

Keyboard and TD Map Reference

For approval and authorisation of this document, please refer to the IECC Applications Manual Contents, RT/E/S/10040.

This document is the property of Network Rail. It shall not be reproduced in whole or in part, nor disclosed to a third party, without the written permission of the document owner.

**Published and issued by
Network Rail
40 Melton Street
LONDON
NW1 2EE**

© 2003 Network Rail

This page is intentionally blank

Amendment Record

Issue	Date	Description
4	29/10/1999	Converted to Word format by AEA Technology Rail. Same as issue 3 except for corrections to the titles and references of cross-referenced documents.
5	10/02/2003	Railtrack amended to Network Rail. Section 4.1.11 'Message' format definition amended in accordance with SAO-IEC-R103. Minor typographical corrections.

Author	Approved	Date	Authorised	Date
--------	----------	------	------------	------

Contents

1	Introduction	1
2	Assumptions And Restrictions	2
3	Overview Of Components	3
4	File Preparation	4
4.1	GIF	4
4.2	CREATE FILES	9
5	Operation	11
5.1	BASIC OPERATION	11
5.2	OPTIONAL SWITCHES	11
6	Error Messages	12
6.1	BERTH NAME <TEXT> NOT FOUR CHARACTERS	12
6.2	CANNOT FIND SUBSYSTEM IN CPF - EXIT FROM TD MAPPER	12
6.3	CANNOT OPEN FILE <TEXT> FOR OUTPUT	12
6.4	ERRORS IN ACTION CODE CLASSIFICATION TABLE	12
6.5	ERRORS IN MESSAGE TEMPLATES POINTER TABLE	12
6.6	ERRORS IN PARAMETER TYPES TABLE	13
6.7	BAD ATTRIBUTE IN LINE <TEXT>	13
6.8	BAD BACKGROUND IN LINE <TEXT>	13
6.9	BAD FOREGROUND IN LINE <TEXT>	13
6.10	BAD SIZE IN LINE <TEXT>	13
6.11	BAD X CO-ORDINATE IN LINE <TEXT>	13
6.12	BAD Y CO-ORDINATE IN LINE <TEXT>	13
6.13	CANNOT OPEN <TEXT> FOR INPUT	13
6.14	UNKNOWN TD TYPE IN LINE <TEXT>	14
6.15	SSI <TEXT> NOT FOUND IN CPF	14
6.16	STRING TOO LONG	14
6.17	SUBSYSTEM <TEXT> IS NOT AN SDS OR FB	14
6.18	SUBSYSTEM <TEXT> IS NOT IN CPF	14
6.19	TEMPLATE STRING IS TOO LONG	14
6.20	THE CPF CANNOT BE LOADED	14
6.21	THE RELATIONAL ALARM NAME <TEXT> DOES NOT BEGIN WITH AN 'A'	14
6.22	THE SSI IDENTITY <TEXT> IS NOT IN THE SSI FILES	15
6.23	THE ADDRESS <TEXT> IS NOT ONE CHARACTER IN SIZE	15

6.24	THE ALARM NAME <TEXT> IS NOT IN THE GIF	15
6.25	THE ALARM TEXT <TEXT> IS GREATER THAN EIGHT CHARACTERS IN LENGTH	15
6.26	THE AREA NAME <TEXT> IS NOT IN THE CPF	15
6.27	THE NEW TEMPLATES HAVE NOT BEEN READ IN CORRECTLY	15
6.28	THE RAPID ADDRESS <TEXT> IS NOT RECOGNISED	15
6.29	TIPLOC CODE CHARACTER <TEXT> IS NOT ALPHANUMERICAL	15
6.30	TIPLOC CODE CHARACTER <TEXT> IS NOT UPPER CASE	15
6.31	TIPLOC CODE <TEXT> ABBREVIATED TO EIGHT CHARACTERS	16
6.32	UNABLE TO OPEN RGI FILE <TEXT> FOR INPUT	16
6.33	USAGE: <TEXT> [-CHV] <X> <FILE>	16
6.34	SYNTAX ERROR	16
6.35	CANNOT OPEN <TEXT> FOR INPUT	16
6.36	LINE <TEXT> ITEM OUT OF RANGE	16
6.37	LINE <TEXT> <NAME> ALREADY DECLARED	16
6.38	<TEXT> UNEXPECTED	16
6.39	SUBSYSTEM <TEXT> IS NOT A MASTER	17
7	Relational/General Interface File Preparation	18
7.1	ERROR REPORTING	18
7.2	WORKED EXAMPLE OF INPUT FILE	20
8	Appendix	21
8.1	WORKED EXAMPLE OF INPUT FILE	21
8.2	SYNTAX	24

1 Introduction

This document is the reference manual for the Keyboard and Train Describer Map data compiler and contains information concerning the following subjects :-

- preparing the input files for the Keyboard and TDMAP data compiler;
- installation of the Keyboard and TDMAP data compiler;
- precise input file syntax description;
- error messages and their interpretation;
- setting up TD maps data;
- a worked example of the Keyboard and TDMAP data compiler source files.

Users of Automated Data Preparation (ADP) tools may use this document to check and modify ADP prepared data.

2 Assumptions And Restrictions

- 2.1 The Central Parameter File may be referred to within this document by its abbreviation CPF.
- 2.2 The Keyboard and TDMap Input File (General Input File) may be referred to within this document by its abbreviation GIF.
- 2.3 The Relational/General Interface File may be referred to within this document by its abbreviation RGI.
- 2.4 The reader of this document is assumed to be familiar with the content of the CPF.
- 2.5 The reader of this document is assumed to be familiar with the screen design utility, 'Create'.
- 2.6 The Keyboard and TD Map data compiler is known throughout this document as 'the compiler'.
- 2.7 Create always names files in upper case, and the Keyboard and TD Map compiler expects its Create input files to have upper case names.
- 2.8 The contents of the RGI file are assumed to be correct.

3 Overview Of Components

The compiler consists of a number of separate components :-

george;
tdmapper;
modpre.a;
preprocessor;
catfiles;
clearup;
compile;
messages.

The compiler is run by executing the main program, “george”, which then runs the other components as necessary.

Copyright Network Rail
Provided by IHS under license with Network Rail
No reproduction or networking permitted without license from IHS

4 File Preparation

The compiler must be run once for each display subsystem or fringe box. Before the compiler can be used, the following files must have been prepared :-

CPF;

GIF;

the Create listings for any TD maps;

the screen dumps for any TD maps.

For a display subsystem, additional file(s) must be prepared :-

RGI;

Keyboard Cross Reference File (if required).

The RGI is prepared using the IECC Keyname Compiler, and is described in section 7 of this document. Preparation of the CPF is described in detail in SAO-IEC-HD-43 "Central Parameter File Reference". The other files are described in sections 4.1, 8.1.1 and 4.2 of this document.

The GIF contains data which is specific to the IECC site. This always needs to be prepared for each site.

This section describes the input file format and the effect of the various data fields on compiler operation. The syntax is described in English, but for a more rigorous description of the input file syntax, refer to section 8.2.

4.1 GIF

This file contains all the data required by the compiler that is not included in the CPF, the Create files and the RGI.

The GIF is described below. There is an example of the GIF in section 4.1, 8.1.1.

Data should appear in the following order.

4.1.1 CPF statement

CPF: {FILE: "../SERVICE/cpf1"}

The **CPF:** statement must be present and is used to inform the compiler that the following is the file path to the CPF. The **FILE:** statement must be present and is used to set the search path used by the compiler to load the CPF. The file path name and the **FILE:** statement must be delimited by curly brackets.

Copyright Network Rail
Provided by IHS under license with Network Rail
No reproduction or networking permitted without license from IHS

4.1.2 RGI statement

RGI: {FILE: "../SDS/NORTHERN/ifx"}

The **RGI:** statement must be present and is used to inform the compiler that the following is the file path to the RGI.

For a display subsystem :-

The **FILE:** statement must be present and is used to set the search path used by the compiler to load the RGI.

For a fringe box :-

The RGI is not required so no file path is necessary. However, the curly brackets must be opened and closed as follows :-**RGI:** {}

4.1.3 SUBSYSTEM statement

The format of this section is as follows :-**SUBSYSTEM:** {Subsystem}

Where :-

Subsystem The logical name of a subsystem, as entered in the CPF

The **SUBSYSTEM:** statement must be present and is used to tell the compiler to load the identity of the subsystem for which the data is being set up. The subsystem must be a display subsystem or a fringe box. The entry must be delimited by curly brackets.

4.1.4 BERTHS statement

The format of this section is as follows :-**BERTHS:** {Berth Type Display Accessibility RapidAddress }

Where :-

Accessibility	The keyword ACCS if the berth is accessible for keyboard commands or - if not
Berth	A four character TD berth name
Display	The keyword DISP if the berth is used in the relational displays data or NOTDISP if not
RapidAddress	The letter to be used as a rapid address, if the berth has one, or - if not
Type	One of the keywords APPR, REMO, IMAG, NORM and DEPA depending on whether the berth is an approach berth, an approach berth with is not displayed, an imaginary berth, a normal displayed berth or a departure berth

The **BERTHS:** statement must be present and is used to tell the compiler to load the information regarding berths. There can be zero or more berths entered here, in any order. If no berths are entered then no data field should follow the BERTHS: statement, and the next entry should be the ACI: statement.

For a display subsystem, this should contain information regarding each of the berths in this IECC that the subsystem might need to know about, for display or stepping purposes.