

United States
Environmental Protection
Agency

Office of Air Quality
Planning and Standards
Research Triangle Park, NC 27711

EPA-454/R-93-049
December 1993

AIR



STANDARDS

FOR

DEVELOPMENT/ENHANCEMENT

OF

OAQPS TTN

BULLETIN BOARDS





EPA-454/R-93-049

STANDARDS
FOR
DEVELOPMENT/ENHANCEMENT
OF
OAQPS TTN
BULLETIN BOARDS

U.S. Environmental Protection Agency
Region 5, Library (PL-12J)
77 West Jackson Boulevard, 12th Floor
Chicago, IL 60604-3590

December 1993

U. S. ENVIRONMENTAL PROTECTION AGENCY
Office of Air Quality Planning and Standards
Technical Support Division
Research Triangle Park, NC 27711

Region 5
77 West Jackson Boulevard, 12th Floor
Chicago, IL 60604-3590

DISCLAIMER

This report has been reviewed by the Office of Air Quality Planning and Standards, EPA, and approved for publication. Mention of trade names or commercial products is not intended to constitute endorsement or recommendation for use.

TABLE OF CONTENTS

1	Introduction	1-1
1.1	What is the Office of Air Quality Planning and Standards Technology Transfer Network (OAQPS TTN)	1-1
1.2	What is The Bread Board System (TBBS)	1-1
1.3	What is The Data Base System (TDBS)	1-1
2	Bulletin Board System Development	2-1
2.1	TBBS Development	2-1
2.1.1	TBBS SDL Source Format	2-2
2.2	TDBS Development	2-5
2.2.1	TDBS Specific Commands	2-8
2.2.2	TDBS Do's and Don'ts	2-13
2.2.3	Source Code Format and Documentation	2-14
2.2.4	Compiler Options	2-16
2.2.5	Performance Measurements	2-17
2.3	Contractors' Role	2-18
2.3.1	Policies Concerning Contractors	2-18
2.4	Standards and Conventions	2-19
3	Bulletin Board System Maintenance	3-1
3.1	TBBS Maintenance	3-1
3.1.1	Menu naming conventions	3-1
3.1.2	SDL menu maintenance	3-1
3.1.3	Menu structure	3-2
3.1.4	Directories	3-3
3.1.5	Archived Files	3-4
3.1.6	TTN file structure and file management ...	3-4
3.2	TDBS Maintenance	3-5
3.2.1	Indexed Data Bases	3-5

4	Bulletin Board System Operators' Responsibilities and Duties	4-1
4.1	SYSOP Responsibilities	4-1
4.2	SYSOP Duties	4-1
4.2.1	Directory and File Maintenance	4-1
4.2.2	Archiving Files	4-3
4.2.3	Preparing ASCII Text Files	4-4
4.2.4	Aquiring File Download Reports	4-5
4.2.5	Processing Mailing Lists	4-5
4.2.6	Searching User Identification	4-5
4.2.7	Remote SYSOP File Transfers	4-5
4.2.8	Message Board Frequency Checks	4-6

SECTION 1

INTRODUCTION

The Office of Air Quality Planning and Standards (OAQPS) Technology Transfer Network (TTN) is an electronic bulletin board system that is used to facilitate communications and disseminate technical information among EPA staff, EPA Regional Offices, and State and local agencies. The TTN also provides communication and information services to private industry, environmental consultants, educational institutions and individuals engaged in environmental issues and projects. The TTN is comprised of over 15 sub bulletin board systems that cover a wide variety of air pollution topics. The purpose of this document is to set standards for the development, enhancement and maintenance of bulletin boards within the TTN.

1.1. What is TBBS

The Bread Board System (TBBS) is electronic bulletin board software that is used to develop and customize telecommunications systems. It will function with up to 64 lines at speeds up to 14,400 bps. Lines can be connected either through serial ports to modems, data adapters to various networks or hard wired locally. The software offers over 40 separate menu commands, supports public and private message boards and a system of privilege levels and user authorization flags. TBBS supports a wide range of file transfer protocols such as XMODEM, YMODEM, SEALink, KERMIT and ZMODEM. TBBS operates on DOS computers through the fastest 80486 machines on the market. TBBS is written and supported by eSoft, Inc., 15200 East Girard Avenue, Aurora, Colorado.

1.2. What is TDBS

The Data Base System (TDBS) is a special option module that provides the capability to write dBase language programs that can be run under a TBBS electronic bulletin board system. It fully supports the dBase III+ standard for explicit file and record locking capabilities for explicit program control. TDBS is compatible with the dBase III+ language and provides many TDBS extensions to the language. The full screen interactive commands (such as EDIT, BROWSE etc.) are not implemented due to the restriction of operation through a serial port.

SECTION 2

Bulletin Board System Development

Bulletin boards for the OAQPS TTN are developed in-house by MSS staff members or on contract to private companies with a knowledge of TBBS, TDBS and dBase/Clipper. For contract development a formal proposal and plan must be submitted for approval by TTN staff. An organizational structure and index of the BBS being developed must be presented to the TTN staff. Whether boards are developed in-house or on contract, there are certain standards that must be followed in order to provide a network that is similar in design, is efficient and has the same look and feel to the user community. These sections address guidelines for TBBS and TDBS development.

2.1. TBBS Development

Bulletin boards that use only those functions provided by TBBS are the most efficient and responsive bulletin boards that can be developed for the TTN. As a result, TTN developers should make every effort to use pure TBBS whenever possible. Each bulletin board developed for the TTN must follow, as an absolute minimum, the TBBS standards outlined below:

- 1) All menu compilation will be conducted in-house by TTN staff. Contractors must submit ASCII source files, in the specified format, that represent the menus required for the particular BBS application.

- 2) The main menu of each BBS must be a TBBS menu, not a TDBS program.

- 3) Make use of TBBS menus and functions as much as possible rather than TDBS programs.

- 4) TTN main menus will retain roughly the same box structure common to all TTN BBS main menus with the sysop name and phone number indicated at the top.

- 5) All TTN bulletin board systems must have alerts to keep users informed and up-to-date regarding changes and additions to the BBS. Alerts files are to be incorporated as text (ASCII) files.

6) The return to Top menu from the main menu of each BBS must use the "<-> Return to TOP Menu" convention and a type 12 menu entry with Opt Data = 0.

7) The "<=> Return to Previous Menu" must use a type 12 menu entry with Opt Data = 1.

8) Return to menus more than 1 level deep must use a type 12 menu entry with Opt Data = N, where N is the number of levels to return.

9) Each TBBS menu must be located in the appropriate and approved DOS directory as assigned by the system manager.

10) All menu names consist of MENUxxyy.CTL where the "xxyy" must be assigned by the system manager. The "xx" portion is used to designate the BBS to which the menu belongs.

2.1.1 TBBS SDL Source Format

TBBS menus are written in SDL (System Development Language), compiled and tested prior to being placed on the TTN. The source files for the menus must adhere to the following requirements:

1) The first lines of the .SDL menu source file will identify the associated bbs, the name of the person who wrote the code (and their company if a contractor), and a list of all menus the file creates. The listing of menus to be indented to show the relationship of which menu calls which menu. For example:

```
*****
;*
;*          EMTIC BBS
;*
;* Consists of:
;*   MenuEM01.ct1   The alerts menu for the EMTIC BBS
;*   MenuEM02.ct1   The main menu for the EMTIC BBS
;*   MenuEM03.ct1   The upload area menu
;*
;* Written by:
;*   Tom Faulkner
;*   BBS Consultants, Inc.
;*   919-683-2078
;*
;* Date written:
;*   August 7, 1993
;*
;* Modification History:
;*   08/07/93   Initial Release
;*
*****
```


2) All equates and macros must be together as the first code items after the identification block. Each equate must be on a separate line and come before macros. Each Macro: keyword must have an explicit EndMacro: keyword.

3) All menus must be commented with a header block that identifies the name of the menu file, and menus that call it, and any menus it calls. The header block will be set off from the rest of the code with a row of "*"s above and below the header as shown here

```

;*****
;*
;*                               MenuEM02.ct1
;*
;*   Called by:
;*       MenuEM01.ct1
;*
;*   Calls:
;*       MenuEM03.ct1
;*       MenuEM04.ct1
;*       MenuEM05.ct1
;*
;*****

```

Macros should be identified in a similar manner and identify any menus that use them. For example:

```

;*****
;*
;*                               Macro: TopMenu
;*
;*   Action: Returns to top level menu
;*
;*   Used in:
;*       MenuEM01.ct1
;*       MenuEM02.ct1
;*       MenuEM03.ct1
;*
;*****
;
Macro: TopMenu
    Entry:
        <-> Return to TTN BBS
        @AnyUser
        Key=- Type = 12 Opt Data=0
EndMacro:

```

NOTE: Macros must be defined before they are called. In this case, the macro "AnyUser" must have been defined prior to the TopMenu macro.

4) Each Menu: keyword must have a matching explicit EndMenu: keyword.

5) Each Macro: keyword must have a matching explicit EndMacro: keyword.

6) Each Title: keyword must have a matching explicit EndTitle: keyword.

7) Menus for each bbs are contained in one file determined by the System Manager. MEDIT Compatible: OFF must be placed before any menu definitions.

8) The Billing: directive must not be used.

9) The Remark:/EndRemark: directives must not be used.

10) Each menu entry must contain all keyword modifiers in the following format:

Entry:

[text, if any]

Priv=<pv> A1=<f1> A2=<f2> A3=<f3> A4=<f4> IBM=<flg> ANS=<flg>

Key=<character> Type=<type> Opt Data=[optional data, if any]

pv = minimum privilege level for use of the entry.

f1 through f4 = 8 character flags

Macros may be defined at the beginning of the source file to avoid the tedious retyping of the same information.

11) The List:/NoList: keywords may only be used for repetitive code that is included in a macro

12) Any code between an If:/IfNot: and its associated EndIf: directives must be indented 3 spaces (SDL ignores leading spaces on compilation and this will not affect the resulting menus)

13) Source code for menus is to be provided to the System Manager on disk as well as printed. Indentation and alignment is to be done with spaces only, tabs are not permitted. In printed form, it must use a 12 point, non-proportional spaced font.

14) There must be blank line with only a ";" in column one to separate lines containing the keywords Menu:, Title:, Entry:, EndMenu:, and Macro: from the preceding line.

15) Keywords must be spelled out completely and must be in mixed upper and lower case as shown here - Macro:, EndMacro, Equate:, Title:, EndTitle:, Menu:, EndMenu:, Entry:, Key=, Type=, Priv=, A1=, A2=, A3=, A4=, ANS=, IBM=, Opt Data=, List:, NoList:, Page:, If:, IfNot:, and EndIf:.

16) Control characters, if any, needed in a menu must be included using the ^ convention - i.e. a Ctrl-A would be included as ^A in the source code. The SDL compiler will make the substitution. There will be no literal control characters in the source file.

17) Escape characters, if needed, will be included in the source file as ^[.

18) All menus must be readable and operational for callers with or without ANSI capability as well as those with and without the ability to display IBM graphics.

2.2. TDBS Development

Bulletin boards that have special needs and requirements not provided by TBBS functions can use TDBS programs. However, TDBS programs should be used sparingly and only if there is absolutely no TBBS function that will do the job. TDBS programs must be coded as efficiently as possible, whether done in-house or on contract. dBase programs for TDBS, if not coded properly, can be extremely slow and cause TTN system degradation for all users.

A TDBS program begins with one or more ASCII source files containing dBase language instructions. These source files are compiled by the TDBS compiler and become a single "token program" (.TPG) file. The TDBS option module then executes these token (.TPG) files as specified by TBBS menu entries.

The target level of compatibility for TDBS is the dBase III+ program language. TDBS is not 100% compatible with dBase III+ because of the on-line environment in which TDBS programs are executed. TDBS is a compiler while dBase III+ is an interpreter. Thus some dBase commands are not present in TDBS. TDBS has some memory limit constraints which dBase III+ does not have.

Because TDBS runs under TBBS, extensions have been added to the language to take advantage of that environment. These enhancements are not backward compatible with other dBase language dialects. TDBS implements transparent locking and automatic locking in addition to the dBase standard explicit file and record locking for shared file access. This allows TDBS to handle multiuser file accesses smoothly in ways which are impossible in most dBase language dialects.

In order to develop TDBS programs for TBBS, familiarity with the dBase III+ language is required as is a knowledge concerning the TDBS dialect, as well as an understanding of the TBBS multi-user environment.

TDBS is frequently compared to the CA-Clipper dBase language compiler. While TDBS does not have the richness of commands and functions of CA-Clipper, it does have certain flat file handling functions not present in CA-Clipper as well as functions to read and write to the userlog which are totally unique to TDBS. In program design and operation there are several significant differences of which a programmer familiar with CA-Clipper must be aware.

First, in CA-Clipper, the programmer writing a multi-user application must be keenly aware of file and record locking whereas TDBS handles these functions transparently when working with database files. Placing un-needed file and record locks will slow a TDBS program down as well as being more difficult to maintain.

Second, multi-user applications in CA-Clipper operate on a network where each user has the power of a complete CPU for their exclusive use. A common mistake made by CA-Clipper programmers when writing TDBS code is using a tight do-nothing loop with the Inkey() function while waiting for a key press. For example:

```
DO WHILE Inkey() = 0
ENDDO
```

Such code utilizes 100% of the CPU resulting in very lengthy pauses for users on other lines. It is totally unacceptable in a multi-tasking environment like TDBS to have code that results in a 100% duty cycle. Use Inkey(240), instead, since it produces virtually no system impact for 4 minutes (240 seconds) then trap the case where no key was pressed and gracefully exit the program.

The multi-tasker in TDBS can only do its job between program instructions. Instructions like INDEX ON or COPY TO provide no opportunity for the multi-tasker to function since they are single instructions. For instance, a benchmark test to create an index on a 45 character field of an 11,483 item database using a 486/33 took only 12 seconds to index in CA-Clipper in one particular program but took 6 MINUTES AND 33 SECONDS in TDBS. During that time, the TDBS system appeared to freeze to callers on other lines.

Each bulletin board developed for the TTN that makes use of TDBS programs must follow the standards listed below:

- 1) If any indexes are used, they must be created and maintained off-line by use of either a standalone dBase III+ program, or a CA-Clipper compiled program. The standalone program must have the capability to reindex and pack the databases. Source code for this program must be provided and be formatted the same as for the on-line portion. All indexes must be dBase III+ compatible .NDX format. Even if a maintenance program uses indexes only as an intermediate step, it must still be dBase III+ compatible, .NDX files and not the CA-Clipper default .NTX files.

2) Unless specified otherwise in the design requirements, it is not necessary to code the program for use by those callers who do not have ANSI capability. If the callers must have ANSI capability, the TDBS program must correctly detect callers who either have ANSI turned off in their userlog setting or who are calling using a terminal program that does not support ANSI. If the userlog entry has ANSI turned off but the results of the TBBS ANSI poll indicate the callers terminal will support ANSI, the caller is to be advised to change their setting. If the results of the ANSI poll indicate the caller's terminal does not support ANSI, the program must provide the user a message explaining that ANSI capability is required for TDBS applications.

3) Complete documentation must be provided for each TDBS program, including commented source code in the format specified in section 2.2.3.

4) Training and documentation must be provided for sysops and special remote operators of TDBS programs.

5) No TDBS program is to have a global glossary size of more than 4300 bytes. If it is not possible to reduce the global glossary size below this figure by making variables private and reusing the names, then the TDBS program must be separated into functional groups and these groups compiled into separate .TPG files. This is more efficient and less taxing on TDBS resources. This can prevent the application from running out of memory and give each functional group more memory to use. See section 2.2.4 on specifying compiler options.

6) It is the developers' responsibility to thoroughly test TDBS systems before forwarding them to the TTN staff. Not only must the program be tested for logical errors and ease of use, as would normally be required of any program development, but they also must be tested for CPU and disk loading in a multi-user environment. See section 2.2.5 on performance measurements.

7) Developer must maintain meticulous version control records of all file changes, additions, or modifications to TDBS source code. The date, explanation, and author of any modifications must be documented in comments at the beginning of the source code to avoid sending wrong versions of source code files that cause obscure errors that are difficult to resolve. The most recent changes are to be listed first and the source code re-documented, as explained in section 2.2.3 on source code format and documentation.

8) Screens must be refreshed efficiently in order to prevent long and tedious pauses while screens are displayed. Refresh segments of the screen or full screen as appropriate, rather than the entire screen each time. Screens will not refresh as quickly, even at 14,400bps, over a modem as they will appear to refresh on the local console.

Drawing a screen in a strict upper left to bottom right sequence may take considerably longer than sending all information in one color, changing to a highlight color, and using cursor position sequences.

9) Programs must be tested with each TTN access method to determine an optimum level of presentation that is acceptable for all methods. TTN access through the LAN via CrossTalk interprets colors differently and refreshes the screen differently than other software packages. Other methods to test include modem, ethernet, and Internet.

10) Programs of a complex nature that are suspected of causing system degradation must be bench marked and tested in a hidden area on the TTN primary system or on the TTN backup system.

2.2.1 TDBS Specific Commands

There are specific TDBS commands that can and cannot be safely used without causing an undue system impact. These are explained below:

?/??	Can always be safely used.
@...CLEAR	The @ <row>, <col> CLEAR works much faster than the @ <row1>, <col1> CLEAR TO <row2>, <col2> form. The first positions the cursor at the specified row and column then sends a clear to end-of-screen command. The latter sends spaces for each position.
@...SAY...GET	No significant system impact.
@ ...TO	No significant system impact.
ACCEPT...TO	No significant system impact.
APPEND BLANK	No significant system impact.
APPEND FROM	This command should not be used with the default scope of ALL except on small files with not more than 200 to 500 records, depending on the number of fields in the database.
AVERAGE	This command may be used on small files with not more than 200 to 500 records depending on the number of fields to be averaged.
CLEAR	No significant system impact.
CLEAR ALL	No significant system impact.

CLEAR GETS	No significant system impact.
CLEAR MEMORY	No significant system impact.
CLEAR TYPEAHEAD	No significant system impact.
CLOSE	No significant system impact.
CONTINUE	No significant system impact.
COPY TO	This command should not be used with the default scope of ALL except on small files with not more than 200 to 500 records depending on the number of fields to be copied.
COPY FILE	Use the command only on files smaller than 250k. On larger files, use the FBREAD/FBWRITE commands inside a loop with programmed pauses to reduce system loading.
COPY STRUCTURE	No significant system impact.
COPY STRUCTURE EXTENDED	No significant system impact.
COUNT TO	This command should not be used with the default scope of ALL except on small files with not more than 200 to 500 records.
CREATE	Should not be routinely used.
CREATE FROM	Should not be routinely used. Create all needed databases off-line and ZAP them instead.
DECLARE	No significant system impact.
DELETE	This command should not be used with the default scope of ALL except on small files with not more than 200 to 500 records.
DIR	No significant system impact.
DO	No significant system impact.
DO CASE	No significant system impact.
DO WHILE	No significant system impact.
DOTBBS	No significant system impact.
EJECT	No significant system impact.
ERASE	No significant system impact.

FBREAD	No significant system impact.
FBWRITE	No significant system impact.
FCLOSE	No significant system impact.
FCREATE	No significant system impact.
FIND	No significant system impact.
FLFIND	Causes significant system impact if more than a few hundred kilobytes are read before a match or End of File is hit.
FLREAD	No significant system impact.
FLWRITE	No significant system impact.
FOPEN	No significant system impact.
FSEEK	No significant system impact.
GO/GOTO	No significant system impact.
HALT	No significant system impact.
IF	No significant system impact.
INDEX ON	This command may only be used on empty databases without specific permission of the TTN System Manager.
INPUT	No significant system impact.
LOCATE	This command should not be used with the default scope of ALL except on small files with not more than 200 to 500 records.
NOTE/*&&	No significant system impact.
ON DISCONNECT	No significant system impact.
ON ERROR	No significant system impact.
ON ESCAPE	No significant system impact.
ON KEY	No significant system impact.
ON NEWMAIL	No significant system impact.
PARAMETERS	No significant system impact.

PRIVATE	No significant system impact.
PROCEDURE	No significant system impact.
PUBLIC	No significant system impact.
QUIT	No significant system impact.
READ	No significant system impact.
RECALL	No significant system impact.
RELEASE	No significant system impact.
RENAME	No significant system impact.
REPLACE	No significant system impact.
RESTORE	No significant system impact.
RETURN	No significant system impact.
RETURN TO MASTER	No significant system impact.
SAVE	No significant system impact.
SEEK	No significant system impact.
SELECT	No significant system impact.
SET ALTERNATE	No significant system impact.
SET BELL	No significant system impact.
SET CENTURY	No significant system impact.
SET COLOR	No significant system impact.
SET CONFIRM	No significant system impact.
SET CONSOLE	No significant system impact.
SET DATE	No significant system impact.
SET DECIMALS	No significant system impact.
SET DELETED	No significant system impact.
SET DELIMITERS	No significant system impact.
SET DEVICE	No significant system impact.

SET DISCONNECT	No significant system impact.
SET DISPLAY RULES	No significant system impact.
SET DIVIDE BY ZERO	No significant system impact.
SET EDITOR	No significant system impact.
SET ESCAPE	No significant system impact.
SET EXACT	No significant system impact.
SET EXCLUSIVE	No significant system impact.
SET FILTER	No significant system impact.
SET FIXED	No significant system impact.
SET FORMAT	No significant system impact.
SET FUNCTION	No significant system impact.
SET INDEX	No significant system impact.
SET INTENSITY	No significant system impact.
SET MARGIN	No significant system impact.
SET MEMOWIDTH	No significant system impact.
SET ORDER	No significant system impact.
SET PRINT	No significant system impact.
SET PRINTER TO	No significant system impact.
SET PROCEDURE	No significant system impact.
SET RELATION	No significant system impact.
SET SOFTSEEK	No significant system impact.
SET TYPEAHEAD	No significant system impact.
SET UNIQUE	No significant system impact.
SET UPDATE BELL	No significant system impact.
SKIP	No significant system impact.
STORE	No significant system impact.

SUM	This command should not be used with the default scope of ALL, except on small files with not more than 200 to 500 records depending on the number of fields to be summed.
TEXT	No significant system impact.
TYPE	No significant system impact.
UNLOCK	No significant system impact.
USE	No significant system impact.
USE MAILBOX	No significant system impact.
WAIT	No significant system impact.
ZAP	No significant system impact.

2.2.2 TDBS Do's and Don'ts

Because TDBS is a multi-user operating system it is important to conserve system resources to prevent one TDBS application from monopolizing the CPU and/or disk. The programmer must always keep in mind that the CPU and disk resources are being shared by up to 65 simultaneous users and that no one program can use all the resources for more than a couple of seconds without a noticeable pause to other users whether they are using the same TDBS program or just downloading a file.

The following is a list of required dos and don'ts for program development:

- 1) Do not use more than 4300 bytes for the global glossary.
- 2) Do use short variable names and reuse variable names. Simple counter variable should never be more than two characters in length.
- 3) Do not open all files at the beginning of each program and then close them at the end.
- 4) Do open files only as needed. No file should be opened or closed more than once in a program.
- 5) Do not assume that the end user will know anything whatsoever about how to use the program.

6) Do give clear, concise instructions and prompts. The F1 key must not be used for any purpose other than to display a help file. Avoid commonly used communication software key sequences such as ESC, HOME, PgUp, PgDn and function keys.

7) Do not specify literal paths in the source code.

8) Do use the TDBS Homopath() function and make all other required directories a sub-directory to the one where the actual .tpg file is located.

9) Do not use "do nothing" loops. While putting the CPU into a 100% duty cycle loop when waiting for the user to press a key is not harmful in a single user program or on a network where each user has a CPU, it is not acceptable in a multi-user environment like TDBS.

10) Do not index non-empty databases on-line.

11) Do not over-modularize the code. Calling a procedure takes a finite amount of time. In other programming languages, the inefficiencies of calling procedures/functions is made up for by having all the executable code in memory at the same time. It's unlikely any non-trivial program written in TDBS will fit completely into memory, so it can be more efficient to simply repeat code blocks. Use procedures for clarity only.

2.2.3 Source Code Format and Documentation

The developer must adhere to the following format and documentation conventions for all programs developed for use with the TTN BBS.

1) All identifiers will start with a lower case letter indicating their type:

- c = character type identifiers,
- n = numeric type identifiers,
- l = logic type identifiers,
- d = date type identifiers,
- f = database field names, and
- p = procedures.

The next character, if not a numeral, is to be capitalized. Subsequent characters should be upper/lower case as appropriate for easy comprehension. Identifiers should be as short as possible and in no case may an identifier be more than 8 characters in length. Examples of acceptable names are nl, nCount, cName, lIsOK, dStart, pGetIn. Identifiers must be declared PRIVATE unless specifically needed to be PUBLIC, and names should be reused in procedures. In

other programming languages, using long, descriptive variable names is often considered good style; however, in TDBS it will make the global glossary larger and the program will run slower. Avoid them.

2) TDBS commands must all be in UPPER CASE.

3) TDBS functions must start with an upper case letter, followed by all lower case letters.

4) Source code must be readable and contain liberal comments.

5) Statements that are subordinate to an IF statement, a DO WHILE, or DO CASE statement must be indented 3 characters. ENDIF, ENDDO, and ENDCASE statements are to be aligned with their matching IF, DO WHILE, or DO CASE commands. ELSE statements are also to be aligned with their matching IF statements. CASE statements are to be aligned with the DO CASE statement.

6) Never use a variable where a numeric constant can be used. Any place a numeric constant is used there must be a comment explaining its significance. For example, instead of defining ESC = 27, and later using:

```
IF Inkey() = ESC,
```

```
    use
```

```
IF Inkey() = 27 && Escape
```

This is exactly opposite of the current fashion and is done to conserve global glossary space.

7) In original source code, lines that are all comments should begin with "****". Revisions to code must be indicated by comments which not only describe the problem fixed, but also the initials of the programmer and the date revised.

8) No line of code or comments can be more than 80 characters in length.

9) All source code for TDBS programs should use SNAP! v5.02, if available, with settings compatible with the requirements above. Both source code and "action" diagrams are to be submitted.

2.2.4 Compiler Options

The compiler command line options that effect performance of TDBS programs are:

```
/GLOSS xxxx,  
/DIBUF,  
/FGLOSS xxx, and  
/GETPOOL xxx.
```

The /GLOSS xxxx option should always be used. The xxxx is the number of additional global glossary bytes reserved for run time use. The number cannot be less than 128 bytes, nor more than 12,000 bytes. Normally, the programmer will specify just enough to allow for any run time macro expansion of identifier names. Under some circumstances, specifying a larger number of bytes along with the /DIBUF option can cause some parts of a program to work faster. If the /GLOSS option is used with more than 128 bytes reserved, the reason must be documented as comments near the beginning of the source code. In no case is the number of global glossary bytes reported by the compiler to exceed 4300 bytes. Less is better.

The /DIBUF option should always be used. It will never cause a decrease in performance; however, the compiler may ignore the directive depending on the other settings. Use of the /DIBUFF option permits dynamic run time allocation of any unused glossary space for use by the program.

The /FGLOSS xxxx is not normally needed, but may be useful in tight memory situations. It is only used if the compiled program gives an "out of memory" error but will work in a test compile without the /DIBUF option. This option specifies the number of glossary bytes to keep free for a single instruction. The default is 256 bytes but it can be made larger or smaller if required. Larger if an "out of memory" error occurs. Smaller if the program can be made to run faster by freeing up unneeded bytes.

The /GETPOOL xxxx option should always be used. The xxx is the number of bytes the compiler reserves for use on any GETs before a READ command. It should be set to the maximum size of any pending GET command in the program.

The /REL11 option should never be used. All other command line arguments are optional at the description of the programmer but must be specified in the documentation. To insure consistency between revisions, the programmer should use, and provide, a batch file with the command line arguments included. The batch file should have the same name as the main program. For example:

```
c:\tbbs\tdbs myprog,myprog /DIBUF /GLOSS 128 /GETPOOL 64
```

would be the contents of MYPROG.BAT to compile the MYPROG.PRG file. If a .TDB multiple file specification file is used, the example would be:

```
c:\tbbs\tbbs @myprog myprog,myprog /DIBUF /GLOSS 128 GETPOOL 64
```

The programmer is responsible for applying all required modifications to their copy of MLTBBS.EXE, TDBS.EXE, and TDBSOM.EXE for all testing and compiling. Optional modifications will be installed, or not installed, as they are on the main TTN BBS. The latest modifications are in the file UPDATE.BIN and are applied using the UPDATE.EXE program. Both are available from the eSoft support board at 303-699-8222 and may be downloaded by the registered owner of TBBS, or a person they designate.

2.2.5 Performance Measurements

It cannot be overemphasized that testing a TDBS program on a development machine with only a single user is not adequate. In actual use the TTN BBS may have as many as 65 simultaneous users (the local console is line 0). If a given instruction takes only 10 seconds to perform and 65 users invoke it, it will cause the system to take 10 minutes, 50 seconds for all users to be serviced. Statistically this will never happen; however, the programmer must not assume that the program they are developing is the only one running. The problem is not just that it might take the user of a specific program a long time, but that poor design can cause other users to see a noticeable delay even if they are not in a TDBS program.

The TBBS scheduler can only operate between instructions; therefore, the restriction is not how long a procedure takes, but rather, how long will any single instruction take. For instance an APPEND FROM statement is only one instruction. If the program appends a very large database, the system could slow down unacceptably. If the same action were done inside a DO WHILE loop with a scope of one record at a time, it would take longer to append the file, but the scheduler would have ample opportunity to service other lines.

An instruction that is used only once a month might be acceptable even if it takes 10 seconds to perform, while an instruction that is used several times in an hour would be considered unacceptable even at 1 second. In development, any single instruction that takes more than 2 seconds should be closely examined for alternatives. Any single instruction that takes more than 5 seconds to perform must be specifically approved by the TTN System Manager.

In addition to single instruction time, the program must not load the TTN BBS to a total system response time of more than 150ms or a system turnaround time of more than 400ms averaged over a 6 second period. This should be measured during the heaviest loading period on the day, unless specifically waived by the TTN System Manager.

2.3 Contractors' Role

Contractor support can be utilized to develop and maintain bulletin board systems for the TTN. However, a contractor cannot assume the same role as an EPA employee with the same authority, responsibility and freedom to maintain a TTN BBS. A work assignment for support of the TTN is to be treated the same as any other work assignment. When files, documents, memos etc. are placed on the TTN for public viewing and downloading by the user, the EPA system operator is the person ultimately responsible for the data being placed on the TTN BBS even though support is received from a contractor.

2.3.1 Policies concerning Contractors

Due to the nature of the information that is offered, decisions about material to be released must remain the responsibility of the Environmental Protection Agency (EPA) staff. If extramural support is used for matters dealing with the TTN, care must be taken to avoid the impression that contractors are speaking for the Agency. In mid-March of 1993, a memo was sent to all Technical Support Division Directors and Branch chiefs who sponsor TTN bulletin boards, outlining certain guidelines concerning contractor support. To this end, the following guidelines should be followed with regard to contractor support:

First, systems operators for individual bulletin boards are to be EPA employees and are to be identified on the main menu of each bulletin board.

Second, systems operators are to be members of and attend monthly meetings of the Systems Operators Management Team (SOMT). However, contractors may accompany a member to the meetings.

Third, contractors shall not make unilateral decisions as to what information is to be updated, deleted or placed on a bulletin board.

Fourth, any activity that would give the impression that contractor personnel are performing decision making activities relative to information on the TTN shall be avoided.

Contractors may develop new applications for the expansion and enhancement of bulletin boards. However, before work of this nature takes place, it shall be coordinated with the Chief, Modeling Support Section and his staff for software and hardware compatibility. The complexity of the TTN system requires that new applications, before design and development, be reviewed for compatibility with data base requirements and system limitations.

2.4 Standards and Conventions

There are certain standards and conventions that must be followed for developing bulletin board systems for the TTN. These are of a general nature that apply whether the BBS application is all TBBS or a combination of TBBS and TDBS. These standards and conventions are presented as follows:

1) Files offered for downloading from the TTN must not be of a size that is impractical to download. Files that range between 700k and 1MB are extremely troublesome and sometimes impossible to download. Files offered for downloading should not exceed 1MB in size.

2) Files offered for downloading must conform to a file naming convention for extension names that will identify to the user the type of file being offered. This naming convention is as follows:

- .ZIP - Files that have been archived using the PKZIP archiver.
ZIP'ed files are dearchived using PKUNZIP dearchiver.
- .TXT - ASCII text files that can be read on the screen or downloaded and processed with a text editor or word processor (unarchived).
- .DOC - ASCII text files of documentation for a program or system (unarchived).
- .WPF - WordPerfect 5.1 files. (unarchived)
- .EXE - A PC executable program that can be executed after downloading by typing the "EXE" name. Files with this extension can also be self-extracting archived files that when executed will dearchive.
- .COM - A PC executable module that can be executed after downloading by typing the COM name.
- .BAS - Basic programs that are run under GW-Basic and BASIC interpreters or compiled BASIC programs.

.WK1 - Lotus 1-2-3 worksheet files.

.TIF - Graphics files for graphics compatible programs such as Lotus, Freelance, WordPerfect, Aldus Pagemaker, etc.

.DBF - Data base files.

.DRW - Freelance draw files.

3) Instructions for displaying text files on the screen from download directories should read:

"To display a TXT file to your screen type <P 1 filename>"

SECTION 3

Bulletin Board System Maintenance

3.1. TBBS Maintenance

Once a bulletin board is in place, the task of maintaining the bulletin board requires a knowledge of TBBS menu conventions and procedures, menu structures, and file management conventions. Menu maintenance is done in-house by TTN staff members only.

3.1.1 Menu naming conventions

MENUnnnn.CTL Where nnnn is any character, numeric or combination

Example: MENUch02.CTL - CHIEF main menu

3.1.2 SDL Menu Maintenance

The first step in creating a new menu is to simply copy an existing menu that closely resembles the one you wish to create. Use the DOS copy command to copy a CTL file as follows:

COPY MENUCH18.CTL MENUCH19.CTL

The name of the new menu is automatically inserted in the appropriate places in the new menu.

a) Decompiling:

SDLDC ch19.sdl/m:ch19 Decompiles menuch19.ctl into a ASCII file "ch02.sdl". If the /m:ch02 is left off, TBBS will decompile every menu in the current directory.

b) Modifications are made with an ASCII, text editor.

c) Graphics characters needed for menus can be obtained from any good graphics program that allows graphic characters to be incorporated into the menus while using your text editor.

d) Compiling:

SDL ch19.sdl Compiles the sdl file back into a CTL menu, menuch19.ctl.

Menus with text strings longer than 255 characters will cause a compiler error. To avoid or correct this error, insert the following compiler directive:

MEDIT COMPATIBLE: OFF

This directive can be placed prior to the MENU: xxxx statement replacing the ";" in column one. This will allow Titles and Entry text strings up to 1,000 characters in length.

3.1.3. Menu Structure

Each menu entry contains 4 basic parts:

- a) ENTRY:
- b) Text where applicable
- c) Privilege and authorization flags
- d) Key, type and opt data

Privilege level range: 0 thru 253 - user defined
 254 - junior sysop
 255 - sysop

Authorization flags:

A(1)..... A(2)..... A(3)..... A(4).....

Only the TTN System Manager can assign privilege levels and authorization flags.

Key: <n> where n is any character desired for a menu key, must be enclosed in < > in the text part of the menu.

Type: Choose a numerical type from the menu section of the TBBS User's Manual.

Opt data: Can be a pathname, 4 character menu, or a pseudo directory name with options.

***** Important Note:

When coding returns from menus accessed with the type 5 menu (goto) always use a type 12 menu with a numeric code in the Opt data field. For example to return one level back from a menu accessed with a type 5 would be:

Key= - Type= 12 Opt Data=1

Do not use type 5 menu types to return to previous menus. This causes menu stack overflow errors. Each menu that is accessed adds to a menu stack. The type 12 return subtracts from the menu stack. A menu stack overflow error will cause an abrupt return to the top menu along with displaying an error message. TBBS provides only 20 menu levels deep before a stack overflow occurs.

The convention for TTN menu returns is as follows:

Key= - Return to a main menu or top menu
Key= = Return to a previous menu

Returns to a main menu should identify the main menu it is returning to in the text of the return option. The return from the main menu of a BBS within TTN to the Top menu in the system must be coded as a type 12 and opt data of 0. This sends control to the Top menu in TTN, clears the menu stack and puts the user in the Top menu of TTN where he has the option to select another BBS.

TTN BBS conventions for privilege levels and authorization flags:

PRIV= 50 and A(1)..... The TTN unregistered user
PRIV=100 The TTN registered user

Example authorization flags:

A(1).X..... The EMTIC sysop
A(1)..X..... The CHIEF sysop

3.1.4. Directories

Directory files, known as pseudo directories, must be developed and maintained as an integral part of a BBS. These directories are used by the system to show the BBS user the list of files that are available for downloading and/or displaying on the screen. The directory files are pointed to by path statements in the OPT DATA part of a menu. Directories are ASCII text files and do not have to be compiled. However, they must have "DIR" as a filename extension.

A "FAR" file is another form of directory file used by the system. It must have "FAR" as a filename extension. "FAR" files are also ASCII text files that can be created and modified using only a text editor. "FAR" files are extremely useful when data is to be presented that represents distinct categories. Examples would be data that can be categorized geographically or collected by some generic type. Each "FAR" file must reference a set of "DIR" files.

The following is an example of a place where "DIR" files are stored on the TTN for the SCRAM BBS:

```
E:\SCRAM\DIRS
```

DOS directories must be set up to hold the files required to support a BBS. DOS directories must be set up with a root directory entry with the same acronym that identifies the BBS. The following is an example of the DOS directories required to support the EMTIC BBS.

```
E:\EMTIC\TXT
      \DIRS
      \UTILS
      \PUBDOM
```

All menu "CTL" files, "DIR", "FAR" and DOS directory files generated to hold archived and/or text files must be described and documented for each particular BBS.

Conventions to use for making directories for new BOARDS:

- o Use the actual DOS file name. Do not use generic names for files. The file name will indicate the type of file by the extension such as TXT, ZIP, ARC, COM, EXE. This eliminates confusion when downloading.

3.1.5 Archived Files

Use only the version of ZIP as directed by the TTN System Manager to archive files. ZIP creates archive files that include, in most cases, multiple files. ZIP files provide an efficient and cost saving means to transfer files. ZIP files take up less space on the TTN disk drives, this is important since files and systems are becoming larger and larger.

3.1.6. TTN File Structure and File Management

The file structure defined for the TTN provides the best possible method of file management for a DOS oriented system with multiple bulletin boards. A root directory is setup for each BBS using the BBS acronym. The following is an example:

```
E:\CHIEF\dirs
  \txt
  \sims
  \utils
  \memos
  \spec
  \efdocs
  \eigdocs
```

This file structure allows the BBS to expand in two directions if necessary to allow for expansion, even if the initial planning for growth could not be anticipated.

Certain file management techniques must be employed to conform to the restrictions imposed by DOS. Sub-directories can be added at any level of the BBS structure. However, the DOS guide and reference indicate that MS-DOS runs slowly if there are more than 150 files and sub-directories in one directory. Technical notes concerning DOS file management indicate that no more than 250 files should be in a directory. This is not a restriction on how many files can be listed in a TBBS pseudo directory, only on how many files can be physically placed in a DOS directory.

TTN staff and contractors should keep a close watch on directories to be sure that this convention is adhered to so the number of files in any one directory does not exceed the suggested limit.

BBSs developed that control files via TDBS programs should employ path techniques up front that will allow for easy modification when new directories are required. The Homepath() function should be used to determine the location of the .TPG file and all referenced directories must be sub-directories to the Homepath() to allow for easy modification of path names when it is necessary to expand a BBS to include additional directories or to move a BBS to another DOS drive.

3.2 TDBS Maintenance

TDBS on-line programs, once installed, require maintenance procedures to ensure that the programs operate efficiently and do not cause TTN system lockups. Maintenance requirements are to be documented by the programmer.

3.2.1 Indexed Data Bases

All indexed data bases associated with the TTN must be reindexed on a scheduled basis in order to ensure that indexes do not become corrupted and as a result cause TTN system lockups. Certain data bases that capture data on a weekly basis must be reset during scheduled maintenance periods.

SECTION 4

Bulletin Board System Operators Responsibilities and Duties

4.1 SYSOP Responsibilities

Each SYSOP is responsible for the integrity of all data offered on the TTN for his/her BBS. The accuracy of data and wording of descriptions and text is very important. Be prepared to answer questions and inquiries about the TTN and questions concerning system access. The timely removal or replacement of outdated alerts and bulletins is important. Be able to answer questions concerning your TTN application and be able to direct inquiries to the appropriate contacts in your Organization. The following are specific responsibilities:

- 1) Directory and file maintenance
- 2) Archiving files
- 3) Preparing ASCII text files
- 4) Acquiring File download reports
- 5) Processing Mailing lists
- 6) Searching User identification
- 7) Remote SYSOP file transfers
- 8) Message board frequency checks

4.2 Sysop Duties

The following instructions and suggestions are presented to assist the sysop with the duties involved in fulfilling sysop responsibilities.

4.2.1 Directory and File Maintenance

Files presented on the TTN for downloading or viewing are actually stored in DOS directories on the TTN computer. The files are made available to TTN users through the use of "pseudo" directories that present the user with a directory of files for either downloading to their computer or viewing on the screen. Development and modification to these files can be made as required

and uploaded or made available to the appropriate TTN SYSOP. These directory files are ASCII text files that can be prepared and modified with a text editor. The directory files are identified by their file name extension. The directory files that you need to maintain are identified by the file name extension "DIR". The following is an example of a pseudo directory file used in the system:

```

!
!
!
!
!
!
!   Files identified as "ZIP" files must be de-archived.
!   Files identified as "TXT" or "DOC" files can be
!   typed to the screen by entering <P 1 filename>.
!
!  NAME           Bytes      Date           Description
!-----
e:\emtic\utils\readtsar.txt READTSAR.TXT "Read First" TSAR desc
e:\emtic\utils\tsar.exe TSAR.EXE TSAR Program "Self Dearchiving"
e:\emtic\utils\schwindt.zip SCHWINDT.ZIP Version 2 Phil Schwindt
!>Sampling Programs, 7 in all.

```

Note that the directory consists of header information and comments indicated with a ! in column one. Each entry below the header and comment information consists of one entry for each file offered for downloading or displaying. Each entry consists of the following 5 items of information:

- | | |
|----------------------|--|
| 1) Drive designation | (DOS Drive designation: x:) |
| 2) Directory | (DOS directory where the file resides: \xxxx\) |
| 3) DOS File Name | (DOS file name: xxxxxxxx.xxx) |
| 4) BBS File Name | (BBS file name: xxxxxxxx.xxx) |
| 5) File Description | (File Description: 45 characters per line) |

o The DOS Drive designation where the file is located is always coded in columns 1 and 2.

o The DOS directory where the file resides is coded immediately following the drive designation, enclosed in backward "\" slashes.

- o The DOS file name is coded immediately following the DOS directory including the full DOS name and file extension. The file name conforms to the DOS file naming convention of up to 12 characters that includes up to 8 characters followed by a "." and up to 3 characters for a file name extension.

- o A blank space separates the DOS file name and the BBS file name.

- o The BBS file name is the file name that is presented to the BBS user to display or download. The BBS file name and DOS file name are equated. The two names are usually the same name. It is preferable to code the BBS name in upper case in order to be more prominent when presented on the BBS.

- o A blank space separates the BBS file name and the file description.

- o The file description can be up to 45 characters in length. If more than one line is needed to describe a file, code a "!" in column 1 followed by a ">" and the text of the continued description. The BBS will align the continued description directly beneath the previous line of description when the directory is processed by the BBS software.

If a file is to be added to the BBS for downloading or viewing, an entry must be made in the appropriate pseudo directory. That pseudo directory along with the file must then be made available to the TTN system.

If a file is to be deleted, the appropriate entry must be deleted from the pseudo directory and that pseudo directory made available to the TTN system.

If a change is required in the description of a file, or a change is required in the text, header or comment part of a pseudo directory, the directory can be edited with your ASCII text editor and the appropriate changes made.

4.2.2 Archiving Files

Archived files are used to save space and time. Archived files reduce the amount of space required to store files on the TTN hard drives. Archived files reduce the amount of time required to transfer files via telephone lines that results in a substantial savings in telephone costs for the TTN user.

All files that require archiving must be archived using the approved and licensed version of PKZIP authorized by the System Manager. That version is currently "2.04G". EPA has a site license to distribute PKUNZIP to TTN users to be used for

dearchiving files downloaded from the TTN. You can get a copy of the appropriate archiver plus documentation from a TTN staff member if you do not already have the necessary programs and documentation.

As a general rule archiving produces between 45% and 70% compression, depending upon the type of file being compressed and the archiver used to do the compression. Archiving should take into consideration the worst case TTN user. That is, the TTN user with the slowest modem. The various TTN user connect speeds include 1200, 2400, 9600 and 14,400. An archived file offered for downloading from the TTN that is over 350K will result in a download time for the 1200 bps user that will take over an hour to transfer. When large applications are required, multiple archives should be used with some logical sequence in order to offer the application or system with choices and without excessive transfer times.

4.2.3 Preparing ASCII Text Files

Files that are to be displayed on the screen by TTN users must be in ASCII format. Word processor files such as WordPerfect cannot be displayed for viewing on the screen while in a BBS session. All ASCII text files that are to be displayed on the screen must conform to the following specifications:

- 1) Leave column 80 blank, do not code anything in column 80.

When text files are produced with word processors and editors, a Carriage Return (CR), Line Feed (LF) is inserted to indicate end of record. When the BBS presents a text file and encounters a Cr/LF in column 80, double spacing occurs and the text is not presented as originally prepared. All carriage returns and line feeds should be stripped out.

- 2) Leave one blank line at the end of the text file.

When the file is displayed on the screen, a blank line at the end of the file will separate the text of the file from the BBS prompt line.

- 3) Use the Textin/Textout function of WordPerfect, if WordPerfect is used to produce text files. Carriage control and line feeds will most likely be in the files produced with this function. They will have to be removed with a text editor.

The best method for producing text files with WordPerfect using the Textin/Out function is to set both left and right margins to .5 and use Courier, 10 pitch. This setup fills out the screen very well without placing any characters in column 80.

Text files that are included in archives need not conform to this convention and text can be coded to include column 80, if so desired.

4.2.4 Aquiring File Download Reports

A download report is forwarded to each SYSOP the week following the regular Monday morning maintenance period from 8:00 am to 12:00 noon. This report can be used to track and report on files that are downloaded from your BBS. The report lists the files downloaded from each BBS on the TTN on a weekly basis by file name, and number of times downloaded. You can write your own programs or use your own editors to summarize or compile weekly user statistics for your BBS as required.

4.2.5 Processing Mailing Lists

For boards that provide registration for mailing lists, it is the sysop's responsibility to develop and/or acquire software necessary to process your mailing list including file maintenance and mailing labels. For some boards these registration files are dBase, for some they are ASCII files.

4.2.6 Searching User Identification

TTN's on-line user registry provides first/last name, city/state, and board of primary interest for each registered user. When a user registers for TTN services, he is asked to identify a primary board. The board he chooses as his board of primary interest is reflected in the on-line user registry. If user identification is desired, it is your responsibility to capture that information from the registry.

The on-line registry is available from the Systems Utilities Menu. The entire registry can be downloaded or searched on-line using a text string as a search key. For example, the search string "CHIEF" will display users who have indicated that their board of primary interest is CHIEF.

4.2.7 Remote SYSOP file transfers

A special hidden menu entry and directory for transferring files between each remote SYSOP and the central SYSOP is provided for some boards. The menu entry is not seen by anyone except the remote SYSOP, those appointed by him and the central SYSOP. The menu entry is provided as a means for remote SYSOPS and the central SYSOP to transfer files back and forth as required, in order to maintain each BBS. Uploads to the central SYSOP can be made via

the normal upload menu. Downloads from the central SYSOP can be retrieved via the special menu entry seen only by the remote SYSOP and central SYSOP. Files can be deleted by the person who uploaded the files, usually the remote SYSOP. The name of this special menu varies from BBS to BBS, depending upon the name agreed upon when the BBS was designed. This menu entry is not seen or accessible by anyone except the remote SYSOP.

4.2.8 Message board frequency checks

If a BBS has a public message board for communication between users on subjects and topics pertaining to your particular board, then frequent access to that board is required. You may be able to help solve problems or respond to questions proposed in your public board. Your public board is an open forum for problems, questions, comments and editorials concerning your BBS. You can enter your public message board, request reverse order and read the latest messages that have been written for all to see and respond to, if appropriate. You can also delete messages from your public board that you feel are inappropriate or outdated. The option to delete public messages is seen only by you and those appointed by you to have the same privileges.

TECHNICAL REPORT DATA
(Please read Instructions on the reverse before completing)

1. REPORT NO. EPA-454/R-93-049		2.		3. RECIPIENT'S ACCESSION NO.	
4. TITLE AND SUBTITLE Standards for Development and Enhancement of OAQPS TTN Bulletin Boards				5. REPORT DATE December 1993	
				6. PERFORMING ORGANIZATION CODE	
7. AUTHOR(S)				8. PERFORMING ORGANIZATION REPORT NO.	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Tom Faulkner BBS Consultants, Inc. Durham, NC 27701				10. PROGRAM ELEMENT NO.	
				11. CONTRACT/GRANT NO. EPA Contract No. 3D1540NASA	
12. SPONSORING AGENCY NAME AND ADDRESS U.S. Environmental Protection Agency Office of Air Quality Planning and Standards Technical Support Division Research Triangle Park, NC 27711				13. TYPE OF REPORT AND PERIOD COVERED Final Report	
				14. SPONSORING AGENCY CODE	
15. SUPPLEMENTARY NOTES EPA Work Assignment Manager: Herschel W. Rorex					
16. ABSTRACT This document provides standards and a guideline for the development of bulletin boards by Agency personnel or under contract for the OAQPS Technology Transfer Network (TTN). The objective is to provide for the development of bulletin boards for the TTN that are efficient and with no impact on the overall performance of the TTN.					
17. KEY WORDS AND DOCUMENT ANALYSIS					
a. DESCRIPTORS		b. IDENTIFIERS/OPEN ENDED TERMS		c. COSATI Field, Group	
TTN TBBS TDBS					
18. DISTRIBUTION STATEMENT Release Unlimited		19. SECURITY CLASS (This Report) Unclassified		21. NO. OF PAGES 22	
		20. SECURITY CLASS (This page) Unclassified		22. PRICE	