Bellcore

Bell Communications Research

89-10

TECHNICAL ADVISORY
TA-STS-000780
Issue 2, March 1989

Generic Requirements for a Bellcore 3270 High Level Language Application Program Interface (HLLAPI)

Technical Advisory
TA-STS-000780
Issue 2, March 1989

This document replaces: TA-STS-000780
Issue 1, March 1988

TECHNICAL ADVISORY
NOTICE OF DISCLAIMER

This Technical Advisory is published by Bell Communications Research, Inc. (Bellcore) to inform the industry of Bellcore's preliminary view of generic requirements for a 3270 High Level Language Application Program Interface (HLLAPI) and to solicit industry comments thereon. It is a preliminary document subject to review and change. As such, future issues of this information may differ extensively in content and format.

Bellcore reserves the right to revise this document for any reason, including but not limited to, conformity with standards promulgated by various agencies, utilization of advances in the state of the technical arts, or the reflection of changes in the design of any equipment, techniques or procedures described or referred to herein.

BELLCORE MAKES NO REPRESENTATION OR WARRANTY, EXPRESS OR IMPLIED, WITH RESPECT TO THE SUFFICIENCY, ACCURACY, OR UTILITY OF ANY INFORMATION OR OPINION CONTAINED HEREIN. BELLCORE EXPRESSLY ADVISES THAT ANY USE OF OR RELIANCE UPON SAID INFORMATION OR OPINION IS AT THE RISK OF THE USER AND THAT BELLCORE SHALL NOT BE LIABLE FOR ANY DAMAGE OR INJURY INCURRED BY ANY PERSON ARISING OUT OF THE SUFFICIENCY, ACCURACY, OR UTILITY OF ANY INFORMATION OR OPINION CONTAINED HEREIN.

This document is not to be construed as a suggestion to any manufacturer to modify or change any of its products, nor does this document represent any commitment by Bellcore, any Bellcore Client Company (BCC)† or any Regional affiliate thereof to purchase any product whether or not it provides the described characteristics.

Readers are specifically advised that each BCC or Regional affiliate may have requirements different from the 3270 PC HLLAPI generic requirements described herein. Therefore, any vendors, or manufacturers of products should communicate directly with a BCC or Regional affiliate thereof to ascertain that company's needs, specifications, and actual requirements.

Nothing contained herein shall be construed as conferring by implication, estoppel or otherwise any license or right under any patent, whether or not the use of any information herein necessarily employs an invention of any existing or later issued patent.

Bellcore does not recommend products and nothing contained herein is intended as a recommendation of any product to anyone.

† Bellcore Client Companies are Ameritech, Bell Atlantic, BellSouth, NYNEX, Pacific Bell, Southwestern Bell, U S West, Southern New England Telephone, and Cincinatti Bell, Inc.

If further information regarding technical content is required, please contact:

District Manager
Computing Environments
Bell Communications Research, Inc.
6 Corporate Place
Room PYA-1J-277
Piscataway, New Jersey 08854
(201) 699-3462

For other information, contact:

Information Exchange Management
Bellcore
445 South Street - Room 2J146
P. O. Box 1910
Morristown, N. J. 07960-1910

CONTENTS

(BLANK PAGE)

GENERIC REQUIREMENTS FOR A BELLCORE 3270 HIGH LEVEL LANGUAGE APPLICATION PROGRAM INTERFACE (HLLAPI)

1. INTRODUCTION

1.1 Purpose and Scope of Document

This Technical Advisory documents the proposed generic requirements for a Bellcore 3270 High Level Language Application Program Interface (HLLAPI) for 3270 connected workstations or processors. These workstations or processors can be based on any architecture and run any operating system (e.g., DOS, OS/2™ , UNIX™). These generic requirements are based on the following:

- The function calls, interface definition, and operating environment of earlier versions of HLLAPI (up to and including Version 3.00) initially designed for use on 3270 PC type workstations, and
- Current and future needs of Bellcore CDS (Centrally Developed Systems) developers. These needs were identified through a forum of Bellcore developers either currently using HLLAPI or expressing an interest in future development.

This Technical Advisory and the proposed generic requirements were motivated by requests to Bellcore from the Regions/Bellcore Client Companies (BCCs) to support 3270 connected workstations or processors from different vendors. Since the 3270 HLLAPI programming interface was historically different for most vendors, Bellcore was required to maintain a different application version for each supported workstation or processor vendor.

One of Bellcore's objectives is to design software that is hardware independent, thus removing application dependencies for a specific vendor, allowing our customers to purchase equipment most suitable to their individual requirements. In keeping with these objectives, this document is a high level specification of the minimum programming interface and support requirements. In addition it is Bellcore's aim to design software using common, industry standard interfaces. Each Region/BCC may have requirements different from those described herein; therefore, a vendor may wish to communicate directly with the Regions/BCCs before deciding the extent to which it implements these 3270 HLLAPI generic requirements.

1.2 Organization

Attachment 1 contains additional generic requirements for a 3270 HLLAPI (based on the needs of Bellcore CDS developers) currently not supported in either existing or earlier versions of 3270 HLLAPI (up to and including Version 3.00). These requirements are currently needed to support customer requests for both enhancements to existing and new Bellcore applications. Note that this list of 3270 HLLAPI enhancements will evolve as additional functionality is needed by CDS developers. Attachment 2 provides a brief summary of each 3270 HLLAPI function call and is intended for reference only.

2. GENERAL INFORMATION

2.1 General Description of 3270 HLLAPI

3270 HLLAPI provides a set of functions that are embedded in the code of an application, that can access host sessions connected through the same 3270 connected workstation or processor. 3270 HLLAPI is a high level interface to a low level API integrated into the 3270 Control Software. Note that this control software was originally developed by IBM and known as the 3270 PC Control Program or 3270 Workstation Program. Whereas the low

OS/2 is a trademark of IBM Corp.

UNIX is a registered trademark of AT&T

level API can only be accessed through assembly language, 3270 HLLAPI calls can be embedded into the source code of a high level language (e.g., Pascal, C). Each session type (e.g., Host, PC or local processing, Notepad, Alternate, and Workstation Control) is supported under 3270 HLLAPI through the concept of "Presentation Spaces". A presentation space is a virtual screen buffer containing the entire contents of a single screen of information. Each connected host is only aware of its presentation space. All interaction with the host session is performed through its presentation space.

The 3270 HLLAPI functions were designed to support, through programming interfaces, all the functions available to the user through the keyboard under the 3270 Control Software. Some of the capabilities available through 3270 HLLAPI are as follows:

- *Unattended operation or Programmed Operator* - Any combination of keystrokes routinely entered by a user can be saved and played back at predefined times. Examples of the Programmed Operator concept include automated host Logon/Logoff and automated execution of 3270 HLLAPI based application programs.
- *Create a single input screen or view to multiple host applications* - This is useful in situations in which the user is entering identical or similar information into two or more host applications. 3270 HLLAPI can be programmed to provide a single view into multiple host applications and then populate the data fields of each application with the information entered by the user.
- *Data Extraction* - Transfer data between different applications resident on the same or different hosts. Data can also be transferred into a local processing or notepad session for use by an application resident on the local workstation.
- *File Transfer* - Allows files to be uploaded to the host or downloaded from the host into the workstation's local processing session. SEND and RECEIVE functions for VM/CMS and MVS/TSO should be provided.
- *Improve the host user interface* - Traditional host interfaces that are terse and uninformative can be modified using 3270 HLLAPI to display easy to read messages specific to an individual application. These messages may prompt for required input or display a more user friendly message suggesting a possible action by the user. Custom data entry screens for host applications can also be programmed to replace existing screens without modifying the host application.

2.2 Components of 3270 HLLAPI

3270 HLLAPI may have up to three parts:

1. Resident Interface Module (HLLAPI.EXE) - Executable portion of 3270 HLLAPI loaded prior to calling 3270 HLLAPI functions and remains resident in memory. (PC implementation only.)
2. Language Interface Module (LIM) - The LIM is the interface between the programming language and HLLAPI.EXE. A unique LIM is required for each supported high level programming language used to call 3270 HLLAPI functions. At a minimum a LIM should be provided for both C and Pascal. The LIM becomes a part of the 3270 HLLAPI application when "linked". (PC implementation only.)
3. 3270 HLLAPI Function Calls - High level calls that in turn call the respective low level API functions of the 3270 Control Software. Earlier versions of 3270 HLLAPI supports 44 functions divided into the following categories:
    - *Operator Services* - Support the interface between the 3270 HLLAPI application and HLLAPI.EXE in the PC implementation.
    - *Presentation Services* - Provide the ability to control presentation spaces. Some of the functions provided include search, copy, query, and connect/disconnect capabilities.
    - *Device Services* - Support 3270 HLLAPI interaction with the keyboard.
    - *Communications Services* - Provide file transfer capabilities
    - *System Services* - Provide access to the Operating System (OS) to invoke other, non-HLLAPI applications or OS commands.

- *Utility Services* - Provide functions to allocate storage or perform conversions between presentation space positions and device row/column coordinates.

2.3 Operating Environment

The 3270 connected workstation or processor should support a DFT (Distributed Function Terminal) connection to an IBM 3274/3174 or equivalent cluster-controller and should operate under a 3270 Control Software program. This Control Software should contain the low level API functions necessary to implement 3270 HLLAPI. 3270 HLLAPI should not require any host software in addition to the software necessary for the 3270 connected workstation or processor and its Control Software.

Note that when implemented on the PC, the resident portion of 3270 HLLAPI, HLLAPI.EXE, should not occupy more than 30KB of RAM.

3. FEATURES AND FUNCTIONALITY

3.1 3270 Control Software

A 3270 Control Software program with functionality equivalent to the IBM 3270 PC Workstation Program (IBM Part Number 74X9921 or 75X1088) is necessary in order to support the 3270 HLLAPI functions specified by the generic requirements set forth in this Technical Advisory.

Note: The IBM 3270 PC Workstation Program replaced the IBM 3270 PC Control Program V3.0.

3.2 Multiple DOS Session Support (PC implementation only)

Support for Multiple DOS sessions provided by the IBM 3270 PC Control Program Version 2.10 and higher is not included in these generic requirements. Operating System functionality should not be provided by the 3270 Control Software, but rather by the host Operating System of the 3270 connected workstation or processor. Bellcore CDS applications should not develop dependencies on the 3270 PC Workstation Program's method of supporting multiple DOS sessions.

3.3 3270 HLLAPI Interrupt Vector (PC implementation only)

On the PC, the 3270 HLLAPI interrupt vector is fixed at 7F hex. The 3270 HLLAPI V1.0 and V2.0 interrupt vectors, 44 hex and 45 hex, should also be supported for older applications using these versions of HLLAPI.

Note that all ongoing Bellcore application development should use interrupt 7F. It is also suggested that older Bellcore applications using interrupts 44 and 45 hex, should be upgraded to support interrupt 7F.

3.4 Session Types Supported

1. *Four Host sessions (PC implementation), Sixteen Host Sessions (UNIX)* - These host sessions should be able to connect to a 3270 Synchronous host. During customization of the 3270 Control Software, the user should be able to configure the number of host sessions.
2. *Two local Notepad sessions* - Historically provided by PC implementations of the 3270 Control Software for convenience to the user and as a local buffer for use by other 3270 HLLAPI applications running on the workstation. (PC implementation only.)
3. *One local processing session* - Provided to run the application software using embedded 3270 HLLAPI functions. This local session could operate under any operating system (e.g., DOS, OS/2, UNIX).
4. *Alternate Presentation Space* - Additional buffers that can be defined by a 3270 HLLAPI application. These alternate presentation spaces can be formatted as either a host or display native to the local workstation hardware. 3270 keystroke emulation and the 3270 synchronous data format should be supported in the alternate presentation space.

5. *Work Station Control (WS CTRL)* - Not currently supported under this standard except for the following two commands:

    - CA (Change Active window) - Allows the active screen profile and presentation space (session) to be changed.

    - QA (Query Active window) - Returns the short name of the window or presentation space currently connected in the specified screen profile.

    Since there is no industry standard for 3270 PC type workstations, Work Station Control may change as new hardware devices are introduced by hardware manufacturers. Application dependencies developed on today's 3270 workstations may require modifications in the future to support new hardware devices.

3.5 3270 HLLAPI Language Interface Module (LIM)

At a minimum, Language Interface Modules (LIMs) for the latest versions of the following languages should be provided by the hardware vendor.

1. Microsoft® C (PC implementation only)
2. Microsoft Pascal (PC implementation only)

3.5.1 3270 HLLAPI Functions Supported

At a minimum, the following 3270 HLLAPI functions and function numbers should be supported. Note that the 44 functions listed below are not sequentially numbered. Attachment 2 provides a brief summary of each function.

1. Function 1 - Connect Presentation Space
2. Function 2 - Disconnect Presentation Space
3. Function 3 - Send Key
4. Function 4 - Wait
5. Function 5 - Copy Presentation Space
6. Function 6 - Search Presentation Space
7. Function 7 - Query Cursor Location
8. Function 8 - Copy Presentation Space to String
9. Function 9 - Set Session Parameters
10. Function 10 - Query Sessions
11. Function 11 - Reserve
12. Function 12 - Release
13. Function 13 - Copy OIA
14. Function 14 - Query Field Attribute
15. Function 15 - Copy String to Presentation Space
16. Function 16 - WS Ctrl

Microsoft is a registered trademark of Microsoft Corporation

17. Function 17 - Storage Manager
18. Function 18 - Pause
19. Function 20 - Query System
20. Function 21 - Reset System
21. Function 22 - Query Session Status
22. Function 23 - Start Host Notification
23. Function 24 - Query Host Update
24. Function 25 - Stop Host Notification
25. Function 30 - Search Field
26. Function 31 - Find Field Position
27. Function 32 - Find Field Length
28. Function 33 - Copy String to Field
29. Function 34 - Copy Field to String
30. Function 35 - Define Presentation Space
31. Function 36 - Switch Presentation Space
32. Function 37 - Display Cursor
33. Function 38 - Display Presentation Space
34. Function 39 - Delete Presentation Space
35. Function 50 - Start Keystroke Intercept
36. Function 51 - Get Key
37. Function 52 - Post Intercept Status
38. Function 53 - Stop Keystroke Intercept
39. Function 54 - Get 3270 AID Key
40. Function 90 - Send File
41. Function 91 - Receive File
42. Function 92 - Invoke DOS Program
43. Function 93 - DOS Redirect
44. Function 99 - Convert Position to RowCol

Functions 90 (Send File) and 91 (Receive File) are not recommended for use in Bellcore CDS applications since these functions are expected to be included in other communications interfaces, such as the IBM OS/2 Extended Edition Communications Manager.

3.6 Format of 3270 HLLAPI LIM Function Call

The Language Interface Module (LIM) expects four parameters for each 3270 HLLAPI function call. If all four parameters are not required, the function should be able to disregard the excess parameters. Each 3270 HLLAPI function call should be in the following format:

HLLLIM (PARAM1, PARAM2, PARAM3, PARAM4) where HLLLIM designates the language specific (LIM).

3.6.1 Calling Parameters:

The following parameters are passed to the Resident Module when the 3270 HLLAPI function is called:

- PARAM1: 3270 HLLAPI Function Number - two byte, full word integer;
- PARAM2: Data String - string of characters or concatenated data items. This data string contains the information necessary to execute the function;

  Note: Additional data items will be required for asynchronous sessions. In order to preserve consistency with existing 3270 HLLAPI functions, this information should be concatenated to the existing data string rather than modifying the currently defined string.

- PARAM3: Data Length - two byte, full word integer - Length of the character string or concatenated list of data items;
- PARAM4: Presentation Space Position - Value associated with the IBM 3278/79 or equivalent Display Station terminal screen sizes to be emulated. Below is the range of screen sizes that should be supported for each type of session or display terminal:
  - PC - 1 to 2000 characters or bytes
  - Model 2 and Notepads - 1 to 1920 characters or bytes
  - Model 3 - 1 to 2560 characters or bytes
  - Model 4 - 1 to 3440 characters or bytes
  - Model 5 - 1 to 3564 characters or bytes

  Note: Additional display screen sizes will be needed for future support of asynchronous terminals.

3.6.2 Return Parameters:

The following parameters are returned from the Resident Module after a 3270 HLLAPI call:

- PARAM1: 3270 HLLAPI Function Number - (Note that the function number is always returned);
- PARAM2: Data String - Returns appropriate information for each specific function. See the IBM 3270 HLLAPI Reference Manual for the data string definition of each 3270 HLLAPI function.

  Note: Additional information will be returned in the future for asynchronous sessions. In order to preserve consistency with existing 3270 HLLAPI functions, this information should be concatenated to the existing data string rather than modifying the currently defined string.

- PARAM3: Data Length - A numeric value giving the resultant length of the data string or the presentation space position;
- PARAM4: Return Code - Numeric return code for each individual function. Usually consistent with the return codes listed below. This structural consistency should be maintained so that common error handling routines can be used by 3270 HLLAPI application programs.

  Note: Additional return codes may be required in the future for asynchronous sessions.

  - Return Code 0 - Valid return or no updates since the last 3270 HLLAPI call was made;
  - Return Code 1 - Invalid presentation space or an invalid 3270 Control Software feature selected (Note that this return code may be different in non-PC environments). Presentation space may not have been connected, configured, selected during customization of the 3270 Control Software, or labeled with a valid name;

— Return Code 2 - Invalid function specified or an error was detected in the function parameter list;

— Return Code 3 - A WS Ctrl (Work Station Control) action has occurred (Only the Change Active Window and Query Active Window WS Ctrl functions are specified under this 3270 HLLAPI standard);

— Return Code 4 - Function could not be performed since target presentation space was busy (in XCLOCK or XSYSTEM state);

— Return Code 5 - Function could not be performed for reasons other than those specified in Return Code 4;

— Return Code 6 - A data error occurred due to an invalid parameter;

— Return Code 7 - Invalid presentation space position;

— Return Code 8 - A 3270 HLLAPI functional procedure error was encountered (e.g., executing functions out of order or use of conflicting functions);

— Return Code 9 - A system error was encountered;

— Return Code 10 - 3270 HLLAPI function not available with the 3270 Control Software version being used;

— Return Code 11 - Requested resource was unavailable.

3.7 Options for 3270 HLLAPI Function 9, Set Session Parameters

- OLDRET and NEWRET (default) - 3270 HLLAPI should support both the OLDRET and NEWRET options. 3270 HLLAPI Function 7 (Query Cursor Location), function 10 (Query Sessions), and function 6 (Search Presentation Space) are affected by this option. Specifying the OLDRET option returns the "result code" for these functions in the "return code" parameter for compatibility with IBM 3270 PC HLLAPI V1.0 and V2.0. Specifying the NEWRET option returns the "result code" in the "length" parameter as in IBM 3270 PC HLLAPI V3.0.

  Note: Applications using Version 1.0 or 2.0 of 3270 HLLAPI should be converted to run under 3270 Control Software equivalent in functionality to the IBM 3270 PC Workstation Program and 3270 PC HLLAPI V3.0 or higher. As of this writing, IBM no longer markets the 3270 PC Control Program V1.22, V2.1, and V3.0.

- ATTRB and NOATTRB (default) - Setting the ATTRB option will return all codes that do not have an ASCII equivalent (except EAB - Extended Attribute Byte) as the original value. All attribute bytes must have a value greater than hex 'C0'. NOATTRB converts all unknown codes to blanks.

- EAB and NOEAB (default) - Setting the EAB option will return the presentation space data with EABs and setting NOEAB with return only data with no EAB.

- STRLEN (default) and STREOT - STRLEN causes all strings to be passed with their exact length. STREOT allows strings to be terminated with an EOT (End of Text) indicator and does not use the exact string length.

- EOT=n - Permits specification of an End of Text indicator. Binary zero is the default.

- CONPHYS and CONLOG (default) - In the PC implementation, CONPHYS causes a physical connect and jump to the requested presentation space when calling function 1, Connect Presentation Space. CONLOG causes a logical connect and does not jump to the requested presentation space, in the PC implementation. Under UNIX, a connect physical (CONPHYS) must return control of the keyboard to the user after the host connecton is established. Connect Logical (CONLOG) prohibits any user interaction under UNIX. The DISPLAY attribute, which is set by a function 9 call, is related to CONLOG in that it allows the user to see the results of the executing workflow on the terminal.

- ESC=n - Specify the escape character for keyboard mnemonics. The at-sign "@" is the default character.

- AUTORESET (default) and NORESET - AUTORESET attempts to reset all inhibited conditions in a 3270 HLLAPI application by prefixing a reset to all keys sent with Function 3, Send Key. NORESET inhibits AUTORESET.

- SRCHALL (default) and SRCHFROM - SRCHALL causes the Search function to search the entire presentation space. SRCHFROM causes the search to start from a designated position.
- SRCHFRWD (default) and SRCHBKWD - SRCHFRWD causes the Search function to search the presentation space in an ascending direction. SRCHBKWD causes the presentation space to be searched in a descending direction.
- TRON and TROFF (default) - Turns trace ON and OFF respectively.
- TWAIT (default), LWAIT, and NWAIT - TWAIT causes function 4, Wait, to wait approximately one minute before timing out on a XCLOCK or XSYSTEM. LWAIT causes the Wait function to wait until XCLOCK or XSYSTEM clears. NWAIT causes the Wait function to check the status and return immediately.
- FPAUSE (default) and IPAUSE - FPAUSE causes a "full duration" pause, whereas IPAUSE causes an interruptible pause.
- QUIET and NOQUIET (default) - QUIET stops messages from being sent to the screen. In the PC implementation, QUIET uses DOS INT #21 hex. NOQUIET permits display of messages to the screen.
- TIMEOUT=N and TIMEOUT=0 (default) - TIMEOUT=N allows the user to set the number of 30-second cycles accepted before issuing a CTRL + BREAK. TIMEOUT=0 causes timeout messages to be displayed every 30 seconds until the user presses CTRL + BREAK.

3.8 Definition of Operator Information Area (OIA) for Function 13

In the PC implementation, 3270 HLLAPI Function 13 returns a data string containing the current OIA information for the presentation space of the connected session. The format and contents of this data string should be consistent in all implementations of 3270 HLLAPI to allow application programs using this function to return and interpret valid data. The format of this data string as defined in the 3270 HLLAPI Programming Guide (see external references below) is as follows:

- Position 1 - OIA Format Byte for the 3270 PC. (Note that this byte is set to 1 for the 3270 PC).
- Positions 2-81 - OIA Image Group
- Positions 82-103 - OIA Group Indicator Meanings

4. REFERENCES

4.1 External References

For details regarding I/O parameters, error codes, etc. refer to the High Level Language Application Program Interface Programming Guide for the IBM 3270 Personal Computer (document 59X9474), or an equivalent document from other manufacturers of 3270 connected workstations or processors.

4.2 Glossary

APPC - Advanced Program to Program Communications
BCC - Bellcore Client Company
CDS - Centrally Developed Systems
DFT - Distributed Function Terminal
ECF - Enhanced Connectivity Function
HLLAPI - High Level Language Application Program Interface
OIA - Operator Information Area
OS/2 - Operating System/2
SRPI - Server Requestor Program Interface

ATTACHMENT 1

ADDITIONAL BELLCORE GENERIC REQUIREMENTS FOR 3270 HIGH LEVEL LANGUAGE API (HLLAPI)

This list represents Bellcore's current anticipated generic requirements for 3270 HLLAPI in CDS development. Updates to this list may be made periodically as either development requirements change or the existing 3270 HLLAPI support becomes insufficient in certain areas.

Asynchronous Communications:

Support for Asynchronous ASCII communications - Each host session should be able to connect to either an Asynchronous or 3270 Synchronous host. During customization of the 3270 Control Software, the user should be able to configure a host session as either an Asynchronous or 3270 Synchronous terminal.

Asynchronous support under 3270 HLLAPI should be made as transparent as possible to the programmer or user. Where possible asynchronous support should be built into each of the HLLAPI presentation space functions. The format of the presentation space (host and alternate), data description fields (data modify tags), host field identifiers, and host notification fields should correspond to the terminal type (Asynchronous or Synchronous) configured in the 3270 Control Software. Additional information that is sent or returned to the HLLAPI application program for asynchronous support through data strings or error return fields, should be appended to the current value or parameter lists for that field. Initial support for asynchronous terminal formats should include the following:

- VT100 type terminal
- VT52 type terminal
- Teletype 33

Support should be provided for dialing remote asynchronous host computers using a modem connected to a serial RS-232 port, at transmission speeds of 300, 1200, and 2400 baud.

Since the operator cannot stop the scrolling of data to the screen from an asynchronous host, HLLAPI would only be able to process the information remaining on the screen after the data had finished scrolling. HLLAPI should include a buffer to store the entire List that scrolled off the screen. HLLAPI could then process the data in this buffer.

Support for control keys used on asynchronous terminals should be provided by the HLLAPI and the Workstation Program.

Note: At this time Bellcore does not have generic requirements for an asynchronous HLLAPI interface.

Additional 3270 HLLAPI Functions and Modifications to Existing Functions:

The following 3270 HLLAPI functions should be added to V3.0's functionality:

- *Notify Completion of Host Screen Update - The function should signal the 3270 HLLAPI application program when the host has successfully updated the screen and the 3270 Control Software has finished processing the message. Currently, the 3270 Control Software notifies the 3270 HLLAPI application when the screen update begins to be processed, rather than when the screen update is complete. Therefore, as it stands today, if a 3270 HLLAPI application begins to access a presentation space when notification of screen update is received, the program may read either the old, non-updated screen or a partially updated screen.*

  Because of this problem, Bellcore applications must keep copies of the screen in a buffer and compare the *contents of the buffer with the screen to determine if host update occurred within a specific period of time. This not only takes up CPU resources, but is unreliable in cases where host response time is abnormally slow.*

- *Function 24, Query Host Notification, should be modified to include an option to ignore the attribute bytes. Currently, when an attribute byte is sent with no visible screen change, a signal is sent to the 3270 HLLAPI application indicating a host update. This is confusing to the 3270 HLLAPI application with unpredictable results.*

Requirement for FAR Pointers in 3270 HLLAPI (PC implementation only):

The 3270 HLLAPI LIMs and function call parameters should support both "far" and "near" pointers. This will allow the 3270 HLLAPI application program and the data to be in separate 64KB memory segments as required in a multitasking environment.

```
hllapi (function, data, length, retcode);

int *function, *length, *retcode
char *data
```

Continued Support for the 3270 HLLAPI Interface:

3270 HLLAPI should be maintained as a supported interface in future communications and workstation products. In addition, the base set of functions defined in the generic requirements in this Technical Advisory should be maintained in subsequent releases of 3270 HLLAPI for the next 5 years.

Reserved Function Numbers for User Enhancements:

The 3270 HLLAPI function numbers 200 to 300 should be reserved for Bellcore CDS or end-user-written functions in all future versions of 3270 HLLAPI.

Note: It is not recommended that Bellcore CDS developers create their own 3270 HLLAPI functions since non-IBM Control Software may not be low level compatible with IBM's Workstation Program, thus causing compatibility problems with Bellcore written functions. In addition, the low level API support in the 3270 Control Software may change in future versions, forcing Bellcore to rewrite the added 3270 HLLAPI functions.

3270 HLLAPI Support for Alternate Input Devices:

Support for optional input devices including the light pen and mouse should be considered in future versions of 3270 HLLAPI.

ATTACHMENT 2

LIST OF 3270 HLLAPI FUNCTIONS BY FUNCTION NUMBER

For details regarding the input and output parameters of each of the below functions, refer to the High Level Language Application Program Interface Programming Guide for the IBM 3270 Personal Computer (document 59X9474) or equivalent for other manufacturers of 3270 connected workstations or processing.

Function Number 1 - Connect Presentation Space

Establishes the connection between the 3270 HLLAPI application program and the designated presentation space.

Function Number 2 - Disconnect Presentation Space

Disconnects the connection between the 3270 HLLAPI application program and the designated presentation space.

Function Number 3 - Send Key

Sends a string of keystroke(s) (maximum 255) to the connected presentation space. Since the length of this string is required, an EOT indicator may be used instead of the length variable. The Send Key function can be used to simulate an operator entering transactions to the connected host session.

Function Number 4 - Wait

Checks the status of the current connected presentation space. For example, Wait checks that the 3270 HLLAPI application is connected to a valid session, tests whether the keyboard is locked, or tests whether a system error has occurred. If the 3270 host session is waiting on a host response, Wait will cause the 3270 HLLAPI application to wait one minute to see if the host response was made. The Wait function waits on either the XCLOCK or XSYSTEM status to clear.

Function Number 5 - Copy Presentation Space

Copies the contents of the connected presentation space into a data area defined by the 3270 HLLAPI application program. Function 5 uses the EAB and NOEAB options set in Function 9.

Function Number 6 - Search Presentation Space

Searches the connected presentation space for a specified string. Function 6 searches the screen for keyword messages or status words (e.g., VM READ, RUNNING, MORE...), or for prompting messages (e.g., ENTER PASSWORD). Keywords expected to be on the screen can be found and used for validation.

Function Number 7 - Query Cursor Location

Returns the position of the cursor in the connected presentation space. Compatibility of Function 7 with 3270 HLLAPI V1.0 and V2.0 is affected by the OLDRET and NEWRET options set in Function 9.

Function Number 8 - Copy Presentation Space to String

Copies all or part of the current presentation space to a string defined in the 3270 HLLAPI application program. The options ATTRB/NOATTRB and EAB/NOEAB set by Function 9 affect this function.

Function Number 9 - Set Session Parameters

Allows default 3270 HLLAPI session options to be changed. For example, both the EOT (End of Transmission) character and the compatibility options, OLDRET and NEWRET, can be set. In general, function 9 allows the 3270 HLLAPI application program to set the protocol used to interact with the presentation spaces.

Function Number 10 - Query Sessions

Returns the number of valid connected sessions configured in the 3270 Control Software. This function returns the short session name, long session name, session type, and the presentation space size in a non-formatted string. A maximum of seven presentation spaces in the PC implementation and sixteen presentation spaces under UNIX is specified by this standard. Function 10 is affected by the OLDRET and NEWRET options set in Function 9.

Function Number 11 - Reserve

Locks out other sources of keystroke input (e.g., the operator) from the connected presentation space. The presentation space remains locked until Function 12, Release, is called.

Function Number 12 - Release

Unlocks the connected presentation space locked by Function 11, Reserve.

Function Number 13 - Copy OIA

Returns the OIA (Operator Information Area) information for the connected presentation space.

Function Number 14 - Query Field Attribute

Returns the attribute byte of the field currently pointed to in the connected presentation space. Extended attribute bytes (EABs) are ignored.

Function Number 15 - Copy String to Presentation Space

Copies an ASCII data string directly into the connected presentation space at the location currently specified by the "PS Position" parameter.

Function Number 16 - WS Ctrl (Workstation Control)

Allows a 3270 HLLAPI application to use WS Ctrl features of the 3270 Control Software without actually entering keystrokes. Because of the lack of standardization in 3270 workstation hardware, it is not recommended to build dependencies on specific workstation control features that may change with new hardware. Therefore only the CA, Change Active Window, and QA, Query Active Window, commands of Function 16 are included in this standard.

The CA option changes the active screen profile and presentation space that the 3270 HLLAPI application considers to be active.

Function Number 17 - Storage Manger

Allows the definition of storage blocks for use by a 3270 HLLAPI application without allocating large blocks of storage for individual 3270 HLLAPI functions. The following four calls can be made from the storage manager:

- Get Storage (subfunction 1) - Allocates a portion of the storage block (defined by HLLBASM.EXE in the PC implementation) for a 3270 HLLAPI application. A maximum of 12 blocks can be allocated.
- Free Storage (subfunction 2) - Frees the block of storage allocated by Get Storage.
- Query Free Storage (subfunction 3) - Returns the size of the largest block of available free storage.
- Free All Storage (subfunction 4) - Frees all allocated storage blocks defined by Get Storage. These freed blocks of storage are reallocated into a block memory that is the same size as the original block of storage (requested when HLLBASM.EXE was loaded in the PC implementation). To remove HLLBASM.EXE from memory (in the PC implementation) requires a re-boot.

Function Number 18 - Pause

Allows a 3270 HLLAPI application to wait a specified length of time for a host update event to occur. Pause should be used in place of hard coded "timing loops" that wait a specified period of time. In order for a host event to end a Pause, function 23, Start Host Notification, must have been called. Pause should not be used for long time durations (e.g., several hours), to wait for times of the day when system performance is better, or for 3270 HLLAPI applications requiring a high-resolution timer.

Function Number 20 - Query System

Can be used to determine system information, such as the level and version number of 3270 HLLAPI, LIM, and 3270 Control Software. This function is very useful to the service coordinator when diagnosing problems flagged by system errors (return code 9).

Function Number 21 - Reset System

Reinitializes the resident interface module 3270 HLLAPI. Connected presentation spaces are disconnected, reserved sessions are released, alternate presentation spaces are deleted, and session parameter options are reset to their default values. This function is used during system initialization or at program termination.

Function Number 22 - Query Session Status

Helps to prevent system fatal function calls by providing session specific information, determining whether a 3270 HLLAPI application is well-behaved, and determining if a presentation space is suspendable.

Function Number 23 - Start Host Notification

Initiates the monitoring of host session events, that allow the 3270 HLLAPI application to determine if a host presentation space or the host OIA has been updated. After calling this function, function 24, Query Host Update, can be used to determine what specific host event occurred.

Function Number 24 - Query Host Update

Checks to see if the host has updated a presentation space or OIA since the last time this function was called. Function 23, Start Host Notification, must be called before a 3270 HLLAPI application can use this function.

Function Number 25 - Stop Host Notification

Ends a 3270 HLLAPI application's ability to monitor host events using function 24, Query Host Update.

Function Number 30 - Search Field

Looks for a string in a specified field within the active presentation space. If the string is found, this function returns the position of the string relative to the beginning of the presentation space (e.g., row 1, column 1 position is expressed as 1) in decimal format. Both protected and unprotected fields in a field-formatted presentation space can be searched.

Function Number 31 - Find Field Position

Returns the starting position of a specified field in the current connected presentation space. The specified field can be either protected or unprotected in a field-formatted presentation space.

Function Number 32 - Find Field Length

Returns the length of a specified field in the current connected presentation space. The length is the number of characters found from the beginning of the specified field up to either the character before the next attribute byte or the end of the presentation space. The field inspected can be either protected or unprotected in a field-formatted presentation space (host or alternate presentation space with 3270 keystroke emulation). This function can be used in the validation of data fields.

Function Number 33 - Copy String to Field

Copies a character string into a designated field located in the current connected presentation space. Both protected and unprotected fields in a field-formatted presentation space (host and alternate presentation space with 3270 keystroke emulation) can be specified.

Function Number 34 - Copy Field to String

This function copies a string of characters from the beginning of a specified field in the current connected presentation space into a string defined by the 3270 HLLAPI application. The length and starting position of the field to be copied can be found by calling functions 32, Find Field Length, and function 31, Find Field Position. This function can be used with both protected and unprotected fields in a field-formatted presentation space (host and alternate presentation space with 3270 keystroke emulation).

Function Number 35 - Define Presentation Space

Creates an alternate presentation space for the base PC presentation space, leaving the contents of the base presentation space intact. Alternate presentation spaces can have the format of a host or PC session. One example of using alternate presentation spaces is to create windows on the screen for displaying help or tutorial messages.

Function Number 36 - Switch Presentation Space

Redirects output from the base presentation space to an alternate presentation space created using function 35. The base presentation space contains the locally processed 3270 HLLAPI application. This function should not be used with sessions configured for 3270 keystroke emulation.

Function Number 37 - Display Cursor

Allows the 3270 HLLAPI application to move the display cursor to a specified position in the base local processing session or alternate presentation space. For example, this function could be used to position the cursor at the next input position on the screen.

Function Number 38 - Display Presentation Space

Used to redraw alternate or base local processing presentation spaces, after each data update.

Function Number 39 - Delete Presentation Space

Deletes an alternate presentation space created using Function 35, Define Presentation Space.

Function Number 50 - Start Keystroke Intercept

Allows a locally processed application referred to as a filtering application, to intercept all keystrokes entered into a presentation space by the operator. This function cannot be used with the WS CTRL session.

Function Number 51 - Get Key

Used to receive keystrokes from the sessions specified by Function 50, Start Keystroke Intercept. Once the 3270 HLLAPI application receives the keystrokes they may be processed, accepted, or rejected. This function is frequently used for validating keystrokes as they are entered.

Function Number 52 - Post Intercept Status

Informs the 3270 Control Software whether a keystroke obtained using Function 51, Get Key, was accepted or rejected by the 3270 HLLAPI application. If the keystroke was rejected, the operator is signaled by an audible "beep". This function cannot be used with a WS CTRL session.

Function Number 53 - Stop Keystroke Intercept

Cancels the 3270 HLLAPI application's ability to intercept keystrokes initiated by Function 50, Start Keystroke Intercept, thereby ending the keystroke filtering process.

Function Number 54 - Get 3270 AID Key

Allows the use of 3270 AID (Attention Identifiers) keys in an alternate presentation space configured for 3270 keystroke emulation. The AID keys are non-ASCII control keys (e.g., Enter, Clear, PF keys, and PA keys). The AID key mnemonics returned by the Get 3270 AID Key function are as follows:

1 - PF1
2 - PF2
3 - PF3
4 - PF4
5 - PF5
6 - PF6
7 - PF7
8 - PF8
9 - PF9
a - PF10
b - PF11
c - PF12
d - PF13
e - PF14
f - PF15
g - PF16
h - PF17
i - PF18
j - PF19
k - PF20
l - PF21
m - PF22
n - PF23
o - PF24
x - PA1
y - PA2
z - PA3

C - Clear
E - Enter
J - Alt + CrSel
Q - Alt + Attn

Function Number 90 - Send File

Called by a 3270 HLLAPI application to send a file from the local processing session to a host session. This file transfer capability can be used to transfer data or functions (i.e., load modules) to be processed locally in the workstation or at the remote host. Send File can also be used to send multiple screens of data too large to efficiently copy using one of the 3270 HLLAPI copy functions. (In the PC implementation, the SEND executable file must be located in the same subdirectory as the 3270 HLLAPI program when using a fixed disk.)

In order to use this function the programmer must disconnect the host session, release the alternate presentation spaces, or end keystroke intercept.

Function Number 91 - Receive File

Receives a file into the local processing session from a host session. The Receive File function can be called from a 3270 HLLAPI application. (In the PC implementation, when using a fixed disk, the RECEIVE executable file must be located in the same subdirectory as the 3270 HLLAPI program.) As with the Send File function, the programmer must not have issued any function calls (connect, reserve, start keystroke intercept) prior to using this function.

Function Number 92 - Invoke DOS Program

Allows a 3270 HLLAPI application to invoke another application. The application must be executable (and have a .COM or .EXE extension in the PC implementation). The 3270 HLLAPI program cannot call another program that uses 3270 HLLAPI, since HLLAPI does not support recursion. Function 93, DOS Redirect, must be used if DOS resident functions (e.g., copy) are called.

Function Number 93 - DOS Redirect (PC implementation only)

Allows the 3270 HLLAPI application to call any other application with a ".COM" or ".EXE" extension, to call any DOS function, or to call DOS batch files. This provides essentially the same capabilities available to the user at the DOS prompt. This function cannot be used to invoke another application using 3270 HLLAPI. It is not recommended to call either Send File or Receive File using this function.

Function Number 99 - Convert Position or RowCol

Converts a screen display row/column coordinate into a presentation space positional value. Likewise, a presentation space positional value can be converted into a display row/column coordinate.

This is the only function that returns a status code in the "return code" field. Common error handling routines need to detect this function and properly interpret the status code containing the presentation space position or column.