



A3.B = Main menu select key code in range 1-5. A value in this range specifies which option is to be selected from the directory main menu. The menu is not displayed.

If the value specified is not in the required range.
the main menu is displayed.

Telephone Directory Application

PSD 76.97.22

The information in this document is proprietary to ICL, and is supplied to you in confidence on the understanding that you will not disclose it to third parties or reproduce it, and that you will use it solely for the purpose of developing applications software for use with the ICL product or products described in this document.



0. DOCUMENT CONTROL

0.1 Contents

0. DOCUMENT CONTROL

- 0.1 Contents List
- 0.2 Changes Since Previous Issue
- 0.3 Document Predecessors
- 0.4 Changes Forecast
- 0.5 Document Cross Reference

1. GENERAL

- 1.1 Scope
- 1.2 Introduction
- 1.3 Terminology

2. SUMMARY FOR MANAGEMENT

3. APPLICATIONS INTERFACES

- 3.1 Procedural Interface
- 3.2 Application Chaining Interface

0.2 Changes Since Previous Issue

This is the first issue.

0.3 Document Predecessors

None

0.4 Changes Forecast

None

0.5 Document Cross References

- 0.5.1 PSD 76.97.3.1 OPD Kernel Specification
- 0.5.2 R51002/01 OPD Handbook
- 0.5.3 PSD 76.97.3.2 OPD Director Facilities for Application Writers



1. GENERAL

1.1 Scope

This document defines the interfaces supported by the OPD Telephone Directory Application which enable other OPD applications to interrogate the telephone directory databases or to invoke the Telephone Directory Application.

1.2 Introduction

The OPD Telephone Directory Application provides the OPD user with facilities for the creation, maintenance and display of two separate directories, the Telephone Directory and the Computer Services Directory. These two logical directories are contained within the same Kernel segment (ref 0.5.1).

The user interface to the TDA is described in ref 0.5.2.

In addition to the user interface, the TDA supports interfaces which allow other OPD applications to read specified directory entries and to start the TDA. No facilities are provided for the application to amend the directory.

1.3 Terminology

TDA - Telephone Directory Application



**Product
specification**

**Company
restricted**

PSD 76.97.22

Issue 1/1

Sheet TD-4

2. SUMMARY FOR MANAGEMENT

This document defines the interfaces that allow OPD applications to interrogate the Telephone Directory and the Computer Services Directory.

The procedures allow details of a directory entry with a specified name or shortcode to be obtained. There is no provision for the application to amend the directory.



**Company
restricted**

PSD 76.97.22

Issue 1/1

Sheet TD-5

3. APPLICATIONS INTERFACES

3.1 Procedural Interface - Overview

The Telephone Directory Application provides a number of procedures which allow other applications in the OPD to search the directory for a specified entry and obtain some or all of the data contained in the directory entry. Procedures are also provided to manipulate the data returned.

The procedural interface is invoked by obeying the instruction

TRAP

#T.TDA

The particular procedure required is specified by an action value in DO.B.

The action values are specified in the INCLUDE file TDVALUES.DG.

The procedures are described in the following sections.

3.1.1 GIVE VOICE DIALABLE NUMBER FOR DIRECTORY ENTRY WITH GIVEN SHORTCODE

3.1.1.1 Call parameters

DO.B : Action value = TD.VN SC

AO.L : Absolute address of area to receive the dialable number string. (The area must be at least 25 bytes long.)

A1.L : Absolute address of area to receive the voice extension number string. (The area must be at least 6 bytes long.)

D1.L : Shortcode:

D1.B defines length of shortcode string as 1, 2 or 3 characters

Bytes 0-2 contain the shortcode.

For example the shortcode A01 would be represented as

D1	41	30	31	03
----	----	----	----	----

D2.B : Event check request flag

=0 if no checks on the event Notification Requests are required

≠0 Check for events periodically. If any event is pending the function is aborted with an error response. The



event will not be cleared by this function

3.1.1.2 Return parameters

D0.L positive or 0:

The function was successful and D0.B contains the number of bytes in the dialable number string. The contents of D1.B, D2.W and D3.W are then as follows:

D1.B : Line access indicator:

1 or 2 = number is only allowed on specified line
0 = number allowed on either line

D2.W : Charge band, left justified with D2.B = 0 if only one character in length.

/0 = timing inhibited for this entry
0 = no chargeband

D3.W : Length of voice extension number string. Zero if field not present in the directory entry.

D0.L negative:

The function failed. D0.B contains one of the following error response codes:

OR: Invalid shortcode: shortcode does not start with an alphabetic character or contains non-alphanumeric characters

NF: Entry with requested shortcode not found, or
Voice directory does not exist, or
Entry with specified shortcode does not have a voice number field

SL: Load store in progress

UV: Event occurred. An event which the calling activity has requested notification of has occurred

3.1.1.3 Description

This function searches the Voice directory, from the beginning, looking for an entry with a short code field that matches the one specified. If such an entry is found, the voice 'dialable' number field is converted to an autodial sequence (see section 3.1.5.3) and this string placed in the buffer specified in A0.L. If the entry contains a voice extension number, this is returned in the buffer specified in A1.L.

If no entry is found with the specified shortcode before the end of the voice directory is encountered, or if the directory entry found has no dialable voice number field, an error



response NF is returned.

3.1.2 SEARCH TELEPHONE DIRECTORY FOR SPECIFIED ENTRY AND RETURN SELECTED FIELDS

3.1.2.1 Call parameters

D0.B : Action Value = TD.VM_GNR

D1.W : Length of parameter block

D2.B : Event check request flag:

0 = No checks on the event notification register are
required

≠0 = Check events periodically. If any event is pending
the function is aborted with an error response. The
event is not cleared by the procedure

A0.L : Absolute address of parameter block

Format of the parameter block (word aligned):

Word 0: Field selection bit map. This bit map specifies which
fields in the located directory entry are to be
returned in the parameter block. A bit set to 1 in the
map requests that the field is returned.

Bit definitions:

<u>Name</u>	<u>Number</u>	<u>Field requested</u>
TD.SHORTCODE	0	Shortcode
TD.NAME	1	Name
TD.INITIALS	2	Initials
TD.TITLE	3	Title
TD.VOICENO	4	Voice number
TD.VEXT	5	Voice extension number
-	6	= 0
TD.DATANO	7	Data dialable number
TD.DEXT	8	Data extension number
TD.CHARGE	9	Charge band (one or two characters; / = charging inhibited for this entry)
TD.DESCRPTION	10	Description
TD.VACCESS	11	Voice line access indicator (Number may only be dialled on specified line)
TD.DACCESS	12	Data line access indicator

Byte 2 : Search criteria selection:

= 0 search for entry with specified shortcode

= 1 search for entry with specified name

= 2 search for entry with specified name and initials

Bytes 3 - n : Search string(s). Each string takes the following



form:

Byte 0 = string length
Bytes 1 - n = string

If a type 2 search is specified, the name string
precedes the initials string.

3.1.2.2 Return parameters

D0.L positive or 0:

The function completed successfully, and D1.W contains the
number of bytes transferred.

The parameter block contents are as follows:

Word 0 Bit map of fields returned

Bytes 2 - n Fields transferred from the directory entry in the
'order' given by the field selection bit map
definition, i.e. shortcode first

Each field has the following format:

Byte 0 Inclusive byte count
1 - n String

D0.L negative:

The procedure failed. D0.B contains one of the following
response codes:

OR: Invalid shortcode

NF: Entry not found

UV: Event occurred

SL: Load store in progress

B0: Insufficient space in the parameter block to receive all
of the requested fields. Fields that will fit into the
parameter block have been transferred. Incomplete fields
are not transferred

D1.W contains the length of parameter block required to receive
all the requested fields.

3.1.2.3 Description

This function searches the Telephone Directory for an entry
that matches the specified search criteria. The entry may be
selected by match of shortcode, name or name and initials. The



search uses the standard telephone directory search rules.
These are:

- 1) An upper case letter matches its lower case equivalent.
- 2) In the Initials field, the characters space and . are equivalent.

When an entry is successfully located, the fields specified are transferred to the parameter block if they are present in the directory entry. A bit map is returned in the parameter block defining which fields were successfully transferred.

3.1.3 SEARCH COMPUTER SERVICES DIRECTORY FOR SPECIFIED ENTRY AND RETURN SELECTED FIELDS

3.1.3.1 Call parameters

D0.B : Action value = TD.CS_GNR

D1.W : Length of parameter block (bytes)

D2.W : Event check request flag:

0 = No event check performed

≠0 = Check for events periodically. If any event is pending, the function is aborted with the appropriate error response. The event is not cleared by the procedure

A0.L : Address of parameter block.

Format of the parameter block (word aligned):

Word 0 : Field selection bit map. This map specifies the fields in the located directory entry which if present are to be returned. A bit set to 1 requests that the field is returned.

Bit definitions:

<u>Name</u>	<u>Number</u>	<u>Field requested</u>
TD.SHORTCODE	0	Shortcode
TD.NAME	1	Service name
TD.SERVICENO	6	Service number
TD.CHARGE	9	Charge band
TD.DESCRPTION	10	Description
TD.PROFILE	14	Profile
TD.CSACCESS	15	Line access indicator

Byte 2 : Search criteria selection:



- = 0 search for entry with specified shortcode
- = 1 search for entry with specified name
- = -1 select entry with specified cell tag

If byte 2 = 0 or 1, byte 3 contains the string length and bytes 4 onwards contain the string.

If byte 2 = -1, bytes 4 - 5 contain the cell tag of a valid directory entry.

3.1.3.2 Return parameters

D0.L positive or 0:

The function completed successfully and D1.W contains the number of bytes of data returned in parameter block.

The parameter block contents are as follows:

- Word 0 Bit map of fields returned
- Byte 2 - n Fields transferred from the directory entry in the order given by the field selection bit map (i.e. shortcode first). Each field has the following format:
- Byte 0 = Inclusive byte count
- 1 - n = Field data

D0.L negative:

The function failed. D0.B contains one of the following response codes:

- OR : Invalid shortcode
- NF : Entry not found
- BP : Bad parameter
- UV : Unexpected event occurred
- SL : Load store in progress
- BO : Insufficient space in the parameter block to hold all the requested fields. Fields that will fit into the parameter block have been transferred. Incompatible fields are not transferred. The bit map in word 0 of the parameter block indicates which fields were transferred

D1.W contains the length of parameter block required to receive all the requested fields.



3.1.3.3 Description

This procedure searches the Computer Services directory for the specified entry by name, shortcode or cell tag. When the entry is selected by name or shortcode the standard directory search rules apply, that is, an upper case letter matches its lower case equivalent.

When an entry is successfully located the fields requested which are present in the entry are transferred to the parameter block.

A bit map is returned in word 0 of the parameter block defining which fields were successfully transferred.

3.1.4 EXTRACT FIELD FROM PARAMETER BLOCK AS RETURNED BY TD.VM_GR AND TD.CS_GNR

3.1.4.1 Call parameters

DO.B : Action value = TD.EXTRACT

DI.W : Field number (0-15)

A0 : Address of parameter block

3.1.4.2 Return parameters

DO.L positive or 0:

The function was successful. DO.L contains the length of the field and A0 the address of the first character of the field.

DO.L negative:

The procedure failed. DO.B contains one of the following error response codes:

NF: Field not present

BP: Bad parameter

3.1.4.3 Description

This procedure takes the parameter blocks as returned by TD.VM_GNR (or TD.CS_GNR) and the required field number. Using the bit map in the parameter block, it returns the length of the field and the address of the first data character.



3.1.5 CONVERT NUMBER FROM DIRECTORY FORMAT TO AN AUTODIAL SEQUENCE

3.1.5.1 Call parameters

D0.B : Action value = TD.CADS

D1.W : Length of number to be converted

A0.L : Address of number string to be converted

A1.L : Address to receive the autodial sequence

3.1.5.2 Return parameters

D0.L contains the length of the autodial sequence.

Error returns: None.

3.1.5.3 Description

This procedure converts a telephone number string, as held in the telephone directory to an autodial sequence.

The conversion algorithm is:

Characters 0 - 9 * and #	Unchanged
space and -	Removed
! ¤ " \$ % & ')	Converted to the pause character (.)
() + , . /)

3.2 Application chaining interface - overview

The TDA may be started by another application using the Director START APPLICATION procedure (ref 0.5.3). The initiating application passes parameters to the TDA via the registers defined below. The application must also pass the foreground to the TDA.

3.2.1 PARAMETERS

To select the Telephone Directory, the program name specified in the START APPLICATION procedure must be A.

To select the Computer Services directory, the program name is A CS

When the initiating application calls the Director procedure to start the TDA, it must have set up the parameters in the registers defined below. Director will ensure that these parameters are carried over to the TDA.