Policing of Automatic Redial Attempts on OPD

PSD 76.97.41

Each country has its own PTT regulations governing the automatic dialling of telephone calls. The software described in this document, Bad Number List Processor (BNLP), is an aid to complying with such regulations. Its use does not guarantee compliance with national PTT regulations: that is the responsibility of the software vendor.

0 DOCUMENT CONTROL

0.1 Contents List

0.2 Documentation Cross-Reference

1 OPD Kernel Specification. PSD 76.97.3.1 Issue 5/0.
R. R. Walton 29-Mar-85

2 OPD Director Facilities for Application Writers. PSD 76.97.3.2
Issue 3/1. R. M. Mahon. 8-Dec-84

3 OPD Layered Communications. PSD 76.97.11 Issue 2/1.
P. J. Vickers. 20-Nov-85

4 ICL One Per Desk Advanced Operations. Technical Publication
R51024/01. Section 5. International Computers Ltd 1984.

5 British Standard BS6789:1985 Apparatus with one or more particular functions for connection to the British Telecommunications public switched telephone network. Part 3: Apparatus with auto-calling, auto-answering and auto-clearing facilities. Section 3.1: Specification for auto-calling facilities.

6 British Standard BS6154: 1981 Method for Defining Syntactic Metalanguage.

7 Bad Number List Processor Component Product Design Document.
Issue 1/0. P. T. Brown [tbs]

8 Bad Number List Processor Alpha Test Specification.
Issue 1/0. M. J. Higham [tbs]

9 Register of PTT Requirements. as at 30-Aug-85
J. C. P. Burleigh, International Marketing Requirements, PCBC

0.3 Changes Forecast

Later versions of this product will not be expected to work with Mark 1 code (see Section 1.3). Hence the initialisation and TRAP-handling code will be omitted. Also see Section 12.

0.4 Changes from the Previous Issue

Although there has been much change of detail and organization of this document since the last issue, there have been no major changes. The scope of the document has been clarified, and an appendix added which summarizes the information pertinent to someone trying to use the component described herein.

1 GENERAL

1.1 Scope

It is advisable to read section 1.2 (Introduction) before this section.

To define the scope of this document accurately it is necessary to use several terms defined in British Standard 6789 (reference 5). These terms are printed in this section in bold type and their definitions are reproduced in section 1.3.

The BNLP does not provide the **initiation feature** or the **resolution feature**. It provides primitives which enable software which uses them to comply with rules governing the **repeat attempts feature.** These primitives only deal with the number of, and interval between, re-dial attempts. The requirements for the **initiation** and **resolution** features (and any other requirements imposed) must be policed by other components.

The BNLP does not provide the means to control **voice alert** repeat attempts : specifically, it does not check that the number being called is not a public emergency authority or the BT emergency (999) service. In the terminology of BS 6789 it only provides **mode 1** working.

The primitives described do not of themselves enforce compliance with any standards. Only correct use of them will ensure compliance. See the Appendix (Programmer's Guide).

Implementation and design details are not supplied, but the detailed interfaces to be used by other software components are given.

1.2 Introduction

One per Desk (OPD) software components which make use of the OPD data communications facilities may detect failed calls and, on behalf of the user, autonomously attempt to repeat the failed call. This involves instructing the OPD Telephone Handler to re-dial a number to which a previous call has failed. Such an action is called an Automatically Generated Re-dial Attempt (AGRA).

Post, Telephone, and Telecommunications authorities (PTTs) and Recognised Private Operating Agencies (RPOAs) in various countries have specified rules governing the making of AGRAs. This document describes a software component, the Bad Number List Processor (BNLP), which enables software to comply with these rules. It is written primarily in terms of the rules to be followed for the British Telecommunications (BT) Public Switched Telephone Network (PSTN). These are given in British Standard 6789 (reference 5). However, it is believed that the requirements of most other PTTs and RPOAs can be met by alteration of the repeat attempt pattern (the number of, and interval between, repeat attempts). The provisions for changing this pattern are covered in section 3.2.5 (Encoding of CMOS Entry). The requirements for some PTTs are covered in section 3.1.4 (Foreign Language Requirements).

1.3 Terms and Abbreviations used in this Document

AGRA — Automatically Generated Re-dial Attempt. An attempt, initiated by software, to place a call to a number to which a previous call attempt had failed.

BABT — British Approvals Board for Telecommunications.

BEM — Base Expansion Module

BEM code — This is the version of the Base issued in the BEM. It incorporates a version of Application Handler which returns a successful response to a GIVE SYSTEM VERSION NUMBERS call, and a version of Housekeeping which uses JSR calls to interface to BNLP.

BNL — Bad Numbers List. A list of telephone numbers to which the most recent attempts to establish data calls have failed.

BNLP — Bad Number List Processor. The software component which maintains the Bad Number List and provides interfaces to it.

BT — British Telecommunications

capsule — A unit approximately one inch by one inch by one-quarter of an inch that plugs into a Rompack.

CCC — Common Concern Codes. These are failure codes from T-Link.

EPC — Evade Police Character. A character preceding a dial string which causes circumvention of checks for re-dial attempts.

failed — for a discussion of what constitutes a failure, for the purposes of an AGRA, see section 3.2.2.

initiation feature — A part of auto-calling in which the instant at which to start sending the first digit of the network address is determined. (See reference 5).

JSR — Jump to Subroutine. A 68000 mnemonic. Non-Mark 1 code makes use of this method to call routines.

Mark 1 code

This is the version of the Base issued in Mark 1 machines. It incorporates a version of Application Handler which returns ERR.BP to a GIVE SYSTEM VERSION NUMBERS call, and a version of Housekeeping which uses TRAP calls to interface to BNLP.

Mark 2 code
This is the version of the Base issued in Mark 2 machines. It incorporates a version of Application Handler which returns a successful response to a GIVE SYSTEM VERSION NUMBERS call, and a version of Housekeeping which uses JSR calls to interface to BNLP.

mode 1 A mode of operation of the apparatus in which the time-out is not more than 1 min when an ineffective call condition is received. (See reference 5).

non-Mark 1 code
BEM code or Mark 2 code.

NTE Name Table Entry.

octet A collection of eight bits. (This term is exactly equivalent to an eight-bit byte.)

OPD ICL One Per Desk product.

PTT Post, Telephone and Telecommunications.

RAM Random Access Memory - this is read-write memory.

repeat attempts feature
A part of auto-calling in which the instant for making a fresh call attempt to a number over the PSTN is determined, after an ineffective attempt to the same number. (See reference 5).

resolution feature
A part of auto-calling in which the outcome of a call attempt is determined. (See reference 5).

ROM Read Only Memory

Rompack This is a unit about four inches by three inches by one inch which plugs into an OPD and into which are plugged capsules.

RPOA Recognised Private Operating Agency.

T-Link The component specified in reference 3.

TRAP A 68000 mnemonic. Mark 1 code software uses this method to call routines.

voice alert A verbal message. (See reference 5).

68000 The OPD processor chip

$ Values preceded by "$" are hexadecimal. For example, "$13" represents hexadecimal 13, decimal 19.

2 SUMMARY FOR MANAGEMENT

OPD components which use data communications, such as Messaging and Interfile, may make several attempts to transfer a body of data (a file or a message) from the originating OPD to the destination. If the first attempt fails, subsequent attempts may be initiated autonomously by software (the first having been initiated by the user). BABT has laid down rules (against which approvals tests will be performed) regarding the interval between such re-dial attempts, and the number of attempts which may be made to any particular destination (characterised by a telephone number).

There may be more than one component within an OPD making such re-dial attempts, and it is therefore necessary to supply an independent component which polices re-dial attempts on behalf of them all. This document describes the Bad Number List Processor (BNLP) which provides the application-independent interface to the Bad Number List.

Note that the BNLP does not initiate re-dial attempts, it merely checks that they are permissible within the rules it has available. The rules are encoded in a record in CMOS memory, and will have to be configured for each national variant of OPD. Information covering the requirements of currently considered nations is included in this document.

3 TECHNICAL DESCRIPTION

3.1 Overall Description.

3.1.1 Features and Characteristics.

The first release of BNLP is designed to be included in ROM capsules with components which automatically generate re-dial attempts. Later releases of OPD Base Functional Software will include the BNLP functions and interfaces described in this document. It was not possible to include BNLP functions (described in Issue 1/1 of PSD 76.97.11, an earlier issue of reference 3) in Mark 1 code, and this document describes a 'retro-fit' approach (implemented entirely in software), which provides the previously removed interfaces.

The BNLP initialisation code, which performs the retro-fit, will determine which type of OPD it is executing in. If it is in a Mark 1 OPD, it will perform the necessary actions. If it is in a non-Mark 1 OPD it will perform no action, the BNL functions having been otherwise provided.

3.1.2 Product Use.

Any component which makes Automatically Generated Re-dial Attempts (AGRAs) should include a copy of the BNLP in its ROM capsule. There is no provision for release of the component on cartridge. Aspects of the design circumvent any problems attendant on there being more than one copy of the BNLP in an OPD at any time. If BNL processing is not being done, it is not permissible to make AGRAs.

Immediately before making a call, an application which generates AGRAs should check with the BNLP that the call is permitted (the interfaces used for this are described in section 4). The response from the BNLP might indicate either that a call to the required number cannot be made yet, or that it should not be made at all. In the former case the application should wait before re-checking with the BNLP. In the latter, the operation which required the call (transfer of a message or file, for example) should be marked as having failed, and should not be rescheduled until the user has intervened explicitly to request another set of attempts.

If the call proceeds, the application should monitor the outcome of the call, to determine whether the call was successful (in network terms - see s. 3.2.2 below). If it was, the BNLP should be notified of success. This will cause it to remove any entry for that number currently held in the BNL. It is recommended that this notification be used even if the call was not an AGRA. A call that succeeds in network terms may be a failure as far as the application is concerned.

If the call fails in network terms, the BNLP should already have been updated (s. 4.2.2.2). Numbers are checked by the BNLP in a normalised form, so that a given number is recognized regardless of pauses, etcetera. For a more detailed explanation of normalization see section 4.2.1. A detailed description of the use of the BNL interfaces is given in section 4.2.

3.1.3 Limitations

Software components which use AGRAs and which handle their own communications must contain their own criteria for call success and failure. Some guidance is given in section 3.2.2 for components which use the T-Link interfaces. Later releases of T-Link will provide BNL processing on behalf of applications using T-Link for communications.

3.1.4 Foreign Language Requirements.

3.1.4.1 Language Requirements.

The only natural language text processed by the BNLP is that contained in telephone numbers. The BNLP must recognise and process the character codes for the digits "0" to "9" ($30..$39), "A" to "D" ($41..$44), "*" ($2A), "#" ($CA) and "X" ($58).

For USA variants, the character "#" is encoded $23.

The BNLP component produces no text messages. The Housekeeping Clear Call-Failed Records feature (ref 4) makes use of natural language text and will have to be changed for each foreign language variant.

3.1.4.2 Foreign PTT Requirements.

In addition, different territories will have different rules governing the number and frequency of re-dial attempts. These rules will be encoded in a CMOS record which will be inspected by BNLP on power-up and reset (s. 3.2.5). If there isn't a CMOS record at power-up, or if the record is deemed invalid, a default record is written to CMOS. The contents of this default record will vary between territories.

The requirements which are currently known are as follows (see reference 9) :

Country	Max. Number of Retries	Min. Interval (in seconds) between Retries
Australia	3	30,180,180
Denmark	9	none specified
Eire	3	5,60,60
Hong Kong	3	none specified
New Zealand	0	-
Norway	14	5,60,60,60,...

South Africa	10	60,60,60,...
Sweden	9	60,60,60,...
United States of America		no restrictions

3.2 Detailed Descriptions.

3.2.1 Use of Automatically Generated Re-dial Attempts.

Any software component which initiates background data calls on behalf of a user may have criteria determining the success or the failure of the call. In the case of a Messaging application, for example, the measure of success of a call is that the message was received and stored by the destination machine, entire and unchanged. Underlying this, there may be many indicators of failure, which may occur during different phases of the call. Each indicator may suggest a different re-dial strategy to the application (including not trying at all).

Most PTTs and RPOAs specify limits to AGRAs, and it is the purpose of the BNLP to police them within these limits. However, these limits do not specify what might be an optimal strategy for any particular application or (sometimes) for any particular network. Each application may, therefore, adopt its own strategy for AGRAs, and some notes for guidance are given in section 3.2.4 of this document. Where the strategy chosen conflicts with the attachment rules of the RPOA or PTT, correct use of the BNLP interfaces will ensure compliance with those rules.

3.2.2 Criteria for Call Success and Failure.

A call may 'fail' at many stages. It may be during dialling (because a task of higher priority has seized the line), during physical connection establishment (because the called party did not comply with V.25 procedures) or at any subsequent stage. In general, attachment rules are concerned solely with calls which fail properly to establish physical connection. This may be for a variety of reasons, such as:

- the number dialled is incorrect

- the called station is engaged

- the called station has a faulty modem

- there is a network routing fault.

(See page 6 of reference 5.) Other types of call failure are not of concern to the ROPA or PTT. For example, a file transfer application would class a call as a failure if the distant machine's file store were full, although to the network the call would be classed as successful. The application may well block subsequent calls to that number for the purposes of file transfer, because they would have a high probability of failure. Another application making calls to the same number might be successful (because it was retrieving, rather than sending, files). Thus some indicators of failure are of interest only to one application, and should not be used to block calls made by others.

It is often not possible for an application to determine the real cause of failure of a call. For example, unless the underlying modem hardware is equipped with busy-tone detection, an application cannot be informed that the called station was engaged. Applications must, therefore, use the returned failure indication conservatively, assuming where doubt exists that the call failure was due to physical connection establishment problems, and that the BNLP should be notified of the failure. Failure to do this may result in failure to achieve approval to attach OPD to a particular network, or at least result in a prohibition of use of the offending application in an OPD connected to a particular network.

An application which operates at the data link layer (or equivalent) may use the reception of a recognised sequence of characters (such as a frame header) to indicate call success. The further from the physical layer that an application operates, the more pessimistic must be its assessment of the causes of call failure, except where very explicit indications of failure are provided by the underlying layers.

Examples of indicators of call failure which are of common concern (common concern codes - CCCs) to all applications have already been given. The remainder of this section lists the failure indications returned by T-Link (ref 3), which should be interpreted by an application as physical connection establishment failures requiring an addition (or update) to be made to the Bad Number List, as described in section 4.

Error code	Description
$45	Link layer detected modem fault. This may occur at the start of a call because of V.25 failure. If data has already been received through the T-Link Link layer, then this is not a CCC.
$46	Timeout matured in Link layer. As for code $45.
$4B	Link request retransmission count exceeded. Often caused by network noise (or engaged tones) causing the modem to indicate connection.
$56	Physical layer failed to connect.
$09	Modem channel failed after call connection. As for code $45
$18	Error reported from Telephone Handler on autodialling.

3.2.3 Limitations on Re-dial Attempts.

The rules imposed on AGRAs by PTTs and RPOAs vary widely. This document discusses those specified for the BT PSTN in ref 5, which describes the conditions of test, detection of equipment engaged tone and other matters. The contents of ref 5 will not be reproduced here and the reader is urged to consult that document.

Consideration of the operating environment of OPD, and its use for both data and voice communications, has led to the conclusion that an adaptation of call pattern B (as defined in reference 5) will be used in releases for the BT PSTN.

Call attempt	Minimum duration between attempts for pattern B
Initial attempt	
}	5s
1st repeat	
}	2min
2nd repeat	
}	2min
3rd repeat	
}	2min
4th repeat	
}	2min
5th repeat	
}	2min
6th repeat	
	End

The call pattern to be used for the BT PSTN will impose a 5 minute pause between all call attempts, up to a maximum of 7 attempts (6 repeats). Note that the imposed minimum pause need not be the pause used by applications - as described in the next section they may use longer, and varied pauses.

3.2.4 Choice of Intervals between Re-dial Attempts.

This section is advisory only. It has no bearing on the rest of the document

Modelling of call patterns and re-dial attempts has led to the conclusion that the optimum re-dial strategy for applications such as Messaging or File Transfer is one of random exponential backoffs. Put simply, this means that the interval between subsequent attempts should increase (almost) exponentially, and that the intervals themselves should be randomised within given limits.

A very simple case where a fixed-interval redial strategy will fail is that in which two machines commence dialling each other at the same time. They would repeatedly fail to make connection (because the network would return an equipment busy signal) at (for example) 18:00, 18:05, 18:10, 18:15 and so on. If the intervals were randomised to some time between 5 and 10 minutes, the probability of collision would be considerably lower.

The strategy of increasing the interval between subsequent call attempts arises from the fact that, as a multi-function workstation, the call-holding pattern differs substantially from that of either a voice-only or a data-only station. When used for terminal access, data calls may well exceed an hour's duration, and the choice of a fixed-interval repeat strategy might well exhaust all repeat attempts within that period. It is therefore recommended that the intervals between repeat attempts be set as follows:

Initial attempt				
	}	5 - 10 min	(randomised)	delay
1st repeat				
	}	15 - 30 "	"	"
2nd repeat				
	}	60 - 90 "	"	"
3rd repeat				
	}	240 - 270 "	"	"
4th repeat				
	}	360 - 390 "	"	"
5th repeat				
	}	480 - 510 "	"	"
6th repeat				
		END		

This strategy provides seven attempts in a minimum of 19hrs 20mins and a maximum of 21hrs 40mins. Modelling on information currently available has shewn that, with very high probability, the call will be successful by the third attempt (2nd repeat).

3.2.5 Encoding of CMOS entry.

As described above, the rule to be policed by the BNLP will be held in CMOS so that it may be configured. When a BNL is created (because there is none extant), BNLP will fetch the rule from CMOS record 24. It will be stored in the memory segment containing the BNL. If BNLP detects that there is no CMOS record 24, it will create one with default values (which will vary for different national variants of the code). If there is an entry in CMOS record 24, and it does not conform to a specified format, it will be deleted and overwritten with the default value as described.

The aim of these procedures is to allow the rule to be altered if circumstances should make that desirable, but only with the use of a 'special tool' (Configurator or some other program created expressly for this purpose). (See section 4.1.1.3 of reference 5.) Additionally, it makes the rules being followed available to an application which may then check <u>before</u> making a re-dial attempt that the attempt is permitted or <u>not</u>.

The format of CMOS record 24 is given below, using the conventions of ref 6:

```
Re-dial_rule = " BNLATTEMPTS", max attempts, {attempt interval},
               checksum
               (* The first field of the record serves as a
                  human-readable identifier, for use mainly by
                  development and support staff. *);

max_attempts = word
               (* This is the maximum number of redial attempts,
                  and is limited to 65535, which is excessive, and
                  in practice must be less than 11. For a value of
                  n, this permits n+1 attempts in total. *);

attempt_interval = word
                   (* This specifies the minimum number of seconds
                      before the re-dial attempt. The value of max_
                      attempts determines the number of attempt_
                      interval entries in the record. *);

checksum = word
           (* The checksum will be computed by the BNLP and checked
              against this value. If they do not correspond the
              default value will be used as described above. The
              aim is to ensure that a 'special tool' (rather than
              just a BASIC program) is required to change the
              entry. *);
```

As an example, the call pattern to be followed for the United Kingdom would be encoded as follows:

```
$7E424E4C 41545445 4D505453 0006012C 012C012C 012C012C 012C302A
                             \  /\  / \  /\  / \  /\  / \  /\  /
   B N L  A T T E  M P T S    \/  \/   \/  \/   \/  \/   \/  \/
                               |   |____|___|____|___|____|   |
                               |              |               |
                      number of re-tries      |          checksum
                  (ie number of intervals)    |
                                     300 seconds (5 mins.) each
```

Note that if max_attempts is given a value greater than $000A, the CMOS entry will be overwritten by BNLP. That is, call pattern C of ref 5 can not be supported.

The reason for making the repeat attempts pattern configurable is to allow more flexibility. The BNLP must work in OPDs which have not been "prepared" by the inclusion of CMOS record 24 (after a

CMOS memory corruption for example), so the code must contain a default value for the record. Each country has different requirements, so there must be national variants of the code. However, after release of BNLP in a given country, it may become apparent that a different repeat attempts pattern would be more favourable. The CMOS record will allow this change without having to re-release BNLP. The checksum incorporated in the record will ensure that only patterns supplied by ICL can be used. A policy decision has been taken that on a device such as the OPD it is not unreasonable to require user intervention after ten unsuccessful retries to a number, hence the limit of ten intervals in the CMOS record.

3.2.6 BNLP Initialisation Procedures.

An initialisation procedure for BNLP will be auto-started on power-up or reset of the machine. This will permit the BNLP to function correctly in Mark 1 OPDs, and will avoid clashes if there is more than one copy of the BNLP in an OPD. It will not perform initialisation in a non-Mark 1 OPD. The procedures are required because the version of Housekeeping in Mark 1 machines uses an obsolete TRAP interface to the BNLP.

The initialisation consists of altering Director's tables so that TRAP calls to TLINK (" L   ") are intercepted by the BNLP. If the initialisation has taken place, a Name Table Entry (NTE) will have been created with the name LREDIRECTOR and having as its value the address of the entry point of TLINK.

3.2.7 BNLP TRAP Entry Processing.

The position after initialisation is as follows : all TRAP and START APPLICATION calls directed at TLINK will be intercepted by BNLP. If the call is a TRAP to T.TLINK with value L.REMNUM or L.GETNTH in D1.B, the call will be dealt with by BNLP. All other calls will be passed unchanged to the address contained in NTE LREDIRECTOR. If, for some reason, the entry cannot be found, return will be made to the caller with value ERR.NF in D0.L.

3.2.8 Establishing the need for BNL Processing

This paragraph applies to T-Link only. For other protocols, see their PSDs.

Later releases of the T-Link protocol will add numbers to, and remove them from, the BNL on the application's behalf. In a Mark 1 OPD (which will not support the GIVE PROGRAM HEADER FIELD (D.GIVPROG) Director interface), an application may determine whether the T-Link it is using will perform this service by testing the response to a LOAD PROGRAM call (ref 2 s. 4.9.3) using " LBUFFERMGMT" as the name of the target program. If the response is ERR.NF, then the application must take responsibility for BNL interactions.

In non-Mark 1 OPDs, the application should request the header field with identifier 2 from program " L " using the GIVE PROGRAM HEADER FIELD interface. This field, if returned, will be one octet in length, and its contents will be undefined. If the field is returned, T-Link will handle interactions with the BNL on the application's behalf.

3.3 Environmental, Portability, and Extensibility Requirements

No requirements established.

The design of the CMOS entry only allows repeat attempt patterns with 10 or less retries.

3.4 Security

There is a software 'signature' on the CMOS record which should prevent unauthorized alteration of it.

4 INTERFACES

There are two sets of applications interfaces to the BNL described in the following sections 4.1 and 4.2. Those described in 4.1 were defined in Issue 1/1 (an earlier issue) of ref 3. They are provided solely for the Mark 1 Housekeeping Clear Call-Failed Records feature (ref 4), which has been written to use TRAP interfaces. The second set of interfaces, described in section 4.2, are those to which all new applications should be written. They allow for the BNLP to be held in paged ROM.

4.1 TRAP Interfaces.

These interfaces will not be supported beyond Mark 1 machines. Any attempt to use them in later machines will return with ERR.BP.

To use these calls, an application should include the file **TLINK.L9VALUES.DG** which contains the required action values. The calls themselves are made by obeying the instruction:

TRAP #T.TLINK

with a particular action value in DO.B and additional parameters as described below. Processing of the call may consume up to 256 octets of the application's stack, and care should be taken to ensure that sufficient stack is available when the call is made.

4.1.1 REMOVE NUMBER FROM BNL

TRAP Name:	T.TLINK
Action Value (DO.B):	L.CONTROL

Additional Call parameters:

D1.B	L.REMNUM
D2.B	Length of telephone number string
AO.L	RAM address of the buffer holding the telephone number string

Return values:

DO.L	zero if the number has been removed
	ERR.NF if the telephone number was not found in the BNL

4.1.2 GET NTH ENTRY FROM BNL

TRAP Name: T.TLINK
Action Value (D0.B): L.CONTROL

Additional Call parameters:

D1.B L.GETNTH

D2.L Entry to be obtained. D2.L has the value 0 for the first entry, 1 for the second, and so on. (This is different from the JSR interface.)

D3.B Length of buffer into which the entry in the BNL (if found) will be copied.

A1.L Address of the buffer into which the entry in the BNL (if found) will be copied.

Return Values:

D0.L The length of the number returned.

ERR.NF if there was no nth entry.

ERR.OR if the buffer length was insufficient. (In this case, as much of the entry as will fit will have been copied to the user's buffer.)

If an nth entry is found in the BNL, it will be copied into the user's buffer. The format will be as follows (octet numbers are given as offsets from the value in A1):

octets 0..3	octet 4	octet 5	octets 6..n
Last Attempt Time	Attempts	Length of Number	Number

The Last Attempt Time is in seconds since 00:00hrs, 1-Jan-1970.

The length of the buffer should therefore be six octets more than the length of the number. In general, numbers are limited to 40 characters, including embedded space and pause characters, so a buffer of 46 octets may often suffice.

4.2 JSR Interfaces

All applications requiring BNL processing functions should use the interfaces described below. Note that the TRAP interfaces described in s. 4.1 will not be supported beyond Mark 1 machines.

To use these interfaces, an application should include the file **BNLP.B9.DG**, which contains the required action values.

The application should use the Director LOAD PROGRAM interface (ref 2), with A0 pointing to a buffer containing the name " BNLP ". On return from the call, A0 will contain the entry point of the BNLP.

Calls to the BNLP are then made with the registers set as described below, and by executing a JSR (Ax), where 'Ax' is the address register containing the address of the entry point of the BNLP. The calling application should ensure that there is at least 256 octets of its stack available when the JSR is executed, since this can be used by the BNLP for storing registers and for workspace.

Once the application has ceased to use the BNLP (for example at the end of a call), it should use the Director RELEASE PROGRAM (ref 2) interface, which de-registers its use of the BNLP.

4.2.1 Normalisation of Telephone Numbers.

As adduced in s. 3.1.2, all comparisons of telephone numbers during BNL processing operate on normalised forms. This section describes the normalisation carried out before searches and comparisons are made.

> If there is an Extension field included in the string (that is, some characters following an 'X' in the string) it is ignored.
>
> All characters other than "0".."9" ($30..$39), "A".."D" ($41..$44), "*" ($2A) and "#" ($CA in UK, $23 in USA) are discarded.

Note that entries are not held in a normalised form, but in the form in which they were first added to the BNL. This ensures that entries presented to the user through the Housekeeping interface will be familiar. Extension fields are not stored in the BNL.

Phone numbers up to 255 characters in length can be held in the BNL.

4.2.2 Sequence of use of BNLP interfaces.

This section describes the sequence in which the BNLP interfaces should be used by an application making a call. For calls which have been initiated manually, there is in the BT PSTN no requirement for a test to be made for previously failed calls to that number. The first attempt to transmit a message (from the OPD Messaging application) is an example of such a call. For simplicity, implementors may chose to ignore this distinction between manually and automatically initiated calls. However, the sequence of events to be followed in either case is given below.

4.2.2.1 Manually initiated calls.

There is no requirement for any interaction with the BNL for manually initiated calls. However, after a successful manually initiated call the application should use the REMOVE NUMBER FROM BNL interface described in section 4.2.5. (It is not neccessary for applications which do not generate AGRAs to do this.)

If the call is to be made through a version of TLINK which will handle BNL processing on the application's behalf (see s. 3.2.2), then the number must be preceded by an Evade Police Character (see reference 3 for further information), which prevents a check being made on the number before dialling. If the call succeeds, T-Link will remove an entry for the number in the BNL if one exists.

Note : If a number is removed from the BNL (by use of the Housekeeping Clear Call-Failed Records utility) while a repeat attempts sequence is in progress, the effect is to begin a new sequence. This is, in effect, algorithm C of the British Standard (reference 5). See the appendix.

4.2.2.2 Automatically initiated calls.

Immediately before a call is made, an application must ascertain whether it is permitted, and reserve space for an entry in the BNL. If there is no room to expand the BNL to take the new entry, then the call should not be made (in circumstances of such acute memory shortage it is unlikely that the call would succeed at all). The application therefore uses the ADD NUMBER TO BNL interface (s. 4.2.3), which will create an entry for the number if none exists, and will return the status of the number if there is already an entry. The status will indicate whether the call may proceed at once, may proceed later, or may not proceed at all. If the call may proceed, it will also increment the count of failed attempts to the number, and will re-set the time at which the last call to the number has been made.

If the call fails, the application need take no action, since the number has already been added to the BNL (or its entry updated). If it succeeds, however, the REMOVE NUMBER FROM BNL interface (s. 4.2.4) should be used instead.

The SEARCH BNL FOR NUMBER interface (s. 4.2.5) is provided to allow an application to make an enquiry about the status of a number. The

GET NTH ENTRY FROM BNL interface (s. 4.2.6) is provided for the Housekeeping Clear Call-Failed Records feature (reference 4), which presents a list of BNL entries to the user. It is available for use by other applications without restriction.

4.2.3 ADD NUMBER TO BNL

Action Value (DO.B) B.ADDNUM

Additional call parameters:

D1.B	Length of telephone number string
AO.L	RAM address of buffer holding telephone number string.

Return values:

DO.L	zero if the number has been added or the entry updated - the call may proceed
	ERR.OM if there is no room for the new entry.
	ERR.OR if no more calls may be made to this number (until its entry is cleared by manual intervention).
	ERR.IU if the call may not be made yet (in which case see D1.W).
D1.W	If the response in DO.B is ERR.IU, D1.W contains the number of seconds which must elapse before another call to this interface could be successful.

4.2.4 REMOVE NUMBER FROM BNL

Action Value (DO.B) B.REMNUM

Additional call parameters:

D1.B	Length of telephone number string
AO.L	RAM address of buffer holding telephone number string.

Return values:

DO.L	zero if the number has been removed
	ERR.NF if no entry for the number could

be found.

4.2.5 SEARCH BNL FOR NUMBER

Action Value (D0.B) B.FINDNUM

Additional call parameters:

D1.B	Length of telephone number string
D3.B	Length of buffer into which the number's entry in the BNL is to be copied.
A0.L	RAM address of buffer holding telephone number string.
A1.L	Address of the buffer into which the number's entry into the BNL will be copied.

Return values:

D0.L	zero if the number has been found and the entry copied to the caller's buffer ERR.NF if no entry for the number could be found. ERR.OR if the buffer supplied was too short for the data. (In this case, as much of the entry as will fit will have been copied to the user's buffer.)

If an entry for the specified number is found in the BNL, it will be copied into the user's buffer. The format will be as follows (octet numbers are given as offsets from the value in A1):

octets 0..3	octet 4	octet 5	octets 6..n
Last Attempt Time	Attempts	Length of Number	Number

The Last Attempt Time is in seconds since 00:00hrs, 1-Jan-1970.

The length of the buffer should therefore be six octets more than the length of the number. In general, numbers are limited to 40 characters, including embedded space and pause characters, so a buffer of 46 octets should always suffice.

4.2.6 GET NTH ENTRY FROM BNL

Action Value (D0.B): B.GETNTH

Additional Call parameters:

D2.L Entry to be copied. D2.L has the value 1 for the first entry, 2 for the second, and so on.

D3.B Length of buffer into which the entry in the BNL (if found) will be copied.

A1.L Address of the buffer into which the entry in the BNL (if found) will be copied.

Return Values:

D0.L The length of the number returned.

ERR.NF if there was no nth entry.

ERR.OR if the buffer length was insufficient. (In this case, as much of the entry as will fit will have been copied to the user's buffer.)

If an nth entry is found in the BNL, it will be copied into the user's buffer. The entry remains in the BNL. The format will be as follows (octet numbers are given as offsets from the value in A1):

octets 0..3	octet 4	octet 5	octets 6..n
Last Attempt Time	Attempts	Length of Number	Number

The Last Attempt Time is in seconds since 00:00hrs, 1-Jan-1970.

The length of the buffer should therefore be six octets more than the length of the number. In general, numbers are limited to 40 characters, including embedded space and pause characters, so a buffer of 46 octets should always suffice.

5 STANDARDS

The product described in this document is designed to allow OPD applications which make data calls to conform to standards governing AGRAs. In the United Kingdom the appropriate standard is BS 6789 part 3, section 3.1 (reference 5).

This product only covers the number of, and intervals between, AGRAs. The British Standard also lays down requirements in other areas, these must be addressed by other components. In the language of the standard, the OPD always works in Mode 1, and call pattern B is complied with. The initiation and resolution features described in the standard are not provided or monitored by the component described in this document.

With regard to section 7.2.3 of the British Standard, the algorithm to be followed can be any of the three options given. The implementation of the chosen algorithm is the responsability of the software that calls the primitives. (See the appendix.)

The standards governing AGRAs in countries other than the UK are held by the International Marketing Requirements team (reference 9).

6 PRODUCT PERFORMANCE

No noticeable degredation to user-perceived response times.

7 HARDWARE CONFIGURATION AND SOFTWARE ENVIRONMENT

The BNLP is designed to run in both Mark 1 and later OPDs (Mark 1 with BEM, and Mark 2). It uses the software interfaces provided by Kernel (ref 1) and Director (ref 2).

The CMOS record consumes up to 36 octets. The Bad Number List itself consumes a variable amount of mobile RAM. A 512-octet block can hold 18 20-digit numbers. There is no software upper limit on the amount of RAM consumed by the BNL. The code is expected to consume less than 2K of ROM. The routines run on the caller's stack, and require less than 256 octets of stack space.

8 COMPATIBILITY WITH OTHER ICL PRODUCTS.

It is a requirement that releases of Housekeeping in later versions of OPD Base software (BEM and Mark 2) no longer use TRAP interfaces to T-Link to access the BNL, but instead use LOAD PROGRAM and JSR interfaces through " BNLP   ".

It is assumed that releases of the T-Link protocol which provide BNL processing on behalf of an application will also provide memory management through JSRs to a program named " LBUFFERMGMT", and that they will have a program header field with identifier 2 and length one octet. (See reference 3)

9 RELIABILITY, RESILIENCE, MAINTENANCE AND USABILITY.

9.1 Reliability

No requirement established.

9.2 Resilience

The Bad Number List itself is checksummed, which aids detection of (unintentional) memory corruption.

9.3 Issue, Installation, and Maintenance

This product will only be issued as an integral part of other ROM-based products. As such, there is no field-maintenance possible, and errors found will be corrected by further releases of the product. The CMOS record can be altered in the field to give different repeat attempt patterns, but a special software 'tool' will be required (usually Configurator).

9.4 Usability

Usability issues are confined to the user interface to the BNL, described in ref 4. There are no marketing requirements.

10 VALIDATION AND PERFORMANCE PROVING TESTS.

Alpha-tests will be specified in a separate document (ref 8). Validation will be performed by the Personal Computers Business Centre validation unit. Ultimately, the tests specified in Appendixes A and B of reference 5 will be undertaken.

11 TEST EQUIPMENT AND TEST SOFTWARE

Testing will require some test harnesses to be written in BASIC. These will be used to exercise the primitives. Testing should also be performed on applications which use the BNLP to check for correct use of the primitives.

No special test hardware will be neccessary.

12 ENHANCEMENT CAPABILITY

There are no enhancements planned for BNLP. A possibility is to allow the user to set a maximum size for the BNL. The design of the current implementation is modular, allowing enhancements and changes to be made at least possible cost. In particular, the rules for the number of, and interval between, repeat attempts is easily changed.

13 DOCUMENTATION

This document is intended to form the basis of a chapter of a reference guide for implementors. User documentation for the Housekeeping Clear Call-Failed Records interface is already provided in ref 4.

Appendix PROGRAMMER'S GUIDE

Establishing the requirement for BNL Processing

No program may generate AGRAs unless BNL processing is being done.

If a program which generates AGRAs will only be run on OPDs which contain a version of T-Link which handles BNL processing, and uses T-Link to generate all its calls, then it need do nothing special. All other such programs, including those for which there is any doubt about the matter (for example, ones in capsules) must at least check that T-Link deals with it.

The procedure to follow is described in section 3.2.8.

If T-Link does not do the processing, or if T-Link is not present and no other protocol is doing it, then the program can only proceed with AGRAs if it does the processing itself. To ensure that it can do so, it should include a copy of the BNLP code.

The rest of this appendix only applies to programs which intend to call the BNLP.

BNL processing

The interfaces to support BNL processing, and the method of calling them, are specified in section 4.2. Use of LOAD and RELEASE PROGRAM are assumed henceforth. The first attempt to any number has a different status from any subsequent re-tries, see the next section. Any re-try attempt should follow the following sequence:

i) call ADD NUMBER with the relevant parameters

ii) if the response is ERR.OM or ERR.OR, give up

iii) if the response is ERR.IU, then wait the returned number of seconds, or longer, then re-call ADD NUMBER (ie go to step i))

iv) only when having received a zero response from ADD NUMBER, make the call - **immediately**. It is vital that there is no delay between the successful return from ADD NUMBER and the dial attempt. All other necessary work must have been done before the call to ADD NUMBER. No screen i/o, potential wait for resources, etcetera should intervene.

v) if the call is a failure in network terms (see section 3.2.2) then no action need be taken.

vi) if the call succeeds, then call REMOVE NUMBER.

Algorithms

What may be the first try of a number as far as the program is concerned may not be the first as far as the machine is concerned - another program may already have tried it and failed. Consequently, the program must implement one of the three algorithms presented in section 7.2.3 of the British Standard (reference 5). The following paragraphs describe the options for the first try of a number.

If algorithm A is being implemented, the sequence of events is:
i) proceed for the first number as for the re-tries (ie as in the previous section).

If algorithm B is being implemented, the sequence of events is:
i) make dial attempt
ii) if network success, REMOVE NUMBER (ignore failures)
iii) if network failure, SEARCH BNL FOR NUMBER
iv) if found number, do nothing, start re-try sequence
v) if did not find number, ADD NUMBER (ignore failures)

If algorithm C is being implemented, the sequence of events is:
i) REMOVE NUMBER (ignore failures)
ii) proceed as for re-tries (previous section)

Note : Octet 4 returned by GET NTH ENTRY and by SEARCH BNL FOR NUMBER is the number of attempts not the number of re-tries. (Even this is slightly misleading, because if algorithm B is being used, initial attempts of repeat sequences other than the first do not count.)