

Displays Relational Data Guide

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 Railtrack. It shall not be reproduced in whole or in part without the written permission of the Railtrack Professional Head of Signalling Asset Management.

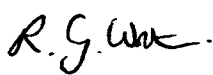
**Published and issued by
Director, Asset Management
Railtrack plc
Railtrack House
Euston Square
LONDON
NW1 2EE**

© 1999 RAILTRACK PLC

This page is intentionally blank

Amendment Record

Issue	Date	Description
3	29/10/99	Converted to Word format by AEA Technology Rail. Same as issue 2.

Author	Approved	Date	Authorised	Date
		29/10/99		12/10/00

Copyright Network Rail

Contents

1 Introduction	1
2 Overview Of Relational Data Files	2
2.1 RELATIONAL DATA FILE EXTENSIONS	2
2.2 RELATIONAL DATA FILE CONTENTS	2
2.3 IDENTITIES	3
2.4 GENERAL SYNTAX	3
2.5 FURTHER INFORMATION	3
3 Trigger Data	4
3.1 CONDITIONS	4
3.2 DISPLAY COMMANDS	6
4 Panel Requests	8
4.1 POINTS	8
4.2 ROUTES	8
4.3 AUTOSIGNALS AND EMERGENCY REPLACEMENTS	8
4.4 AUTOBUTTONS	8
4.5 RELEASES	9
4.6 ARS SUBAREAS	9
4.7 LATCHES	9
5 Points	10
6 Signals	11
7 Track Circuits	13
8 Routes	14
8.1 NORMAL ROUTE	14
8.2 LONG ROUTE	14

9 Subroutes And Suboverlaps	16
10 Releases	18
11 Subareas	19
12 Latches	20
12.1 NORMAL LATCH	20
12.2 POSSESSION OR ISOLATION REMINDER LATCH	20
12.3 AUTOMATIC PANEL REQUEST TRANSMITTER LATCH	21
13 Alarms	22
14 HIT Areas	23
15 Isolation Areas	24
16 Train Description Berths	25
17 Train Describer Steps	26
18 Initialisation	28
19 Signal Posts	29
19.1 CHANGES REQUIRED TO RELATIONAL DATA AND INTERLOCKING IDENTITY FILES	29
19.2 POST DATA COMPILER	30

1 Introduction

The purpose of this guide is to describe the processes involved in generating the Relational Data for the Displays (DIS/SDS/RDF) subsystem. It describes a process which complements the Automated Data Preparation (ADP) tools and utilities. This guide is useful for the generation of the following :-

- data which ADP does not prepare automatically e.g. special indications, features at the fringes of the scheme, trigger data, releases and alarms; and
- minor changes to ADP prepared data where re-running the ADP tools and utilities would be more time consuming.

This guide gives an overall description of the relation data files followed by details of data preparation for each individual file. The source data required for each is listed at the beginning of the relevant section.

Additionally, the necessary amendments to various relational data and interlocking identity files required to ensure signal posts display in white when a route is set are described, together with the operation of the **posttool** data compiler required as a pre-processor to compiling the relational data with the relational data compiler **asstest**.

The following abbreviations will be used throughout the document :-

SL	Screen Layout
TD	Train Descriptor
TDN	Train Description

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

2 Overview Of Relational Data Files

There are 15 relational data files in total. Each file has a unique name and extension which is generated by ADP. Data is normally initially entered using ADP, thereafter, amendments can be entered directly into the files if this is less time consuming.

2.1 RELATIONAL DATA FILE EXTENSIONS

A unique name is associated with each Displays workstation. To this name, a three character extension (*.xxx) is appended to this name forming a unique filename for each file. The contents of each file and its extension are given below.

<u>Contents</u>	<u>Extension</u>
Panel requests	.pqs
Points	.pnt
Signals	.snl
Track circuits	.tct
Routes	.rot
Subroutes and suboverlaps	.srt
Releases	.rel
Subareas/general	.sba
Latches	.lth
Alarms	.alm
Hit areas	.hit
Isolation areas	.iso
Train description berths	.dsb
Train describer stepping data	.stp
Initialisation	.int

ADP will generate all 15 files, even though some of them may be empty. This is to ensure that the relational data compiler functions correctly. It should be noted that ADP does not generate the files on a one-to-one basis i.e. data is taken from various parts of the database when constructing the files.

2.2 RELATIONAL DATA FILE CONTENTS

In the following sections of this guide, the data required in each relational data file is described. Each section describes the type of information required in each file and the syntax which must be used to express this information, giving a single example. Some information described is not necessarily required for every identity in a particular file. In some sections, the inter-relationships between that particular relational data file and others (and also Image and/or Create) are given where thought relevant.

2.3 IDENTITIES

Various forms of signalling identities are generated and used by the ADP database and are contained within the relational data files. As these forms are referenced in the following sections, an explanation of each is given here.

2.3.1 SSI Identities

These are the identities given in the actual SSI identity files received from SSI data preparers. They include virtually all possible printable characters and can be up to a maximum of 24 characters in length. These will be referenced as SSI-NAME in the following sections.

2.3.2 Full Identities

These identities are the SSI identities with all relational data syntax characters removed e.g. full stop. Under some circumstances, mainly for clarity, these identities will require amending. These will be referenced as FULL-NAME in the following sections.

2.3.3 IECC Identities

These identities are constructed from the Full identities by truncating them where necessary to 8 characters. Normally the 8 characters are the first 7 plus the last character of the Full identity, but this is not always the case as they require to be unique. These will be referenced as IECC-NAME in the following sections.

2.4 GENERAL SYNTAX

Each relational data file contains a number of entries, each of which has the general form :-

<SSI-NAME> = IECC-NAME
 <quantity/type statement>
 <associated panel requests list>
 <trigger data>
 <location data>
 <cross reference data>

.

Statements in angle brackets <> denotes that these are optional.

Trigger data describes the conditions under which some display changes occur and what those changes are, and also controls TD stepping. This is described in the next section.

Comments can be inserted at the end of any line, preceded by a space and slash character, or can be on a separate line starting with a slash.

Each entry is limited to a maximum of 70 lines, each up to a maximum of 80 characters.

2.5 FURTHER INFORMATION

Further information on producing relational data through use of ADP can be found in Section 5.4 of Part 3 of the ADP User Guide, reference SCU-D-ADP-UG-31.

3 Trigger Data

Trigger data, describing the conditions under which some display changes occur, has the following format :-

```
$<S>(  if      {CONDITION(S)}
      then
          {DISPLAY-COMMAND(S)}
      ...../additional "if ... then" clauses
      if      {CONDITION(S)}
      then
          {DISPLAY-COMMAND(S)}
      else
          {DISPLAY-COMMAND(S)}
      )
```

Note “if”, “then” and “else” must be lower case.

Trigger data expresses the CONDITIONS under which any DISPLAY-COMMAND is executed, according to whether the CONDITIONS are true or false, and describes those commands. This is normally used to provide special display requirements in addition to standard requirements produced by other means. Note that only the final statement can contain an “else” clause. A DISPLAY-COMMAND can only contain one TD label (see section 17) and these cannot be used in an “else” clause.

A similar statement format is used for the control of TD stepping, where only one DESCRIPTION-STEP replaces DISPLAY-COMMAND(S). This is described further in section 17.

If the initial “\$” is followed immediately by “S”, the trigger is a ‘special case’ where the DISPLAY-COMMAND(S) may be different from that usually associated with the function. Any other commands in the statement are **not** executed if such a special trigger statement is present, whether it evaluates true or false. This allows a special display to occur **instead** of a standard display.

3.1 CONDITIONS

The CONDITION(S) clause is of the form :-

IECC-NAME_STATE <& IECC-NAME_STATE>

STATE is a code for equipment states and is dependent on the equipment type.

Only ‘and’ conditions or possible, ‘not’ and ‘or’ are **not** available. However, ‘or’ conditions can be dealt with by adding further “if ... then” statements containing the alternative conditions. This form must also be generated (using the opposite STATE on the CONDITION(S)) if more than one “else” clause would otherwise be required.

STATE can be any one of the following :-

Points:	kn	keyswitch normal
	nc	normal control
	nd	normal detection
	kr	keyswitch reverse
	rc	reverse control

	rd	reverse detection
	dn	not disabled normal
	dr	not disabled reverse
Signals:	lp	lamp proved
	am	auto mode (for emergency replacement, see section6)
	fl	approach locking free
	ac	aspect control
	g	green
	fyy	flashing double yellow
	yy	double yellow
	fy	flashing yellow
	y	yellow
	rs	red & subsidiary off
	r	red
	mr	most restrictive
	nlp	not lamp proved
	bp	button pulled
Track Circuits:	o	occupied
	c	clear
Routes:	s	set
	xs	not set
	f	free
	l	locked
Subroutes:	f	free
	l	locked
Latches:	s	set
	xs	clear
Alarms:	sna	failed & not acknowledged
	sa	failed & acknowledged
	xsna	recovered & not acknowledged
	xsa	recovered & acknowledged

For STATE codes associated with TD stepping controls, see section17.

If there is no STATE code provided to specify a required state then a bit pattern can be specified instead. “1” is to be used to indicate that a bit should be set, and “0” that a bit should be clear. “X” is used to indicate that a bit can be either set or clear i.e. don’t care. Bit patterns should be in double quotes e.g.

S1_”10XXXXXX”

3.2 DISPLAY COMMANDS

The DISPLAY-COMMAND clause is of the form :-

[LIST-ITEM-NAME_QUALIFIER <ACTION-CODE>]

or

[STEP-LABEL(S)]

For the latter, see section 17.

If QUALIFIER = prev, the required list item is already displayed and is to change status according to the ACTION-CODE, which must be present.

If QUALIFIER = new, the required list item is not already displayed and if this is to be in the default status i.e. as originally generated, no ACTION-CODE need be provided.

ACTION-CODE has the format :-

flash_STATE, f(STATE), b(STATE)

where:	for flash	STATE is on, off or prev
	for f(oreground colour)	STATE is COL or prev
	for b(ackground colour)	STATE is COL or prev

If any of the above are omitted, the default state is displayed for that selection.

If STATE = prev, the currently displayed state is maintained. This allows, for example, signal aspects to be changed without affecting a collar (background), or a collar to be changed without affecting the aspect (foreground).

COL can be any one of the following :-

Background:	bk	black
	p	pink
	s	stone
	c	cyan
Foreground:	bk	black
	mg	magenta
	r	red
	y	yellow
	p	pink
	gn	green
	s	stone
	db	dark blue
	dr	dark red
	c	cyan
	gy	grey
	dgn	dark green
	wh	white

The legal combinations of display commands are as follows :-

[LIST-ITEM-NAME_prev, b(COL)]