7 VLM6 Modules and VLM6-VLM6 Interface Files

Note: ***This section does not cover general editing of VLM6 modules or VLM6-VLM6 interface files. See section 5.2.2.2 for this information.***

VLM6-VLM6 interface files define the mnemonics passed between the VLM6 modules in two separate WESTRACE applications over a network. A VLM6 can communicate vitally with up to 16 other VLM6-based WESTRACE applications via individual network sessions.

A separate and unique interface file is required for every instance of a VLM6 communicating with another VLM6.

Figure 5.17 illustrates how data from the VLM6-VLM6 interface file is combined with data from the two VLM6-based VLM Installation files by the GCSS compiler to produce VLM PROM data for the two WESTRACE applications.

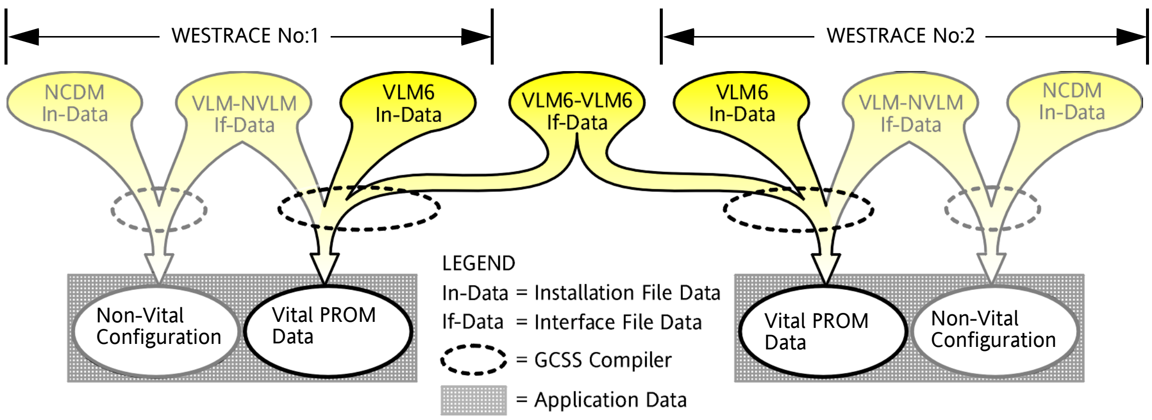


Figure 5.17 VLM6-VLM6 interface file and VLM6 Installation data—merging

The VLM6-VLM6 interface file can be an existing file created through the other VLM6 or be newly created in the current VLM6.

It contains:

- input and output mnemonics passed between the two VLM6-based VLM installations;
- configuration control history of the interface file itself.

VLM6-VLM6 Interface File Inputs and Outputs

The input and output mnemonics in a VLM6-VLM6 interface file are always seen from the point of view of the installation in which the interface file was opened. That is, a mnemonic that is an input when viewed through one VLM6 installation is seen as an output when viewed through the other VLM6 installation.

VLM6-VLM6 Interface File Creation or Selection and Editing[4]

A VLM6-VLM6 interface file can be created or selected and edited in either of the associated VLM6-based VLM installations.

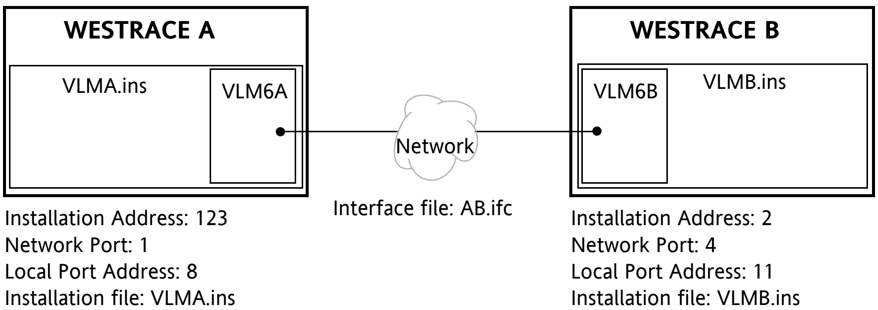
Once the interface file is set up in terms of installation addresses and port addresses, the mnemonics can be changed through either installation.

Note: ***You must use the same VLM6-VLM6 interface file for both WESTRACE applications that are communicating vitally.***

5.2.7.1 Configuring VLM6 Installations that must Communicate

This section will use a simple example to illustrate setting up VLM6 modules that are required to communicate and the necessary VLM6-VLM6 interface file. The example uses simple file names and installation addresses to simplify the explanation. Real life VLM6 installations will use file names that relate to the real WESTRACE applications and 6 digit installation addresses (most likely).

Consider two VLM6 installations with a single network session.



4 Invensys Rail recommends the VLM6-VLM6 interface file be created by editing the VLM6 module in one VLM installation and then selecting and attaching the interface file to the VLM6 in the other VLM installation.

Note: ***There is a fixed relationship between the Network Ports and the Local Port Addresses within a VLM installation:***

- ***Network Port 1 has Local Port Address 8***
- ***Network Port 2 has Local Port Address 9***

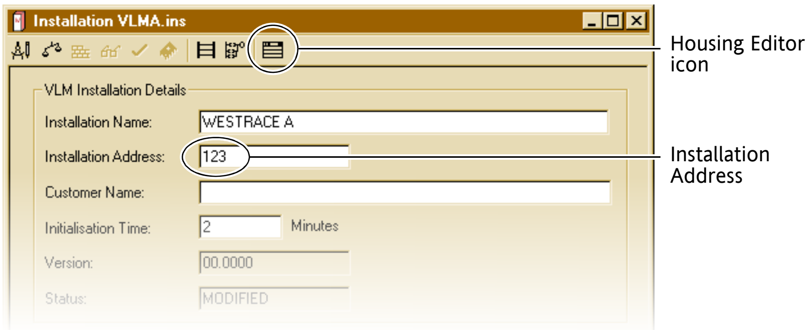
  etc, up to
- ***Network Port 16 has Local Port Address 23.***

5.2.7.1.1 First VLM6 Installation

With the first VLM6 installation active (VLMA in this example):

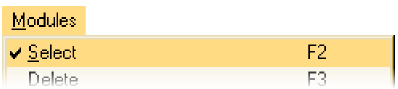
a) Complete the following fields as required:
- **Installation Name**
- **Installation Address**
- **Customer Name**
- **Initialisation Time**

b) Ensure the installation address is entered correctly (“123” in this example).



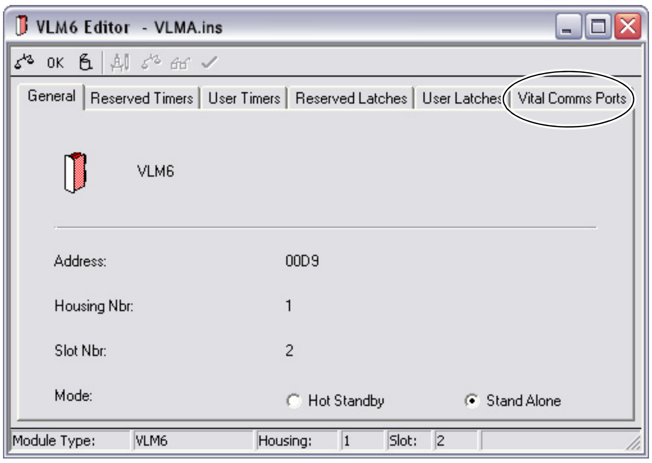
c) Click the Housing Editor icon to open the Housing Editor.

d) Choose **Select**.



Alt m s
F2

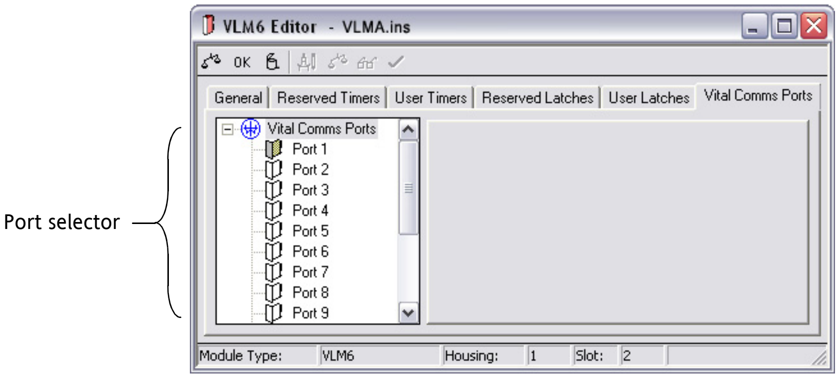
e) Double-click on the VLM6 module in the Housing Editor to open the Module Editor.



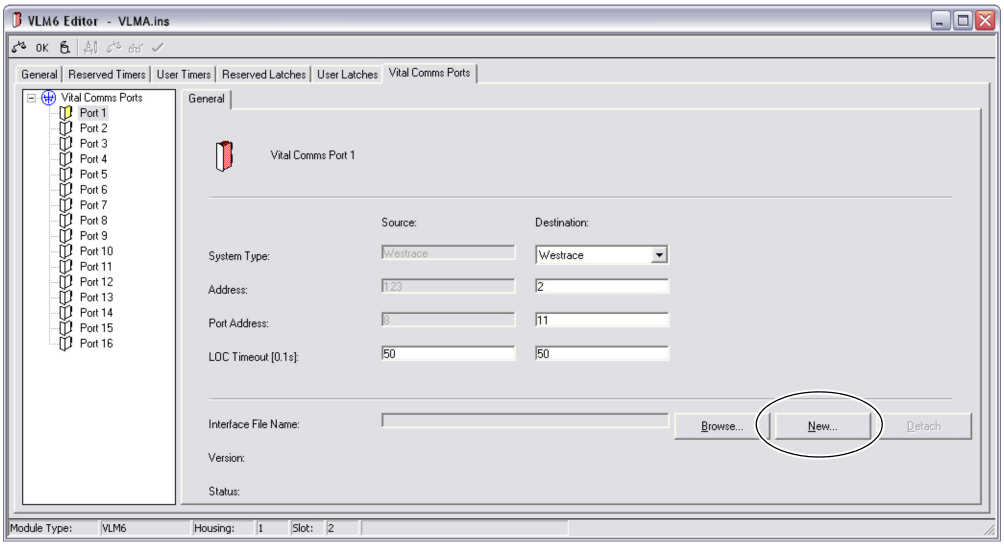
f) Set the operating mode as “Hot Standby” or “Stand Alone” as appropriate.

g) Select the **User Timers** and **User Latches** tabs in turn and edit as appropriate. See section 5.2.2.2 for assistance.

h) Select the **Vital Comms Ports** tab.

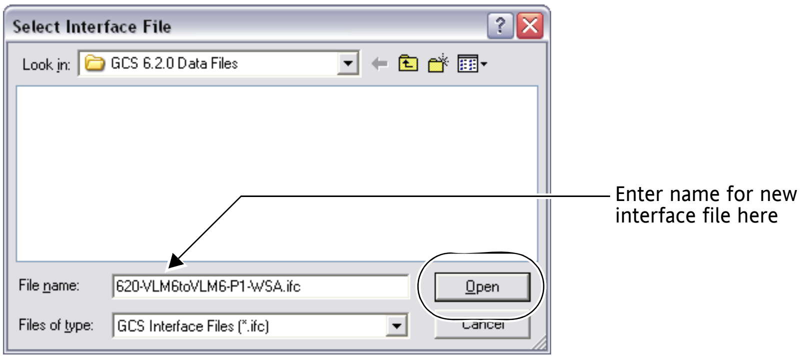


i) Select a communications port from the Port Selector (Port 1 in this example).



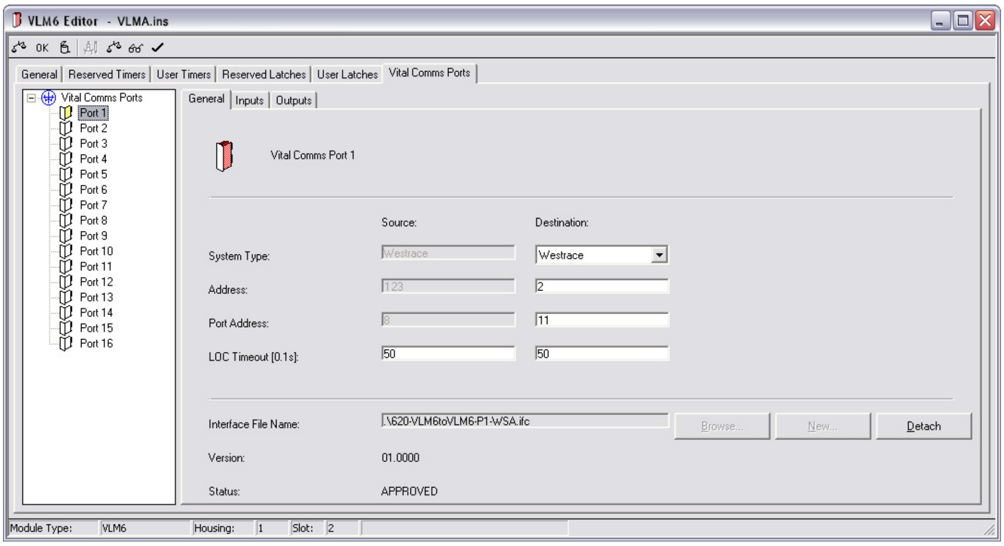
Create a New VLM6-VLM6 Interface File

j) Select the **New** button to open the Select Interface dialog box.



k) Enter a name for the new interface file.

l) Select the **Open** button.

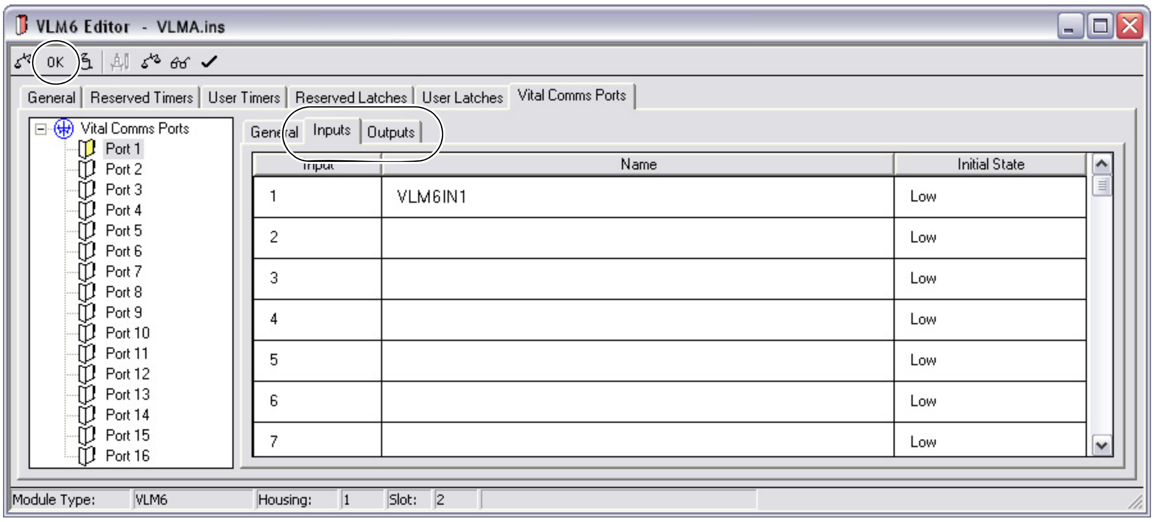


m) For the destination VLM Installation, enter values for:
- System type (WESTRACE in this example);
- Address (“2” in this example);
- Local port address (“11” in this example);
- LOC timeout.

See section A.3.16 for further details.

n) Enter a Loss of Communication (LOC) timeout value (in tenths of a second) for the current VLM Installation. See section A.3.16 for further details.

o) Use the **Input** and **Output** tabs in turn to enter mnemonics for transferring logic states between the two WESTRACE applications.



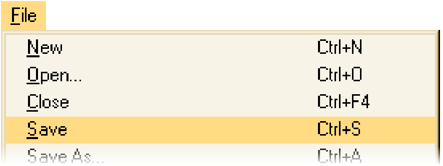
For each required input or output you will need:
- a valid mnemonic name (see section 2.4);
- the initial state, either high or low (see section 2.4.1).

Remember, an input in the “Source” installation will be an output in the “Destination” installation.

See 5.2.2.2 for general assistance in entering and editing mnemonics.

p) Use the **OK** button to save the interface file.

q) Use **File Save** to save the installation file.



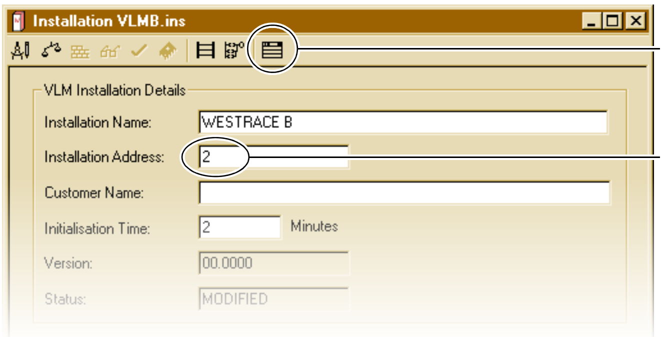
Alt f s
Ctrl+s

5.2.7.1.2 Second VLM6 Installation

With the second VLM6 installation active (VLMB in this example):

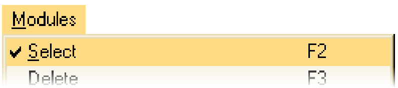
a) Complete the following fields as required:
- **InstaLlation Name**
- **Installation Address**
- **Customer Name**
- **Initialisation Time**

b) Ensure the installation address is entered correctly (“2” in this example).



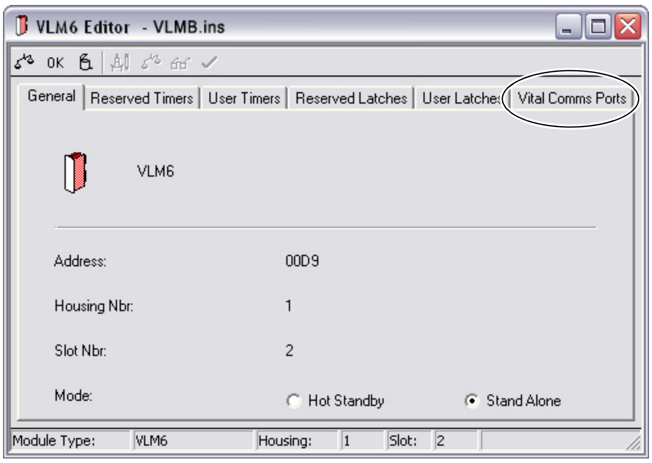
c) Click the Housing Editor icon to open the Housing Editor.

d) Choose **Select**.



Alt m s
F2

e) Double-click on the VLM6 module in the Housing Editor to open the Module Editor.

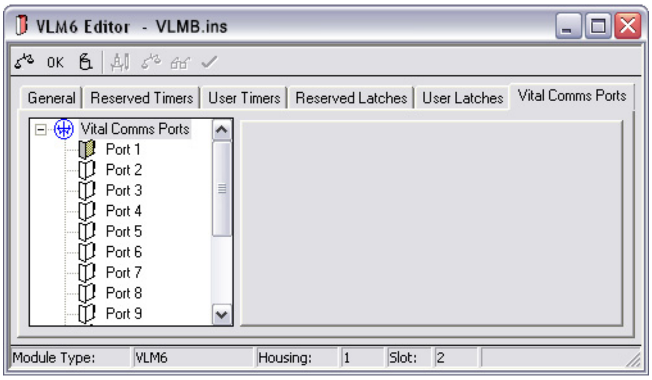


f) Set the operating mode as "Hot Standby" or "Stand Alone" as appropriate.

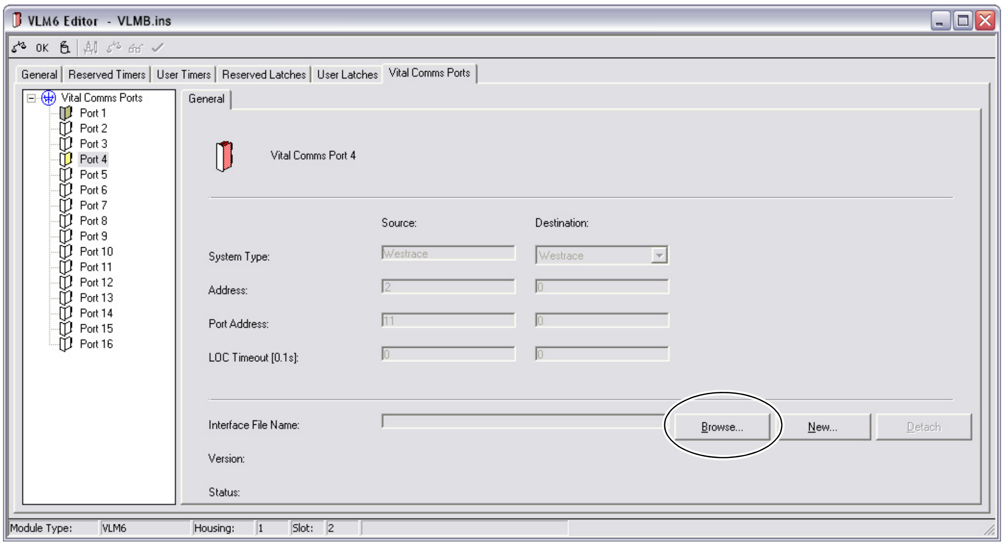
g) Select the **User Timers** and **User Latches** tabs in turn and edit as appropriate. See section 5.2.2.2 for assistance.

h) Select the **Vital Comms Ports** tab.

Port selector

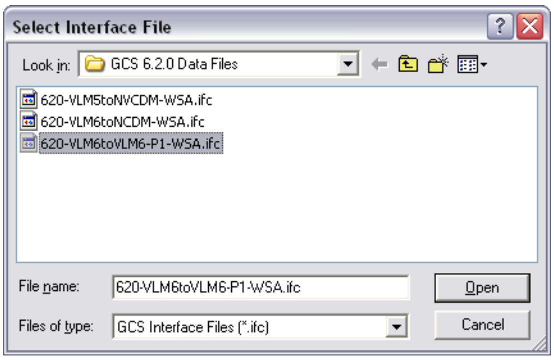


i) Select a communications port from the Port Selector (Port 4 in this example).



Select the VLM6-VLM6 Interface File

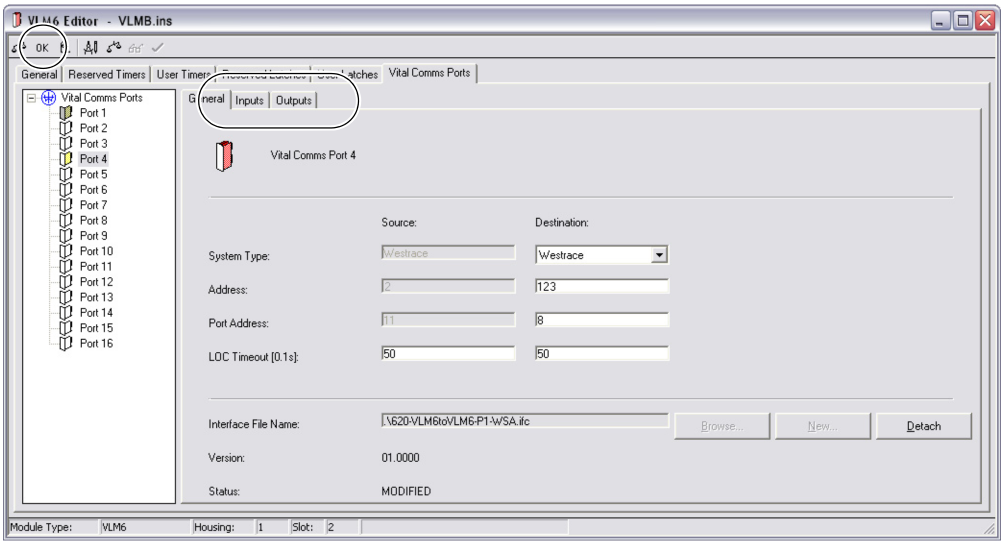
a) Select the **Browse** button to open the Select Interface dialog box.



Select existing interface file from here

b) Select the desired interface file from the list (`620-VLM6toVLM6-P1-WSA.ifc` in this example).

c) Select the **Open** button.

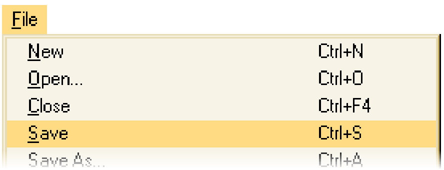


The “Destination” fields and **Input** and **Output** tab sheets are automatically filled with information extracted from the interface file. There should not be any errors. See *Installation File Errors* below for possible errors.

d) Select the **Input** and **Output** tabs in turn to check the input and output mnemonics. Mnemonics that were inputs in the first VLM installation should be outputs in the second VLM installation and vice versa.

e) Use the **OK** button to save the interface file (there should not be any changes unless you changed the I/O mnemonics).

f) Use **File Save** to save the installation file (VLMB.ins in this example).

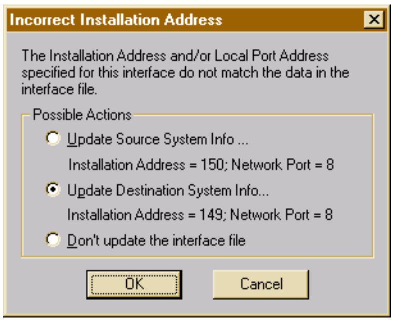


Alt f s
Ctrl+s

Installation File Errors

The “Incorrect Installation Address” dialog box is displayed when the Installation Address or Local Port Address in the interface file doesn’t match the values set in the current VLM installation. This could arise because:

- the wrong interface file was selected;
- incorrect values were entered for the “Destination” installation;
- an existing VLM6-VLM6 interface file was selected with the intention of re-using it, perhaps because most of the input and output mnemonics are what you want.



You have two options:

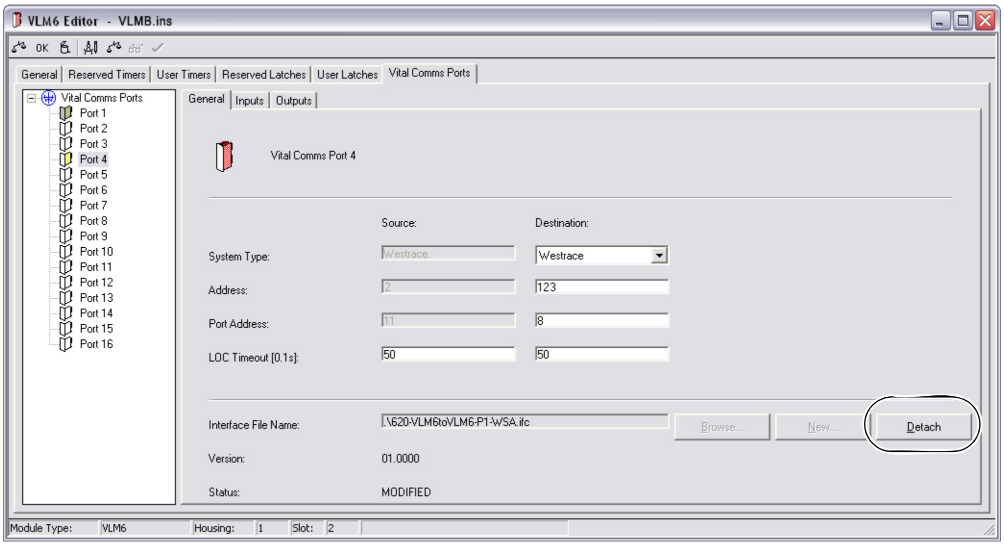
- Detach the interface file and select the correct one;
- Select one of the possible actions on the Incorrect Installation Address dialog box:
  - Update source system;
  - Update destination system information;
  - Don't update the interface file.

Take care when choosing one of the latter options. Thoroughly check the interface and installation files to ensure you get what you expect.

Changing the VLM6-VLM6 Interface File

a) Select the **Detach** button to disassociate the interface file from the VLM Installation.

b) Select or create an interface file as described above.



5.2.7.2 Checking and Approving VLM6-VLM6 Interface Files

The VLM6-VLM6 interface file can be checked and approved in either of the associated VLM installations.

Consistency checking can be done at any time during the development of the interface file. It **must** be done at the completion of interface file development and **before** a Check record is added.

An authorized Signalling Engineer (see table 3.5 on page 3-15) **must** check the input and output mnemonics plus their initial states **before** adding a Check record.

An authorized Signalling Engineer (see table 3.5 on page 3-15) **must** raise the VLM6-VLM6 interface file to the “Approved” status **before** the associated VLM6 installations can be raised to the “Approved” status. Refer to section 7.1.1.3 to ensure you are checking and approving the interface file at the correct time.

Note: ***Changing the status of the interface file may also affect the current status of the associated VLM6 installation files. See section 7.1.2.***

Select in turn and respond to the related dialog boxes:

- **Check Consistency**;
- **Add Check record**;

  GCSS adds a Check record to the interface file and the file is saved. A copy of the interface file is added to the Log file and the Log file is saved.
- **Add Approval record**.

  GCSS adds an Approval record to the interface file and the file is saved. A copy of the interface file is added to the Log file and the Log file is saved.

5.2.8 NVLM modules and VLM-NVLM Interface Files—VLM Installations

The VLM-NVLM interface file defines the mnemonics passed between the VLM and the NVLM within the same WESTRACE application.

Figure 5.18 illustrates how data from the VLM-NVLM interface file is combined with data from the VLM Installation file by the GCSS compiler to produce VLM PROM data.

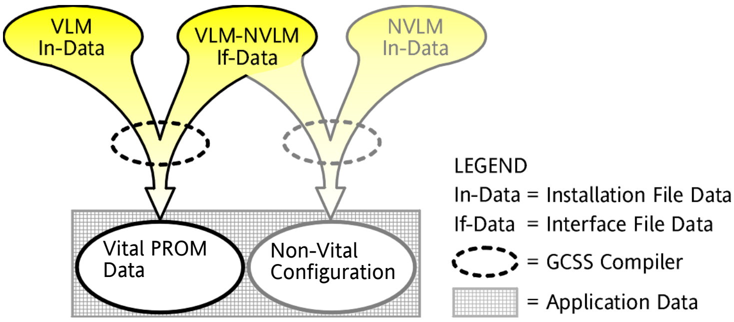


Figure 5.18 VLM-NVLM interface file and VLM Installation data—merging

The VLM-NVLM interface file can be an existing file or a newly created file.

It contains:

- VLM type and hot-standby or stand-alone configuration;
- NVLM address with the VLM Installation;
- NVLM Module Sequence Number within the VLM Installation;
- Inputs and outputs shared between the VLM and NVLM installations. See *VLM-NVLM Interface File Inputs and Outputs* below;
- Configuration control history of the interface file itself.

VLM-NVLM Interface File Inputs and Outputs

The input and output mnemonics are always seen from the point of view of the current installation:

- When viewed, entered or edited through the NVLM in the VLM installation:
  - inputs are for information received from the associated NVLM installation;
  - outputs are for information sent to the associated NVLM Installation;
- When viewed, entered or edited through the NVLM Installation:
  - inputs are for information received from the associated VLM installation;
  - outputs are for information sent to the associated VLM Installation.

5.2.8.1 Creating, Selecting or Changing VLM-NVLM Interface Files

Although a VLM-NVLM interface file can be created and edited, or selected, attached and then edited in either of the associated VLM and NVLM installations, the recommended procedure is to:

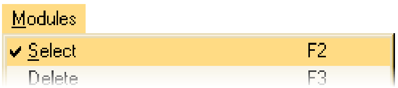
- create and edit the VLM-NVLM interface file by editing the NVLM module in the VLM installation, and then;
- select and attach the interface file to the NVLM module when editing the NVLM installation.

Once created and correctly edited in one installation, the interface file need only be selected and attached in the other installation.

Note: ***The same VLM-NVLM interface file must be attached to both associated VLM and NVLM installations.***

With the Housing Editor for the VLM installation active:

a) Choose **Select**.



Alt m s
F2

b) Double-click on the NCDM or NVC/DM module to open the Module Editor.

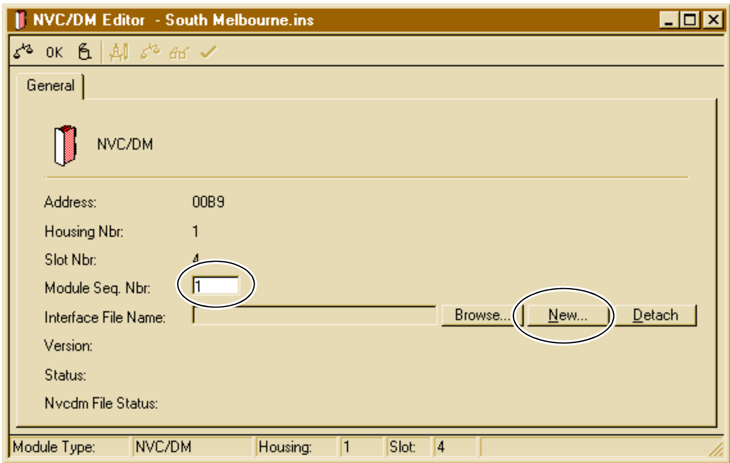
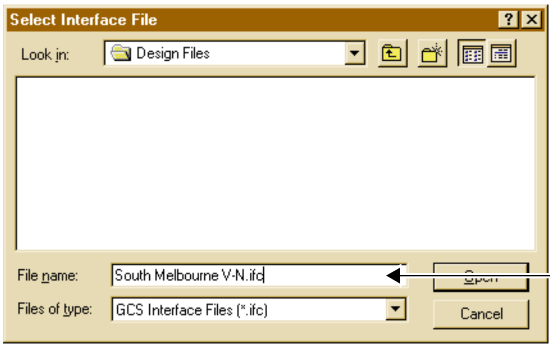


Figure 5.19 NVLM Module Editor—VLM installation

c) Ensure the Module Sequence Number is entered correctly. NVLM modules are always 1 (see table 5.1).

5.2.8.1.1 Creating a New VLM-NVLM Interface File

a) Select the **New** button (see figure 5.19) to open the Select Interface dialog box.



Enter name for new interface file here

b) Enter a name for the VLM-NVLM interface file.

c) Select the **Open** button.

d) Edit the input and output mnemonics as required. See section 5.2.8.2.

5.2.8.1.2 Selecting an Existing VLM-NVLM Interface File

See section 5.3.9.1.2.

5.2.8.1.3 Changing the VLM-NVLM Interface File

See section 5.3.9.1.3.

5.2.8.2 Editing VLM-NVLM Interface File Inputs and Outputs

Access to the interface file input and outputs is through Input and Output tabs. See figure 5.20.

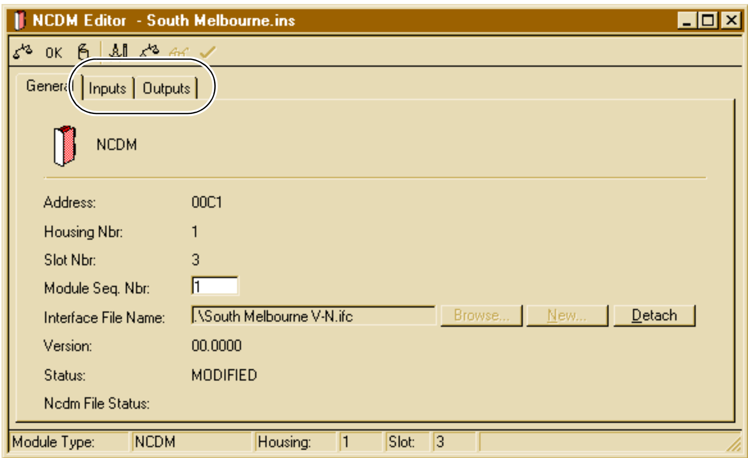


Figure 5.20 NVLM Module Editor—VLM-NVLM interface file selected—VLM installation

Select these tabs in turn to *set the interface file inputs and outputs from the point of view of the VLM installation*. See *VLM-NVLM Interface File Inputs and Outputs* on page 5-30.

For each input or output you will need:

- a valid mnemonic name (see section 2.4);
- the initial state, either high or low (see section 2.4.1).

See section 5.1.6.3 for assistance in editing the list of mnemonics.

See Appendix A, pages A-17 and A-24 for further assistance.

5.2.8.3 Checking and Approving VLM-NVLM Interface File

Consistency checking can be done at any time during the development of the interface file. It **must** be done at the completion of interface file development and **before** a Check record is added.

An authorized Signalling Engineer (see table 3.5 on page 3-15) **must** check the input and output mnemonics plus their initial states **before** adding a Check record.

An authorized Signalling Engineer (see table 3.5 on page 3-15) **must** raise the VLM-NVLM interface file to the “Approved” status **before** the associated VLM and NVLM installations can be raised to the “Approved” status. Refer to sections 7.1.1.2 and 7.1.1.3 to ensure you are checking and approving the interface file at the correct time.

Note: ***Changing the status of the interface file may also affect the current status of the associated VLM and NVLM installation files. See section 7.1.2.***

The VLM-NVLM interface file can be checked and approved in either of the associated VLM and NVLM installations; however, Invensys Rail recommends the VLM-NVLM interface file be checked and approved via the NVLM in the VLM installation **after** it has been selected and attached to the NVLM installation.

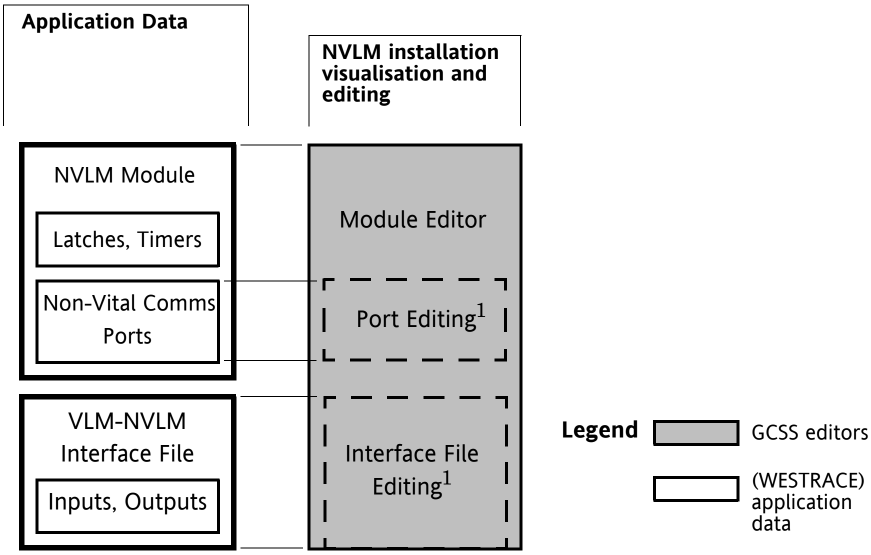
Select in turn and respond to the related dialog boxes:

- **Check Consistency**;
- **Add Check record**;

  GCSS adds a Check record to the interface file and the file is saved. A copy of the interface file is added to the Log file and the Log file is saved.
- **Add Approval record**.

  GCSS adds an Approval record to the interface file and the file is saved. A copy of the interface file is added to the Log file and the Log file is saved.

5.3 Module Editing—NVLM Installations



1. Port and interface file editing is via a child window opened from a tab sheet within the Module Editor window. See sections section 5.3.8 and section 5.3.9.

Figure 5.21 Module editing—NVLM Installation

Summary of Module Editing

General editing of the NVLM module in an NVLM installation comprises the following basic steps:

a) Complete the NVLM Installation window data fields. See section 5.3.1.

b) Activate the module editor. See section 5.3.2.

c) Set general parameters, ports, user timers, time-of-day timers, user latches and reset latches as required and assign a VLM-NVLM interface file. See section 5.3.3.

d) Check module data validity. See section 5.3.4.

e) Save the module data. See section 5.3.5.

f) Print the Module Editor Report. See section 5.3.6.

g) Close the module editor. See section 5.3.7.

h) Save the NVLM installation. See section 3.10.

5.3.1 NVLM Installation Window

a) Complete these fields as required:
- Installation name
- Installation address
- Customer name
- Initialization time, and
- Comments

b) Ensure the installation address is identical to the installation address assigned to the associated VLM installation.

c) Ensure the “Comments” fields are filled correctly when the NVLM is an NCDM supporting network communications. See section 2.6.4.4.

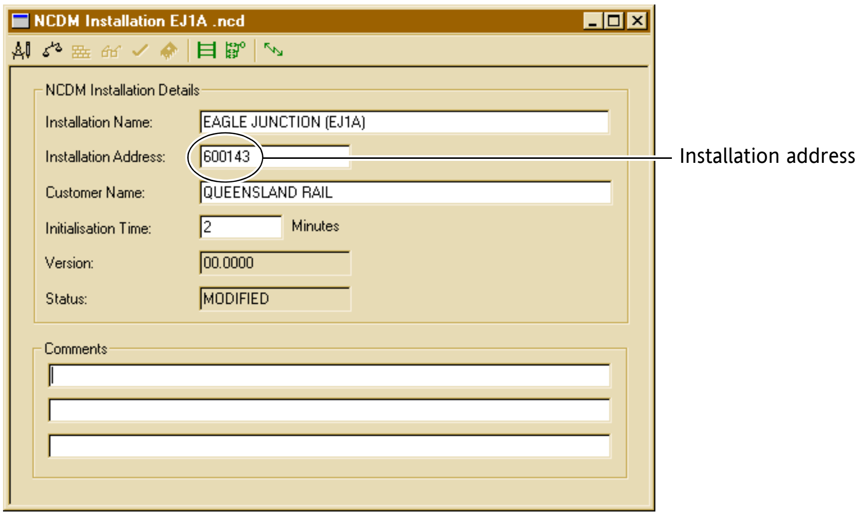


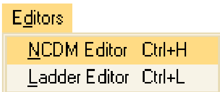
Figure 5.22 NVLM Installation window—typical

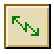
5.3.2 Module Editor Activation

The only module edited in an NVLM installation is the NCDM or NVC/DM. The Module Editor title bar shows the name of the non-vital module being edited.

a) Select the NVLM installation window to make it active.

b) Select **NCDM Editor** or **NVC/DM Editor** as appropriate.

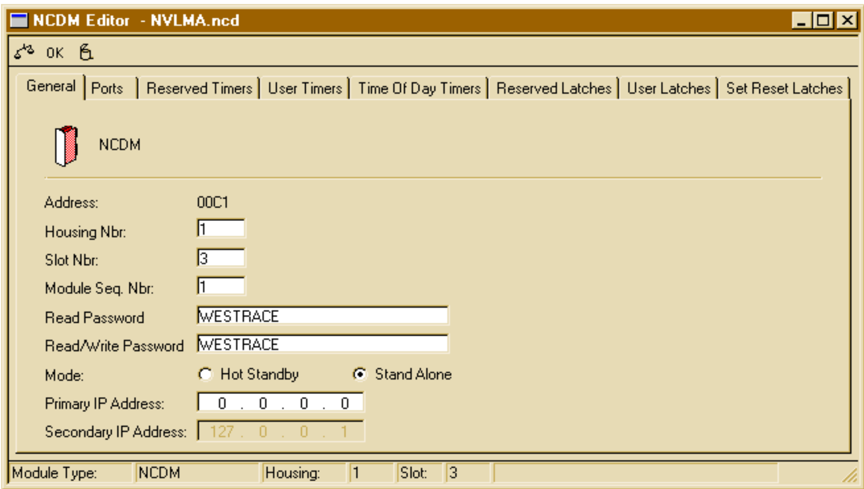




Alt d n
Ctrl+h

The example NVLM installation in this section is for an NCDM.

5.3.3 General NVLM Editing—NVLM Installations



For each NVLM there are tabs for:

- *General parameters*—can be edited. See Appendix A, pages A-17 and A-24;
- *Ports*—can be edited. See section 5.3.8. A VLM-NVLM interface file must be assigned to the NVLM module. See sections 5.3.8 and 5.3.9;
- *Reserved Timers*—read only. Cannot be edited. See sections 2.4.6.2 and 2.4.6.3;
- *User Timers*—can be edited. See Appendix A, pages A-23 and A-28;
- *Time Of Day Timers*—can be edited. See Appendix A, pages A-23 and A-28;
- *Reserved Latches*—read only. Cannot be edited. See sections 2.4.5.2 and 2.4.5.3;
- *User Latches*—can be edited. See section 2.5.3 and Appendix A, pages A-23 and A-28;
- *Set Reset Latches*—can be edited. See section 2.5.4 and Appendix A, pages A-23 and A-28.

Select each editable tab in turn and:

- add or remove mnemonics as appropriate;
- set values and other parameters.

See section 5.1.6.3.5 for assistance.

5.3.4 NVLM Module Data Validity Check

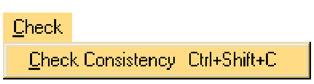
This procedure should be performed when editing is complete and before the module editor is closed. It could also be advantageous to check module data validity at selected stages while entering and editing module mnemonics.

This procedure:

- performs data validity checks on the module information set;
- checks the ranges of all numerical fields;
- checks that the module information forms a consistent set;
- checks that an interface file is attached to the VLM port (if a VLM interface is defined).

Any inconsistencies are reported in an alert message box.

Select **Check Consistency**.



Alt c c
Ctrl+Shift+c

Note: ***This action will not find duplicated sequence numbers. They are found by the Installation Consistency Check. See section 7.2.***

5.3.5 Saving Module Data

These editors work on a copy or image of the module data and only writes over the original module data when properly closed. See figure 5.2.

Select the **OK** button on the Ports Editor toolbar and then the Module Editor toolbar before saving the installation.

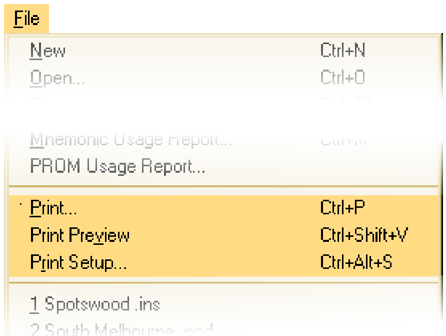
OK

An appropriate warning message is displayed if the Module Editor is open when the installation is saved or closed with unsaved edits.

5.3.6 Module Editor Report

This printed report replicates the layout of the Module Editor window. If any list box is too wide to fit on a single page, it is continued on subsequent pages. In these circumstances, the column containing the row number is reproduced on each page.

a) Select **Print** or **Print Preview** to display the Print Module dialog box.



Alt f v
Ctrl+p

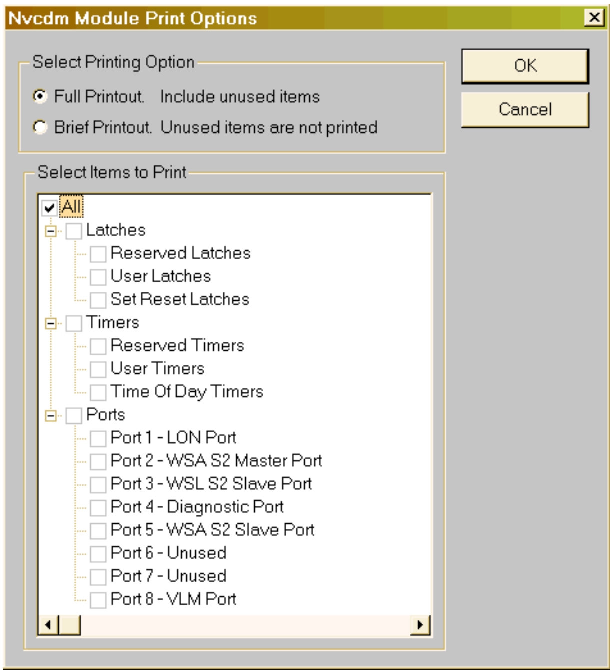


Figure 5.23 Print Module dialog box

b) Select the type of report required:

Full Printout—shows all list box fields in the Module Editor window regardless of whether they contain data or not.

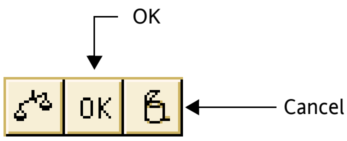
Brief Printout—only shows those list box fields that contain data Use this option unless you specifically need the blank fields included in the full printout.

c) Select items to print by clicking the appropriate check boxes.

The resulting report will be similar to the example in figure B.15 on page B-19.

Note: ***The examples shown by figures B.14 and B.15 were generated for the same NVC/DM module in matching VLM and NVLM installations.***

5.3.7 Closing the Module Editor



Close the editor window by clicking the:

- **OK** button to:
  - perform data validity check;
  - save the data and close the editor if there are no inconsistencies.

    Otherwise inconsistencies are reported and the editor remains open.
- **Cancel** button (or **Window Close** button) to:
  - abandon any changes made to the data.
  - close the editor after confirmation.

5.3.8 Port Editing—NVLM Installations

Select the Port tab from the Module Editor to display the available ports for the NVLM.

The NCDM supports four ports; all can be configured for a choice of port types in any order.

The VLM-NVLM interface file can be assigned to any port. The port number on the GCSS user interface (see figure 5.24) is not directly associated with any physical port, however the lowest numbered serial configured port on the GCSS user interface maps to physical serial port 2 on the NCDM NCD PFM. The other serial configured port (if any) maps to physical serial port 3 on the NCDM NCD PFM.

The NVC/DM supports eight ports; only six can be configured for a choice of port types:

- Port 1 is reserved;
- Port 8 is dedicated to the VLM interface. Port 8 is where you attach the VLM-NVLM interface file;
- Ports 2, 3, 4, 5, 6, and 7 on the GCSS user interface (see figure 5.24) map directly to the same numbered physical serial ports on the NVC/DM CIMPM.

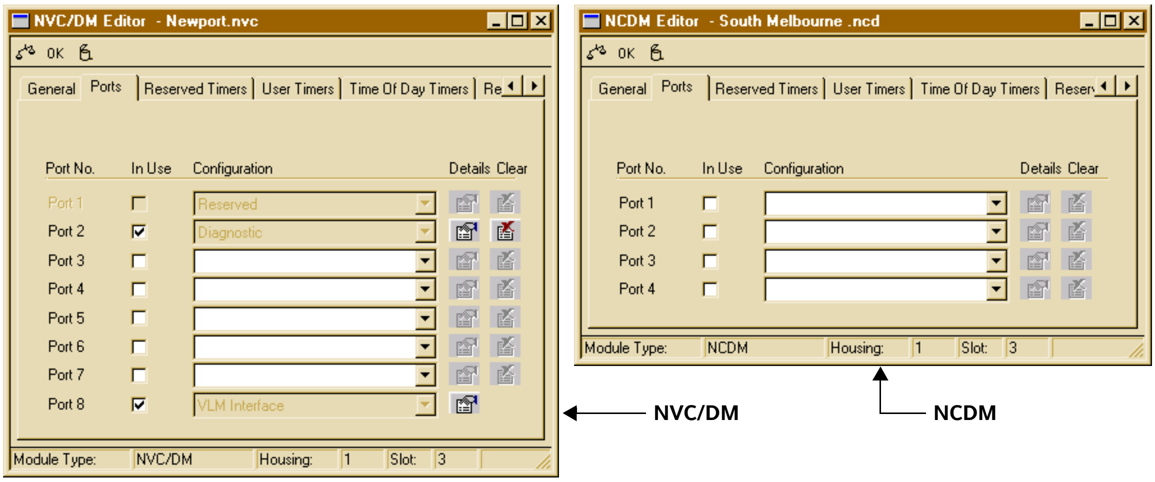


Figure 5.24 NVLM Module Editor—Ports tab

Figure 5.24 shows the Port tab sheet for the NCDM and NVC/DM modules. Figure 5.25 identifies the functional features of each port field.

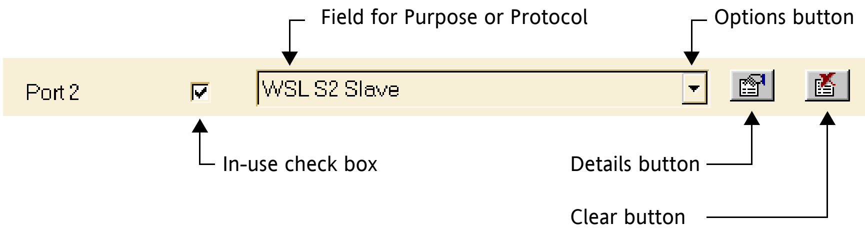


Figure 5.25 Port buttons and In-use check box

In-use check-box—read only indicator of port status. Shows a tick when one or more of the port properties has changed.

Field—shows the purpose or protocol supported by the port. Filled by selecting the Options button.

Options button—select to display and choose from the available function and protocol options. See section 5.3.8.1.

Details button—select to open the Port Editor for the port.

Clear button—select to clear the current configuration of the port. Also makes the port inactive.

A “greyed out” port indicates it cannot be changed until the **Clear** button is used.

Caution: ***When cleared, all data for that port is lost.***

5.3.8.1 Port Function and Protocol Options

PORT	Diagnostic	WSL/S2 (Server)[1]	WSA/S2 (Server)[1]	WSA/S2 (Client)[1]	VLM Interface
1 (Reserved)					
2 (serial)	✓	✓	✓	✓	
3 (serial)	✓	✓	✓	✓	
4 (serial)	✓	✓	✓	✓	
5 (serial)		✓	✓	✓	
6 (serial)		✓	✓	✓	
7 (serial)		✓	✓	✓	
8 (IMB)					✓

Table 5.2 Port Function and Protocol Options—NVC/DM

1 Server = Slave or Field, Client = Master or Office

Port	Diagnostic (Serial)[1]	WSL S2 Server (Serial)[1]	WSA S2 Client (Serial)[1]	WSA S2 Server (Serial)[1]	VLM Interface	Network				
						Diagnostic	WSL S2 Server[2]	WSA S2 Server[2]	WSA S2 Client[2]	WESTRACE Vital
1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
3	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 5.3 Port functions and protocol options—NCDM

1 NCDM modules have only two physical serial ports, therefore only two serial protocols can be in use at any time.

2 Server = Slave or Field, Client = Master or Office

5.3.8.2 Set Up a Port—NVLM Installation

a) Select the **Options** button and then select from the drop-down list.

b) Select the **Details** button to invoke the Port Editor.

The Port Editor interface is similar to the Module Editor. It has Tab sheets and often a multi-level tree structure. See figure 5.26 for an example.

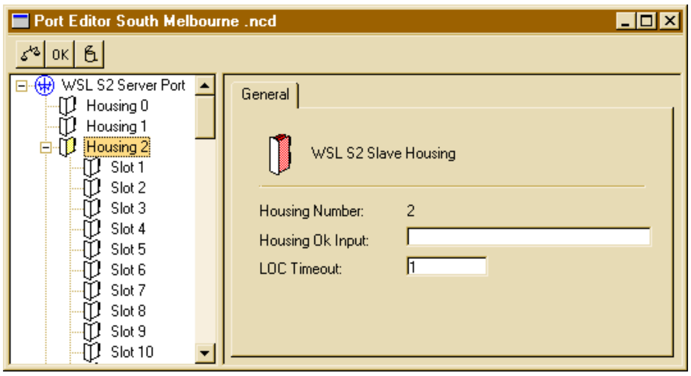


Figure 5.26 Setup a port—example—NVLM Installation

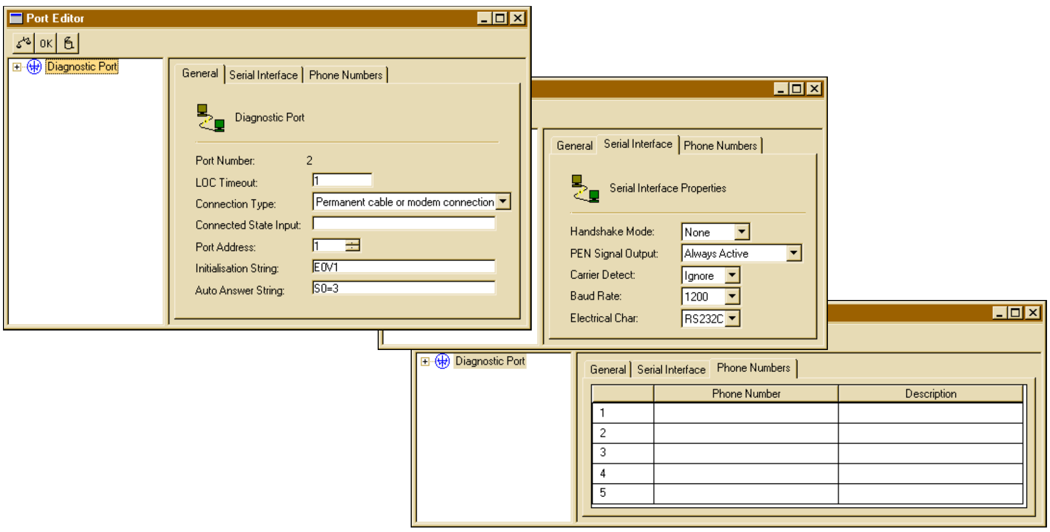
c) Select each tree level (as appropriate) and each editable tab in turn and:
- add or remove mnemonics as appropriate (see sections 2.4 and 5.1.6);
- set values and other parameters (see sections 5.3.8.2.1 and 5.3.8.2.2 below).

Note: ***Right-click with the mouse on the Session icon to display then select protocols for NCDM network sessions. See section 5.3.8.2.1.***

5.3.8.2.1 NCDM Protocols

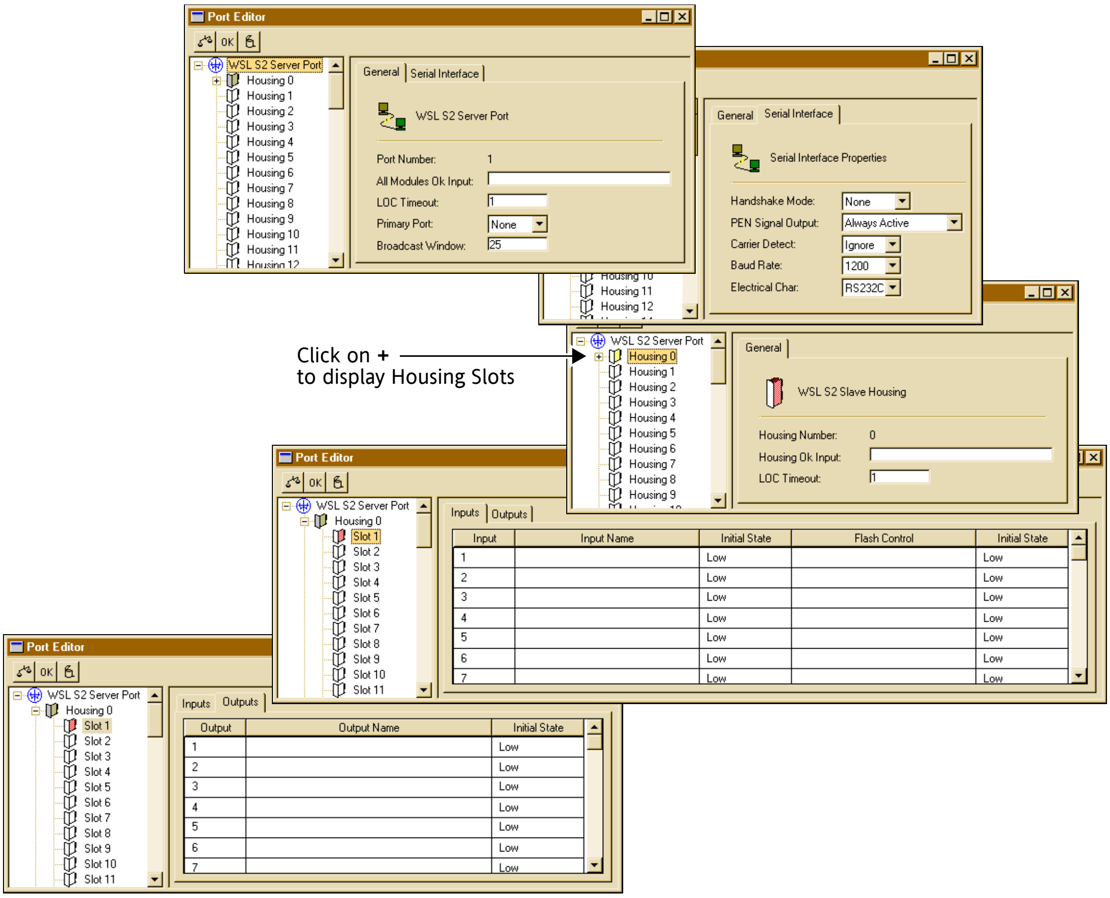
Terminology: Server = Slave or Field, Client = Master or Office

Diagnostic (Serial)



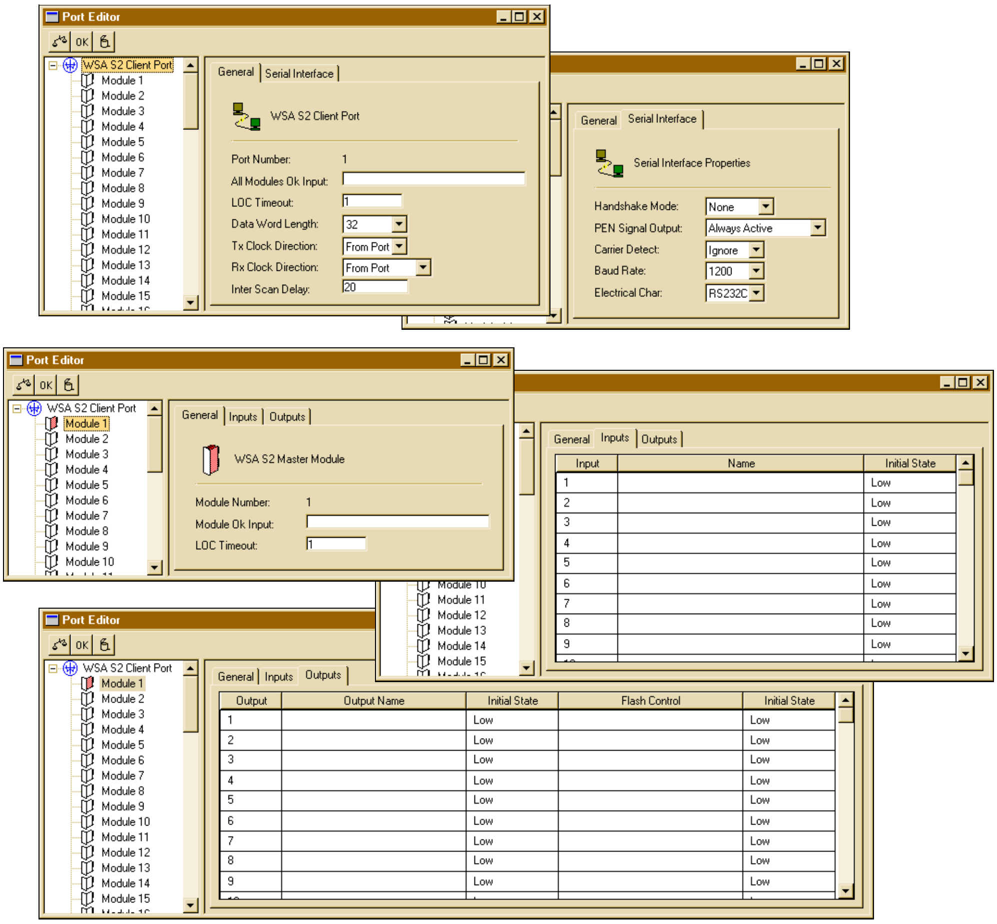
Also see section 2.6.4.6.1 and Appendix A, page A-18.

WSL S2 Server (Serial)



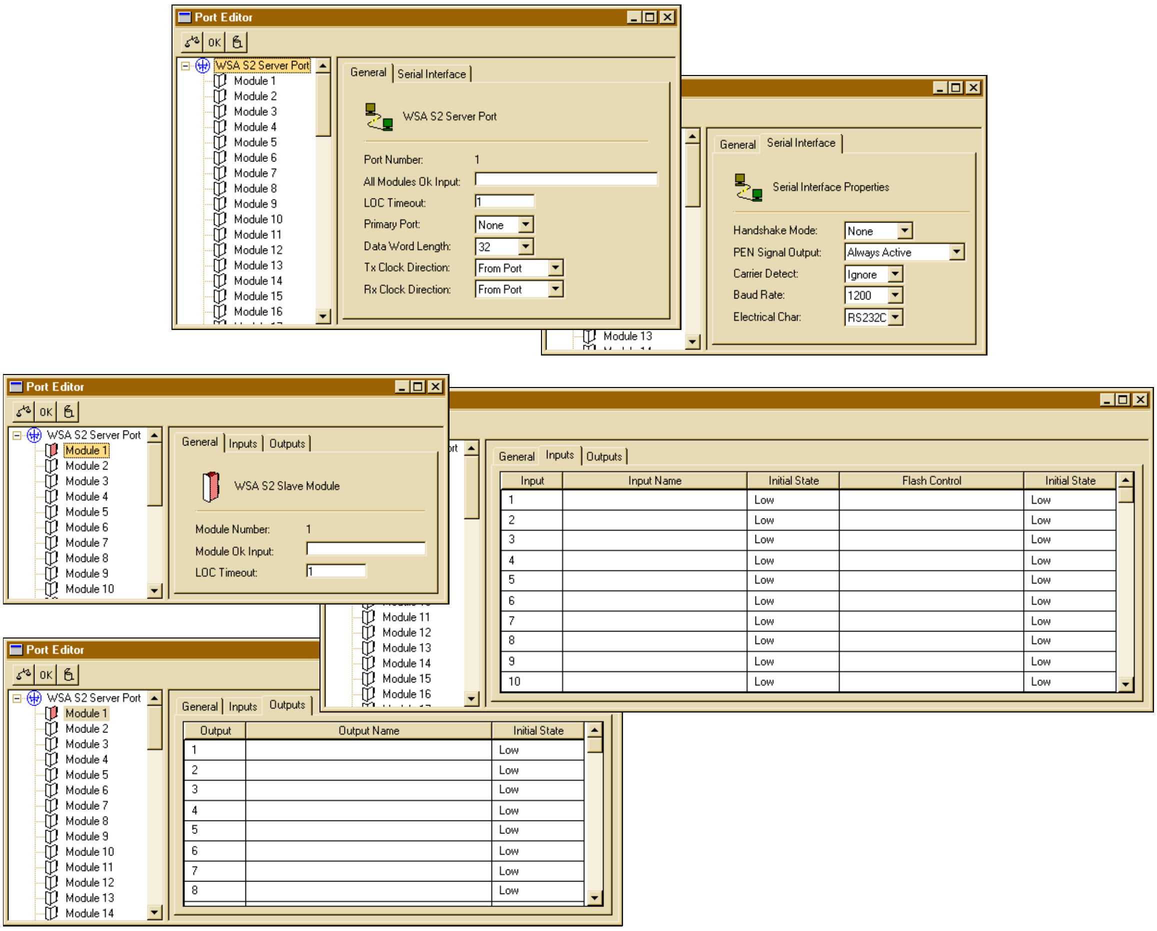
Also see sections 2.6.4.6.2, 2.6.4.6.3 and Appendix A, page A-18.

WSA S2 Client (Serial)



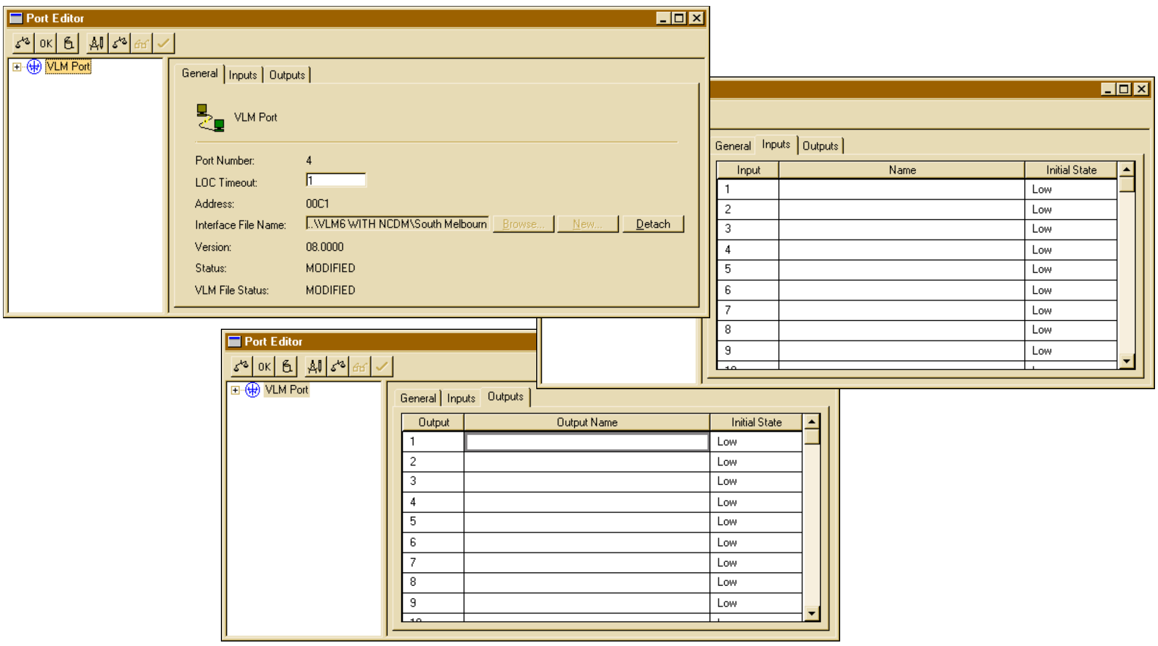
Also see sections 2.6.4.6.2, 2.6.4.6.5 and Appendix A, page A-19.

WSA S2 Server (Serial)



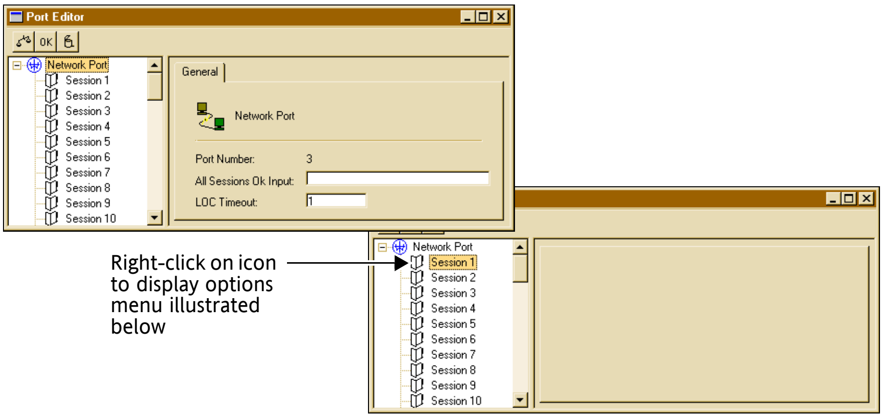
Also see sections 2.6.4.6.2, 2.6.4.6.4 and Appendix A, page A-20.

VLM Interface



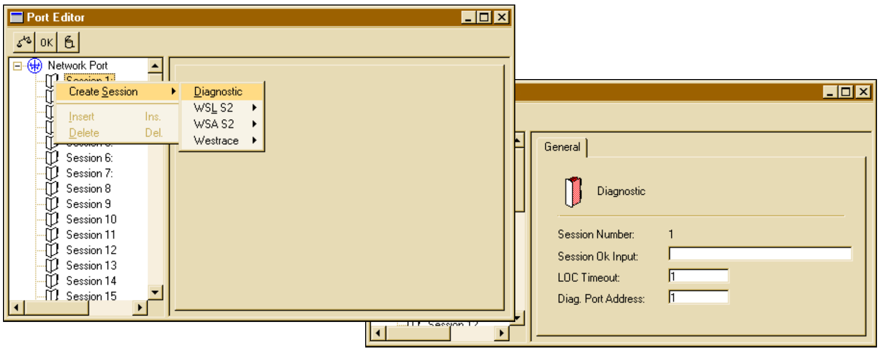
Also see section 5.3.9.1 and Appendix A, page A-21.

Network



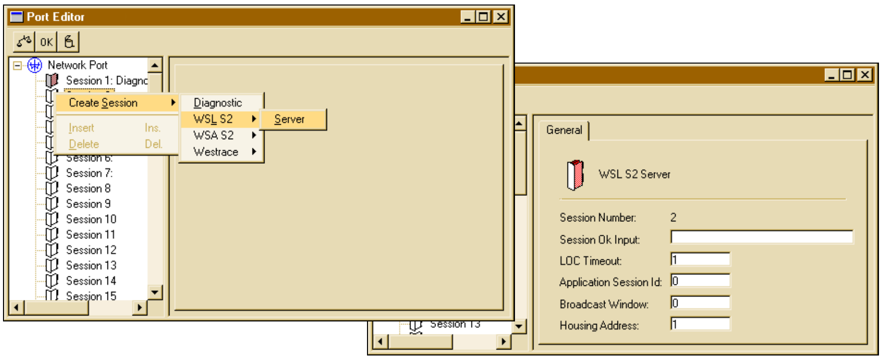
Also see section 2.6.4.4 and Appendix A, page A-21.

Diagnostic—Session



Also see section 2.6.4.4.1 and Appendix A, page A-21.

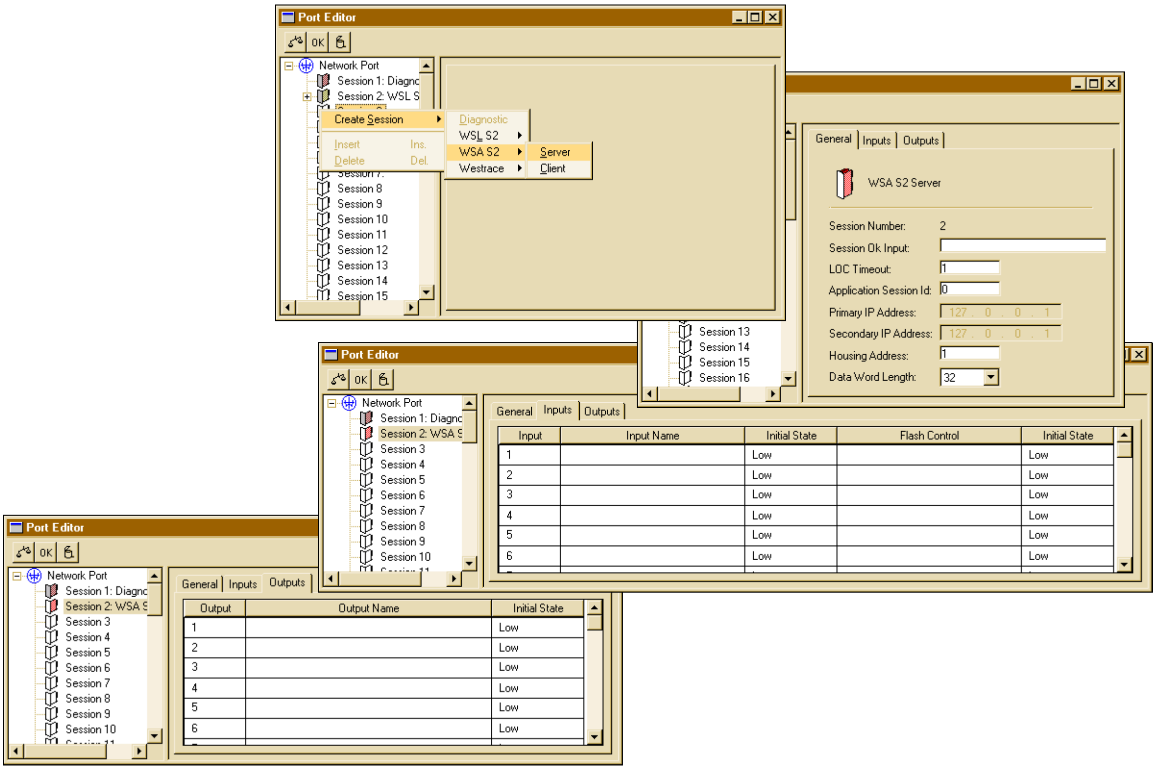
WSL S2 Server—Session



Also see section 2.6.4.4.2 and Appendix A, page A-21.

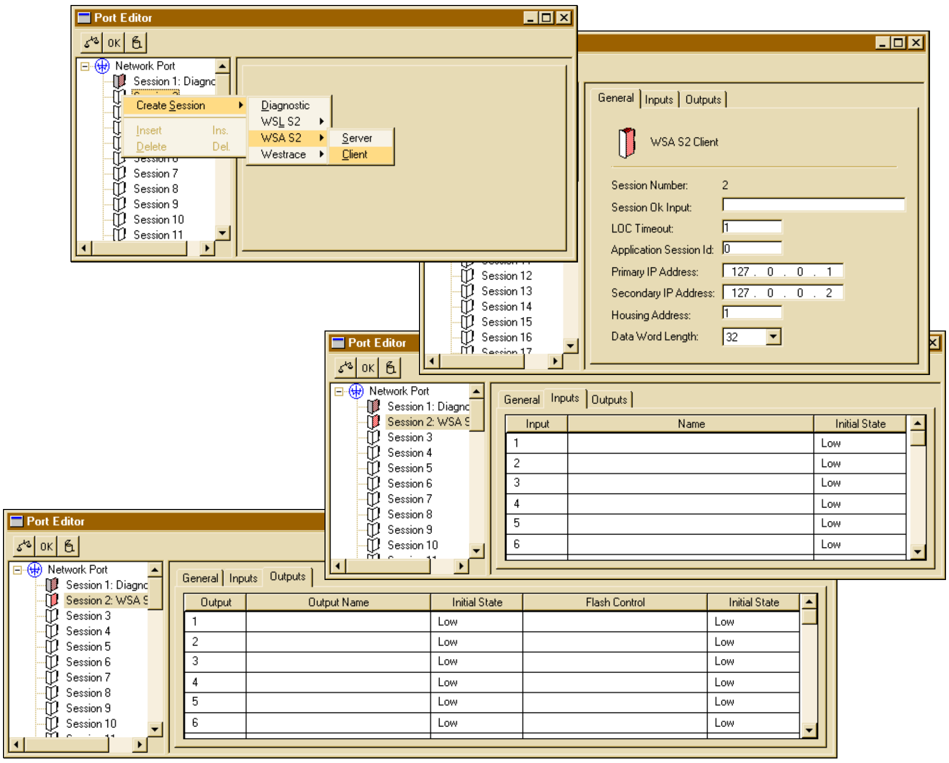
Note: ***Each diagnostic and telemetry session must have a unique port address within an installation. Do not use the same address for diagnostic and telemetry.***

WSA S2 Server—Session



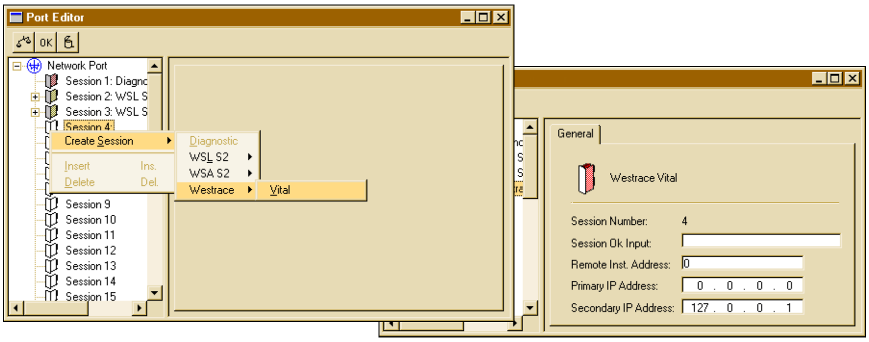
Also see section 2.6.4.4.3 and Appendix A, page A-22.

WSA S2 Client—Session



Also see section 2.6.4.4.4 and Appendix A, page A-22.

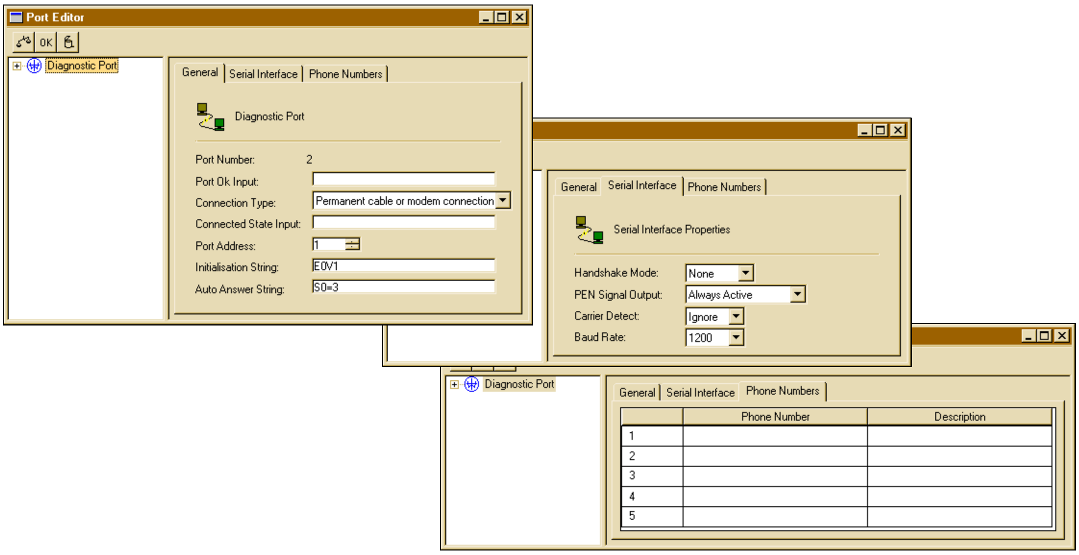
WESTRACE Vital—Session



Also see section 2.6.4.4.5 and Appendix A, page A-23.

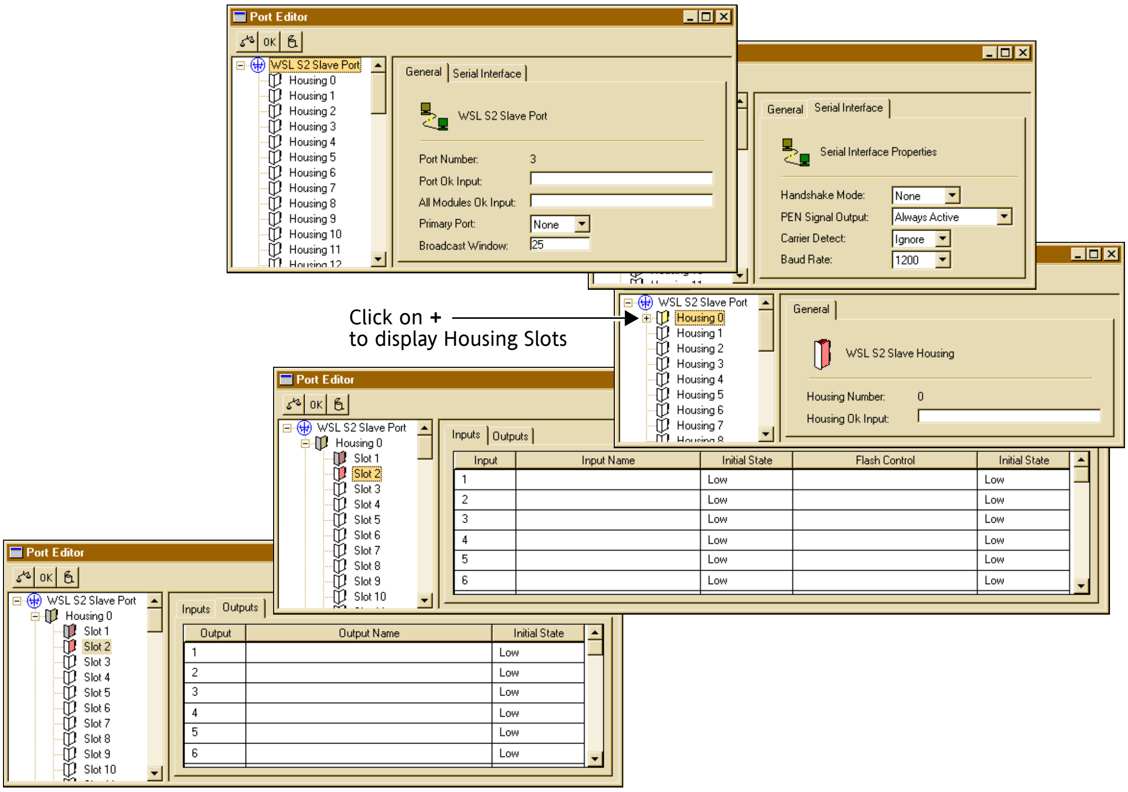
5.3.8.2.2 NVC/DM Protocols

Diagnostic (Serial)



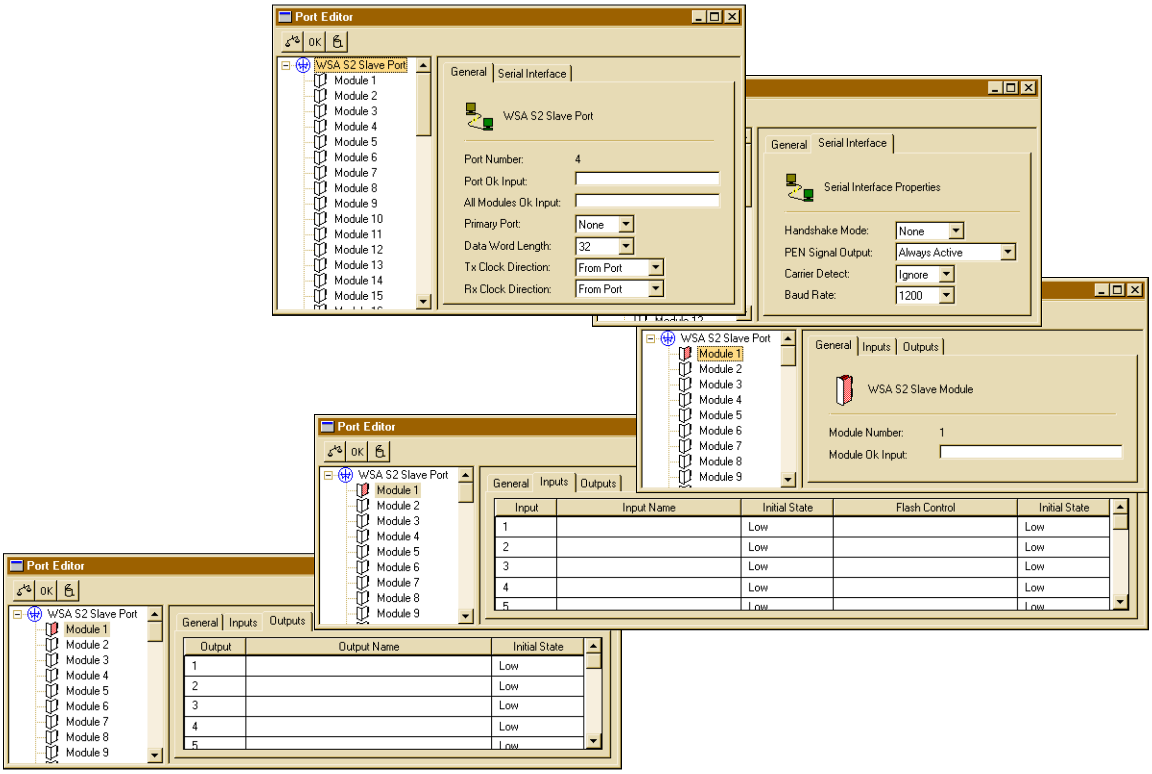
Also see section 2.6.7.4.1 and Appendix A, page A-24.

WSL/S2 Slave (Serial)



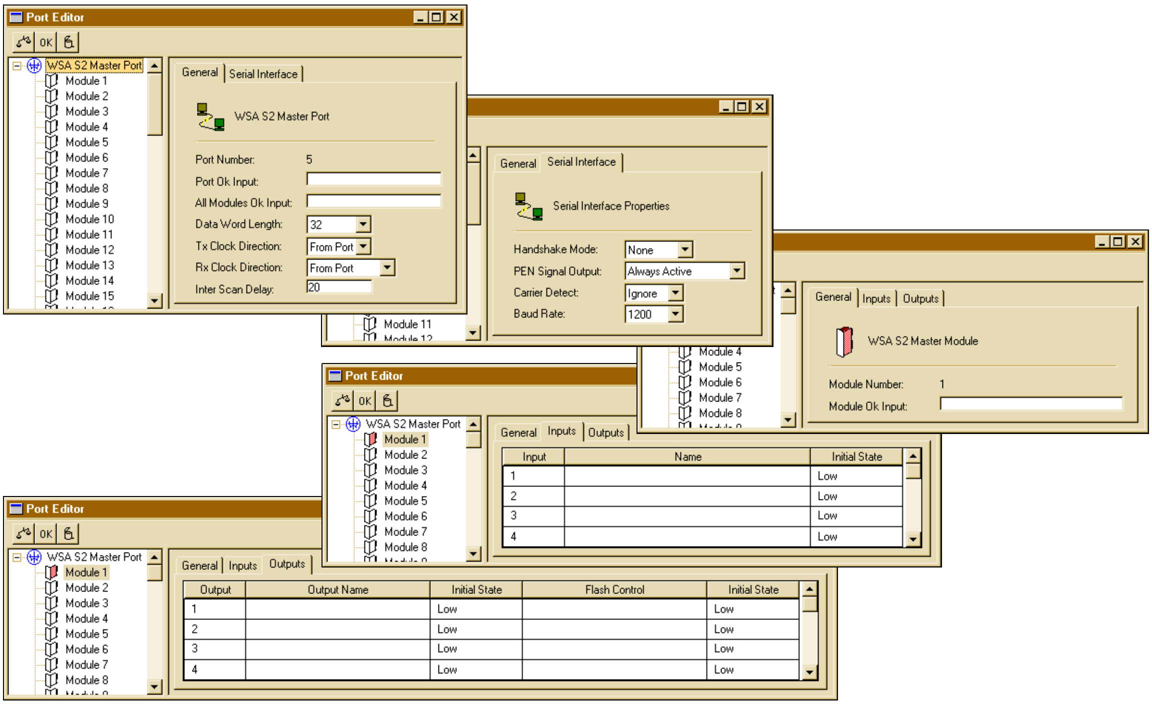
Also see sections 2.6.7.4.2, 2.6.7.4.5 and Appendix A, page A-25.

WSA/S2 Slave (Serial)



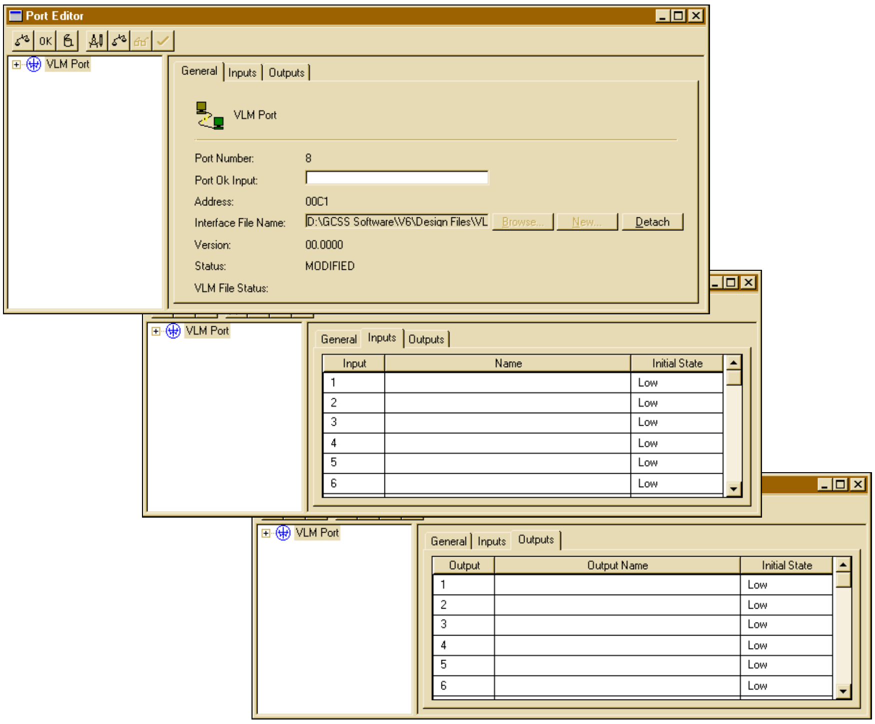
Also see sections 2.6.7.4.2, 2.6.7.4.5 and Appendix A, page A-26.

WSA/S2 Master (Serial)



Also see sections 2.6.7.4.2, 2.6.7.4.4 and Appendix A, page A-26.

VLM Interface (Serial)



Also see section 5.3.9.1 and Appendix A, page A-27.

5.3.9 VLM-NVLM Interface Files—NVLM Installations

The VLM-NVLM interface file defines the mnemonics passed between the VLM and the NVLM within the same WESTRACE application.

Figure 5.27 illustrates how data from the VLM-NVLM interface file is combined with data from the NVLM Installation file by the GCSS compiler to produce Non-Vital Configuration.

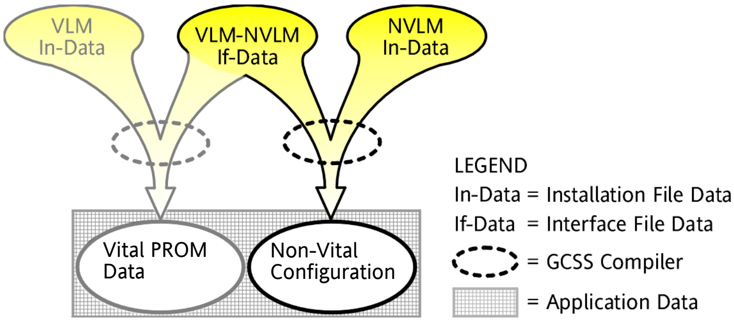


Figure 5.27 VLM-NVLM interface file and NVLM Installation data—merging

The VLM-NVLM interface file can be an existing file or a newly created file. It contains:

- VLM type and hot-standby configuration;
- NVLM address with the VLM Installation;
- NVLM Module Sequence Number within the VLM Installation;
- Inputs and outputs shared between the VLM and NVLM installations. See *VLM-NVLM Interface File Inputs and Outputs* below;
- Configuration control history of the interface file itself.

VLM-NVLM Interface File Inputs and Outputs

The input and output mnemonics are always seen from the point of view of the current installation:

- When viewed, entered or edited through the NVLM Installation:
  - inputs are for information received from the associated VLM installation;
  - outputs are for information sent to the associated VLM Installation;
- When viewed, entered or edited through the NVLM in the VLM installation:
  - inputs are for information received from the associated NVLM installation;
  - outputs are for information sent to the associated NVLM Installation.

5.3.9.1 Creating, Selecting or Changing VLM-NVLM Interface File

Although a VLM-NVLM interface file can be created and edited, or selected, attached and then edited in either of the associated VLM and NVLM installations, the recommended procedure is to:

- create and edit the VLM-NVLM interface file by editing the NVLM module in the VLM installation, and then;
- select and attach the interface file to the NVLM module when editing the NVLM installation.

Once created and correctly edited in one installation, the interface file need only be selected and attached in the other installation.

Note: ***The same VLM-NVLM interface file must be attached to both associated VLM and NVLM installations.***

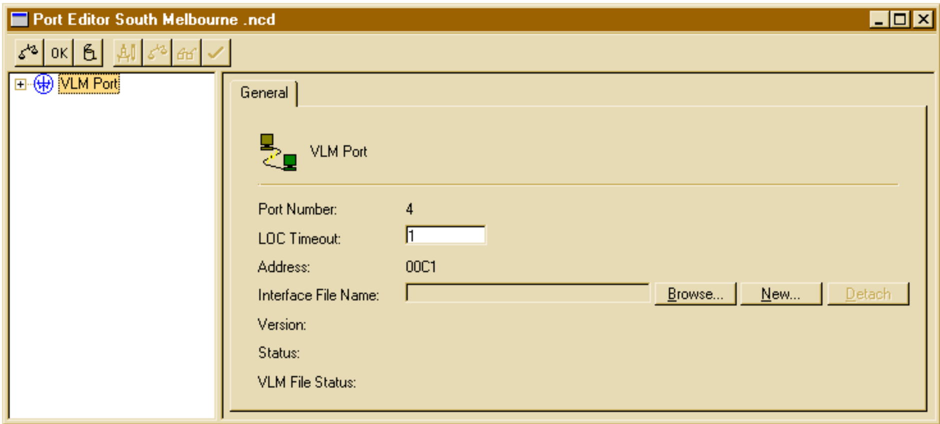


Figure 5.28 NVLM Port Editor—NVLM Installation

5.3.9.1.1 Creating a New VLM-NVLM Interface File

See section 5.2.8.1.1.

5.3.9.1.2 Selecting an Existing VLM-NVLM Interface File

a) Select the **Browse** button to open the Select Interface dialog box.

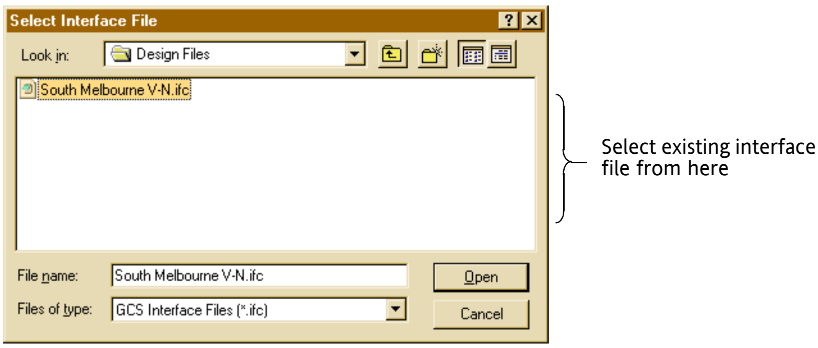


Figure 5.29 Interface file—selecting an existing file

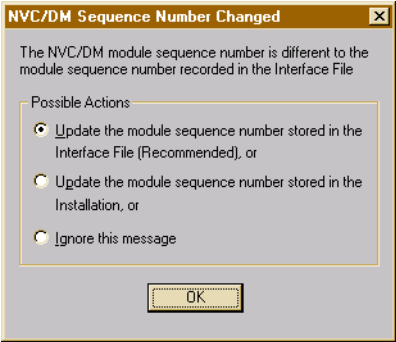
b) Select a file from the dialog box.

c) Select the **Open** button.

d) Edit the input and output mnemonics as required. See section 5.2.8.2.

Installation File Errors

An error will occur if the module sequence number for the NVLM stored in the interface file is different to the number in the current installation (see figure 5.19).



You have two options:

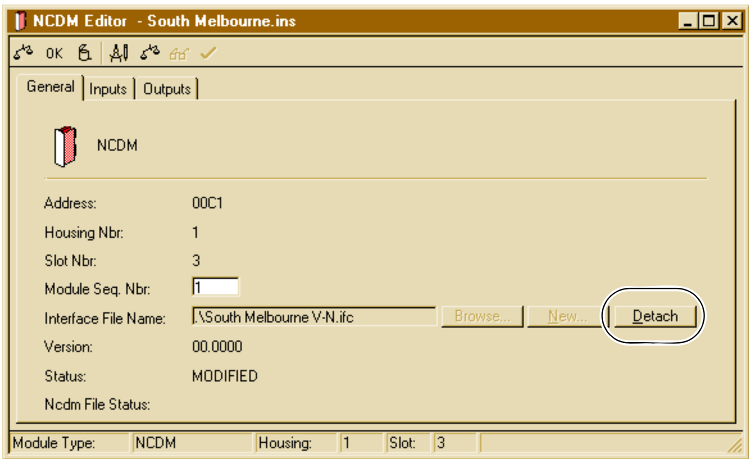
- Detach the interface file and select the correct one.
- Select one of:
  - Update the module sequence number stored in the interface file;
  - Update the module sequence number stored in the installation;
  - Ignore the message (not recommended).

Take care when choosing one of the latter options. Thoroughly check the interface and installation files to ensure you get what you expect. Remember, the interface file contains input and output mnemonics passed between the VLM and NVLM modules.

5.3.9.1.3 Changing the VLM-NVLM Interface File

a) Select the **Detach** button to disassociate the interface file from the installation.

b) Select or create an interface file as described above.



5.3.9.2 Editing VLM-NVLM Interface File Inputs and Outputs

Note: ***It should not be necessary to edit the VLM-NVLM interface file here if it was correctly created and edited as described in section 5.2.8.2.***

Access to the interface file input and outputs is through tabs adjacent to the General tab. See figure 5.30.

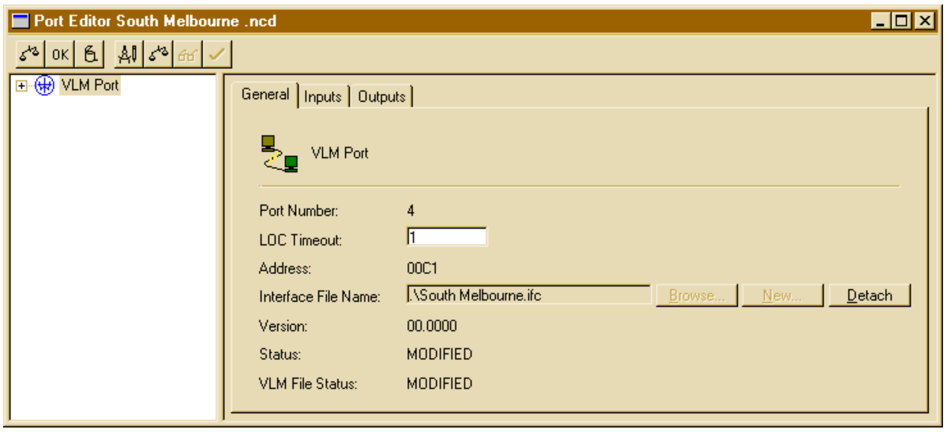


Figure 5.30 NVLM Port Editor—NVLM Installation—VLM-NVLM interface file selected

Use these tabs sheets to *set the interface file inputs and outputs from the point of view of the HVLM installation*. See *VLM-NVLM Interface File Inputs and Outputs* on page 5-51.

For each input or output you will need:

- a valid mnemonic name (see section 2.4);
- the initial state, either high or low (see section 2.4.1).

See section 5.1.6.3 for assistance in editing the list of mnemonics.

See Appendix A, pages A-17 and A-24 for further assistance.

5.3.9.3 Checking and Approving the Interface File

Invensys Rail recommends the VLM-NVLM interface file be checked and approved via the NVLM in the VLM installation **after** it has been selected and attached to the NVLM installation.

See section 5.2.8.3.

6. LADDER LOGIC

This chapter describes the Ladder Logic Editors for VLM and NVLM installations, and introduces GCSS templates.

6.1 Overview

6.1.1 Ladder Logic Editors

All Ladder Logic Editors operate in the same manner but are differentiated by the colour of the ladder:

- the VLM editors use black;
- the NVLM editors use green.

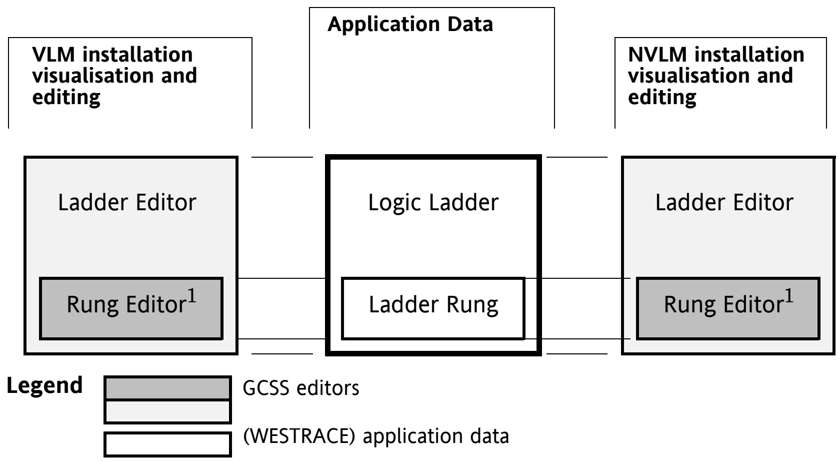
A **Ladder Logic Editor** comprises three parts:

- the Ladder Editor;
- the Rung Viewer (or Locked Rung Viewer for locked rungs);
- the Rung Editor.

Use the:

- **Ladder Editor** for adding and deleting rungs and for getting an overview of the entire ladder logic (see section 6.2);
- **Rung Viewer** to view the underlying logic of *all unlocked* ladder rungs in an installation (see section 6.4), or **Locked Rung Viewer** to view the underlying logic of a single locked ladder rung (see section 6.5);
- **Rung Editor** to design vital ladder logic for the VLM and non-vital ladder logic for the NVLM (see section 6.3).

Start the Ladder Logic Editor after first opening the Ladder Editor.



1. Rung Editor is started from within the Ladder Editor.

Figure 6.1 Application Data—applying ladder logic editors

6.1.2 Locked (Template) Rungs

All template rungs (ie rungs created in version 3.0 of the GCS Templates Tool[1] and "exported then saved" to a GCSS installation) are initially **locked**.

When installations containing locked rungs are opened in GCSS 7.0.0 and later, they are identified:

- in the Ladder Editor window by the symbol shown in figure 6.2;
- in the Ladder Editor Report by the letter “L” after the rung index number (figure B.13);
- in the Rung Viewer Report by the word “Locked” after the rung index number (figures 6.12 and B.11).

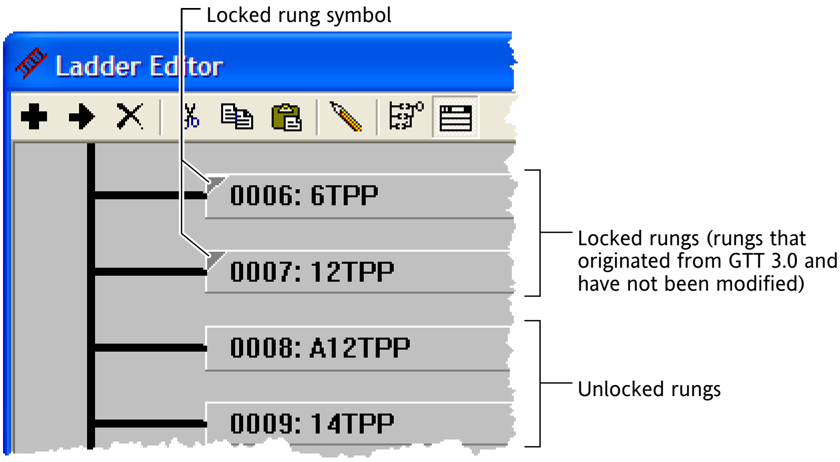


Figure 6.2 Locked and unlocked rungs in the GCSS Ladder Editor

In the Ladder Editor, locked rungs:

- must be unlocked to be edited by either:
  - right-clicking and selecting **Unlock Rung** from the context menu;
  - choosing **Unlock Rung** from the **Rung** menu.

  You can also use this method to unlock a group of adjacent locked rungs. Select the rungs first as described in “Selecting Contiguous Rungs” on page 6-4.
- may be cut, copied and pasted individually, or as a contiguous group (see section 6.2.3.1.2).
- are automatically unlocked when pasted.

Unlocked rungs cannot be re-locked.

[1] See section 2.1.2.4.

6.2 The Ladder Editor

6.2.1 Opening the Ladder Editor

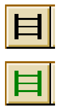
a) Select the Installation window to make it active.

b) Select **Ladder Editor**.

VLM Installation

NVLM Installation

In this example, the NVLM is an NVC/DM



Alt d l
Ctrl+l

6.2.2 Ladder Editor Window

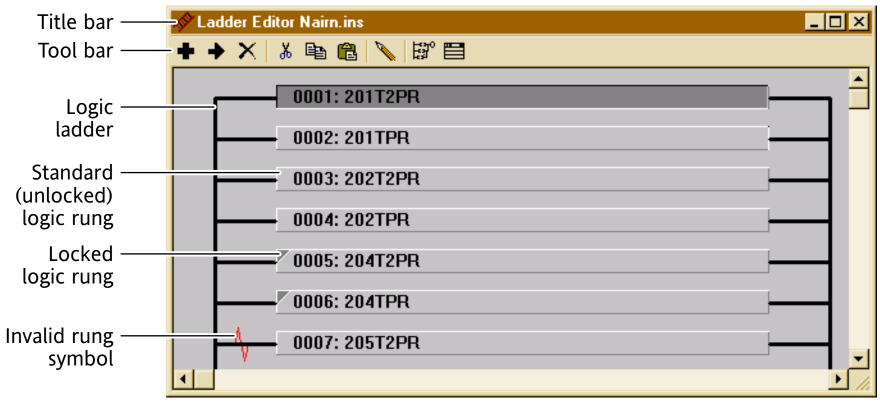


Figure 6.3 Ladder Editor—typical window

The window title bar shows the name of the installation and the body of the window shows the logic ladder for the installation.

Each rung in the ladder comprises:

- the rung's index number (starting at 0001, up to the maximum for the logic module);
- the rung's name (0–15 characters)—that is, the name of the coil represented by the rung;

and may also have:

- a red "break" symbol, which indicates that the rung has not passed a consistency check;
- a triangle in the upper left corner, which indicates a locked rung (section 6.1.2).

The default ladder for new installations consists solely of a special "END rung" which cannot be edited or deleted.

A ladder must never exceed the number of rungs shown in table 2.22 on page 2-25.

Practical systems are often limited to fewer rungs by either logic capacity or available executing time. The reduction in capacity is related to logic complexity and will mostly be more than what is suggested by the typical maximum values also shown in table 2.22.

6.2.2.1 Scrolling the Ladder Editor Window

The content of the Ladder Editor window can be scrolled in exactly the same way as a Windows list box. The scroll bars can be clicked or dragged by the mouse or the normal Windows keyboard commands can be used.

6.2.2.2 Ladder Editor Toolbar

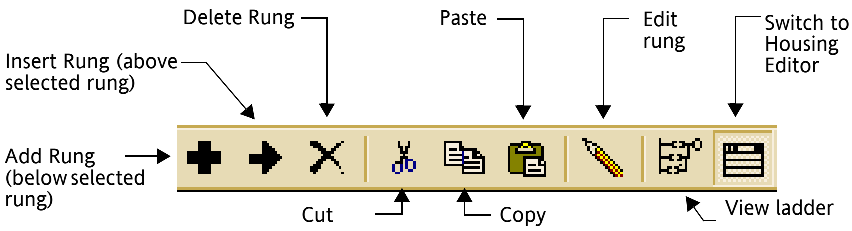


Figure 6.4 Ladder Editor toolbar

This toolbar has buttons for quick activation of the basic operations performed on a ladder. Equivalent menu items are available from the installation menu bar when a ladder editor is active.

6.2.3 Editing the Ladder

This section describes the basic editing operations that can be performed on a selected unlocked[2] rung. All of these operations can be activated by:

- selecting the appropriate menu item;
- selecting the appropriate button from the Ladder Editor toolbar;
- using the assigned accelerator keys;
- right-clicking within the Ladder Editor window to display the context menu.

6.2.3.1 Selecting Rungs

6.2.3.1.1 Selecting a Single Rung

Select by:

- Clicking the rung with the mouse.
- Moving the selector bar (highlight) up and down the editor window using the **Up Arrow** and **Down Arrow** keys.

6.2.3.1.2 Selecting Contiguous Rungs

Select by:

- Clicking an initial rung, then holding down the **Shift** key and clicking the last rung of the group. All rungs between and including the two are selected.
- Clicking an initial rung, then holding down the **Shift** key and using the **Up Arrow** or **Down Arrow** keys to extend the selection.

[2] To edit a locked rung you must unlock it first—see section 6.1.2.

6.2.3.2 Adding Rungs

Rungs can be added by inserting new empty rungs or by pasting copies of existing rungs (see section 6.2.3.4).

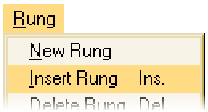
New logic ladders consist solely of one rung, the special "END rung" which cannot be edited or deleted. All other rungs must be added above the END rung.

There are two options for adding a new empty rung to the ladder:

- Below a selected rung by selecting **New Rung**.

Alt r n

- Above a selected rung (or the END rung) by selecting **Insert Rung**.



Alt r i
Insert

6.2.3.3 Deleting Rungs

Select a rung or rungs (see section 6.2.3.1) in the ladder then select **Delete Rung**.

Alt r d
Delete

A confirmation alert box is displayed which must be acknowledged (or cancelled) before the rung is deleted.

Note: ***The END rung or a rung that is currently being edited cannot be deleted.***

6.2.3.4 Cut, Copy and Paste of Rungs

Whole rungs or contiguous blocks of rungs can be cut or copied from the Ladder Editor to the clipboard, then pasted from the clipboard into the Ladder Editor of the same installation, or of a different installation.

This applies to locked rungs (section 6.1.2) as well as unlocked rungs, but note that locked rungs are automatically unlocked when pasted.

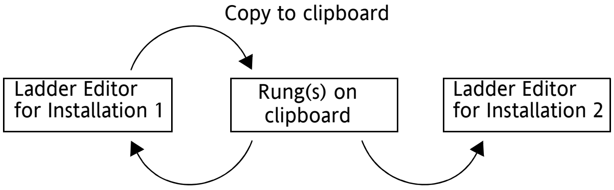
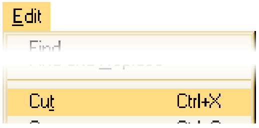


Figure 6.5 Copy, Cut and Paste using the clipboard

The clipboard stores the rung information as ASCII text.

6.2.3.4.1 Cutting

Select the rung(s)[3] and then select **Cut**.



Alt e t
Ctrl+x

The selected rung(s) is removed from the Ladder Editor and stored in the clipboard from where it can be pasted later.

Caution: ***The order of rungs in the Ladder Editor may be critical for correct operation. Therefore take great care when cutting rungs.***

Note: ***A rung cannot be cut to the clipboard if it is currently being edited.***

6.2.3.4.2 Copying

Select the rung(s)[3] and then select **Copy**.

Alt e c
Ctrl+c
Ctrl+Insert

The selected rung(s) is copied to the clipboard from where it can be pasted later.

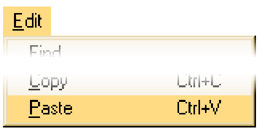
Note: ***A rung cannot be copied to the clipboard if it is currently being edited.***

6.2.3.4.3 Pasting

a) Activate the Ladder Editor of the target installation.

b) Select the rung in the ladder which is to follow the pasted rung(s).

c) Select **Paste**.



Alt e p
Ctrl+v
Shift+Insert

The rung(s) is inserted into the logic ladder immediately above the currently-selected rung. The same rung(s) can be pasted more than once by repeating the paste operation.

[3] See section 6.2.3.1.

Caution: ***The order of rungs in the Ladder Editor may be critical for correct operation. Therefore take great care when pasting rungs.***

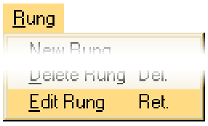
Note: ***Locked rungs are automatically unlocked when pasted.***

Pasting rungs employs the same placement rules as adding rungs. For example, rungs cannot be pasted into a ladder which already has the maximum number of rungs.

6.2.3.5 Selecting a Rung for Editing

You can only edit rungs that are unlocked. See section 6.1.2.

In the Ladder Editor window, select the rung and then select **Edit Rung** (not available for locked rungs).



Alt r e
Return

Alternatively, do one of the following:

- Select the rung and then press the **Return** or **Enter** key (unlocked rungs only).
- Double-click the required rung. (For locked rungs, this opens the Locked Rung Viewer instead of the Rung Editor—see section 6.5.)
- Select the rung, right-click the Ladder Editor window, then select **Edit Rung** from the context menu (unlocked rungs only).

Only one rung of a particular ladder can be edited at a time. If a rung in the ladder is already being edited, trying to edit another rung has no effect. The Rung Editor for the installation must be closed (by selecting the **OK** or **Cancel** buttons from its toolbar) in order to edit another rung.

The rung currently being edited is shown with red text in the Ladder Editor. See section 6.3 for rung editing.

6.2.3.6 Finding and Replacing Mnemonics within Rungs

The GCSS has the ability to find and replace mnemonics within highlighted rungs. See Chapter 9 for details.

6.2.4 Ladder Editor Report

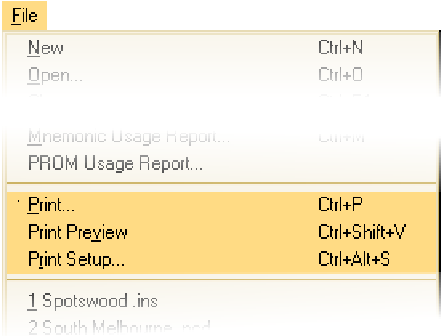
This printed report replicates the layout of the Ladder Editor window. The ladder is continued on subsequent pages when it is too big to fit on a single page. The page orientation can be changed if needed.

With the Ladder Editor active:

a) Select **Print Setup** and change the page orientation or select another printer if necessary.

See Appendix B, section B.2 for further assistance.

b) Select **Print** or **Print Preview**.



Alt f r
Ctrl+Alt+s

Alt f p
Ctrl+p

Alt f v
Ctrl+Shift+v

The report can be sent to the printer after previewing in on screen. See figure B.13 on page B-17 for a typical Ladder Editor report.

6.2.5 Closing the Ladder Editor Window

Closing the Ladder Editor for an installation will also close the Rung Editor and Rung Viewer windows if either (or both) are open.

Click on the **Close** button at the top right corner of the Ladder Editor window.

6.3 The Rung Editor

Use the Rung Editor to add or modify signalling logic by drawing or pasting relay equivalent circuits and defining each function with a mnemonic.

Invoke the Rung Editor from either of:

- the Ladder Editor (see section 6.2.3.5);
- the Rung Viewer (see section 6.4.3).

Only unlocked rungs open the editor (locked rungs open the Locked Rung Viewer—see section 6.5). Unlock rungs as described in section 6.1.2 to edit them.

6.3.1 Rung Editor Window

The Rung Editor window displays the components of the rung fitted to a grid (or array of cells).

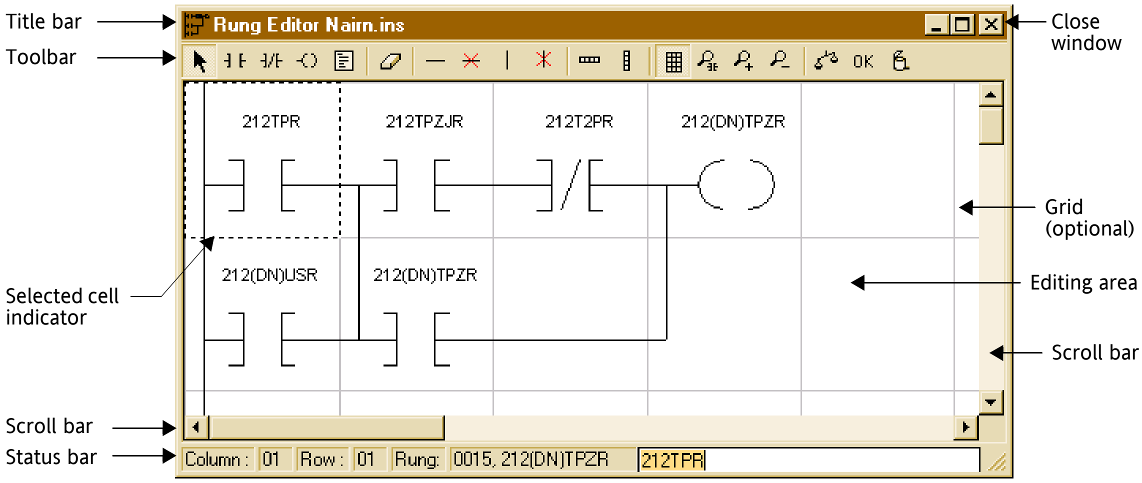


Figure 6.6 Rung Editor window

Title bar—displays the name of the installation.

Toolbar—for fast access to commands (see figure 6.8).

Editing area—with an optionally visible placement grid.

Selected cell indicator—outlines the cell subject to editing.

Scroll bars—moves the editing area within the window to allow access to all 16 rows by 32 columns of possible cell positions.

Status bar—provides details of the selected cell and rung (see figure 6.7).

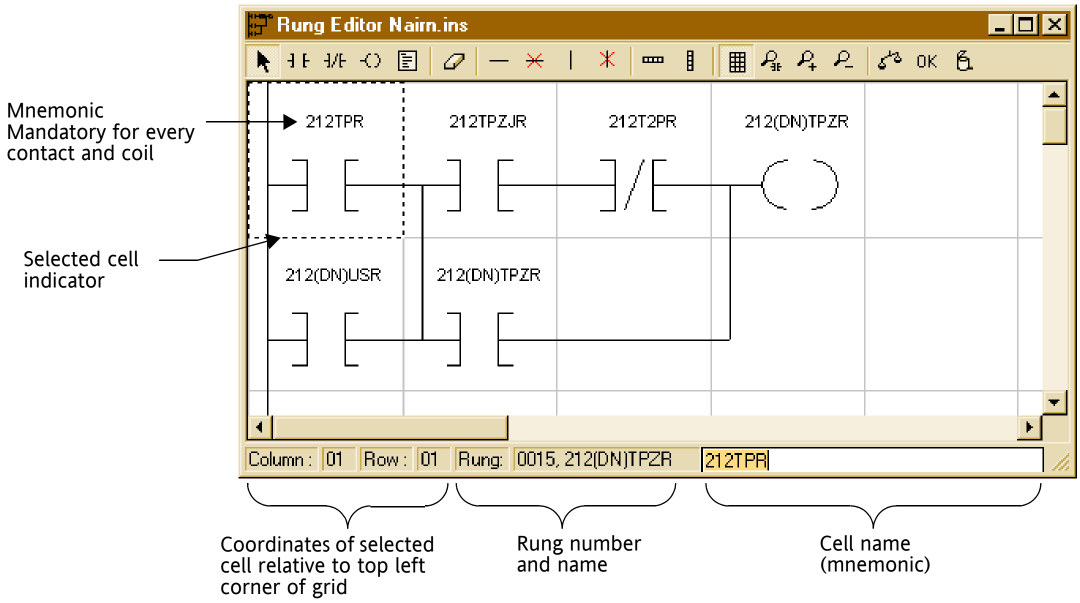


Figure 6.7 Rung Editor window—navigation information

6.3.2 Rung Editor Toolbar and Menus

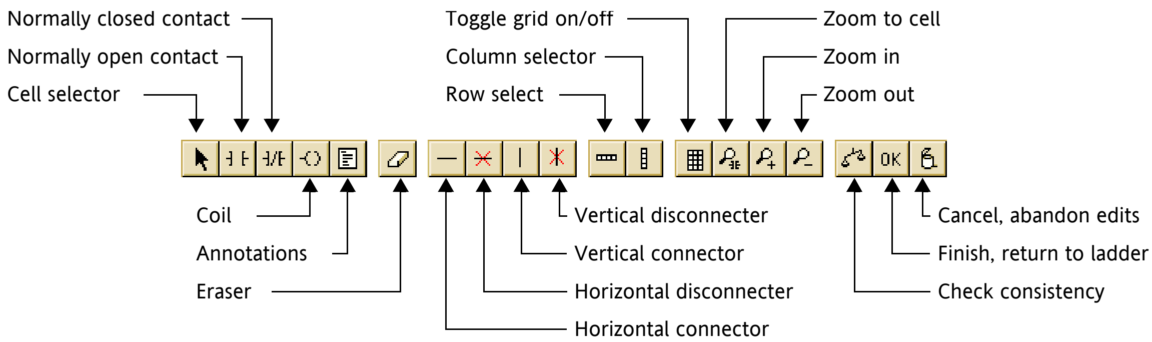


Figure 6.8 Rung Editor toolbar

Use the toolbar buttons for quick activation of the basic operations performed on a rung. The Tools, Check and Grid menus (figure 6.9) have corresponding menu items. Some also have accelerator keys.

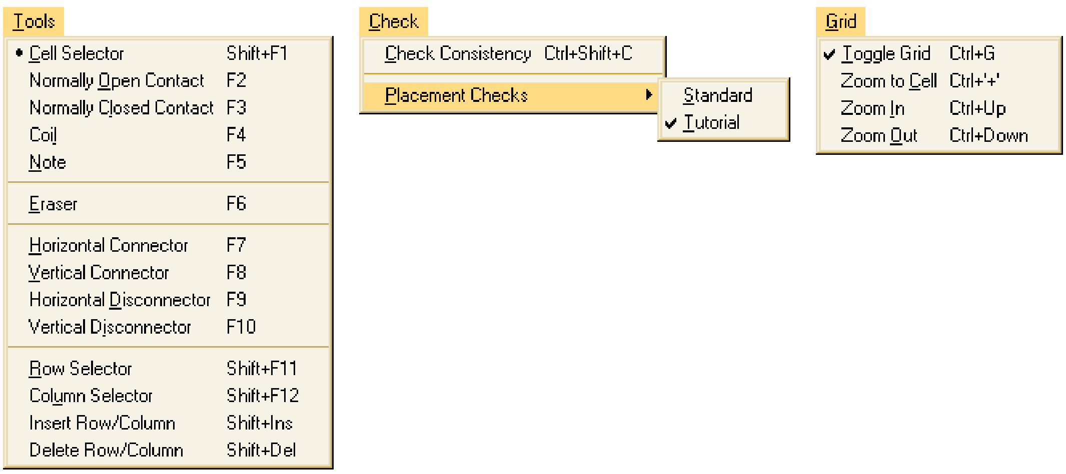
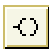
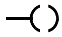


Figure 6.9 Rung Editor menus

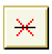
For many of these tools described in sections 6.3.2.1 to 6.3.2.5, the appearance of the mouse pointer changes to reflect the selected tool.

6.3.2.1 Cell Tools

Toolbar Button	Menu Command & Function	Inserts into Cell:
	**Cell Selector**—select an individual cell for working one cell at a time (rather than a row or column of cells). A dashed perimeter outline indicates the selected cell.	
 [1]	**Normally Open Contact**—use to place a normally open contact in a cell.	
 [1]	**Normally Closed Contact**—use to place a normally closed contact in a cell.	
	**Coil**—use to place a coil in a cell. The coil must be placed on the first row of a rung and in the right-most occupied cell of the rung logic.	
	**Annotations**—use to place a note in a cell.	
	**Eraser**—use to remove the entire content a selected cell. Removes the symbol and the connectivity.	

1 The symbol on the toolbar button depends on the notation style selected on the View menu.

6.3.2.2 Connectivity Tools

Toolbar Button	Menu Command & Function	Inserts into Cell:
	**Horizontal Connector**—use to place a horizontal connector in a cell. The contents of the selected cell are connected to the horizontally adjacent cell (left or right).	——
	**Vertical Connector**—use to place a vertical connector in a cell. The contents of the selected cell are connected to the row above.	|
	**Horizontal Disconnecter**—use to remove a horizontal connection.	
	**Vertical Disconnecter**—use to remove a vertical connection.	

6.3.2.3 Row and Column Selectors

Toolbar Button **Menu Option & Function**

Row Selector—use to select the entire row that includes the selected cell.

The following can be done once a row is selected:

- Insert a new row above the selected row by selecting **Insert Row/Column** from the Tools menu (see figure 6.9);
- Delete the selected row by selecting **Delete Row/Column** from the Tools menu (see figure 6.9).

Column Selector—use to select the entire column that includes the selected cell.

The following can be done once a column is selected:

- Insert a new column before the selected column by selecting **Insert Row/Column** from the Tools menu;
- Delete the column by selecting **Delete Row/Column** from the Tools menu.

Insert Row/Column—use to place a new row above the row selected with the **Row Selector**, or a new column to the left of the column selected with the **Column Selector**.

Delete Row/Column—use to remove the row selected with the **Row Selector**, or the column selected with the **Column Selector**.

6.3.2.4 Grid

Toolbar Button **Menu Command & Function**

Toggle Grid—toggles the grid lines on and off. Cells are always placed on the grid whether it is visible or not.

Zoom to Cell—enlarges the selected cell to the maximum available size.

Zoom In—increases the grid size so that fewer cells can be viewed, perhaps with more detail.

Zoom Out—reduces the grid size so that more cells can be viewed, perhaps with less detail. At smaller grid sizes, an entire rung may be viewable without scrolling.

6.3.2.5 Completion of a Rung

Toolbar Button **Menu Command & Function**

Check Consistency—checks the contents of the rung and reports on the rungs validity and any violations of the rung consistency check rules (see section 6.3.8).

OK—finishes rung editing and returns the results of the edited rung to the ladder. The Rung Editor window is closed. Tool bar button only.

Cancel—causes the edit to be abandoned. The results are not stored and the Rung Editor window is closed. Tool bar button only.

Close window—button at top right hand corner of window has the same behaviour as Cancel. Tool bar button only.

6.3.3 Rung Content

The content of a rung comprises:

- a default power supply rail;
- 1 to 16 rows each containing 32 cells.

Valid cell content and connectivity must comply with a number of rules. Compliance with these rules is checked before the edited content of a rung is written back to the ladder (see section 6.3.8).

Every contact and coil must be assigned a mnemonic. Mnemonics are entered by selecting the cell and typing a name into the cell name field. Remember that the order of contacts in a rung can be important. See section 2.5.5.

Mnemonics behave in the same way as a conventional relay circuit in that the state of the contact (open or closed) follows the state of the coil (dropped or picked).

Every mnemonic used in the ladder must be defined in one of the module definitions as an:

- input;
- output;
- latch;
- timer input or output, or;
- reserved mnemonic (with appropriate permissions).

The GCSS will generate errors for undefined mnemonics.

Finding and Replacing Mnemonics within a Rung

The GCSS has the ability to find and replace mnemonics in the rung being edited. See Chapter 9 for details.

6.3.4 Status Bar

The status bar displays the following information.

- **Cell position**—the position of the selected cell relative to the top left corner of the grid.

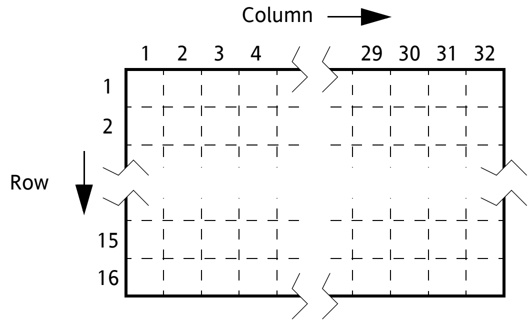


Figure 6.10 Cell positions in a rung

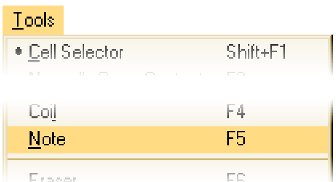
- **Rung number**—a sequential number in a range from 0001 to the maximum possible for the logic module.
- **Rung name**—same as the coil name (up to 15 characters). Changing the name of the cell containing the coil changes the rung name.
- **Cell name**—up to 15 characters. The name of a selected cell appears in the cell name field where it can be altered by typing at the keyboard.

6.3.5 Rung Annotation

The Rung Editor's **Note** command lets you:

- attach optional information to each rung as an aid to logic design and troubleshooting—unlocked rungs only;
- see notes originally attached to the rung in the GCS Templates Tool — template-originated rungs only.

Select **Note** to display the Rung Annotations dialog box.



Alt t n
F5

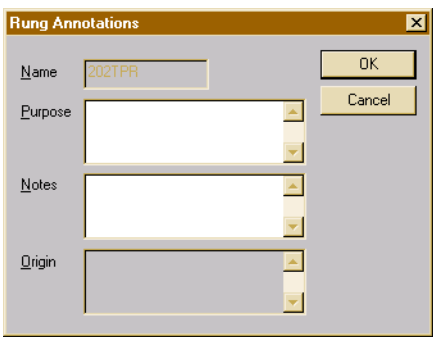


Figure 6.11 Rung Annotations dialog box

Rung annotation has three fields:

- **Purpose**—an overview of what the Rung is intended to achieve.
- **Notes**—additional information the designer believes relevant.
- **Origin**—shows notes from the template used to create the rung (not editable from the GCSS; the field is blank if the rung was not prepared by the template tool).

See also “Showing or Hiding Annotations” on page 6-19.

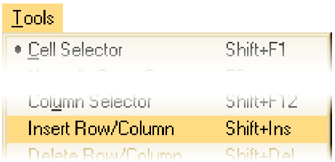
6.3.6 Cell Insert

This is effectively the same as inserting a column.

To insert a cell:

a) Select a cell with the Cell Selector tool.

b) Select **Insert Row/Column** from the Tools menu to insert a new column before the selected cell.

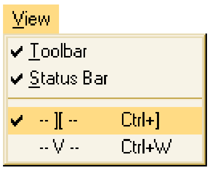


Shift Ins

6.3.7 Alternative Relay Contact Notation

The Rung Editor offers two notations for displaying relay contacts.

Choose between **--][--** and **--V--** from the View menu.



Ctrl+]
Ctrl+w

6.3.8 Rung Consistency Checks

Rung consistency checks are performed whenever:

- **Check Consistency** is selected from the Check menu.
- The **Check** button is selected from the Rung Editor toolbar.
- The **OK** button is selected from the Rung Editor toolbar.

The following checks are performed:

- The rung must contain a coil.
- The rung must not contain any open circuits.
- The rung must not contain any short circuits.
- Every contact in the rung must be named.

An appropriate message is displayed if the rung fails on any of these checks.

Whenever the **OK** button is selected and the rung fails the consistency check, a warning message is displayed but the rung can be saved and the Rung Editor exited.

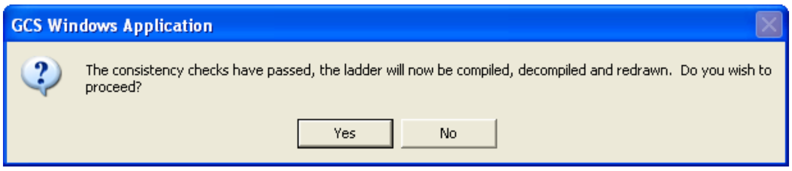
A saved rung that failed the consistency checks is shown with a break symbol (to the left) in the Ladder Editor window.

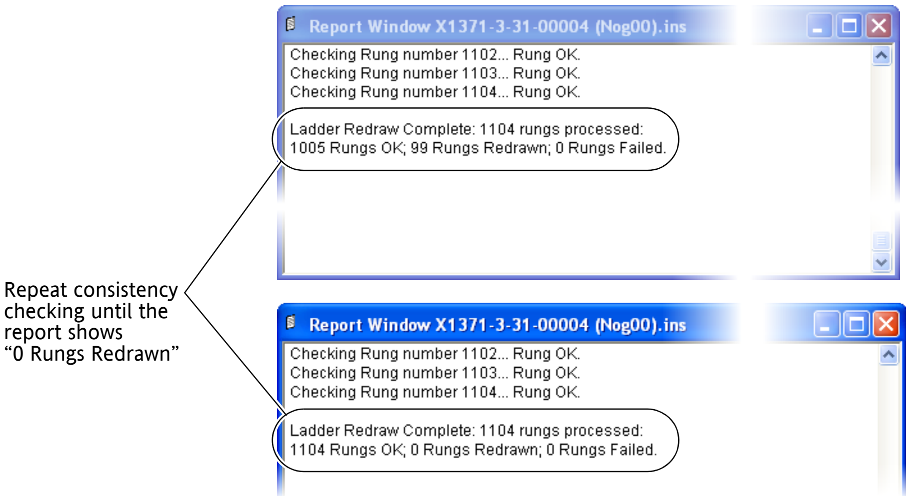
6.3.9 Rung Redraw and Consistency Check

All ladder rungs are also checked immediately after the installation Consistency Checks.

Every rung is checked and redrawn (if necessary) to its actual representation in the WESTRACE equipment. Therefore every rung should be manually checked against the signalling requirements to ensure the equipment contains the intended logic.

Consistency checking must be performed until no rungs are redrawn. Use the report window to see if any rungs were redrawn during the consistency check.





6.3.10 Rung Consistency Check Error Reporting

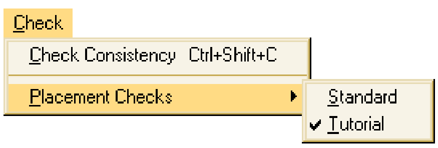
Rungs that fail consistency checks have an error reported and the offending is cell highlighted.

a) Dismiss the error reporting dialog box.

b) Select **Zoom to cell** and the view will automatically scroll and display the offending cell.

6.3.11 Element Placement Checks

Two levels are available when adding contacts a rung.



Standard—allows contacts to be added without the extra checking of the tutorial mode (for advanced users).

Tutorial—extra checks are performed every time a contact is placed in the rung. This helps you avoid creating an invalid rung.

The Rung Editor defaults to the last selected mode.

Note: ***These options do not affect the Rung Consistency Checks.***

Basic Checking Performed by Both Modes

Only allows a coil to be added to the last column.

Extra Checking Performed by the Tutorial Mode

- There must be nothing to the right or below a coil.
- All contacts must be to the left of the coil (if there is one).
- The cell to the left of a contact or horizontal connection, must contain a Right connection.
- There must be at least one contact between the coil and the power rail.

6.3.12 Closing the Rung Editor

The Rung Editor window can be closed by selecting the:

- **OK button**—accepts any rung changes, performs a rung consistency check, redraws the rung, and closes the Rung Editor.

- **Cancel button**—abandons any changes made to the rung and closes the Rung Editor. Confirmation is required before abandoning any changes.

- **Window Close button**—has the same effect as selecting the Cancel button.

6.3.13 Saving the Contents of the Rung Editor

The Rung Editor works on a copy of the rung data and only writes over the original rung data when the **OK** button icon is selected.

The Rung Editor must therefore be closed using the **OK** button before saving or closing the installation in order to save the current rung edits.

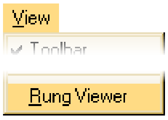
An appropriate warning message is displayed if the Rung Editor is open when the installation is saved or closed.

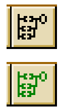
6.4 The Rung Viewer

The Rung Viewer is a single window that provides a detailed view of all rungs in a ladder (figure 6.12). Its purpose is to display the underlying logic of all ladder rungs.

6.4.1 Opening the Rung Viewer

Select **Rung Viewer** with the Ladder Editor active.





Alt v r

Alternatively, do one of the following:

- Select the rung in the Ladder Editor window.
- Right-click on the Ladder Editor window and select **Rung Viewer** from the context menu.

6.4.2 Rung Viewer Window

The Rung Viewer window displays the ladder as a series of expanded rungs showing the underlying logic. The name and component cells of each rung are displayed.

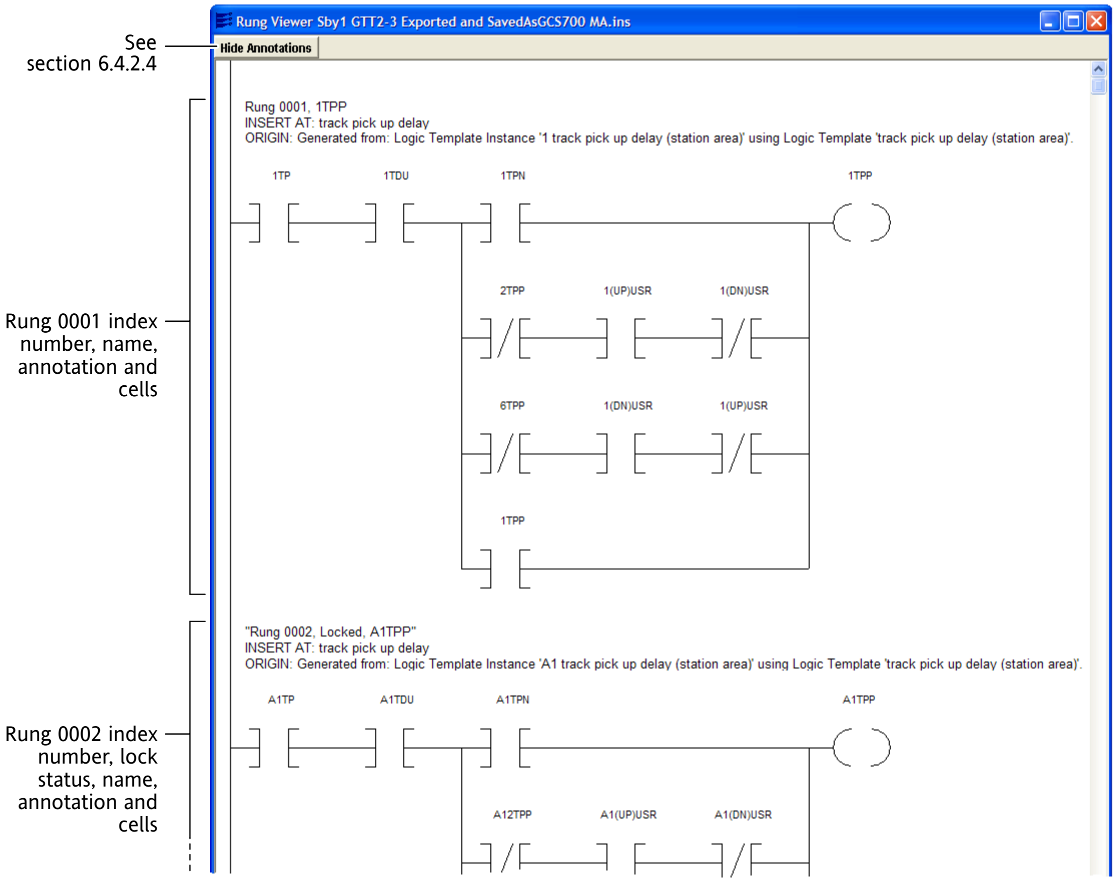
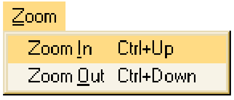


Figure 6.12 Rung Viewer window

6.4.2.1 Zooming the View

To increase the size of the rungs:

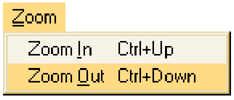
Select **Zoom In**.



Alt z i
Ctrl+Up

To decrease the size of the rungs:

Select **Zoom Out**.



Alt z o
Ctrl+Down

6.4.2.2 Scrolling the Rung Viewer Window

The content of the Ladder Editor window can be scrolled in exactly the same way as a Windows list box. The scroll bars can be clicked or dragged by the mouse or the normal Windows keyboard commands can be used.

6.4.2.3 Change of Notation

See section 6.3.7.

6.4.2.4 Showing or Hiding Annotations

Annotations (notes) can be added in the Rung Editor (section 6.3.5) and the GCS Templates Tool (section 6.1.2). These annotations can be shown or hidden in the Rung Viewer and the Rung Viewer Report (figures 6.13 and B.11).

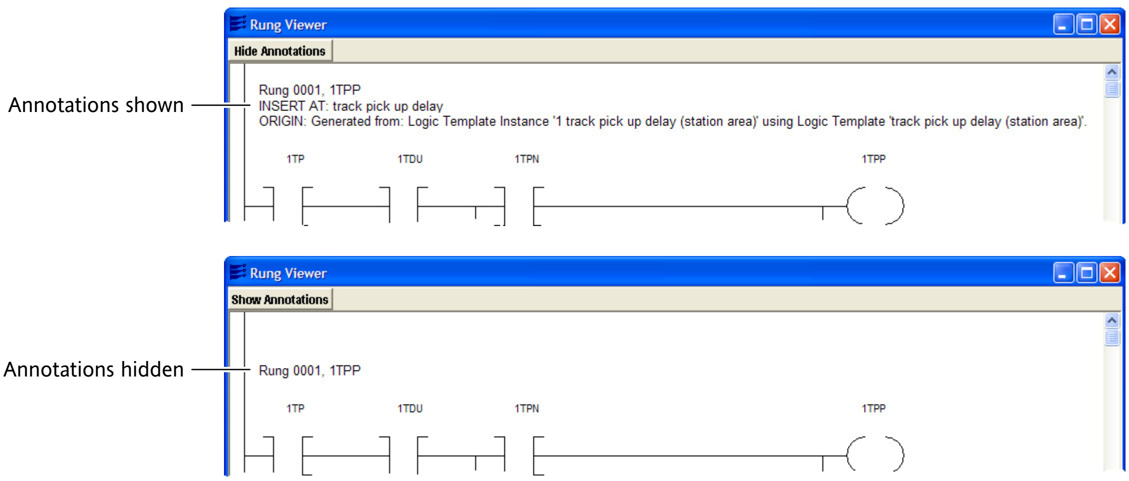
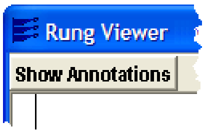


Figure 6.13 Shown and hidden annotations

To show annotations:

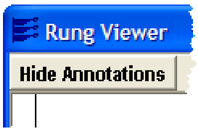
Click the **Show Annotations** button (or right-click in the Rung Viewer and select **Show Annotations** from the context menu).



Ctrl+D

To hide annotations:

Click the **Hide Annotations** button (or right-click in the Rung Viewer and select **Hide Annotations** from the context menu).



Ctrl+B

6.4.3 Opening the Rung Editor from the Rung Viewer

The Rung Editor can be started from the Rung Viewer—unlocked rungs only (see section 6.5 for locked rungs).

Double-click anywhere within the unlocked rung to be edited.

The Rung Editor window opens and, in the Rung Viewer, the name of the rung selected for editing changes colour to red.

Only one rung from a ladder can be edited at a time, so double-clicking a rung has no effect when another rung from the same ladder is already being edited.

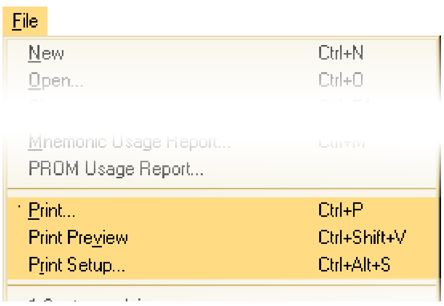
6.4.4 Rung Viewer Report

With the Rung Viewer open:

a) Select **Print Setup** and change the page orientation or select another printer if necessary.

See Appendix B, section B.2 for further assistance.

b) Select **Print** or **Print Preview** to generate a report of the logic ladder rungs.



Alt f r
Ctrl+Alt+s

Alt f p
Ctrl+p

Alt f v
Ctrl+Shift+v

See figure B.11 on page B-15 for an example Rung Viewer Report.

6.4.5 Closing the Rung Viewer Window

Click the **Close** button at the top right corner of the Rung Viewer window.

6.5 The Locked Rung Viewer

The Locked Rung Viewer provides a detailed view of the underlying logic of a single locked rung.

In the Locked Rung Viewer:

- only one rung is displayed (as in the Rung Editor);
- editing is not possible;
- only the Toggle Grid, zoom, Check Consistency and Cancel commands and buttons are functional;
- rung annotations are not displayed.

To see the ladder logic of a *locked* rung, the Locked Rung Viewer (displays the logic of one locked rung) is an alternative to the Rung Viewer (displays the logic of all rungs).

6.5.1 Opening the Locked Rung Viewer

Double-click a locked rung in the Ladder Editor, or double-click a locked rung in the Rung Viewer (figure 6.14).

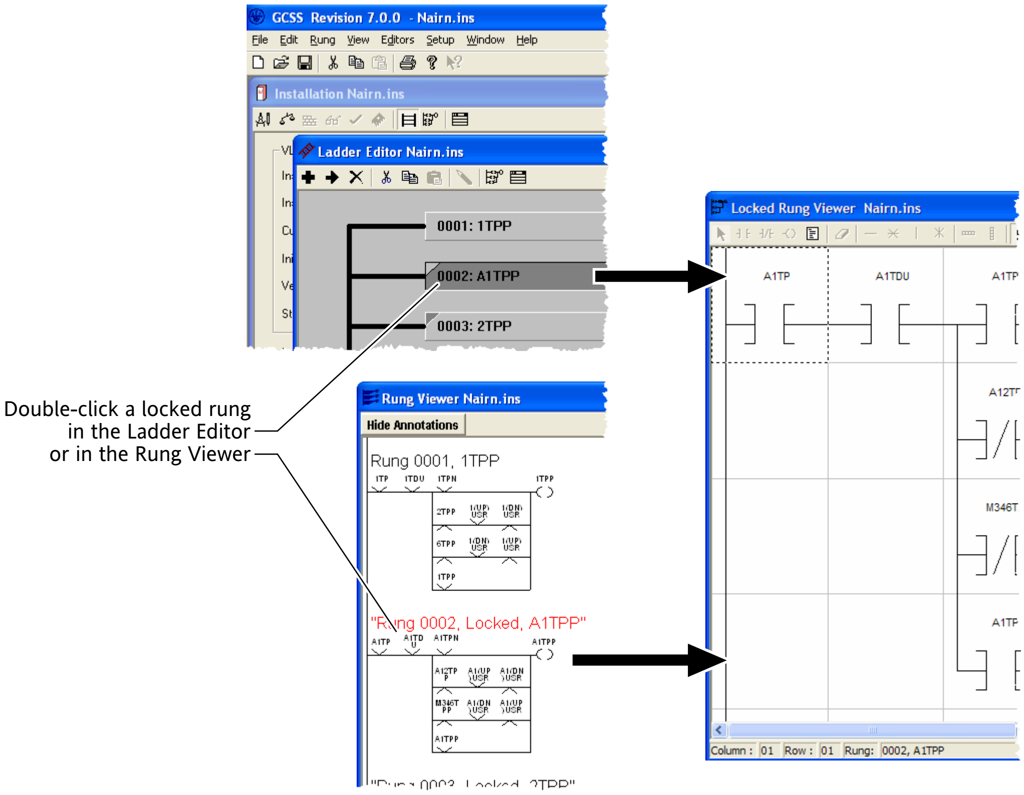


Figure 6.14 Two methods of opening the Locked Rung Viewer

6.5.2 Locked Rung Viewer Window

The Locked Rung Viewer window displays the component cells of the selected locked rung.

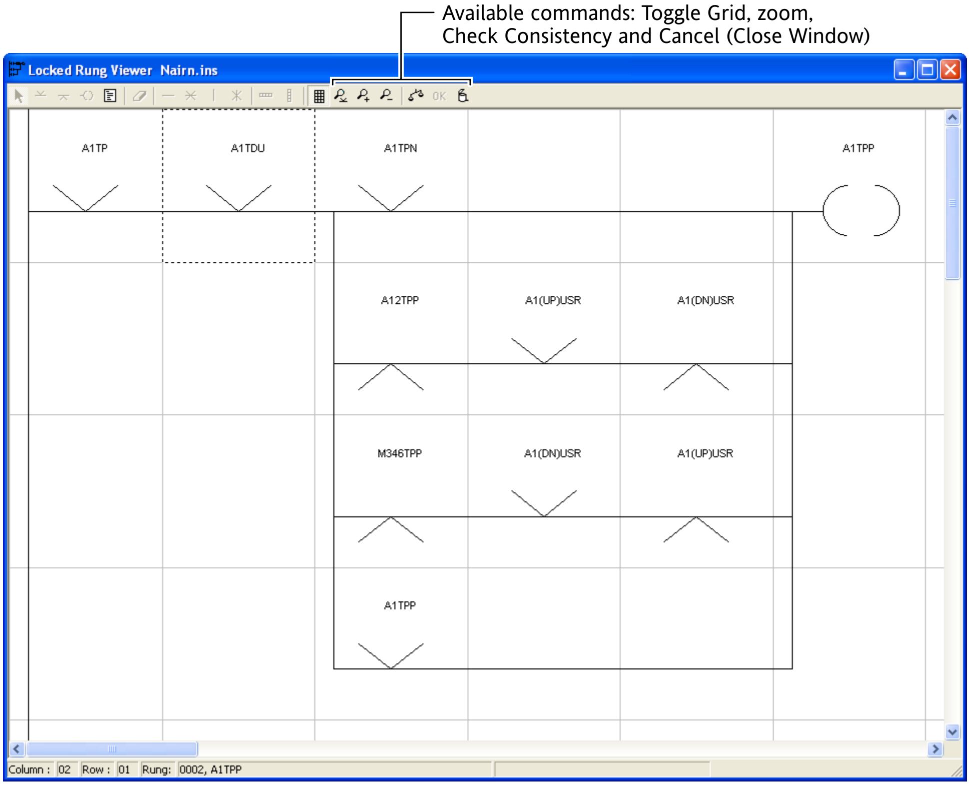


Figure 6.15 Locked Rung Viewer window

7. Consistency, Compiling, Checking and Approval of Installations

Although the primary focus of this section is VLM and NVLM installations, it does mention interface files at appropriate places.

7.1 Overview

Signal Engineers must ensure that all WESTRACE application logic is error-free and functionally correct before it is used on a railway. This applies equally to vital and non-vital installation data and the content of interface files.

Hot-standby systems do not have additional checking or approval processes.

Table 3.4 on page 3-15 shows the categories of Signalling Engineers that have the authority to perform consistency checking, compiling, manual checking and approving of installations.

This chapter describes the steps:

a) **Check consistency**—a manually-initiated automatic check performed by the GCSS to detect syntactic and semantic errors. See section 7.2 and table 3.4.

b) **Compile installation**—see section 7.3 and table 3.4.

c) **Manually check logic**—see section 7.4 and table 3.4.

d) **Test logic principles**—see section 7.4 and table 3.4.

e) **Add Manual Check Record**—see section 7.4 and table 3.4.

   Installations (and interface files) must be signed off as “checked” before they can be approved.

f) **Approve installation**—approval of fully checked logic by an authorized Signal Engineer. See section 7.5 and table 3.4.

Mnemonic usage reports and difference lists can be used in conjunction with reports from these functions to assist in resolving errors and warnings that may be generated.

7.1.1 Implications for Different WESTRACE Designs

There are three basic cases:

- a simple application based on a VLM alone
- an application based on a VLM and NVLM
- multi VLM6-based applications that are required to communicate with each other

7.1.1.1 Stand-Alone VLM Installation-Based WESTRACE application

The VLM may be a HVLM128, HVLM128a, VLM5 or VLM6.

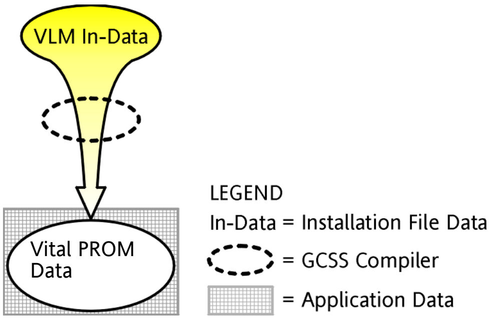


Figure 7.1 Application—VLM Installation only

In this case, the procedure is:

a) Check consistency—section 7.2.
b) Compile installation—section 7.3.
c) Manually check logic and test logic principles—section 7.4.
d) Add manual check record—section 7.4.
e) Approve installation—section 7.5.

7.1.1.2 Application Comprising VLM and NVLM Installations

The VLM can be an HVLM128, HVLM128a, VLM5 or VLM6 and the NVLM an NCDM or NVC/DM.

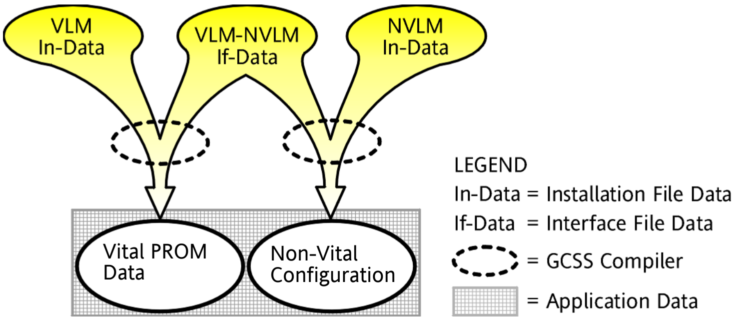


Figure 7.2 Application—VLM and NVLM Installations

Perform consistency checking, manual checking and approval in the following order:

a) The VLM-NVLM Interface file, through either the VLM (preferred) or NVLM installation:
   i. Check consistency—see section 5.2.8.3.
   ii. Manually check input and output mnemonics—see section 5.2.8.3.
   iii. Add Manual Check Record—see section 5.2.8.3.
   iv. Approve interface file by adding Approval Record—see section 5.2.8.3.

Adding a Manual Check record or Approval record to the interface file will force the status of the associated VLM and NVLM installations to change from "Approved" (if approved) to "Modified". See section 7.1.2.

b) The NVLM installation:
   i. Check consistency—section 7.2.
   ii. Compile installation—section 7.3.
   iii. Manually check logic and test logic principles—section 7.4.
   iv. Add manual check record—section 7.4.
   v. Approve installation—section 7.5.

c) The VLM installation. Perform the same procedure as for the NVLM installation.

7.1.1.3 Two VLM6-Based applications Required to Communicate

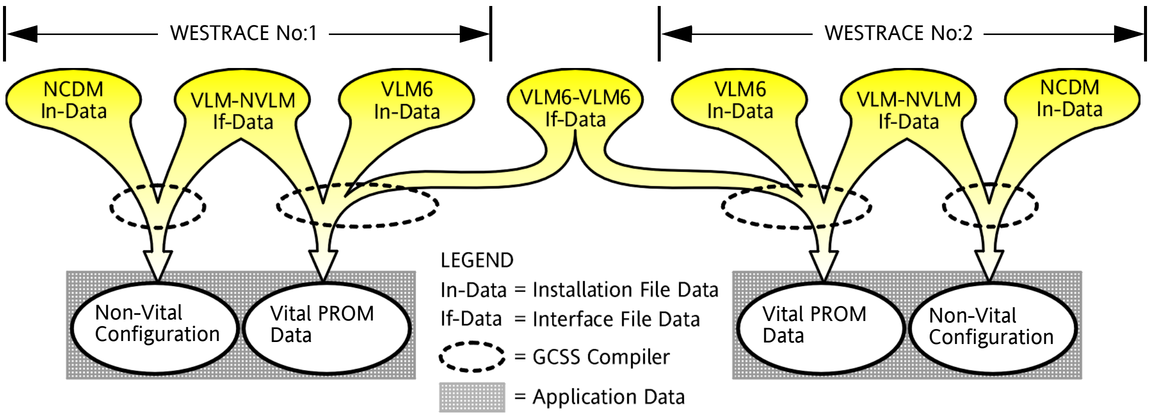


Figure 7.3 Application—two interconnected VLM6-based WESTRACE applications

Perform consistency checking, manual checking and approval in the following order:

a) The VLM-NVLM Interface file for both WESTRACE applications, through either the VLM (preferred) or NVLM installation:
   i. Check consistency—see section 5.2.8.3.
   ii. Manually check input and output mnemonics—see section 5.2.8.3.
   iii. Add Manual Check Record—see section 5.2.8.3.
   iv. Approve interface file by adding Approval Record—see section 5.2.8.3.

   Adding a Manual Check record or Approval record to the interface file will force the status of the associated VLM and NVLM installations to change from "Approved" (if approved) to "Modified". See section 7.1.2.

b) The VLM6-VLM6 Interface file, through the VLM6 module in either of the two WESTRACE applications:
   i. Check consistency—see section 5.2.7.2.
   ii. Manually check input and output mnemonics—see section 5.2.7.2.
   iii. Add Manual Check Record—see section 5.2.7.2.

iv. Approve interface file by adding Approval Record—see section 5.2.7.2.

Adding a Manual Check record or Approval record to the interface file will force the status of the associated VLM6 installations to change from “Approved” (if approved) to “Modified”. See section 7.1.2.

c) The NVLM installations in both WESTRACE applications:

i. Check consistency—section 7.2.

ii. Compile installation—section 7.3.

iii. Manually check logic and test logic principles—section 7.4.

iv. Add manual check record—section 7.4.

v. Approve installation—section 7.5.

d) The VLM installations in both WESTRACE applications. Perform the same procedure as for the NVLM installation.

7.1.2 Impact of Changing Interface File Status Upon Installation Status

Changing the status of any interface file will change the status of any associated installation from “Approved” (if approved) to “Modified”.

The GCSS will display dialog boxes similar to those shown in figure 7.4 next time an affected installation is opened.

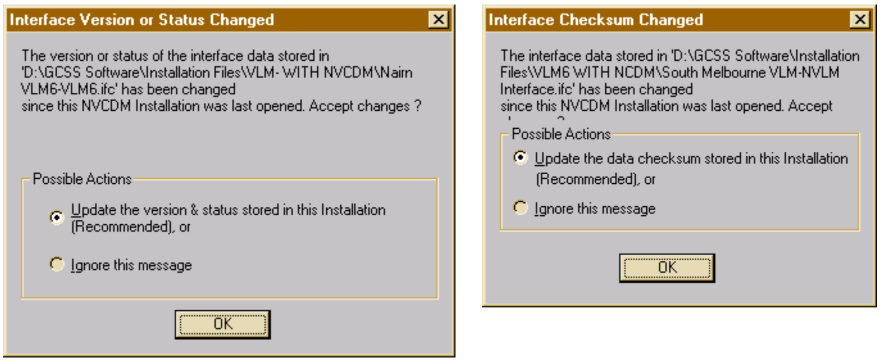


Figure 7.4 Interface Version or Status Changed and Interface Check Sum Changed dialog boxes

Selecting the **OK** button to accept the recommendation will cause the status of the VLM or NVLM installation to revert to “Modified”. Selecting “Ignore this message” before selecting the **OK** button will leave the VLM or NVLM installation status unchanged.

7.2 Consistency Checking of Installations

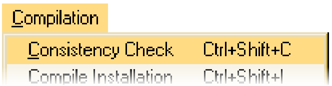
Check consistency of installations as required during development and always before compiling.

GCSS checks and redraws every rung in the logic ladder to its actual representation in the WESTRACE equipment when the installation passes the Consistency Checks. The GCSS displays a warning message before this action is taken to allow cancellation of the operation.

Manually check redrawn ladder rungs against the signalling requirements to ensure the equipment contains the intended logic representation.

7.2.1 Consistency Check Activation

Select the installation window and then select **Consistency Check**.



Alt c c
Ctrl+Shift+c

Note: ***An installation cannot be checked for consistency while the module or rung editor is active.***

7.2.2 Consistency Check Report

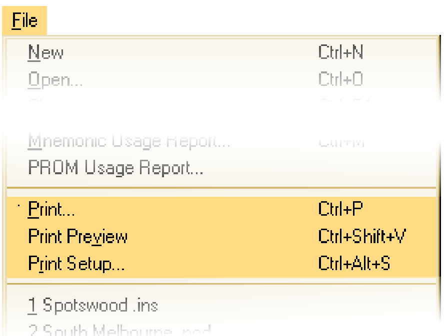
A consistency check activates the Report window where progress of the check and any inconsistencies are displayed. The report also shows the date, time and a statement of whether the installation passed or failed the checks.

With the Report Window open:

a) Select **Print Setup** and change the page orientation or select another printer if necessary.

See Appendix B, section B.2 for further assistance.

b) Select **Print** or **Print Preview** to generate a report of the logic ladder rungs.



Alt f r
Ctrl+Alt+s

Alt f p
Ctrl+p

Alt f v
Ctrl+Shift+v

See section B.5.14 on page B-20 for an example.

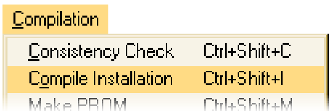
7.3 Compiling Installations

The installation must pass the consistency check before compiling can be performed. Manual compiling must be performed before a manual Check record can be added. VLM and NVLM installations are compiled independently.

Note: ***The GCSS automatically compiles installations after installation approval and immediately before downloading Vital PROM Data to PROMs or downloading the Non-Vital Configuration to the NVLM as described in Chapter 8.***

7.3.1 Compilation Activation

Select the installation window and then select **Compile Installation** from the Compilation menu.



Alt c o
Ctrl+Shift+m

7.3.2 Compilation Report

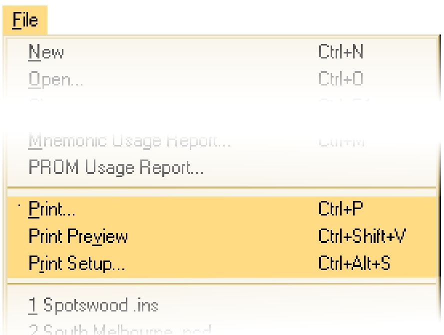
The GCSS Report window displays compilation progress and any errors that may occur. See Appendix C for an explanation of compilation errors.

With the Report Window open:

a) Select **Print Setup** and change the page orientation or select another printer if necessary.

   See Appendix B, section B.2 for further assistance.

b) Select **Print** or **Print Preview** to generate a report of the logic ladder rungs.



Alt f r
Ctrl+Alt+s

Alt f p
Ctrl+p

Alt f v
Ctrl+Shift+v

See section B.5.15 on page B-21 for an example.

7.4 Manual Checking of Installations

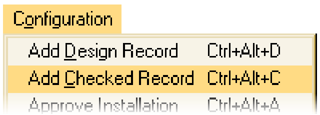
An appropriately authorized Signalling Engineer must manually check the installation design and its logic for compliance with factors such as control tables, standard circuits, and the company methodology.

Manual checking must include:

- housing layout and selection of WESTRACE modules;
- user-defined mnemonics;
- both vital and non-vital ladder logic;
  Ensure that no vital logic that should be in the VLM Installation is included with the non-vital logic of the NVLM Installation.

An authorised Signal Engineer (see table 3.4 on page 3-15) may add a check record after confirming the installation or interface file meets the signalling requirements. A Manual Check record **must** be added before the installation can be approved.

a) Select the installation window and select **Add Checked Record** to open the Add Checked Record dialog box.



Alt o c
Ctrl+Alt+c

Figure 7.5 Add Check Record dialog box

b) Select the **OK** button.

GCSS adds a Checked record to the installation file and the file is saved. A copy of the installation file is added to the Log file and the Log file is saved.

7.5 Approving Installations

Approval of installations must only be performed by an authorised Signal Engineer (see table 3.4 on page 3-15) and only when the engineer is satisfied that the design is ready for service. Approving an installation is analogous to signing the “Approved” field in a drawing or other document.

An installation can only be approved after all associated interface files (if any) have been approved.

A Checked record (see section 7.4) must be added to an installation before an Approval record can be added.

An installation must be approved before it can be used to create Vital PROM Data or Non-Vital Configuration data.

7.5.1 Approval Activation

a) Select the installation window and then select **Approve Installation** to open the Approval Installation dialog box.

Alt o a
Ctrl+Alt+a

Figure 7.6 Approve Installation dialog box

b) Select the **OK** button.

An Approval record is added to the installation file and the file is saved. A copy of the installation file is added to the Log file and the Log file is saved.

7.5.2 Approval Report

GCSS automatically generates the Approval Report in the Report window. An approval report contains a single checksum value which is a checksum of all the information in the installation that is compiled and stored in the WESTRACE equipment.

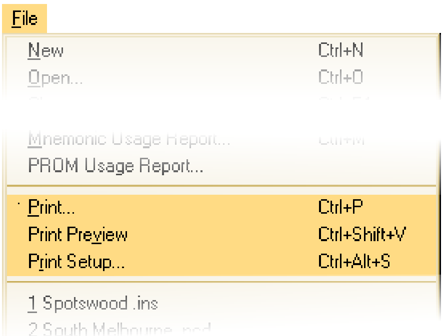
The Approval Report must be printed and signed by the authorised approving Signal Engineer whenever an installation is approved. It provides paper evidence of approval and is used during the Installation check function on commissioning. The Approving Authority must have an adequate system for filing and retaining these reports.

With the Report Window open:

a) Select **Print Setup** and change the page orientation or select another printer if necessary.

   See Appendix B, section B.2 for further assistance.

b) Select **Print** or **Print Preview** to generate a report of the logic ladder rungs.



Alt f r
Ctrl+Alt+s

Alt f p
Ctrl+p

Alt f v
Ctrl+Shift+v

See section B.5.17 on page B-23 for an example.

8. Data Download, Upload and Compare

This chapter describes:

- generating and downloading compiled VLM installation data to the WESTRACE PROMs—section 8.1 (page 8-2);
- generating and downloading compiled NVLM data to the WESTRACE Non-Vital Configuration—section 8.2 (page 8-46);
- uploading from the PROMs or the Non-Vital Configuration to re-create installations, or to compare with the original installation data or other installations—section 8.3 (page 8-48).

These processes are shown in figure 8.1 beginning at step 6. (Figure 8.1 is derived from figure 2.44 and section 1.10.)

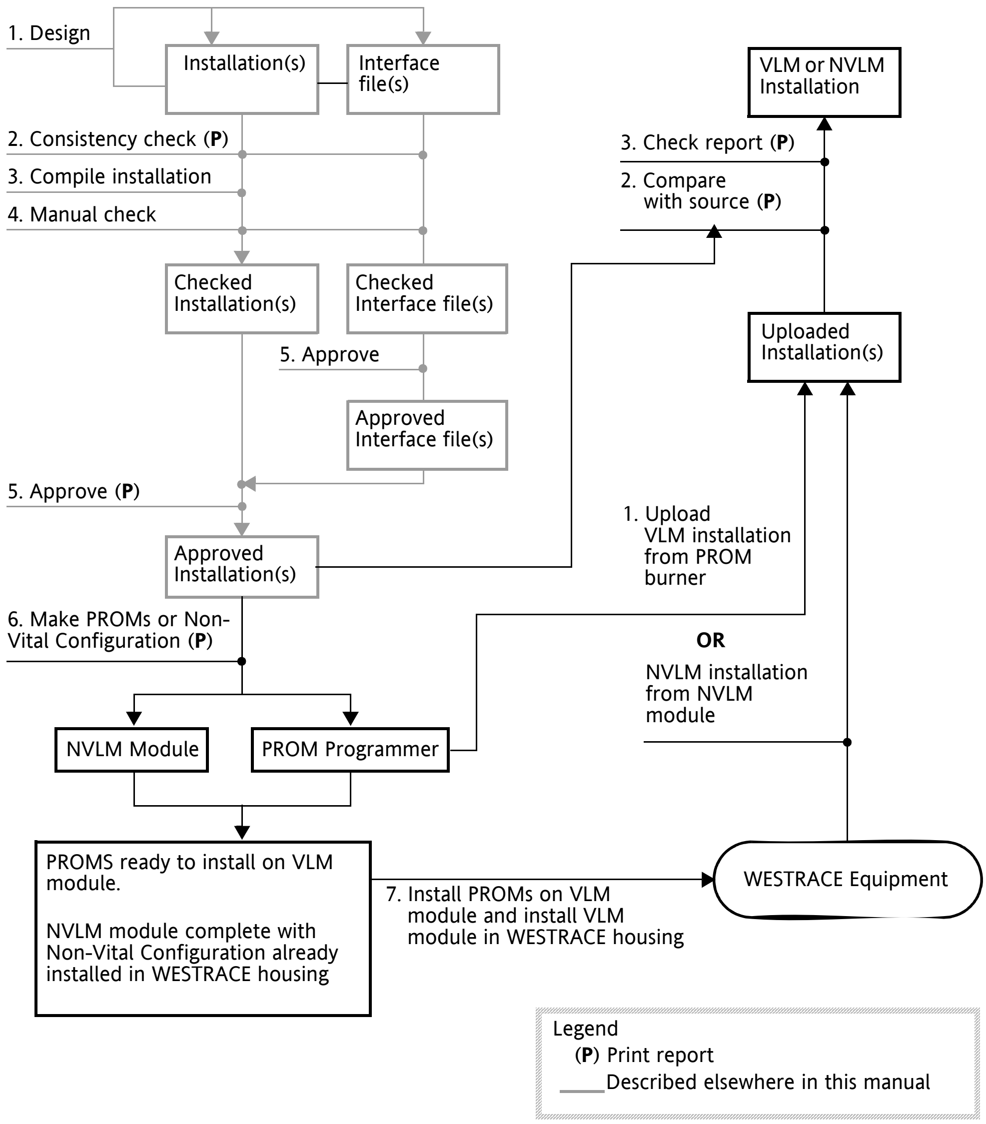


Figure 8.1 Data download, upload and compare

8.1 VLM Installations: Make PROMs

8.1.1 Overview of PROM Programming

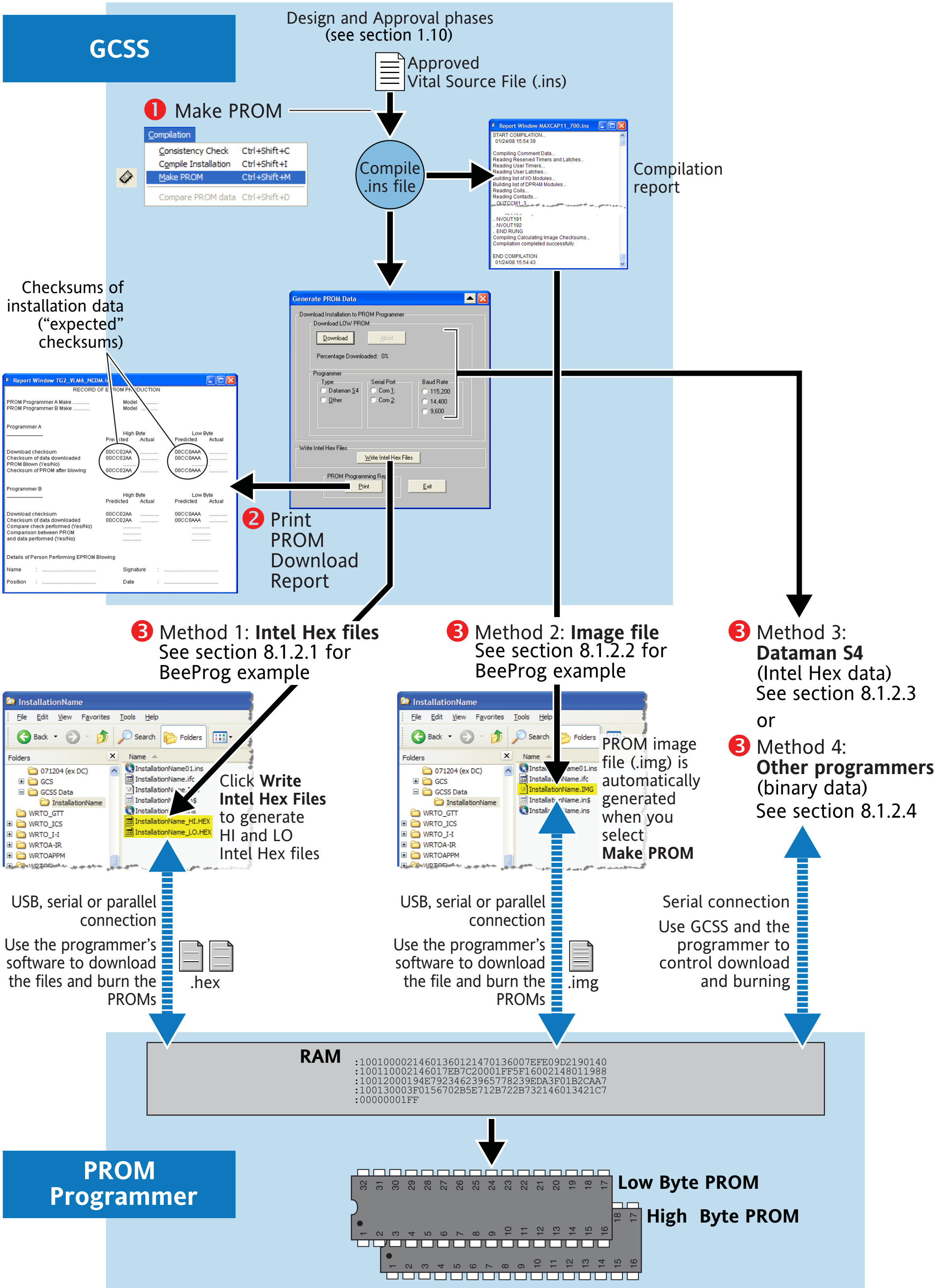


Figure 8.2 PROM programming overview—downloading and burning

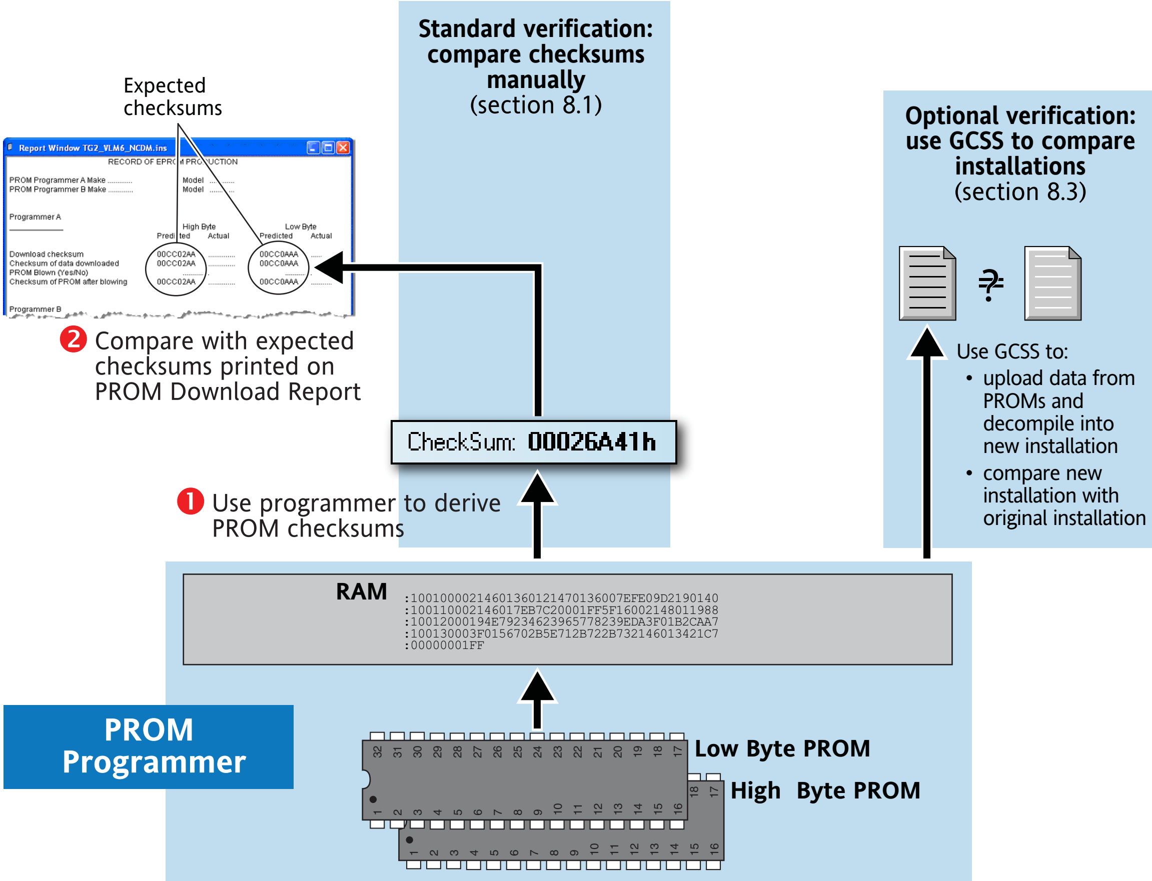


Figure 8.3 PROM programming overview—verifying

Relates to design cycle:
- task E on page 1-9
- task 39 on page 1-13

You do the following to program (ie download and burn) the two PROMs—High Byte and Low Byte—required for each VLM installation (figures 8.2 and 8.3), and then verify them:

a) Use GCSS to:

   i. Print an installation-specific report that you use later to verify and approve the PROMs.

   ii. Compile the installation data and save the resulting image file (.img) to the installation's directory.

b) Download the compiled data to a PROM programmer.

   GCSS makes the data available in various formats and, depending on your programmer, it can be downloaded by means of USB, serial or parallel connections.

c) Use the PROM programmer to burn the data into the two PROMs.

d) Verify the PROM contents.

Lastly, you fit the PROMs to the VLM and install the VLM into the WESTRACE.

Much of this is initiated from GCSS's Generate PROM Data dialog box (figure 8.4).

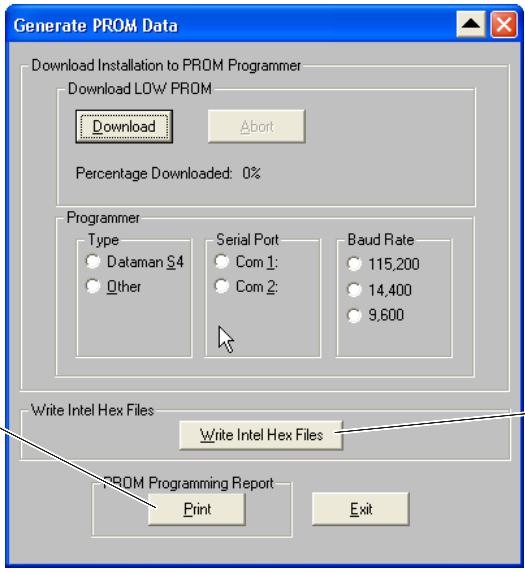


Figure 8.4 Generate PROM Data dialog box—"command central" for PROM data

The data burnt into the PROMs is safety-critical, but the PROM programmer is not. Therefore you must follow the procedures specified in this manual and [ICS] to ensure that the data is correct and uncorrupted. In particular, this requires:

- verification of the installation data checksums when burning PROMs (described below);
- verification of the data in the installed WESTRACE system using ICS (see reference [ICS]).

8.1.1.1 Recommended PROM Programmers

We recommend the ELNEC BeeProg programmer and the Dataman S4 programmer, and this chapter includes the following examples of their use:

- section 8.1.2.1, "Procedure 1: Intel Hex Files—BeeProg USB Example" (preferred for GCSS 7.1.x and later)
- section 8.1.2.2, "Procedure 2: Image File—BeeProg USB Example" (in current use, but replace with above procedure if practicable)
- section 8.1.2.3, "Procedure 3: Dataman S4" (preferred procedure for Dataman programmers)

	BeeProg	**Dataman S4**
Supports **Intel Hex** data file type (see section 8.1.1.2)	✓ Preferred	✓
Supports **image** data file type (see section 8.1.1.2)	✓	✓
USB comms	✓	✗
Serial comms	✗	✓
Parallel comms	✓	✗

Table 8.1 Recommended PROM programmers

For other supported programmers, use the example procedures in this chapter as a guide and refer to the specific programmer's manual if you need more detail.

Note: ***Serial programmers used with GCSS must comply with the parameters listed in section 10.1.3.***

8.1.1.2 Installation Data Formats

GCSS 7.1.x can generate installation data for PROM burning in four formats:

- **Low and High Intel Hex files (_LO.hex and _HI.hex)**
  GCSS 7.1.x can generate PROM data in the form of two Intel® Hex[1] files, one for the Low Byte PROM (eg InstallationName_LO.hex) and the other for the High Byte PROM (eg InstallationName_HI.hex)—see figure 8.2.

  You initiate this using the **Write Intel Hex Files** button in figure 8.4.

  Section 8.1.2.1 gives an example of PROM programming using Intel Hex files (uses the BeeProg programmer).
- **Binary image data file (.img)**
  GCSS automatically compiles the PROM data as a binary image file (eg InstallationName.img) the first time you begin the “Make PROM” procedure. See figure 8.2.

  Section 8.1.2.2 gives an example of PROM programming using an image file (uses the BeeProg programmer).
- **Intel Hex data** (Dataman serial data)
  GCSS has download and programming functionality specifically designed for the Dataman S4 programmer. GCSS can generate Intel Hex data and make it available to a specified COM port on the computer. Then you use GCSS and the Dataman to control the downloading of this serial data to the programmer and its burning into the PROMs.

  See section 8.1.2.3.
- **Binary data** (“Other” serial programmers)
  GCSS can provide the installation data in binary format to a specified COM port on the computer. This can be used with any serial PROM programmer that supports binary-formatted data, 9600 Baud (or better) and RTS handshaking. Verification methods will depend on the programmer.

  See section 8.1.2.4.

8.1.1.3 PROM Download Report

In the procedures below you are instructed to print, complete and retain a PROM Download Report (also known as a PROM Programming Report or a Record of EPROM Production; figure B.6). See figure 8.2 and the **Print** button in figure 8.4.

Note: ***Ignore the greyed-out areas of the report shown in figure B.6 for WESTRACE applications using WRSA-supplied equipment.***

This report is installation-specific—ie it contains the checksums predicted by GCSS for the specific installation’s High Byte and Low Byte

[1] Intel Hex format: http://en.wikipedia.org/wiki/Intel_hex

data to be sent to the PROM programmer. Later, for validation, you compare these expected checksums with the checksums derived from the actual PROMs after programming, and record this information on the report.

The actual checksums must be identical to the predicted checksums. You must discard, or erase and re-program, any PROM for which the actual checksum is not identical to the expected checksum.

SAFETY WARNING

A PROM Download Report must be printed, completed and retained for every pair of PROMs (High Byte and Low Byte) used in a WESTRACE application.

8.1.1.4 Errors

Errors can occur when:

- installation data is compiled into the .ins installation file—compilation errors;
- the compiled data is downloaded to the PROM programmer—communications errors;
- the PROM programmer burns the PROMs—burn errors.

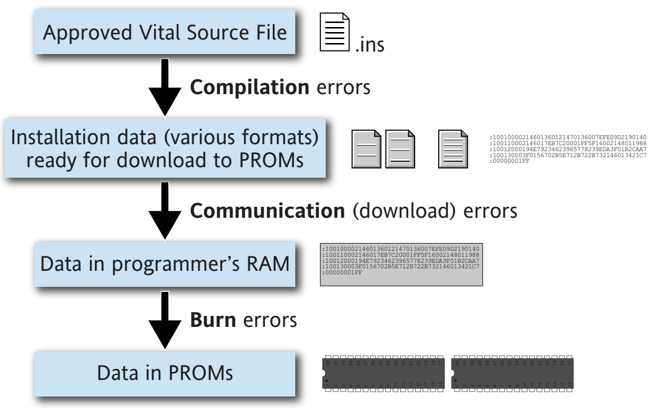


Figure 8.5 Errors can occur during compile, download or burn

To expose any errors that occur, validation steps are included in the GCSS procedures below, and in the subsequent ICS procedures (see [ICS]).

8.1.2 Making PROMs

Contents of this section:

8.1.2.1 Procedure 1: Intel Hex Files—BeeProg USB Example

This section gives an example of programming VLM PROMs from GCSS-generated Intel Hex files (filenames end with _HI.hex and _LO.hex).

A BeeProg programmer, connected to the computer by USB, is used in this example.

Note: ***This section is only applicable to versions of GCSS that can generate Intel Hex files, ie GCSS version 7.1.x and later.***

Figure 8.6 ELNEC BeeProg PROM programmers

Note: ***Although these instructions refer to the basic BeeProg, they can be used, with minor adaptation, for BeeProg+ and SmartProg2 programmers.***

8.1.2.1.1 Preliminaries

a) Ensure that the installation's Vital Source File has been approved (see section 1.10).

b) Install the BeeProg software on the computer as described in section 8.1.2.5.1.

c) Use a USB cable to connect the BeeProg to the computer, then power the BeeProg.
See section 8.1.2.5.2, "Setting Up the BeeProg" for full details, especially if this is the first time the BeeProg has been connected to this computer.

d) Start the BeeProg Control Program PG4UW (figure 8.7) and establish communication with the BeeProg as described in section 8.1.2.5.3.

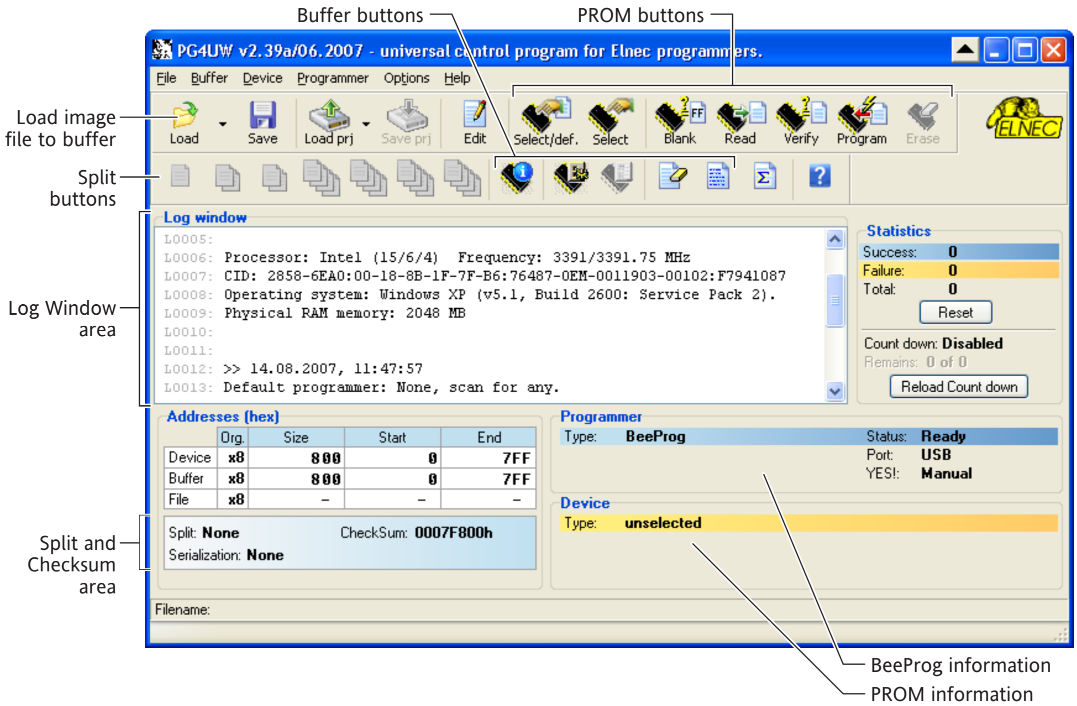


Figure 8.7 BeeProg PG4UW main screen—initial view

e) Erase the PROMs (new and used) as described in section 8.1.2.5.4.

f) Label one PROM “Low” and the other “High”.

Use a sticker, or write directly on the PROM. We recommend that you also label each PROM with the installation name and the date.

g) Select the type (manufacturer and part number) of the PROMs you are using (eg **AMD Am29F010B** or AMD Am28F010 for WESTRACE):

TIP: The PG4UW software remembers the last-used device type. Therefore, if the required PROM type is already showing in the Device Type area (figure 8.14), you can skip to step **h** below.

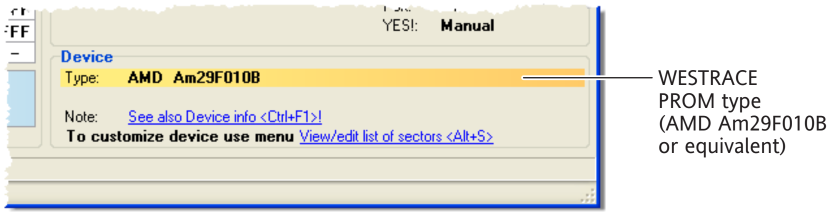
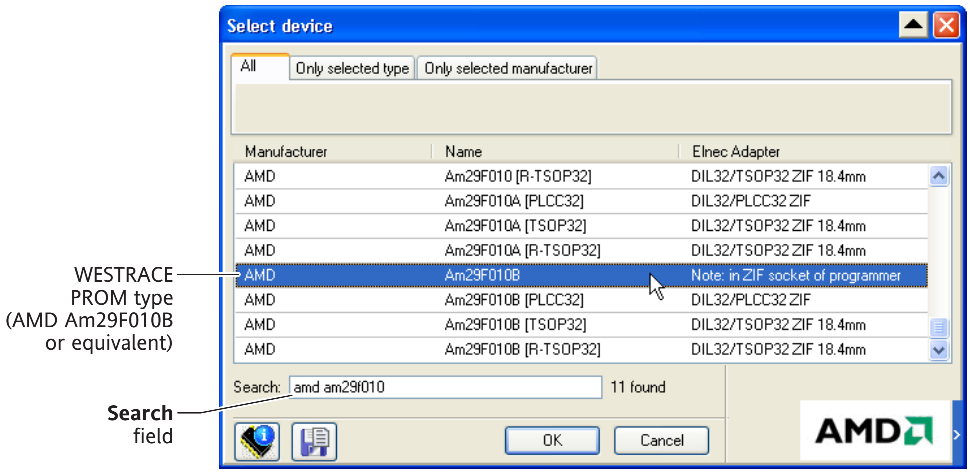


Figure 8.8 PG4UW remembers the previous session’s device type

i. Click the **Select** button (or press **Alt+F5**) to select from a complete list of devices.

You can also click the **Select/default** button (or press **F5**) to select from a list of recent devices.

The Select Device dialog box opens.

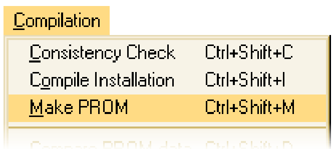


ii. Highlight the device—**AMD Am29F010B** (or equivalent) for WESTRACE—then click **OK**.

TIP: The list reduces as you type the device name into the Search field.

h) In GCSS:

i. Choose **Compilation** > **Make PROM**.



Alt c m
Ctrl+Shift+m

ii. If the “Recompile now?” dialog box opens, click **OK**.

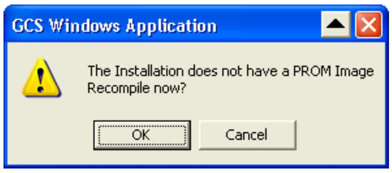


Figure 8.9 “Recompile now?” dialog box

GCSS generates the PROM image file (.img) and opens a report window that lists the (re)compilation details.

iii. Wait for the Generate PROM Data dialog box to open.

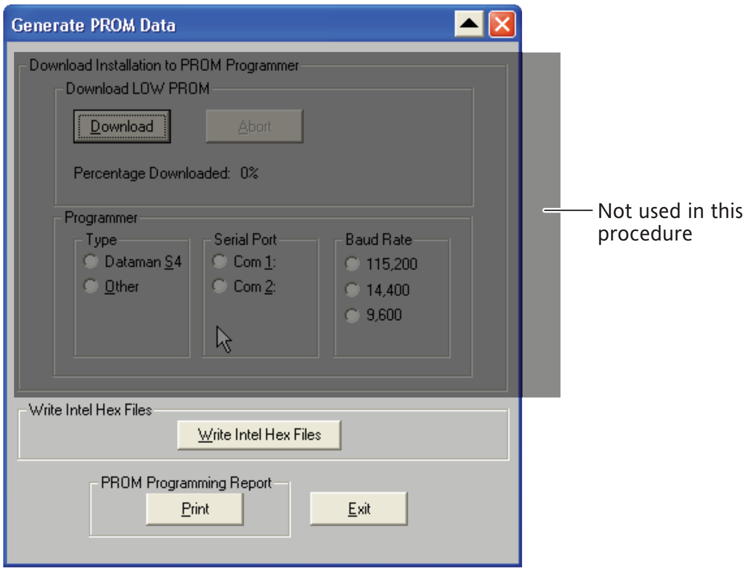


Figure 8.10 Generate PROM Data dialog box (Intel Hex files, BeeProg, USB procedure)

i) In the Generate PROM Data dialog box:

i. Click **Write Intel Hex Files** to create the _HI.hex file and the _LO.hex file in the installation directory (figures 8.2 and 8.11).

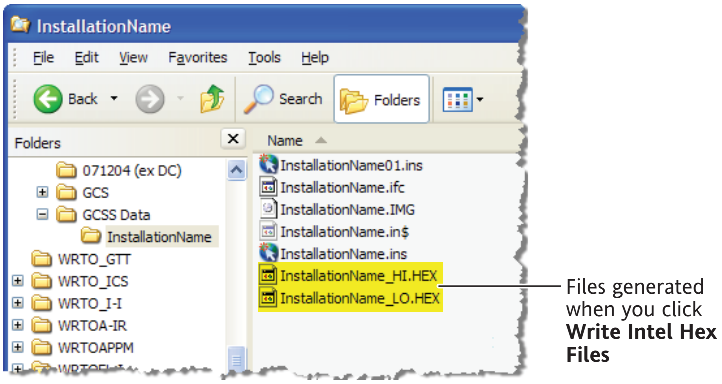


Figure 8.11 Installation directory containing Intel Hex files

ii. Click **Print** to print a PROM Download Report (see section 8.1.1.3).

The Print dialog box allows you to change paper orientation or select another printer if necessary. If you want to view the report before printing, cancel the Print dialog box and choose **File** > **Print Preview**.

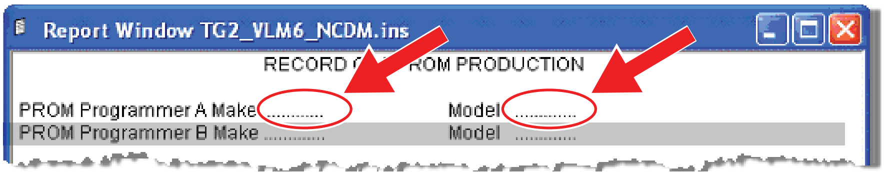
SAFETY

WARNING

A PROM Download Report (figure B.6) must be printed, completed and retained for every pair of PROMs (High Byte and Low Byte) used in a WESTRACE application.

iii. Click **Exit** in the dialog box to close it.

j) Complete the programmer “Make” and “Model” fields on the printed PROM Download Report (Make = **ELNEC**; Model = **BeeProg**, **BeeProg+** or **SmartProg2**).



8.1.2.1.2 Program the Low Byte PROM

a) Erase the BeeProg’s buffer memory:

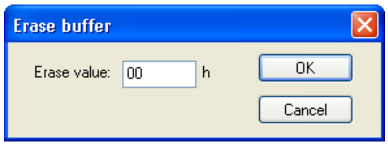
i. Click the **Buffer Erase** button  (or press **Ctrl+F2**).

NOTE: Do not confuse this with the Device Erase button  .

The Erase Buffer dialog box opens.

ii. Type **00** (zero zero) and click **OK**.

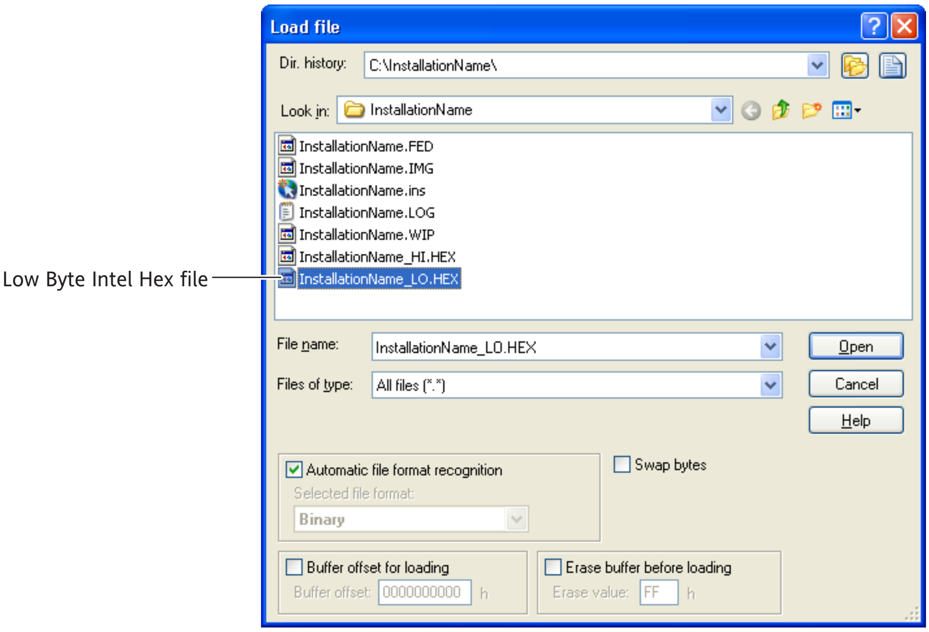
The PG4UW Log Window shows “Buffer is erased with value 00h.”



b) Download the Low Byte Intel Hex file to the BeeProg’s buffer[2]:

i. Click the **Load** button (or press **F3**).

The Load File dialog box opens.



ii. Locate the Low Byte Intel Hex file (eg InstallationName_LO.hex) in the installation directory, highlight it and click **Open**.

The PG4UW Log Window shows “File loading successful!”

The Low Byte Intel Hex file is now loaded into the BeeProg buffer, ready for burning into the Low Byte PROM.

c) Insert the Low Byte PROM into the BeeProg.

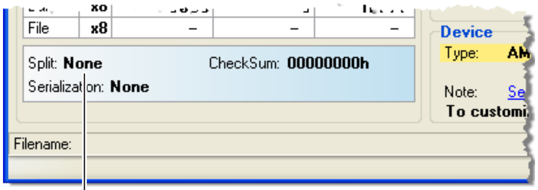
[2] Also known as RAM, Random Access Memory or “memory”

d) Specify splitting:


Split:
None

i. Click the **Split: None** button.

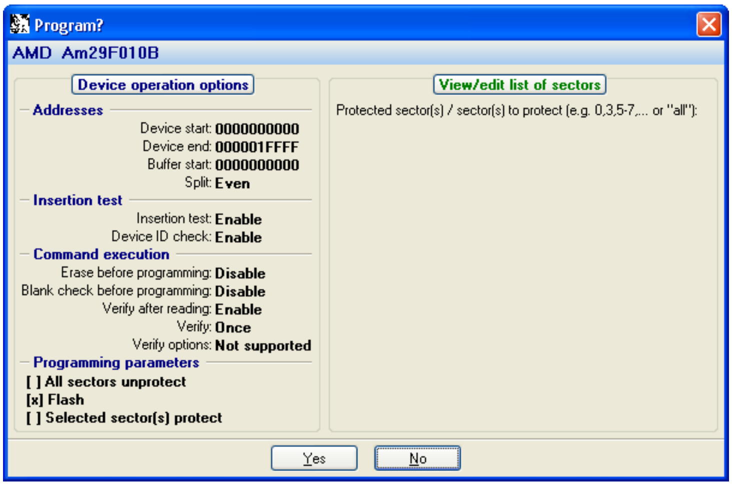
ii. Ensure that Split: **None** is shown in the lower left corner of the PG4UW main screen.



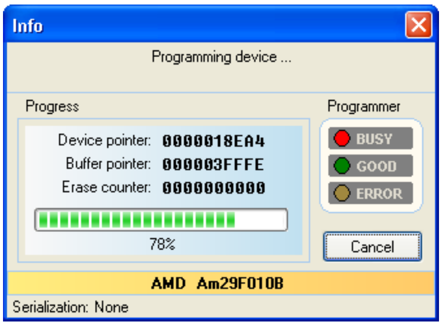
Split setting (**None** when burning from Intel Hex files)

e) Burn ("blow") the Low Byte PROM:

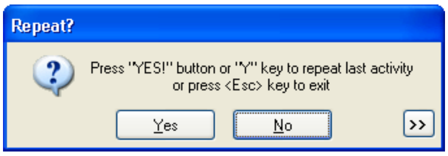
i. Click the **Program** button (or press **F9**).
The Program? dialog box opens.



ii. Click **Yes**.
An Info dialog box reports the progress.



The Repeat? dialog box opens at the end of the download.



iii. If the Info dialog box header bar changes to green and displays "Programming device - O.K.", click **No** in the Repeat? dialog box.
If the Info dialog box reports an error, repeat the erase and burn procedures. Then if the Info dialog box still reports an error, repeat the erase and burn procedures with a new PROM.

f) Use the BeeProg **Verify** function to confirm that the BeeProg's buffer contents and the PROM contents are the same:

i. Click the **Verify** button (or press **F8**).

An Info dialog box reports the progress.

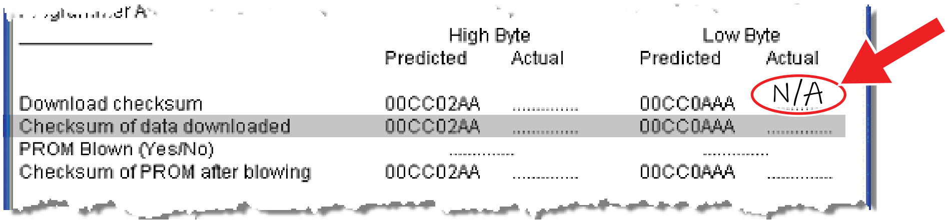
The Repeat? dialog box opens at the end of the process.

ii. If the Info dialog box header bar changes to green and displays "Verifying device - O.K.", click **No** in the Repeat? dialog box.

If the Info dialog box reports an error, repeat the erase and burn procedures. Then if the Info dialog box still reports an error, repeat the erase and burn procedures with a new PROM.

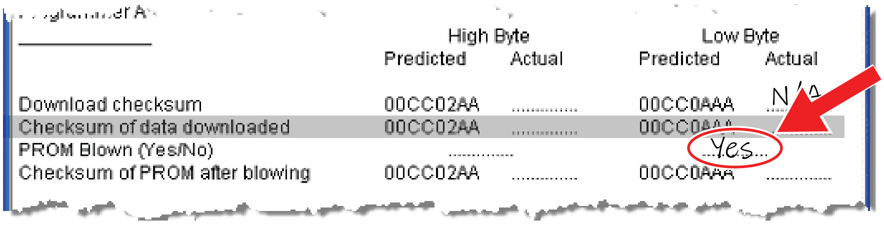
g) In the PROM Download Report:

i. Write **Not Applicable** in the Low Byte "**Download checksum**" field.



	High Byte		Low Byte	
	Predicted	Actual	Predicted	Actual
Download checksum	00CC02AA	....	00CC0AAA	N/A
Checksum of data downloaded	00CC02AA	....	00CC0AAA	....
PROM Blown (Yes/No)	....		....	
Checksum of PROM after blowing	00CC02AA	....	00CC0AAA	....

ii. Write **Yes** in the Low Byte "PROM Blown" field on the report.



	High Byte		Low Byte	
	Predicted	Actual	Predicted	Actual
Download checksum	00CC02AA	....	00CC0AAA	N/A
Checksum of data downloaded	00CC02AA	....	00CC0AAA	....
PROM Blown (Yes/No)	....		Yes	
Checksum of PROM after blowing	00CC02AA	....	00CC0AAA	....

The Low Byte PROM is now programmed.

8.1.2.1.3 Program the High Byte PROM

a) Repeat the steps in section 8.1.2.1.2, "Program the Low Byte PROM" but use the High Byte PROM instead, and the High Byte fields on the PROM Download Report.

The High Byte PROM is now programmed.

8.1.2.1.4 Verify the PROMs

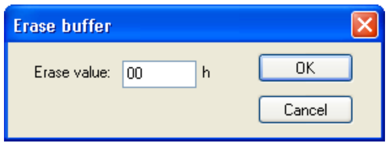
Note: ***In this section, do not use the Buffer > Checksum command (or the button) because this does not calculate checksums correctly.***

a) Verify the burnt Low Byte PROM by comparing the checksum of its contents (derived using the BeeProg) with the value printed on the PROM Download Report:

i. Insert the Low Byte PROM into the BeeProg.

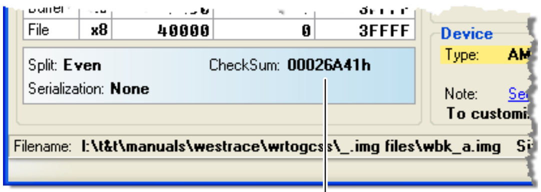
ii. Click the **Buffer Erase** button (or press **Ctrl+F2**) to clear the BeeProg buffer.
Do not confuse this with the Device Erase button .
The Erase Buffer dialog box opens.

iii. Type **00** (zero zero) and click **OK**.
The PG4UW Log Window shows "Buffer is erased with value 00h."



iv. Click the **Read** button (or press **F7**).
The contents of the PROM are read into the BeeProg buffer.

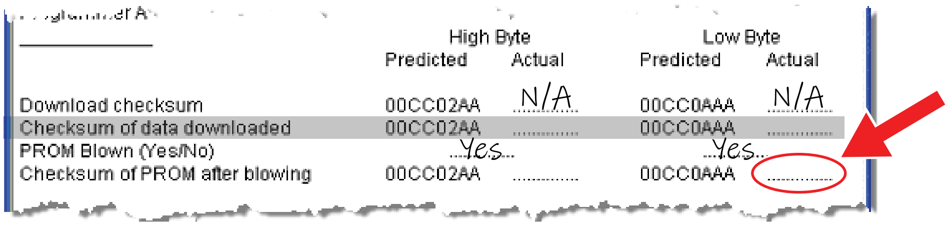
The checksum is shown near the lower left corner of the PG4UW main screen.



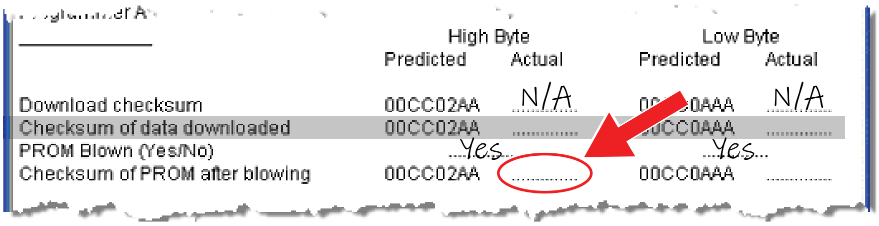
Checksum of PROM contents

v. Compare the checksum shown in the lower left corner of the main screen with the Low Byte "**Checksum of PROM after blowing**" printed on the PROM Download Report. They must be identical.
If the checksums differ, either discard the PROM, or erase and re-program it.

vi. Write the checksum in the Low Byte "Checksum of PROM after blowing" field on the report.



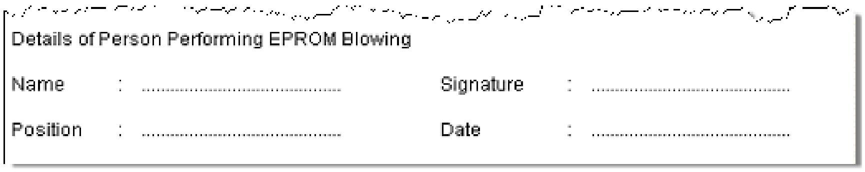
b) Remove the Low Byte PROM and repeat step **a** for the High Byte PROM.



	High Byte Predicted	High Byte Actual	Low Byte Predicted	Low Byte Actual
Download checksum	00CC02AA	N/A	00CC0AAA	N/A
Checksum of data downloaded	00CC02AA	....	00CC0AAA	....
PROM Blown (Yes/No)	Yes		Yes	
Checksum of PROM after blowing	00CC02AA	....	00CC0AAA	....

8.1.2.1.5 Final Details

a) Complete the “Details…” fields at the bottom of the report.



Details of Person Performing EPROM Blowing

Name	:	Signature	: ....
Position	:	Date	: ....

b) File the PROM Download Report with the project documentation.

The PROMs have now been successfully programmed and verified, and are ready to be installed in the WESTRACE VLM.

TIP: For additional security prior to installing the PROMs on site, you can also carry out the procedure presented in section 8.3, “Uploading, Decompiling and Comparing Installations (Optional Verification Method)”.

8.1.2.2 Procedure 2: Image File—BeeProg USB Example

This section gives an example of programming VLM PROMs from the image file (.img) that GCSS automatically generates when you begin the "Make PROM" procedure (figure 8.2).

A BeeProg programmer, connected to the computer by USB, is used in this example.

Figure 8.12 ELNEC BeeProg PROM programmer

Note: ***Although these instructions refer to the basic BeeProg, they can be used, with minor adaptation, for BeeProg+ and SmartProg2 programmers.***

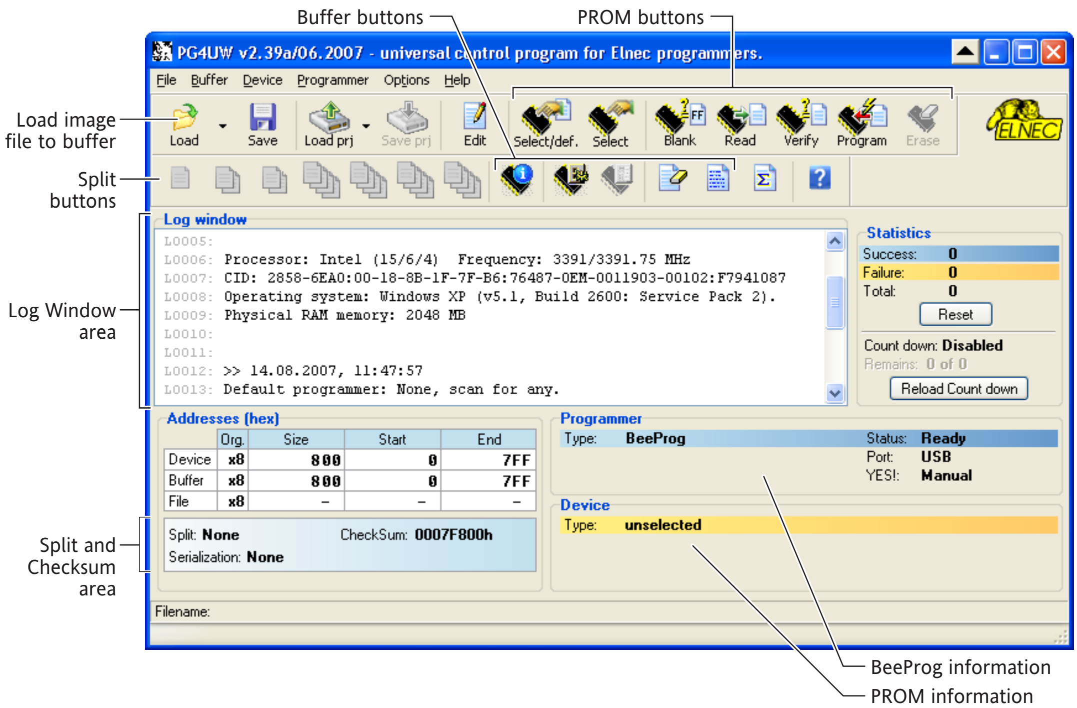


Figure 8.13 BeeProg PG4UW main screen—initial view

8.1.2.2.1 Preliminaries

a) Ensure that the installation's Vital Source File has been approved (see section 1.10).

b) Install the BeeProg software on the computer as described in section 8.1.2.5.1.

c) Use a USB cable to connect the BeeProg to the computer, then power the BeeProg.

See section 8.1.2.5.2, “Setting Up the BeeProg” for full details, especially if this is the first time the BeeProg has been connected to this computer.

d) Start the BeeProg Control Program PG4UW (figure 8.13) and establish communication with the BeeProg as described in section 8.1.2.5.3.

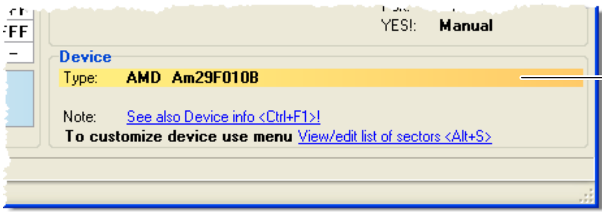
e) Erase the PROMs (new and used) as described in section 8.1.2.5.4.

f) Label one PROM “Low” and the other “High”.

Use a sticker, or write directly on the PROM. We recommend that you also label each PROM with the installation name and the date.

g) Select the type (manufacturer and part number) of the PROMs you are using (eg **AMD Am29F010B** or AMD Am28F010 for WESTRACE):

TIP: The PG4UW software remembers the last-used device type. Therefore, if the required PROM type is already showing in the Device Type area (figure 8.14), you can skip to step **h** below.



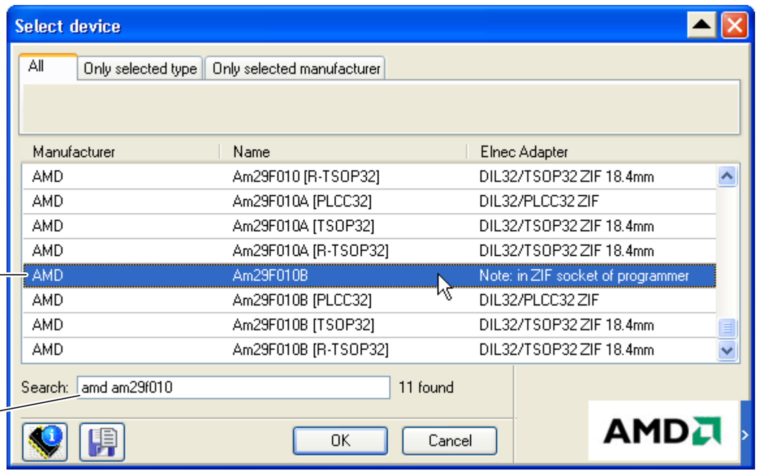
WESTRACE PROM type (AMD Am29F010B or equivalent)

Figure 8.14 PG4UW remembers the previous session’s device type

i. Click the **Select** button (or press **Alt+F5**) to select from a complete list of devices.

You can also click the **Select/def**ault button (or press **F5**) to select from a list of recent devices.

The Select Device dialog box opens.



WESTRACE PROM type (AMD Am29F010B or equivalent)

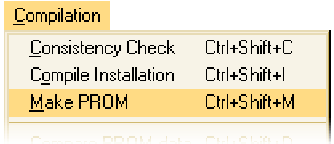
Search field

ii. Highlight the device—**AMD Am29F010B** (or equivalent) for WESTRACE—then click **OK**.

TIP: The list reduces as you type the device name into the Search field.

h) In GCSS:

i. Choose **Compilation** > **Make PROM**.



Alt c m
Ctrl+Shift+m

ii. If the “Recompile now?” dialog box opens, click **OK**.

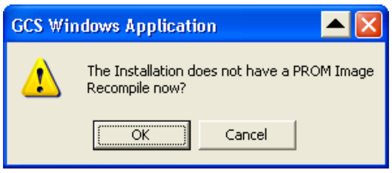


Figure 8.15 “Recompile now?” dialog box

GCSS generates the PROM image file (.img) and opens a report window that lists the (re)compilation details.

iii. Wait for the Generate PROM Data dialog box to open.

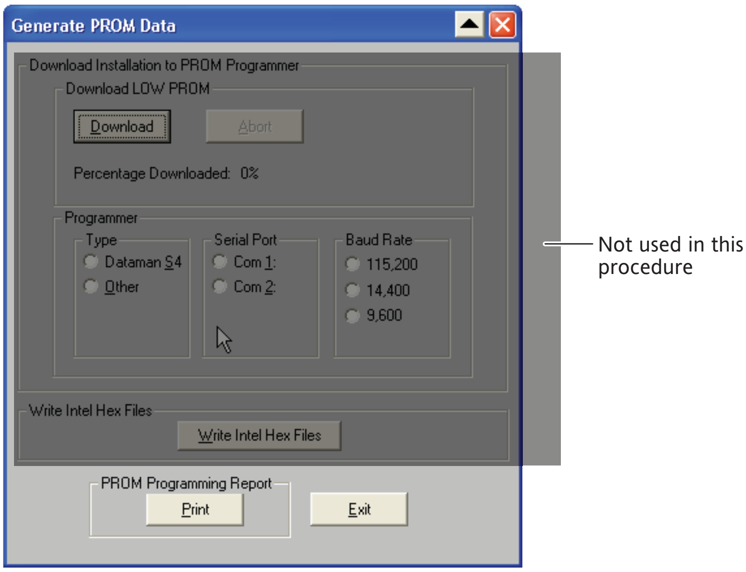


Figure 8.16 Generate PROM Data dialog box (image file, BeeProg, USB procedure)

i) In the Generate PROM Data dialog box:

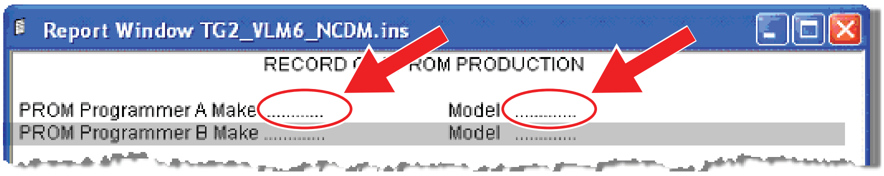
i. Click **Print** to print a PROM Download Report (see section 8.1.1.3).

The Print dialog box allows you to change paper orientation or select another printer if necessary. If you want to view the report before printing, cancel the Print dialog box and choose **File** > **Print Preview**.

A PROM Download Report (figure B.6) must be printed, completed and retained for every pair of PROMs (High Byte and Low Byte) used in a WESTRACE application.

ii. Click **Exit** in the dialog box to close it.

j) Complete the programmer "Make" and "Model" fields on the printed PROM Download Report (Make = **ELNEC**; Model = **BeeProg**, **BeeProg**+ or **SmartProg2**).



8.1.2.2.2 Download the Image File to the BeeProg

a) Erase the BeeProg's buffer memory:

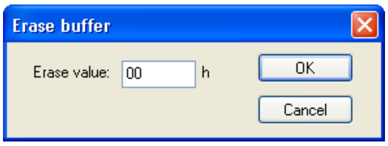
i. Click the **Buffer Erase** button  (or press **Ctrl+F2**).

NOTE: Do not confuse this with the Device Erase button  .

The Erase Buffer dialog box opens.

ii. Type **00** (zero zero) and click **OK**.

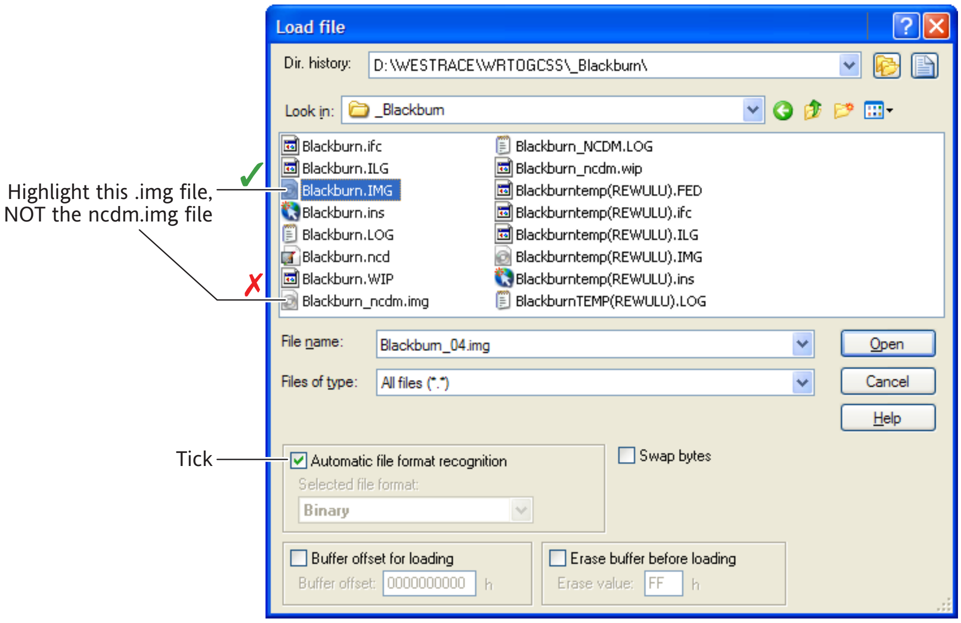
The PG4UW Log Window shows "Buffer is erased with value 00h."



b) Download the image file (eg StationName.img) to the BeeProg's RAM (Random Access Memory, "memory" or "buffer"):

i. Click the **Load** button (or press **F3**).

The Load File dialog box opens.



ii. Locate the image file and highlight it.

You generated the image file earlier (step **h** on page 8-18). It will be in the same directory as the installation files (.ins, .nvc, .ncd) and interface files (.ifc).

NOTE: There may be several image files in the installation directory—be sure to choose the one with the shortest name, eg **StationName.img** and NOT **StationName_ncdm.img**.

iii. Ensure that **Automatic file format recognition** is ticked.

iv. Click **Open**.
The PG4UW Log Window shows “File loading successful!”
The image file is now loaded into the BeeProg RAM, ready for burning into the PROMs.

8.1.2.2.3 Burn the Low Byte PROM

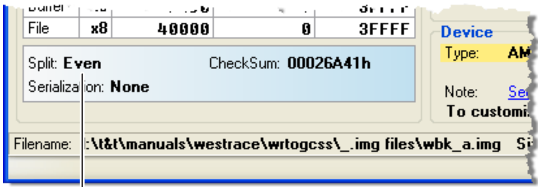
a) Insert the Low Byte PROM into the BeeProg.

b) Specify splitting:

Split: Even

i. Click the **Split: Even** button.

ii. Ensure that Split: **Even** is shown in the lower left corner of the PG4UW main screen.

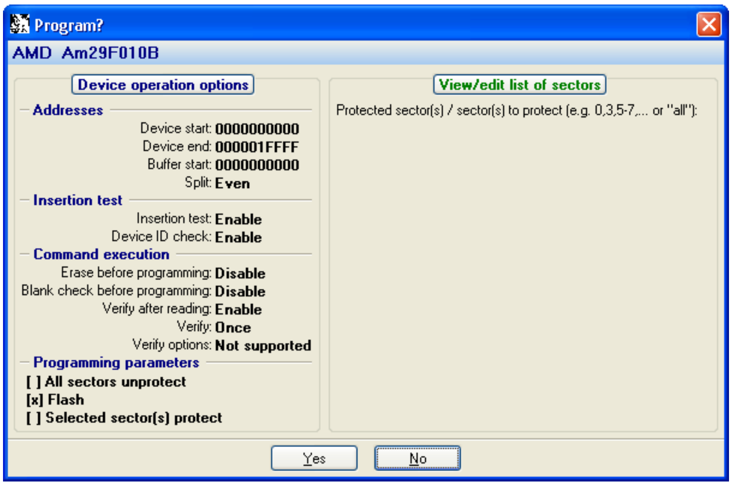


Split setting (**Even** for Low-Byte WESTRACE PROMs)

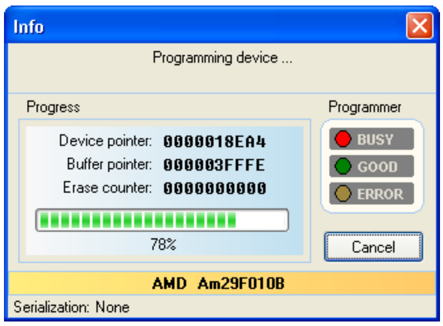
c) Burn the Low Byte PROM:

Program

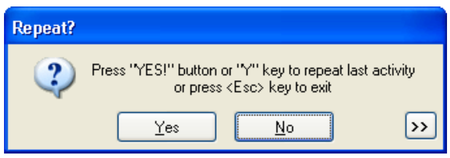
i. Click the **Program** button (or press **F9**).
The Program? dialog box opens.



ii. Click **Yes**.
An Info dialog box reports the progress.



The Repeat? dialog box opens at the end of the download.



iii. If the Info dialog box header changes to green and displays "Programming device - O.K.", click **No** in the Repeat? dialog box.

If the Info dialog box reports an error, repeat the erase and burn procedures. Then if the Info dialog box still reports an error, repeat the erase and burn procedures with a new PROM.

d) Use the BeeProg's Verify function to confirm that the BeeProg's buffer contents and the PROM contents are the same:

i. Click the **Verify** button (or press **F8**).

An Info dialog box reports the progress.

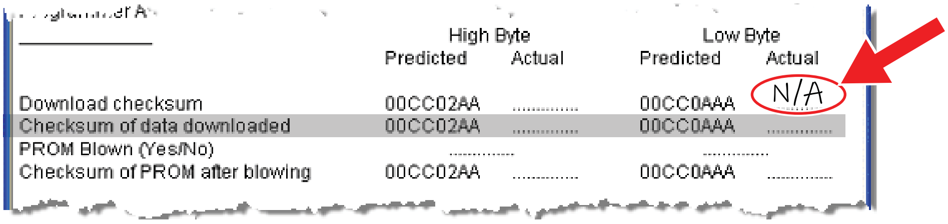
The Repeat? dialog box opens at the end of the process.

ii. If the Info dialog box header changes to green and displays "Verifying device - O.K.", click **No** in the Repeat? dialog box.

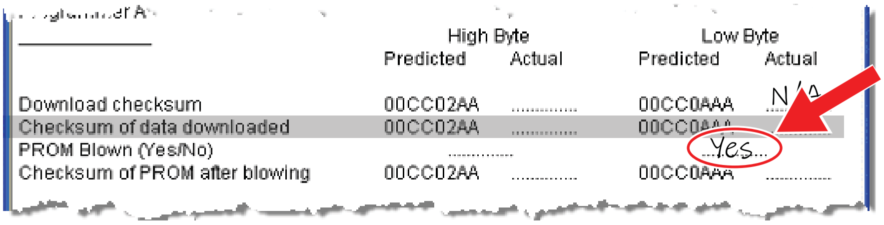
If the Info dialog box reports an error, repeat the erase and burn procedures. Then if the Info dialog box still reports an error, repeat the erase and burn procedures with a new PROM.

e) In the PROM Download Report:

i. Write **Not Applicable** in the Low Byte "**Download checksum**" field.



ii. Write **Yes** in the Low Byte "PROM Blown" field on the report.



The Low Byte PROM is now programmed.

8.1.2.2.4 Burn the High Byte PROM

a) Repeat the steps in section 8.1.2.2.3, "Burn the Low Byte PROM" using the High Byte PROM instead, and making appropriate changes:

Split: Odd

- Click the **Split: Odd** button when specifying the splitting for the High Byte PROM, and ensure that Split: **Odd** is shown in the lower left corner of the PG4UW main screen.
- Use the **High** Byte fields on the PROM Download Report.

The High Byte PROM is now programmed.

8.1.2.2.5 Verify the PROMs

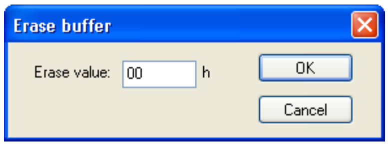
Note: ***In this section, do not use the Buffer > Checksum command (or the button) because this does not calculate checksums correctly.***

a) Verify the burnt Low Byte PROM by comparing the checksum of its contents (derived using the BeeProg) with the value printed on the PROM Download Report:

i. Insert the Low Byte PROM into the BeeProg.

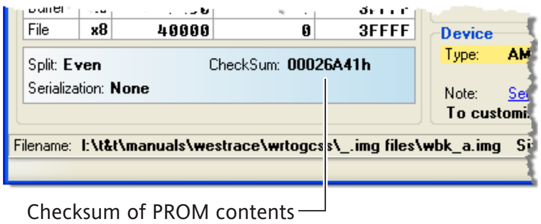
ii. Click the **Buffer Erase** button (or press **Ctrl+F2**) to clear the BeeProg buffer.
Do not confuse this with the Device Erase button.
The Erase Buffer dialog box opens.

iii. Type **00** (zero zero) and click **OK**.
The PG4UW Log Window shows "Buffer is erased with value 00h."



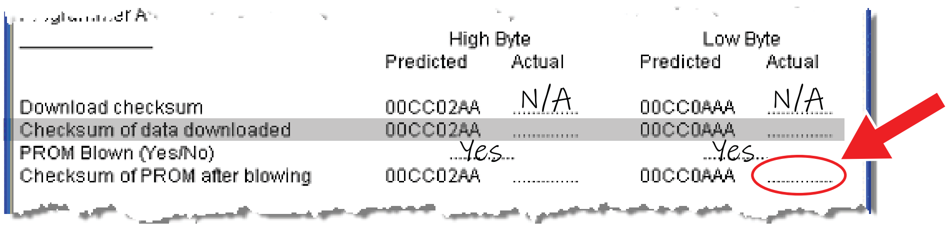
iv. Click the **Read** button (or press **F7**).
The contents of the PROM are read into the BeeProg buffer.

The checksum is shown near the lower left corner of the PG4UW main screen.



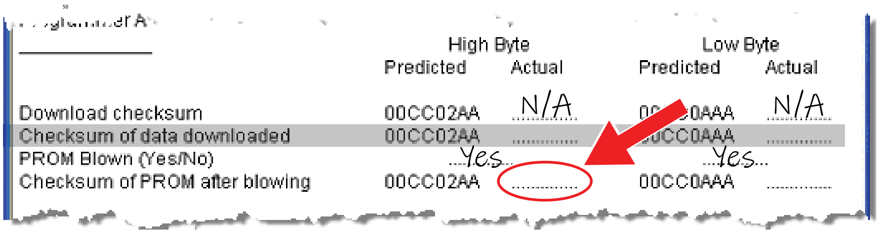
v. Compare the checksum shown in the lower left corner of the main screen with the Low Byte "**Checksum of PROM after blowing**" printed on the PROM Download Report. They must be identical.
If the checksums differ, either discard the PROM, or erase and re-program it.

vi. Write the checksum in the Low Byte "Checksum of PROM after blowing" field on the report.



	High Byte Predicted	High Byte Actual	Low Byte Predicted	Low Byte Actual
Download checksum	00CC02AA	N/A	00CC0AAA	N/A
Checksum of data downloaded	00CC02AA	....	00CC0AAA	....
PROM Blown (Yes/No)		Yes		Yes
Checksum of PROM after blowing	00CC02AA	....	00CC0AAA	....

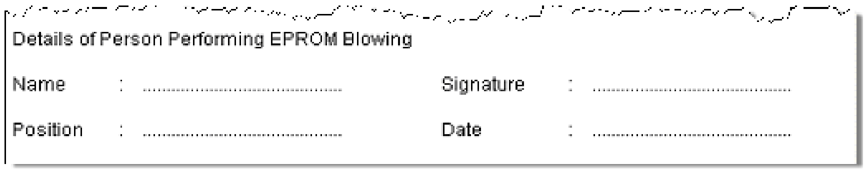
b) Remove the Low Byte PROM and repeat step **a** for the High Byte PROM.



	High Byte		Low Byte	
	Predicted	Actual	Predicted	Actual
Download checksum	00CC02AA	N/A	00CC0AAA	N/A
Checksum of data downloaded	00CC02AA	....	00CC0AAA	....
PROM Blown (Yes/No)	Yes		Yes	
Checksum of PROM after blowing	00CC02AA	....	00CC0AAA	....

8.1.2.2.6 Final Details

a) Complete the “Details…” fields at the bottom of the report.



Details of Person Performing EPROM Blowing

Name : .... Signature : ....

Position : .... Date : ....

b) File the PROM Download Report with the project documentation.

The PROMs have now been successfully programmed and verified, and are ready to be installed in the WESTRACE VLM.

TIP: For additional security prior to installing the PROMs on site, you can also carry out the procedure presented in section 8.3, “Uploading, Decompiling and Comparing Installations (Optional Verification Method)”.

8.1.2.3 Procedure 3: Dataman S4

This section explains how to use GCSS to burn and verify VLM PROMs using the Dataman S4 serial PROM programmer.

Figure 8.17 Dataman S4 PROM programmer

8.1.2.3.1 Preliminaries

a) Ensure that the installation's Vital Source File has been approved (see section 1.10).

b) Connect the Dataman S4 to a serial port on the computer (figure 10.1 gives cable details), then power the Dataman.

c) Configure the Dataman:

INSTRUCTION	DISPLAY	COMMENT
Press **Setup.**	**> SETUP**	Red Button.
File: INTEL.		Use the ← → keys to select. This prepares the Dataman to receive Intel Hex data.
Press **Enter.**		
Baud: 9600.		Use the ← → keys to select. You may select a higher Baud rate (to minimise download time), but your selection here must match the computer's Baud rate, which you select later (see step **h**-i on page 8-26, and figure 8.19).
Press **Enter.**		
Auto—press **Spacebar.**		
Handshake: /RTS.		Use the ← → keys to select. "RTS" may cause some problems. Try "NONE" if problems are encountered.
Press **Enter.**		
Download Sound off.		Use the ← → keys to select.
UL BYTES/LINE 20.		
Press **Enter.**		
File: INTEL.		

Table 8.2 Configuring the Dataman

INSTRUCTION	DISPLAY	COMMENT
Press **Esc.**		
Press **RCVE.**	**> Receive Intel**	Grey Button.
Press **Enter.**		Red Button.
	ESC >	Dataman is ready for download.

Table 8.2 Configuring the Dataman

d) Erase the PROMs (new and used):

INSTRUCTION	DISPLAY	COMMENT
Insert one of the PROMs.		
Press **FUNC.**	*	Grey Button.
Press **BURN.**	> **>ERASE** **Intel 28F010** **12.00V Burn Pin 1**	Green Button.
Press **Enter.**	**Intel 28F010** **12.00V** **Writing all zeros** **Address ---- ****	---- ** These mnemonics toggle while program function is running.
	Address ---- ** **Erasing IFF00** **DONE** >	Dataman sounds once when complete.
Repeat for the other PROM.		

Table 8.3 Erasing PROMs (Dataman)

e) Label one PROM "Low" and the other "High".

Use a sticker, or write directly on the PROM. We recommend that you also label each PROM with the installation name and the date.

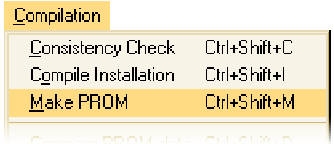
f) Select the PROM type (manufacturer and part number):

INSTRUCTION	DISPLAY	COMMENT
Press **PROM**		
Select either Intel PROM P28F0I0, or	**Intel 28F010** **12.00V Burn Pin 1** **10uS No Overprog** **ALg 3F, Ref 081E**	Use the ↑↓ keys to select PROM manufacturer. Use the ← → keys to select PROM part number.
AMD PROM.	**AMD 28F010** **12.00V Burn Pin 1** **10uS No Overprog** **ALg 70, Ref 001B**	
Press **Enter.**		To accept selection.

Table 8.4 Selecting the PROM Type (Dataman)

g) In GCSS:

i. Choose **Compilation** > **Make PROM**.



Alt c m
Ctrl+Shift+m

ii. If the "Recompile now?" dialog box opens, click **OK**.

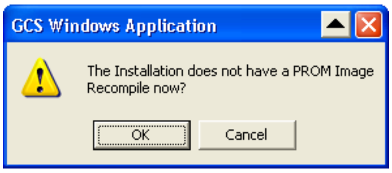


Figure 8.18 "Recompile now?" dialog box

GCSS generates the PROM image file (.img) and opens a report window that lists the (re)compilation details.

iii. Wait for the Generate PROM Data dialog box to open.

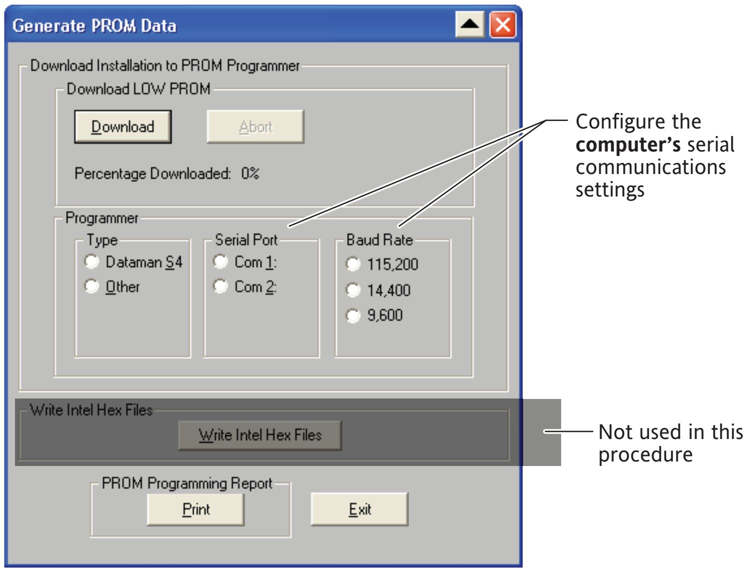


Figure 8.19 Generate PROM Data dialog box (Dataman procedure)

h) In the Generate PROM Data dialog box:

i. Select **Dataman S4**, the computer's **Serial Port** to which the Dataman is connected, and the **Baud Rate** to which you configured the Dataman (step **c** on page 8-24).

ii. Click **Print** (at the bottom of the dialog box) to print an installation-specific PROM Download Report (see section 8.1.1.3).

The Print dialog box allows you to change paper orientation or select another printer if necessary. If you want to view the report before printing, cancel the Print dialog box and choose **File** > **Print Preview**.

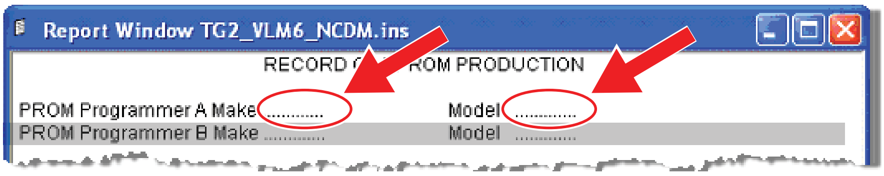
SAFETY
WARNING

A PROM Download Report (figure B.6) must be printed, completed and retained for every pair of PROMs used in a WESTRACE application.

iii. Click **Download** (at the top of the dialog box).
Note that you can click **Abort** at any time to stop the download process.

iv. Click **Exit** in the dialog box to close it.

i) Complete the programmer “Make” and “Model” fields on the printed PROM Download Report (Make = **Dataman**, Model = **S4**).



8.1.2.3.2 Program the Low Byte PROM

a) Download the Low Byte data to the Dataman:

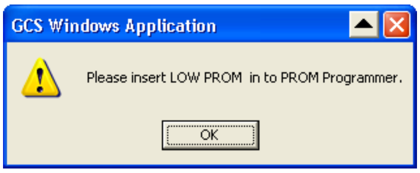
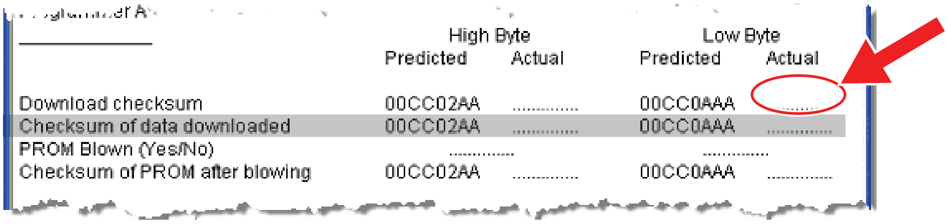
INSTRUCTION	DISPLAY	COMMENT
		Message on PC.
Click **OK** on PC.		Two beeps from Dataman when download starts; no change to Dataman screen.
		Percentage download on PC starts: 0—100% (when completed).
	> RECEIVE INTEL **> ESC** **>**	Dataman sounds one beep when download completed.
Burn the low PROM and read Checksum		Message on PC.
Insert the PROM labelled “Low” into the Dataman and lock it down.		
Press **Sum.**		Grey Key—to get checksum of contents of Dataman RAM.
	Address IFFXX **Same** **> CHECKSUM RAM** **00000,0FFFF**	Use the → key and mnemonic pad to set 0FFFF for HVLM128 or HVLM128A, 1FFFF for VLM5 or VLM6 if required.
Press **Enter.**		Dataman sounds five beeps for HVLM128 or HVLM128A, ten beeps for VLM5 or VLM6.
	> CHECKSUM RAM **00000,0FFFF** **SUM = 005FD7AA** **>**	Should match the Low Byte “**Download Checksum**” on the PROM Download Report (figure B.6).

Table 8.5 Downloading Low Byte data to the Dataman [sheet 1 of 2]

INSTRUCTION	DISPLAY	COMMENT
Compare the checksum currently displayed by the Dataman (this is the checksum of the data in the Dataman's RAM, after download from GCSS) with the Low Byte "**Download checksum**" shown on the PROM Download Report (printed in step **ii** on page 8-26)—they must be identical. (If the checksums differ, either discard the PROM, or erase and re-program it.) Write the value in the Low Byte "Download checksum" field on the report.		



	High Byte Predicted	High Byte Actual	Low Byte Predicted	Low Byte Actual
Download checksum	00CC02AA	....	00CC0AAA	....
Checksum of data downloaded	00CC02AA	....	00CC0AAA	....
PROM Blown (Yes/No)		....		....
Checksum of PROM after blowing	00CC02AA	....	00CC0AAA	....

Table 8.5 Downloading Low Byte data to the Dataman [sheet 2 of 2]

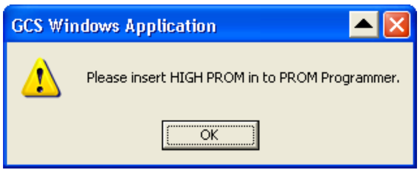
b) Burn ("blow") the Low Byte PROM:

INSTRUCTION	DISPLAY	COMMENT
Press **Burn.**	**>BURN** **Intel 28 F010** **12.00V Burn Pin 1** **00000—IFFFF = 00000**	Green Button. PROM LABEL:
		KBS V01.0U Lo 0060A9AA 15.7.96
Press **Enter.**	**Intel 28F010** **12.00V Burn Pin 1** **00000—IFFFF = 00000** **Address ---- ****	---- ** These mnemonics toggle while program function is running.
		Dataman sounds several beeps when completed.
	00000—IFFFF = 00000 **Address IFFXX** **Same** **>**	
Click **OK** on PC.		

Table 8.6 Burning the Low Byte PROM (Dataman)

8.1.2.3.3 Program the High Byte PROM

a) Download the High Byte data to the Dataman:

INSTRUCTION	DISPLAY	COMMENT
		Message on PC.
Click **OK** on PC.		Two beeps from Dataman when download starts; no change to Dataman screen.
		Percentage download on PC starts: 0—100% (when completed).
	> RECEIVE INTEL **> ESC** >	Dataman sounds one beep when download completed.
Burn the high PROM and read Checksum		Message on PC.
Insert the PROM labelled "High" into the Dataman and lock it down.		
Press **Sum.**		Grey Key—to get checksum of contents of Dataman RAM.
	Address IFFXX **Same** **> CHECKSUM RAM** **00000,0FFFF**	Use the → key and mnemonic pad to set 0FFFF for HVLM128 or HVLM128A, 1FFFF for VLM5 or VLM6 if required.
Press **Enter.**		Dataman sounds five beeps for HVLM128 or HVLM128A, ten beeps for VLM5 or VLM6.
	> CHECKSUM RAM **00000,0FFFF** **SUM = 005FD7AA** >	Should match the High Byte "Download Checksum" printed on the PROM Download Report (figure B.6).

Compare the checksum currently displayed by the Dataman (this is the checksum of the data in the Dataman's RAM, after download from GCSS) with the High Byte "**Download checksum**" shown on the PROM Download Report (printed in step **ii** on page 8-26)—they must be identical. (If the checksums differ, either discard the PROM, or erase and re-program it.)

Write the value in the High Byte "Download checksum" field on the report.

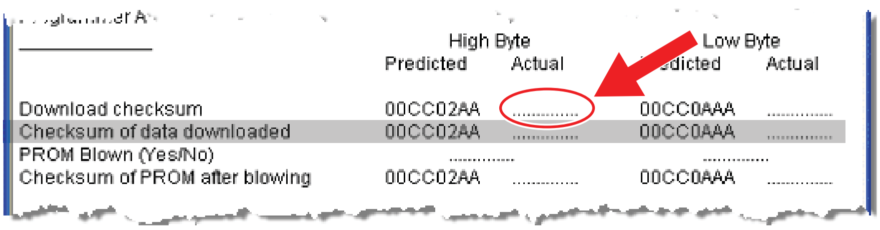


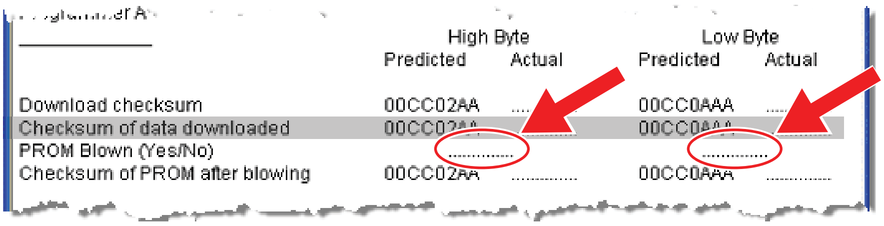
Table 8.7 Downloading High Byte data to the Dataman

b) Burn ("blow") the High Byte PROM:

INSTRUCTION	DISPLAY	COMMENT
Press **Burn.**	**>BURN** **Intel 28 F010** **12.00V Burn Pin 1** **00000—IFFFF = 00000**	Green Button. PROM LABEL:
		KBS V01.0U Lo 0060A9AA 15.7.96
Press **Enter**	**Intel 28F010** **12.00V Burn Pin 1** **00000—IFFFF = 00000** **Address ---- **** **#**	---- ** # These mnemonics toggle while program function is running.
		Dataman sounds six beeps when completed.
	00000—IFFFF = 00000 **Address IFFXX** **Same** **>**	
Click **OK** on PC.		

Table 8.8 Burning the High Byte PROM (Dataman)

c) Fill in the "PROM Blown (Yes/No)" fields on the PROM Download Report.



8.1.2.3.4 Verify the PROMs

INSTRUCTION	DISPLAY	COMMENT
Insert one of the burnt PROMs into the Dataman.		
Press **LOAD.**		
	> Load 28F010 **00000—IFFFF = 00000**	
Press **Enter.**		
		Dataman sounds several beeps while image loads into RAM.
	00000—IFFFF = 00000 **>**	
Press **SUM.**		Grey key—to get check sum.

Table 8.9 Verifying burnt PROMs (Dataman) [sheet 1 of 2]

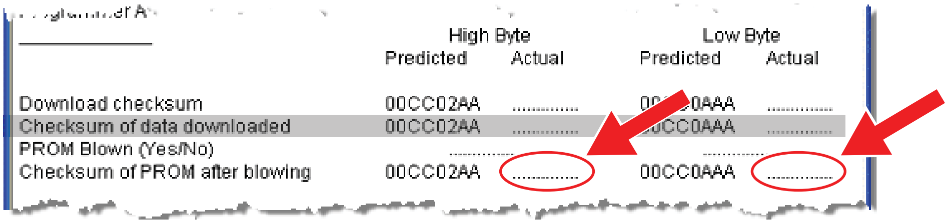
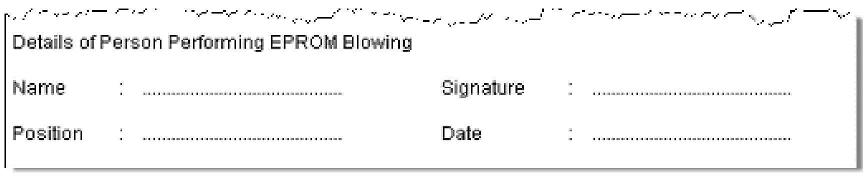
INSTRUCTION	DISPLAY	COMMENT
	> CHECKSUM RAM 00000,0FFFF	
Press **Enter.**		Dataman sounds five beeps for HVLM128 or HVLM128A, ten beeps for VLM5 or VLM6.
	SUM = 005A5DAA **>**	Should match the appropriate “Checksum of PROM after blowing” printed on the PROM Download Report (figure B.6).
Compare the checksum displayed by the Dataman with the appropriate “**Checksum of PROM after blowing**” shown on the PROM Download Report—they must be identical. (If the checksums differ, either discard the PROM, or erase and re-program it.) Write the value in the appropriate “Checksum of PROM after blowing” field on the report.		
		
Insert the other burnt PROM into the Dataman and repeat the procedure.		

Table 8.9 Verifying burnt PROMs (Dataman) [sheet 2 of 2]

8.1.2.3.5 Final Details

a) Complete the “Details…” fields at the bottom of the report.



b) File the PROM Download Report with the project documentation.

The PROMs have been successfully programmed and verified, and are ready to be installed in the WESTRACE VLM.

c) Click **Exit** in the Generate PROM Data dialog box (figure 8.19).

TIP: For additional security prior to installing the PROMs on site, you can also carry out the procedure presented in section 8.3, “Uploading, Decompiling and Comparing Installations (Optional Verification Method)”.

8.1.2.4 Procedure 4: “Other” Serial Programmers

GCSS can communicate with other serial PROM programmers to control data download and PROM burning. Data is transmitted low byte first followed by high byte, in binary format with Baud rate configured by the user to be either 115,200, 14,400 or 9,600 Baud.

The procedure below is an outline of the use of GCSS to burn and verify VLM PROMs using an “other” serial programmer. See also:

- section 8.1.2.3, “Procedure 3: Dataman S4” as a general guide;
- your serial programmer’s manual for detailed information.

8.1.2.4.1 Preliminaries

a) Ensure that the installation’s Vital Source File has been approved (see section 1.10).

b) Connect the “other” PROM programmer to a serial port on the computer (figure 10.1 shows a typical cable), and power the programmer.

c) Configure the programmer for communication with GCSS.

Refer to the manual for your specific PROM programmer.

d) Erase the PROMs (new and used).

Refer to the manual for your specific PROM programmer.

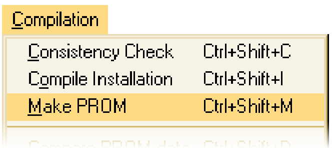
e) Label one PROM “Low” and the other “High”.

Use a sticker, or write directly on the PROM. We recommend that you also label each PROM with the installation name and the date.

f) Select the PROM type (manufacturer and part number)

g) In GCSS:

i. Choose **Compilation** > **Make PROM**.



Alt c m
Ctrl+Shift+m

ii. If the “Recompile now?” dialog box opens, click **OK**.

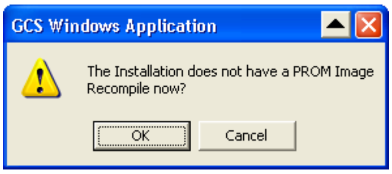


Figure 8.20 “Recompile now?” dialog box

iii. Wait for the Generate PROM Data dialog box to open.

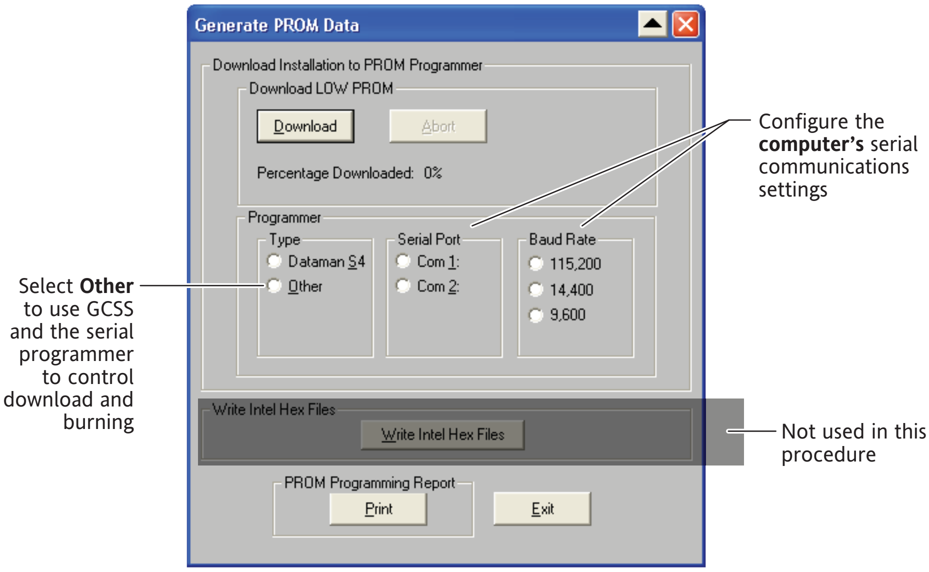


Figure 8.21 Generate PROM Data dialog box (“Other” programmer procedure

h) In the Generate PROM Data dialog box:

i. Select **Other**, the computer’s **Serial Port** to which the programmer is connected, and the **Baud Rate** to which you configured the programmer (step **c** on page 8-32).

ii. Click **Print** (at the bottom of the dialog box) to print an installation-specific PROM Download Report (see section 8.1.1.3).

The Print dialog box allows you to change paper orientation or select another printer if necessary. If you want to view the report before printing, cancel the Print dialog box and choose **File** > **Print Preview**.

SAFETY WARNING

A PROM Download Report (figure B.6) must be printed, completed and retained for every pair of PROMs used in a WESTRACE application.

iii. Click **Download** (at the top of the dialog box).

Note that you can click **Abort** at any time to stop the download process.

8.1.2.4.2 Program the PROMs

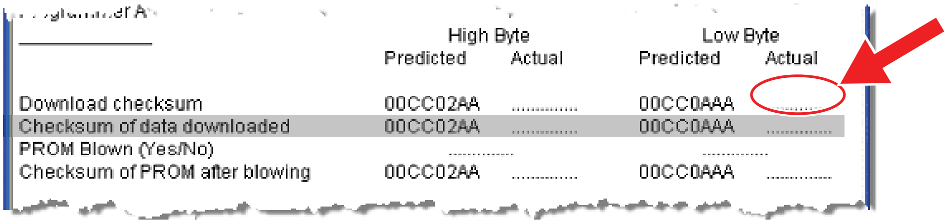
a) Download the Low Byte data to the programmer.

Refer to the manual for your specific PROM programmer.

b) Compare the checksum currently displayed by the programmer (this is the checksum of the data in the programmer’s RAM, after download from GCSS) with the Low Byte “**Download checksum**” shown on the PROM Download Report (printed in step **ii** on page 8-33)—they must be identical.

If the checksums differ, either discard the PROM, or erase and re-program it.

c) Write the value in the Low Byte "Download checksum" field on the report.



	High Byte Predicted	High Byte Actual	Low Byte Predicted	Low Byte Actual
Download checksum	00CC02AA	....	00CC0AAA	....
Checksum of data downloaded	00CC02AA	....	00CC0AAA	....
PROM Blown (Yes/No)	....		....	
Checksum of PROM after blowing	00CC02AA	....	00CC0AAA	....

d) Burn the Low Byte PROM.

Refer to the manual for your specific PROM programmer.

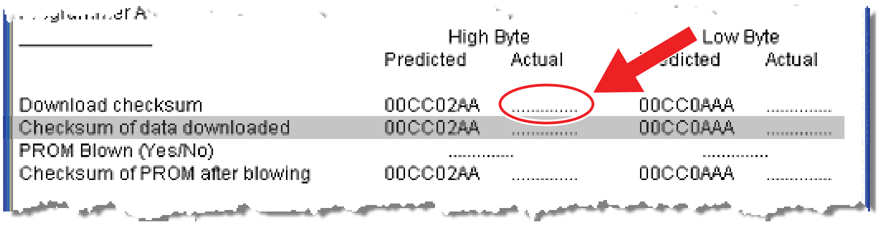
e) Download the High Byte data to the programmer.

Refer to the manual for your specific PROM programmer.

f) Compare the checksum currently displayed by the programmer (this is the checksum of the data in the programmer's RAM, after download from GCSS) with the High Byte "**Download checksum**" shown on the PROM Download Report—they must be identical.

If the checksums differ, either discard the PROM, or erase and re-program it.

g) Write the value in the High Byte "Download checksum" field on the report.

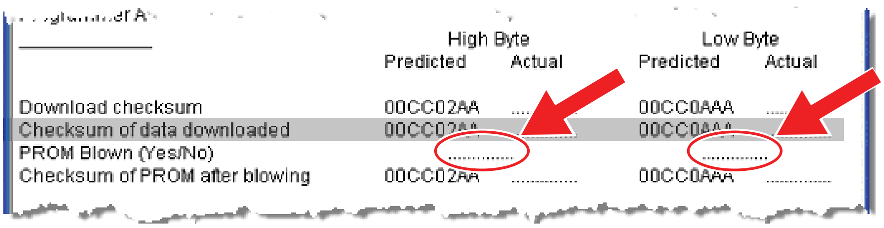


	High Byte Predicted	High Byte Actual	Low Byte Predicted	Low Byte Actual
Download checksum	00CC02AA	....	00CC0AAA	....
Checksum of data downloaded	00CC02AA	....	00CC0AAA	....
PROM Blown (Yes/No)	....		....	
Checksum of PROM after blowing	00CC02AA	....	00CC0AAA	....

h) Burn the High Byte PROM.

Refer to the manual for your specific PROM programmer.

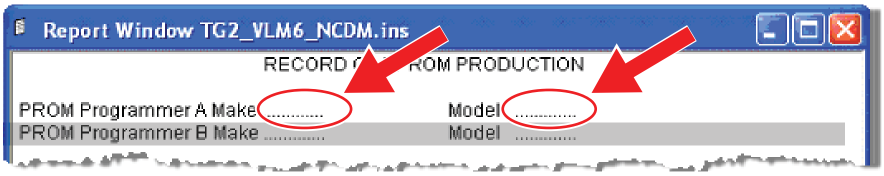
i) Fill in the "PROM Blown (Yes/No)" fields on the PROM Download Report.



	High Byte Predicted	High Byte Actual	Low Byte Predicted	Low Byte Actual
Download checksum	00CC02AA	....	00CC0AAA	....
Checksum of data downloaded	00CC02AA	....	00CC0AAA	....
PROM Blown (Yes/No)	....		....	
Checksum of PROM after blowing	00CC02AA	....	00CC0AAA	....

j) Click **Exit** in the Generate PROM Data dialog box to close it.

k) Complete the programmer "Make" and "Model" fields on the printed PROM Download Report.



8.1.2.4.3 Verify the PROMs

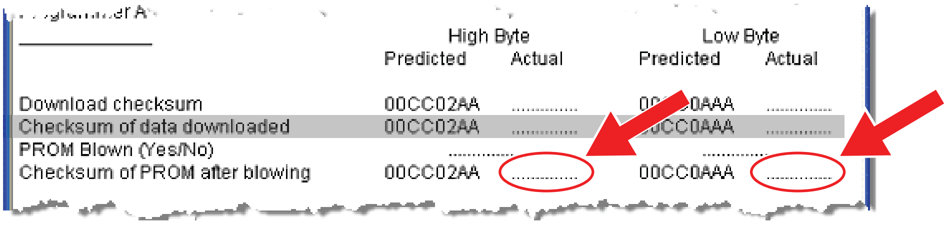
a) Insert one of the burnt PROMs into the programmer, load the PROM contents back into the programmer's RAM and read the checksum.

Refer to the manual for your specific PROM programmer.

b) Compare the checksum displayed by the Dataman with the appropriate "**Checksum of PROM after blowing**" shown on the PROM Download Report—they must be identical.

If the checksums differ, either discard the PROM, or erase and re-program it.

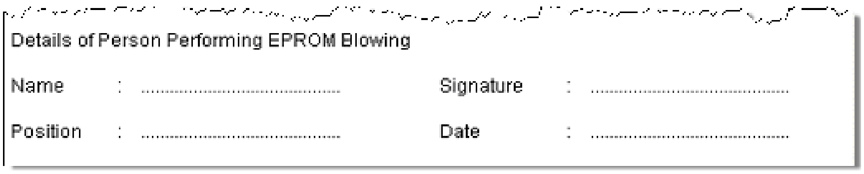
c) Write the value in the appropriate "Checksum of PROM after blowing" field on the report.



d) Insert the other burnt PROM into the Dataman and repeat the procedure.

8.1.2.4.4 Final Details

a) Complete the "Details…" fields at the bottom of the report.



b) File the PROM Download Report with the project documentation.

The PROMs have been successfully programmed and verified, and are ready to be installed in the WESTRACE VLM.

c) Click **Exit** in the Generate PROM Data dialog box (figure 8.19).

8.1.2.5 BeeProg Reference Procedures

The earlier BeeProg examples (sections 8.1.2.1 and 8.1.2.2) refer to the procedures here.

Contents of this section:

8.1.2.5.1 Installing BeeProg Software

a) If installing from an ELNEC installation CD:

   i. Insert the CD into the computer's CD drive and wait for the Setup.exe program to start automatically.

   ii. Select **Software Installation - Programmers**.

b) If installing the latest software from the ELNEC website:

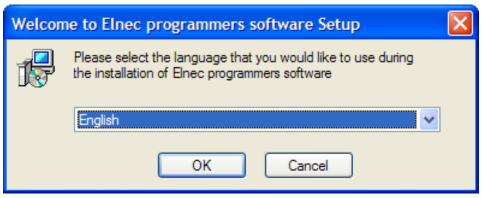
   i. Ensure that the computer's firewall software is enabled, then connect to the internet and download the file **PG4UWarc.EXE** from www.elnec.com/download/#prog.
   We recommend that you use appropriate software to scan the downloaded file for viruses.

   ii. If using a USB cable to connect the BeeProg to the computer, unplug the USB cable from the BeeProg.

   NOTE: The USB cable must not be connected during installation.

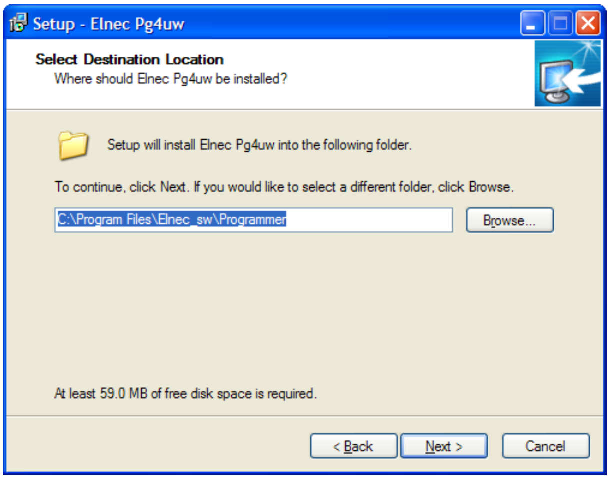
   iii. Run PG4UWarc.EXE.

c) Select **English** then click **OK**.

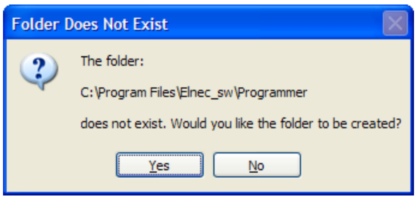


d) In the first wizard screen, click **Next**.

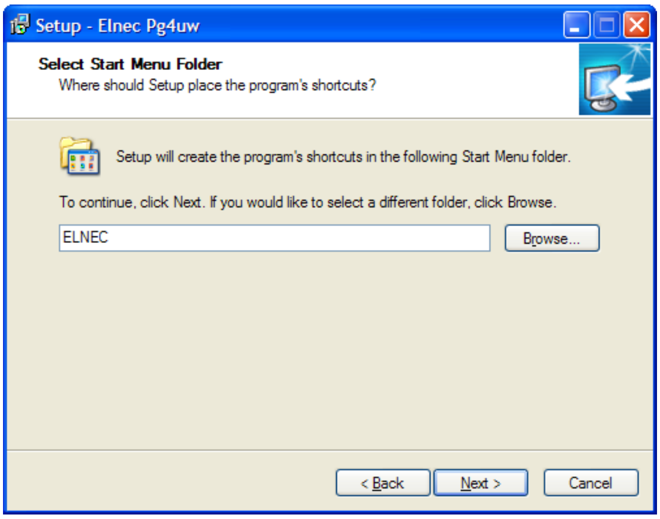
e) In the Select Destination Location screen, click **Next** to accept the default destination (C:\Program Files\Elnec_sw\Programmer).



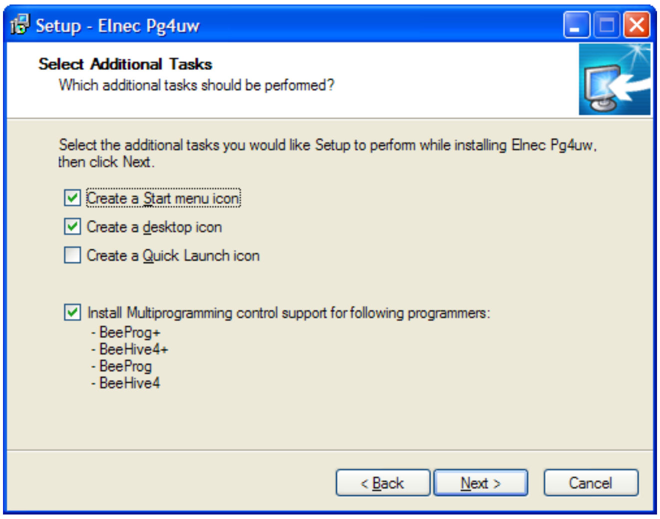
f) In the Folder Does Not Exist dialog box, Click **Yes**.



g) In the Select Start Menu Folder screen, Click **Next**.

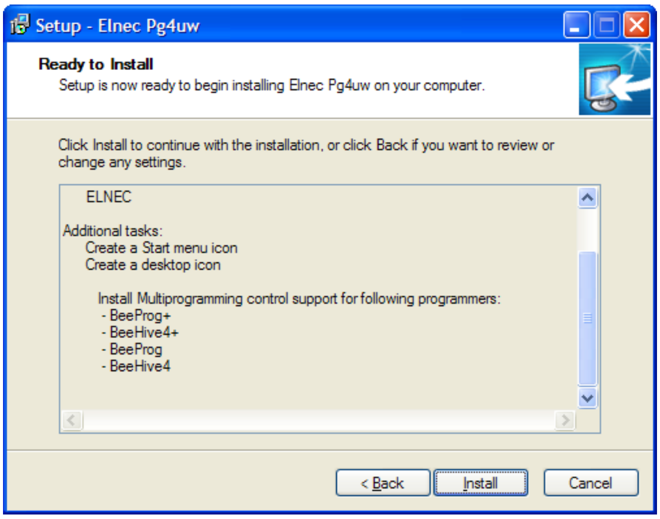


h) In the Select Additional Tasks screen, Click **Next**.

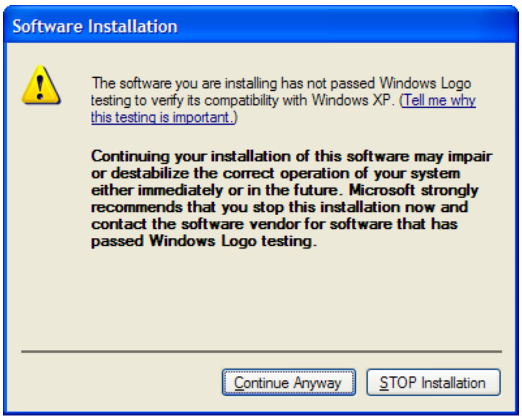


i) In the Ready to Install screen, click **Install**.

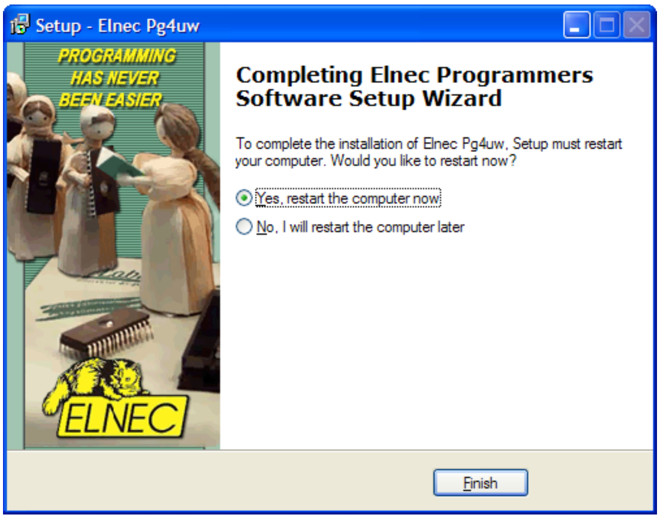
The installation begins.



j) If the Windows Logo Testing dialog box opens, click **Continue Anyway**.



k) In the final screen, ensure that **Yes** is selected then click **Finish** to restart the computer.



8.1.2.5.2 Setting Up the BeeProg

a) Connect the programmer to the computer using a standard USB cable (computer must support USB 2.0), or a standard parallel cable if no USB port exists.

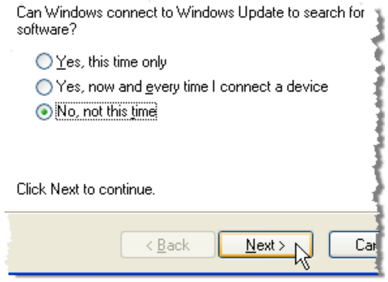
See section 10.1.4, “Cables”.

NOTE: If using a parallel BeeProg communications cable, you must connect this cable **before** powering the BeeProg. (Order is not important if using a USB communications cable.)
When disconnecting the BeeProg, unplug the power cable first, then the parallel communications cable. (Order is not important if using a USB communications cable.)

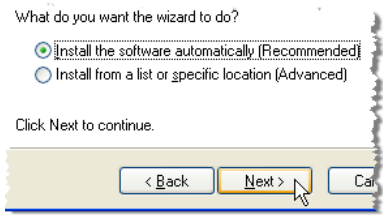
b) Power the BeeProg programmer and verify that its **POWER** LED is lit. (The unit has no power switch.)

c) If the Found New Hardware Wizard opens (occurs the first time the BeeProg is connected to the computer):

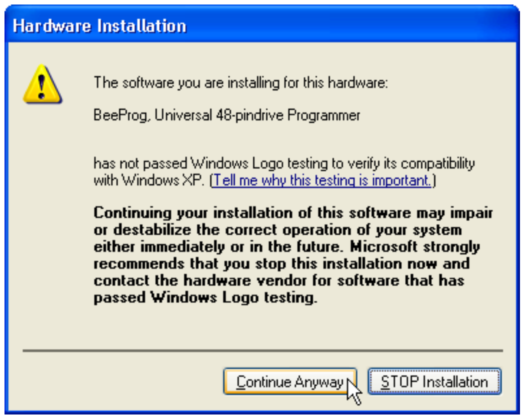
i. Select **No, not this time** and click **Next**.



ii. Select **Install the software automatically** and click **Next**.

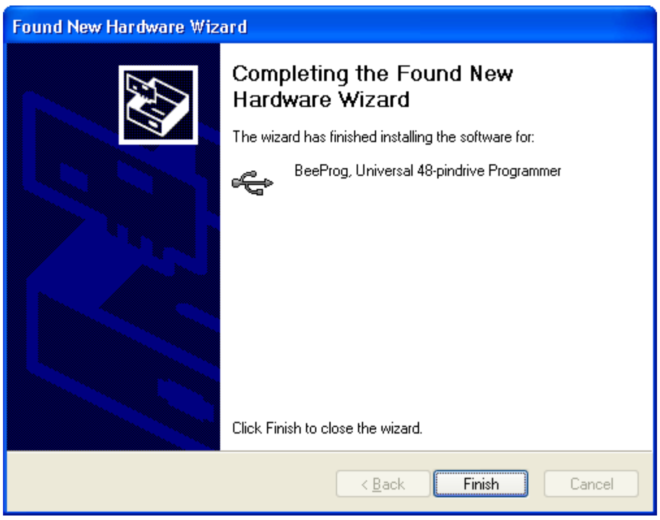


iii. If the Windows Logo Testing dialog box opens, click **Continue Anyway**.



Windows creates a System Restore point, then installs the BeeProg driver files.

iv. Click **Finish**.



8.1.2.5.3 Starting PG4UW

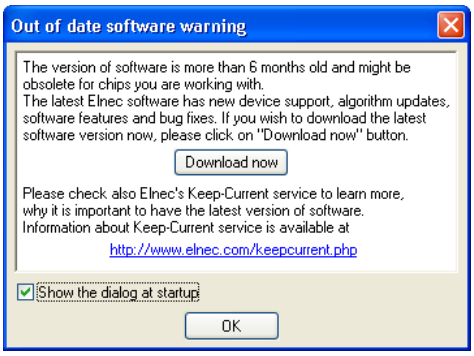
a) Start the BeeProg Control Program **PG4UW** (not PG4UWMC).

You can do this by:

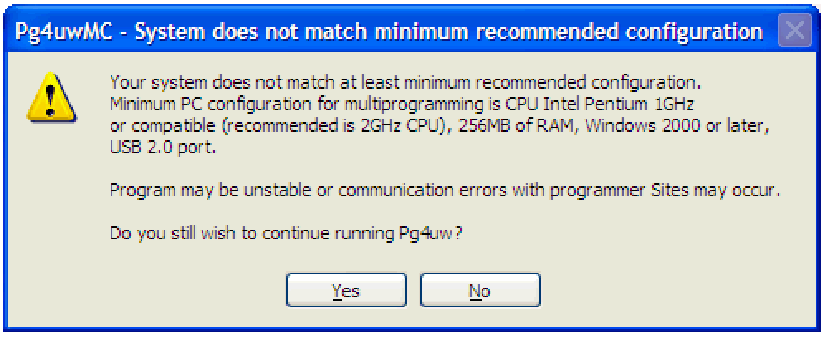
- double-clicking the icon on the desktop, or;
- choosing **Start** menu > **Programs** > **Elnec** > **PG4UW**.

b) If an "Out of Date Software" message appears, either:

- click **Download Now** to obtain the latest BeeProg software, then quit PG4UW, install the new version (see step **b** on page 8-36) and start again from step **a** immediately above, or;
- click **OK** to continue without updating PG4UW.

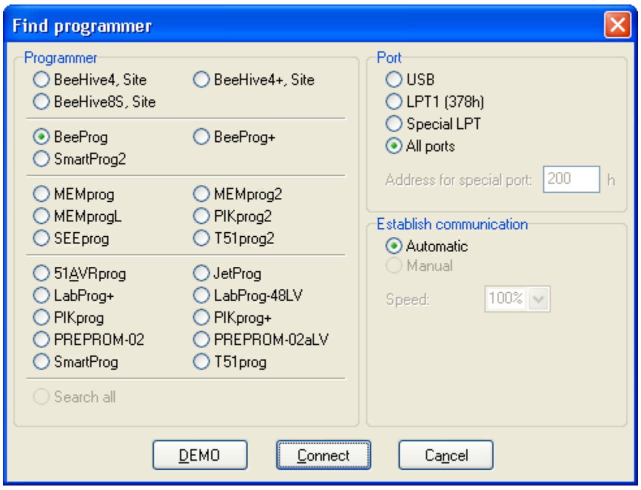


c) If a "System does not match ..." message appears, click **Yes**.



The PG4UW main screen opens, and the Log Window (in the centre, figure 8.13) shows basic computer information.

d) In the Find Programmer dialog box that opens:



i. Ensure that your programmer model (**BeeProg**, BeeProg+ or SmartProg2), **All ports** and **Automatic** are selected.

ii. Click **Connect**.
PG4UW connects to the BeeProg. The Log Window shows connection progress, communication speed and serial number of the connected BeeProg (figure 8.22).

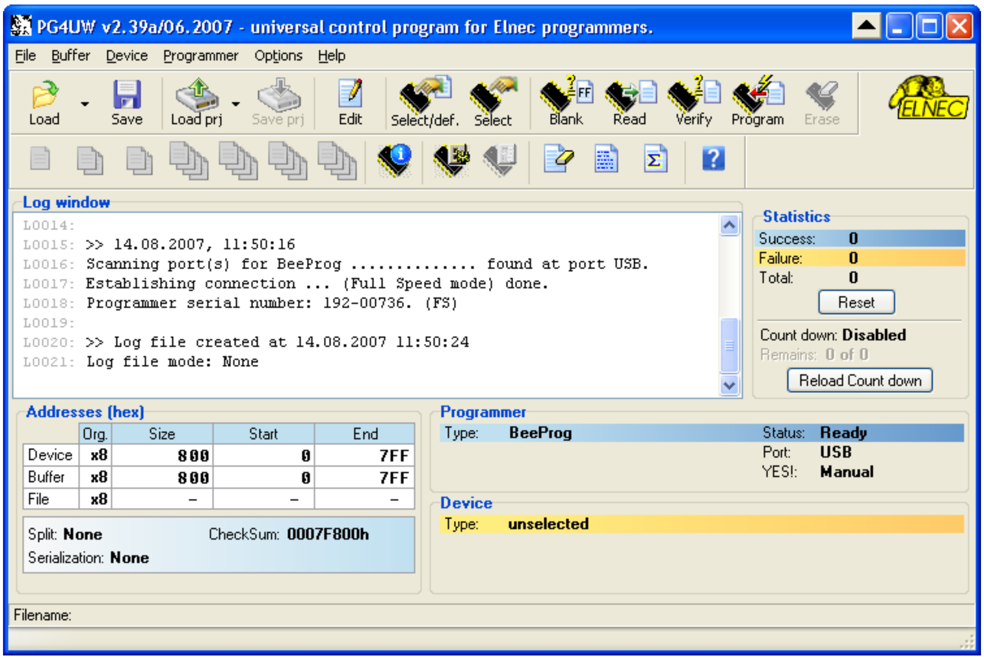
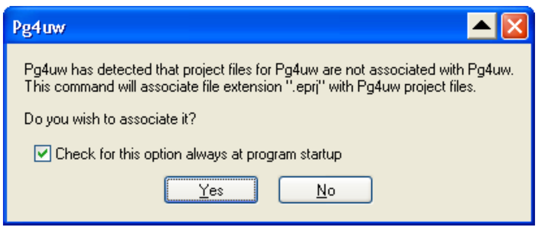


Figure 8.22 BeeProg PG4UW main screen—connected to BeeProg

e) If a PG4UW file association dialog box opens:



i. Ensure that **Check for this option...** is ticked.

ii. Click **Yes**.

The BeeProg and its software are now ready.

8.1.2.5.4 Erasing PROMs (BeeProg)

Always erase every PROM (new and used) prior to programming it.

Caution: ***WESTRACE PROMs can be damaged by static electricity. Take appropriate anti-static precautions when handling these devices.***

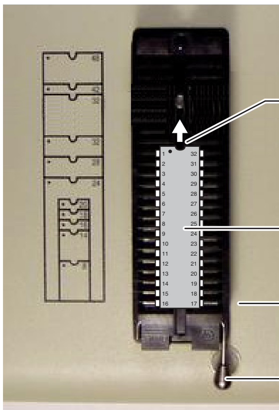
a) Insert one of the PROMs into the BeeProg:

i. Check that the BeeProg's **BUSY** LED is not lit (figure 8.12).

ii. Raise the ZIF[3] socket clamp lever.

iii. Place the PROM into the lower (lever) end of the socket, notch towards the top.

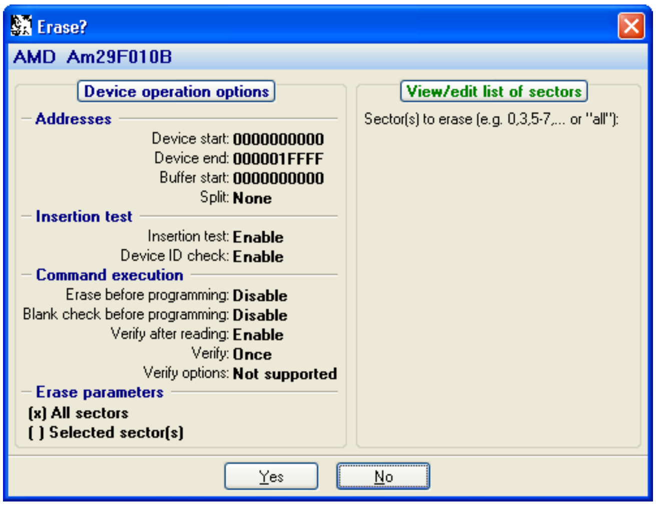
iv. Lower the clamp lever.



b) Click the **Device Erase** button (or press **F10**).

NOTE: Do not confuse this with the Buffer Erase button (Ctrl+F2).

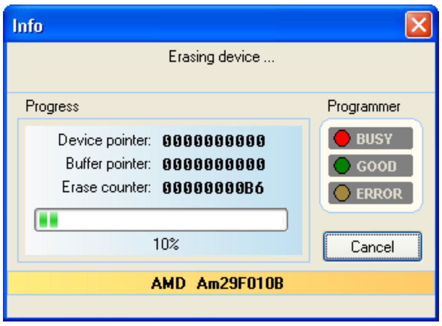
The Erase? dialog box opens.



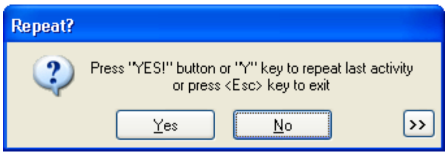
[3] ZIF: Zero Insertion Force

c) Click **Yes**.

An Info dialog box reports the progress.

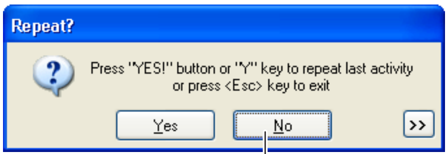


The Repeat? dialog box opens at the end of the process.



d) If the Info dialog box reports successful erasure (green header bar), click **No** in the Repeat? dialog box.

If the Info dialog box reports an error, repeat the process. Then if the Info dialog box still reports an error, repeat the process with another PROM.



Click **NO** to continue

e) Remove the PROM.

NOTE: Only remove PROMs when the BeeProg's **BUSY** LED is not lit.

f) Repeat steps **a** to **e** for the other PROM.

The PROMs have now been erased and are ready for programming.

8.1.3 Download and Upload Timeout (Serial Connection Only)

GCSS limits the serial PROM Download and Upload functions to 4 minutes. If either operation fails to finish within that time, the GCSS assumes an error condition and abandons the operation with an appropriate message.

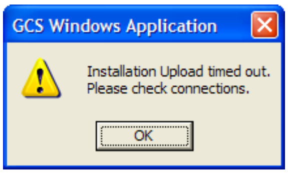


Figure 8.23 “Upload timed out” dialog box

Similarly, if the PROM Programmer does not respond within 2 seconds, the GCSS assumes an error condition and displays an appropriate message.

Upload and Download Timeout Detection

		Dataman S4	Other Serial Programmer
Cable	Link from DTR to DCD	Break in link triggers timeout	–
	No DTR/DCD link	Bad line detection triggers timeout	Bad line detection triggers timeout

Table 8.10 PROM programmer serial timeout detection

8.2 NVLM Installations: Make Non-Vital Configuration and Download

Relates to design cycle:
- task E on page 1-9
- task 40 on page 1-13

This section explains how to compile a Non-Vital Configuration and download it to the NVLM (NCDM or NVC/DM module).

a) Ensure that the installation's Non-Vital Source File has been approved (section 1.10).

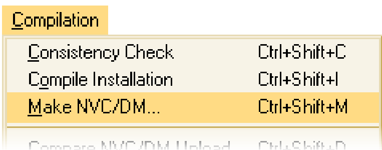
b) Install the NVLM in a WESTRACE housing, connect a suitable serial cable (section 10.1.4.3) between the serial port on the computer and the Production Port on the front of the NVLM (see [APPM]), and power the WESTRACE.

c) Ensure the NVLM module is in Maintenance Mode.

You do this manually, either by:
- connecting a special "maintenance cable" to the Production Port and restarting the NVLM (section 10.1.4.3), or;
- setting on-board switches and restarting the NVLM (section 10.2).

d) In GCSS:

i. Choose **Compilation** > **Make NVC/DM** or **Compilation** > **Make NCDM**).



Alt c m
Ctrl+Shift+m

ii. If the "Recompile now?" dialog box opens, click **OK**.

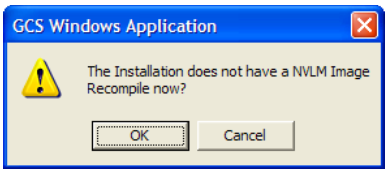


Figure 8.24 "Recompile now?" dialog box

GCSS generates the Non-Vital Configuration data and opens a report window that lists the (re)compilation details.

iii. Wait for the "Download Data" dialog box to open.

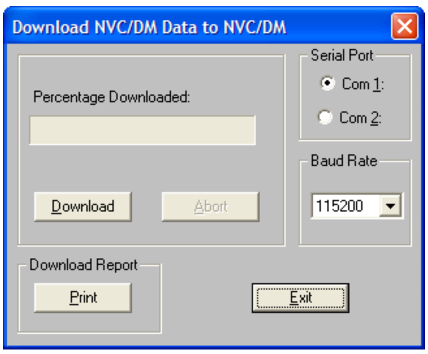


Figure 8.25 "Download Data" dialog box (NVC/DM shown)

e) In the “Download Data” dialog box (figure 8.25):

i. Select the computer’s **Serial Port** that is connected to the NVLM, and the **Baud Rate** (use 115200 unless there is a reason not to).

ii. Click **Print** to print a Non-Vital Configuration Download Report (figure B.8).

The Print dialog box allows you to change paper orientation or select another printer if necessary. If you want to view the report before printing, cancel the Print dialog box and choose **File** > **Print Preview**.

iii. Click **Download**.

The GCSS displays the percentage completed as the image is downloaded.

Note that you can click **Abort** at any time to stop the download process.

iv. When the download is complete, click **Exit** in the dialog box to close it.

f) Complete the “Details…” fields on the Non-Vital Configuration Download Report.

g) File the report with the project documentation.

h) Switch the NCDM or NVC/DM out of Maintenance Mode (section 10.2).

8.3 Uploading, Decompiling and Comparing Installations (Optional Verification Method)

This section describes an additional, optional verification procedure. It is not a required part of the approval process, but may reduce the risk of failures at the ICS stage (ie, it provides additional surety before the PROMs are taken to site).

Contents of this section:

- "VLM Installation Upload and Decompile (Optional VLM Verification: Part 1)"—section 8.3.1—various examples for different programmers
- "NVLM Installation Upload and Decompile (Optional NVLM Verification: Part 1)"—section 8.3.2
- "Comparing Uploaded Installations (Optional VLM and NVLM Verification: Part 2)"—section 8.3.3

To verify a VLM installation, see section 8.3.1 then 8.3.3.
To verify an NVLM installation, see section 8.3.2 then 8.3.3.

8.3.1 VLM Installation Upload and Decompile (Optional VLM Verification: Part 1)

This procedure is optional. (The usual method is to upload from the installed WESTRACE equipment using ICS; see figure 2.44 on page 2-82.)

Relates to design cycle:
- task F on page 1-9
- tasks 41 and 42 on page 1-13

This section explains how to use the GCSS's **Upload from PROM** function to upload data from a set of PROMs and decompile it to create a reproduction of the original VLM installation.

Then, as section 8.3.3 describes, the decompiled installation can be:

- compared with the original VLM installation for the WESTRACE application, perhaps to verify the PROMs before shipment;
- compared with an original or uploaded VLM installation from another similar WESTRACE application;
- used as a source for the creation or modification of another VLM installation.

Note: ***A PROM does not store all installation data so certain data will be missing from an uploaded installation.***

Examples of various upload methods are given:

- Section 8.3.1.1 describes the Intel Hex files method, using the BeeProg (connected by USB) as the example PROM programmer.
- Section 8.3.1.2 specifically describes the use of the Dataman S4 serial programmer to upload the data.
- Section 8.3.1.3 describes the use of GCSS's "Other" serial programmer functionality to upload the data.

8.3.1.1 Upload and Decompile: Intel Hex Files—BeeProg USB Example

This procedure is optional.

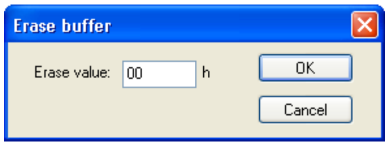
a) Ensure that the BeeProg software PG4UW is installed on the computer, and that the BeeProg is connected to a USB port on the computer (see sections 8.1.2.5.1 and 8.1.2.5.2).

b) Erase the BeeProg's buffer memory:

i. Click the **Buffer Erase** button (or press **Ctrl+F2**).

NOTE: Do not confuse this with the Device Erase button .
The Erase Buffer dialog box opens.

ii. Type **00** (zero zero) and click **OK**.
The PG4UW Log Window shows "Buffer is erased with value 00h."



c) Read the contents of the High Byte PROM into the BeeProg, then upload it to the computer as an Intel Hex file:

i. Insert the High Byte PROM into the BeeProg.

ii. In PG4UW, click the **Read** button (or press **F7**)
The contents of the PROM are read into the BeeProg buffer.

iii. Choose **File** > **Save**.

iv. In the Save Buffer dialog box that opens (figure 8.26), use the Save In field to set your destination directory, type a filename and be sure to add the suffix **HIExPROM**, choose **IntelHEX** as the type, then click **Save**.

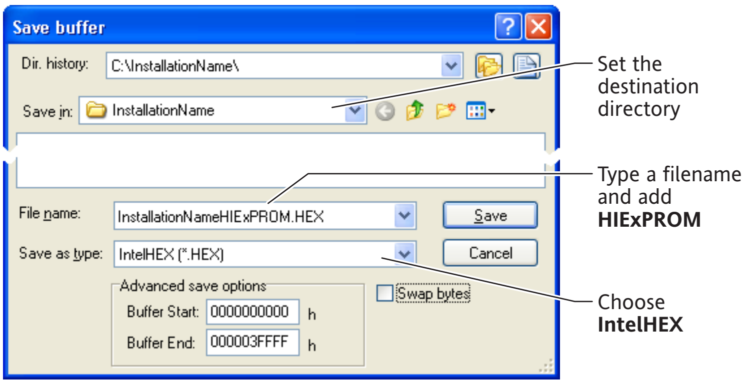
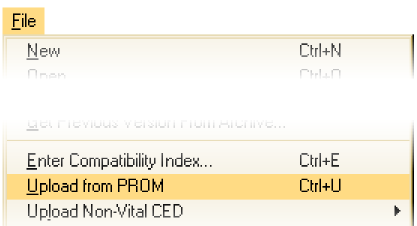


Figure 8.26 Save Buffer dialog box

d) Repeat step **c** using the Low Byte PROM instead, and adding **LOExPROM** to the filename.

e) Now use GCSS to decompile the hex files and re-create the installation:

i. In GCSS, choose **File** > **Upload from PROM.**



Alt f u
Ctrl+u

The Upload PROM Data dialog box opens.

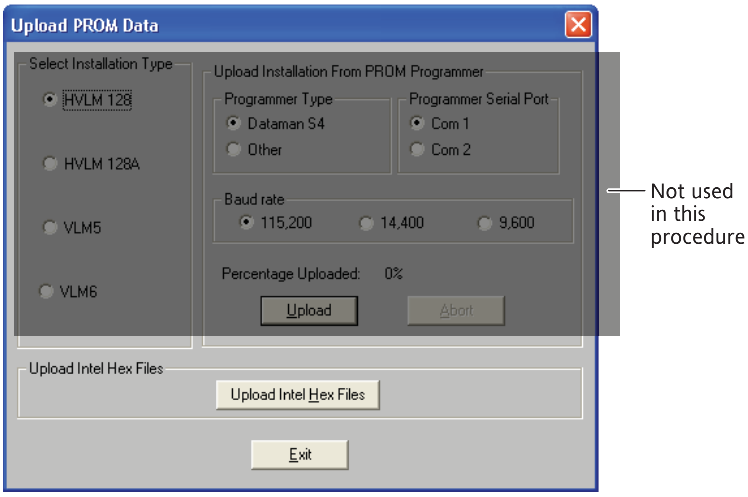


Figure 8.27 Upload PROM Data dialog box

ii. Click **Upload Intel Hex Files**.

iii. In the Select High PROM File dialog box that opens (figure 8.28), locate and select the High PROM Intel Hex file you just uploaded, then click **Open**.

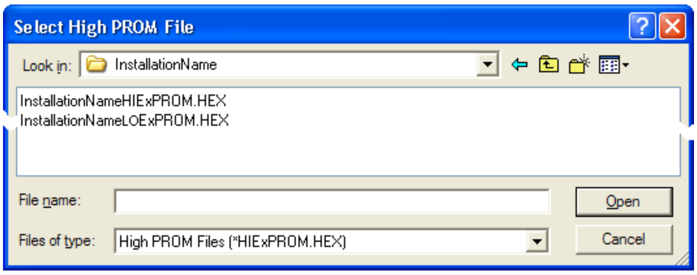


Figure 8.28 Select High PROM File dialog box

iv. In the Select Low PROM File dialog box that opens (figure 8.29), locate and select the Low PROM Intel Hex file you just uploaded, then click **Open**.

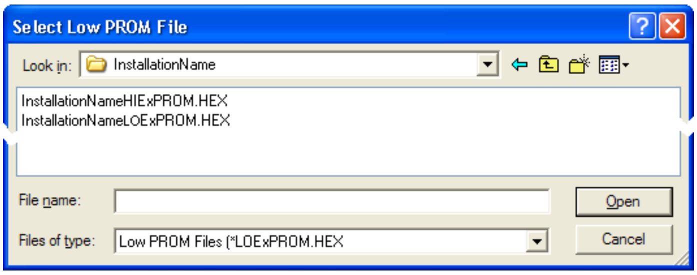


Figure 8.29 Select Low PROM File dialog box

v. A PROM Upload Report opens (see section 8.3.1.4 and figure B.7) that displays decompilation progress, including any decompilation errors (see Appendix D, "Decompiler Error Messages").

vi. Check that "Installation decompiled successfully" appears at the end of the PROM Upload Report.

An unsuccessful decompilation probably indicates that one or both of the PROMs is faulty—ie invalid. If this is the case, discontinue this procedure.

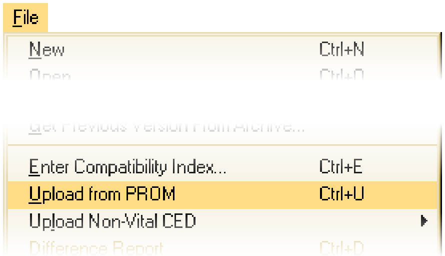
f) Finally, carry out the steps in section 8.3.3 to verify the PROMs by comparing the installation just created by the upload and decompile process with the original VLM installation for the WESTRACE application.

8.3.1.2 Upload and Decompile: Dataman S4

a) Connect the Dataman to a serial port on the computer (figure 10.1 gives cable details), then power the Dataman.

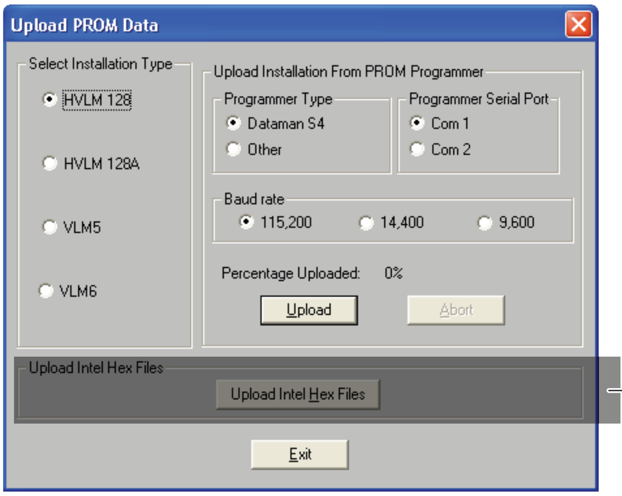
b) Configure the Dataman (see step **c** on page 8-24).

c) In GCSS, choose **File** > **Upload from PROM.**



Alt f u
Ctrl+u

The Upload PROM Data dialog box opens.



Not used in this procedure

Figure 8.30 Upload PROM Data dialog box

d) In the Upload PROM Data dialog box:

   i. Select the Installation Type, **Dataman S4**, the computer's **Serial Port** to which the Dataman is connected, and the **Baud Rate** to which you configured the Dataman (step **c** on page 8-24).

   ii. Click **Upload**.
   Note that you can click **Abort** at any time to stop the upload process.

e) When prompted, upload the contents of the Low Byte PROM to the new GCSS installation file via the Dataman as follows:

INSTRUCTION	DISPLAY	COMMENT
Insert the Low Byte PROM into the Dataman.		
Press **LOAD.**		
	> Load 28F010 **00000—IFFFF = 00000**	

Table 8.11 Uploading Low Byte PROM contents (Dataman) [sheet 1 of 2]

INSTRUCTION	DISPLAY	COMMENT
Press **Enter.**		
		Dataman sounds several beeps while image loads into RAM.
	00000—IFFFF = 00000 **>**	
Press **SUM.**		Grey key—to get check sum.
	> CHECKSUM RAM **00000,0FFFF**	
Press **Enter.**		Dataman sounds five beeps for HVLM128 or HVLM128A, ten beeps for VLM5 or VLM6.
	SUM = 005A5DAA **>**	Confirm that this value matches the appropriate "Checksum of PROM after blowing" printed on the PROM Download Report (figure B.6).
		If the values do not match, the PROM is probably faulty—ie invalid. If this is the case, discontinue this procedure.
Click **OK** on PC.		

Table 8.11 Uploading Low Byte PROM contents (Dataman) [sheet 2 of 2]

f) When prompted, upload the contents of the High Byte PROM to the new GCSS installation file via the Dataman by repeating the steps in table 8.11, but using the High Byte PROM instead.

GCSS uploads the data from the Dataman and automatically decompiles it into a new VLM installation file.

A PROM Upload Report opens (see section 8.3.1.4 and figure B.7) that displays decompilation progress, including any decompilation errors (see Appendix D, "Decompiler Error Messages").

g) Check that "Installation decompiled successfully" appears at the end of the PROM Upload Report.

An unsuccessful decompilation probably indicates one or both of the PROMs is faulty—ie invalid. If this is the case, discontinue this procedure.

h) Finally, carry out the steps in section 8.3.3 to verify the PROMs by comparing the installation just created by the upload and decompile process with the original VLM installation for the WESTRACE application.

8.3.1.3 Upload and Decompile: “Other” Serial Programmers

For uploading data from PROMs, GCSS supports certain serial programmers in addition to the Dataman S4 (discussed in section 8.3.1.2). Like the Dataman, these programmers must be able to accept data in binary format at 115,200, 14,400 or 9600 Baud.

Use of these “other” serial programmers is covered here. The general procedure is similar to the Dataman procedure (section 8.3.1.2) except that you select **Other** in the Upload PROM Data dialog box (figure 8.31) and use your particular programmer’s interface.

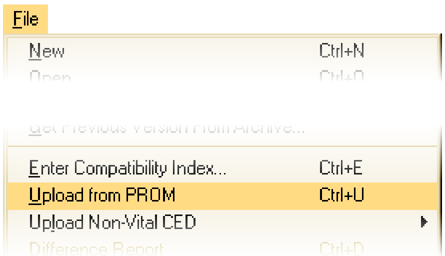
See also:

- section 8.1.2.3, “Procedure 3: Dataman S4” as a general guide;
- your serial programmer’s manual for detailed information.

a) Connect the “other” programmer to a serial port on the computer (figure 10.1 shows a typical cable), then power the programmer.

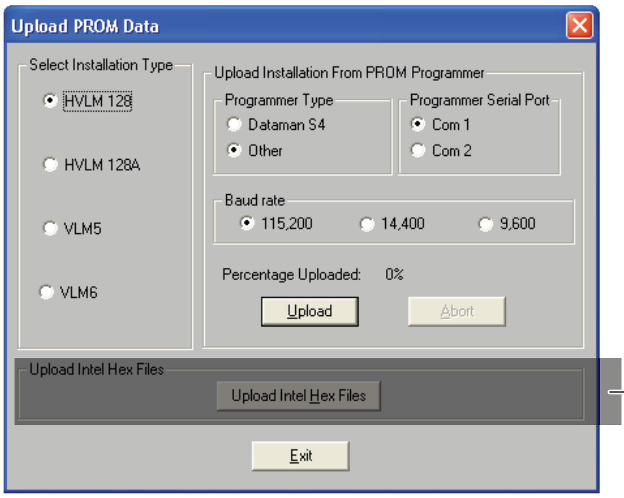
b) Configure the programmer for communication with GCSS.
Refer to the manual for your specific programmer.

c) In GCSS, choose **File** > **Upload from PROM.**



Alt f u
Ctrl+u

The Upload PROM Data dialog box opens.



Not used in this procedure

Figure 8.31 Upload PROM Data dialog box

d) In the Upload PROM Data dialog box:

   i. Select the Installation Type, **Other**, the computer’s **Serial Port** to which the programmer is connected, and the **Baud Rate** to which you configured the programmer (step **b** on page 8-54).

ii. Click **Upload**.

Note that you can click **Abort** at any time to stop the upload process.

e) When prompted, use your programmer to upload the contents of the Low Byte and High Byte PROMs to a new GCSS installation file.

Use section 8.1.2.3 as a guide, and refer to the manual for your specific programmer if necessary.

f) Finally, carry out the steps in section 8.3.3 to verify the PROMs by comparing the installation file just created by the upload and decompile process with the original VLM installation for the WESTRACE application.

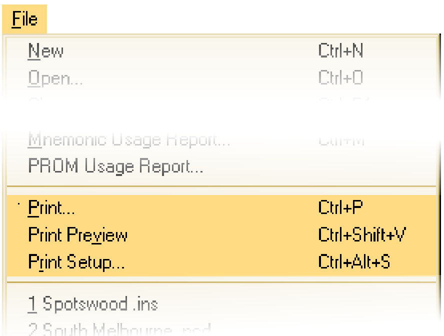
8.3.1.4 PROM Upload Report

The PROM Upload Report (figure B.7) is automatically generated at the end of the upload and decompile procedure.

When the Report Window opens:

a) Choose **File** > **Print Setup** and change the page orientation or select another printer if necessary (see section B.2).

b) Choose **File** > **Print** or **Print Preview**.



Alt f r
Ctrl+Alt+s

Alt f p
Ctrl+p

Alt f v
Ctrl+Shift+v

8.3.2 NVLM Installation Upload and Decompile (Optional NVLM Verification: Part 1)

This procedure is optional. (The usual method is to upload from the installed WESTRACE equipment using ICS; see figure 2.44 on page 2-82.)

Data can be uploaded from an NVLM and decompiled to create a reproduction of the original NVLM installation in a manner similar to a Non-Vital Configuration download. Then the uploaded installation can be:

- used as a source for the creation or modification of another NVLM installation;
- compared with the original NVLM installation for the WESTRACE application, perhaps to ensure the NVLM module is OK before shipment;
- compared with an original or uploaded NVLM installation from another similar WESTRACE application.

Use the PROM Data Comparison report (see section 8.3.3 and section B.5.16) to compare an uploaded and decompiled NVLM installation with another installation.

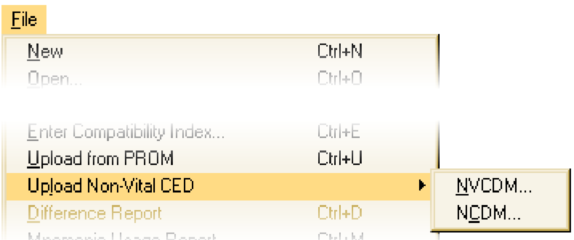
Uploading

a) Connect a suitable serial cable (see section 10.1.4.3) between the computer's serial port and the Production Port on the front of the NVLM.

The NVLM should be installed in a WESTRACE housing.

b) Ensure the NVLM is in Maintenance Mode (see section 10.2).

c) Choose **Upload Non-Vital CED** and then the type of NVLM.



Alt f l n
Alt f l c

The Upload Non-Vital Installation dialog box opens.

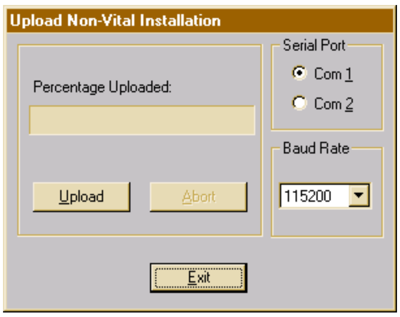


Figure 8.32 Upload Non-Vital Installation dialog box

d) Select the serial port that is connected to the NVLM, and the Baud rate. (Use 115200 unless there is a reason not to.)

e) Click the **Upload** button.

Upload progress is monitored as a percentage of the data to be uploaded.

f) Ensure the NVLM is switched out of Maintenance Mode after the upload when using a standard cable (see section 10.2).

The upload can be abandoned at any stage by clicking **Abort**.

Decompilation

The GCSS automatically decompiles the Non-Vital Configuration into an NVLM installation file after uploading. Decompilation progress, including any decompilation errors, is displayed in a Report Window. See Appendix D for an explanation of possible decompilation errors.

Note: ***Only Ladder Logic and VLM – NVLM Interface File data is decompiled from uploaded Non-Vital Configurations.***

Non-Vital Configuration Upload Report

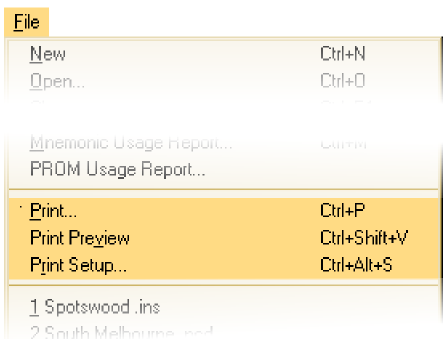
This report is automatically generated during decompilation. It should be signed and stored with the rest of the project documentation.

With the report window open:

a) Choose **Print Setup** and change the page orientation or select another printer if necessary.

See Appendix B, section B.2 for further assistance.

b) Choose **Print** or **Print Preview** to generate a report of the decompilation process.



Alt f r
Ctrl+Alt+s

Alt f p
Ctrl+p

Alt f v
Ctrl+Shift+v

See section B.5.8 on page B-13 for an example.

c) Check that "Installation decompiled successfully" appears at the end of the report.

An unsuccessful decompilation probably indicates that the Non-Vital Configuration is faulty. If this is the case, discontinue this procedure.

d) Finally, carry out the steps in section 8.3.3 to compare the decompiled Non-Vital Configuration with the original approved source data.

8.3.3 Comparing Uploaded Installations (Optional VLM and NVLM Verification: Part 2)

This procedure is a continuation of the optional upload and decompile procedures in sections 8.3.1 and 8.3.2.

After uploading and decompiling the VLM data from the PROMs (section 8.3.1) or the NVLM data from the NCDM or NVC/DM (section 8.3.2), you use the GCSS **Compare** function and the PROM Data Comparison report (section B.5.16) as described below to verify the retrieved data.

Note: ***This verification procedure does not remove the need to use ICS to extract and compare data from the final, installed WESTRACE system. (Later, in ICS, you must do a similar comparison, but using uploaded and decompiled data from the installed PROMs or downloaded NVLM.)***

The ICS procedure provides a final, all-inclusive check of the compile, download and store process through a diverse path right back to the approved source data.

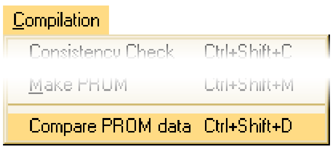
Note: ***Like must be compared with like. It is not possible to compare vital PROM data with Non-Vital Configuration data.***

a) Open the uploaded and decompiled VLM or NVLM installation (see section 8.3.1 or 8.3.2), and the relevant original approved installation.

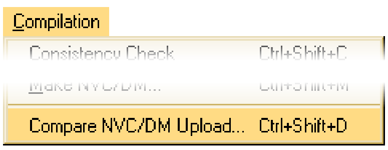
Other installations may also be open.

b) Click either of the two installations you want to compare.

c) Choose **Compilation** > **Compare PROM Data** or **Compilation** > **Compare NVC/DM Upload**.



Ctrl+Shift+d



The Comparison Report dialog box opens.

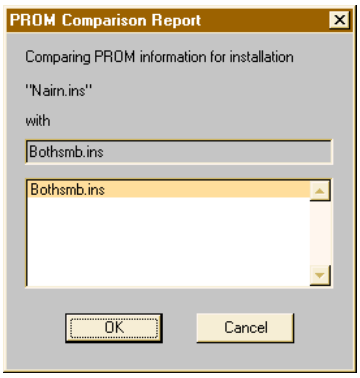


Figure 8.33 Comparison Report dialog box—VLM (PROM) example shown

d) From the list of open installations, select the other of the two installations you want to compare, then click **OK**.

GCSS generates the report and displays it in the Report Window. See section B.5.16, “Compare PROM Data or Compare NVC/DM Upload Report” for an example.

e) Check that the report contains no differences—there should be no difference markers (>>) at the left edge of the report, and the last line of the report should read “Installation Comparison Complete: No Differences Found”.

If no differences are reported, the VLM data burnt into the PROMs or the NVLM data downloaded to the NCDM or NVC/DM is valid.

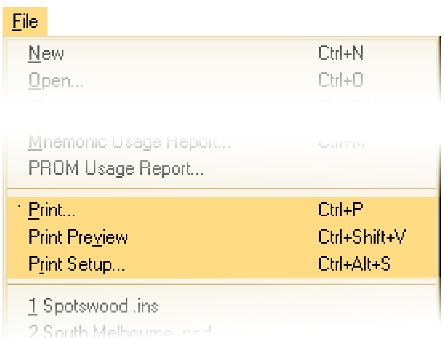
If differences are reported, reprogram the PROMs or download the Non-Vital Configuration again.

f) If you want to print or print-preview the report:

i. Click the Report Window to make it active.

ii. Choose **File** > **Print Setup** and change the page orientation or select another printer (see Section B.2, “Selecting and Setting Up a Printer”).

iii. Choose **File** > **Print** or **File** > **Print Preview**.



Alt f r
Ctrl+Alt+s

Alt f p
Ctrl+p

Alt f v
Ctrl+Shift+v

iv. Complete the Print or Print Preview process as required.

9. Find and Replace Mnemonics

The GCSS has the ability to find and replace mnemonics when the editors are active. There are three options:

- Find;
- Find and Replace;
- Find and Replace All.

All find and replace operations:

- are **case-sensitive**;
- are performed across a single installation only;
- retain the last search string (for quick re-use, the previous search string is "remembered" in the Find and the Find and Replace dialog boxes).

The replacement of any mnemonic causes the status of the installation to change to "Modified".

Table 9.1 defines the scope of the find and replace operations within the context of the GCSS editors.

Active Editor	Scope	
Housing Editor	All modules within the installation	Find: all rungs Replace: unlocked rungs only (mnemonics in locked rungs cannot be replaced)
Module Editor	Vital installation (*.ins)—current module **NOTE: GCSS does not find matches within vital communications ports.**	
	Non-vital installation (*.nvc and *.ncd)—current module—all ports, timers, latches and data fields	
Ladder Editor	Currently-selected rung or rungs (opens each rung in the Single Rung Viewer or Locked Rung Viewer) **NOTE: In the Ladder Viewer, GCSS only searches within rungs that are *selected* (see section 6.2.3.1).**	
Rung Editor	Current rung	
Module Ports Editor	Current port or housing	

Table 9.1 Find and Replace operations—scope

All found mnemonics are displayed in the editor appropriate for the scope of the search. See table 9.2.

Mnemonics found by a search initiated from these editors:	are displayed in these editors:
Ladder or Rung Editor	Rung Editor
Housing or Module Editor	Module Editor
Port Editor	Port Editor

Table 9.2 Search initiation and display—Find

Note: ***GCSS does not find matches within vital communications ports.***

9.1 Find

To find a mnemonic:

a) Open the Housing, Module, Ladder, Rung or Module Ports Editor.

b) If searching the Ladder Editor, select the rung or rungs to be searched—see section 6.2.3.1.

c) Choose **Find** from the **Edit** menu to display the Find dialog Box (figure 9.1).

Alt e f

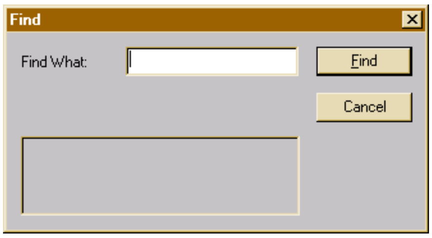


Figure 9.1 Find dialog box

d) Type the search text in the **Find What** field (case-sensitive).

Use wildcard characters to broaden the search—see section 9.4.

e) Click the **Find** button.

GCSS searches for the first occurrence of the search text, and opens an appropriate editor if a match is found (see table 9.2).

The found text is identified in the editor, and the details are shown in the Find dialog box.

f) Click the **Find** button again (or press **Enter**).

GCSS displays the next match.

g) Repeat until one of the following occurs:

- You locate the mnemonic you want.
  Press Cancel to close the Find dialog box.
- GCSS finds the last match (shows the total number of matches found—figure 9.2).
  Click **OK**. Both dialog boxes close.

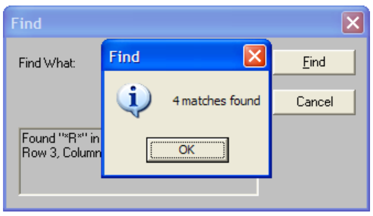


Figure 9.2 “Last match found” dialog box

9.2 Find, Replace and Replace All

Note: ***Reserved mnemonics cannot be replaced.***

Mnemonics in locked rungs cannot be replaced.

To find and replace some or all mnemonics:

a) Open the Housing, Module, Ladder, Rung or Module Ports Editor.

b) If searching the Ladder Editor, select the rung or rungs to be searched—see section 6.2.3.1.

c) Choose **Find and Replace** from the **Edit** menu to display the Find and Replace dialog Box (figure 9.3).

Alt e r

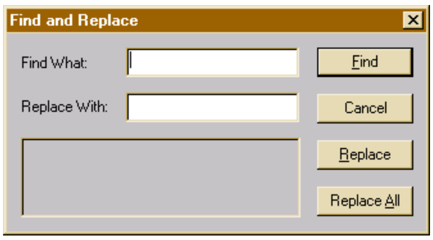


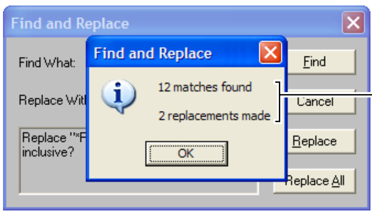
Figure 9.3 Find and Replace dialog box

d) Type the search text in the **Find What** field.
Use wildcard characters to broaden the search—see section 9.4.

e) Type the replacement text in the **Replace With** field.
Replacement text must not:
- contain wildcard characters (see section 9.4);
- exceed 15 characters (also see section 9.3.1).

f) To replace all matches with the replacement text (unlocked rungs only):

i. Click the **Replace All** button.
GCSS replaces the first and subsequent matches without further confirmation, then reports the total number of matches and replacements (figure 9.4).



Matches within locked rungs (10 in this example) are not replaced

Figure 9.4 “Total matches and replacements” dialog box

ii. Click **OK** to close both dialog boxes.

g) To find and replace matches one-by-one:

i. Click the **Find** button.

GCSS searches for the first occurrence of the search text, and opens an appropriate editor if a match is found (see table 9.2).

The found text is identified in the editor, and the details are shown in the Find and Replace dialog box.

ii. Click the **Replace** button.

GCSS carries out the replacement and finds the next occurrence of the search text.

iii. Click the **Replace** button repeatedly until one of the following occurs:

- The required matches are replaced.
  Click **Cancel** to stop the search.
- The last match is replaced (GCSS reports the total number of matches and replacements—figure 9.4).
  Click **OK**. Both dialog boxes close.

GCSS checks the consistency of each replacement. Consistency check failures are handled as described in section 9.3.2.

9.3 Error Conditions

9.3.1 Replace String is Too Long

The replacement string cannot exceed 15 characters.

The GCSS prevents the entry of a replacement string in excess of 15 characters by not recognising character 16 onwards.

9.3.2 Consistency Check Failure

Table shows the outcome of **Find and Replace** consistency check failures.

Find & Replace started from:	Outcome
Housing Editor	Further searching is abandoned and the failure is reported. Replacements made this point remain intact. The last replacement is left on display and selected when the error is acknowledged.
Ladder Editor	The operation continues, but the rung is marked as inconsistent (see section 6.3.8).

Table 9.3 Find and Replace consistency check failures

9.4 Wildcard Characters

Find or Find and Replace operations will only find the mnemonics that exactly match the search pattern. For example, if the search pattern is “ab” then occurrences of “abc”, “cab” and “acb” are ignored.

The scope of a search or replacement can be broadened by including the wildcard character “*” (asterisk) in the search text. The asterisk character represents zero or more characters of any value in the found mnemonic; and can be used at the beginning, end, or middle of a search pattern.

9.4.1 Wildcards in a Find Operation

Table 9.4 illustrates wildcard matches in a find operation, a tick signifies a match and a cross no match.

Search Pattern	Mnemonic									
	abc	**abc1e**	**d1abc**	**ad1bc**	**ad1be2c**	**deabcfg**	**deaf2bc**	**axybcz**	**raxbcyt**	**abcabc**	**cba**
abc	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
abc*	✓	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗
*abc	✓	✗	✓	✗	✗	✗	✗	✗	✗	✓	✗
a*bc	✓	✗	✗	✓	✗	✗	✗	✗	✗	✓	✗
a*b*c	✓	✗	✗	✓	✓	✗	✗	✗	✗	✓	✗
abc	✓	✓	✓	✗	✗	✓	✗	✗	✗	✓	✗
*a*bc	✓	✗	✓	✓	✗	✗	✓	✗	✗	✓	✗
a*bc*	✓	✓	✗	✓	✗	✗	✗	✓	✗	✓	✗
*a*bc*	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✗
a*	✓	✓	✗	✓	✓	✗	✗	✓	✗	✓	✗
*a	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
a	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 9.4 Wildcard matches in Find operations—examples

9.4.2 Use of Wildcards in a Replace Operation

Wildcards cannot be used in the *Replace With* field.

When a wildcard is used:

- at the beginning or end of a search pattern — only the fixed text is replaced by the replacement text.

  For example, if the search pattern is "*abc*" and the replacement text is "efgh" then "xyzabc1" is replaced with "xyzefgh1".

- within the middle of a search pattern — the whole text (excepting wildcards at the beginning or end) is replaced by the replacement text.

  For example, if the search pattern is "*ab*c*" and the replacement text is "efgh" then "xyzabqwc1" is replaced with "xyzefgh1".

Only the first occurrence of a pattern match within a mnemonic is replaced. So, the search pattern "*a*" and replacement text of "d" in the mnemonic "abcabc" will result in the substitution "dbcabc".

Table 9.5 illustrates wildcard pattern replacements using the replacement text 'BOB'. A cross ✗ represents no match.

Search Pattern	Mnemonic										
	abc	**abc1e**	**d1abc**	**ad1bc**	**ad1be2c**	**deabcfg**	**deaf2bc**	**axybcz**	**raxbcyt**	**abcabc**	**cba**
abc	BOB	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
abc*	BOB	BOB1e	✗	✗	✗	✗	✗	✗	✗	BOBabc	✗
*abc	BOB	✗	d1BOB	✗	✗	✗	✗	✗	✗	abcBOB	✗
a*bc	BOB	✗	✗	BOB	✗	✗	✗	✗	✗	BOB	✗
a*b*c	BOB	✗	✗	BOB	BOB	✗	✗	✗	✗	BOB	✗
abc	BOB	BOB1e	d1BOB	✗	✗	deBOBfg	✗	✗	✗	BOBabc	✗
*a*bc	BOB	✗	d1BOB	BOB	✗	✗	deBOB	✗	✗	BOB	✗
a*bc*	BOB	BOB1e	✗	BOB	✗	✗	✗	BOBz	✗	BOBabc	✗
*a*bc*	BOB	BOB1e	d1BOB	BOB	✗	deBOBfg	deBOB	BOBz	rBOByt	BOBabc	✗
a*	BOBbc	BOBbc1e	✗	BOBd1bc	BOBd1be2c	✗	✗	BOBxybcz	✗	BOBbcabc	✗
*a	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	cbBOB
a	BOBbc	BOBbc1e	d1BOBbc	BOBd1bc	BOBd1be2c	deBOBbcfg	deBOBf2bc	BOBxybcz	rBOBxbcyt	BOBbcabc	cbBOB

Table 9.5 Wildcard pattern replacements—examples

10. GCSS Installation and Setup

This chapter describes how to:

- install the GCSS software (with recommendations for network installation);
- set up a password file;
- add and delete users.

It also describes the functions that are available to the Supervisor user.

Note: ***GCSS must not be installed on the same computer as ICS.***

10.1 Workstation Requirements

The GCSS requires:

- a computer complying with the minimum specifications listed in section 10.1.1;
- a printer—see section 10.1.2;
- a PROM programmer—see section 8.1.1.1;
- a PROM programmer serial cable—see section 10.1.4;
- a serial cable for the NVLM—see section 10.1.4.

10.1.1 Computer for GCSS

The minimum specifications required of a PC to operate the GCSS are:

- 500 MHz Intel Pentium P3 (or compatible) processor;
- 256 Mb RAM;
- 1 Gb hard disk with a least 100 Mb of free space;
- SVGA display card (800 x 600 or better);
- Microsoft Windows XP—Service Pack 2;
- CD-R drive or access to the Invensys Rail Tools database.

The minimum possible Windows configuration should be used with only essential background applications running (such as virus checkers).

Note: ***Do not use a screensaver.***

Note: ***We recommend that you enable Windows XP Automatic Updates to maximise the security of your computer and minimise operating system intrusions. (The safety of GCS is embodied in the GCSS and ICS software, and is achieved by following the defined application design process. It is not affected by updates to the Windows XP operating system.)***

See also “Showing Menu Navigation Underlines” on page A-6.

10.1.2 Printer

Local or network printer.

10.1.3 Serial PROM Programmers

GCSS supports the Dataman S4 directly.

Other serial PROM programmers can be used with GCSS provided they can operate with the following parameters:

- 9600 Baud (or better)
- RTS handshaking
- binary data

10.1.4 Cables

10.1.4.1 ELNEC BeeProg Communication Cables

You can connect the BeeProg to the computer using either:

- a standard USB cable (computer must support USB 2.0), or;
- a standard parallel cable connected between the BeeProg's LPT connector and a free Printer port on the computer.

We recommend USB because it is simple to use, and reliable.

10.1.4.2 Dataman S4 Serial Communication Cable

Dataman S24 DB25 Male Pin	Computer DB9 Female Pin
2	3
3	2
4	7
5	8
6	6
7	5
20	4

Figure 10.1 Dataman S4 serial cable pinout

10.1.4.3 NVLM Serial Cables

Two types of serial cable may be used for data transfer between the computer and the NVLM module.

Maintenance Cable

This is an RS232 cable with female DB9 connectors at both ends. Links within the cable select options (figure 10.2):

- pin 8 to pin 7—automatically switches the NVLM into Maintenance Mode;
- pin 9 to pin 7— automatically selects 115200 Baud data transfer rate (otherwise the NVLM uses 9600 Baud).

The NVLM must be restarted for the maintenance setting to take affect.

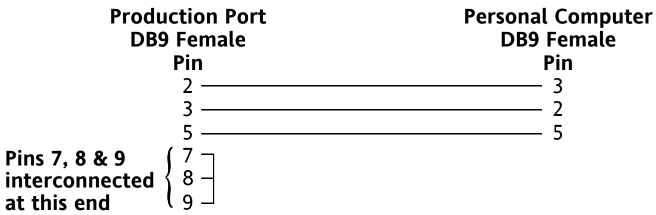


Figure 10.2 Serial cable—NVLM maintenance cable

Standard Cable

This is also an RS232 cable with female DB9 connectors at both ends, but without internal links (figure 10.3). When this cable is used:

- the data transfer rate for the Production port must be selected manually;
- the NVLM must be switched manually to Maintenance Mode, and;
- the NVLM must be restarted for the settings to take affect.

Maintenance Mode and the data transfer rate are selected and set by switches on the NVLM circuit board. See section 10.2.

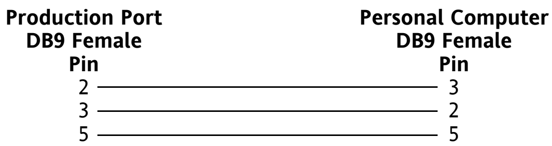


Figure 10.3 Serial cable—NVLM standard cable

10.2 NVLM Maintenance Mode Switch Settings

10.2.1 NCDM

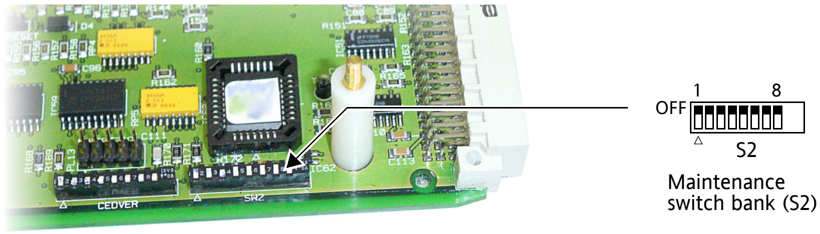


Figure 10.4 Maintenance Mode switches—NCDM

Set the S2 switch bank as follows:

Switch	Setting
1	OFF for Normal mode ON to select Maintenance Mode
2	OFF for 9600 bits per second ON for 115200 bits per second
3 – 8	Leave as is or see reference [APPM]

Table 10.1 Switch Bank S2—valid settings—NCDM

The NCDM will stay in Maintenance Mode until switch 1 is returned to OFF and the system is rebooted.

THE NCDM Watchdog LED flashes red and the display shows **MNT** to indicate when the module is in maintenance mode.

10.2.2 NVC/DM

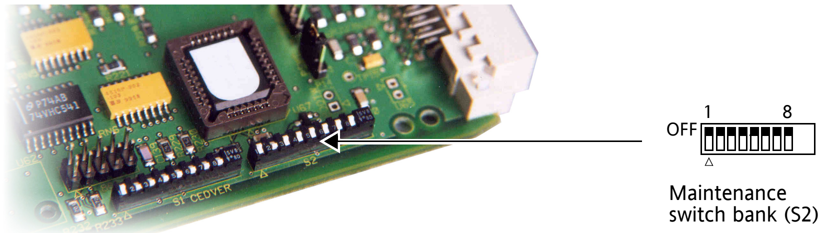


Figure 10.5 Maintenance Mode switches—NVC/DM

Set the S2 switch bank as follows:

Switch	Setting
1	OFF for Normal mode ON to select Maintenance Mode
2	OFF for 9600 bits per second ON for 115200 bits per second
3 – 8	Not used. Normally left in the OFF position.

Table 10.2 Switch Bank S2—valid settings—NVC/DM

The NVC/DM will stay in Maintenance Mode until switch 1 is returned to OFF and the system is rebooted.

THE NVC/DM Watchdog LED flashes red to indicate when the module is in maintenance mode.

10.3 Network Setup

The GCSS can be installed directly onto a PC or onto a network and be run on a local PC. Network file folders (if available) can be used for each type of installation.

Westinghouse Rail Systems Australia installs the GCSS onto PCs but a network installation is possible.

We recommend the following when a network setup of the GCSS is required:

- GCSS application files (`GCSS.EXE` and `GCSS.DAT`) are installed on a network directory that is "read-only" from a user's point of view;
- Data files (installations) and password files are in a network directory that has "read and write" access from a user's point of view;
- Each local PC is configured to run the network copy of GCSS.

  Users can save their installation files to their own choice of directory, preferably a project-based network directory that is regularly backed up. The GCSS will automatically save the associated Log files to the same directory.

See figure 10.6.

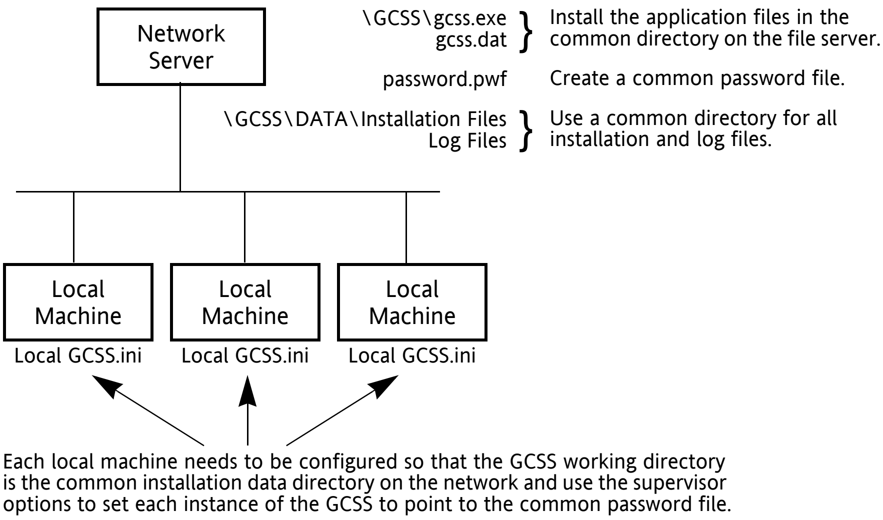


Figure 10.6 Network setup

10.4 Installing the GCSS Software

The GCSS can be installed directly from an installation file in the Design Tools Authority—Tools Database (Lotus Notes), or from CD. The installation file in the tools database can be detached and copied to a local hard drive if preferred.

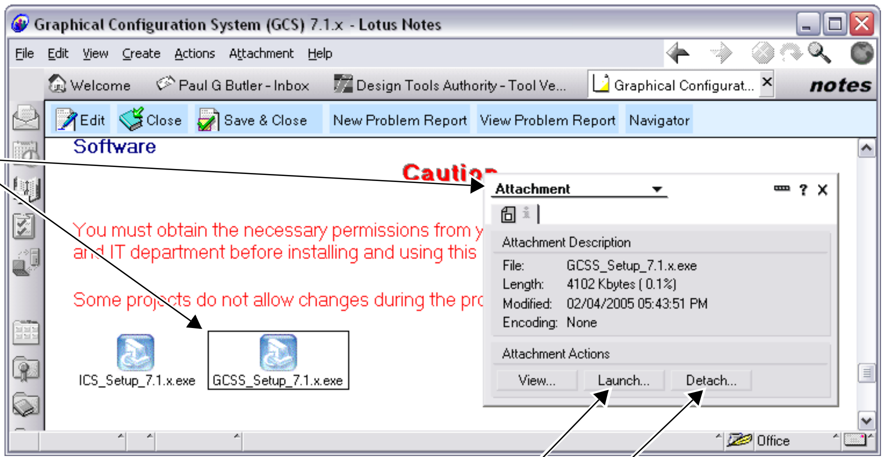
10.4.1 Installing from the Design Tools Authority—Tools Database

a) In Lotus Notes, open the Design Tools Authority—Tools Database, then open the entry for Graphical Configuration System (GCS) version 7.1.x.

b) Scroll down to the Software section (figure 10.7).

c) Double-click the GCSS 7.1.x installation file icon, then choose **Launch** or **Detach** (figure 10.7).

Double-click the GCSS installation icon to open the attachment action dialog box



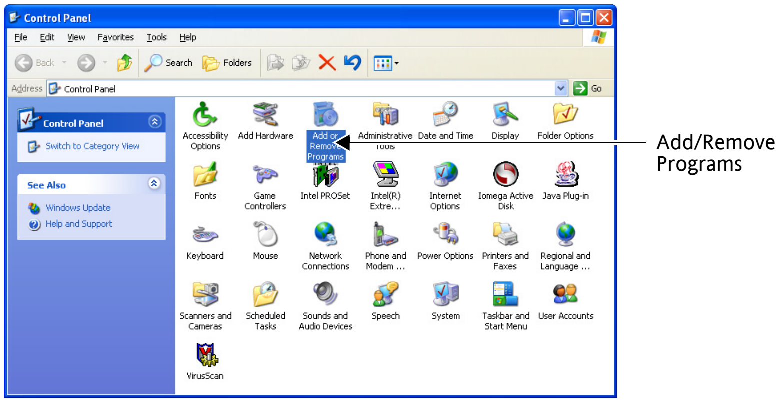
Click the **Launch** button to install GCSS directly from the Lotus Notes database;
OR
Click the **Detach** button to save a copy of the installation file on a local drive.

Figure 10.7 GCS 7.1.x Software section in Lotus Notes

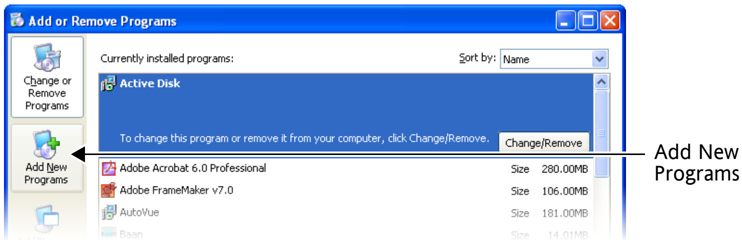
d) Continue as follows:
- at step **h** on page 10-8 when installing directly from Lotus Notes;
- at step **a** below when installing from a saved copy of the installation file (browse for the saved file as shown in figures 10.8 and 10.9).

10.4.2 Installing from a CD ROM or a Saved Copy of the Installation File

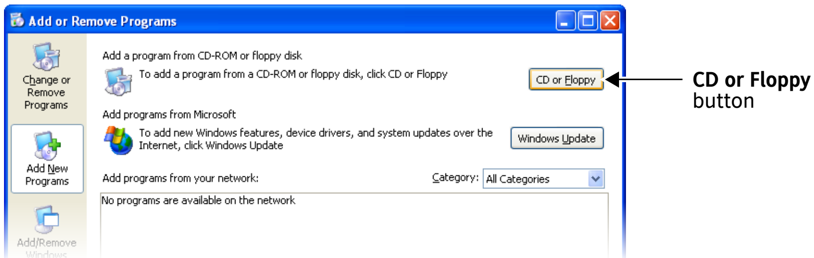
a) Click the **Add/Remove Programs** icon in the Windows Control Panel.



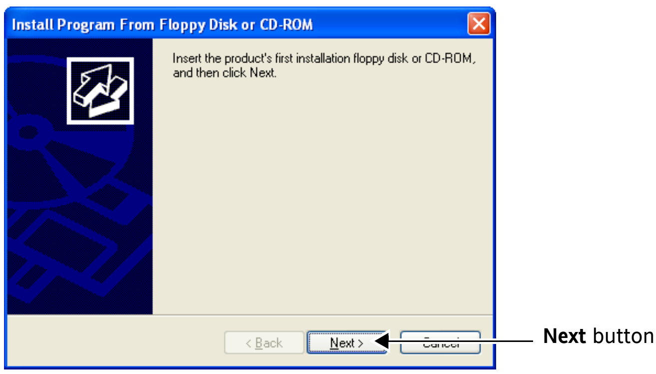
b) Click the **Add New Programs** button.



c) Click the **CD or Floppy** button to start program installation.



d) Insert the installation disk and select the **Next** button. Windows will continue at step **h** when it finds an installation program.



e) Windows will display the dialog box shown in figure 10.8 if it cannot find the installation program.

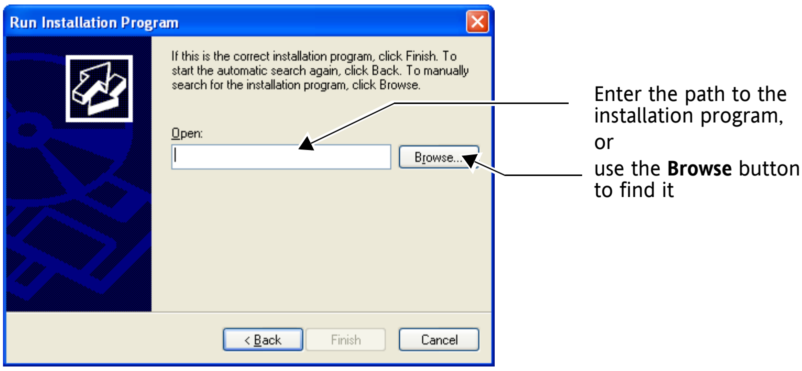


Figure 10.8 Installing GCSS—defining path to installation program—part A

f) Enter the path to the installation program or use the **Browse** button to find the installation program (figure 10.8).

g) Click **Finish** to continue, or click **Cancel** if you cannot find the installation program

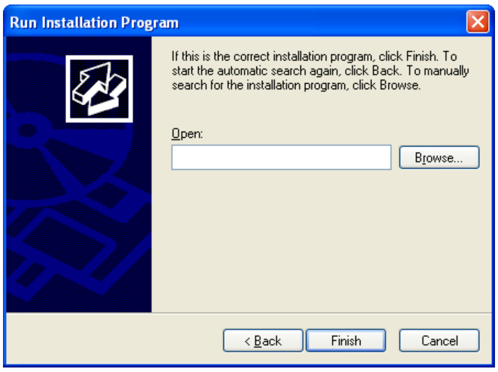
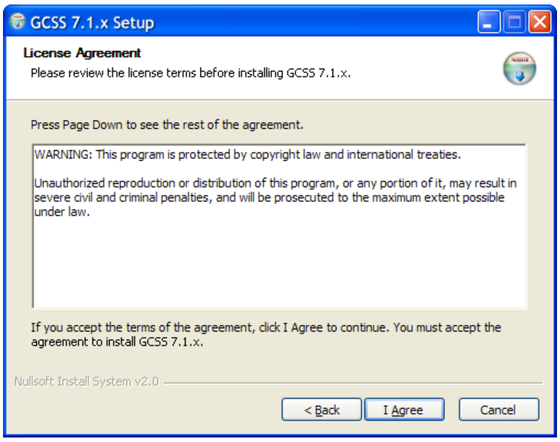


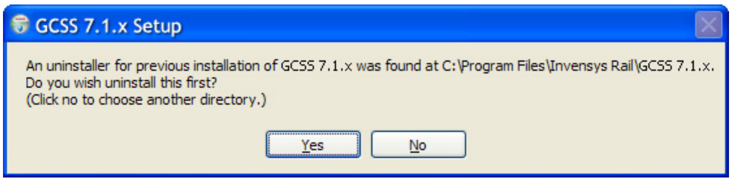
Figure 10.9 Installing GCSS—defining path to installation program—part B

h) In the Setup Wizard dialog box, confirm continuation of the installation process by clicking **Next**, or terminate the process by clicking **Cancel** if you have another program running. Restart at step **a** after closing the other program(s).

i) In the License Agreement dialog box, review the license terms then click **I Agree** if you accept them.



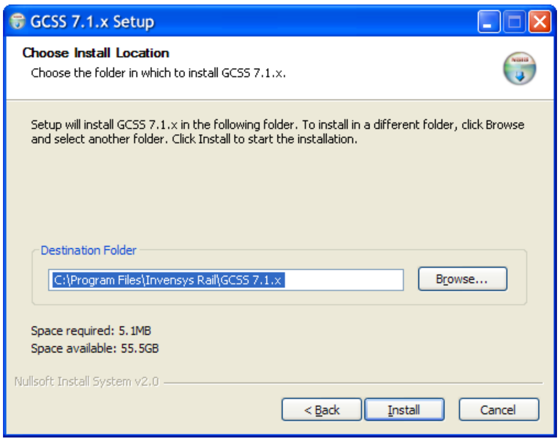
j) If the following dialog box opens:
- click **Yes** to overwrite the existing installation of GCSS 7.1.x—go to step **k**, or;
- click **No** to install the new version in a different location—go to step **m**.



k) Click **Yes** to confirm that you want to overwrite the existing installation.

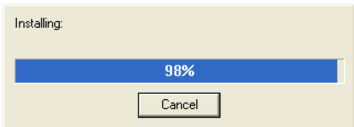
l) Click OK in the "successfully removed" dialog box.

m) The installation program proposes a location for installing the GCSS software.

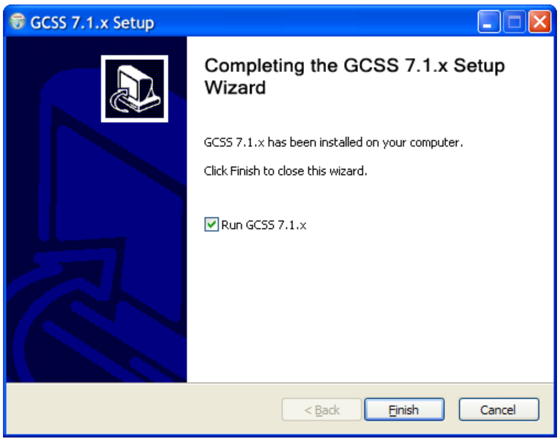


Accept the proposed location, or change it by clicking **Browse**. Then continue by clicking **Next**.

The installer displays a progress bar.



n) Click **Finish** to complete the installation.



The installer places a shortcut to GCSS 7.1.x on the desktop.

Note: ***The GCSS program displays the alert message shown in figure 10.10 when first activated because a password file is not available. The GCSS Supervisor must login as "SUPERVISOR" as described in section 10.5 and setup a password file as described in section 10.5.1.***

Figure 10.10 Password File alert message

10.5 Activating Supervisor Privileges

The first time GCSS is started and any other time Supervisor access is needed, login as the Supervisor user.

a) Start the GCSS.

b) Press any key or click the left mouse button to bypass the startup screen and display the login dialog box.

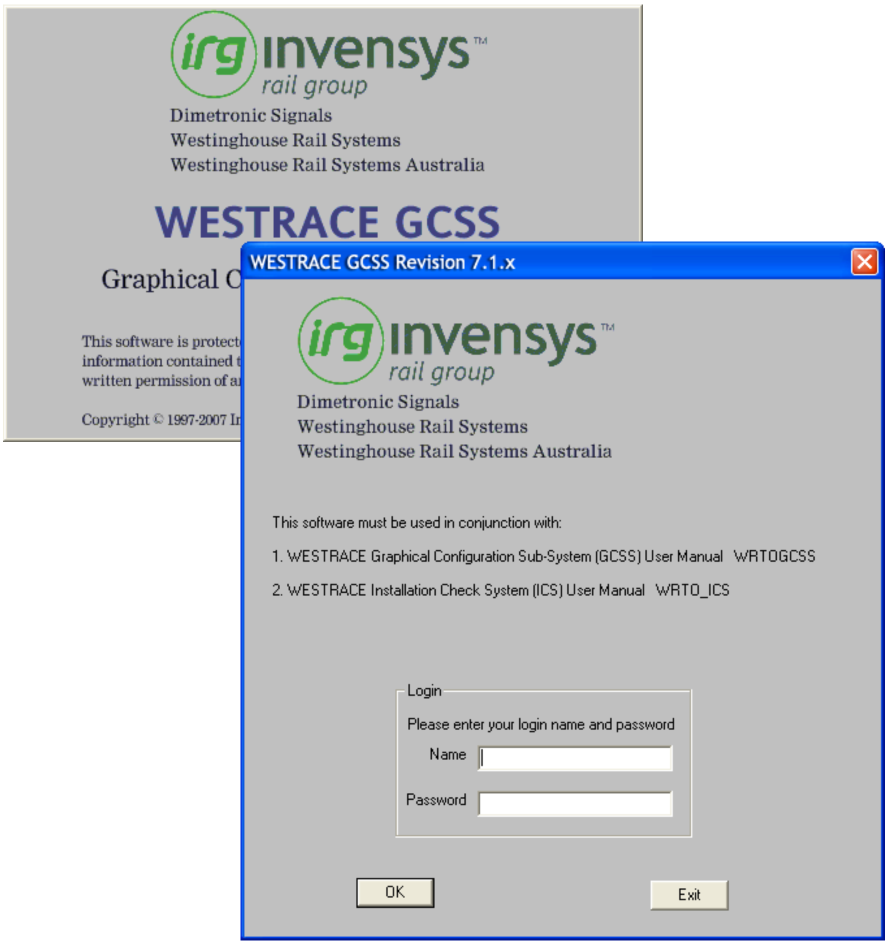


Figure 10.11 Startup screen and Login dialog box

c) Type `SUPERVISOR` in the Name field (GCSS converts name to uppercase).

d) Type the Supervisor password in the Password field (not case-sensitive).

e) Select **OK**.

Note: ***The supervisor password is distributed separately by Invensys Rail.***

The GCSS displays the Compatibility Index dialog box when it is started for the first time. Select the correct Compatibility Index. See section 3.4 for details.

The GCSS displays the Supervisor window:

- At first startup, the Supervisor window will be similar to figure 10.12. No password file is available and there is only one user, the Supervisor.
- At any other time, the Supervisor window will be similar to figure 10.13. A password file is available and there is at least one user in addition to the Supervisor.

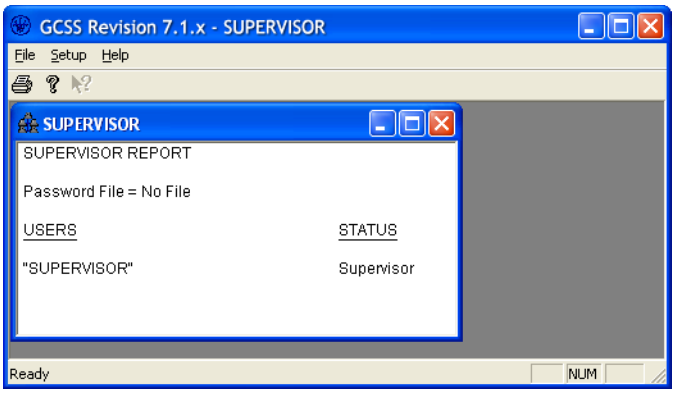


Figure 10.12 Supervisor window at startup

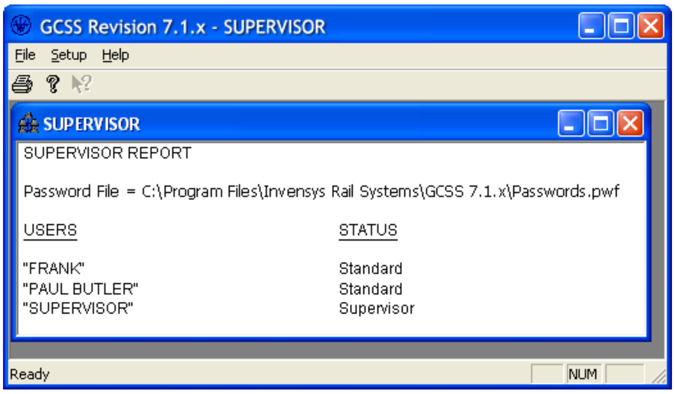


Figure 10.13 Supervisor window at any other time

The Supervisor can:

- set up password files and add users of the GCSS system;
- remove lock files. (If the computer shuts down before an installation is closed in an orderly manner, a lock file remains that prevents anyone opening the installation file. See section 10.5.5.)

The Supervisor cannot create or open any installation files.

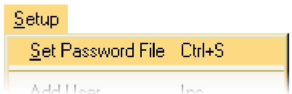
The Supervisor Window cannot be closed—GCSS must be shut down.

10.5.1 Setting the Password File

A GCSS password file must be created or selected before the GCSS can be accessed by users. It is an encrypted file that stores the user name and password for each registered user.

Note: ***Each copy of GCSS must be set up to access a password file.***

a) Select **Set Password File** to display the dialog box shown in figure 10.14.



Ctrl+S

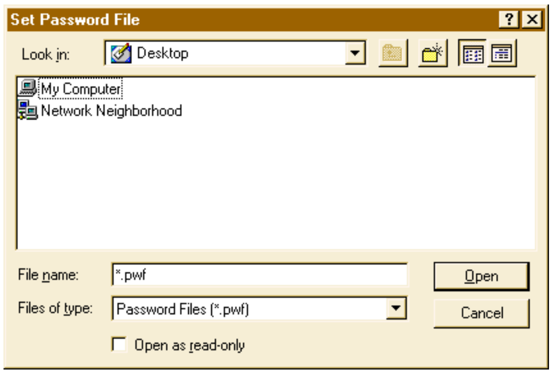


Figure 10.14 Set Password File dialog box

b) Select the desired directory then enter or select a file name:
   - A new name causes GCSS to ask if a new password file is to be created.
   - An existing name causes GCSS to check that the file is a valid password file (and report an error if it isn't).

c) Click **Open**.

The password file name is encrypted and stored in the `gcss.ini` file.

GCSS will show the Password File Alert Message (see figure 10.10) if the password file is missing or corrupted, or if the `gcss.ini` file on a user PC is missing or corrupted. In such a case, the password file must be replaced or recreated.

Caution: ***Create separate password files for GCSS and GTT (GCS Templates Tool). The same user names and passwords can be set up in both password files.***

10.5.1.1 Setting the Password File for Network Use

a) Login as supervisor on one PC.

b) Set up a new password file in a network directory common to all users of the GCSS, perhaps in a network directory used to store the GCSS installation data files.

c) Add all the user names and passwords for the users of the GCSS (refer to section 10.5.2).

d) Close the GCSS application.

e) Login as supervisor on each user PC and use the Set Password file option to set the password file to the configured password file name in the common network directory.

This must be done on every user PC to set up the local `gcss.ini` file.

Note: ***Several password files containing different users can be set up for different GCSS development teams.***

10.5.1.2 Setting the Password File for using GCSS on a local PC

a) Login as supervisor on the local (stand-alone) PC.

b) Setup a local password file in a local directory.

c) Add the appropriate user name as the single user of the system.

d) Close the GCSS application and re-open as the user.

10.5.1.3 Corrupt Password Files

The password file must be re-created:

- when the GCSS cannot open the existing file;
- the supervisor user cannot log in.

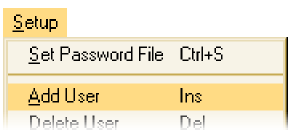
To rectify this situation:

a) Delete the corrupted password file;

b) Log in to the GCSS using the original supervisor password distributed by Invensys Rail.

10.5.2 Adding Users

a) Select **Add User** (in Supervisor view only) to display the dialog box shown in figure 10.15.



Insert

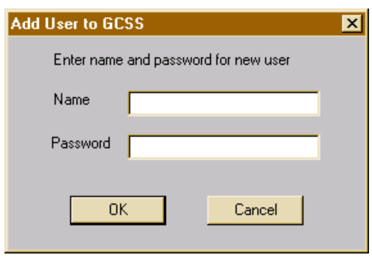


Figure 10.15 Add User dialog box

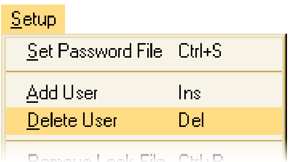
b) Enter:
- the name of the user in the **Name** field;
- a password for the user in the **Password** field.

The password can be changed by the user as described in section 3.3.

Note: ***User names and passwords cannot exceed 15 characters.***

10.5.3 Deleting Users

a) Select **Delete User** from the Setup menu (in Supervisor view only) to display the dialog box shown in figure 10.16.



Delete

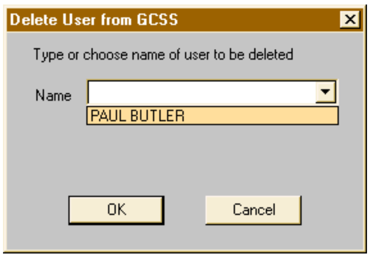


Figure 10.16 Delete User dialog box

b) Enter or choose the name of the user from the drop down list in the Name field.

c) Select **OK** to delete the user (after confirming the action in the confirmation dialog box).

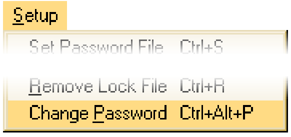
Selecting **Cancel** will abort the operation.

Figure 10.17 Confirm Delete User Dialog Box

Note: ***The Supervisor user cannot be deleted.***

10.5.4 Changing the Supervisor Password

a) Select **Change Password** from the Setup menu (in Supervisor view only) to display the dialog box shown in figure 10.18.



Alt s p
Ctl+Alt+p

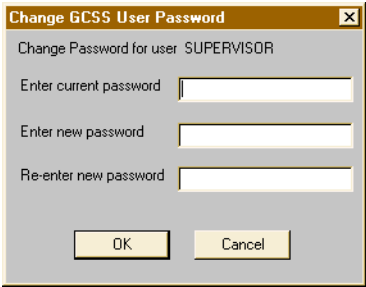


Figure 10.18 Change Supervisor Password dialog box

The labels on each field are self explanatory, however be aware that:

- what you type in a field will not be displayed;
- a password can contain numerals and uppercase and lowercase characters;
- password length can be 1 to 15 characters.

Caution: ***The Supervisor password must be changed to improve GCSS security.***

Note: ***The new Supervisor password is stored in the password file.***

The original Supervisor password distributed by Invensys Rail will not work after the Supervisor password is changed.

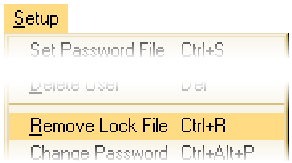
10.5.5 Removing Redundant Lock Files

The GCSS locks an installation file for the exclusive of the user opening it. The lock prevents another user from opening the same file at the same time.

The lock is a hidden read-only file created by GCSS in the same directory as the installation file when the installation file is opened. The lock file is removed when the installation file is closed.

An unwanted lock file can be left behind in circumstances such as a power failure or switching off the PC whilst an installation file is open. The installation file cannot be opened for further editing until the old lock file is removed.

a) Select **Remove Lock File** from the Setup menu to display the dialog box shown in figure 10.19.



Alt s r
Ctrl+f

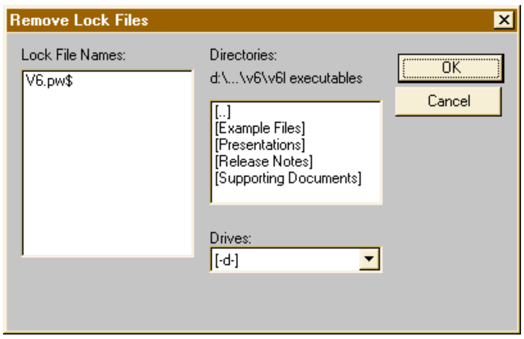


Figure 10.19 Remove Lock File dialog box

b) Select the drive and directory containing the locked installation file.

All lock files in the directory appear in the Lock File Names panel with a `.in$` extension.

c) Select the lock file that matches the locked installation file.

d) Select **OK**.

This will delete the lock file and allow the installation file to be opened for editing.

Password files are locked in a similar manner when the supervisor user logs into the GCSS.

Caution:

Do not remove lock files (file extension `.in$`) for installation files that are currently open.

Do not remove password files (file extension `.pw$`).

10.5.6 User Report

The Supervisor window (see figure 10.20), shows the name of current password file and the user details it contains.

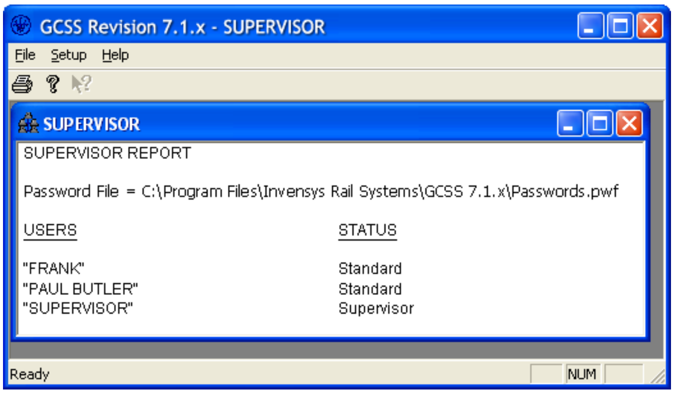
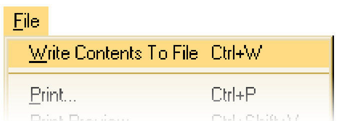


Figure 10.20 Supervisor window

The information in this window can be:

- previewed by selecting **Print Preview** from the File menu;
- printed by selecting **Print** from the File menu (or by choosing the print icon from the main toolbar or by using the accelerator key combination **Ctrl+P**);
- written to a disk file by selecting **Write Contents to File** from the File menu.



Alt f w
Ctl+w

See section B.5.18 on page B-23 for an example report.

APPENDIX A: GCSS INTERFACE

This appendix provides information about the GCSS software not found elsewhere in this manual. Space is provided for your own notes in section A.3, Tab Sheets.

A.1 GCSS Interface

The GCSS has a Multi Document Interface (MDI) application which means that more than one document (installation) can be open at any one time. Each installation and their lower level functions are displayed in child windows.

A.1.1 The Mouse

Many actions in Windows can be performed with the mouse. This section introduces the "mouse" terminology.

Note: ***Always use the left button on the mouse to click unless specifically directed to use the right mouse button.***

There are three actions associated with mouse buttons:

- **Click**—means "select". To Click on a display element (a button, for example) means that you must perform the following steps with the mouse:
  - place the tip of the mouse pointer on the display element;
  - press the mouse button once and release it.
- **Double-click**—means "select and enter". Perform the same steps as for Click, but press the mouse button twice in rapid succession.
- **Drag**—means place the mouse pointer on the required display element, press down on the mouse button and hold it down as you move the mouse. The display element will move with the mouse pointer.

A.1.2 Windows

A window is the work area on the computer screen. There might be a single, large window filling the entire computer screen, or there might be a number of windows side-by-side or overlapping on the screen.

A large window (main or parent window) might contain smaller windows (child windows) for particular purposes. Child windows always remain within the border of the main window.

A window, dialog box (or field) must be active (has the focus of Windows) for it to accept data.

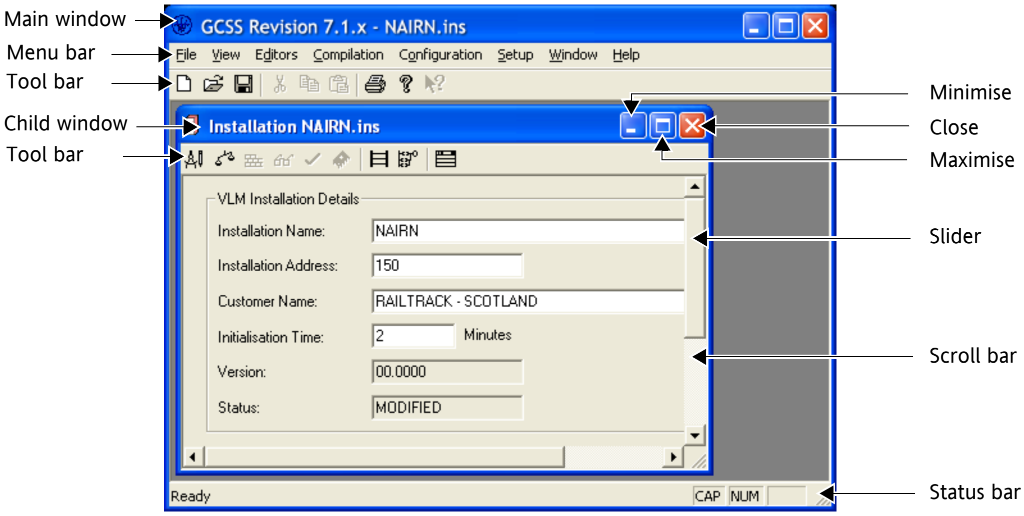


Figure A.1 GCSS windows elements

A.1.3 Making a Window Active

A window is active and ready to accept commands when its title bar is highlighted. A window can be active, and have a highlighted title, even when it is reduced to an icon at the bottom of the screen.

When several windows are visible, click in the required window with the mouse, or use the Window command (**Alt+W**) and select the required window from the window list. Only one window is active at any time.

Note: ***The colour attributes that distinguish an active window from its neighbours depends upon the configuration of Windows.***

A.1.4 Maximising, Minimising and Closing Windows

A window can be in one of three states:

- **Minimised**— reduced to the size of an icon and positioned at the bottom of the main window. An icon can be moved to any position in the main window working area.
- **Maximised**—occupies all of the working area within the main window.
- **Restored**—intermediate in size between an icon and maximum size; has a border that can be dragged to resize the window.

The main application window may itself be minimised to an icon on the windows desktop or maximised to occupy the entire screen.

- Minimise a window by clicking this button: 
- Maximise a window by clicking this button: 
- Restore a window by clicking this button: 
- Close a window by clicking on this button: 

These buttons are in the top right corner of the window.

A.1.5 Scroll Bars

Scrolling means to move up or down, or side to side through a document that's too large to fit in the window.

Scroll the contents of a window by:

- dragging the slider in the scroll bar with the mouse;

  or
- clicking the arrows at the end of the scroll bars with the mouse;

  or
- clicking in the scroll bar with the mouse;

  or
- using special keyboard keys:
  - **Page Up**—scrolls up one page
  - **Page Down**—scrolls down one page
  - **Up Arrow**—scrolls up one line
  - **Down Arrow**—scrolls down one line
  - **Left Arrow**—scrolls left
  - **Right Arrow**—scrolls right
  - **Home**—scrolls to the top of the window
  - **End**—scrolls to the bottom of the window

A.1.6 Menu Bar

This bar is at the top of the main window (see figure A.1). The items available from the menu bar change according to the child window that is currently active.

A.1.7 Context Menus

Context menus allow fast access to commonly used menu items within the context of the current activity. A context menu is invoked by right-clicking (with the mouse) within the workspace of a GCSS window. Each item in the context menu performs the same function as the same named item available from the Menu Bar within the same context.

A.1.8 Toolbars

Selecting a button on a toolbar has exactly the same effect as selecting the corresponding item from a menu.

A.1.8.1 Tool Tips

Every button on a toolbar displays a description of the button's function when the mouse pointer is over the button.

A.1.8.2 Main Tool Bar

This tool bar is at the top of the main window below the menu bar. It contains buttons that provide fast access to the more commonly used features. It can be hidden or displayed by selecting *Toolbar* from the View menu.

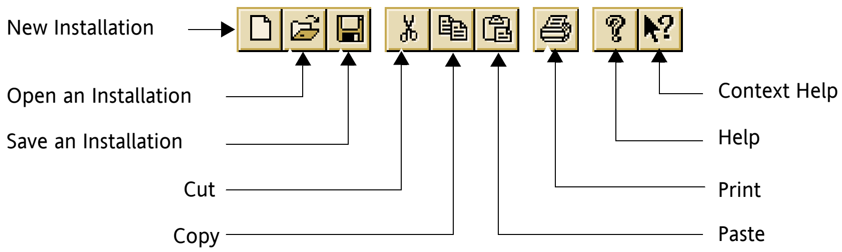


Figure A.2 Main toolbar

A.1.8.3 Child Window Toolbars

Some child windows have tool bars of their own. Use these for fast access to features peculiar to the child window.

A.1.9 Status Bars

A.1.9.1 Main Status Bar

This bar is at the bottom of the main window (see figure A.1). It displays information and the status of the Caps Lock, Num Lock, and Scroll Lock keys.

Select *Status Bar* from the View menu to display or hide the Status Bar.

A.1.9.2 Child Window Status Bar

Some child windows have a status bar containing information relevant to the particular child window.

A.1.10 Help and Version Numbers

Windows help should be available if it was installed on your PC. Online help for the GCSS is not yet available.

See section 1.12, "About GCSS" for GCSS revision and build numbers.

A.1.11 Accelerator Keys

The GCSS offers a variety of keyboard shortcuts to items on the Menu Bar and many underlying menu options. The items that respond to keyboard commands usually feature an underlined character (in the menu label) which is the activating character for the item. Some use a function key (eg **F1**, **F2** etc) or a **Ctrl+** command, which might be an alternative to an **Alt+** command.

Accelerator keys are listed throughout this manual whenever they are available for an operation.

Showing Menu Navigation Underlines

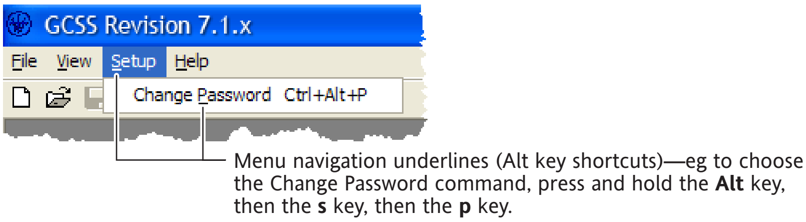


Figure A.3 Menu underlines

You can choose whether menu navigation keys are:

- always underlined or;
- underlined only when you press the Alt key.

a) Choose **Start** menu > (**Settings** >) **Control Panel** > **Display** (or right-click the Windows desktop and choose **Properties** from the context menu).

b) In the Display Properties dialog box (figure A.4), click the **Appearance** tab then the **Effects** button.

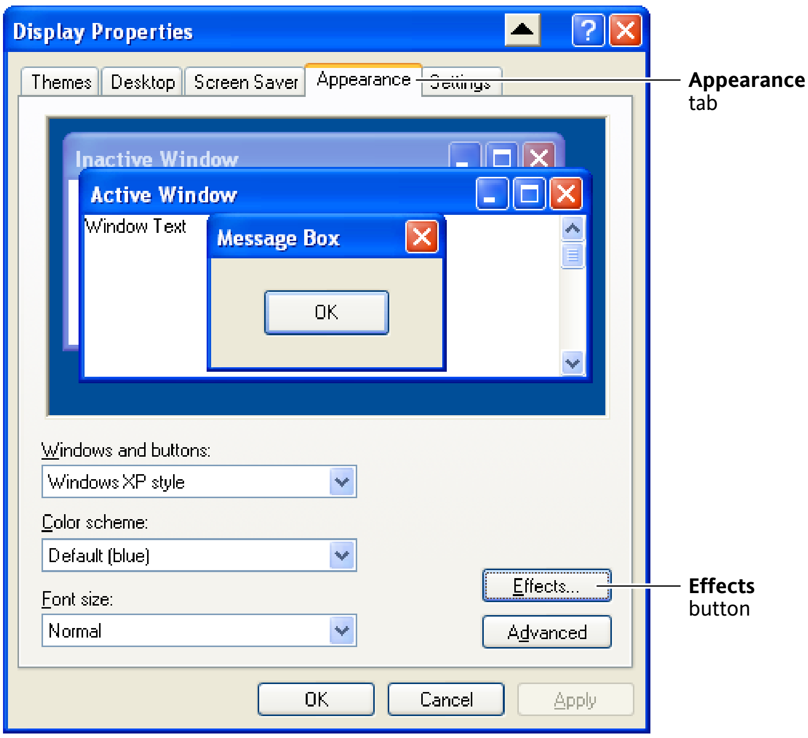


Figure A.4 Display Properties dialog box

c) In the Effects dialog box (figure A.5), tick or clear the **Hide underlined letters for keyboard navigation until I press the Alt key** checkbox as required.

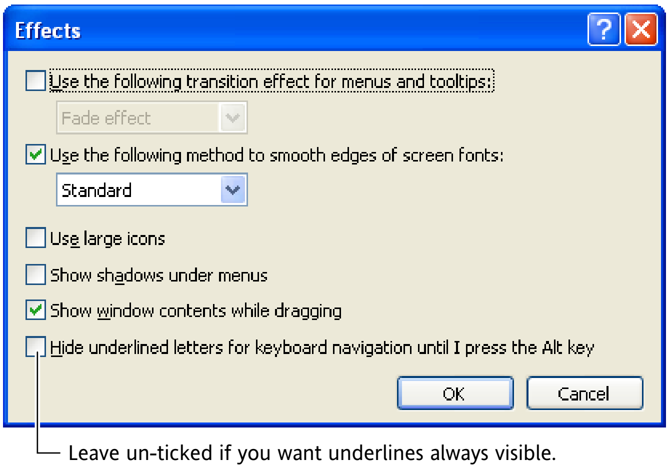


Figure A.5 Effects dialog box

d) Click **OK** to close the Effects dialog box.

e) Click **OK** to close the Display Properties dialog box.

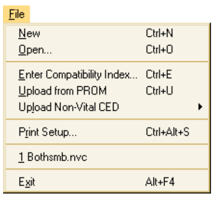
A.2 Menus and Dialog Boxes

A.2.1 File Menu

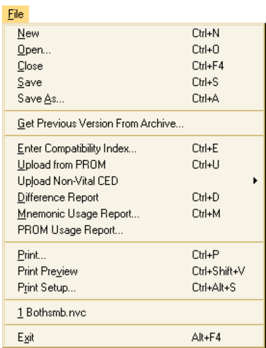
There are six versions of this menu:

- No installation open
- One or more installations open
- Housing editor active
- Module editor active
- Ladder editor active
- Supervisor window open

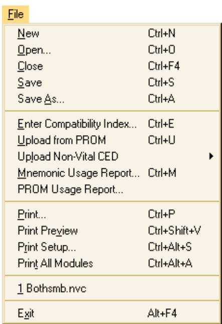
They differ in the number and type of options available.



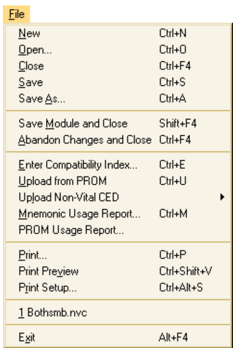
No Installation open



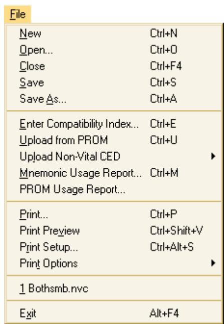
One or more Installations open



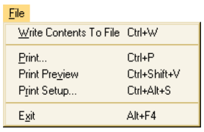
Housing editor active



Module editor active



Ladder editor active



Supervisor window open

New—create a new installation. See section 3.5.1.

Open—open an existing installation. See section 3.5.2 and section A.2.2.1.

Close—close an open installation.

Save—save an open installation (not an open archive) to the same name. See section 3.10.2.1.

Save As—save an open installation to a different name or directory. See section 3.10.2.2 and section A.2.2.2.

Save Module and Close—write recent module edits to disk and closes the Module editor.

Abandon Changes and Close—abandons recent module edits to disk and closes the Module editor.

Get Previous Version From Archive—open a previous version of a currently open installation. See section 3.5.3.

Also use this command to:

- discard the most recent set of changes;
- check changes using the difference report (see section 3.9 and section B.5.2).

Enter Compatibility Index—a number that identifies the company that created the installation. See section 3.4.

Upload From PROM—upload a VLM installation from a set of PROMs. See section 8.3.1.

Upload From Non-Vital CED—upload an NVLM installation from an NVC/DM or NCDM. See section 8.3.2.

Difference Report—compares two installations (or two versions of the one installation) by listing all differences. See section 3.9 and section B.5.2.

Mnemonic Usage Report—lists all mnemonics used in Installation. See section 3.7 and section B.5.3.

PROM Usage Report—lists statistics on the contents of VLM PROMs. See section 3.8 and section B.5.4.

Non-Vital CED Usage Report—lists statistics on the contents of NVLM Non-Vital Configuration. See section 3.8 and section B.5.4.

Write Contents to File—writes the User Report to a disk file. See section 10.5.6 and section B.5.18.

Print—sends a report for the current editor, window, activity etc to the default printer (on the network or locally connected to the user PC). See Appendix B.

Print Preview—displays an on-screen image of a report as it would appear when printed. The report can be sent to the default printer by clicking on the **Print** button.

Print Setup—opens a standard Windows dialog box for selecting and arranging the properties of a local or network connected printer.

Print All Modules—print details of all modules in an installation. See section 4.2.6.2 and section B.5.9.

Print Options—select a single or two column Ladder Editor report

Previous File List—displays the most recently saved installations (up to four). Select a name from the list to open the installation file.

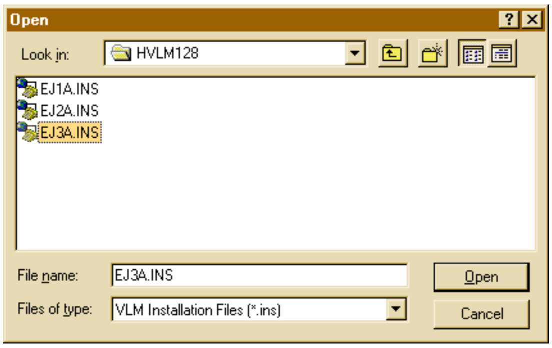
Exit—closes the GCSS and any open installations in an orderly fashion. This is the preferred way to close down the GCSS because it guarantees all files are properly closed before the program terminates.

Note: ***If the GCSS is closed prior to programming PROMs, some work may need repeating. PROMs must be programmed in the same session as Compilation.***

A.2.2 File Menu—Dialog Boxes

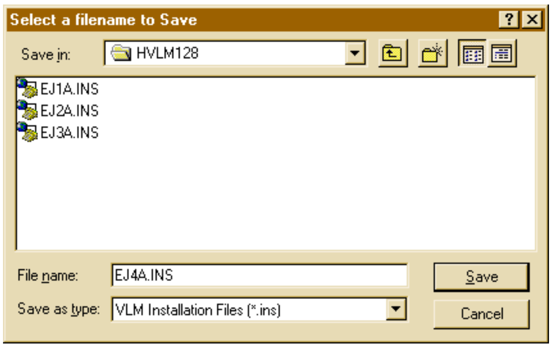
This section covers the GCSS dialog boxes accessible through the File menu.

A.2.2.1 Open



Double-click on the name of the required installation file or click on the file name and then on the **Open** button.

A.2.2.2 Save As



a) Enter a name for the installation file.

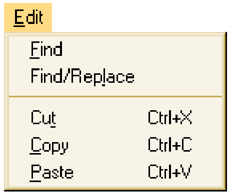
   If a file with the same name already exists in the nominated directory, the GCSS will prompt for permission to overwrite it.

b) Click on the **Save** button.

A.2.3 Edit and Editors Menus

This section summarises the availability of the editing functions within the GCSS editors.

Installation active



Editors active

Cut—removes the selected object.

Active Editor	Available	Comment
Housing Editor	✓	See section 4.2.4.3.
Module Editor	✓	See sections 5.1.6.3.4, 5.1.6.3.5 and 5.1.6.3.6.
Ladder Editor	✓	See sections 6.2.3.4.
Rung Editor	✗	

Copy—copies the selected object.

Active Editor	Available	Comment
Housing Editor	✓	See section 4.2.4.3.
Module Editor	✓	See sections 5.1.6.3.4, 5.1.6.3.5 and 5.1.6.3.6.
Ladder Editor	✓	See sections 6.2.3.4.
Rung Editor	✗	

Paste—pastes the copied object to the selected position.

Active Editor	Available	Comment
Housing Editor	✓	See section 4.2.4.3.
Module Editor	✓	See sections 5.1.6.3.4, 5.1.6.3.5 and 5.1.6.3.6.
Ladder Editor	✓	See sections 6.2.3.4.
Rung Editor	✗	

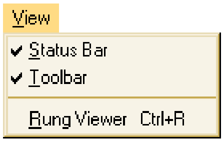
Find—searches mnemonic fields for a user-entered text string.

Active Editor	Available	Comment
Housing Editor	✓	See section 9.1.
Module Editor	✓	"
Ladder Editor	✓	"
Rung Editor	✓	"

Find and Replace— searches mnemonic fields for a user-entered text string and replaces it with the user-entered replacement string.

Active Editor	Available	Comment
Housing Editor	✓	See section 9.2.
Module Editor	✓	"
Ladder Editor	✓	"
Rung Editor	✓	"

A.2.4 View Menu



Toolbar—toggles the GCSS main tool bar on or off.

Status Bar—toggles the GCSS status bar on or off.

Rung Viewer—opens the Rung Viewer (only when the Ladder Editor is active).

A.2.5 Windows Menu

The Windows menu appears only when a child window is open within the GCSS main window.

Cascade—arranges child windows by placing them one behind another, each with a corner exposed.

Tile—arranges child windows in a tiled pattern to fill the main window.

Arrange Icons—arranges icons belonging to any minimised child windows along the bottom of the main window display.

Current file—lists all open child windows. Selecting a window from the list activates it and moves it to the front.

A.3 Tab Sheets

The GCSS representations of all WESTRACE modules use multiple tab sheets. This section provides a brief description of each tab sheet with guidance to field content. Space is provided for you to add your own notes.

User editable fields are indicated with ✓.

All modules show these fields on the "General" tab:

Field	Comment
Address	
Housing Nbr	The number of the physical housing containing the module. 1 to 4
Slot Nbr	The number of the housing slot holding the module. 1 to 15

Most modules show this field on the "General" tab:

Field		Comment
Module Seq Nbr	✓	Defines the place of the module in the processing sequence. 1 to 99

A.3.1 CCM

Also see section 2.4.7.6.

Tab	Field		Comment	Notes
	Local Port Addr.	✓	0 to 7	
	System ID	✓	0 to 99	
	PDV			
Communications	Communications	✓	WESTRACE to WESTRACE WESTRACE to MBP	
	Adj Inst Name	✓	Adjacent installation name.	
	Adj Inst Address	✓	Adjacent installation address. 0 to 9999999	
	Adj Port Address	✓	Adjacent port address. 0 to 7	
	HDLC Status	✓		
	Link Status	✓		
	Message Status	✓		
	Sequence Status	✓		
	Header Status	✓		

Tab	Field		Comment	Notes
Outputs	Output (Id)		1 to 161	
	Output Name	✓		
	Initial State	✓	High or Low	
Inputs	Input (Id)		1 to 161	
	Input Name	✓		
	Initial State	✓	High or Low	

A.3.2 CNVC

Tab	Field		Comment	Notes
General	Field Addr.	✓		
	Data Rate	✓	1200, 2400, 4800	
	Pair	✓	Yes or No	
Outputs	Output (Id)		1 to 240	
	Output Name	✓		
	Initial State	✓	High or Low	
Inputs	Input (Id)		1 to 240	
	Input Name	✓		
	Initial State	✓	High or Low	

A.3.3 DIAG, DM

No user-editable fields.

A.3.4 EVTC

Also see section 2.4.7.5.

Tab	Field		Comment	Notes
General	Local Port Addr.	✓	0 to 7	
	Adjacent Installation Name	✓		
	Adjacent Installation Address	✓	0 to 9999999	
	Adjacent Port Address	✓	0 to 7	
Outputs	Output (Id)		1 to 66	
	Output Name	✓		
	Initial State	✓	High or Low	

Tab	Field		Comment	Notes
Inputs	Input (Id)		1 to 66	
	Input Name	✓		
	Initial State	✓	High or Low	

A.3.5 GPOM110L and GPOM50LL

Also see section 2.4.7.8.

Tab	Field		Comment	Notes
General	Fault Input Name	✓		
Flashing/Dimming	Flash Status Input Name	✓	Not required unless "Flash Control Name" entered.	
	Flash Line Selection	✓	Not available unless "Flash Control Name" entered. Line 1 or 2	
	Flash Mark Space Ratio	✓	Not available unless "Flash Control Name" entered. Min. 5 to 100 in steps of 5. Max. 5 to 100 in steps of 5.	
	Flash Frequency	✓	Not available unless "Flash Control Name" entered. Min. 0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.85, 1.0, 1.15, 1.33 Max. 0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.85, 1.0, 1.15, 1.33	
	Flash De-en Power (% of RMS)	✓	Fixed at 100% unless "Flash Control Name" entered. Range: 0% to 70% in 5% steps.	
	Dim Power (% of RMS)	✓	Fixed at 100% unless "Dimming Control Name" entered. Range: 50% to 90% in 5% steps.	
	Dimming Control Name	✓		
	Dim Status Input name	✓	Not required unless "Dimming Control Name" entered.	
Outputs	Output (Id)		1 to 4	
	Output Name	✓		
	Initial State	✓	High or Low	
	Output Status Name	✓		
	Norm. Current Min.			
	Norm. Current Max.			
	Hot Proving	✓	Yes or No	

Tab	Field		Comment	Notes
	Cold Proving	✓	Yes or No	
	FONZ Current Min.			
	FONZ Current Max.			
	Flash Control Name	✓		
	Init. Flash State	✓	Not available unless "Flash Control Name" entered. Low or High	
	Flash Phase	✓	Not available unless "Flash Control Name" entered. In phase or Anti-phase	
	Dimming	✓	Not available unless "Dimming Control Name" entered. Enabled or Disabled	
	Dim. Current Min.			
	Dim. Current max.			

A.3.6 GPOMFSU

Tab	Field		Comment	Notes
General	Fault Input Name	✓		
Outputs	Output (Id)		1 to 4	
	Output Name	✓		
	Initial State	✓	High or Low	

A.3.7 HVLM128

Tab	Field		Comment	Notes
General	Mode	✓	Hot Standby or Stand-Alone	
Reserved Timers			See section 2.4.6.1.	
User Timers	Timer (Id)		1 to 193	
	Trigger Name	✓		
	Timer Name	✓		
	Duration	✓	0.000 to 0.999, 0 to 99999	
	Units	✓	Hrs, Mins, Secs	
Reserved Latches			See section 2.4.5.1.	
User Latches	Latch (Id)		1 to 2057	
	Name	✓		

A.3.8 HVLM128a

Tab	Field		Comment	Notes
General	Mode	✓	Hot Standby or Stand-Alone	
Reserved Timers			See section 2.4.6.1.	
User Timers	Timer (Id)		1 to 143	
	Trigger Name	✓		
	Timer Name	✓		
	Duration	✓	0.000 to 0.999, 0 to 99999	
	Units	✓	Hrs, Mins, Secs	
Reserved Latches			See section 2.4.5.1.	
User Latches	Latch (Id)		1 to 1800	
	Name	✓		

A.3.9 NCDM (VLM Installation)

Tab	Field		Comment	Notes
General	Interface File Name	✓		
	Version			
	Status			
	Ncdm File Status			
Inputs	Input (Id)		Not available unless “Interface File Name” entered. 1 to 2032	
	Input Name	✓		
	Initial State	✓	High or Low	
Outputs	Output (Id)		Not available unless “Interface File Name” entered. 1 to 2032	
	Output Name	✓		
	Initial State	✓	High or Low	

A.3.10 NCDM (NVLM Installation)

Tab	Field		Comment	Notes
General	Housing Nbr	✓	Defaults to 1 (assumed to be part of aVLM6 installation).	
	Slot Nbr	✓	Defaults to 3 (assumed to be part of a VLM6 installation).	
	Read Password	✓	Defaults to WESTRACE	
	Read/Write Password	✓	Defaults to WESTRACE	

Tab	Field		Comment	Notes
	Mode	✓	Hot Standby or Stand-Alone	
	Primary IP Address	✓		
	Secondary IP Address	✓	Not available unless in "Hot Standby" mode.	
Ports:			0 to 4 ports can be set up to 1 of 4 possible types with up to 6 protocols: • Serial port – Diagnostic; WSL/S2 Server; WSA/S2 Client; WSA/S2 Server; • VLM port – VLM Interface; • Network port – Diagnostic; WSL/S2 Server; WSA/S2 Server; WSA/S2 Client; WESTRACE Vital. Terminology: Server = Slave or Field, Client = Master or Office	
– Diagnostic			Also see section 2.6.4.6.1.	
– General	LOC Timeout	✓	Defaults to 1 second.	
	Connection Type	✓	Permanent cable or modem connection. Intermittent cable connection. Hayes Modem (dial in only). Hayes Modem (dial out only). Hayes Modem (dial in and out).	
	Connected State Input	✓		
	Port Address	✓	Defaults to 1. Range 1 to 127.	
	Initialisation String	✓	Defaults to E0V1	
	Auto Answer String	✓	Defaults to S0=3	
– Serial Interface	Handshake Mode	✓	None or RTS/CTS	
	PEN Signal Output	✓	Transmit Only or Always Active	
	Carrier Defect	✓	Ignore or Used	
	Baud Rate	✓	1200, 2400, 4800, 9600, 19200, 38400, 57600, 64000.	
	Electrical Char.	✓	RS232C or RS485	
– Phone Numbers			0 to 5 phone numbers	
	Phone Number	✓		
	Description	✓		
– WSL S2 Server			Also see section 2.6.4.6.3.	

Tab	Field		Comment	Notes
– General	All Modules OK Input	✓		
	LOC Timeout	✓	Defaults to 1 second.	
	Primary Port	✓	None, 1, 2, 3, 4.	
	Broadcast Window	✓	Defaults to 25.	
– Serial Interface	Handshake Mode	✓	None or RTS/CTS	
	PEN Signal Output	✓	Transmit Only or Always Active	
	Carrier Detect	✓	Ignore or Used	
	Baud Rate	✓	1200, 2400, 4800, 9600, 19200, 38400, 57600, 64000.	
	Electrical Char.	✓	RS232C or RS485	
– WSL S2 Housing			Up to 63 housings (numbers 0 to 63, excluding 13)	
– General	Housings OK Input	✓		
	LOC Timeout	✓	Defaults to 1 second.	
– Housing Slot			Up to 16 slots (numbers 1 to 16)	
– Inputs	Input (Id)		1 to 32	
	Input Name	✓		
	Initial State	✓	High or Low	
	Flash Control	✓		
	Initial State	✓	High or Low	
– Outputs	Output (Id)		1 to 32	
	Output Name	✓		
	Initial State	✓	High or Low	
– WSA S2 Client			Also see section 2.6.4.6.5.	
– General	All Modules OK Input	✓		
	LOC Timeout	✓	Defaults to 1 second (Range 1 to 60)	
	Data Word Length	✓	Range: 32, 48, 64, 96, 128, 256.	
	Tx Clock Direction	✓	From Port or External	
	Rx Clock Direction	✓	From Port, External or Reconstructed.	
	Interscan Delay	✓	Defaults to 20.	
– Serial Interface	Handshake Mode	✓	None or RTS/CTS	

Tab	Field		Comment	Notes
	PEN Signal Output	✓	Transmit Only or Always Active	
	Carrier Detect	✓	Ignore or Used	
	Baud Rate	✓	1200, 2400, 4800, 9600, 19200, 38400, 57600, 64000.	
	Electrical Char.	✓	RS232C or RS485	
- Modules			Up to 62 Master Modules (numbers 1 to 62)	
- General	Module OK Input	✓		
	LOC Timeout	✓	Defaults to 1 second.	
- Inputs	Input (Id)		1 to 32	
	Input Name	✓		
	Initial State	✓	High or Low	
- Outputs	Output (Id)		1 to 32	
	Output Name	✓		
	Initial State	✓	High or Low	
	Flash Control	✓		
	Initial State	✓	High or Low	
- WSA S2 Server			Also see section 2.6.4.6.4.	
- General	All Modules OK Input	✓		
	LOC Timeout	✓	Defaults to 1 second. (Range 1 to 60)	
	Primary Port	✓	None, 1, 2, 3, 4.	
	Data Word Length	✓	Range: 32, 48, 64, 96, 128, 256.	
	Tx Clock Direction	✓	From Port or External	
	Rx Clock Direction	✓	From Port, External or Reconstructed.	
- Serial Interface	Handshake Mode	✓	None or RTS/CTS	
	PEN Signal Output	✓	Transmit Only or Always Active	
	Carrier Detect	✓	Ignore or Used	
	Baud Rate	✓	1200, 2400, 4800, 9600, 19200, 38400, 57600, 64000.	
	Electrical Char.	✓	RS232C or RS485	
- Modules			Up to 62 Slave Modules (numbers 1 to 62)	
- General	Module OK Input	✓		
	LOC Timeout	✓	Defaults to 1 second.	
- Inputs	Input (Id)		1 to 32	

Tab	Field		Comment	Notes
	Input Name	✓		
	Initial State	✓	High or Low	
	Flash Control	✓		
	Initial State	✓	High or Low	
– Outputs	Output (Id)		1 to 32	
	Output Name	✓		
	Initial State	✓	High or Low	
– VLM Interface				
– General	LOC Timeout	✓	Defaults to 1 second.	
	Address			
	Interface File Name	✓		
	Version			
	Status			
	VLM File Status			
– Inputs	Input (Id)		1 to 2032	
	Input Name	✓		
	Initial State	✓	High or Low	
– Outputs	Output (Id)		1 to 2032	
	Output Name	✓		
	Initial State	✓	High or Low	
– Network Port				
– General	All Sessions OK Input	✓		
	LOC Timeout	✓	Defaults to 1 second. (Range 1 to 20)	
– Sessions			Up to 48 sessions (1 to 48) with up to 5 protocols: • Diagnostic; • WSL/S2 Server; • WSA/S2 Server; WSA/S2 Client; • WESTRACE Vital. Terminology: Server = Slave or Field, Client = Master or Office	
– Diagnostic			Also see section 2.6.4.4.1.	
– General	Session Number		Number of session selected under Network Port (1 to 48)	
	Session OK Input	✓		
	LOC Timeout	✓	Defaults to 1 second.	
	Diag Port Address	✓	1 to 127	
– WSL/S2 Server			Also see section 2.6.4.4.2.	

Tab	Field		Comment	Notes
– General	Session Number		Number of session selected under Network Port (1 to 48)	
	Session OK Input	✓		
	LOC Timeout	✓	Defaults to 1 second.	
	Application Session Id.	✓	Defaults to 0 which is not valid. (Range 1 to 255)	
	Broadcast Window	✓	Defaults to 0 which is not valid. (Range 5 to 100)	
	Housing Address	✓	Defaults to 1 (Range 0 to 63, excluding 13.)	
– WSA/S2 Server			Also see section 2.6.4.4.3.	
– General	Session Number		Number of session selected under Network Port (1 to 48)	
	Session OK Input	✓		
	LOC Timeout	✓	Defaults to 1 second.	
	Application Session Id.	✓	Defaults to 0 which is not valid. (Range 1 to 255)	
	Housing Address	✓	Defaults to 1 (Range 1 to 62)	
	Data Word Length	✓	Defaults to 32 (Range 32, 48, 64, 96, 128, 256)	
– Inputs	Input (Id)		1 to 32	
	Input Name	✓	Appropriate mnemonic	
	Initial State	✓	High or low	
	Flash Control	✓	Appropriate mnemonic	
	Initial State	✓	High or low	
– Outputs	Output (Id)		1 to 32	
	Output Name	✓	Appropriate mnemonic	
	Initial State	✓	High or low	
– WSA/S2 Client			Also see section 2.6.4.4.4.	
– General	Session Number		Number of session selected under Network Port (1 to 48)	
	Session OK Input	✓		
	LOC Timeout	✓	Defaults to 1 second.	
	Application Session Id.	✓	Defaults to 0 which is not correct. (Range 1 to 255)	
	Primary IP Address	✓	Defaults to 127.0.0.1	
	Secondary IP Address	✓	Defaults to 127.0.0.1	
	Housing Address	✓	Defaults to 1. (Range 1 to 62)	
	Data Word Length	✓	Defaults to 32 (Range 32, 48, 64, 96, 128, 256)	
– Inputs	Input (Id)		1 to 32	

Tab	Field		Comment	Notes
	Name	✓	Appropriate mnemonic	
	Initial State	✓	High or low	
– Outputs	Output (Id)		1 to 32	
	Output Name	✓	Appropriate mnemonic	
	Initial State	✓	High or low	
	Flash Control	✓	Appropriate mnemonic	
	Initial State	✓	High or low	
– WESTRACE Vital			Also see section 2.6.4.4.5.	
– General	Session Number		Number of session selected under Network Port (1 to 48)	
	Session OK Input	✓		
	Remote Inst. Address	✓	Defaults to 0	
	Primary IP Address	✓	Defaults to 0.0.0.0	
	Secondary IP Address	✓	Defaults to 127.0.0.1	
Reserved Timers			See section 2.4.6.2.	
User Timers	Timer Id		1 to 3000	
	Trigger Name	✓		
	Timer Name	✓		
	Pick Dur.	✓	0.1 seconds to 1000 hours	
	Units	✓	Hrs, Mins, Secs	
	Drop Dur.	✓	0.1 seconds to 1000 hours	
	Units	✓	Hrs, Mins, Secs	
Time Of Day Timers	Timer (Id)		1 to 10	
	Timer Name	✓		
	Time	✓	0.00 to 24.00 (format HH:MM)	
Reserved Latches			See section 2.4.5.2.	
User Latches	Latch (Id)		1 to 15000	
	Name	✓		
Set Reset Latches	Latch Id		1 to 2000	
	Output	✓		
	Set	✓		
	Reset	✓		

A.3.11 NVC232 and NVC422

Tab	Field		Comment	Notes
General	Pair (NVC)	✓	Yes or No	
Outputs	Output (Id)		1 to 64	
	Output Name	✓		
	Initial State	✓	High or Low	
Inputs	Input (Id)		1 to 48	
	Input Name	✓		
	Initial State	✓	High or Low	

A.3.12 NVC/DM (VLM Installation)

Tab	Field		Comment	Notes
General	Interface File Name	✓		
	Version			
	Status			
	Nvcdm File Status			
Inputs	Input (Id)		Not available unless "Interface File Name" entered. 1 to 480	
	Input Name	✓		
	Initial State	✓	High or Low	
Outputs	Output (Id)		Not available unless "Interface File Name" entered. 1 to 480	
	Output Name	✓		
	Initial State	✓	High or Low	

A.3.13 NVC/DM (NVLM Installation)

Tab	Field		Comment	Notes
General	Housing Nbr	✓	Defaults to 1.	
	Slot Nbr	✓	Defaults to 3.	
	Read Password	✓	Defaults to WESTRACE	
	Read/Write Password	✓	Defaults to WESTRACE	
Ports:			0 to 8 ports can be set up. Port 1 is reserved. Port 8 can only be an Interface port if set up.	
– Diagnostic				
– General	Port OK Input	✓		

Tab	Field		Comment	Notes
	Connection Type	✓	Permanent cable or modem connection. Intermittent cable connection. Hayes Modem (dial in only). Hayes Modem (dial out only). Hayes Modem (dial in and out).	
	Connected State Input	✓		
	Port Address	✓	Defaults to 1. Range 1 to 127.	
	Initialisation String	✓	Defaults to E0V1	
	Auto Answer String	✓	Defaults to S0=3	
– Serial Interface	Handshake Mode	✓	None or RTS/CTS	
	PEN Signal Output	✓	Transmit Only or Always Active	
	Carrier Defect	✓	Ignore or Used	
	Baud Rate	✓	1200, 2400, 4800, 9600, 19200, 38400, 57600, 64000.	
– Phone Numbers			0 to 5 phone numbers	
	Phone Number	✓		
	Description	✓		
– WSL S2 Slave				
– General	Port OK Input	✓		
	All Modules OK Input	✓		
	Primary Port	✓	None, 2, 3, 4, 5, 6, 7.	
	Broadcast Window	✓	Defaults to 25.	
– Serial Interface	Handshake Mode	✓	None or RTS/CTS	
	PEN Signal Output	✓	Transmit Only or Always Active	
	Carrier Detect	✓	Ignore or Used	
	Baud Rate	✓	1200, 2400, 4800, 9600, 19200, 38400, 57600, 64000.	
– Housing			Up to 63 housings (numbers 0 to 63 excluding 13)	
– General	Housings OK Input	✓		
– Housing Slot			Up to16 slots (numbers 1 to 16) Only use slots 1 to 15	
– Inputs	Input (Id)		1 to 32	

Tab			Field		Comment	Notes
			Input Name	✓		
			Initial State	✓	High or Low	
			Flash Control	✓		
			Initial State	✓	High or Low	
		Outputs	Output (Id)		1 to 32	
			Output Name	✓		
			Initial State	✓	High or Low	
WSA S2 Slave						
	General		Port OK Input	✓		
			All Modules OK Input	✓		
			Primary Port	✓	None, 2, 3, 4, 5, 6, 7.	
			Data Word Length	✓	Range: 32, 48, 64, 96, 128, 256.	
			Tx Clock Direction	✓	From Port or External	
			Rx Clock Direction	✓	From Port, External or Reconstructed.	
	Serial Interface		Handshake Mode	✓	None or RTS/CTS	
			PEN Signal Output	✓	Transmit Only or Always Active	
			Carrier Detect	✓	Ignore or Used	
			Baud Rate	✓	1200, 2400, 4800, 9600, 19200, 38400, 57600, 64000.	
	Modules				Up to 62 Slave Modules (numbers 1 to 62)	
		General	Module OK Input	✓		
		Inputs	Input (Id)		1 to 32	
			Input Name	✓		
			Initial State	✓	High or Low	
			Flash Control	✓		
			Initial State	✓	High or Low	
		Outputs	Output (Id)		1 to 32	
			Output Name	✓		
			Initial State	✓	High or Low	
WSA S2 Master						
	General		Port OK Input	✓		
			All Modules OK Input	✓		

Tab	Field		Comment	Notes
	Data Word Length	✓	Range: 32, 48, 64, 96, 128, 256.	
	Tx Clock Direction	✓	From Port or External	
	Rx Clock Direction	✓	From Port, External or Reconstructed.	
	Inter Scan Delay	✓	Defaults to 20	
Serial Interface	Handshake Mode	✓	None or RTS/CTS	
	PEN Signal Output	✓	Transmit Only or Always Active	
	Carrier Detect	✓	Ignore or Used	
	Baud Rate	✓	1200, 2400, 4800, 9600, 19200, 38400, 57600, 64000.	
Modules			Up to 62 Master Modules (numbers 1 to 62)	
General	Module OK Input	✓		
Inputs	Input (Id)		1 to 32	
	Input Name	✓		
	Initial State	✓	High or Low	
Outputs	Output (Id)		1 to 32	
	Output Name	✓		
	Initial State	✓	High or Low	
	Flash Control	✓		
	Initial State	✓	High or Low	
VLM Interface				
General	Port OK Input	✓		
	Address			
	Interface File Name	✓		
	Version			
	Status			
	VLM File Status			
Inputs			Not available unless “Interface File name” entered.	
	Input (Id)		1 to 480	
	Input Name	✓		
	Initial State	✓	High or Low	
Outputs			Not available unless “Interface File name” entered.	
	Output (Id)		1 to 480	
	Output Name	✓		

Tab	Field		Comment	Notes
	Initial State	✓	High or Low	
Reserved Timers			See section 2.4.6.3.	
User Timers	Timer Id		1 to 3000	
	Trigger Name	✓		
	Timer Name	✓		
	Pick Dur.	✓	0.1 seconds to 1000 hours	
	Units	✓	Hrs, Mins, Secs	
	Drop Dur.	✓	0.1 seconds to 1000 hours	
	Units	✓	Hrs, Mins, Secs	
Time Of Day Timers	Timer (Id)		1 to 10	
	Timer Name	✓		
	Time	✓	0.00 to 24.00 (format HH:MM)	
Reserved Latches			See section 2.4.5.3.	
User Latches	Latch (Id)		1 to 15000	
	Name	✓		
Set Reset Latches	Latch Id		1 to 2000	
	Output	✓		
	Set	✓		
	Reset	✓		

A.3.14 TCOM

Also see section 2.4.7.7.

Tab	Field		Comment	Notes
General		✓	Standard fields.	
Channels 1 & 2 Functions				
	Change Over #		# = 1 or 2	
	Long Name	✓		
	Control Code #		# = 1 to 13	
	Long Name	✓		
	Code Enable			
	Long Name	✓		
	Channel Fault			
	Long Name	✓		

A.3.15 VLM5

Tab	Field		Comment	Notes
General	Mode	✓	Hot Standby or Stand-Alone	
Reserved Timers			See section 2.4.6.1.	
User Timers	Timer (Id)		1 to 293	
	Trigger Name	✓		
	Timer Name	✓		
	Duration	✓	0 seconds to 59.0 hours	
	Units	✓	Hrs, Mins, Secs	
Reserved Latches			See section 2.4.5.1.	
User Latches	Latch (Id)		1 to 3957	
	Name	✓		

A.3.16 VLM6

Tab	Field		Comment	Notes
General	Mode	✓	Hot Standby or Stand-Alone	
Reserved Timers			See section 2.4.6.1.	
User Timers	Timer (Id)		1 to 293	
	Trigger Name	✓		
	Timer Name	✓		
	Duration	✓	0 seconds to 59.0 hours	
	Units	✓	Hrs, Mins, Secs	
Reserved Latches			See section 2.4.5.1.	
User Latches	Latch (Id)		1 to 3942	
	Name	✓		
Network Ports			1 to 16 Ports Cannot create or select an Interface file until the Installation Address is defined (on the Installation main window). There is a fixed relationship between the Network Port number and the local Port Address: Port 1 has Local Port Address 8, Port 2 has Local Port Address 9 etc, up to Port 16 has Local Port Address 23.	
– General				
– System Type	Source	✓		
	Destination	✓		
– Address	Source	✓		
	Destination	✓		
– Local Port Address	Source	✓	See Network Ports above.	
	Destination	✓		
– LOC Timeout	Source	✓	0.1 of a second. (Range 0 to 20 seconds) Do not use the default value of 0. Use 2.0 seconds if the correct value is not known.	
	Destination	✓	0.1 of a second. (Range 0 to 20 seconds) Do not use the default value of 0. Use 2.0 seconds if the correct value is not known.	
	Interface File Name	✓		
	Version			
	Status			

A.3.17 VLOMFS12, VLOMFS24, VLOMFS110, VLOMFT12, VLOMFT24, VLOMFT110

Also see section 2.4.7.3.

Tab	Field		Comment	Notes
General	Fault Name	✓		
Lamps	Output (Id)		VLOMFS12, VLOMFS24, VLOMFS110: 1 to 6 VLOMFT12, VLOMFT24, VLOMFT110: 1 to 12	
	Output Name	✓		
	Initial State	✓	High or Low	
	Filament Proving Name	✓		
	Flash Name	✓		
	Initial Flash State	✓	High or Low	
	Flash Proving Name	✓		

A.3.18 VLOMSS12, VLOMSS24, VLOMSS110, VLOMST12, VLOMST24, VLOMST110

Also see section 2.4.7.2.

Tab	Field		Comment	Notes
General	Fault Name	✓		
Lamps	Output (Id)		VLOMSS12, VLOMSS24, VLOMSS110: 1 to 6 VLOMST12, VLOMST24, VLOMST110: 1 to 12	
	Output Name	✓		
	Initial State	✓	High or Low	
	Filament Proving Name	✓		

A.3.19 VPIM12, VPIM24, VPIM50

Also see section 2.4.7.1.

Tab	Field		Comment	Notes
General	Fault Name	✓		
Inputs	Input (Id)		1 to 12	
	(Input) Name	✓		
	Initial State	✓	High or Low	

A.3.20 VROM12, VROM 24, VROM50

Also see section 2.4.7.4.

Tab	Field		Comment	Notes
General	Fault Name	✓		
Outputs	Output (Id)		1 to 8	
	Output Name	✓		
	Initial State	✓	High or Low	

A.3.21 VTC232

Also see section 2.4.7.5.

Tab	Field		Comment	Notes
General	Local Port Address	✓		
	Adjacent Installation Name	✓		
	Adjacent Installation Address	✓		
	Adjacent Port Address	✓		
Outputs	Output (Id)		1 to 17	
	Output Name	✓		
	Initial State	✓	High or Low	
Inputs	Input (Id)		1 to 17	
	Input Name	✓		
	Initial State	✓	High or Low	

A.3.22 WCM

Also see section 2.4.7.9.

Tab	Field		Comment	Notes
General	Local Port Address	✓		
Route Pairs	Buttons for route pairs	✓	02/03, 04/05, 06/07… 62/63	
	Output (Id)		1 to 4	
	Output Name	✓		
	Initial State		Low	

Appendix B: Reports

This Appendix summarises the GCSS reporting tools and directs the reader to where further information is available. Some reports are immediately directed to the defined printer and others are displayed on screen, typically in the Report Window from which they can sent to the printer.

B.1 The Report Window

This window is automatically activated when required. It remains open until closed by double-clicking the close box or by closing the installation. The current contents of the window are over written if an operation that writes to the Report Window is performed while the window is open.

The contents of the Report Window can be:

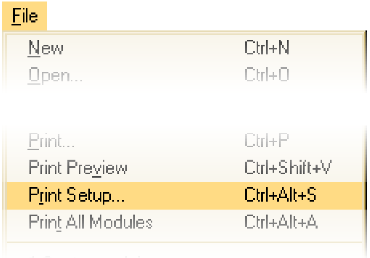
- **Print Previewed**—by selecting **Print Preview** from the File menu to display an image of the report as it would appear when printed.
- **Printed**—by sending it to the defined printer. See section B.2.
- **Written to an ASCII text file**—by selecting **Write Contents To File** from the File menu.

  This displays a standard file selector dialog box that allows you to select or enter an appropriate file name.

Note: ***The information above is also applicable to the User Report and the Supervisor Window.***

B.2 Selecting and Setting Up a Printer

a) Select Print Setup to open the Print Setup dialog box.



Alt f r
Ctrl+Alt+s

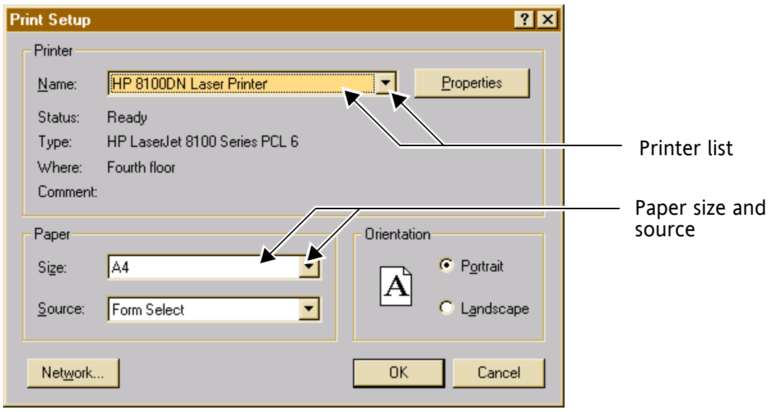


Figure B.1 Print Setup dialog box

b) Select the required printer from the drop-down printer list.

Other printers on the network (if any) can be selected via the **Network** button.

c) Change the printer properties, if needed, by selecting the **Properties** button.

d) Select the paper size and source.

e) Select the paper orientation.

B.3 Printed Report Format

The format of a printed report is usually similar to the on-screen version and each page has a header and footer (apart from the ladder logic reports which have the header and footer on the first page but only the footer on the other pages).

Headers and Footers

The report headers and footers usually include the following:

Header

- Installation name
- Installation address
- Customer
- Approver (or "Not Approved")
- Approval date (if approved)
- GCSS Version
- Source File Checksum

Footer

- Filename
- Installation Data Version
- Approval Status
- Time
- Date
- Page Number

Printing Reports

Use one of the following options to print a report:

- Select **Print** from the File menu;
- Click on the printer icon in the main toolbar;
- Use the accelerator key **Ctrl+P**;
- Select the **Print** button from the print preview display;
- Right-click on the appropriate window and select **Print** from the Context menu.

All options display the standard Microsoft Windows print dialog which allows selection of pages to be printed and if need be, changes to the printer settings.

B.4 Summary of GCSS Reports

Name	Description	Format	See Section
Installation Report	A single page containing header information and details of Installation Address Jumper Settings and Data Version Switches. Included in the header information are the three lines of comments that can be added to the Installation window.	Printed or Previewed	3.6.1 B.5.1
Installation Difference Report	Compares two installations (or two versions of the one installation) by listing all differences. Is useful for checking an installation to ensure that all intended changes were made and that unintended changes were not made.	Report Window	3.9 B.5.2
Housing Configuration Report	A single-page printout containing an illustration of the housings and modules plus the standard header and footer information. It is printed from the Housing Editor window.	Printed or Previewed	4.2.6.1 B.5.11
Module Report	Presents the information displayed by the Module Editor. Two formats are available: • Full—contains all fields on the module regardless or whether they contain information or not.; • Brief—only contains the data entered by the user. Printed from the Module Editor window.	Printed or Previewed	5.2.5 5.3.6 B.5.13
All Modules Report	Prints details of all modules in a housing automatically. Saves having to activate the Module Editor for each module. Can select between a Full or Brief report. Printed from the Housing Editor window.	Printed or Previewed	4.2.6.2 B.5.9
Ladder Editor Report	A textual list of all the rungs in the ladder showing the rung order, rung names, and their consistency. Printed from the Ladder Editor window.	Printed or Previewed	6.2.4 B.5.12
Rung Viewer Report	The Rung Viewer Report consists of a title page followed by a graphical representation of all rungs in the ladder. Printed from the Rung Viewer window.	Printed or Previewed	6.4.4 B.5.10
Consistency Check Report	When the consistency check is activated, the report window for the installation is opened and any inconsistencies will then be displayed in the report window. The report also contains the date, time and a statement of whether the installation passed or failed the checks. Printed from the Installation menu or toolbar.	Report Window	7.2.2 B.5.14
Compilation Report	When the compilation is activated, the report window for the installation is opened to show a record of the compilation process. The report also contains the date, time and a statement of whether the installation passed or failed the compilation. Printed from the Installation menu or toolbar.	Report Window	7.3 B.5.15
Approval Report	Is generated automatically whenever an installation is approved. The Approval report contains a single checksum value which is a checksum of all the information in the installation that is compiled and stored in the WESTRACE Equipment EPROMs.	Report Window	7.5 B.5.17

Table B.1 GCSS Reports—Summary

Name	Description	Format	See Section
VLM PROM Programming Report	Contains predicted checksums of the data sent to the PROM Programmer for both high and low byte PROMs. These checksums should be compared to actual checksums produced by the PROM Programmer. Is generated after a VLM installation is approved and compiled. Printed from the PROM Download dialog box. **This report must be printed, completed and retained.**	Printed or Previewed	B.5.5
NVLM Non-Vital Configuration Programming Report	Is generated after a NCDM or NVC/DM installation is approved and compiled. Printed from the CED Download dialog box. **This report must be printed, completed and retained.**	Printed or Previewed	B.5.7
Data Comparison Report	Compares two installations for differences in their respective downloaded data fields (only the information that can be reproduced from the PROMs upload is compared). The main use of the report is to compare an uploaded installation with the original source used to create the installation. In the case of a VLM, the installation is uploaded from the PROM Programmer rather than from the module itself.	Printed or Previewed	8.3.3 B.5.16 B.5.6 B.5.8
Mnemonic Usage Report	Lists all the names in an installation and where they are defined and used in alphabetical order. A search pattern can be entered for the report so that only names corresponding to the pattern are listed. This can be used to effectively search for a single name or group of names.	Report Window	3.7 B.5.3
PROM Usage Report	Accessible when a VLM installation is selected. Provides an estimate of the PROM space that will be used by the installation. Can be run on installations that are not yet ready to be compiled. Usage percentages above 100% indicate the installation is too large to fit the PROMS.	Report Window	ICS
CED Usage Report	Accessible when an NVLM installation is selected. Is similar to the PROM Usage report but the "Bytes Used" information is only available after the installation is compiled. An NVLM installation must be recompiled after approval to generate the required information.	Report Window	ICS
User Report	Shows the name of the current password file and all users recorded in that file.	Report Window	10.5.6 B.5.18

Table B.1 GCSS Reports—Summary *(Continued)*

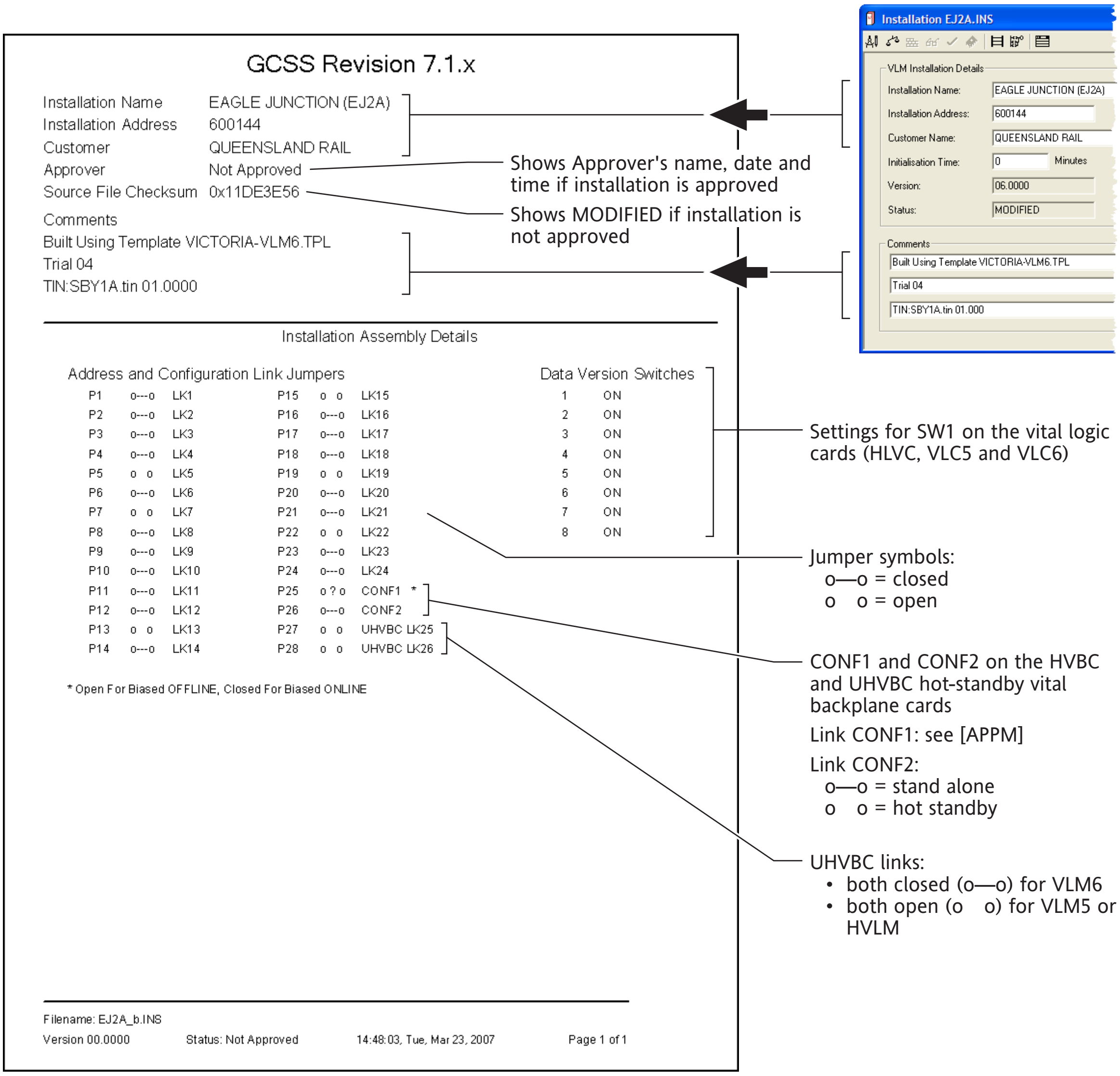
B.5 Example Reports

B.5.1 Installation Report

This is a single printed page that contains:

- header information and comments entered through the installation window;
- a list of installation Address and Configuration Link Jumper settings;
- a list of Data Version Switches.

See section 3.6.1 for generating the report, and reference [APPM] for information on address jumpers and version switches.



GCSS Revision 7.1.x

Installation Name EAGLE JUNCTION (EJ2A)
Installation Address 600144
Customer QUEENSLAND RAIL
Approver Not Approved
Source File Checksum 0x11DE3E56

Comments
Built Using Template VICTORIA-VLM6.TPL
Trial 04
TIN:SBY1A.tin 01.0000

Installation Assembly Details

Address and Configuration Link Jumpers

P1	o---o	LK1	P15	o o	LK15			
P2	o---o	LK2	P16	o---o	LK16			
P3	o---o	LK3	P17	o---o	LK17			
P4	o---o	LK4	P18	o---o	LK18			
P5	o o	LK5	P19	o o	LK19			
P6	o---o	LK6	P20	o---o	LK20			
P7	o o	LK7	P21	o---o	LK21			
P8	o---o	LK8	P22	o o	LK22			
P9	o---o	LK9	P23	o---o	LK23			
P10	o---o	LK10	P24	o---o	LK24			
P11	o---o	LK11	P25	o ? o	CONF1	*		
P12	o---o	LK12	P26	o---o	CONF2			
P13	o o	LK13	P27	o o	UHVBC LK25			
P14	o---o	LK14	P28	o o	UHVBC LK26			

Data Version Switches

1	ON
2	ON
3	ON
4	ON
5	ON
6	ON
7	ON
8	ON

* Open For Biased OFFLINE, Closed For Biased ONLINE

Filename: EJ2A_b.INS
Version 00.0000 Status: Not Approved 14:48:03, Tue, Mar 23, 2007 Page 1 of 1

Figure B.2 Installation Assembly Details Report—example

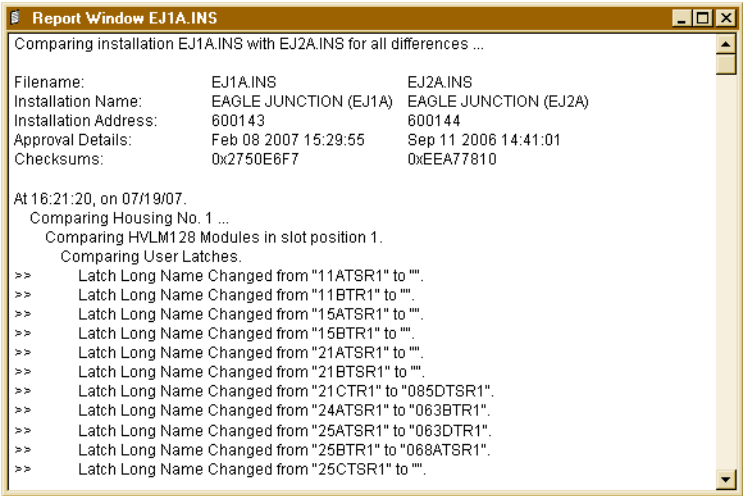
B.5.2 Difference Report

The Difference Report is written to the Report Window where it can be viewed, print-previewed or printed. See section 3.9 for generating the report.

It compares two installations (or two versions of the one installation) by listing all differences, which are marked with >> at the left edge of the report. The report is useful for checking that all intended changes to an installation were made and that unintended changes were not made.

See section 3.9, "Comparing Installations (Difference Report)".

Do not use this report if one of the installations has been re-created by the upload and decompile process (section 8.3)—use the Compare PROM Data or Compare NVC/DM Upload Report instead (section B.5.16).



```
Report Window EJ1A.INS
Comparing installation EJ1A.INS with EJ2A.INS for all differences ...

Filename:              EJ1A.INS              EJ2A.INS
Installation Name:     EAGLE JUNCTION (EJ1A) EAGLE JUNCTION (EJ2A)
Installation Address:  600143                600144
Approval Details:      Feb 08 2007 15:29:55  Sep 11 2006 14:41:01
Checksums:             0x2750E6F7            0xEEA77810

At 16:21:20, on 07/19/07.
   Comparing Housing No. 1 ...
      Comparing HVLM128 Modules in slot position 1.
         Comparing User Latches.
>>          Latch Long Name Changed from "11ATSR1" to "".
>>          Latch Long Name Changed from "11BTR1" to "".
>>          Latch Long Name Changed from "15ATSR1" to "".
>>          Latch Long Name Changed from "15BTR1" to "".
>>          Latch Long Name Changed from "21ATSR1" to "".
>>          Latch Long Name Changed from "21BTSR1" to "".
>>          Latch Long Name Changed from "21CTR1" to "085DTSR1".
>>          Latch Long Name Changed from "24ATSR1" to "063BTR1".
>>          Latch Long Name Changed from "25ATSR1" to "063DTR1".
>>          Latch Long Name Changed from "25BTR1" to "068ATSR1".
>>          Latch Long Name Changed from "25CTSR1" to "".
```

Figure B.3 Installation Differences Report—example

Note:

1. Only the first difference in rungs is reported.

2. The Difference Report command is only available when two or more installation windows are open.

B.5.3 Mnemonic Usage Report

The report is written to the Report Window where it can be viewed, print-previewed or printed. See section 3.7 for generating the report.

This report lists, in alphabetical order, all the names in an installation and the locations where they are defined and used. It is particularly useful when an existing interlocking is being altered. The Signal Engineer can search for all instances of a particular logic function to see if there are any unintended consequences of changes.

Report Window EJ2A.INS

MNEMONIC USAGE REPORT

Long Name:	"(AT)19(1M)GNSR"
Defined in:	Module VPIM50, Housing 3, Slot 9, Seq Nbr 19; Input 8
	Module EVTC, Housing 1, Slot 4, Seq Nbr 1; Output 57
Long Name:	"(AT)ATPR"
Defined in:	Module EVTC, Housing 1, Slot 4, Seq Nbr 1; Output 38
Used in:	Rung 79, "(AT)ATPR"; as Coil
Long Name:	"(AT)ATR"
Defined in:	Module VPIM50, Housing 3, Slot 11, Seq Nbr 20; Input 1
Used in:	Rung 79, "(AT)ATPR"; as Normally Open Contact
	Rung 80, "(AT)ATR1"; as Normally Open Contact
Long Name:	"(AT)ATR1"
Defined in:	Module HVLM128, Housing 1, Slot 1, Seq Nbr 0; User Latch 35
Used in:	Rung 80, "(AT)ATR1"; as Coil
	Rung 79, "(AT)ATPR"; as Normally Open Contact

Figure B.4 Mnemonic Usage Report with default search pattern—example

B.5.4 PROM or Non-Vital Configuration Usage Report

The report is written to the Report Window where it can be viewed, print-previewed or printed. See section 3.8 for generating the report.

Report Window South Melbourne.ins

CED Usage Report   Logic Module Type: VLM6

Capacities

	Used	Available	Free	%
All Names	61	4000	3939	1.5
Timers	7	300	293	2.3
I/O Space	124 bytes	16648 bytes	16524 bytes	0.7
Logic Space	10 bytes	78328 bytes	78318 bytes	0.0

Other Measurements

44	Latches Used
2	Rungs Used
1	Contacts Used

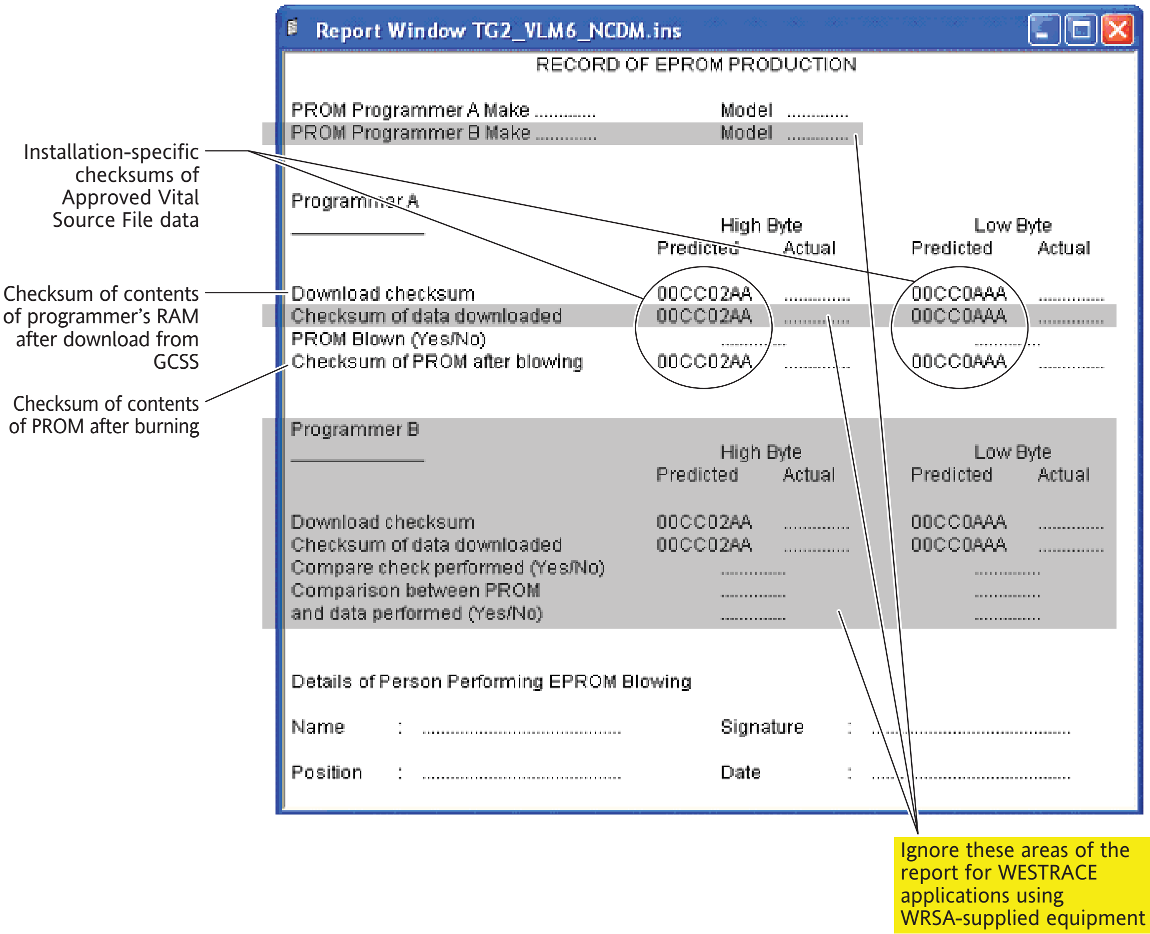
I/O Breakdown by module

Order In PROM	Sequence Number	Housing	Slot	Module Type	Input Names	Output Names	Bytes Used
1	0	1	1	VLM6	0	0	4
2	1	1	3	NCDM	1	1	40
3	2	1	5	VLOMSS12	0	1	36
4	3	1	8	VLOMFT12	0	1	36

Figure B.5 PROM or Non-Vital Configuration Usage Report—example

B.5.5 PROM Download Report (PROM Programming Report or Record of EPROM Production)

The report is written to the Report Window where it can be viewed, print-previewed or printed. See sections 8.1 and 8.2 for generating the report.



RECORD OF EPROM PRODUCTION

PROM Programmer A Make .... Model ....
PROM Programmer B Make .... Model ....

Programmer A

	High Byte Predicted	High Byte Actual	Low Byte Predicted	Low Byte Actual
Download checksum	00CC02AA	....	00CC0AAA	....
Checksum of data downloaded	00CC02AA	....	00CC0AAA	....
PROM Blown (Yes/No)	....		....	
Checksum of PROM after blowing	00CC02AA	....	00CC0AAA	....

Programmer B

	High Byte Predicted	High Byte Actual	Low Byte Predicted	Low Byte Actual
Download checksum	00CC02AA	....	00CC0AAA	....
Checksum of data downloaded	00CC02AA	....	00CC0AAA	....
Compare check performed (Yes/No)	....		....	
Comparison between PROM and data performed (Yes/No)	....		....	

Details of Person Performing EPROM Blowing

Name : .... Signature : ....

Position : .... Date : ....

Figure B.6 PROM Download Report (PROM Programming Report or Record of EPROM Production)—example

B.5.6 PROM Upload Report

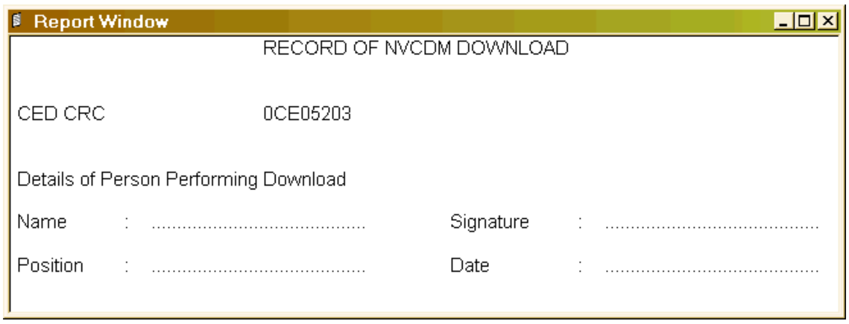
The report is written to the Report Window where it can be viewed, print-previewed or printed.

```
Report Window
START DECOMPILATION...
  08/01/07 18:00:47

Verifying prom checksums...
.. Checksums OK
Decompiling Comment Data...
.. Comment Data OK
Decompiling Mnemonics...
...Mnemonics Decompilation : OK!
Decompiling System Configuration data...
.. System Configuration data OK
Decompiling Initialisation data...
.. Initialisation data OK
Decompiling Timers...
.. Timers OK
Decompiling Modules...
   Decompiled a CNVC module in Housing 1, Slot 6
   Decompiled a CNVC module in Housing 1, Slot 7
   Decompiled a CCM module in Housing 1, Slot 8
   Decompiled a VPIM module in Housing 1, Slot 9
   Decompiled a VPIM module in Housing 1, Slot 11
   Decompiled a VPIM module in Housing 1, Slot 13
   Decompiled a VLOMFT module in Housing 2, Slot 2
   Decompiled a VLOMFT module in Housing 2, Slot 6
   Decompiled a VROM module in Housing 2, Slot 9
   Decompiled a VPIM module in Housing 2, Slot 11
   Decompiled a VPIM module in Housing 2, Slot 13
   Decompiled a VLOMFT module in Housing 3, Slot 2
   Decompiled a VROM module in Housing 3, Slot 5
   Decompiled a VROM module in Housing 3, Slot 7
   Decompiled a VROM module in Housing 3, Slot 9
   Decompiled a VPIM module in Housing 3, Slot 11
   Decompiled a VPIM module in Housing 3, Slot 13
   Decompiled a CNVC module in Housing 1, Slot 3
   Decompiled a CNVC module in Housing 1, Slot 4
.. Modules OK
Decompiling Ladder Logic...
.. v1
.. v2
.. v3
.. v4
.. v5
.. v6
.. v7
.. v8
.. v9
.. v10

.. v464
.. v465
.. v466
.. v467
.. v468
.. v469
.. v470
.. v471
.. v472
.. vpim1
.. vpim2
.. vpim3
.. vpim4
.. vpim5
.. vpim6
.. vpim7
.. vpim8
.. COMMS_OK
.. Ladder Logic OK
Checking for User Latches...
.. User Latches OK
Checking for NVC/DM...
  Creating NVC/DM...
  Copying Inputs/Outputs...
NVC/DM Ok...
Installation Decompiled successfully.

END DECOMPILATION
  08/01/07 18:00:54
```

Figure B.7 PROM Upload Report—example

B.5.7 Non-Vital Configuration Download Report

The report is written to the Report Window where it can be viewed, print-previewed or printed.


Report Window

RECORD OF NVCDM DOWNLOAD

CED CRC 0CE05203

Details of Person Performing Download

Name : .... Signature : ....

Position : .... Date : ....

Figure B.8 Non-Vital Configuration Download Report—example

B.5.8 Non-Vital Configuration Upload Report

The report is written to the Report Window where it can be viewed, print-previewed or printed.

```
Report Window
START DECOMPILATION...
  08/01/07 18:00:54

Checking CRC ...
...CRC Checking : OK!
Decompiling Pointer Data...
.. Pointer decompilation : OK!
Decompiling Comment Data...
...Comment Data Decompilation : OK!
Decompiling System Configuration Data...
...Configuration Data Decompilation: OK!
Decompiling Mnemonics...
...Mnemonics Decompilation : OK!
Decompiling Application Logic Data...
.. vrom1_1
.. vrom1_2

.. vrom1_79
.. vrom1_80
.. vrom11
.. vrom12
.. vrom13
.. nvrom1
.. nvrom2
.. RLatch
.. SLatch1
.. SLatch2

.. SLatch9
.. SLatch10
.. nvrom3
.. Trigger2401
.. Trigger2402

.. Trigger2799
.. Trigger2800
.. TriggerGen1
.. TriggerGen2
.. ResetLatch
.. SLatchVROM4
.. nvrom4
.. RLatchVROM4
.. TrigVROM4Timer
.. SLatch11
.. SLatch12

.. SLatch1899
.. SLatch1900
.. Port41
.. Port42

.. Port48
.. Port49
.. nvrom5
.. nv1
.. nv2

.. nv471
.. nv472
.. Vital1
.. Vital2

.. Vital15
.. Vital16
.. nvrom6
.. nvrom7_1
.. Cycle3_1
.. Cycle2_1
.. Cycle1_1
.. p6m1_s12_out1
.. p6m1_s12_out2

.. p6m1_s12_out29
.. p6m1_s12_out30
.. p6m1_s14_out1
.. p6m1_s14_out2

.. p6m1_s14_out29
.. p6m1_s14_out30
.. p6m1_s16_out1
.. p6m1_s16_out2

.. p6m1_s16_out29
.. p6m1_s16_out30
.. p6m1_s18_out1
.. p6m1_s18_out2

.. p5m18_out249
.. p5m18_out250
.. p5m19_out1
.. p5m19_out2

.. p5m19_out249
.. p5m19_out250
.. p4m20_out1
.. p4m20_out2

.. p4m20_out249
.. p4m20_out250
.. p4m21_out1
.. p4m21_out2

.. p4m21_out249
.. p4m21_out250
.. p4m22_out1
.. p4m22_out2

.. p4m22_out249
.. p4m22_out250
.. p4m23_out1
.. p4m23_out2

.. p4m23_out249
.. p4m23_out250
.. p5m20_out1
.. p5m20_out2

.. p5m20_out249
.. p5m20_out250
.. p5m21_out1
.. p5m21_out2

.. p5m21_out249
.. p5m21_out250
.. p5m22_out1
.. p5m22_out2

.. p5m22_out249
.. p5m22_out250
.. p5m23_out1
.. p5m23_out2

.. p5m23_out249
.. p5m23_out250
.. p4m24_out1
.. p4m24_out2

.. p4m24_out191
.. p4m24_out192
.. p4m26
.. p4m28
.. p5m24_out1
.. p5m24_out2

.. p5m24_out190
.. p5m24_out191
.. p4m30
.. TimerMin_1

.. TimerMin_4
.. TimerMin
.. TimerHour_1

.. TimerHour_4
.. TimerHour
.. SLatch1901
.. SLatch1902

.. SLatch1998
.. SLatch1999
.. nvrom8
.. Cycle3_3
.. Cycle2_3
.. Cycle1_3
.. nvrom7
.. nvrom7_3
.. Trig2801
.. Trig2802

.. Trig2997
...Application Logic Data Decompilation :
Installation Decompiled successfully.
```

Figure B.9 Non-Vital Configuration Upload Report—example

B.5.9 All Modules Report

This multi-page report is available when the housing editor is active. It contains details of *all* modules in an installation.

To generate the report, see section 4.2.6.2.

GCSS Revision 7.1.x

Installation Name	EAGLE JUNCTION (EJ2A)
Installation Address	600144
Customer	QUEENSLAND RAIL
Approver	Not Approved
Source File Checksum	MODIFIED

Module Type: EVTC, Housing: 1, Slot: 5

Module Type:	EVTC
Address:	00A1
Housing Nbr:	1
Slot Nbr:	5
Module Seq. Nbr:	2
Local Port Address:	1
Adjacent Installation Name:	EAGLE J
Adjacent Installation Address:	600143
Adjacent Port Address:	1

Filename: EJ2A.INS
Version 06.0000 Status: Not App

GCSS Revision 7.1.x

Installation Name	EAGLE JUNCTION (EJ2A)
Installation Address	600144
Customer	QUEENSLAND RAIL
Approver	Not Approved
Source File Checksum	MODIFIED

Module Type: EVTC, Housing: 1, Slot: 4

Output	Output Name	Initial State
1	063DTCR	Low
2	068CTSCR	Low
3	068DTCR	Low
4	(AT)DTSCR	Low
5	(DN)AATSCR	Low
6	082ATSCR	Low
7	085DTSPR	Low
8	063BTPR	Low
9		Low
10	063DTPR	Low
11	068ATSPR	Low
12		Low
13	068CTSPR1	Low
14	068DTPR	Low
15	077ATSPR	Low
16	077BTPR	Low
17	078ATSPR	Low
18	078BTPR	Low
19	078CTPR	Low
20		Low
21		Low
22	079CTSPR	Low
23	079DTPR	Low
24	081CTSPR	Low
25	082ATSPR	Low
26	082BTPR	Low
27	084CTSPR	Low
28	084DTPR	Low
29	085ETPR	Low
30	088CTSPR	Low

Filename: EJ2A.INS
Version 06.0000 Status: Not Approved 17:34:26, Wed, Jun 22, 2007 Page 2 of 7

Figure B.10 All Modules Report—typical—selected pages

B.5.10 Rung Viewer Report

See section 6.4.4 for generating the report.

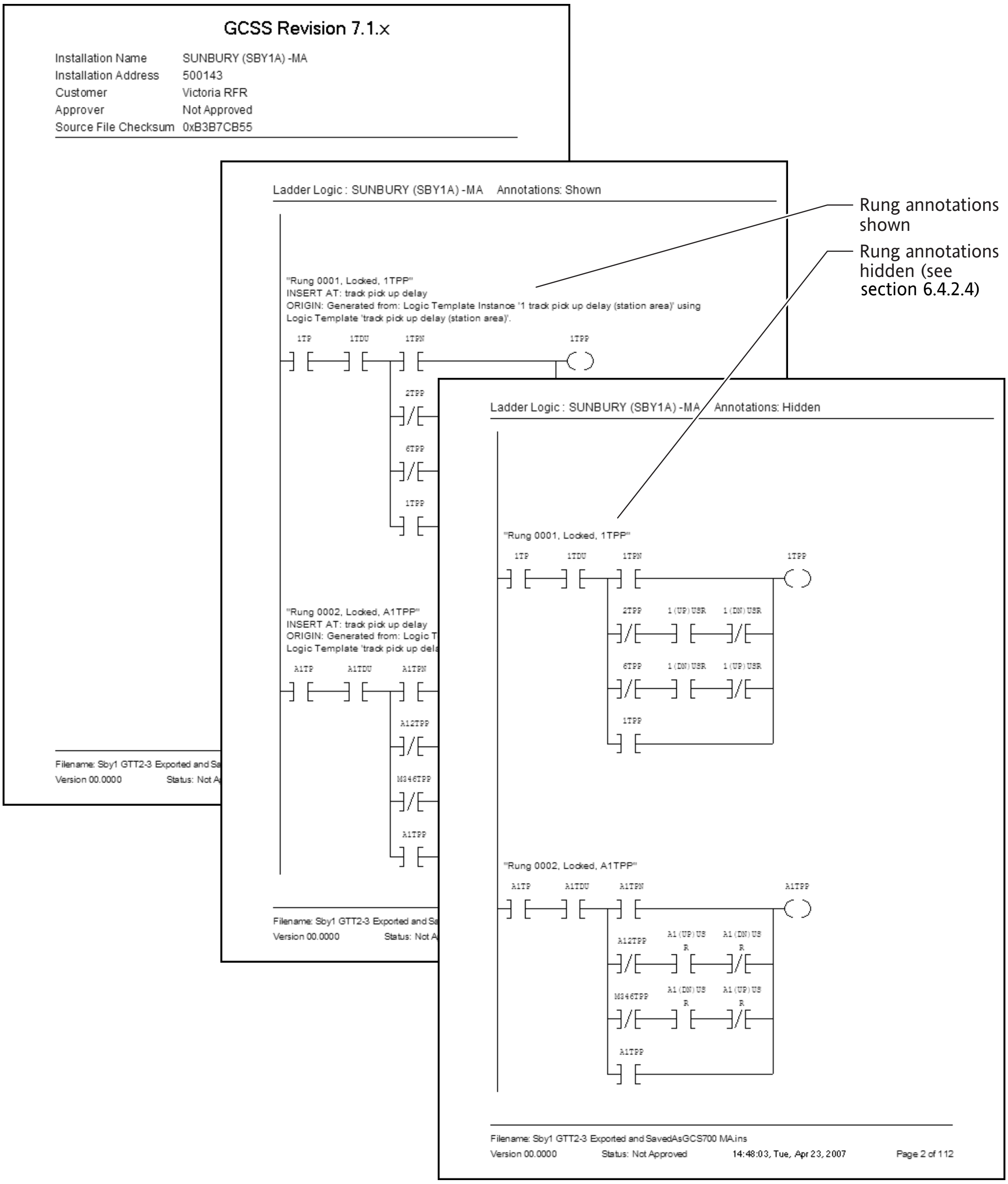


Figure B.11 Rung Viewer Report—example

Hide or show the rung annotations (notes) as described in section 6.4.2.4.

In the Rung Viewer Report, locked rungs (see section 6.1.2) are identified by the word “Locked” after the rung index number.

B.5.11 Housing Editor Report

This report is a single-page printout containing the standard header and footer information and an image of the housings and modules. The image is a representation of the housing view on the screen and shows modules occupy housing slots.

Housings without modules are “unused” and are not printed in certain circumstances.

For example:

- If the first three housings contain modules but housing four is empty, then housing four is not printed.
- If housings one and three contain modules but housings two and four are empty, then housings one, two, and three are printed.

See section 4.2.6.1 for generating the report.

GCSS Revision 7.1.x

Installation Name	EAGLE JUNCTION (EJ1A)
Installation Address	600143
Customer	QUEENSLAND RAIL
Approver	STEVEN BARKER, Feb 08, 2007 15:29:55
Source File Checksum	0x2750E6F7

Installation Assembly Details

Slot

Housing	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
1	DIAG	NVC 422 7	NVC 422 6				EVTC 32		EVTC 4	EVTC 3	EVTC 2	EVTC 1		HVLM 128	
2	NVC 422 8	VROM 50 17		VROM 50 16		VROM 50 15		VROM 50 14		VROM 50 13		VROM 50 12		VROM 50 11	
3	NVC 422 9	VPIM 50 24		VPIM 50 23		VPIM 50 22		VPIM 50 21		VPIM 50 20					
4	NVC 422 10	VPIM 50 31		VPIM 50 30		VPIM 50 29		VPIM 50 28		VPIM 50 27		VPIM 50 26		VPIM 50 25	

Unused Slots require Blanker Plates.

Filename: EJ1A.INS
Version 08.0000 Status: Approved 14:48:03, Tue, Mar 23, 2007 Page 1 of 1

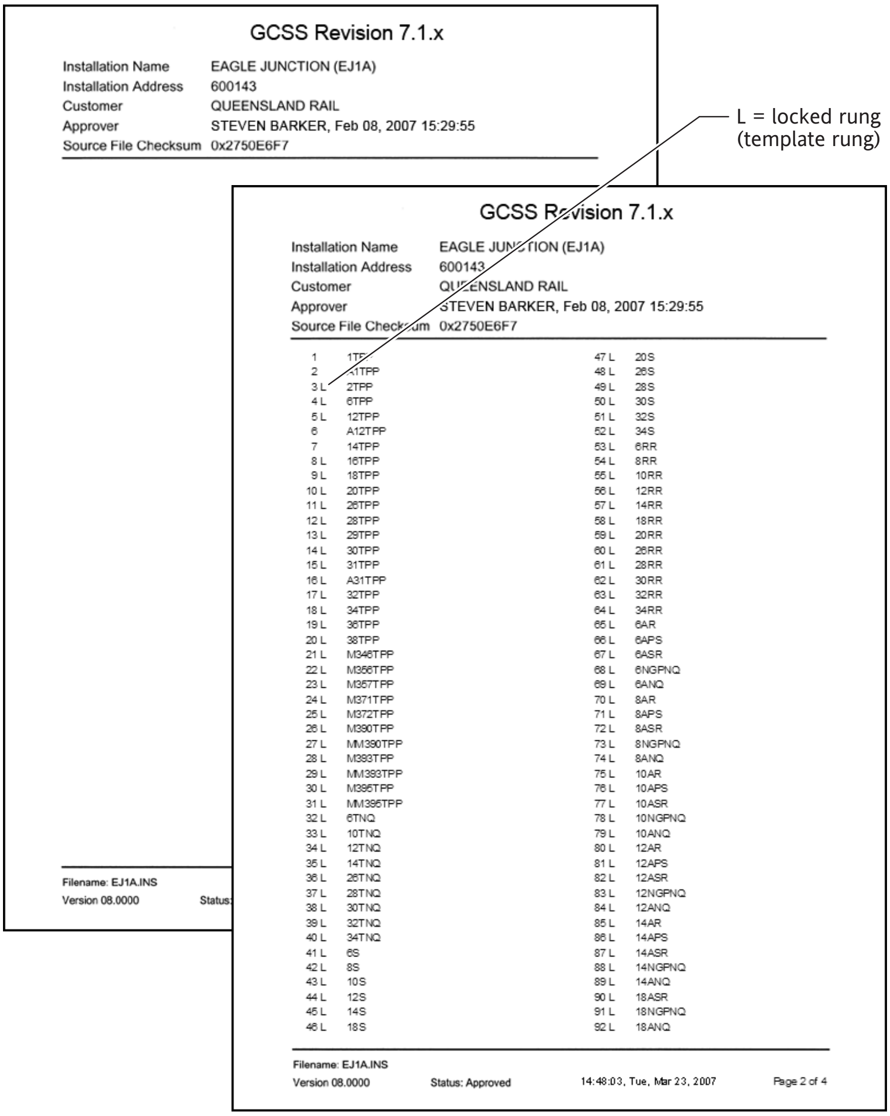
Figure B.12 Housing Report—example

B.5.12 Ladder Editor Report

This printed report replicates the layout of the Module Editor window. If any list box is too wide to fit on a single page, it is continued on subsequent pages. In these circumstances, the column containing the row number is reproduced on each page.

Locked rungs (see section 6.1.2) are identified by the letter “L” after the rung index number.

See section 6.2.4 for generating the report.



GCSS Revision 7.1.x

Installation Name	EAGLE JUNCTION (EJ1A)
Installation Address	600143
Customer	QUEENSLAND RAIL
Approver	STEVEN BARKER, Feb 08, 2007 15:29:55
Source File Checksum	0x2750E6F7

Filename: EJ1A.INS
Version 08.0000 Status:

GCSS Revision 7.1.x

Installation Name	EAGLE JUNCTION (EJ1A)
Installation Address	600143
Customer	QUEENSLAND RAIL
Approver	STEVEN BARKER, Feb 08, 2007 15:29:55
Source File Checksum	0x2750E6F7

1	1TPP	47 L	20S
2	A1TPP	48 L	26S
3 L	2TPP	49 L	28S
4 L	6TPP	50 L	30S
5 L	12TPP	51 L	32S
6	A12TPP	52 L	34S
7	14TPP	53 L	6RR
8 L	16TPP	54 L	8RR
9 L	18TPP	55 L	10RR
10 L	20TPP	56 L	12RR
11 L	26TPP	57 L	14RR
12 L	28TPP	58 L	18RR
13 L	29TPP	59 L	20RR
14 L	30TPP	60 L	26RR
15 L	31TPP	61 L	28RR
16 L	A31TPP	62 L	30RR
17 L	32TPP	63 L	32RR
18 L	34TPP	64 L	34RR
19 L	36TPP	65 L	6AR
20 L	38TPP	66 L	6APS
21 L	M346TPP	67 L	6ASR
22 L	M356TPP	68 L	6NGPNQ
23 L	M357TPP	69 L	6ANQ
24 L	M371TPP	70 L	8AR
25 L	M372TPP	71 L	8APS
26 L	M390TPP	72 L	8ASR
27 L	MM390TPP	73 L	8NGPNQ
28 L	M393TPP	74 L	8ANQ
29 L	MM393TPP	75 L	10AR
30 L	M395TPP	76 L	10APS
31 L	MM395TPP	77 L	10ASR
32 L	6TNQ	78 L	10NGPNQ
33 L	10TNQ	79 L	10ANQ
34 L	12TNQ	80 L	12AR
35 L	14TNQ	81 L	12APS
36 L	26TNQ	82 L	12ASR
37 L	28TNQ	83 L	12NGPNQ
38 L	30TNQ	84 L	12ANQ
39 L	32TNQ	85 L	14AR
40 L	34TNQ	86 L	14APS
41 L	6S	87 L	14ASR
42 L	8S	88 L	14NGPNQ
43 L	10S	89 L	14ANQ
44 L	12S	90 L	18ASR
45 L	14S	91 L	18NGPNQ
46 L	18S	92 L	18ANQ

Filename: EJ1A.INS
Version 08.0000 Status: Approved 14:48:03, Tue, Mar 23, 2007 Page 2 of 4

Figure B.13 Ladder Editor Report—example

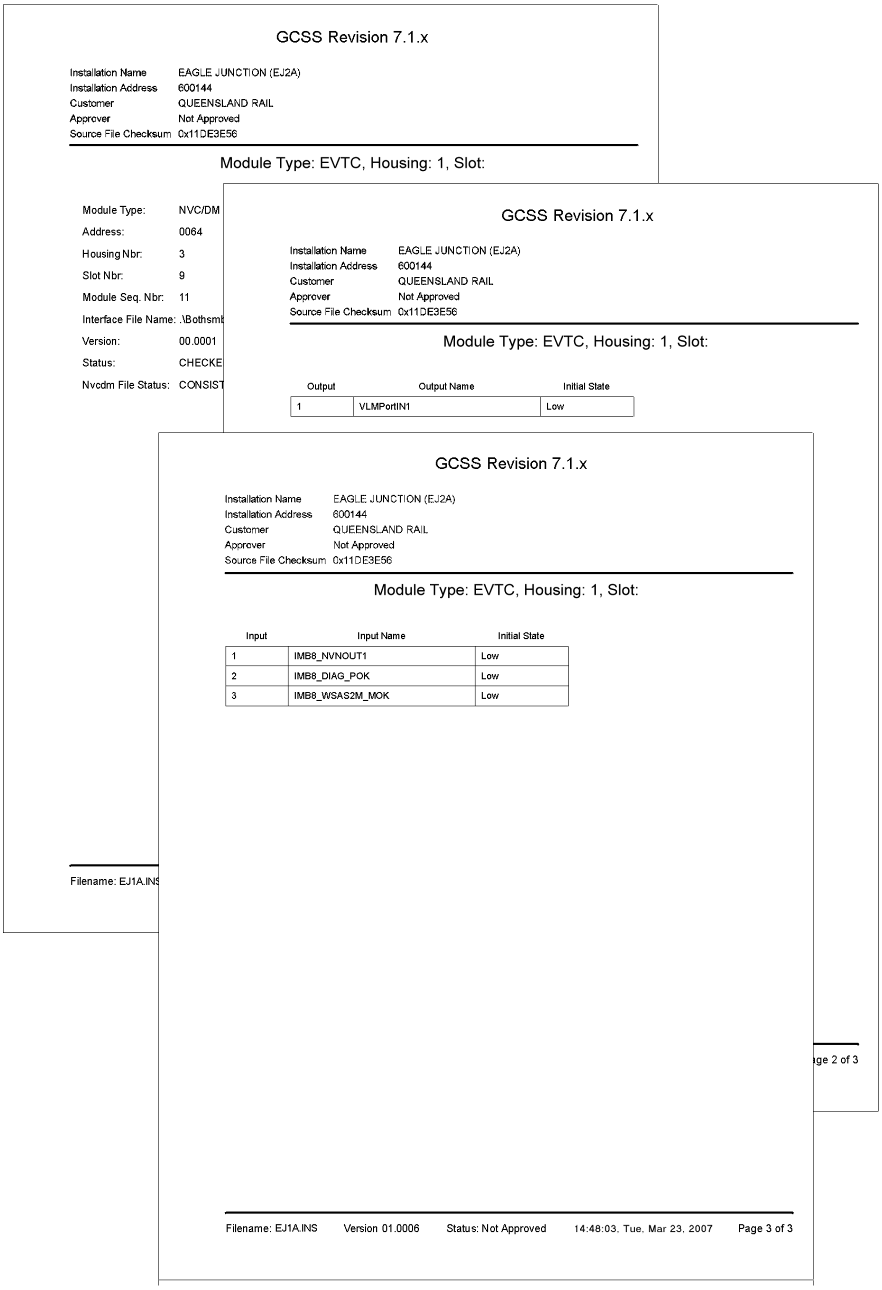
B.5.13 Module Editor Report

This printed report replicates the layout of the Module Editor window. If any list box is too wide to fit on a single page, it is continued on subsequent pages. In these circumstances, the column containing the row number is reproduced on each page.

There are two variants:

- when the Module Editor is invoked from the Housing Editor;
- when the Module Editor is invoked from an NVLM installation window.

The examples in figures B.14 and B.15 were generated for the same NVC/DM module in matching VLM and NVLM installations.



GCSS Revision 7.1.x

Installation Name	EAGLE JUNCTION (EJ2A)
Installation Address	600144
Customer	QUEENSLAND RAIL
Approver	Not Approved
Source File Checksum	0x11DE3E56

Module Type: EVTC, Housing: 1, Slot:

Module Type:	NVC/DM
Address:	0064
Housing Nbr:	3
Slot Nbr:	9
Module Seq. Nbr:	11
Interface File Name:	.\Bothsml
Version:	00.0001
Status:	CHECKE
Nvcdm File Status:	CONSIST

Filename: EJ1A.INS

GCSS Revision 7.1.x

Installation Name	EAGLE JUNCTION (EJ2A)
Installation Address	600144
Customer	QUEENSLAND RAIL
Approver	Not Approved
Source File Checksum	0x11DE3E56

Module Type: EVTC, Housing: 1, Slot:

Output	Output Name	Initial State
1	VLMPortIN1	Low

ge 2 of 3

GCSS Revision 7.1.x

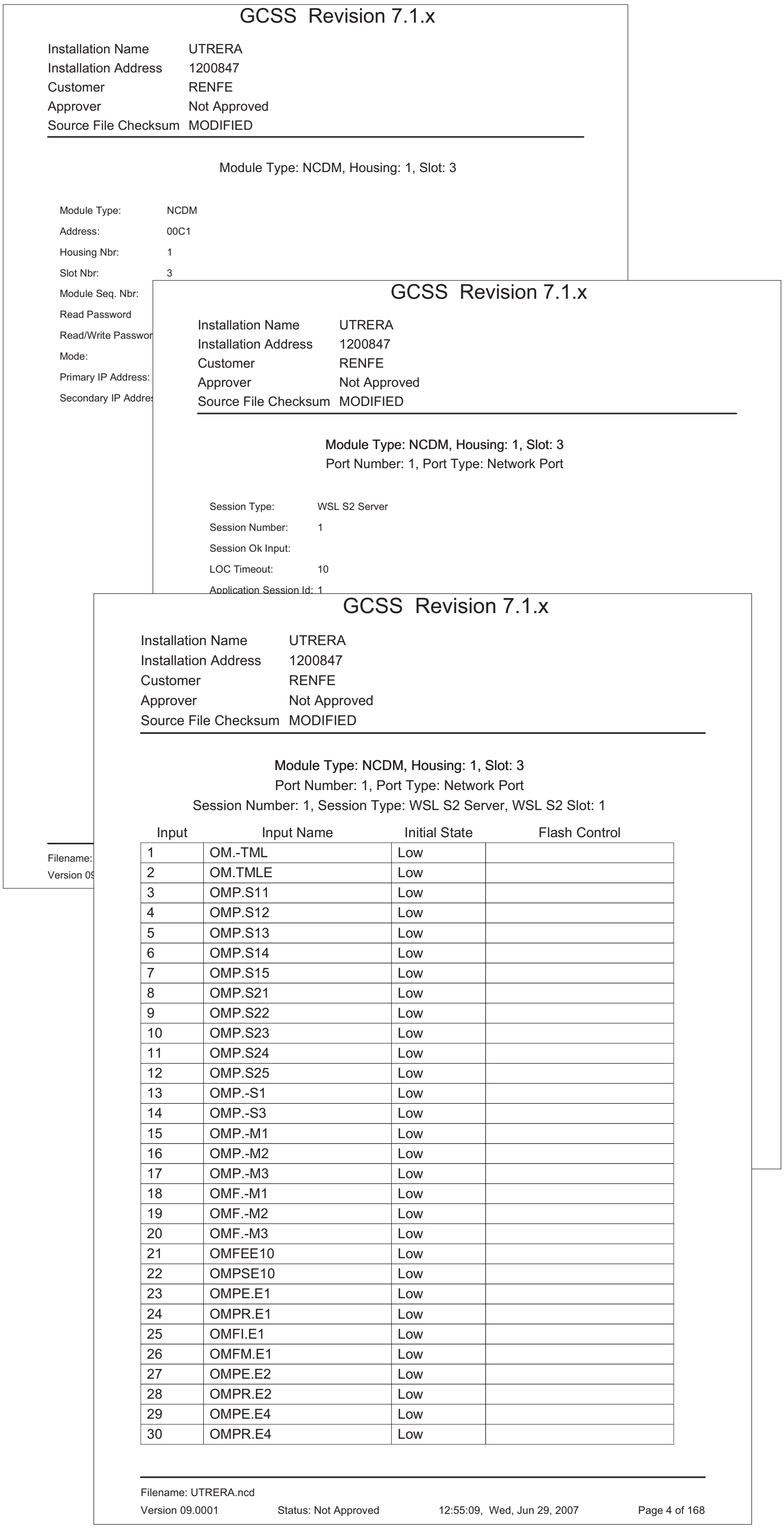
Installation Name	EAGLE JUNCTION (EJ2A)
Installation Address	600144
Customer	QUEENSLAND RAIL
Approver	Not Approved
Source File Checksum	0x11DE3E56

Module Type: EVTC, Housing: 1, Slot:

Input	Input Name	Initial State
1	IMB8_NVNOUT1	Low
2	IMB8_DIAG_POK	Low
3	IMB8_WSAS2M_MOK	Low

Filename: EJ1A.INS   Version 01.0006   Status: Not Approved   14:48:03, Tue, Mar 23, 2007   Page 3 of 3

Figure B.14 Module Editor Report—typical—editor invoked from Housing Editor



GCSS Revision 7.1.x

Installation Name	UTRERA
Installation Address	1200847
Customer	RENFE
Approver	Not Approved
Source File Checksum	MODIFIED

Module Type: NCDM, Housing: 1, Slot: 3

Module Type:	NCDM
Address:	00C1
Housing Nbr:	1
Slot Nbr:	3
Module Seq. Nbr:	
Read Password	
Read/Write Passwor	
Mode:	
Primary IP Address:	
Secondary IP Addre	

Filename:
Version 0

GCSS Revision 7.1.x

Installation Name	UTRERA
Installation Address	1200847
Customer	RENFE
Approver	Not Approved
Source File Checksum	MODIFIED

Module Type: NCDM, Housing: 1, Slot: 3
Port Number: 1, Port Type: Network Port

Session Type:	WSL S2 Server
Session Number:	1
Session Ok Input:	
LOC Timeout:	10
Application Session Id:	1

GCSS Revision 7.1.x

Installation Name	UTRERA
Installation Address	1200847
Customer	RENFE
Approver	Not Approved
Source File Checksum	MODIFIED

Module Type: NCDM, Housing: 1, Slot: 3
Port Number: 1, Port Type: Network Port
Session Number: 1, Session Type: WSL S2 Server, WSL S2 Slot: 1

Input	Input Name	Initial State	Flash Control
1	OM.-TML	Low	
2	OM.TMLE	Low	
3	OMP.S11	Low	
4	OMP.S12	Low	
5	OMP.S13	Low	
6	OMP.S14	Low	
7	OMP.S15	Low	
8	OMP.S21	Low	
9	OMP.S22	Low	
10	OMP.S23	Low	
11	OMP.S24	Low	
12	OMP.S25	Low	
13	OMP.-S1	Low	
14	OMP.-S3	Low	
15	OMP.-M1	Low	
16	OMP.-M2	Low	
17	OMP.-M3	Low	
18	OMF.-M1	Low	
19	OMF.-M2	Low	
20	OMF.-M3	Low	
21	OMFEE10	Low	
22	OMPSE10	Low	
23	OMPE.E1	Low	
24	OMPR.E1	Low	
25	OMFI.E1	Low	
26	OMFM.E1	Low	
27	OMPE.E2	Low	
28	OMPR.E2	Low	
29	OMPE.E4	Low	
30	OMPR.E4	Low	

Filename: UTRERA.ncd
Version 09.0001 Status: Not Approved 12:55:09, Wed, Jun 29, 2007 Page 4 of 168

Figure B.15 Module Editor Report—typical—editor invoked from NVLM installation window

B.5.14 Consistency Check Report

The Report window for the installation is opened when a consistency check is activated. Progress of the consistency check and any inconsistencies are displayed in the Report window. The report also shows the date, time and a statement of whether the installation passed or failed the checks.

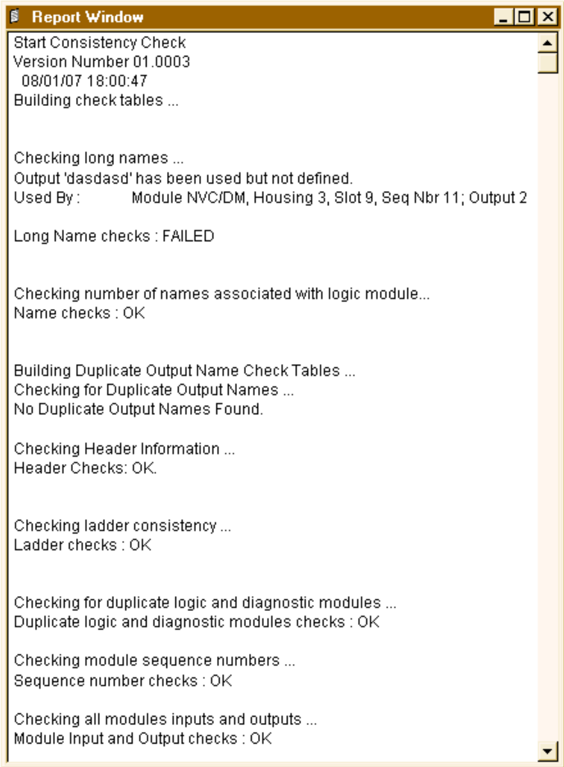


Figure B.16 Consistency Check Report—example

B.5.15 Compilation Report

```
Report Window
Start Compilation

Version Number 01.0006
  08/01/07 18:00:47

Compiling Comment Data...
Reading Reserved Timers and Latches...
Reading User Timers...
Reading User Latches...
Building list of I/O Modules...
Reading Coils...
Reading Contacts...
Checking Inputs...
Checking Outputs...
Checking Coils...
Checking Contacts...
Compiling True Configuration data...
Compiling True Initialisation data...
Compiling True Timer data...
Compiling True IOMapping data...
Compiling True Logic data...
.. WR1O3
.. WR1O2
.. WR1O1
.. EVO1
.. V23O8
.. N42O1
.. FTO6
.. FT12F
.. VR50O8
.. WR15O2
.. FSO6
.. FSO4
.. FTO1
.. FSO1F
.. FTO12
.. FTO1F
.. VR50O5
.. N42O32
.. WR31O4
.. V23O1
.. N42O64
.. FSO1
.. EVO33
.. EVO66
.. FSO4F
.. N23O1
.. WR31O1
.. VR50O1
```

Figure B.17 Compilation Report—example

B.5.16 Compare PROM Data or Compare NVC/DM Upload Report

The report is written to the Report Window where it can be viewed, print-previewed or printed.

It compares two installations (or two versions of the one) by listing all differences, which are marked with >> at the left edge of the report.

See section 8.3.3, "Comparing Uploaded Installations (Optional VLM and NVLM Verification: Part 2)".

Only use this report if one or both of the installations has been re-created by the upload and decompile process (section 8.3). To compare two "original" (ie not re-created) installations, use the Difference Report (section B.5.2).

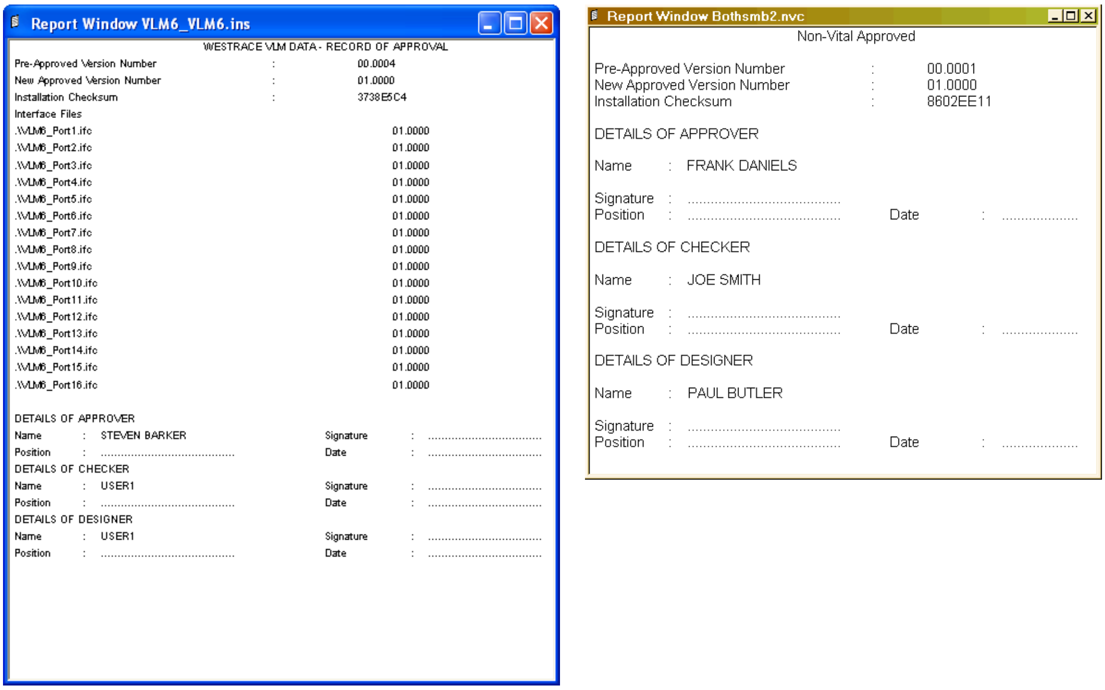
Note: ***Only the first difference in rungs is reported.***

Section 8.3.3, "Comparing Uploaded Installations (Optional VLM and NVLM Verification: Part 2)" explains how to generate this report.

```
Report Window
Comparing information which is sent to PROM for installation Nairn.ins
with that for installation Bothsmb.ins ...

Filename:               Nairn.ins               Bothsmb.ins
Installation Name:      NAIRN                   RA Test 1
Installation Address:   150                     0
Approval Details:       Feb 01 2007 14:14:55    Not Approved
Checksums:              0x50DFCBE3              0xE8D9B94E

At 13:34:01, on 09/11/01.
   Comparing Housing No. 1 ...
      Comparing HVLM128 Modules in slot position 1.
         Comparing User Latches.
>>          Latch[17] PROM Name changed from "10(A)NLR" to "L1".
>>          Latch[23] PROM Name changed from "10(A)RLZR" to "L1200".
>>          Latch[106] PROM Name changed from "10(A)RPR" to "L1201".
>>          Latch[59] PROM Name changed from "10(A)UCR" to "L1600".
>>          Latch[21] PROM Name changed from "10(B)NLR" to "L1601".
>>          Latch[27] PROM Name changed from "10(B)RLZR" to "L2000".
>>          Latch[108] PROM Name changed from "10(B)RPR" to "L2001".
>>          Latch[63] PROM Name changed from "10(B)UCR" to "L2057".
>>          Latch[50] PROM Name changed from "10(OFF)SR" to "L400".
>>          Latch[19] PROM Name changed from "101APZR" to "L401".
>>          Latch[31] PROM Name changed from "101N/RPZR" to "L800".
>>          Latch[93] PROM Name changed from "101NKLPR" to "L801".
>>          90 User Latches removed.
         Comparing Reserved Latches.
         Comparing User Timers.
>>          Timer[1] PROM Name changed from "201TJR" to "TM1".
>>          Trigger[1] PROM Name changed from "201TJR(1)" to "TG1".
>>          Timer 1 Duration Changed.
>>          Timer[2] PROM Name changed from "213TJR" to "TM100".
>>          Trigger[2] PROM Name changed from "213TJR(1)" to "TG100".
>>          Timer 2 Duration Changed.
>>          Timer[3] PROM Name changed from "101NWJR" to "TM200".
>>          Trigger[3] PROM Name changed from "101NWJR(1)" to "TG200".
>>          Timer 3 Duration Changed.
>>          24 User Timers removed.
```

Figure B.18 Compare PROM Data Report—example

B.5.17 Approve Installation Reports


Report Window VLM6_VLM6.ins

WESTRACE VLM DATA - RECORD OF APPROVAL

Pre-Approved Version Number	:	00.0004
New Approved Version Number	:	01.0000
Installation Checksum	:	3738E5C4

Interface Files

.\VLM6_Port1.ifc	01.0000
.\VLM6_Port2.ifc	01.0000
.\VLM6_Port3.ifc	01.0000
.\VLM6_Port4.ifc	01.0000
.\VLM6_Port5.ifc	01.0000
.\VLM6_Port6.ifc	01.0000
.\VLM6_Port7.ifc	01.0000
.\VLM6_Port8.ifc	01.0000
.\VLM6_Port9.ifc	01.0000
.\VLM6_Port10.ifc	01.0000
.\VLM6_Port11.ifc	01.0000
.\VLM6_Port12.ifc	01.0000
.\VLM6_Port13.ifc	01.0000
.\VLM6_Port14.ifc	01.0000
.\VLM6_Port15.ifc	01.0000
.\VLM6_Port16.ifc	01.0000

DETAILS OF APPROVER
Name : STEVEN BARKER   Signature : ....
Position : ....   Date : ....
DETAILS OF CHECKER
Name : USER1   Signature : ....
Position : ....   Date : ....
DETAILS OF DESIGNER
Name : USER1   Signature : ....
Position : ....   Date : ....

Report Window Bothsmb2.nvc

Non-Vital Approved

Pre-Approved Version Number	:	00.0001
New Approved Version Number	:	01.0000
Installation Checksum	:	8602EE11

DETAILS OF APPROVER

Name : FRANK DANIELS

Signature : ....
Position : ....   Date : ....

DETAILS OF CHECKER

Name : JOE SMITH

Signature : ....
Position : ....   Date : ....

DETAILS OF DESIGNER

Name : PAUL BUTLER

Signature : ....
Position : ....   Date : ....

Figure B.19 Approve Installation Reports—examples

B.5.18 User Report

SUPERVISOR REPORT

Password File = D:\GCSS Software\V6\V6L Executables\V6.pwf

USERS	STATUS
"SUPERVISOR"	Supervisor
"PAUL BUTLER"	Standard

Figure B.20 User Report—example

APPENDIX C: COMPILER ERROR MESSAGES

This appendix describes the errors for the 128 k byte EPROM that may appear in the Report Window during compilation.

Error Number	Definition
1	Too much comment data to fit into the comment data area of the EPROM
2	Reserved latch access mode is unknown or illegal
3	Reserved timer trigger access mode is unknown or illegal
4	Reserved timer output access mode is unknown or illegal
5	Timer duration units are unknown or illegal
6	Timer duration is outside the permitted limits. These are: 0 seconds ≤ timer duration ≤ 59 hours 39 minutes 8 seconds
7	Not used
8	The initial state of a module input is not the same as that of a writeable reserved latch or timer trigger that is associated with it.
9	The initial state of a module output is not the same as that of a readable reserved latch or a readable reserved timer output or a timer trigger or a timer output or an input that is associated with it.
10	User timer trigger is defined without the corresponding user timer output being defined
11	User timer output is defined without the corresponding user timer trigger being defined
12	Not used
13	Module input name is the same as a non-writeable reserved latch
14	Module input name is the same as a non-writeable reserved timer trigger
15	Module input name is the same as a timer output
16	Module input name is not unique
17	Contact name or module output name is the same as a non-readable reserved latch
18	Contact name or module output name is the same as a non-readable reserved timer output
19	Contact name or a module output name is the same as a non-readable reserved timer trigger
20	Module sequence number is not unique
21	Module sequence number is not in the range 1 through 99
22	Coil name is not unique
23	Coil name is the name of a timer output
24	Coil name is the name of a module input
25	Output does not have the same name as either a module input or a timer output or a coil
26	User timer trigger does not have the same name as a module input or a coil

Table C.1 Compiler errors—128K Byte EPROM

Error Number	Definition
27	Contact name is not a module input or a module output or a timer trigger or a timer output or a latch
28	Module input does not have the same name as a timer trigger or a module output or a contact
29	Timer output does not have the same name as a module output or a contact
30	Latch does not have the same name as a contact
31	Slot number of the module is not in the range 1 through 15
32	Housing number of the module is not in the range 1 through 4
33	Module is not one of these: VLOMST12, VLOMSS110, VLOMSS24, VLOMSS12, VPIM50, VPIM24, VPIM12, HVLM128A, HVLM128, HVLM64, VLM, VLOMFT110, VLOMFT24, VLOMFT12 VLOMFS110, VLOMFS24, VLOMFS12, VLOMST110, VLOMST24, VLOMST12, CNVC, NVC422, NVC232, EVTC, VTC232, GPOM110L, GPOM50LL, VROM50, VROM24, VROM12, TCOM, GPOMFSU, WCM, CCM, DIAG, VLM5, NVC/DM VLM6 NCDM
34	Number of reserved mnemonics is more than 255
35	To many timers
36	To many mnemonics
37	A ladder logic error has occurred
38	The depth of a ladder logic rung is more than 100. A rung is converted to a tree for compilation. The length (number of nodes) of the longest path from the root to a leaf of the tree is the depth of the rung
39	Too much ladder logic data to fit in the allocated area of the EPROM
40	Module has no inputs or outputs
41	Too much module I/O data to fit in the allocated area of the EPROM
42	Number of I/O modules is more than 57
43	Number of vital I/O modules in the installation is less than 2
44	VLOMFT module flash proving input is used where the corresponding flash control output is not used
45	VLOMFT module lamp proving input or a flash control input or a flash proving input is used where the corresponding lamp control output is not used
46	GPOM module has an odd number of initialisation bytes

Table C.1 Compiler errors—128K Byte EPROM *(Continued)*

Error Number	Definition
47	Entries for the GPOM output Normal Current Minimum and Normal Current Maximum values are not within the following limits: 1 For each GPOM variant, for each GPOM output, if either Hot Proving or Cold Proving or both are selected, the minimum and maximum values for the Normal Current shall lie within or at the range limits as follows (except as in 3 below): GPOM50LL 0.0–0.0213 A GPOM110L 0.0–2.38 A 2 The maximum Normal Current value shall be greater then the minimum Normal Current value (except as in 3 below). 3 If Hot Proving is Not Selected and Cold Proving is Not Selected, then for all GPOM variants the minimum Normal Current shall be 0.0 A and the maximum Normal Current shall be 0.0 A.
48	Entries for the GPOM output Dimming Current Minimum and Dimming Current Maximum values are not within the following limits: 1 For each particular GPOM variant, the maximum and minimum Dim Current values shall lie within or at the range limits and shall be able to be selected from the values specified for the Normal Current values. (see error no 47). 2 The minimum Dim Current shall be less than the maximum Dim Current.
49	Entries for the GPOM output **F**lash **O**ff Non Zero (FONZ) Current Minimum and FONZ Current Maximum values are not within the following limits: 1 FONZ Current Maximum < Normal Current Minimum (see error no 47) This inequality must hold when the current values have been scaled and truncated to integer precision. 2 FONZ Current Maximum < Dimming Current Minimum (see error no 48) This inequality must hold when the current values have been scaled and truncated to integer precision. 3 FONZ Current Minimum < FONZ Current Maximum This inequality must hold when the current values have been scaled and truncated to integer precision, except in the case where FONZ Current Minimum = FONZ Current Maximum = 0.0.
50	The range for flash frequency testing by the GPOM has been set outside the limits: 0.5Hz ≤ minimum flash frequency < maximum flash frequency ≤ 1.33Hz
51	The range for flash mark-space testing by the GPOM has been set outside the limits: 50% ≤ minimum flash% time on < maximum flash% time on ≤ 67%
52	The value for the GPOM dimming energising voltage percentage of normal energising voltage has been set outside the limits: 50% ≤ dimming energising voltage ≤ 70%
53	The value for the GPOM flash de-energising voltage percentage of normal energising voltage has been set outside the limits: 5% ≤ dimming energising voltage ≤ 70%

Table C.1 Compiler errors—128K Byte EPROM *(Continued)*

Error Number	Definition
54	Entries for the GPOMSU output Normal Current Minimum and Normal Current Maximum values are not within the following limits: 0.0 ≤ Normal Current Minimum ≤ 0.0499 0.00078 ≤ Normal Current Maximum ≤ 0.05 Also, Normal Current Minimum < Normal Current Maximum This last inequality must hold when the current values have been scaled and truncated to integer precision.
55	A GPOM module exists where no outputs have been used
56	Dimming is enabled but no dimming control output is defined
57	VLOMS steady lamp control undefined but other controls used.
58	Only one NVC/DM is allowed.
59	No space for extra CNC/NVC module
60	First CNVC slot not used—the first input or output defined on the NVC/DM interface is 240.
61	Too many inputs/outputs on NCDM—the last input or output defined on the NCDM module (when used with an HVLM128, HVLM128a or VLM5) is greater than 480.

Table C.1 Compiler errors—128K Byte EPROM *(Continued)*

APPENDIX D: DECOMPILER ERROR MESSAGES

This appendix describes the errors for the 128 kb EPROM that may appear in the Report Window during decompilation.

The list below summarises the types of errors and where they are described:

Error Number	Definition
1	Error in comment data checksum
2	Error in true data checksum
3	Error in mnemonic data checksum
4	Error in complement data checksum
5	Error in bytewise checksum
6	Not used

Table D.1 Decompilation—Checksum Errors

Error Number	Definition
7	Unexpected NULL character in mnemonic
8	Unexpected NULL character in padding of mnemonic
9	Unexpected character in padding of mnemonic
10	Unexpected padding character in mnemonic NULL area
11	Unexpected character in mnemonic NULL area
12	Mnemonic not contiguous

Table D.2 Decompilation—Mnemonic Errors

Error Number	Definition
13	Comment different in high byte EPROM and low byte EPROM
14	Unexpected title in comment data
15	Comment text improperly terminated
16	Comment text data field empty
17	Comment data colon missing
18	Installation name empty or too big
19	Invalid configuration element type
20	Installation address out of range in comment data
21	Version out of range in comment data
22	Customer name undefined or too long
23	Designer name undefined or too long
24	Checker name undefined or too long
25	Approver name undefined or too long
26	Invalid approval date or time

Table D.3 Decompilation—Comment Errors

Error Number	Definition
27	Initialisation time out of range
28	High byte text inconsistent
29	Low byte text inconsistent
30	Incompatible CSS version
31	Incomprehensible comment
32	Unexpected comment title
33	Non null characters in empty comment data

Table D.3 Decompilation—Comment Errors *(Continued)*

Error Number	Definition
34	True and complement installation addresses are different
35	True installation address parity error
36	The Non-Vital Configuration version numbers in the true and complement data are different.
37	The true Non-Vital Configuration version number is out of range
38	The true and complement compatibility indices are different
39	True compatibility index parity error
40	True system configuration logical state undefined (neither hot standby or stand-alone)
41	Complement system configuration logical state undefined
42	True and complement system configurations (hot standby or stand-alone) are different
43	True flash sync signal logical state undefined
44	Complement flash sync signal logical state undefined
45	True and complement flash sync signals are different
46	Configuration data filler byte error

Table D.4 Decompilation—Configuration Data Errors

Error Number	Definition
47	True and complement number of reserved mnemonics are different
48	Initialisation data filler byte missing
49	True initial state of logic state is undefined
50	Complement initial state of logic state is undefined

Table D.5 Decompilation—Initialisation Data Errors

Error Number	Definition
51	True and complement initial states of logic states are different
52	Unused logic state has non zero initial state
53	Installation address in true data differs from that in complement data
54	Data version in true data differs from that in the complement data
55	EPROM compatibility index incompatible with GCSS
56	Unexpected reserved latch initial state

Table D.5 Decompilation—Initialisation Data Errors *(Continued)*

Error Number	Definition
57	True and complement number of timers used are different
58	Number of timers used is invalid
59	Filler byte after number of timers used is invalid
60	True and complement number of reserved timers are different
61	Invalid number of reserved timers used
62	Filler byte after number of reserved timers used invalid
63	True and complement timer durations are different
64	True timer duration is out of range
65	True and complement timer durations are different for unused timer
66	Invalid unused timer duration
67	Invalid timer trigger or timer output initial state
68	Unexpected reserved timer trigger name
69	Unexpected reserved timer output name
70	Unexpected reserved latch name

Table D.6 Decompilation—Timer Data Errors

Error Number	Definition
71	True and complement number of logic states used are different
72	True number of logic states used is out of range
73	Number of logic states differs from number of mnemonics
74	True and complement primary NVC number are different
75	True Primary NVC number out of range

Table D.7 Decompilation—Module I/O Mapping Data Errors

Error Number	Definition
76	True and complement secondary NVC number are different
77	True secondary NVC number out of range
78	Secondary NVC defined without primary
79	Secondary NVC before primary NVC
80	Primary NVC defined without secondary
81	True and complement number of modules different
82	True number of modules out of range
83	True and complement backplane addresses are different
84	Invalid backplane address
85	True and complement module numbers are different
86	True and complement module types are different
87	Unknown module type
88	True and complement slave kinds are different
89	True and complement input message lengths are different
90	True and complement output message lengths are different
91	True and complement local port addresses are different
92	True and complement adjacent port addresses are different
93	True and complement number of inputs used are different
94	True and complement number of outputs used are different
95	Number of inputs and outputs are both zero
96	True and complement number of initialisation bytes are different
97	True and complement init. data are different
98	True and complement input byte numbers are different
99	True and complement input bit numbers are different
100	True and complement input logic state numbers are different
101	True and complement output byte numbers are different
102	True and complement output bit numbers are different
103	True and complement output logic state numbers are different
104	Module not found in module type table
105	Input message not found in input message table
106	Invalid number of bytes in input message
107	Invalid number of bits in input message
108	Invalid number of bytes in output message
109	Invalid number of bits in output message

Table D.7 Decompilation—Module I/O Mapping Data Errors *(Continued)*

Error Number	Definition
110	Error in filler data after module inputs
111	Error in filler data after module outputs
112	Unused
113	Fixed module data filler error
114	Error in filler data after fixed module data
115	Error in filler data after module init. data
116	Error in module linked list
117	Module linked list is improperly terminated
118	Invalid filler bytes after module header data
119	Unsupported VPIM module
120	Unsupported VLOM module
121	Unsupported VROM module
122	Unsupported VTI module
123	Unsupported VTC module
124	Unsupported NVC module
125	Unsupported TPM module
126	Unsupported DM module
127	Unsupported WCM module
128	Invalid module type
129	Invalid byte and bit address for VPIM
130	Undefined or inappropriate input type for VPIM
131	Fixed module data invalid for VPIM
132	Invalid byte and bit address for VLOMFT input
133	Undefined or inappropriate input type for VLOMFT
134	Invalid byte and bit address for VLOMFT output
135	Undefined or inappropriate output type for VLOMFT
136	Fixed module data invalid for VLOMFT
137	Invalid byte and bit address for VROM input
138	Undefined or inappropriate input type for VROM
139	Invalid byte and bit address for VROM output
140	Undefined or inappropriate output type for VROM
141	Fixed module data invalid for VROM
142	Invalid byte and bit address for VROM input
143	Undefined or inappropriate input type for VTC

Table D.7 Decompilation—Module I/O Mapping Data Errors *(Continued)*

Error Number	Definition
144	Invalid byte and bit address for VTC output
145	Undefined or inappropriate output type for VTC
146	Fixed module data invalid for VTC
147	Invalid byte and bit address for EVTC input
148	Undefined or inappropriate input type for EVTC
149	Invalid byte and bit address for EVTC output
150	Undefined or inappropriate output type for EVTC
151	Fixed module data invalid for EVTC
152	Invalid byte and bit address for NVC input
153	Invalid byte and bit address for NVC output
154	Fixed module data invalid for NVC
155	Fixed module data invalid for DM
156	Invalid byte and bit address for GPOM
157	Undefined or inappropriate input type for GPOM
158	Invalid byte and bit address for GPOM output
159	Undefined or inappropriate output type for GPOM
160	Fixed module data invalid for GPOM
161	Invalid byte and bit address for GPOMSU
162	Undefined or inappropriate input type for GPOMSU
163	Invalid byte and bit address for GPOMSU output
164	Undefined or inappropriate output type for GPOMSU
165	Fixed module data invalid for GPOMSU
166	Rung decompiler error
167	Unexpected token in application logic
168	True and complement unused adjacent installation addresses are different

Table D.7 Decompilation—Module I/O Mapping Data Errors *(Continued)*

Error Number	Definition
169	Cannot insert a row into the rung grid
170	Cannot insert a column into the rung grid
171	Failed to create the end rung
172	Failed to create rung

Table D.8 Decompilation—Rung Errors

Error Number	Definition
173	Invalid end rung operator
174	Invalid root operator
175	Invalid operator
176	Invalid left right
177	Invalid contact assignment
178	Compress failed
179	Invalid link add
180	Invalid link remove
181	Invalid link test
182	Invalid constant symbol
183	No down connection
184	No corresponding top left corner
185	Invalid start for top left corner
186	Invalid cell for top left corner

Table D.8 Decompilation—Rung Errors *(Continued)*

Error Number	Definition
187	Illegal GPOM type
188	Initialisation data incorrect parity in GPOM status byte
189	Unused initialisation bits not zero
190	Minimum FONZ current parity error
191	Maximum FONZ current parity error
192	Minimum dim energised current parity error
193	Maximum dim energised current parity error
194	Minimum normal energised current parity error
195	Maximum normal energised current parity error
196	Minimum FONZ current greater than maximum FONZ current
197	Minimum dim current greater than maximum dim current
198	Minimum normal current greater than maximum normal current
199	Maximum FONZ current greater than minimum normal current
200	Maximum FONZ current greater than minimum dimmed current
201	Minimum flash proving period out of range
202	Maximum flash proving period out of range

Table D.9 Decompilation—GPOM Errors

Error Number	Definition
203	Minimum flash proving byte parity error
204	Maximum flash proving byte parity error
205	Minimum flash percent on out of range
206	Unused
207	Minimum flash percent on byte parity error
208	Maximum flash percent on out of range
209	Unused
210	Maximum flash percent on byte parity error
211	Dimming power level percent on out of range
212	Unused
213	Dimming power level percent byte parity error
214	Flash off percent on out of range
215	Flash off percent on byte parity error
216	GPOMSU unused initialisation bits not zero
217	GPOMSU initialisation data incorrect parity in control byte
218	GPOMSU non zero minimum FONZ byte
219	GPOMSU non zero maximum FONZ byte
220	GPOMSU non zero minimum dim energised current byte
221	GPOMSU non zero maximum dim energised current byte
222	Parity error in GPOMSU initialisation bytes 28 through 33

Table D.9 Decompilation—GPOM Errors *(Continued)*

Error Number	Definition
223	Comment configuration element type wrong size
224	High and low comment headers different
225	Undefined or inappropriate input type for NVC
226	Undefined or inappropriate output type for NVC
227	Invalid byte and bit address for VLOMFS input
228	Undefined or inappropriate input type for VLOMFS
229	Invalid byte and bit address for VLOMFS output
230	Undefined or inappropriate output type for VLOMFS
231	Fixed module data invalid for VLOMFS
232	Invalid byte and bit address for TCOM input

Table D.10 Decompilation—Other Module Errors

Error Number	Definition
233	Undefined or inappropriate input type for TCOM
234	Invalid byte and bit address for TCOM output
235	Undefined or inappropriate output type for TCOM
236	Fixed module data invalid for TCOM
237	Invalid byte and bit address for WCM
238	Fixed module data invalid for WCM
239	Invalid byte and bit address for CCM input
240	Undefined or inappropriate input type for CCM
241	Invalid byte and bit address for CCM output
242	Undefined or inappropriate output type for CCM
243	Fixed module data invalid for CCM
244	Configuration message data invalid for CCM
245	Link information invalid for CCM
246	Undefined or inappropriate input type for VLOMSS
247	Invalid byte and bit address for VLOMSS output
248	Undefined or inappropriate output type for VLOMSS
249	Fixed module data invalid for VLOMSS
250	Undefined or inappropriate input type for VLOMST
251	Invalid byte and bit address for VLOMST output
252	Undefined or inappropriate output type for VLOMST
253	Fixed module data invalid for VLOMST
254	Invalid byte and bit address for VLOMST input
255	Undefined or inappropriate input type for CNVC
256	Invalid byte and bit address for CNVC output
257	Undefined or inappropriate output type for CNVC
258	Invalid byte and bit address for CNVC input
259	Fixed module data invalid for CNVC
260	Data rate invalid for CNVC
261	Invalid byte and bit address for VLOMSS input
262	Invalid NVC pairing
263	Invalid TRUE Non-Vital Configuration layout information
264	Invalid COMP Non-Vital Configuration layout information
265	Invalid NVC/DM housing data
266	Invalid NVC/DM position data

Table D.10 Decompilation—Other Module Errors *(Continued)*

Error Number	Definition
267	Invalid NVC/DM right slot data
268	Invalid NVC/DM left slot data
269	Invalid NVC/DM interface present flag
270	Invalid comment VLM interface version
271	CRM mismatch error
272	OFF set out of range
273	Config element type error
274	Invalid comment Non-Vital Configuration format version
275	VLM Interface version in CONFIG data differs from comment data
276	Non-Vital Configuration format version in CONFIG data differs from comment data
277	True and complement No of DPRAM modules different
278	True of DPRAM modules out of range
279	True and complement first DPRAM module address different
280	True first DPRAM module address out of range
281	Decompile fixed DPRAM module data Filler error
282	True and complement DPRAM module type are different
283	True and complement LOC Timeouts are different
284	Unrecognised DPRAM module type
285	DPRAM module number out of range
286	Network port already exists
287	Network port destination system type different
288	Network port destination system address different
289	Network port source system type different
290	Network port source system address different
291	Network port product data version different
292	Network port compatibility data index different
293	True and complement rung data different
294	There has been an error in the error reporting system

Table D.10 Decompilation—Other Module Errors *(Continued)*

GLOSSARY

Active The window or dialog box on the computer screen that has “focus” is the active window. The window or dialog box in focus is usually highlighted by a change in appearance or colour attributes.

Application Data The logic, information and data component of a complete WESTRACE application. The other component is the WESTRACE hardware.

Application Logic Logic that defines how the inputs and outputs for a particular WESTRACE application are related. See also *Ladder Logic*.

Approach Locking Locking applied to prevent a route normalising when a train is approaching.

ATP Automatic Train Protection. See also *WESTECT ATP*.

Boolean Logic Equations A method to evaluate a particular set of inputs that produces a result that is either “True” or “False”.
True = Logic 1
False = Logic 0

Back Up A saved copy of the latest version of installation data to an alternative storage medium.

Cancel Signal Setting a signal back to red, displaying the stop aspect.

Case-sensitive Refers to the use of uppercase and lowercase letters. For example, if the user name “Allan” was case-sensitive, Allan would need to log onto the system using only the capital letter “A” and lowercase “llan”. No other combinations would work.

The GCSS name and password commands are **not** case-sensitive, and can be typed in any combination of uppercase or lowercase letters.

CCM Configurable Communications Module

CED Configuration Element Data—was synonymous with Application Data in the early days of WESTRACE. The GCSS uses the term when referring to the non-vital installation configuration data.

Client Alternative terms: master or office

CNVC Configurable Non-Vital Communications Module

Compilation The process of creating the installation data from the source file information.

Control Tables A tabular list of controls required to permit the state of routes and points to be altered.

Delete Installation Deletes all versions of all data for the installation.

Diagnostic Module Interface Displays details the correspondence between the mnemonics and the logic status numbers as used in the Diagnostic Module.

Dialog Box Dialog boxes are (usually) small panels displaying buttons or fields for selecting and entering data. They must be responded to and be closed before the GCSS will continue.

EVTC	See *VTC*.
Flashing Lamp Output Module	A WESTRACE module used for vitally controlling steady or flashing signal lamps from a WESTRACE application.
GCS	Graphical Configuration System—the tool by which the Application Engineer can enter, verify and validate the data for a particular installation which is used to configure a WESTRACE system.
GCSS	Graphical Configuration Sub-System—a computer software package. It is used by Railway Signal Engineers to design Railway Signal Interlockings using WESTRACE Vital Logic Equipment.
GPOM	General Purpose Output Module
GPOMFSU	General Purpose Output Module Switching Unit (GPOMFSU2000)
GSIM	GSIM is a computer application that simulates the operation of WESTRACE logic and trackside equipment. It enables Signal Engineers to functionally test WESTRACE vital application logic for a WESTRACE installation.
GTT	Graphical Template Tool—software used to create installation templates for later use with GCSS.
HVLM	Hot-Standby Vital Logic Module—a hot-standby version of the VLM. The HVLM controls the operation of each WESTRACE application. It performs all logic processing and supervises communication between itself and each WESTRACE module.
Hot Standby	The arrangement where two WESTRACEs, a main (online) unit and a standby unit, run in parallel. The standby unit automatically takes over in the event of a fault in the main unit.
Housings	The physical unit used to hold the WESTRACE modules in an installation.
I/O	Input or Output
ICS	Installation Check System—a software program used to check WESTRACE configuration data in an installed system.
INCL	Inter-NCDM Communications Link
Initialisation (System)	This is a time prior to normal operation when the WESTRACE application determines the current state of the external inputs.
Installation	Part of a complete set of WESTRACE Application Data. Often preceded by VLM or NVLM. The complete WESTRACE Application Data can comprise: • a VLM installation alone; • a VLM installation, an NVLM installation plus interface file(s).
Intermediate outputs	Un-required outputs which are activated as part of bad logic design.
Ladder Logic	The system by which Application Logic is expressed.
Latches	Internal logic states within the Application Logic. They do not have a physical input or output.
LOC	Loss of Communication
LOIC	Loss of Input Channel

LOOC	Loss of Output Channel
Mnemonic	Abbreviated names that consists of numbers and letters to represent particular logic states or functions.
Module Bit Allocation	The process of allocating mnemonics to I/O bits on a module.
NCDC	Network Communication Diagnostic Card—the NCDM's main circuit board.
NCDM	Network Communication Diagnostic Module—comprises an NCDC and an NCD PFM.
NVC	Non-Vital Communication Modules—a serial data module used for communicating serially between a WESTRACE applications and a non-vital control system.
NVC/DM	Non-Vital Communications and Diagnostic Module
Non-Vital Telemetry	This is the system of remote control used in conjunction with NVC modules.
Normalise route locking	See *Route Locking*.
NVLM	An acronym representing non-vital logic modules such as the NCDM or NVC/DM.
OPCR	Output Power Control Relay
Output	The output of the WESTRACE such a relay or lamp output.
Parallel Processing	Many relays in an all-relay interlocking can operate independently of other relays in the interlocking, in other words, more than one logic evaluation may be occurring at the same time.
PEN	Port Enabled
Peripheral Interface	The programs and devices used to make peripheral transfers.
Printouts	Printed copy of output data or reports.
PROM	Programmable Read-Only Memory
PROM Blowing Form	A printed record of PROM programming having taken place.
PROM Programming	Storing data in a PROM.
PSU	Power Supply Unit
RAM	Random Access Memory ("memory")
Restore Signal	See *Cancel Signal*.
Route Locking	The holding of points and conflicting routes.
Route Proving	The checking that the route is safe for the passage of a train.
Rung	An element of logic, expressed as a set of Boolean logic equations.
Rung Acceptance	The process of checking a rung for correctness prior to being accepted.
Sequential Processing	WESTRACE logic evaluation that occurs one rung at a time starting from rung 1.
Server	Alternative terms: slave or field

Slot	A space in a housing where WESTRACE modules can be inserted.
Source Files	A collection of files which contain all the data that has been entered into the GCSS.
Steady Lamp Driver Module	This is a WESTRACE input and output module used for vitally controlling steady signal lamps from a WESTRACE application.
TCOM	Track Code Output Module
Timer	A device or circuit that provides time signals at regular, specified intervals for purposes of controlling a sequence of events or synchronising events in separate operations.
Track Plan	The plan of the railway detailing the geographical location of railway equipment including tracks, points and signals.
Vital	A process that can cause or lead to an accident should it fail.
Vital Logic Equipment (WESTRACE)	The physical equipment used in a WESTRACE application.
VLC6	Vital Logic Card for the VLM6.
VLM	An acronym representing the vital logic modules HVLM128, HVLM128a, VLM5 or VLM6.
VLOM	Vital Lamp Output Module—for switching an external lamp supply to signal lamps under control of the VLM logic.
VPIM	Vital Parallel Input Module—used for inputting vital parallel inputs into a WESTRACE application.
VROM	Vital Relay Output Module—for switching an external signalling supply to drive signalling relays under control of the configuration logic.
VTC	Vital Telemetry Continuous Module—provides an interface to other WESTRACE systems over dedicated serial communications channels.
WESTECT ATP	WESTECT Automatic Train Protection—a Westinghouse Brake and Signal Co. proprietary system used to automatically protect a train.
WCM	WESTECT Communication Module—a WESTRACE module used to transmit serial data to a train. Part of the WESTECT ATP system.
Window	Microsoft Windows NT is the computer operating system for the GCSS. The term "window" refers to a bordered frame on the computer monitor where information is displayed or entered.
WRSA	Westinghouse Rail Systems Australia

INDEX

This index identifies material that is referenced in other WESTRACE manuals and is not readily identifiable through the Table of Contents.

Reader's Comments

Westinghouse Rail Systems Australia values your thoughts as the user of this document. We would be grateful to receive your views about this document and welcome any suggestions for improvement. Please send your comments to:

The Manager, Technology and Training
Westinghouse Rail Systems Australia
179–185 Normanby Road (Locked Bag 66), South Melbourne, VIC 3205

Phone +61 3 9233 8840 Fax +61 3 9233 8702 Email wrsa-training@wrsa.com.au

Document Name: **WESTRACE User Manual**
WRTOGCSS-WRSI, Issue 9.0

Document Owner: ______________________________

____

Contact Name: ______________________________

Address: ______________________________

____

____

Email: ______________________________

On a scale of 1 to 10, please rate this document for being suitable to your needs. A rating of 0 means this document is of no benefit; a rating of 10 means it is completely satisfactory for the intended purpose.

Rating: 0 – 5 – 10 (**no benefit** … **very useful**)

Comments: ______________________________

____

____

____

____

____

____

____

Signed: ____________________ Dated: ____________________

Comments are continued on the following ________ pages.