



BMS/TS 8 User's Guide

GT Software, Inc.
1314 Spring Street NW
Atlanta, Georgia USA 30309
404-253-1300
<www.gtsoftware.com>

Copyright and Legal Notices



Copyright © 1981, 2003 GT Software, Inc. All rights reserved.

Information in the product documentation is subject to change without notice and does not represent a commitment on the part of GT Software, Inc. The software described in the documentation is furnished under a license agreement. The software may be used or copied only in accordance with the terms of the agreement. It is against the law to copy the software on any medium except as specifically allowed in the license or nondisclosure agreement. The product documentation may be reproduced as needed by the licensee for use by or distribution to authorized employees of the licensee. All copies must include this copyright statement, or otherwise clearly indicate that GT Software holds the copyright to this material.

GT Software, Inc.
1314 Spring Street NW
Atlanta, Georgia USA 30309
Phone: 404-253-1300
Fax: 404-253-1314
E-mail: sales@gtsoftware.com or support@gtsoftware.com
URL: <http://www.gtsoftware.com/>

ASSIST/GT, NOVATION, and WQL/GT are registered trademarks of GT Software, Inc.

ASSIST/TS, AssistVision, BMS/GT, BMS/TS, DataAccess, DOCM/GT, Interactive Books, Interactive Portal for DB2, IPortal/GT, MENU/GT, Novation, PREVAL/GT, Process Extender for CICS, READ/GT for BookManager in CICS, tcACCESS, TUTOR/GT, VisualConnect, and WebBooks are trademarks of GT Software, Inc.

CICS, OS/390, z/OS, and S/390 are trademarks of IBM Corporation <www.ibm.com>.

Java and Solaris are trademarks of Sun Microsystems <www.sun.com/>.

UNIX is a registered trademark of The Open Group <www.opengroup.org/>.

Windows is a registered trademark of Microsoft Corporation <www.microsoft.com/>.

All other brand or product names are trademarks or registered trademarks of their respective companies.

Notice

This PDF version of the BMS/TS product documentation was produced using HTMLDOC
<<http://www.easysw.com/htmldoc/>>, which is available under the GNU General Public License
<<http://www.fsf.org/copyleft/gpl.html>>.

The complete BMS/TS documentation is supplied with the product in HTML format. This PDF is provided for the convenience of those customers who prefer hardcopy documentation.

BMS/TS User's Guide Home

[Introduction to BMS/TS](#)

[Working with Maps](#)

- [Design a Map](#)
- [Design a Map with Arrays](#)
- [Redesign a Map](#)
- [Redesign a Map Using Protected Redesign](#)
- [Display a Map](#)
- [Display Fields](#)
- [Build a Page](#)
- [Submit a Map](#)
- [Submit a Map using ADS](#)
- [List Maps in the Current Library](#)

[Working with Input Rules](#)

- [Input Rule Commands](#)
 - ◆ [ATTRIB](#)
 - ◆ [CALL](#)
 - ◆ [CURS](#)
 - ◆ [EDIT](#)
 - ◆ [FILL](#)
 - ◆ [IF, ELSE, ENDIF](#)
 - ◆ [JUST](#)
 - ◆ [LET](#)
 - ◆ [OPTION](#)
 - ◆ [PAD](#)
 - ◆ [REJ](#)
 - ◆ [RETN](#)
 - ◆ [SEL](#)
 - ◆ [STRIP](#)
 - ◆ [TROFF](#)
 - ◆ [TRON](#)
- [Common Uses of Input Rules](#)
- [Create Input Rules](#)
- [Edit Input Rules](#)
- [Test Input Rules](#)
- [Implement Input Rules](#)
- [Using Input Rules With Conversational Programs](#)

[Working with Tables](#)

- [Design a Table](#)
- [Redesign a Table](#)

- [Test a Table](#)
- [Submit a Table](#)
- [List Tables in the Current Library](#)
- [Using BMS/TS Tables in Your Application](#)

[Working with Demos](#)

- [Design a Demo](#)
- [Redesign a Demo](#)
- [Run a Demo](#)
- [List Demos in the Current Library](#)

[Working with Help Screens](#)

- [Create a Help Screen](#)
- [Chained Help Screens](#)
- [Selection \(Branched\) Help Screens](#)
- [Add a Help Screen to the Transaction Monitor](#)

[Performing Library Functions](#)

- [Manage Library Members in a Single Library](#)
 - ◆ [Copy Library Member](#)
 - ◆ [Display a Member](#)
 - ◆ [Erase Library Member](#)
 - ◆ [Rename Library Member](#)
 - ◆ [Reset Library Member](#)
 - ◆ [Add, Change, or Delete Password](#)
- [Transferring Library Members Between Libraries](#)
- [Link Libraries](#)
- [List Members in a Library](#)

[Working with the 3270 Bridge](#)

- [3270 Bridge Support](#)
- [BMS/TS Bridge Web Services](#)

[Appendix A, Using the BMS/TS Line Editor](#)

[Appendix B, Performing Batch Functions](#)

- Using [GTBCDMS](#) to Convert a DMS Panel
- Using [GTBCONV](#) to Convert/Import BMS Maps
- Using [GTBPUNCH](#) to Generate BMS Source Code
- Using the [GTBUTIL](#) Batch Utility Program
 - ◆ [ALTER](#)
 - ◆ [BACKUP](#)
 - ◆ [CATAL](#)
 - ◆ [COPY](#)

- ◆ CREATE
- ◆ DELETE
- ◆ DISKSEL
- ◆ DLIST
- ◆ MOVE
- ◆ PASSWRD
- ◆ PRINT
- ◆ PROD
- ◆ PUNCH
- ◆ RENAME
- ◆ RESET
- ◆ RESTORE
- ◆ SETFILE
- ◆ TAPESEL

[Appendix C, BMS/TS Transaction Codes](#)

[Appendix D, Support for Multi-Byte Character Sets](#)

[System Messages](#)

[Index](#)

[Getting Technical Support](#)

[Copyright and Legal Notices](#)

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Understanding Maps](#)
- [Access the Maps Menu](#)
- [Maps Menu Options](#)

Related topics:

- [Design a Map](#)
- [Design a Map with Arrays](#)
- [Redesign \(Edit\) a Map](#)
- [Display Map Fields](#)
- [Build Page](#)
- [List Maps](#)
- [Submit Maps](#)
- [Appendix D, Support for Multi-Byte Character Sets](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Working with Maps

The Maps Menu provides options which allow you to:

- Create and update maps using the BMS/TS Map Options and Copy/Move functions,
- Test maps created for page build applications using the Page Build Option,
- Submit maps to the CICS system library using the online Submit Option,
- Display a list of maps stored in the current BMS/TS library.

Understanding Maps

The symbolic map (also called DSECT for Assembler or copybook for COBOL, PL/1, and RPG) defines map fields used to store variable data referred to in a COBOL, PL/1, RPG, or Assembler language program. You place symbolic maps into a copy library and include them in the program when you compile the code. BMS/TS creates physical and symbolic maps (in the language you choose) which you can submit to your system library using the BMS/TS online [Submit Option](#).

Note: This document uses the term symbolic map for both DSECT and copybook.

BMS/TS keeps you from having to create BMS macro source code which you would then have to assemble in order to produce the physical and symbolic maps. If you want the BMS source, however, you can generate it using the BMS/TS [Macro Generator](#) batch function.

Access the Maps Menu

To start BMS/TS, enter the transaction GTB from a blank CICS screen. The BMS/TS Main Menu displays (view [sample](#)).

Select the Maps option to display the Maps Menu (view [sample](#)).

Maps Menu Options

When you select Maps from the BMS/TS main menu, the Maps menu displays the options described below.

- Design** Design a new map. The steps for designing a map include:
 - Naming the map,
 - Selecting options for the programming language, the types of terminals in your installation, and other characteristics of the map,
 - Positioning and typing literals (constants such as titles, field prompts, and labels) on the blank design screen, and indicating the start of input/output fields by positioning and typing a special character,
 - Defining fields with field names, lengths, data types, attributes, etc.,
 - Viewing the map exactly as it will appear to your end–users,
 - Designing your map with field, column, user–defined, or line arrays.

- Redesign** Change or modify elements of an existing map. To redesign a map, you follow the same steps used to design a map (except selecting the language and map options). BMS/TS provides a Copy/Move function to aid in the redesign of your maps.

- Protected Redesign** Change the literals (constants such as titles, field prompts, and labels), attributes, and initial values of an existing map. BMS/TS does not allow you to save changes to elements or information used by the application program (the symbolic map). This ensures that you do not have to alter the application program to accommodate changes made to your map.

- Display a Map** Display an existing map as it will appear to the end–user.

- Display Fields** Display a list of the fields defined in the map, including field names, data types, lengths, and other attributes.

- Build Page** Define a list of existing maps, and have BMS/TS display them as a single page. You can preview a page to ensure that you have created and placed the component maps correctly. However, you must still include paging routines in your application program.

- Submit** Catalog your maps to the CICS system library.

- List** Display a list of maps stored in the current BMS/TS library.

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Select a Map](#)
- [Define Map Options](#)
- [Design the Screen](#)
- [Define Fields](#)
- [Verify the Map](#)

Related topics:

- [Working with Maps](#)
- [Appendix D, Support for Multi-Byte Character Sets](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Design a Map

When you design a map, you will:

- Name the new map, and select or create a mapset name.
- Define options for the programming language, the user's terminal type, and other characteristics of your map.
- Design the screen literals (constants such as titles, field prompts, and labels), and indicate the start of input/output fields.
- Define map fields (field characteristics and attributes).
- Verify the map.

Select a Map

Begin by naming the new map, and selecting or creating a mapset name for your map.

1. From the Maps Menu, select the Maps option and press Enter to display the Map Options Menu (view [sample](#)).
2. From the Map Options Menu, select the Design option and press Enter to display the Select a Map screen (view [sample](#)).
3. Specify a mapset name (up to 7 characters) in the Mapset field.
Note: You can press F2 to display a list of mapsets in the current BMS/TS library.
4. Specify a map name (up to 7 characters) in the Map name field.
Note: you can press F2 to display a list of maps in the current BMS/TS library.
5. If you want to protect the map with a password, type the password in the Password field.
Note: If you assign a password to your map, you must enter the password each time you edit the map with BMS/TS.
6. Press Enter. BMS/TS displays the Map Definition Options screen (view [sample](#)).

Define Map Options

After you [select a map](#), the Map Definition Options screen appears (view [sample](#)). On this screen you will specify the map's characteristics.

1. Press Tab to move to the entry fields you want to modify, then type over the currently displayed value.

Note: Press F1 from any parameter or field on the screen to display online help. You can review a parameter or field's purpose, valid entries, and examples. Options for Language, Upper and Lower Case, Sound Alarm, Resets, Extended Attributes, and Page Build normally stay constant at a company, project, or application level.

2. Repeat step 1 until you make all of your modifications.
3. Press Enter. BMS/TS displays the Design screen (view [sample](#)).

Design the Screen

After you [define the map options](#), the Design screen appears (view [sample](#)). On this screen you will design the physical map and indicate the start of input/output fields.

1. When the Design screen displays, press the ErEOF key (Erase End of Field) to remove the displayed instructions and list of function key assignments. (The table below lists the [function keys](#) you can use in designing a map.)
2. Use the Arrow keys to move the cursor to the location on screen where you want to begin designing the map. Keep in mind that if you enter data in the first byte (row 1, column 1) of the map, you can enter attribute bytes only. Place literals (constants such as field prompts and labels) in any position other than the first byte.
3. Type literals (constants such as field prompts and labels).

Note: Type a Change Field Character to mark the first position of each input/output field. The default Change Field Character is a plus sign (+). However, you can use any other character (as long as the character does not appear on your map) by defining it as the Change Field Character on the [Map Definition Options](#) screen.

4. Place the cursor where you want it positioned when the map is displayed to the end-user.
5. Press Enter. BMS/TS displays the [Field Definition](#) screen for the first field marked with a Change Field Character.

Note: BMS/TS skips the Field Definition screen if your map contains no Change Field Characters (indicating input or output fields on screen).

The following table lists the function keys you can use to help design the map.

Fkey	Description
F1	Display help for a specific field. Press F1 again for screen-level help.
F2	Duplicate the line where the cursor is positioned.
F3	Assign default values to all fields and advances directly to the Map Verification screen.
F4	Return to the Map Definition Options screen.
F5	

Check. Tells BMS/TS that you have modified current fields (usually during an update) using Insert and/or Delete. For this reason, BMS/TS then realigns the symbolic and physical maps so that no information is lost when you exit the screen.

- F6** Save your work up to this point.
 - F7** Delete the line where the cursor is positioned.
 - F8** Refresh the screen from the previously saved image.
 - F9** Insert a blank line where the cursor is positioned.
 - F10** Go to the Copy/Move Functions screen.
-

Define Map Fields

After you [design the screen](#), the Field Definition screen appears (view [sample](#)). On this screen you will define field characteristics and attributes for your map.

Note: BMS/TS displays an altered version of the Field Definition screen if you use the Assembler language. These variations are: the Edit Pattern fields do not display; the name field is 8 bytes long.

The top section of the Field Definition screen shows the map you designed, with the first field highlighted. BMS/TS places a ? at the highlighted field to identify the field you are currently defining.

1. Type a descriptive name in the field name field.
 2. Press Tab to move to the parameters you want to select or change, then type over the currently displayed value. You must designate a field size if your program refers to a field.
 3. After you define all of the parameters for this field, press Enter. BMS/TS moves to the next field containing a Change Field Character --- marking it with a "?" --- so you can define its parameters.
 4. Repeat steps 1 through 3 until you have defined the parameters for all of the fields on this screen.
 5. Press Enter. BMS/TS displays the [Map Verification](#) screen.
-

Verify the Map

After you [define the map fields](#), the Map Verification screen appears (view [sample](#)). This screen displays the new map as it will appear in the application. You can then visually verify the layout of your map, as well as type in sample data to test edit patterns and data type restrictions.

Note: When present, protected fields appear with a V at every fifth byte, a number at every tenth byte and four asterisks between, for the length of your defined field:
(****V****1****V****2 etc.)

1. Review the displayed data to verify the appearance of the map.

2. (optional) You can enter sample data into the fields displayed on the screen.

Note: BMS/TS does not validate the sample data unless you have written and implemented [input rules](#) for the map. BMS/TS limits only the length of the field according to the definition you have specified.

3. After keying sample data into all desired fields, press Enter. BMS/TS displays the sample data as it would appear when your map is being used.

4. Review the displayed data to verify the appearance of the map. At this point, you have the following options:

- ◆ If you want to save the map, continue at step 5.
- ◆ If the map's appearance is acceptable but you want to reposition the cursor on screen, continue at step 6.
- ◆ If you want to display all spaces you placed in the map as question marks (?), continue at step 7.
- ◆ If you want to correct a field prompt, label, or change a field definition, continue at step 8.

5. If the map's appearance is acceptable, press Enter. BMS/TS saves the map and redisplay the Select a Map screen.

Note: BMS/TS takes you straight to the [Input Rule Editor](#) screen when you press Enter if you selected Y (Yes) for input validation rules on the Map Definition Options screen.

6. If the map's appearance is acceptable but you want to reposition the cursor on screen, press F2, then place your cursor where you want it positioned for the end-user when the map appears on screen. Press F2 again to accept this cursor position, then continue at step 5 of this section.

7. If you want to display all spaces you placed in the map as question marks (?), press F5. To redisplay the screen without the question marks, press F5 again. Then continue at step 5 of this section. Note: This shows you which bytes of the map are true spaces (hexadecimal 40) instead of nulls (hexadecimal/binary 00).

8. If you want to correct a field prompt, label, or change a field definition, press F4. The Map Verification screen redisplay. Notice that the fields of the map are now preceded with an @. The @ character is the Start Field Character. It precedes all input and/or output fields.

9. If you want to correct a literal, perform the following steps. If you want to correct a field definition, continue at step 10.

- a. Use the Arrow keys to move the cursor to the field prompt or label you want to change.
- b. Enter the correct information over the existing literal.
- c. Press Enter. BMS/TS redisplay the Map Verification screen with the new information.
- d. Repeat steps 1–5 to verify the map with the new information.

10. To correct a field definition, perform the following steps:

- a. Press Tab to move to the field you want to redefine.
- b. Type a Change Field Character (default is +) in front of all fields that you want to change, then press Enter.
- c. BMS/TS displays the Field Definition screen for each field marked with a Change Field Character. (Keep in mind that the displayed fields are still preceded by @ characters.)

d. d. After redefining all desired fields, press Enter. BMS/TS returns to the Map Verification screen. Repeat steps 1–5 to verify the map with the new information.

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Design a Map with Field Arrays](#)
- [Design a Map with Column Arrays](#)
- [Design a Map with User-Defined Arrays](#)
- [Design a Map with Line Arrays](#)

Related topics:

- [Working with Maps](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Design a Map with Arrays

You can define BMS/TS maps with field, column, user-defined, or line arrays.

- **Field Arrays** -- Select a field to repeat, and the number of times BMS/TS should repeat the field, side-by-side across a map.
- **Column Arrays** -- Select a field to repeat, and the number of times BMS/TS should repeat the field, down the map.
- **User-defined Arrays** -- Select a field to repeat, and the number of times BMS/TS should repeat the field, anywhere on the map.
- **Line Arrays** -- Select a line to repeat, and the number of times BMS/TS should repeat the line, down the map. Keep in mind that a line can contain multiple fields.

Design a Map with Field Arrays

Designing a map with field arrays lets you select a field to repeat, and the number of times BMS/TS should repeat the field, side-by-side, across a map.

Perform the following steps to design a map with field arrays. To begin, you will [select a map](#), [define map options](#), and [design the screen](#). When you complete the map design screen, the [Field Definition](#) screen appears (view [sample](#)).

The top section of the Field Definition screen contains the map you have created, with the first field highlighted. BMS/TS places a "?" at the highlighted field to identify the field you are currently defining.

1. Type a descriptive name in the field name field.
2. Press Tab to move to the Field Occurs field. This parameter lets you specify the total number of times you want the current field to appear on the map.

Note: BMS/TS uses the ending attribute of the current field for the beginning attribute of all other fields in the array.

3. Indicate the number of times you want the field to appear and press Tab.

For example, if you want the field to appear a total of 3 times side-by-side across the map, type 3 and press Tab.

The cursor moves to the Field Occurs Type field.

4. Identify the type of array: **F** (field array), **C** (column array), or **U** (user-defined array). Type F to define a field array.
 5. Press Tab to move to any other parameters you want to select or change, then type over the currently displayed value. Keep in mind that you must designate a field size if your program refers to a field.
 6. After you define all of the parameters for this field, press Enter. BMS/TS moves to the next field containing a Change Field Character -- marking it with a "?" -- so you can begin defining its parameters.
 7. Repeat steps 1–6 until you have defined the parameters for all of the fields on this map.
 8. Press Enter. BMS/TS displays the Map Verification screen. The map now contains the field array.
 9. [Verify](#) the map.
-

Design a Map with Column Arrays

Designing a map with column arrays lets you select a field to repeat, and the number of times BMS/TS should repeat the field down the map.

Perform the following steps to design a map with column arrays. To begin, you will [select a map](#), [define map options](#), and [design the screen](#). When you complete the map design screen, the [Field Definition](#) screen appears (view [sample](#)).

The top section of the Field Definition screen contains the map you have created, with the first field highlighted. BMS/TS places a "?" at the highlighted field to identify the field you are currently defining.

1. Type a descriptive name in the field name field.
2. Press Tab to move to the Field Occurs field. This parameter lets you specify the total number of times you want the current field to appear on the map.

Note: BMS/TS uses the ending attribute of the current field for the beginning attribute of all other fields in the array.

3. Indicate the number of times you want the field to appear and press Tab. For example, if you want the field to appear 3 times down the map, type 3 and press Tab.

The cursor moves to the Field Occurs Type field.

4. Identify the type of array: **F** (field array), **C** (column array), or **U** (user-defined array). Type C to define a column array.
5. Press Tab to move to any other parameters you want to select or change, then type over the currently displayed value. Keep in mind that you must designate a field size if your program refers to a field.

6. After you define all of the parameters for this field, press Enter. BMS/TS moves to the next field containing a Change Field Character --- marking it with a "?" --- so you can begin defining its parameters.
 7. Repeat steps 1–6 until you have defined the parameters for all of the fields on this map.
 8. Press Enter. BMS/TS displays the Map Verification screen. The map now contains the column array.
 9. [Verify](#) the map.
-

Design a Map with User–Defined Arrays

Designing a map with user–defined arrays lets you select a field to repeat, and the number of times BMS/TS should repeat the field, anywhere on the map. This gives you the greatest flexibility in designing a map with arrays.

Perform the following steps to design a map with user–defined arrays. To begin, you will [select a map](#), [define map options](#), and [design the screen](#).

When you complete the map design screen, the [Field Definition](#) screen appears (view [sample](#)).

The top section of the Field Definition screen contains the map you have created, with the first field highlighted. BMS/TS places a "?" at the highlighted field to identify the field you are currently defining.

1. Type a descriptive name in the field name field.
2. Press Tab to move to the Field Occurs field. This parameter lets you specify the total number of times you want the current field to appear on the map.

Note: BMS/TS uses the ending attribute of the current field for the beginning attribute of all other fields in the array.

3. Indicate the number of times you want the field to appear and press Tab. The cursor moves to the Field Occurs Type field.
4. Identify the type of array: **F** (field array), **C** (column array), or **U** (user–defined array). Type U to define a user–defined array.
5. Press Tab to move to any other parameters you want to select or change, then type over the currently displayed value. Keep in mind that you must designate a field size if your program refers to a field.
6. After you define all of the parameters for this field, press Enter. BMS/TS redisplay the Design screen.
7. Use the Arrow keys to move the cursor to the location where you want the first field to appear and press Enter. Note: You cannot place a user–defined array to the left, or above, the original field or line on which you are basing the array.

BMS/TS again displays the Design screen, which now includes the first array.

8. After you have defined all of the arrays for this map, BMS/TS again redisplay the Design screen, which now contains all of the current arrays. The message Press Enter to accept the marked occurrences appears at the bottom of the screen.

Verify the placement of the arrays and press Enter.

9. BMS/TS returns to the Field Definition screen so that you can now define the parameters for the next field containing a Change Field Character.

Notice that BMS/TS continues to prompt you to position the cursor for the location of each array until you have defined all of the arrays. Remember, you cannot place a user-defined array to the left, or above, the original field or line on which you are basing the array.

Note: The user-defined arrays for the first field now display as * characters.

10. Repeat steps 1 through 9 until you have defined the parameters for all of the fields on this screen.
11. Press Enter. BMS/TS displays the Map Verification screen. The map now contains the user-defined array.
12. [Verify](#) the map.

The table below lists the function keys you can use while defining arrays.

Fkey	Description
F1	Display screen level help for this screen.
F3	Cancel this function and exit this screen.
F5	Refresh the screen. Removes the instruction line from the bottom of the map. Press F5 again to redisplay the instruction line.
F11	Undo. Cancel all arrays you have placed on the screen, then redisplay the screen so you can reposition the arrays.

Design a Map with Line Arrays

Designing a map with line arrays lets you select a line to repeat, and the number of times BMS/TS should repeat the line, in a map. Line arrays can contain multiple fields.

Perform the following steps to design a map with line arrays. To begin, you will [select a map](#) and [define map options](#). After you define the map options, the map design screen appears (view sample).

1. Press ErEOF to remove the displayed instructions and list of function key assignments.
2. [Design](#) the map.
3. Type literals (constants such as field prompts and labels).

Note: Type a Change Field Character to mark the first position of each input/output field. The default Change Field Character is a plus sign (+). However, you can use any character you like by defining the Change Field Character at the [Map Definition Options](#) screen.

4. Place a Change Field Character (default is +) as the first character of the line being used to define a line array.
5. Place the cursor where you want it positioned when the map appears on screen for the end-user.

For example, if you want the map to display to the end-user in the first field that requires data, place the cursor in the first position of the first input field on the map.

6. Press Enter. BMS/TS displays the [Field Definition](#) screen (view [sample](#)).

The top section of the Field Definition screen contains the map you have created. Notice that BMS/TS places a "?" at the beginning of the line you are using to define a line array.

7. Type a descriptive name in the field name field.
8. Press Tab to move to the Line Occurs field. This parameter lets you specify the total number of times you want the current line to appear on the map.
9. Indicate the number of times you want the line to appear and press Tab. For example, if you want to repeat this line 3 times down the map, type 3 and press Tab.
10. Press Tab to move to any other parameters you want to select or change, then type over the currently displayed value.
11. After you define all of the parameters for this line array, press Enter. BMS/TS moves to the next field containing a Change Field Character -- marking it with a "?" -- so you can begin defining its parameters.

Note: If the next field containing a Change Field Character happens to fall within the line array, BMS/TS identifies the field with the message *Array Member*.

12. Repeat steps 7–11 until you have defined the parameters for all of the fields on this screen.
13. Press Enter. BMS/TS displays the Map Verification screen. The map now contains the line array.
14. [Verify](#) the map.

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Select a Map to Redesign](#)
- [Redesign \(Edit\) a Map](#)
- [Copy/Move a Field or Block of Fields](#)

Related topics:

- [Working with Maps](#)
- [Appendix D, Support for Multi-Byte Character Sets](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Redesign (Edit) a Map

This section describes how to modify an existing map with BMS/TS. You can:

- Update, modify, and/or correct the map.
- Add, delete, or move fields, change field attributes, and change order and/or lengths of fields.
- Revise literals (constants such as titles, field prompts, and labels).
- Redefine fields by selecting or accepting various field characteristics and/or attributes.
- Verify the revised map as it will appear in your application, and enter sample data to test edit patterns and data type restrictions.

Keep in Mind!

You do not need to reassemble or recode the application program if your changes are limited to literals, repositioning fields without changing their order, or changing extended attributes.

However, you must modify the application code if you add or delete fields referred to by your program.

Select a Map to Redesign

1. From the Main Menu, select the Maps option and press Enter.
2. From the Maps Menu, select the Maps option and press Enter.
3. From the Map Options Menu, select the Redesign option and press Enter. The Select a Map screen displays (view [sample](#)).
4. Type a mapset name (up to 7 characters) in the Mapset field.

Note: To display a list of mapsets in the current BMS/TS library, press F2.

5. Type a map name (up to 7 characters) in the Map name field. To display a list of maps in the current BMS/TS library, press F2.
 6. If you password-protected the map, type the password in the Password field.
 7. Press Enter. BMS/TS displays the map you want to redesign.
-

Redesign (Edit) a Map

Redesigning a map lets you modify the appearance and/or content of an existing map.

After you [select](#) a map to redesign, BMS/TS displays the map you want to redesign.

Note: BMS/TS precedes each field in the above sample with a Start Field Character (@). Unless you change the default values on the Map Definition Options screen when you initially design the map, BMS/TS uses the @ sign as the Start Field Character.

1. Use the Arrow keys to position the cursor at the field you want to change.
2. At this point, you have the following options:
 - a. To modify literals or a field's initial values, simply type over the existing data. Keep in mind that if you make an invalid change, BMS/TS restores the original contents of the field.
 - b. To redefine fields, type a Change Field Character (the default is +) before each field you want to change.
3. Repeat steps 1 and 2 to modify any other fields on the screen.
4. Place the cursor where you want it positioned when the map appears on screen for the end-user.
5. Press Enter. BMS/TS displays one of two screens:
 - a. If you added or modified any field on your map, the Field Definition screen appears.
 - b. Otherwise, BMS/TS displays your modified map on the Map Verification screen.
6. Verify the appearance of the map and press Enter. BMS/TS returns to the Select a Map screen.

The table below lists the functions keys that are available as you redesign the map.

Function Key	Description
F1	Help. Displays help information for a specific field. Press F1 again for help concerning the Redesign screen.
F2	Dup Line. Duplicates the line where the cursor is positioned.
F3	Skip Fld. Assigns default values to all fields (new, modified, or unchanged) and advances directly to the Map Verification screen.
F4	Options. Returns to the Map Definition Options screen.
F5	Check. Tells BMS/TS that you have modified current fields (usually during an update) using Insert and/or Delete. For this reason, BMS/TS then realigns the symbolic and physical maps so that no information is lost when you exit the screen.
F6	Hold. Saves your work up to this point.
F7	Del line. Deletes the line where the cursor is positioned.
F8	Undo. Refreshes the screen from the previously saved image.

- F9** Add Line. Inserts a blank line where the cursor is positioned.
- F10** Action. Goes to the [Copy/Move Functions Menu](#).

Copy/Move a Field or Block of Fields

You can copy or move a field, or a block of fields, to another location on your map.

After you [select](#) a map to redesign, BMS/TS displays the map you want to redesign.

1. Press F10 to display the Copy/Move Functions Menu (view [sample](#)).
2. Select the appropriate option from the menu:

If you want to...	Then...
Copy a field to another location	Select the Copy Field option and press Enter.
Move a field to another location	Select the Move Field option and press Enter.
Copy a block of fields to another location	Select the Copy Block option and press Enter.
Move a block of fields to another location	Select the Move Block option and press Enter.

BMS/TS then redisplay your map.

Note: BMS/TS provides an instruction line at the bottom of your map to walk you through the copy/move process. The instructions BMS/TS displays depend on the function you have chosen.

3. Move the cursor to the location on screen from which you want to begin copying or moving.

If you want to...	Then...
Copy or move a field to another location	<ol style="list-style-type: none"> 1. Use the Arrow keys to position your cursor at the Start Field Character (@) of the field you want to copy or move. 2. Press Enter.
Copy or move a block of fields to another location	<ol style="list-style-type: none"> 1. Use the Arrow keys to position your cursor at the top left corner of the block of fields you want to copy or move. 2. Press Enter. 3. Mark the end of the block by moving to the bottom right corner of the block. 4. Press Enter.

4. Use the Arrow keys to position your cursor at the location where you want BMS/TS to copy or move the field or block of fields.
5. Press Enter. BMS/TS copies or moves the field, or block of fields, to the desired location on

your map.

Note: When you move a field or a block of fields, BMS/TS deletes the original field or block of fields.

You can **not** copy or move a field or a block of fields into the middle of a group of fields, line array, or field array. You must move the field to a location where there are no existing fields.

When copying or moving a field, BMS/TS does **not** copy or move the Ending Attribute of a field. You must move this manually.

6. Verify the location of the copied or moved fields and press Enter. BMS/TS returns to the Copy/Move Functions Menu.
7. Press F3. BMS/TS returns to your map. Notice that it now reflects the changes you made when you copied or moved a field or block of fields.
8. You can continue to [redesign the map](#).

The table below lists the function keys that are available during the copy/move operations.

Function Key	Description
F1	Help. Provides screen level help for this screen.
F3	Exit. Cancels and returns you to the Redesign screen without saving your changes.
F5	Refresh. Removes the instruction line from the bottom of your map. Press F5 again to redisplay the instruction line.
F11	Undo. Lets you cancel the previous step of the Copy/Move function.

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Select a Map for Protected Redesign](#)
- [Edit a Map Using Protected Redesign](#)

Related topics:

- [Working with Maps](#)
- [Appendix D, Support for Multi-Byte Character Sets](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Edit a Map using Protected Redesign

The Protected Redesign feature lets you modify a map safely, so that the modification does not force you to recode your application.

Specifically, this feature restricts you from changing the order or size of any field referred to by your application. And, if you make an invalid change, BMS/TS restores the contents of the original field.

Note: You cannot change the order or size of named fields (fields referred to by your application program).

Select a Map for Protected Redesign

1. From the Main Menu, select the Maps option and press Enter.
2. From the Maps Menu, select the Maps option and press Enter.
3. From the Map Options Menu, select the Redesign (Protected) option and press Enter. The Select a Map screen displays (view [sample](#)).
4. Type a mapset name (up to 7 characters) in the Mapset field.

Note: To display a list of mapsets in the current BMS/TS library, press F2.

5. Type a map name (up to 7 characters) in the Map name field.

Note: To display a list of maps in the current BMS/TS library, press F2.

6. If you password-protected the map, type the password in the Password field.
7. Press Enter. BMS/TS displays the map you want to redesign.

Edit a Map Using Protected Redesign

Redesigning a map with protected redesign lets you modify the appearance and/or content of an existing map. However, you cannot change the order and size of named fields (those referred to by the application program).

After you [select](#) a map for protected redesign, BMS/TS displays the selected map.

Note that BMS/TS precedes each field with a Start Field Character (@). Unless you change the default values on the Map Definition Options screen when you initially design your map, BMS/TS uses the @ sign as the Start Field Character.

1. Use the Arrow keys to move the cursor to the area of the map you want to change.
2. At this point, you have the following options:
 - a. To modify literals or a field's initial values, simply type over the existing data. Keep in mind that if you make an invalid change, BMS/TS restores the original contents of the field.
 - b. To redefine fields, type a Change Field Character (the default is +) before each field you want to change.
3. Repeat steps 1 and 2 to modify any other fields on the screen.
4. Place the cursor where you want it positioned when the map appears to the end-user.
5. Press Enter. BMS/TS displays one of two screens:
 - a. If you added or modified any field on your map, the Field Definition screen appears. You can only change the Field Intensity, Color, and Hilite field definitions.
 - b. If you modified literals or a field's initial values, BMS/TS displays the modified map on the Map Verification screen.
6. Verify the appearance of the map and press Enter. BMS/TS returns to the Select a Map screen.

The table below lists the functions keys that are available as you redesign the map.

Function Key	Description
F1	Help. Displays help information for a specific field. Press F1 again for help concerning the Redesign screen.
F2	Dup Line. Duplicates the line where the cursor is positioned.
F3	Skip Fld. Assigns default values to all fields (new, modified, or unchanged) and advances directly to the Map Verification screen.
F4	Options. Returns to the Map Definition Options screen.
F5	Check. Tells BMS/TS that you have modified current fields (usually during an update) using Insert and/or Delete. For this reason, BMS/TS then realigns the symbolic and physical maps so that no information is lost when you exit the screen.
F6	Hold. Saves your work up to this point.
F7	Del line. Deletes the line where the cursor is positioned.
F8	Undo. Refreshes the screen from the previously saved image.
F9	Add Line. Inserts a blank line where the cursor is positioned.
F10	Action. Goes to the Copy/Move Functions Menu .

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Select a Map to Display](#)
- [Review the Map](#)

Related topics:

- [Working with Maps](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Display a Map

Selecting a map lets you choose an existing map to display. BMS/TS displays the map exactly as it will appear in your application.

You can verify the layout of the map, and type in sample data to test edit patterns and data type restrictions.

Select a Map to Display

1. From the Main Menu, select the Maps option and press Enter.
2. From the Maps Menu, select the Maps option and press Enter.
3. From the Map Options Menu, select the Display a Map option and press Enter. The Select a Map to Display screen displays (view [sample](#)).
4. Type a mapset name (up to 7 characters) in the Map set field. To display a list of mapsets in the current BMS/TS library, press F2.
5. Type a map name (up to 7 characters) in the Map name field.
Note: To display a list of maps in the current BMS/TS library, press F2.
6. If you password-protected the map, type the password in the Password field.
7. Press Enter. BMS/TS displays the desired map.

Review the Map

Reviewing a map lets you review a displayed map. You can verify that the layout of your map is accurate, and type in sample data to test edit patterns and data type restrictions.

After you select a map to display, the map displays as it will appear in the application.

Note: When present, protected fields appear with a V at every fifth byte, a number at every tenth byte and four asterisks between, for the length of your defined field (****V****1****V****2 etc.).

1. Review the displayed data to verify the appearance of the map.

2. If desired, you can enter sample data into the fields displayed on the screen.

Note: BMS/TS does not validate the sample data unless you have written and implemented [input rules](#) for the map. BMS/TS limits only the length of the field according to the definition you have specified.

3. After keying sample data into all desired fields, press Enter. BMS/TS displays the sample data as it would appear when your map is being used.

4. Press F3. BMS/TS takes you back to the Select a Map to Display screen.

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Select a Map to Display Field Names](#)
- [Review Map Fields](#)

Related topics:

- [Working with Maps](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Display Map Fields

You can display the fields of your map, in the order in which they occur.

You can also view the associated field attributes for the specified map.

Note: This procedure is for review purposes only. You cannot update any fields.

Refer to [Redesign a Map](#) or [Redesign a Map with Protected Redesign](#) if you need to edit fields.

Select a Map to Display Field Names

Selecting a map lets you choose an existing map whose fields you want to display. You can review the map's fields with their associated attributes.

1. From the Main Menu, select the Maps option and press Enter.
2. From the Maps Menu, select the Maps option and press Enter.
3. From the Map Options Menu, select the Display Fields option and press Enter. The Display Field Names screen displays (view [sample](#)).
4. Type a mapset name (up to 7 characters) in the Set Name field. To display a list of mapsets in the current BMS/TS library, press F2.
5. Type a map name (up to 7 characters) in the Map Name field. To display a list of maps in the current BMS/TS library, press F2.
6. If you password-protected the map, type the password in the Password field.
7. Press Enter. BMS/TS displays a table containing the fields from the specified map (view [sample](#)).

Review Map Fields

Reviewing fields lets you inspect the attributes of the fields on your map.

Note: This procedure is for review purposes only. You cannot update any fields. Refer to [Redesign a Map](#) or [Redesign a Map with Protected Redesign](#) if you need to edit fields.

After you [select](#) a map to display field names, the Field Name Display screen displays (view [sample](#)). This screen includes a table that contains the fields from the desired map.

The Field Name Display screen includes the following information for each field on your map:

Information	Description
Field Name	Name of the field as defined on the Field Definition screen.
Row	Location of the field on map: starting row number.
Col	Location of the field on map: starting column number.
Len	Length of the field in bytes.
Attributes	Extended field attributes: <ul style="list-style-type: none">• Field Intensity (Normal, Bright, Dark)• Field Type (Autoskip, Protected, Normal, Unprotected)• Field Color (Blue, Red, Purple, Green, Turquoise, Yellow, Neutral)• Field Hilite (Blinking, Reverse Video, Underline)

You can use the following function keys to help you review fields:

Function Key	Description
F1	Help. Display help information for this screen.
F3	Exit. Cancel this option and returns to the Display Field Names screen.
F5	Rloc. Repeat a previous search.
F7	Bkwd. Scroll backward one screen (if applicable).
F8	Fwd. Scroll forward one screen (if applicable).
F9	ScHf. Toggle between half-page and full-page scrolling.
F12	Cursor. Move the cursor to the command line.

When you are finished reviewing the map fields, press F3. BMS/TS returns to the Display Field Names screen.

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Perform the Page Build](#)
- [Verify the Page Build](#)
- [View the Sample Page–Build Map](#)

Related topics:

- [Working with Maps](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Build a Page

When you build a page, you test the layout of individual maps that combine to create a single map.

Before You Begin

You must code all appropriate paging commands in your application program.

When designing the individual maps for the page, you must select the appropriate page–build map definitions for each map. These definitions include:

- Map start row,
- Map start column,
- Map type,
- Map justification.

That is, you need to know where each map starts, the type of map, and where the map appears on screen.

Press F1 from the Map Definition Options screen for more information on these map definitions.

Perform Page Build

Performing a page build lets you select existing mapsets and maps for your page.

1. From the Main Menu, select the Maps option and press Enter.
2. From the Maps Menu, select the Maps option and press Enter.
3. From the Map Options Menu, select the Page Build option and press Enter. The Page Build screen displays (view [sample](#)).

Notice that Mapset and Member are the only two columns which contain entry fields.

4. Type the first mapset name (up to 7 characters) for the page in the Mapset field. (If all the maps for the page belong to the same set, you only need to enter the mapset name once.)
5. Type the first map name (up to 7 characters) for the page in the Member field. (If you are using all of the members in a set, place an asterisk (*) in this field.)

Note: Your entries for the Page Build options on the Map Definition Options screen determine the data displayed in the Map type, Justify, and Comments fields of this screen.

6. Press Tab to move to line 2 of the Mapset column.
 7. Repeat steps 1 through 3 until you have listed all of the maps for your page.
 8. Press Enter. BMS/TS displays your page.
-

Verify the Page Build

Page build verification lets you visually check the layout and integration of the various map components of your page.

After you complete the [page build](#), the Page Build Verification screen appears.

After you verify the appearance of the map, press Enter. BMS/TS returns to the Page Build screen.

View the BMS/TS Sample Page–Build Map

Viewing the sample page–build map lets you visually check the layout and integration of the various map components of a sample page.

1. From the Main Menu, select the Maps option and press Enter.
2. From the Maps Menu, select the Maps option and press Enter.
3. From the Map Options Menu, select the Page Build option and press Enter. The Page Build screen displays (view [sample](#)).
4. Type GTBDEMO in the Mapset field.
5. Type * in the Member field.
6. Press Enter to display the [sample page–build map](#).

Note: The sample map consists of the header, the body (detail) of the map, and the trailer. Keep in mind that you could use the header and trailer repeatedly and change the detail map for a series of maps.

7. Verify the appearance of the map and press Enter. BMS/TS returns to the Page Build screen.
-

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Sample Page-Build Map

```

*****
*                                     * *                                     *
*           Header map1                * *           Header map2                *
*           GTBDEMO H1                  * *           GTBDEMO H2                  *
*****
*****
*                                     * *                                     *
*           Header map3                * *           Header map4                *
*           GTBDEMO H3                  * *           GTBDEMO H4                  *
*****
*****
This would be a detail line one          GTBDEMO D1
This is detail line two                   GTBDEMO D2
This would be a detail line one          GTBDEMO D1
This is detail line two                   GTBDEMO D2
This would be a detail line one          GTBDEMO D1
This is detail line two                   GTBDEMO D2
This would be a detail line one          GTBDEMO D1
This is detail line two                   GTBDEMO D2
This would be a detail line one          GTBDEMO D1
*****
*                                     * *                                     *
*                                     * *           Trailer map                *
*                                     * *           GTBDEMO T1*                *
*****

```

Contents:

- [Select a Map to Submit](#)
- [Submit a Map](#)

Related topics:

- [Working with Maps](#)
- [Support for Multi-Byte Character Sets](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Submit a Map

This section describes how to catalog maps in your CICS system library.

Select a Mapset/Map to Submit

Selecting a map lets you choose the map(s) to catalog in your CICS system library. Perform the following steps to select a map to submit.

1. From the Main Menu, select the Maps option and press Enter.
2. From the Maps Menu, select the Maps option and press Enter.
3. From the Map Options Menu, select the Submit option and press Enter. The Select a Map to Submit screen displays (view [sample](#)).
4. Type the mapset name (up to 7 characters) of the map you want to submit in the Set name field. To display a list of mapsets in the current BMS/TS library, press F2.
5. Because you should submit complete mapsets, you should generally leave the Member name field blank. However, if you want a specific map to be resident in memory, type the name of the map (up to 7 characters) in the Member name field.

Note: You can only submit Assembler language maps as resident maps.

6. If you password-protected the map, type the password in the Password field.
 7. Press Enter. BMS/TS displays the Batch Job Submission screen (view [sample](#)).
-

Submit a Mapset/Map

You can specify the parameters to submit a group of BMS/TS maps to the CICS system library.

When you [select](#) a map to submit, the Batch Job Submission screen appears. Note that the screen you see depends on your operating system (view [sample](#) screen).

Note that BMS/TS automatically places the set and member names in the appropriate fields based on the information you entered in the Select a Map to Submit screen.

1. Type the name of the JCL member you want to use in the JCL member name field, or press Tab to accept the default (**SUBMIT**). This name must coincide with the library names used in your company.
2. Type any BMS/TS Options, Mapset Definition Options, or JCL Parameters Options that you want to change in the appropriate fields, or press Tab to accept the defaults.
3. Press Enter. BMS/TS begins submitting the map and displays the message Job Submission in Progress. When the submit is complete, the Operation Complete message appears and the cursor returns to the Set or Type name field.
4. Type the mapset name (up to 7 characters) of the next map you want to submit in the Set name field. To display a list of mapsets in the current BMS/TS library, press F2.
5. Because you should submit complete mapsets, you should generally leave the Member name field blank. However, if you want a map to be resident in memory, type the name of the map (up to 7 characters) in the Member name field.

Note: You can only submit Assembler language maps as resident maps.

6. Repeat steps 1–5 until you have completed submitting your maps.
7. Press F3. BMS/TS returns to the Map Options menu.

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Select a Map to Submit](#)
- [Submit a Map](#)

Related topics:

- [Submit a Map](#)
- [Working with Maps](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Submit a Map with ADS (3270 Bridge Support)

Though the map submission process is the same for ADS maps, there are some additional considerations.

Select a Mapset/Map to Submit

On the Select a Map to Submit screen (Maps> Maps> Submit), you specify the mapset name.

Note: The IBM 3270 Bridge requires a suffix which is appended to the mapset name. If your mapset name is 7 characters, the suffix is limited to one additional character (A–Z, 0–9). This means that you can have no more than 36 maps in the mapset.

If your mapset name is 6 (or fewer) characters, the suffix can be two additional characters (A–Z, 0–9, AA–ZZ, 00–99). You can have as many as 72 maps in the mapset.

After you complete this screen and press Enter, BMS/TS displays the Batch Job Submission screen (view [sample](#)).

Submit a Mapset/Map

On the Batch Job Submission screen, you can specify the parameters to submit BMS/TS maps to the CICS system library.

1. Type the name of the JCL member you want to use in the JCL member name field, or press Tab to accept the default (**SUBMIT**). This name must coincide with the library names used in your company.

Refer to the **SUBMSHTM** member on the GT\$FILE for a sample of the JCL needed to submit maps for use with the 3270 Bridge. The JCL must include the @HTML parameter and may include the following options:

BRCOMP Indicates whether you want to compress the HTML code.

- ◆ BRCOMP=Y compresses the HTML; this is the default value.

- ◆ BRCOMP=N does not compress the HTML.

Note: HTML is generated, like the BMS macros, with each tag on its own line. This creates requirements for large buffer sizes used by the CICS Bridge. To reduce these buffer requirements, use BRCOMP=Y to combine tags on lines where room allows.

BRLOOK Indicates how you want the HTML screens to display.

- ◆ BRLOOK=IBM produces a light gray background similar to that of the BMS macro template generation.
- ◆ BRLOOK=GT produces a black background similar to a 3270-style display. This is the default value.

Remember, you can edit the resulting HTML to customize the displays.

BRTYPE Indicates the type of field names.

- ◆ BRTYPE=1 represents simple field names as implemented in Transaction Server version 1.
Type 1 example: &FLDNAME (uses full field name).
- ◆ BRTYPE=2 represents the field naming conventions used in Transaction Server version 2 in *rrcccllll* (row, column, length) format.
Type 2 example: &F010020032_FLDNAME (uses up to 21 characters of name).

If you are unsure which to use, BRTYPE=1 is the only option supported by all versions of Transaction Server. Default is 1.

2. Type any BMS/TS Options, Mapset Definition Options, or JCL Parameters Options that you want to change in the appropriate fields, or press Tab to accept the defaults.

You must set Generate ADS to **Y** and 3270 Template to **Y**.

3. Press Enter. BMS/TS begins submitting the mapset and displays the message Job Submission in Progress. When the submit is complete, the Operation Complete message appears and the cursor returns to the Set or Type name field.

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- List Maps in the Current Library

Related topics:

- [Working with Maps](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

List Maps in the Current Library

You can view a list of all maps stored in your current library using the Listing of Maps screen.

Note: You can also use the [Listing of Members](#) screen to perform file maintenance on a map in the library.

List Maps in the Current Library

To list all maps in the current library:

1. From the Main Menu, select the Maps option and press Enter.
2. From the Maps Menu, select the Maps option and press Enter.
3. From the Map Options Menu, select the List option and press Enter. The Listing of Members screen displays (view [sample](#)).
4. Review the [displayed information](#) (described below). You can use the [function keys](#) listed at the bottom of the screen to help you review the list of members.
5. To view a specific mapset, type the mapset name at the command prompt and press Enter. The list scrolls to the desired set name.
6. To exit this screen, press F3 to return to the Map Options Menu.

The Listing of Members screen includes the fields described in the table below.

Field	Description
?	Entry field for single character command. Press F1 for details.
Set/Type	Mapset name.
Member	Map name.

Flags	Status code(s) for map:
	<ul style="list-style-type: none"> * Map is password-protected. P Map is in production status (locked); cannot be updated. U Map is currently being updated. C,A,P,R Language indicator: C (COBOL), A (Assembler), P (PL/1), R (RPG). E Map has extended attributes. H,T Map is header (H) or trailer (T) of page-built screen. F,L Map is justified as first or last on a page-built screen. V Map has input validation rules.
Date	Date the map was last updated. By default, the date is displayed in <i>mmdccyy</i> format; if European dates are used, the date is formatted as <i>ddmmccyy</i> .
Time	Time of day that map was last updated.
Description	Comments on member as entered in Map Definition Options screen.

The following fields are displayed when you press F2 for the alternate display.

OPID	Terminal or operator ID (operator ID used whenever present).
SIZE	Size of map in row/column format.
START	Starting position of map on screen in row/column format. Or, may be described using the codes: N (next row or column) or S (same row or column).
MAP	Length of the physical map.
DSECT	Length of the symbolic map.

You can use the function keys described in the following table to navigate the list.

Fkey	Description
F1	Display screen level help for this screen.
F2	View the alternate display for this screen. Press F2 to toggle between these two displays.
F3	Exit -- cancel this option and return to the Map Options menu.
F7	Scroll backward one screen (if applicable).
F8	Scroll forward one screen (if applicable).

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Input Rule Options](#)
- [Input Rule Editor](#)
- [Input Rule Syntax](#)
- [Reserved Words](#)
- [Organization](#)

Related topics:

- [Common Uses of Input Rules](#)
- [Create Input Rules](#)
- [Edit Input Rules](#)
- [Test Input Rules](#)
- [Implement Input Rules](#)
- [Using Input Rules with Conversational Programs](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Working with Input Rules

Input Rules are programming [commands](#) that evaluate and respond to user input. You write input rules for a map using a line editor provided by BMS/TS. After writing input rules, you tell BMS/TS to monitor user input by adding an entry to the Transaction Monitor. BMS/TS then responds to user entries based on your input rules.

You can use input rules to:

- Force the user to enter data in a field,
- Force the user to fill a field with data,
- Format dates,
- Format numeric fields,
- Initialize empty fields,
- Perform range checks,
- Force field values,
- Perform table look-ups,
- Return custom error messages.

For pseudo-conversational programs, BMS/TS evaluates and responds to user input before calling the application program. If necessary, the user must correct the input before BMS/TS passes it to the application program.

For conversational programs, you can create the same form of interaction by adding a simple "call" to the application. Place this "call" in front of your application program's "receive" command. As a result, your program interacts with the input rules in the same way a pseudo-conversational program would. See [Using Input Rules With Conversational Programs](#) for details on writing this "call."

Input Rule Options

You have the following options for using input rules with BMS/TS:

- **Design** — Select the map for which you want to create input rules, then write input rules using the [Input Rule Editor](#) screen. You can also test the functionality of the input rules.
- **Redesign** — Update the existing input rules for a map. You can select the map containing the input rules you want to edit, then revise the input rules using the [Input Rule Editor](#) screen. You can also test the functionality of the input rules.
- **Test** — Test the functionality of a map's input rules. You can select the map containing the input rules you want to test, then key data and watch the responses as they will appear to the end-user.

Input Rule Editor

You write input rules using the Input Rule Editor (view [sample](#)).

Note: The Desc field contains the description (if applicable) that you entered at the Map Definition Options screen when you designed the map.

The editor screen consists of a line command (LC) field and four entry fields, all in a columnar format. You create input rules by completing the entry fields with the aid of line commands. The table below describes each of these fields.

Field	Description												
LC	Line command. Entry field for single character line editing command (insert, delete, copy, etc.). Place the cursor in the LC field and press F1 for details on these commands. For a complete description of line commands, see Appendix A, Using the Line Editor .												
Cmd	Command. BMS/TS performs an action based on the command entered in this field. See Using Input Rule Commands for details.												
-A- Field	Usually a field name from your map. BMS/TS performs the action described in the Cmd field on this field. BMS/TS can also perform the action if a test of this field proves to be true. You describe the test using Cond and -B- Field or Constant fields.												
Cond	Condition. BMS/TS performs the action on the -A- Field so that the condition described here exists. BMS/TS can also perform the action if the condition described here exists. Valid codes are: <table><tbody><tr><td>LS or <</td><td>Less than</td></tr><tr><td>GT or ></td><td>Greater than</td></tr><tr><td>NE or <> or ≠</td><td>Not equal to</td></tr><tr><td>LE or <=</td><td>Less than or equal to</td></tr><tr><td>GE or >=</td><td>Greater than or equal to</td></tr><tr><td>EQ or =</td><td>Equal to</td></tr></tbody></table>	LS or <	Less than	GT or >	Greater than	NE or <> or ≠	Not equal to	LE or <=	Less than or equal to	GE or >=	Greater than or equal to	EQ or =	Equal to
LS or <	Less than												
GT or >	Greater than												
NE or <> or ≠	Not equal to												
LE or <=	Less than or equal to												
GE or >=	Greater than or equal to												
EQ or =	Equal to												
	For specific examples of these fields, see Input Rule Syntax (below).												
-B- Field or Constant	Either a constant value or another field from your map. The contents of this field tell BMS/TS how to modify the -A- Field. If desired, you can also compare the -A- Field to the contents of this field.												

Input Rule Syntax

The physical layout of the Input Rule Editor helps create the basic syntax of all input rules. Depending on the individual command, the syntax normally serves to perform one of two functions:

Cmd	-A- Field	Cond	-B- Field or Constant
<Do this>	<to this field>	<so that it is>	<this>
or...			
<Do this>	<to this field>	<if it is this>	<compared to this>

Here are some specific examples of basic syntax:

Cmd	-A- Field	Cond	-B- Field or Constant
LET	CUSTOMER-NUMBER	EQ	TELEPHONE NUMBER
REJ	CREDIT-AMOUNT	GT	10000

In the first example, the input rules tell BMS/TS to fill a field named Customer-Number with the data entered in the same map's Telephone-Number field.

In the second example, the input rules tell BMS/TS to reject the data entered in a map's Credit-Amount field if the value is greater than 10,000.

Reserved Words

With reserved words, you can have BMS/TS perform particular actions based on system information, such as the ID of the operator, the date, or the press of a function key by the user. Use these reserved words in conditional logic statements (IF, ELSE, ENDIF). Place the reserved words in the -A- Field. You can choose from the following reserved words:

Reserved Word	Description
*AID	Number of the 2-digit function key pressed by the user.
*CCYY	Current year in 4-digit format.
*CCYYDATE	Current date. By default, the format is <i>mmdccyy</i> ; if European dates are used, the format is <i>ddmmccyy</i> .
*DATE	Current system date in <i>mm/dd/yy</i> format.
*DAY	Current day of the month in <i>dd</i> format.
*ERROR	Special field used for holding and displaying custom error messages.
*FIRST	Number of validation passes for the current mapset and map name. The default is '1' (first pass).
*MONTH	Current month in <i>mm</i> format.
*OPID	3-character operator ID.
*Rnn-Cnn-Lnn	

Field on a map described by that field's attribute position (one position to the left of the actual data):

- *Rnn* – Row number of the field,
- *Cnn* – Column number of the field,
- *Lnn* – Length of the field (not including the attribute byte).

***TRAN** Current transaction code.

***YEAR** Current year in yy format.

Here is an example of the input rule syntax when using reserved words:

Cmd	-A- Field	Cond	-B- Field or Constant
IF	CUSTOMER-NUMBER	GT	9999
LET	*ERROR	EQ	'NUMBER TOO LARGE'
ENDIF			

In this example, the input rules tell BMS/TS to write a custom error message to the screen -- NUMBER TOO LARGE -- if the user enters a number in the Customer-Number field that is greater than 9,999.

Organization

The order of your input rules is important. BMS/TS reads and processes input rules for each field on a map from top to bottom, starting with the first field. Follow the guidelines listed below when organizing your input rules.

- Place data fields in a logical order.

The order of the fields validated in your input rules should follow the order in which the user will enter data. Ideally, the user can make corrections to invalid entries one field at a time in the same order that the fields appear on the map.

- Move data before you edit it.

Move data to a field (using the **LET** command) before you perform any edits on the field (using the **EDIT** command). This ensures that the proper edits are applied to the data.

- Edit and strip input data before you validate it.

Place **EDIT** and **STRIP** commands immediately after **LET** commands but before any other commands associated with a particular data field. This ensures that edits are applied to the proper data.

- Keep conditional logic to a minimum.

BMS/TS can support up to 31 levels of nested **IF**, **ELSE**, **ENDIF** commands. However, because you cannot indent statements in the **Input Rule Editor**, these levels may be hard to identify. For this reason, you should place heavily nested conditional logic in your

application for easier identification. You can then use the [CALL](#) command to pass control to your application program.

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Input Rule Commands](#)

Related topics:

- [Working with Input Rules](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Input Rule Commands

This section provides links to detailed information about the individual input rule commands you can use to write input rules.

Command	Description
ATTRIB	Set field attribute.
CALL	Call a command-level program.
CURS	Move the cursor to a particular field.
EDIT	Edit operator input.
FILL	Initialize empty fields.
IF, ELSE, ENDIF	Use conditional logic.
JUST	Force left/right justification.
LET	Move data to a field.
OPTION	Define the number of error messages displayed.
PAD	Pad the field with a trailing constant.
REJ	Reject field input on a condition.
RETN	Return control to the application program.
SEL	Select field input on a condition.
STRIP	Strip trailing characters.
TROFF	Turn off input rule trace.
TRON	Turn on input rule trace.

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [ATTRIB Syntax](#)
- [ATTRIB Parameters](#)
- [ATTRIB Example](#)

Related topics:

- [Working with Input Rules](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

ATTRIB Input Rule Command

The ATTRIB input rule command lets you automatically change field attributes (such as intensity) if the user enters invalid data.

Use the ATTRIB command, along with the [EDIT](#) or [REJ](#) input rule commands, to change the attributes of fields if the user enters invalid data. Keep in mind that ATTRIB has no effect if an error does not occur.

ATTRIB Syntax

Use the following syntax with the ATTRIB command:

Cmd	-A- Field	Cond	-B- Field or Constant
ATTRIB	Must be a map field, and must correspond with the field name you defined at the Field Definition screen when designing the map.	EQ or =	Positional parameter(s) describing how BMS/TS should change the field attributes when the user enters invalid data.

ATTRIB Parameters

ATTRIB supports three positional parameters. When using multiple parameters, separate them with commas.

A. The first parameter indicates the original attribute values of the field.

Parameter	Attribute
A	Unprotected input field, normal intensity.
I	Unprotected input field, bright intensity.
(Unprotected input field, dark intensity.
J	Unprotected numeric field, normal intensity.
R	Unprotected numeric field, bright intensity.
)	Unprotected numeric field, dark intensity.

B. The second parameter indicates what color you want to make the field when BMS/TS detects an

error.

Parameter	Attribute
B	Blue
G	Green
N	Neutral
P	Pink
R	Red
T	Turquoise
Y	Yellow

C. The third parameter indicates how you want to highlight the field when BMS/TS detects an error.

Parameter	Attribute
B	Blinking
R	Reverse Video
U	Underline

ATTRIB Examples

In the following examples, note that you can use single character [parameters](#) or you can spell out the parameters. Either method is correct.

Example: ATTRIB command using all three parameter

Cmd	-A- Field	Cond	-B- Field or Constant
REJ	CUSTOMER-NUMBER	GT	9999
ATTRIB	CUSTOMER-NUMBER	EQ	A,B,U

In this example, BMS/TS rejects any entries in the Customer-Number field that are greater than 9,999. If an error occurs, BMS/TS also underlines the Customer-Number field, and makes it blue in color, until the user enters a valid number.

Example: ATTRIB command using only third parameter position

Cmd	-A- Field	Cond	-B- Field or Constant
REJ	CUSTOMER-NUMBER	GT	9999
ATTRIB	CUSTOMER-NUMBER	EQ	.,UNDERLINE

In this example, BMS/TS rejects any entries in the Customer–Number field that are greater than 9,999. If an error occurs, BMS/TS underlines the Customer–Number field until the user enters a valid entry.

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [CALL Format](#)
- [CALL Syntax](#)
- [CALL Example](#)

Related topics:

- [Working with Input Rules](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

CALL Input Rule Command

Use the CALL command to run a command-level program when you want to perform edits or input validation routines that are not possible with BMS/TS.

After you define the fields you want to pass in the communications area of the linkage section of your command-level program, BMS/TS passes the contents of the -B- Field, the field name, and the field length to the command-level program. Your program must then pass a code back to BMS/TS telling it whether to display an error message. You can designate in your program the wording of the displayed error message.

CALL Format

Use the following format to define the fields you want to pass in the communications area within the linkage section of your application program:

```
01 EXAMPLE-COMMUNICATIONS-AREA.
   10 FILLER                PIC X.
   10 COM-RETURN            PIC X.
   10 COM-FIELD-NAME        PIC X(30).
   10 COM-FIELD-LENGTH      PIC XX.
   10 FILLER                PIC X(4).
   10 COM-ERROR-MSG         PIC X(80).
   10 COM-FIELD             PIC X(256).
```

Field	Description
FILLER	Reserved for BMS/TS.
COM-RETURN	User return code. Valid codes are: <ul style="list-style-type: none"> 0 (zero) Do not display any error message. S Display the default BMS/TS error message B605 Field contains invalid data. U Display error message defined in COM-ERROR-MSG.
COM-FIELD-NAME	Map field name.
COM-FIELD-LENGTH	Map field length.

FILLER	Reserved for BMS/TS.
COM-ERROR-MSG	User error message.
COM-FIELD	Field contents.

CALL Syntax

Use the following syntax with the CALL command:

Cmd	-A- Field	Cond	-B- Field or Constant
CALL	Must be an executable program name having a corresponding CSD entry.	<i>Not used.</i>	Must be a map field, and must correspond with the field name you defined at the Field Definition screen when designing your map.

CALL Example

Cmd	-A- Field	Cond	-B- Field or Constant
CALL	PRGMNAME		CUSTOMER-NUMBER

In the above example, the CALL command passes the contents and length of a field named Customer-Number to the application program prgname.

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [CURS Syntax](#)
- [CURS Example](#)

Related topics:

- [Working with Input Rules](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

CURS Input Rule Command

The CURS input rule command lets you move the cursor to the beginning of a named field and set the field attribute to bright intensity.

You then have the option of moving your own error message into the *ERROR reserved word, or into your own error message field.

CURS is usually used with [IF](#) and [ENDIF](#) conditional commands.

CURS Syntax

Use the following syntax with the CURS command:

Cmd	-A- Field	Cond	-B- Field or Constant
CURS	Must be a map field, and must correspond with the field name you defined at the Field Definition screen when designing your map.	<i>Not used.</i>	<i>Not used.</i>

CURS Example

Cmd	-A- Field	Cond	-B- Field or Constant
IF	CUSTOMER-NUMBER	GT	9999
CURS	CUSTOMER-NUMBER		
LET	*ERROR	EQ	'INVALID CUSTOMER NUMBER'
ENDIF			

In this example, BMS/TS rejects any entries in the Customer-Number field that are greater than 9,999. BMS/TS then returns the cursor to the Customer-Number field, and displays the custom error message "INVALID CUSTOMER NUMBER".

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Contents:

- [EDIT Syntax](#)
- [EDIT Pattern](#)
- [EDIT Reserved Words](#)
- [EDIT Example](#)

Related topics:

- [Working with Input Rules](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

EDIT Input Rule Command

The EDIT input rule command lets you review user entries.

Use the EDIT command with an edit pattern to read an entry by the user and, if necessary, display the correct format. This capability lets BMS/TS reject all improperly entered numeric fields.

EDIT Syntax

Use the following syntax with the EDIT command:

Cmd	-A- Field	Cond	-B- Field or Constant
EDIT	Must be a map field that corresponds with the field name defined at the Field Definition screen when designing the map.	<i>Not used.</i>	Edit pattern or reserved word.

Edit Pattern

Write the edit pattern using the following three characters:

Character	Function
S	Enables the user to enter a negative number, using a minus sign (-) in front of the number. However, the number can be either positive or negative. Note: You must place the S first in the edit pattern if you want to allow entry of negative numbers.
V	Indicates the position of an implied decimal point.
9	Identifies a numeric digit.

Note: You can use parenthetical expressions in the edit pattern. For example, S9(4)V99 is the same as S9999V99.

EDIT Reserved Words

You can use the EDIT command with the following reserved words:

Reserved Word	Function
*MUSTENTER	Ensures that the user enters data in a specific field.
*MUSTFILL	Ensures that the user enters a character for each position of the specified field.
*DATE	Reads a date entered by the user and, if necessary, displays the correct date format. You can abbreviate the *DATE reserved word as *D.

*DATE supports any combination of the following parameters:

MM	Month
DD	Day of the month
YY	2–digit year
CCYY	4–digit year
YYYY	4–digit year
/	Date divider
– (hyphen)	Date divider
(space)	Date divider

Note: Separate *DATE from the parameters with a colon. For example: *D:MM/DD/CCYY.

EDIT Example

Cmd	–A– Field	Cond	–B– Field or Constant
EDIT	CUSTOMER–NUMBER		*MUSTENTER
EDIT	CUSTOMER–NUMBER		*MUSTFILL
EDIT	CUSTOMER–NUMBER		9999
EDIT	PAYMENT–DATE		*D:MM/DD/YYYY
EDIT	PAYMENT–AMOUNT		S9(4)V99

In this example BMS/TS forces the user to enter a 4–digit number in the Customer–Number field. BMS/TS also forces the user to enter the date in the Payment–Date field in *mm/dd/yyyy* format (with slashes as date dividers). Lastly, BMS/TS forces the user to enter an amount in the Payment–Amount field according to the following rules:

- Entry can be positive or negative (through edit pattern 'S').
- Entry must be numeric and 1 – 4 digits (through edit pattern '9(4)').
- Format input for two decimal positions, regardless of whether user enters decimal point (through edit pattern 'V').

Keep in mind that when formatting numeric fields, BMS/TS recognizes a trailing space as a field delimiter and therefore automatically removes any trailing spaces (and any other subsequent characters) from the user entry. For example, BMS/TS sees a user entry of 1599space95 as 1599.

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [FILL Syntax](#)
- [FILL Example](#)

Related topics:

- [Working with Input Rules](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

FILL Input Rule Command

The FILL command lets you place a single character or numeric digit in a field not entered by the user.

Keep in mind that the FILL command has no effect if the user enters data in the field.

FILL Syntax

Use the following syntax with the FILL command:

Cmd	-A- Field	Cond	-B- Field or Constant
FILL	Must be a map field, and must correspond with the field name you defined at the Field Definition screen when designing the map.	EQ or =	A single character or numeric digit. You must enclose character literals in single quotes (for example, A = 'A').

FILL Example

Cmd	-A- Field	Cond	-B- Field or Constant
FILL	CUSTOMER-NUMBER	EQ	0

In the above example, BMS/TS places a zero (0) in a field named Customer-Number if a user leaves this field blank.

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [IF,ELSE,ENDIF Syntax](#)
- [IF,ELSE,ENDIF Examples](#)

Related topics:

- [Working with Input Rules](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

IF, ELSE, ENDIF Input Rule Commands

The IF, ELSE, ENDIF commands let you add conditional logic to your input rules.

These commands allow you to evaluate a user's entry in a field, or to test a reserved word. By also using the [LET](#), [REJ](#), [CURS](#), [ATTRIB](#), and other commands, BMS/TS can respond to the conditions you establish.

Note: When using the [EDIT](#) command to compare entries in numeric fields, you must place the EDIT command before the IF logic.

BMS/TS executes all input rule commands following a true "IF" condition until finding an "ELSE" or "ENDIF" command. BMS/TS executes the commands after the "ELSE" command only when the "IF" condition is false. Because the "ENDIF" command terminates the "IF" condition, you must have an ENDIF for every IF.

BMS/TS can support up to 31 levels of nested IF, ELSE, ENDIF commands. However, because you cannot indent statements in the Input Rule Editor, these levels may be hard to identify. You should therefore place heavily nested conditional logic in your application for easier identification. You can then use the CALL command to pass control to your application program.

IF, ELSE, ENDIF Syntax

Use the following syntax with the IF, ELSE, ENDIF commands:

Cmd	-A- Field	Cond	-B- Field or Constant
IF	Must be a map field* or reserved word.	< or LS > or GT <> or NE <= or LE >= or GE = or EQ	Any valid map field, reserved word, or constant value.
ELSE, ENDIF			

Note: A map field must correspond with the field name you defined at the Field Definition screen when you designed the map.

IF, ELSE, ENDIF Examples

Example: Conditional logic using IF, ELSE, and ENDIF

Cmd	-A- Field	Cond	-B- Field or Constant
IF	CUSTOMER-NUMBER	LS	9999
REJ	CREDIT-AMOUNT	GT	100.00
ELSE			
REJ	CREDIT-AMOUNT	GT	200.00
ENDIF			

In this example, BMS/TS first checks the user's entry in a field named Customer-Number. If the field entry is less than 9,999, BMS/TS rejects all entries in a field named Credit-Amount that exceed \$100.00. If the entry in the Customer-Number field is greater than 9,999, BMS/TS rejects all entries in the Credit-Amount field that exceed \$200.00.

Example: Conditional logic using only IF and ENDIF

Cmd	-A- Field	Cond	-B- Field or Constant
IF	CUSTOMER-NUMBER	GT	9999
LET	*ERROR	EQ	'INVALID CUSTOMER NUMBER'
ENDIF			

In the above example, BMS/TS rejects any entries in the Customer-Number field that are greater than 9,999. BMS/TS then displays the custom error message "INVALID CUSTOMER NUMBER".

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [JUST Syntax](#)
- [JUST Example](#)

Related topics:

- [Working with Input Rules](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

JUST Input Rule Command

The JUST command lets you force a field entry to left or right justification.

Use the JUST command to reformat entered data, making it either left or right justified on screen.

The JUST command works like the CICS BMS JUST function except that it also removes trailing spaces. That is, BMS/TS recognizes a trailing space as a field delimiter and therefore automatically removes any trailing spaces (and any other subsequent characters) from the user entry.

For example, BMS/TS sees a user entry of 1599[space]95 as 1599.

JUST Syntax

Use the following syntax with the JUST command:

Cmd	-A- Field	Cond	-B- Field or Constant
JUST	Must be a map field, and must correspond with the field name you defined at the Field Definition screen when designing the map.	<i>Not used.</i>	*RIGHT or *R or *LEFT or *L

JUST Example

Cmd	-A- Field	Cond	-B- Field or Constant
JUST	CUSTOMER-NUMBER		*RIGHT

In this example, BMS/TS forces the user's entry in the Customer-Number field to be right-justified.

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Contents:

- [LET Syntax](#)
- [LET Example](#)

Related topics:

- [Working with Input Rules](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

LET Input Rule Command

The LET command can place data in a field.

Use the LET command (often with conditional logic) to move the value or contents of the –B– Field or Constant to the –A– Field. This lets you move data to a field for display purposes, or to force a constant or value into a field. Keep in mind, though, that when moving the value of one field to another along with the EDIT command, you should place the LET command before the EDIT command. This ensures that BMS/TS edits the proper field.

Note: You should use this routine only when there are a few possible values for the –A– Field. If the –A– Field can contain numerous values, you should [create a table](#) and write a table look-up routine.

LET Syntax

Use the following syntax with the LET command:

Cmd	–A– Field	Cond	–B– Field or Constant
LET	Must be a map field or the reserved word *ERROR.	EQ	Any valid map field, reserved word, or constant value.

The map field must correspond with the field name you defined on the Field Definition screen when you designed the map.

LET Example

Cmd	–A– Field	Cond	–B– Field or Constant
IF	CUSTOMER–NUMBER	GT	9999
LET	*ERROR	EQ	'INVALID CUSTOMER NUMBER'
ENDIF			

In this example, BMS/TS rejects any entries in the Customer–Number field that are greater than 9,999. BMS/TS then displays the error message "INVALID CUSTOMER NUMBER".

[Top of page](#)

Contents:

- [OPTION Syntax](#)
- [OPTION Parameters](#)
- [OPTION Example](#)

Related topics:

- [Working with Input Rules](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

OPTION Input Rule Command

The OPTION command lets you define the number of error messages BMS/TS displays at one time. You can also use the OPTION command to define the attributes of the error message(s).

Use the OPTION command with the reserved word *ERROR to specify the number of error messages displayed simultaneously. This allows you to stack multiple BMS/TS error messages on top of each other at the bottom of the screen.

OPTION Syntax

Use the following syntax with the OPTION command:

Cmd	-A- Field	Cond	-B- Field or Constant
OPTION	Must be a map field, and must correspond with the field name you defined at the Field Definition screen when designing the map.	EQ or =	Positional parameter(s) describing how BMS/TS should display the error message.

OPTION Parameters

OPTION supports three positional parameters. When using multiple parameters, separate them with commas.

- A. The first parameter indicates the desired number of lines of error messages.
- B. The second parameter indicates what color you want to make the error message.

Parameter	Attribute
B	Blue
G	Green
N	Neutral
P	Pink
R	Red

T	Turquoise
Y	Yellow

C. The third parameter indicates how you want to highlight the error message.

Parameter	Attribute
B	Blinking
R	Reverse Video
U	Underline

OPTION Example

Cmd	-A- Field	Cond	-B- Field or Constant
OPTION	*ERROR	EQ	5,R,R

This example defines 5 lines for display of error messages, and the messages display in red and reverse video.

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [PAD Syntax](#)
- [PAD Example](#)

Related topics:

- [Working with Input Rules](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

PAD Input Rule Command

The PAD command lets you fill a field with any trailing character you desire.

Use the PAD command to fill the spaces following the input data with the character you designate.

PAD Syntax

Use the following syntax with the PAD command:

Cmd	-A- Field	Cond	-B- Field or Constant
PAD	Must be a map field, and must correspond with the field name you defined at the Field Definition screen when designing the map.	EQ or =	Single character literal or numeric digit. You must enclose character literals in single quotes (for example, A = 'A').

PAD Example

Cmd	-A- Field	Cond	-B- Field or Constant
PAD	CUSTOMER-NUMBER	EQ	0

In this example, BMS/TS adds trailing zeros (0) to the entry in the Customer-Number field. For example, if the length of the Customer-Number field were set to 10 characters and the user entered 99 in the field, BMS/TS would pad the field with 8 zeros: 9900000000.

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [REJ Syntax](#)
- [REJ Example](#)

Related topics:

- [Working with Input Rules](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

REJ Input Rule Command

The REJ command lets you use a condition to reject a user entry in a field.

Use the REJ command to evaluate user entry in a field and subsequently reject invalid entries. The REJ command rejects the field entry if the condition specified is true. You can create an unlimited number of REJ commands for a field.

Note: REJ ignores fields that do not contain data. Use the [EDIT](#) command with the reserved word *MUSTENTER to force the user to enter data in a field. The REJ command will then evaluate the field entry.

REJ Syntax

Use the following syntax with the REJ command:

Cmd	-A- Field	Cond	-B- Field or Constant
REJ	Must be a map field and must correspond with the field name you defined at the Field Definition screen when you designed the map.	LS or < GT or > NE or <> LE or <= GE or >= EQ or =	Any valid map field, reserved word, or constant value.

REJ Example

Cmd	-A- Field	Cond	-B- Field or Constant
EDIT	CUSTOMER-NUMBER	EQ	*MUSTENTER
REJ	CUSTOMER-NUMBER	GT	9999

In this example, BMS/TS first forces the user to enter data in the Customer-Number field. BMS/TS then rejects all entries that are greater than 9,999.

[Top of page](#)

Contents:

- [RETN Syntax](#)
- [RETN Example](#)

Related topics:

- [Working with Input Rules](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

RETN Input Rule Command

The RETN command lets you return control from the input rules to your application program.

Use the RETN command with conditional logic to pass control from the input rules to your application program. Keep in mind that the RETN command passes control to the application program regardless of whether BMS/TS detects any errors through other input rule commands. In other words, RETN bypasses input validation.

Note: Place your RETN command at the top of your input rules to keep BMS/TS from processing data fields unnecessarily.

RETN Syntax

Use the following syntax with the RETN command:

Cmd	-A- Field	Cond	-B- Field or Constant
RETN	<i>Not used.</i>	<i>Not used.</i>	<i>Not used.</i>

RETN Example

Cmd	-A- Field	Cond	-B- Field or Constant
IF	*AID	EQ	'11'
RETN			
ENDIF			
REJ	CUSTOMER-NUMBER	GT	9999

In this example, BMS/TS returns control to your application program if the user presses F11. If the user does not press F11, BMS/TS checks the Customer-Number field and rejects all entries that are greater than 9,999.

[Top of page](#)

[User's Guide Home](#)

Contents:

- [SEL Syntax](#)
- [SEL Example](#)

Related topics:

- [Working with Input Rules](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

SEL Input Rule Command

The SEL command lets you evaluate an entry in a field and accept the entry if it is valid.

Use the SEL command to evaluate a field entry using a condition, then accept the entry if the condition proves to be true. You can create an unlimited number of SEL commands for a field.

You usually use the SEL command in combination with [REJ](#) to further define valid field entries. When using both SEL and REJ commands, remember to place your SEL commands before your REJ commands to ensure that BMS/TS does not reject valid entries.

Note: SEL ignores fields that do not contain data. Use the [EDIT](#) command with the reserved word *MUSTENTER to force the user to enter data in a field. The SEL command will then evaluate the field entry. Also, remember to place the EDIT command before the SEL command for the same field.

SEL Syntax

Use the following syntax with the SEL command:

Cmd	-A- Field	Cond	-B- Field or Constant
SEL	Must be a map field and must correspond with the field name you defined at the Field Definition screen when you designed the map.	LS or < GT or > NE or <> LE or <= GE or >= EQ or =	Any valid map field, reserved word, or constant value.

SEL Example

Cmd	-A- Field	Cond	-B- Field or Constant
EDIT	CUSTOMER-NUMBER	EQ	*MUSTENTER
SEL	CUSTOMER-NUMBER	EQ	12345
REJ	CUSTOMER-NUMBER	GT	9999

In this example, BMS/TS first forces the user to enter data in the Customer-Number field. BMS/TS then reviews the field and rejects all entries that are greater than 9,999. However, the SEL command lets

BMS/TS accept a Customer–Number entry of 12,345, even though it is greater than 9,999.

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [STRIP Syntax](#)
- [STRIP Example](#)

Related topics:

- [Working with Input Rules](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

STRIP Input Rule Command

The STRIP command lets you remove trailing characters used to initialize an entry field.

For example, if you preformat entry fields with an underscore (or other character), you can use STRIP to take the underscores out before sending the data to your application program. This keeps the user from having to press ErEOF before completing a preformatted field. You can also use STRIP to remove any trailing spaces.

You can create an unlimited number of STRIP commands for a field.

STRIP Syntax

Use the following syntax with the STRIP command:

Cmd	-A- Field	Cond	-B- Field or Constant
STRIP	Must be a map field, and must correspond with the field name you defined at the Field Definition screen when designing the map.	EQ or =	Single character literal or numeric digit. You must enclose character literals in single quotes.

STRIP Example

Cmd	-A- Field	Cond	-B- Field or Constant
STRIP	CUSTOMER-NUMBER	EQ	'_'
STRIP	CUSTOMER-NUMBER	EQ	' '

In this example, BMS/TS removes all trailing underscores (_), as well as trailing spaces, from the Customer-Number field after the user completes the field and presses Enter.

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Contents:

- [TROFF Syntax](#)
- [TROFF Example](#)

Related topics:

- [Working with Input Rules](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

TROFF Input Rule Command

The TROFF command lets you turn off the BMS/TS input rule trace tool.

Use the TROFF command to terminate the trace debugging tool at a certain point in your input rules.

Note: See the [TRON](#) command for more information on the trace debugging tool.

TROFF Syntax

Use the following syntax with the TROFF command:

Cmd	-A- Field	Cond	-B- Field or Constant
TROFF	<i>Not used.</i>	<i>Not used.</i>	<i>Not used.</i>

TROFF Example

You could use the TRON and TROFF commands to help find a problem with nested IF commands:

Cmd	-A- Field	Cond	-B- Field or Constant
TRON			
IF	MONTHS-ACCOUNT-OPEN	LS	12
IF	OUTSTANDING-BALANCE	GT	1000
IF	PURCHASE AMOUNT	GT	2500
REJ	CASH-DOWN	EQ	0
ENDIF			
ELSE			
REJ	CASH-DOWN	LS	2500
ENDIF			
ELSE			
REJ	CASH-DOWN	LS	1250

ENDIF
TROFF

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [TRON Syntax](#)
- [TRON Example](#)

Related topics:

- [Working with Input Rules](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

TRON Input Rule Command

The TRON command lets you turn on the BMS/TS input rule trace tool.

Use the TRON command to initiate the trace debugging tool at a certain point in your input rules. With input rule trace turned on, BMS/TS performs the following functions:

- From the point of the TRON command forward, BMS/TS begins stepping through your input rules one line at a time, beginning at the top and reading down.
- BMS/TS displays the result of each of your input rules.
- BMS/TS continues this action until it reads a TROFF command.

Note: Place the TRON and [TROFF](#) commands at different places throughout your input rules to locate problems in troubled sections.

TRON Syntax

Use the following syntax with the TRON command:

Cmd	-A- Field	Cond	-B- Field or Constant
TRON	<i>Not used.</i>	<i>Not used.</i>	<i>Not used.</i>

TRON Example

You could use the TRON and TROFF commands to help find a problem with nested IF commands:

Cmd	-A- Field	Cond	-B- Field or Constant
TRON			
IF	MONTHS-ACCOUNT-OPEN	LS	12
IF	OUTSTANDING-BALANCE	GT	1000
IF	PURCHASE AMOUNT	GT	2500
REJ	CASH-DOWN	EQ	0
ENDIF			

```
ELSE
REJ    CASH-DOWN           LS    2500
ENDIF
ELSE
REJ    CASH-DOWN           LS    1250
ENDIF
TROFF
```

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- Common Uses for Input Rules

Related topics:

- [Working with Input Rules](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Common Uses for Input Rules

Theoretically, you can use the input rule commands to write an infinite number of different input rules. Chances are, however, that you will use several input validation routines regularly.

You are most likely to use the input rule commands (either alone or in combination) to:

- Force entry of data in a field,
- Force complete entry of a field,
- Format dates,
- Format numeric fields,
- Initialize empty fields,
- Perform range checks,
- Force field values,
- Perform table look-ups,
- Return custom error messages.

Force Entry of Data in a Field

Forcing entry of data in a field ensures that the user enters data in the specified field.

Use the [EDIT](#) command with the reserved word *MUSTENTER to define input rules to force entry of data in a field. This ensures that the user enters data in a field before BMS/TS calls the application program. For example, to force the user to enter data in the Customer-Number field, you would use the following data:

Cmd	-A- Field	Cond	-B- Field or Constant
EDIT	CUSTOMER-NUMBER		*MUSTENTER

The field name you enter in the -A- Field must correspond with the field name you defined at the Field Definition screen when you originally designed the map.

Force Complete Entry of a Field

Forcing complete entry of a field ensures that the user enters a character for each position of the specified field.

Use the [EDIT](#) input rule command, along with the reserved words *MUSTENTER and *MUSTFILL, to force the user to completely fill the specified field with data (according to the field length defined at the Field Definition screen when you designed your map).

Keep in mind that *MUSTFILL alone does not require a field to be input: it simply requires the user to fill the field if the user actually enters data. Using *MUSTENTER and *MUSTFILL together ensures that the user enters the assigned number of characters in a field before BMS/TS calls the application program.

For example, to force the user to place a character in each position of the Customer–Number field, you would use the following data:

Cmd	–A– Field	Cond	–B– Field or Constant
EDIT	CUSTOMER–NUMBER		*MUSTENTER
EDIT	CUSTOMER–NUMBER		*MUSTFILL

The field name you enter in the –A– Field must correspond with the field name you defined at the Field Definition screen when you originally designed the map.

Format Dates

Formatting dates lets you review all user–entered dates to ensure that they are in the correct format for your application.

Use the [EDIT](#) command with the reserved word *DATE (or *D) to read a date entered by the user and, if necessary, display the correct date format. This capability lets BMS/TS reject all improperly entered dates.

The reserved word *DATE supports any combination of the following parameters:

Parameter	Attribute
MM	Month
DD	Day of the month
DDD	Day of the year (Julian date)
YY	2–digit year
CCYY	4–digit year
YYYY	4–digit year
/	Date divider
–	Date divider
(space)	Date divider

Use the EDIT command, with the reserved word *DATE, to define input rules that will review an entry in a date field. For example, to force the user to enter the date in *mm/dd/yy* format (with slashes as date dividers), you would use the following data:

Cmd	–A– Field	Cond	–B– Field or Constant
EDIT	DATE		*D:MM/DD/YY

To force the user to enter the date in *mm/dd/ccyy* format with hyphens as date dividers), you would use the following data:

Cmd	-A- Field	Cond	-B- Field or Constant
EDIT	DATE		*D:MM-DD-CCYY

To force the user to enter a Julian (YYDDD) date:

Cmd	-A- Field	Cond	-B- Field or Constant
EDIT	DATE		*D:YYDDD

The field name you enter in the -A- Field must correspond with the field name you defined at the Field Definition screen when you originally designed the map.

Format Numeric Fields

Formatting numeric fields lets you review all user-entered numeric data to ensure that it is in the correct format for your application.

Use the [EDIT](#) command with an edit pattern to read a numeric entry by the user and, if necessary, display the correct format. This capability lets BMS/TS reject all improperly entered numeric fields.

Write the edit pattern using the following three characters:

Character	Function
S	Allows the user to enter a negative number, using a minus sign (-) in front of the number. However, the number can be either positive or negative. Note: You must place the S first in the edit pattern if you want to allow entry of negative numbers.
V	Indicates the position of an implied decimal point.
9	Identifies a numeric digit.

Note: You can use parenthetical expressions in your edit pattern. For example, S9(4)V99 is the same as S9999V99.

Also, keep in mind that when formatting numeric fields, BMS/TS recognizes a trailing space as a field delimiter and therefore automatically removes any trailing spaces (and any other subsequent characters) from the user entry. For example, BMS/TS sees a user entry of 1599[space]95 as 1599.

Use the EDIT command, with an edit pattern, to define input rules that will review a numeric entry in a data field. For example, to force the user to enter numerics into a field named Customer-Number, you could use the following data:

Cmd	-A- Field	Cond	-B- Field or Constant
EDIT	CUSTOMER-NUMBER		S9(6)V99

In this example, the input rules ensure the following:

- Entry can be positive or negative (through edit pattern 'S').
- Entry must be numeric and 1 – 6 digits (through edit pattern '9(6)').
- Format input for two decimal positions, regardless of whether user enters decimal point (through edit pattern 'V').
- Remove any trailing spaces (by default).

Format Numeric Field Entries

BMS/TS can accept a variety of user entries when formatting numeric fields. For example, if the edit pattern were S999999V99, the user could enter any of the following numbers and BMS/TS would still see the entry as 00000300:

3
3.00
\$3.00
+3

BMS/TS also accepts percentages in one of two ways. Both of the following have the same value:

.15
15%

Initialize Empty Fields

Initializing empty fields lets you place data in a field which the user has left empty.

Use the [FILL](#) command to fill a field not entered by the user with a single character or numeric digit. Keep in mind that the FILL command does nothing if the user enters data in the field.

The FILL command will place a single character or numeric digit in a field not entered by the user. For example, to place a zero (0) in a field named Customer–Number if a user leaves this field blank, you would use the following data:

Cmd	–A– Field	Cond	–B– Field or Constant
FILL	CUSTOMER–NUMBER	EQ	0

Perform Range Checks

Performing range checks lets you ensure that a user entry falls within a specified range.

- Use a single REJ command to reject a user entry that falls outside of a specified range.
- Use two REJ commands for the same field if your range has both an upper and lower limit.

For example, to reject a user's entry in a field named Customer–Number if the entered data is greater than 9,999, you would use the following data:

Cmd	–A– Field	Cond	–B– Field or Constant
-----	-----------	------	-----------------------

REJ CUSTOMER-NUMBER GT 9999

To reject a user's entry in a field named Customer-Number if the entered data is greater than 9,999 or less than 1,000, you would use the following data:

Cmd	-A- Field	Cond	-B- Field or Constant
REJ	CUSTOMER-NUMBER	LS	1000
REJ	CUSTOMER-NUMBER	GT	9999

Force Field Values

Forcing field values lets you determine the value of one field based on the value of another map field.

Use conditional logic and the LET command to read an input field and then place the appropriate value in another field (determine the value of one field based on the value of another field).

Note: You should use this routine only when there are a few possible values for the other field. If the other field can contain numerous values, you should create a table and write a table look-up routine.

For example, to determine the value of a field named Division based on the value of a field named Zip-Code, you could use the following data:

Cmd	-A- Field	Cond	-B- Field or Constant
IF	ZIP-CODE	LS	25000
LET	DIVISION	EQ	'1'
ELSE			
IF	ZIP-CODE	GE	25000
LET	DIVISION	EQ	'2'
ENDIF			
ENDIF			

In this example, BMS/TS fills the Division field with 1 if the Zip-Code is less than 25,000. If the Zip-Code is equal to or greater than 25,000, BMS/TS fills the Division field with 2.

Perform Table Lookups

Table lookups ensure that the user enters a valid value in a field. You can perform two types of table lookups:

- Use the [REJ](#) command to first compare a field entry to a table of valid entries for that field, then reject an entry that does not match an entry in the table.

- Use conditional logic with the **LET** and **REJ** commands to first compare a field entry to a table of valid entries for that field, then return the corresponding object entry from the table.

For example, to validate an entry in a field named Customer–Number, you could enter the following data:

Cmd	–A– Field	Cond	–B– Field or Constant
REJ	CUSTOMER–NUMBER	NE	*TABLE ACODES

To validate an entry in a field named State–Abbreviation and then use the corresponding data from the table to fill the State–Name field, you could enter the following data:

Cmd	–A– Field	Cond	–B– Field or Constant
IF	STATE–ABBREVIATION	EQ	*TABLE STATES
LET	STATE–NAME	EQ	*TABLE STATES
ELSE			
REJ	STATE–ABBREVIATION	NE	*TABLE STATES
ENDIF			

In this example, BMS/TS first validates the user's entry in the State–Abbreviation field using a table of valid entries. If the entry is valid, BMS/TS then uses information from the same table to complete the State–Name field on the map. If the entry is invalid, BMS/TS simply rejects the user's entry.

Return Custom Error Messages

Returning custom error messages lets you create and display your own customized error messages.

Use conditional logic, the **LET** command, and the reserved word ***ERROR** to display a specific message in the event that the user makes an input error.

Move data to and from the ***ERROR** field using the **LET** command. Then, by imbedding the **LET** command in conditional logic, you can automatically move the contents of ***ERROR** to the screen if the condition tested in the logic is true.

For example, to display the message "INVALID CUSTOMER NUMBER" if the user enters a number in the Customer–Number field that does not appear in the table CNCODES, you could enter the following data:

Cmd	–A– Field	Cond	–B– Field or Constant
IF	CUSTOMER–NUMBER	NE	*TABLE CNCODES
LET	*ERROR	EQ	'INVALID CUSTOMER NUMBER'
LET	*R25–C01–L79	EQ	'RE–ENTER NUMBER OR CANCEL'
REJ	CUSTOMER NUMBER	NE	*TABLE CNCODES

ENDIF

Notice in the above sample that after displaying the initial error message, the input rules use the LET command and field coordinates to display REENTER NUMBER OR CANCEL beginning on row 25, column 1 of the map.

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Select the map](#)
- [Define the input rule\(s\)](#)
- [Test the input rule\(s\)](#)

Related topics:

- [Working with Input Rules](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Create Input Rules

You can create (define) input rules for an existing map. You will:

- [Select the map,](#)
- [Define the input rule\(s\),](#)
- [Test the input rule\(s\).](#)

Select a Map

Perform the following steps to select a map to which you want to add input rules.

1. From the Main Menu, select the Maps option and press Enter to display the Maps Menu (view [sample](#)).
 2. From the Maps Menu, select the Rules option and press Enter to display the Rules Options Menu (view [sample](#)).
 3. Select the Design option and press Enter to display the Select a Map screen (view [sample](#)).
 4. Type the name of the mapset in the Mapset field. To display a list of mapsets in the current BMS/TS library, press F2.
 5. Type the name of the map in the Map Name field. To display a list of maps in the current BMS/TS library, press F2.
 6. If you password-protected the map, type the password in the Password field. Otherwise, skip this field.
 7. Press Enter. BMS/TS displays the Input Rule Editor screen (view [sample](#)).
-

Define Input Rules

After you select a map, the Input Rule Editor screen appears (view [sample](#)). Notice that the cursor is in the Cmd column on the first entry line.

1. Enter information into the Cmd, –A– Field, Cond, and –B– Field or Constant fields. Press Tab to move from field to field.

Note: See [Using Input Rule Commands](#) for a list of valid input rule commands.

Hint: You can enter a question mark (?) in –A– Field or –B– Field or Constant if you want BMS/TS to display the field names on your map. This is especially useful if you forget the name of a field when defining input rules.

2. Repeat step 1 for all fields on the map for which you want to define input rules.

Note: BMS/TS initially provides 17 lines for input rule entries. If you fill all 17 lines with data, press Enter to insert an additional line. Then, as you fill each new line with data and press Enter, BMS/TS will insert an additional line. This allows you to define an unlimited number of input rules.

3. After defining all input rules for this map, press F3. BMS/TS saves the input rules and displays the Map Verification screen.
-

Test Input Rules

Testing input rules lets you display the map as it will appear in your application. You can then type in sample data to test your input rules.

After you complete the Input Rule Editor screen, the Map Verification screen appears (view [sample](#)).

Note: When present, protected (output) fields appear with a V at every fifth byte, a number at every tenth byte and four asterisks between, for the length of your defined field: (****V****1****V****2 etc.).

1. Enter sample data into the input fields displayed on the screen.
2. After keying sample data into all desired fields, press Enter. If your entries are valid, BMS/TS displays the sample data as it would appear when your map is being used. If a specific field entry is invalid, BMS/TS displays an error message based on the input rule for that field.
3. Repeat steps 1 and 2 to test the input rules for all desired fields.

Note: You can press F4 to return to the Input Rule Editor screen to modify or add input rules.

4. Press Clear. The Select a Map screen redisplay. You can repeat the process to define input rules for another map, or press F3 to exit.
-

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Select the map](#)
- [Edit the input rule\(s\)](#)
- [Test the input rule\(s\)](#)

Related topics:

- [Working with Input Rules](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Edit Input Rules

You can edit the input rules that you previously defined for a map. You will:

- [Select the map](#),
- [Edit the input rule\(s\)](#),
- [Test the input rule\(s\)](#).

Select a Map

Perform the following steps to select a map containing the input rules you want to edit.

1. From the Main Menu, select the Maps option and press Enter to display the Maps Menu (view [sample](#)).
 2. From the Maps Menu, select the Rules option and press Enter to display the Rules Options Menu (view [sample](#)).
 3. Select the Redesign option and press Enter to display the Select a Map screen (view [sample](#)).
 4. Type the name of the mapset in the Mapset field. To display a list of mapsets in the current BMS/TS library, press F2.
 5. Type the name of the map in the Map Name field.
Note: To display a list of maps in the current BMS/TS library, press F2.
 6. If you password-protected the map, type the password in the Password field. Otherwise skip this field.
 7. Press Enter. BMS/TS displays the Input Rule Editor screen (view [sample](#)).
-

Edit Input Rules

After you select a map, the Input Rule Editor screen appears (view [sample](#)).

1. Press Tab to move the cursor to the first field you want to edit.

2. Enter information into the field to revise the existing input rules.

Note: See [Using Input Rule Commands](#) for a list of valid input rule commands.

Hint: Enter a question mark (?) in –A– Field or –B– Field or Constant if you want BMS/TS to display the field names on your map. This is especially useful if you forget the name of a field when editing input rules.

3. Press Tab to move to the next field you want to edit.
4. Repeat steps 1 through 3 for all fields on the map for which you want to edit existing input rules or add new input rules.

Note: BMS/TS initially provides 17 lines for your input rule entries. If you fill all 17 lines with data, press Enter. BMS/TS inserts an additional line. Then, each time you fill a line with data and press Enter, BMS/TS will insert an additional line. This allows you to enter an unlimited number of input rules.

5. After you have edited the input rules for this map, press F3. BMS/TS saves the input rules and displays the Map Verification screen.

Test Input Rules

Testing input rules lets you display the map as it will appear in your application. You can then type in sample data to test your new or revised input rules.

After you complete the Input Rule Editor screen, the Map Verification screen appears (view [sample](#)).

Note: When present, protected (output) fields appear with a V at every fifth byte, a number at every tenth byte and four asterisks between, for the length of your defined field: (****V****1****V****2 etc.).

1. Enter sample data into the input fields displayed on the screen.
2. After keying sample data into all desired fields, press Enter. If your entries are valid, BMS/TS displays the sample data as it would appear when your map is being used. If a specific field entry is invalid, BMS/TS displays an error message based on the input rule for that field.

For example, if you entered 999 in the Customer Number field, BMS/TS would display the message "Value is not in range of field: GT 100".

3. Repeat steps 1 and 2 to test the input rules for all desired fields.

Note: You can press F4 to return to the Input Rule Editor screen to modify or add input rules.

4. Press Clear. The Select a Map screen redisplay. You can repeat the process to define input rules for another map, or press F3 to exit.

[Top of page](#)

Contents:

- [Select a Map](#)
- [Test Input Rules](#)

Related topics:

- [Working with Input Rules](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Test Input Rules

You can test the functionality of input rules without actually running the application.

To test input rules, you display a single map and enter data into the map's fields. BMS/TS reads and validates your entries according to your input rules. Use this option to test or demonstrate the operation of your map.

Select a Map

1. From the Main Menu, select the Maps option and press Enter.
2. From the Maps Menu, select the Rules option and press Enter.
3. Select the Test option and press Enter. The Select a Map to Display screen appears (view [sample](#)).
4. Type the name of the mapset in the Map set field. To list mapsets in the current BMS/TS library, press F2.
5. Type the name of the map in the Map name field.

Note: To display a list of maps in the current BMS/TS library, press F2.

6. If you password-protected the map, type the password in the Password field. Otherwise, skip this field.
7. Press Enter. BMS/TS displays the Input Rule Editor screen (view [sample](#)).

Test Input Rules

After you complete the Input Rule Editor screen, the Map Verification screen appears.

Note: When present, protected (output) fields appear with a V at every fifth byte, a number at every tenth byte and four asterisks between, for the length of your defined field: (****V****1****V****2 etc.).

1. Enter sample data into the input fields displayed on the screen.
2. After keying sample data into all desired fields, press Enter.
 - ◆ If your entries are valid, BMS/TS displays the sample data as it would appear when your map is being used.

- ◆ If your entries are invalid, BMS/TS displays an error message based on the input rule.
3. Repeat steps 1 and 2 to test the input rules for all desired fields.
 4. Press Clear. The Select a Map screen redisplay.
 5. Press F3 to exit, or select another map to test.
-

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Select a Transaction Monitor](#)
- [Edit the Transaction Monitor](#)
- [Reload the Transaction Monitor](#)

Related topics:

- [Working with Input Rules](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Implement Input Rules

You will use the BMS/TS Transaction Monitor to identify all maps in your application program which contain input rules.

BMS/TS will verify that the user input for a given map agrees with the appropriate input rules before passing the data to the application program.

Note: You do not need to make an entry in the Transaction Monitor to simply test input rules.

After you add an entry to the Transaction Monitor, you must reload the monitor into memory so that BMS/TS can correctly validate user entries.

To implement your input rules, you must also replace your program name in the Program Control Table (PCT) with the BMS/TS program name, GTBTHLP.

Because you may not have access to either the Transaction Monitor or the PCT, contact your Product Administrator for assistance, if necessary.

Select a Transaction Monitor

Perform the following steps to select the Transaction Monitor so you can add an entry to implement your input rules.

1. From the Main Menu, select the System option and press Enter. The System Functions Menu displays (view [sample](#)).
2. From the System Functions Menu, select the Transaction Monitor option and press Enter. The Transaction Monitor Options Menu displays (view [sample](#)).
3. Select the Edit Monitor option and press Enter. The Select a Transaction Monitor screen displays (view [sample](#)).
4. Type the name of the Transaction Monitor you want to edit in the Transaction Monitor Name field.

Remember, in almost all cases the Transaction Monitor you will use is called Monitor. Because Monitor is also the default entry for this field, you can simply press Tab to accept the default and move to the Password field.

5. If you password-protected the Transaction Monitor, type the password in the Password field. Otherwise skip this field.
 6. Press Enter. BMS/TS displays the Transaction Monitor Control screen (view [sample](#)).
-

Edit the Transaction Monitor

Editing the Transaction Monitor lets you add an entry to define which transactions BMS/TS should monitor for input rules. You can also use this option to modify an existing entry in the Transaction Monitor.

After you select a Transaction Monitor, the Transaction Monitor Control screen appears (view [sample](#)).

1. Press Tab to move to any LC (Line Command) field.
2. Type I (Insert) and the number of lines you want to insert. For example, to insert two lines, you would type I2.

Note: BMS/TS automatically sorts all entries when you exit the Transaction Monitor. This means that you can add entries in any order using a line command to insert blank lines. See Appendix A, [Using the Line Editor](#), for a description of the line commands you can use for this screen.

3. Press Enter. BMS/TS inserts the desired number of lines, then moves the cursor to the Tran field of the first blank line.
4. Type the transaction code that you want BMS/TS to monitor for input rules. Enter the transaction code as it appears in the PCT.
5. Press Tab to move to the Fkey field, then type *VAL. This tells BMS/TS to use the appropriate input rules to validate all screen input from the maps associated with the transaction code you entered in step 4.
6. Press Tab to move to the File field, then type the prefix of the library where the transaction is stored. You can also leave this field blank to accept the default, **GT**.
7. Press Tab to move to the Description field, then type any comments you want regarding the entry. You can leave this field blank to accept the default value, ***BMS/TS VALIDATION***.

Note: The Set/Type and Member fields are associated with help screens and are not required at this point.

8. Repeat steps 4 through 7 until you have completed your entries.
9. Press F3. BMS/TS displays the message SORT Processing. BMS/TS then clears the screen and displays a screen prompting you to load the Transaction Monitor.
10. Press Enter if you want to load the Transaction Monitor now. Press Clear to exit this screen and return to the Select a Transaction Monitor screen.

- ◆ If you press Enter, BMS/TS loads the Transaction Monitor (which immediately implements the input rules), then returns to the Select a Transaction Monitor screen. You do **not** have to complete the steps to reload the monitor (below).
 - ◆ If you press Clear, the changes you made to the input rules will remain intact but will not be recognized by BMS/TS until you [reload](#) the Transaction Monitor.
-

Reload the Transaction Monitor

After you edit the Transaction Monitor, you must reload the monitor before the changes take effect. If you do not reload the monitor, your input rules are saved, but BMS/TS does not apply them to user input.

1. From the Transaction Monitor Options Menu, select the Refresh Monitor option and press Enter. The Reload Monitor screen displays (view [sample](#)).
2. Press Enter to reload the Transaction Monitor.

Note: BMS/TS can only load a Transaction Monitor into CICS memory if the monitor has a set name of *SYSTEM and a member name of MONITOR. You can create a Transaction Monitor table with another name, but you must rename it *SYSTEM/MONITOR or merge it with *SYSTEM/MONITOR before you can load it.

BMS/TS reloads the Transaction Monitor, then displays the message System Monitor Initialization Complete.

3. Press Enter. BMS/TS returns to the Select a Transaction Monitor screen.
 4. Press F3 to exit.
-

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Add Command–level Call](#)
- [Modify COMMAREA](#)

Related topics:

- [Working with Input Rules](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Using Input Rules With Conversational Programs

Pseudo–conversational programs return control to CICS at each screen "read". Each time such an application program returns control to CICS, BMS/TS can intervene and take control itself. This allows BMS/TS to use input rules to evaluate and respond to user entries before calling the application program.

Conversational programs, however, do not return control to CICS at each screen "read". Therefore, BMS/TS does not have an opportunity to redirect user entries through your input rules. For conversational programs, you can create the same form of interaction by placing a simple "call" in front of your application program's "receive" command. As a result, your program interacts with your input rules in the same way a pseudo–conversational program would.

The following modifications allow your input rules to interface with a conversational program:

- Add a command–level call to your application program.
- Modify the COMMAREA of your application program.

Add a Command–Level Call to Your Application

You can add a command–level call to your application program so that the BMS/TS input rules can interface with a conversational program.

Place a "call" command in front of your application program's "receive" command. Write the single line command–level call exactly as follows:

```
EXEC CICS LINK PROGRAM('GTBVALR') COMMAREA('VALCOMM') LENGTH(36)
EXEC CICS RECEIVE MAP(. . .
```

Note: The second line of the above example represents the beginning of your existing RECEIVE command.

Modify the COMMAREA of Your Application

You can modify the COMMAREA of your application program so that the BMS/TS input rules can interface with a conversational program.

Add the following information to the COMMAREA of your application program:

```
000400 01 VALCOMM.  
000410 03 FILLER          PIC X(8) VALUE 'GTBVALR'.  
000420 03 FILLER          PIC X   VALUE 'C'.  
000430 03 FILLER          PIC X   VALUE LOW-VALUE.  
000440 03 SET-NAME        PIC X(8) VALUE 'ssssssss'.  
000450 03 MAP-NAME        PIC X(8) VALUE 'mmmmmmm'.  
000460 03 FILLER          PIC X(8) VALUE SPACE.  
000470 03 DDNAME-PFIX    PIC X(2) VALUE 'xx'.
```

In the sample code above:

ssssssss Mapset name used in the RECEIVE command.

mmmmmmm Map name used in the RECEIVE command.

xx Two character prefix of the library containing the map. The default library is **GT**.

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Table Types](#)
- [Table Options Menu](#)

Related topics:

- [Design a Table](#)
- [Redesign \(Edit\) a Table](#)
- [Test a Table](#)
- [Submit a Table](#)
- [List Tables](#)
- [Using Tables in an Application](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Working with Tables

A table is a collection of data in the form of an array of rows and columns. Each row represents a record, and the columns represent adjacent fields.

In BMS/TS, a row of information is an entry. You can define entries in a BMS/TS table as character, numeric, alphanumeric, or hexadecimal data. However, an entry cannot exceed 200 bytes, regardless of the number of columns.

You can use BMS/TS tables with your input rules to perform table look-ups. This lets you check for valid codes during data entry validation, and indicate how BMS/TS should respond to these entries. In the same way that input rules eliminate the need for writing input validation routines in your application program, tables keep you from having to create databases or files to compare to input validation routines.

You can also use table lookups to return cross-referenced information to a field on your map based on the user's entry.

Table Types

BMS/TS lets you construct either a one or two column table. The first column contains the argument. The second column, if you choose to create one, contains the object.

One-Column Tables

One-column or "argument-only" tables list all valid entries for a particular field on your map. To use a one-column table, you perform two steps:

1. Construct a table listing all valid inputs,
2. Write an input rule telling BMS/TS to reject all input that does not match an entry in the table.

Two-Column Tables

Two-column or "argument/object" tables let you return cross-referenced information to a field on your map based on the user's entry. To use a two-column table, you perform two steps:

1. Create a table that lists all valid entries for a particular field in the argument column and the corresponding information in the object column,
 2. Write an input rule so that if the user entry matches an argument in the table, BMS/TS displays the corresponding object in a specified field on the map.
-

Table Options Menu

When you select the Tables option from the Main Menu, the Table Options Menu displays (view [sample](#)). This menu offers the following options:

- Design** Create a new table. You will:
- Name the table.
 - Define the number of columns, the length of entries, and each entry's data format (alphanumeric, character, numeric, or hexadecimal).
 - Enter your list of valid entries into the argument and object columns.
- Redesign** Change or modify any or all elements of an existing table. To redesign a table, you step through the same options that you performed when you created a table, including adding, changing, or deleting table entries.
- Test** Test the accuracy of individual entries. You can type sample arguments, then review the corresponding objects returned by BMS/TS.
- Submit** Catalog your tables into the CICS system library.
- List** Display a list of tables stored in the current BMS/TS library.

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Select \(Name\) the Table](#)
- [Define Data Types and Lengths](#)
- [Enter Table Information](#)

Related topics:

- [Working with Tables](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Design a Table

When you design a new table, you will:

- Name the table and (optionally) protect the table with a password.
- Define the data types and lengths of the table's arguments and objects.
- Type the list of valid entries into the argument and object columns.

Select (Name) the Table

Selecting a table lets you name the new table, and protect it with a password (if desired).

1. From the Main Menu, select the Maps option and press Enter. The Maps Menu displays (view [sample](#)).
 2. From the Maps Menu, select the Tables option and press Enter. The Table Options Menu displays (view [sample](#)).
 3. From the Table Options Menu, select the Design option and press Enter. The Select a Table screen displays (view [sample](#)).
 4. Type a name (1–8 characters) in the Table name field. To display a list of tables in the current BMS/TS library, press F2.
 5. If you want to protect the table with a password, type the password in the Password field. Remember that if you assign a password to the table, you must enter the password each time you edit the table with BMS/TS.
 6. Press Enter. BMS/TS displays the Define Data Types and Lengths screen (view [sample](#)).
-

Define Data Types and Lengths

Defining data types and lengths of your table lets you specify an individual table's arguments and objects.

Note: Tables and their corresponding map fields must have the same data types and lengths.

After you [select](#) (name) the table, the Define Data Types and Lengths screen appears (view [sample](#)).

1. Type a brief description (1–35 characters) of the table in the Description field.
 2. Press Tab to move to the Argument type field.
 3. Type the argument type in the Argument type field. This defines the format of the data from the field on your map. The valid argument type codes appear on screen.
 4. Type the argument length in the Argument length field. Because the argument and object lengths combined cannot exceed 200, the maximum length for the Argument length field is 199 if you are designing an argument/object table, or 200 if you are designing an argument-only table.
 5. Press Tab to move to the Object type field.
 6. Specify the object type:
 - ◆ If you want to design an argument-only table (one column), type A (alphanumeric). You must enter an object type even if your table will not include an object.
 - ◆ If you want to design an argument/object table (two columns), type the appropriate object type. This defines the format of the data returned from the table to your map. The valid object type codes appear on screen.
 7. Specify the object length:
 - ◆ If you want to design an argument-only table (one column), press Tab to skip the Object length field. Notice that you design an argument-only table by defining an object type but setting the object length to zero.
 - ◆ If you want to design an argument/object table (two columns), type the object length in the Object length field. This defines the length of the data that BMS/TS can return to your map. Because the argument and object lengths combined cannot exceed 200, the maximum length for the Object length field is 199.
 8. Press Enter. BMS/TS displays the Table Editor screen (view [sample](#)).
-

Enter Table Information

Entering table information lets you display the BMS/TS Table Editor screen so you can define the argument and object entries for your table.

Note: BMS/TS initially provides 17 lines for table entries. If you fill all 17 lines with data, press Enter to insert an additional line. Then, as you fill each new line with data and press Enter, BMS/TS will insert an additional line.

You can add table entries in any order. BMS/TS automatically sorts the table in ascending order by argument when you exit the table editor.

After you [define the data types and lengths](#), the Table Editor screen appears (view [sample](#)). The Desc field contains the description that you entered at the Define Data Types and Lengths screen when you defined the table (if any).

1. Type an argument entry in the first field of the Argument column. This defines the data from the field on your map.

At this point you have several options, depending on whether are entering table information for an argument-only table, or for an argument/object table.

If you are entering data for an...	Then...
Argument-only table (one column)	Press Tab to move to the second field of the Argument column, and then repeat step 1 to enter all desired arguments. Then continue at step 6.
Argument/Object table (two columns)	Complete steps 2 through 7.

2. Press Tab to move to the first field of the Object column.
3. Type the corresponding object entry. This defines the data that BMS/TS returns to your map.
4. Press Tab to move to the second field of the Argument column.
5. Repeat steps 1–4 to define all arguments and objects for your table.
6. Press F3. BMS/TS displays the message SORT – PROCESSING as it sorts your entries. The Select a Table screen then displays.
7. You must [reload the Transaction Monitor](#) in order to access the table you just created.

The table below lists the function keys you can use while entering table information.

Function Key	Description
F1	Help. Displays help information for a specific field. Press F1 again for help concerning the screen you are using.
F3	Exit. Saves your entries and exits to the Select a Table screen.
F4	Options. Returns to the Define Data Types and Lengths screen.
F5	Rloc. Repeats a previous search.
F6	Rchg. Repeats a previous change string command.
F7	Bkwd. Scrolls backward one screen (if applicable).
F8	Fwd. Scrolls forward one screen (if applicable).
F9	ScHf. Lets you toggle between half-page and full-page scrolling.
F12	Cursor. Moves the cursor to the command line.

[Top of page](#)

Contents:

- [Select a Table to Redesign](#)
- [Update Data Types and Lengths](#)
- [Update Table Information](#)

Related topics:

- [Working with Tables](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Redesign (Edit) a Table

When you redesign (edit) an existing table with BMS/TS, you can:

- Change the data types and lengths of your table's arguments and objects.
- Edit, delete, or add to your list of valid entries in the argument and object columns.

Select a Table to Redesign

1. From the Main Menu, select the Maps option and press Enter.
2. From the Maps Menu, select the Tables option and press Enter.
3. From the Table Options Menu, select the Redesign option and press Enter. The Select a Table screen displays (view [sample](#)).
4. Type the name of the table (up to 8 characters) you want to redesign in the Table name field. To display a list of tables in the current BMS/TS library, press F2.
5. If you password-protected the table, type the password in the Password field.
6. Press Enter. BMS/TS displays the Define Data Types and Lengths screen (view [sample](#)).

Update Data Types and Lengths

Updating data types and lengths of your table lets you edit an individual table's arguments and objects.

Note: Tables and their corresponding map fields must have the same data types and lengths.

After you [select](#) a table to edit, the Define Data Types and Lengths screen appears (view [sample](#)).

1. (Optional) Edit the current description of your table.
2. Press Tab to move to the Argument type field.
3. Type the new Argument type code. The valid argument type codes appear on screen.

4. Repeat steps 2 and 3 to edit all desired fields.
5. Press Enter. BMS/TS displays the Table Editor screen (view [sample](#)).

Update Table Information

Updating table information lets you display the BMS/TS Table Editor screen so you can edit, add to, or delete the entries for your table.

Note: The Table Editor initially provides 17 lines for table entries. If you fill all 17 lines with data, press Enter to insert an additional line. Then, as you fill each new line with data and press Enter, BMS/TS will insert an additional line.

You can enter your entries in any order. BMS/TS automatically sorts the table in ascending order by argument when you exit the table editor.

After you complete the Define Data Types and Lengths screen, the Table Editor screen appears (view [sample](#)).

1. Press Tab to move to the field you want to update.
2. Update all desired information by typing over the existing data.
3. Repeat steps 1 and 2 to edit all fields on the screen.
4. Press F3. BMS/TS displays the SORT – Processing message as it sorts your entries. The Select a Table screen then displays.

Note: When reading your table, BMS/TS looks for the table in the load module first. If you have already submitted the table that you are redesigning, you must [resubmit](#) your updated table to place it in the load module. If you do not submit the revised table, BMS/TS continues to read the original from memory.

5. You must [reload the BMS/TS Transaction Monitor](#) in order to access the table you just updated.

You can use the following function keys to help you update table information:

Function Key	Description
F1	Help. Displays help information for a specific field. Press F1 again for help concerning the screen you are using.
F3	Exit. Saves your entries and exits to the Select a Table screen.
F4	Options. Returns to the Define Data Types and Lengths screen.
F5	Rloc. Repeats a previous search.
F6	Rchg. Repeats a previous change string command.
F7	Bkwd. Scrolls backward one screen (if applicable).
F8	Fwd. Scrolls forward one screen (if applicable).

- F9** ScHf. Lets you toggle between half–page and full–page scrolling.
- F12** Cursor. Moves the cursor to the command line.

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Select a Table to Test](#)
- [Test Table](#)

Related topics:

- [Working with Tables](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Test a Table

You will want to test the accuracy of a table before you use it in an application.

When you test a table, you sequentially browse through the information in the table, enter sample arguments, then have BMS/TS return the corresponding objects or errors.

Select a Table to Test

1. From the Main Menu, select the Maps option and press Enter.
2. From the Maps Menu, select the Tables option and press Enter.
3. From the Table Options Menu, select the Test option and press Enter. The Select a Table to Test screen displays (view [sample](#)).
4. Type the name of the table (1–8 characters) you want to test in the Table name field. To display a list of tables in the current BMS/TS library, press F2.
5. If you password-protected the table, type the password in the Password field.
6. Press Enter. BMS/TS displays the Test Table screen (view [sample](#)).

Test Table

After you [select](#) a table to test, the Test Table screen appears (view [sample](#)).

1. Perform the desired test function:

If you want to...	Then...
Browse through the table sequentially	Press F8 to display the first argument and its corresponding object (if applicable). After you begin reviewing arguments, you can also press F7 to display the previous argument and its corresponding object (if applicable).
Enter a sample argument to view the corresponding objects	Type an argument and press Enter. If the argument is valid, BMS/TS displays the corresponding object. If not, BMS/TS displays an error message.

2. Press F3. The Select a Table to Test screen redisplay.

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Select a Table to Submit](#)
- [Submit a Table](#)

Related topics:

- [Working with Tables](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Submit a Table

This section describes how to catalog tables in the CICS system library.

You need to create a JCL submit member for tables. See the *BMS/TS Administrator's Guide* for instructions on [creating a JCL submit member](#).

Submitting a table lets you catalog a table in the CICS system library. After you catalog a table, it becomes a load module. Converting a table into a load module enables your input rules to access the table faster.

Select a Table to Submit

Selecting a table lets you choose the table to catalog in your CICS system library.

1. From the Main Menu, select the Maps option and press Enter.
2. From the Maps Menu, select the Tables option and press Enter.
3. From the Table Options Menu, select the Submit option and press Enter. The Select a Table to Submit screen displays (view [sample](#)).
4. Type the name (up to 8 characters) of the table you want to submit in the Table name field.

Note: To display a list of tables in the current BMS/TS library, press F2.

5. If you password-protected the table, type the password in the Password field.
6. Press Enter. BMS/TS displays the Batch Job Submission screen (view [sample](#)).

Submit Table

You can specify the parameters necessary to submit a group of BMS/TS tables to your CICS system library.

After you [select](#) a table to submit, the Batch Job Submission screen appears. Note that the screen you see depends on your operating system (view [sample OS/390](#) screen).

1. Type the JCL member name you wish to use in the JCL member name field, or press Tab to accept the default (**SUBTAB**). This name must coincide with the library names used in your company.

2. Type any JCL Parameters Options in the appropriate fields, or press Tab to accept the defaults (BMS/TS automatically fills these fields if you have created a default JCL Submit Member).
 3. Press Enter. BMS/TS begins submitting the map and displays the message Job Submission in Progress. When the submit is complete, the Operation Complete message appears and the cursor returns to the Set or Type name field.
 4. Type the name (up to 8 characters) of the next table you want to submit in the Set or Type name field. To display a list of mapsets in the current BMS/TS library, press F2.
 5. Repeat steps 1–4 until you have submitted all tables.
 6. Press F3. BMS/TS returns to the Table Options menu.
-

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- List Tables in the Current Library

Related topics:

- [Working with Tables](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

List Tables in the Current Library

You can view a list of all tables stored in the current CICS system library.

List Tables in the Current Library

To list all tables in your current library:

1. From the Main Menu, select the Maps option and press Enter.
2. From the Maps Menu, select the Tables option and press Enter.
3. From the Table Options Menu, select the List option and press Enter. The Listing of Members screen displays (view [sample](#)).
4. Review the [displayed information](#) (described below). You can use the [function keys](#) listed at the bottom of the screen to help you review the list of members.
5. If you want to advance the list to a specific table, type the table name at the command prompt.
6. Press Enter. The list scrolls to the desired table name.
7. To exit this screen, press F3 to return to the Table Options Menu.

The Listing of Members screen includes the fields described in the table below.

Field	Description
?	Entry field for single character command. Press F1 for details. See Review List of Members for valid commands .
Set/Type	Always *TABLE.
Member	Table name.
Flags	Status code(s) for the table: <ul style="list-style-type: none"> * Table is password-protected. P Table is in production status (locked); cannot be updated.

U Table is currently being updated.

- Date** Date the table was last updated. By default, the date is displayed in *mmddccyy* format; if European dates are used, the date is formatted as *ddmmccyy*.
- Time** Time of day that the table was last updated.
- Description** Description of table as entered in Define Data Types and Lengths screen.

The following fields are displayed when you press F2 for the alternate display.

- OPID** Terminal or operator ID (operator ID used whenever present).
- SIZE** *Not applicable.*
- START** *Not applicable.*
- MAP** *Not applicable.*
- DSECT** *Not applicable.*

You can use the function keys described in the following table to navigate the list.

Fkey	Description
F1	Display screen level help for this screen.
F2	View the alternate display for this screen. Press F2 to toggle between these two displays.
F3	Exit — cancel this option and return to the Map Options menu.
F7	Scroll backward one screen (if applicable).
F8	Scroll forward one screen (if applicable).

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [COBOL Call or Link to the BMS/TS Table Processor](#)
- [Assembler Call or Link to the BMS/TS Table Processor](#)

Related topics:

- [Working with Tables](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Using BMS/TS Tables in Your Application

You can access load module tables or online tables created with BMS/TS from your own application. You can create a table for BMS/TS to use, along with input rules, when validating user entries. These "table look-ups" ensure that the user enters valid data in a field.

BMS/TS can access two types of tables:

- Online tables (tables stored in the VSAM internal file), or...
- Load module tables (tables stored in the BMS/TS LOAD module).

You can access load module or online tables created with BMS/TS directly from within a CICS application or through batch program calls. However, to access either type of table, you must first write a COBOL or Assembler program that calls or links to the BMS/TS table processor (which interfaces with tables created with BMS/TS). You use Table User Interface (TUI) control blocks to do this. You can use the same TUI control block for both online and batch environments.

Remember, the body of your program must include a call routine to the BMS/TS table processor. The call to the table processor has two stages. First, it must locate the table you wish to process. Secondly, it must search the table for the value it is to find.

COBOL Call or Link to the BMS/TS Table Processor

You can write a COBOL program that calls or links to the BMS/TS interface programs that access tables created with BMS/TS.

Your COBOL program must include the symbolic map needed to establish names that represent control block descriptions. (View [sample symbolic map](#).)

Parameters (COBOL Call or Link to BMS/TS Table Processor)

Your COBOL program includes required, optional, and returned parameters.

Required Parameters

These are the minimum parameters that your application program must fill in before the COMMAREA/Linkage area is passed to BMS/TS:

Parameter	Description	Definition
TUI-DIS-TABLE-NAME	Table Name	Name of the table to be accessed.
TUI-ACTION-CODE	Action Code	Tells the table processor what action to take on the table.
TUI-DIS-ARG-OBJ	Argument	Value to be looked up in the table: <ul style="list-style-type: none">• If the argument is null, then the first object in the table will be returned.• If the argument is null and the TUI-TABLE-LOCATE-CODE is NEXT, then the first object in the table will be returned.• If the argument is null and the TUI-TABLE-LOCATE-CODE is PRIOR, then the last object in the table will be returned.• If the argument is not null and the TUI-TABLE-LOCATE-CODE is EQUAL, then the object that matches the argument will be returned.

Optional Parameters

You can also pass these parameters to the BMS/TS table processor to further tailor the table look-up:

Parameter	Description	Definition
TUI-DIS-SRC-LENGTH	Object Return Length	Number of bytes of the object to be returned.
TUI-DIS-KEY-LENGTH	Argument Key Length	Length of the argument to be used in LOCATE comparison.
TUI-TABLE-LOCATE-CODE	Table Locate	Allows NEXT/PRIOR/EQUAL/LIST. This indicates the type of search to be conducted: search for a single table or a list of tables. <ul style="list-style-type: none">• EQUAL instructs the table processor to locate the table named in the table name field.• If the argument is given as nulls, NEXT is interpreted as the last entry in the list and PRIOR is interpreted as the first entry in the list.

TUI-OBJECT-LOC-CODE	Object Locate	<p>Allows NEXT/PRIOR/EQUAL.</p> <ul style="list-style-type: none"> • LIST can be used to bulk load/unload files. This enables you to search multiple tables (up to 21). The list must terminate with x'FF' and can exceed the values given. If LIST is passed, then the TUI-LST will be used to determine the list of tables to be loaded.
TUI-TABLE-TYPE	Table Type	<p>Tells the processor how to search for the specified table:</p> <ul style="list-style-type: none"> • EQUAL instructs the processor to find an argument in a specified table that matches the argument. • NEXT and PRIOR instruct the processor to find an argument one entry in the table previous to or following the current entry. (If the argument is null, then NEXT = FIRST and PRIOR = LAST). • 'T' -- Search VSAM internal file only. • 'V' -- Search the VSAM internal file first, then the LOAD module. • 'P' -- Search the LOAD module only; nulls force the default: search the LOAD module first, then the VSAM internal file.

Returned Parameters

The BMS/TS table processor can also pass these parameters back to your application program following the table lookup:

Parameter	Description	Definition
TUI-TABLE-TYPE	Table Type	<p>Indicates where the specified table was found.</p> <ul style="list-style-type: none"> • T -- Table found in a test state on the VSAM GT\$FILE. • V -- Table found on the VSAM file (but not as part of the load module). • P -- Table found as part of the load module.
TUI-ARGUMENT-LENGTH	Argument Length	Length of the full argument found by the table.
TUI-OBJECT-LENGTH	Object length	Length of the object found by the table.
TUI-ARGUMENT-TYPE	Argument Type	Type of the argument: A (Alphanumeric), C (Character), N (Numeric), or H (Hexadecimal).

TUI-OBJECT-TYPE	Object Type	Data type of the object returned to you in the object field: A (Alphanumeric), C (Character), N (Numeric), or H (Hexadecimal).
TUI-DIS-ARG-OBJ	Full Argument Requested object	In this field, you will receive first the argument, then the object that matches the argument.

Program (COBOL Call or Link to the BMS/TS Table Processor)

The COBOL program you write lets you access load module or online tables created with BMS/TS directly from within a CICS application or through batch program calls. Samples of both batch and online programs appear below.

Remember, the body of your program must include a call routine to the BMS/TS table processor. This call must first locate the table you wish to process, then search the table for the desired value.

Sample Programs

[Sample Batch Program](#) (COBOL Call or Link to Table Processor)

This sample batch program reads the table 'STATES' from the beginning to the end and prints the results. Notice that in order to access the table, the program calls the BMS/TS GTBTUIB program and passes the TUI control block.

[Sample Online Program](#) (COBOL Call or Link to Table Processor)

This sample online program reads the table 'STATES' from the beginning to the end and prints the results. Notice that in order to access the table from within a CICS program, the online program links to the BMS/TS GTBTUIO program and passes the address of the TUI control block.

Assembler Call or Link to the BMS/TS Table Processor

You can write an Assembler program that calls or links to the BMS/TS interface programs that access tables created with BMS/TS.

Your Assembler program must include the symbolic map needed to establish names that represent control block descriptions. (View [sample symbolic map](#).)

Parameters (Assembler Call or Link to the BMS/TS Table Processor)

Your Assembler program includes required, optional, and returned parameters.

Required Parameters

These are the minimum parameters that your application program must fill in before the COMMAREA/Linkage area is passed to BMS/TS:

Parameter	Description	Definition
-----------	-------------	------------

TUIDISTN	Table Name	Name of the table to be accessed.
TUIACTCD	Action Code	Tells the table processor what action to take on the table.
TUIDISAO	Argument	Value to be looked up in the table: <ul style="list-style-type: none"> • If the argument is null, then the first object in the table will be returned. • If the argument is null and the TUITLCCD is NEXT, then the first object in the table will be returned. • If the argument is null and the TUITLCCD is PRIOR, then the last object in the table will be returned. • If the argument is not null and the TUITLCCD is EQUAL, then the object that matches the argument will be returned.

Optional Parameters

You can also pass these parameters to the BMS/TS table processor to further tailor the table lookup:

Parameter	Description	Definition
TUIDISOS	Object Return Length	Number of bytes of the object to be returned.
TUIDISAS	Argument Key Length	Length of the argument to be used in LOCATE comparison.
TUITLCCD	Table Locate	Allows NEXT/PRIOR/EQUAL/LIST. This indicates the type of search to be conducted: search for a single table or a list of tables. <ul style="list-style-type: none"> • EQUAL instructs the table processor to locate the table named in the table name field. • If the argument is given as nulls, NEXT is interpreted as the last entry in the list and PRIOR is interpreted as the first entry in the list. • LIST can be used to bulk load/unload files. This enables you to search multiple tables (up to 21). The list must terminate with x'FF' and can exceed the values given. If LIST is passed, then the TUI-LST will be used to determine the list of tables to be loaded.
TUIOLCCD	Object Locate	Allows NEXT/PRIOR/EQUAL. <ul style="list-style-type: none"> • EQUAL instructs the processor to find an argument in a specified table that matches the argument. • NEXT and PRIOR instruct the processor to find an argument one entry in the table previous to or following the current entry. (If the argument is null, then NEXT = FIRST and PRIOR = LAST).
TUITLBTP	Table Type	Tells the processor how to search for the specified table:

- **'T'** -- Search VSAM internal file only.
- **'V'** -- Search the VSAM internal file first, then the LOAD module.
- **'P'** -- Search the LOAD module only; Nulls force the default: search the LOAD module first, then the VSAM internal file.

Returned Parameters

The BMS/TS table processor can also pass these parameters back to your application program following the table lookup:

Parameter	Description	Definition
TUITBTP	Table Type	Indicates where the specified table was found. <ul style="list-style-type: none"> • T -- Table found in a test state on the VSAM GT\$FILE. • V -- Table found on the VSAM file (but not as part of the load module). • P -- Table found as part of the load module.
TUIARGLN	Argument Length	Length of the full argument found by the table.
TUIOBJLN	Object length	Length of the object found by the table.
TUIARGTP	Argument Type	Type of the argument: A (Alphanumeric), C (Character), N (Numeric), H (Hexadecimal).
TUIOBJTP	Object Type	Data type of the object returned to you in the object field: A (Alphanumeric), C (Character), N (Numeric), H (Hexadecimal).
TUIDISAO	Full Argument Requested object	In this field, you will receive first the argument, then the object that matches the argument.

Program (Assembler Call or Link to the BMS/TS Table Processor)

The Assembler program you write lets you access load module or online tables created with BMS/TS directly from within a CICS application or through batch program calls.

Remember, the body of your program must include a call routine to the BMS/TS table processor. This call must first locate the table you wish to process, then search the table for the desired value.

In order to access the table from within a batch program, the program calls the BMS/TS GTBTUIB program and passes the TUI control block. To access the table from within a CICS program, the online program links to the BMS/TS GTBTUIO program and passes the address of the TUI control block.

Note: You must refer to the symbolic map from your program with the Assembler COPY <DSECTNAME> command.

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Sample Symbolic Map (Assembler Call or Link to the BMS/TS Table Processor)

Name	Description	
TUIDS	DSECT	
TUIXITPG	DS	CL8 EXIT PROGRAM NAME
TUICALPG	DS	CL8 CALLERS PROGRAM NAME
TUIXITTR	DS	CL4 EXIT TRANSACTION NAME
TUICALTR	DS	CL4 CALLERS TRANSACTION NAME
*		
TUIACTCD	DS	CL1 REQUESTED ACTION CODE
TUIACDTV	EQU	C'0' TABLE VERIFY X'FO'
TUIACDTL	EQU	C'1' TABLE LOAD X'F1'
TUIACDTD	EQU	C'2' TABLE DELOAD X'F2'
TUIACDL	EQU	C'3' TABLE LOAD/DELOAD
TUIACDTS	EQU	C'4' TABLE SEARCH X'F4'
TUIACDLS	EQU	C'5' TABLE LOAD/SEARCH
TUIACDSD	EQU	C'6' TABLE SEARCH/DELOAD
TUIACDT1	EQU	C'7' TABLE LOAD/SEARCH/DELOAD
TUIACDXB	EQU	C'8' EXIT CALL (BATCH ONLY)
*		
TUITLCCD	DS	CL1 TABLE LOCATE CRITERIA
TUITCDEQ	EQU	C'0' EQUAL TO TABLE NAME
TUITCDNX	EQU	C'1' NEXT TABLE NAME
TUITCDPR	EQU	C'2' PRIOR TABLE NAME
TUITCDLS	EQU	C'8' USE LIST IN TUITLSAD
*		
TUIOLCCD	DS	CL1 OBJECT LOCATE CRITERIA
TUIOCDEQ	EQU	C'0' EQUAL
TUIOCDNX	EQU	C'1' NEXT
TUIOC DPR	EQU	C'2' PRIOR
*		
TUIINTCD	DS	CL1 INTERNAL ONLY FUNCTIONS
*		
TUIARGLN	DS	H ARGUMENT LENGTH
TUIOBJLN	DS	H OBJECT LENGTH
*		
TUIARGTP	DS	CL1 ARG TYPE
TUIATPAN	EQU	C'A' ALPHANUMERIC
TUIATPCH	EQU	C'C' CHARACTER
TUIATPNM	EQU	C'N' NUMERIC
TUIATPHX	EQU	C'H' HEXADECIMAL
*		
TUIOBJTP	DS	CL1 OBJ TYPE
TUIOTPAN	EQU	C'A' ALPHANUMERIC
TUIOTPCH	EQU	C'C' CHARACTER
TUIOTPNM	EQU	C'N' NUMERIC
TUIOTPHX	EQU	C'H' HEXADECIMAL
*		
TUIRTNCD	DS	CL1 RETURN CODE
TUIRC DOK	EQU	X'00' SUCCESSFUL OPERATION
TUIRCDEA	EQU	C'A' ARGUMENT ERROR
TUIRCDEE	EQU	C'E' END OF FILE
TUIRCDEF	EQU	C'F' FUNCTION ERROR
TUIRCDET	EQU	C'T' TABLE ERROR
TUIRCDEX	EQU	C'X' FATAL ERROR
*		
TUITBLTP	DS	CL1 TABLE TYPE
TUITTPPV	EQU	X'00' PHASE/VSAM SEARCH

TUITTPTT	EQU	C'T'	TEST - VSAM ONLY NO LOAD
TUITTPTV	EQU	C'V'	LOAD VSAM FILE
TUITTPTP	EQU	C'P'	LOAD PHASE
*			
	ORG	TUIRTNCD	
TUIDISRC	DS	CL1	RETURN CODE
TUIDISTT	DS	CL1	TABLE TYPE
TUIDISTN	DS	CL8	TABLE NAME
TUIDISAS	DS	H	ARGUMENT KEY LENGTH
TUIDISOS	DS	H	OBJECT LENGTH
TUIDISAO	DS	CL200	ARGUMENT/OBJECT
TUIDISSZ	EQU	*-TUIDS	CONTROL BLOCK SIZE
	SPACE	2	
TUIMINLN	EQU	*-TUIDS	MINIMUM LENGTH
	SPACE	2	
	ORG	TUIRTNCD	
TUILST	DS	0C TABLE	LOAD/DELOAD LIST
TUILSTRC	DS	CL1	RETURN CODE
TUILSTTT	DS	CL1	TABLE TYPE
TUILSTTN	DS	CL8	TABLE NAME (X'FF'= END OF TABLE)
TUILSTLN	EQU	*-TUILST	SIZE OF LIST ENTRY
TUILSTS	DS	21CL(TUILSTLN)	REMAINING IN MIN DSECT
TUILSTCT	EQU	(*-TUILST)/TUILSTLN	NUMBER OF LST ENTRIES
TUILSTEN	DS	C	X'FF' = END OF DISPLAY
TUILSTSZ	EQU	*-TUIDS	LENGTH OF LST CONTROL BLOCK

Sample Symbolic Map (COBOL Call or Link to BMS/TS Table Processor)

```

05  TUI-EXIT-PROGRAM          PIC X(8) VALUE 'GTBTUIB'.
05  TUI-CALL-PROGRAM          PIC X(8) VALUE 'GTBTDISB'.
05  TUI-EXIT-TRAN             PIC X(4) VALUE 'BTCH'.
05  TUI-CALL-TRAN             PIC X(4) VALUE 'BTCH'.
*
05  TUI-ACTION-CODE           PIC X.
    88 TUI-ACT-VERIFY          VALUE '0'.
    88 TUI-ACT-LOAD            VALUE '1'.
    88 TUI-ACT-DELOAD          VALUE '2'.
    88 TUI-ACT-LOAD-DELOAD     VALUE '3'.
    88 TUI-ACT-SEARCH          VALUE '4'.
    88 TUI-ACT-LOAD-SEARCH     VALUE '5'.
    88 TUI-ACT-SEARCH-DELOAD   VALUE '6'.
    88 TUI-ACT-LOAD-SEARCH-DELOAD VALUE '7'.
    88 TUI-ACT-EXIT-BATCH     VALUE '8'.

05  TUI-TABLE-LOCATE-CODE     PIC X.
    88 TUI-TBLLOC-EQUAL        VALUE '0'.
    88 TUI-TBLLOC-NEXT         VALUE '1'.
    88 TUI-TBLLOC-PRIOR        VALUE '2'.
    88 TUI-TBLLOC-LIST         VALUE '8'.
*
05  TUI-OBJECT-LOCATE-CODE    PIC X.
    88 TUI-OBJLOC-EQUAL        VALUE '0'.
    88 TUI-OBJLOC-NEXT         VALUE '1'.
    88 TUI-OBJLOC-PRIOR        VALUE '2'.

05  FILLER                    PIC X.
*
05  TUI-ARGUMENT-LENGTH       PIC 9(3) COMP VALUE 0.
05  TUI-OBJECT-LENGTH         PIC 9(3) COMP VALUE 0.
*
05  TUI-ARGUMENT-TYPE         PIC X.
    88 TUI-ARG-ALPHNUM         VALUE 'A'.
    88 TUI-ARG-CHAR            VALUE 'C'.
    88 TUI-ARG-NUM             VALUE 'N'.
    88 TUI-ARG-HEX            VALUE 'H'.

05  TUI-OBJECT-TYPE           PIC X.
    88 TUI-OBJ-ALPHNUM         VALUE 'A'.
    88 TUI-OBJ-CHAR            VALUE 'C'.
    88 TUI-OBJ-NUM             VALUE 'N'.
    88 TUI-OBJ-HEX            VALUE 'H'.
*
05  TUI-SEARCH.

    10 TUI-RETURN-CODE        PIC X.
        88 TUI-RTN-OK          VALUE LOW-VALUE.
        88 TUI-RTN-ARGUMENT-ERR VALUE 'A'.
        88 TUI-RTN-END-OF-TABLE VALUE 'E'.
        88 TUI-RTN-FUNCTION-ERR VALUE 'F'.
        88 TUI-RTN-TABLE-ERR   VALUE 'T'.
        88 TUI-RTN-FATAL-ERR   VALUE 'X'.
        88 TUI-RTN-OBJ-ERR     VALUE 'O'.
*
    10 TUI-TABLE-TYPE         PIC X.

```

```

      88 TUI-TBL-ANY                VALUE LOW-VALUE.
      88 TUI-TBL-TEST              VALUE 'T'.
      88 TUI-TBL-FILE              VALUE 'V'.
      88 TUI-TBL-PHASE             VALUE 'P'.
*
10  FILLER                        PIC X(220).
*
05  TUI-DISPLAY REDEFINES TUI-SEARCH.

10  TUI-DIS-RTN                   PIC X.
10  TUI-DIS-TBL-TYPE              PIC X.
10  TUI-DIS-TBL-NAME              PIC X(8).
10  TUI-DIS-KEY-LENGTH            PIC S9(4) COMP.
10  TUI-DIS-SRC-LENGTH            PIC S9(4) COMP.
10  TUI-DIS-ARG-OBJ               PIC X(200).
10  FILLER                        PIC X(6).
*
05  TUI-LST REDEFINES TUI-SEARCH.
10  TUI-LST OCCURS 22 TIMES.
    15 TUI -LST-RTN                PIC X.
    15 TUI-LST-TABLE-TYPE          PIC X.
    15 TUI-LST-TABLE-NAME         PIC X(8).

```

Sample Batch Program (COBOL Call or Link to Table Processor)

This sample batch program reads the table 'STATES' from the beginning to the end and prints the results. Notice that in order to access the table, the program calls the BMS/TS GTBTUIB program and passes the TUI control block.

```

CBL LIB,APOST,NOXREF,NOSEQ
* OLD COMP PARMS *CBL LIB, STATE,APOST,NOXREF,NOSYM,NOCLIST,NOSEQ
IDENTIFICATION DIVISION. TST00190
PROGRAM-ID. GTBTDISM. TST00200
*****
* GTBTDISM - EXAMPLE OF A USER ONLINE TABLE INTERFACE *
* *
* LINKS TO GTBTUIB WITH CONTROL BLOCK CONTAINING TABLE *
* AND SEARCH INFORMATION *
* *
* PROGRAM DUTIES: PRINT CONTENTS OF A TABLE *
*****
ENVIRONMENT DIVISION. TST00390
CONFIGURATION SECTION.
SPECIAL-NAMES.
    C01 IS PAGE-TOP.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
* SELECT PRINT-FILE ASSIGN TO SYS015-UR-1403-S.
DATA DIVISION. TST00400
FILE SECTION.
*FD PRINT-FILE
* DATA RECORD IS PRINT-LINE
* FORMAT IS FIXED
* LENGTH IS 132
* LABEL RECORDS OMITTED.
WORKING-STORAGE SECTION. TST00410
01  I PIC S9(4) COMP VALUE 0.
01  PRINT-LINE                PIC X(132) .
01  HEADING-LINE              PIC X(80)   VALUE 'EXAMPLE OF COBOL BATCH PROGRAM - TABLE: S
01  ERROR-LINE.
    05  FILLER                 PIC X(18)   VALUE 'LAST RETURN CODE:'.
    05  ERROR-CODE             PIC X.
01  FMT-LINE.
    05  FILLER                 PIC X(10)   VALUE 'ARGUMENT:'.
    05  PRINT-ARGUMENT         PIC X(2)   VALUE SPACES.
    05  FILLER                 PIC X(5)   VALUE SPACES.
    05  FILLER                 PIC X(10)  VALUE 'OBJECT:'.
    05  PRINT-OBJECT           PIC X(20)  VALUE SPACES.
01  ARG-OBJ.
    05  ARG                    PIC X(2) .
    05  OBJ                    PIC X(20) .
01  LOC-VALUES.
    05  LOC-EQUAL              PIC X      VALUE '0'.
    05  LOC-NEXT               PIC X      VALUE '1'.
    05  LOC-PRIOR              PIC X      VALUE '2'.
01  ACT-VALUES.
    05  ACT-VERIFY             PIC X      VALUE '0'.
    05  ACT-LOAD               PIC X      VALUE '1'.
    05  ACT-DELOAD             PIC X      VALUE '2'.
    05  ACT-LOAD-DELOAD        PIC X      VALUE '3'.
    05  ACT-SEARCH             PIC X      VALUE '4'.
    05  ACT-EXIT               PIC X      VALUE '8'.

```

```

05  ACT-LOAD-SEARCH          PIC X          VALUE '5'.
05  ACT-SEARCH-DELOAD       PIC X          VALUE '6'.
01  TABLE-TEST             PIC X          VALUE 'T'.
01  TUI-CONTROL-BLOCK.
05  TUI-EXIT-PROGRAM        PIC X(8)         VALUE 'GTBTUIB'.
05  TUI-CALL-PROGRAM        PIC X(8)         VALUE 'GTBTDISB'.
05  TUI-EXIT-TRAN           PIC X(4)         VALUE 'GT '.
05  TUI-CALL-TRAN           PIC X(4)         VALUE 'GT '.
05  TUI-ACTION-CODE         PIC X.
88  TUI-ACT-VERIFY          VALUE '0'.
88  TUI-ACT-LOAD            VALUE '1'.
88  TUI-ACT-DELOAD          VALUE '2'.
88  TUI-ACT-LOAD-DELOAD     VALUE '3'.
88  TUI-ACT-SEARCH          VALUE '4'.
88  TUI-ACT-LOAD-SEARCH     VALUE '5'.
88  TUI-ACT-SEARCH-DELOAD   VALUE '6'.
88  TUI-ACT-LOAD-SEARCH-DELOAD VALUE '7'.
88  TUI-ACT-EXIT-BATCH     VALUE '8'.
05  TUI-TABLE-LOCATE-CODE   PIC X.
88  TUI-TBLLOC-EQUAL        VALUE '0'.
88  TUI-TBLLOC-NEXT         VALUE '1'.
88  TUI-TBLLOC-PRIOR       VALUE '2'.
88  TUI-TBLLOC-LIST        VALUE '8'.
05  TUI-OBJECT-LOCATE-CODE  PIC X.
88  TUI-OBJLOC-EQUAL        VALUE '0'.
88  TUI-OBJLOC-NEXT         VALUE '1'.
88  TUI-OBJLOC-PRIOR       VALUE '2'.
05  FILLER PIC X.
05  TUI-ARGUMENT-LENGTH     PIC S9(3)        COMP VALUE +0.
05  TUI-OBJECT-LENGTH       PIC S9(3)        COMP VALUE +0.
05  TUI-ARGUMENT-TYPE       PIC X.
88  TUI-ARG-ALPHNUM         VALUE 'A'.
88  TUI-ARG-CHAR            VALUE 'C'.
88  TUI-ARG-NUM             VALUE 'N'.
88  TUI-ARG-HEX             VALUE 'H'.
05  TUI-OBJECT-TYPE         PIC X.
88  TUI-OBJ-ALPHNUM         VALUE 'A'.
88  TUI-OBJ-CHAR            VALUE 'C'.
88  TUI-OBJ-NUM             VALUE 'N'.
88  TUI-OBJ-HEX             VALUE 'H'.
05  TUI-SEARCH.
10  TUI-RETURN-CODE         PIC X.
88  TUI-RTN-OK              VALUE LOW-VALUE.
88  TUI-RTN-ARGUMENT-ERR    VALUE 'A'.
88  TUI-RTN-END-OF-TABLE    VALUE 'E'.
88  TUI-RTN-FUNCTION-ERR    VALUE 'F'.
88  TUI-RTN-TABLE-ERR       VALUE 'T'.
88  TUI-RTN-FATAL-ERR       VALUE 'X'.
88  TUI-RTN-OBJ-ERR         VALUE 'O'.
10  TUI-TABLE-TYPE          PIC X.
88  TUI-TBL-ANY             VALUE LOW-VALUE.
88  TUI-TBL-TEST            VALUE 'T'.
88  TUI-TBL-FILE            VALUE 'V'.
88  TUI-TBL-PHASE           VALUE 'P'.
10  FILLER                  PIC X(220).
05  TUI-DISPLAY REDEFINES TUI-SEARCH.
10  TUI-DIS-RTN             PIC X.
10  TUI-DIS-TBL-TYPE        PIC X.
10  TUI-DIS-TBL-NAME        PIC X(8).
10  TUI-DIS-KEY-LENGTH      PIC S9(4) COMP.
10  TUI-DIS-SRC-LENGTH      PIC S9(4) COMP.
10  TUI-DIS-ARG-OBJ         PIC X(200).
05  FILLER REDEFINES TUI-SEARCH.

```

```

10 TUI-LST OCCURS 22 TIMES.
15 TUI-LST-RTN PIC X.
15 TUI-LST-TABLE-TYPE PIC X.
15 TUI-LST-TABLE-NAME PIC X(8) .

PROCEDURE DIVISION. TST00980
HOUSEKEEPING. TST00980
* OPEN OUTPUT PRINT-FILE. TST01040
MOVE HEADING-LINE TO PRINT-LINE.
* WRITE PRINT-LINE AFTER ADVANCING PAGE-TOP.
MOVE ALL 'T' TO TUI-TABLE-TYPE.
MOVE 'STATES' TO TUI-DIS-TBL-NAME.
MOVE ACT-LOAD TO TUI-ACTION-CODE.
PERFORM CALL-GTTABLE-PROCESSOR.
MOVE ACT-SEARCH TO TUI-ACTION-CODE.
MOVE 'STATES' TO TUI-DIS-TBL-NAME.
MOVE LOW-VALUES TO TUI-TABLE-LOCATE-CODE.
MOVE LOW-VALUES TO TUI-OBJECT-LOCATE-CODE.
MOVE 'GA' TO TUI-DIS-ARG-OBJ.
PERFORM CALL-GTTABLE-PROCESSOR THRU PRINT-ARG-END
VARYING I FROM 1 BY 1 UNTIL I > 1500.
* WRITE PRINT-LINE AFTER ADVANCING 2.
MOVE ACT-EXIT TO TUI-ACTION-CODE.
PERFORM CALL-GTTABLE-PROCESSOR.
* CLOSE PRINT-FILE.
GOBACK.
CALL-GTTABLE-PROCESSOR. TST00980
CALL 'GTBTUIB' USING TUI-CONTROL-BLOCK.
PRINT-ARG.
IF TUI-RTN-OK
MOVE TUI-DIS-ARG-OBJ TO ARG-OBJ
MOVE ARG TO PRINT-ARGUMENT
MOVE OBJ TO PRINT-OBJECT
MOVE FMT-LINE TO PRINT-LINE
* WRITE PRINT-LINE AFTER ADVANCING 1 LINE.
DISPLAY 'VALUE OF I IS ' I.
DISPLAY ' ' PRINT-LINE.
PRINT-ARG-END.

```

Sample Online Program (COBOL Call or Link to Table Processor)

This sample online program reads the table 'STATES' from the beginning to the end and prints the results. Notice that in order to access the table from within a CICS program, the online program links to the BMS/TS GTBTUIO program and passes the address of the TUI control block.

```

IDENTIFICATION DIVISION.
PROGRAM-ID. 'GTBOLTIP'.
AUTHOR. GT SOFTWARE INC.
REMARKS.
*****
* GTBOLTIP - EXAMPLE OF A USER ON-LINE TABLE INTERFACE *
*
* LINKS TO GTBTUIO WITH THE COMMAREA CONTAINING TABLE *
* AND SEARCH INFORMATION *
*
* PROGRAM DUTIES: ACQUIRE TABLE INFORMATION FROM BMS/TS*
* FOR ONLINE USE *
*****
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. IBM-370.
OBJECT-COMPUTER. IBM-370.
INPUT-OUTPUT SECTION.
DATA DIVISION.
WORKING-STORAGE SECTION.
01  HEADING-LINE                PIC X(80)    VALUE 'EXAMPLE OF COBOL BATCH PROGRAM - TABLE: STATES'.
01  ERROR-LINE.
    05  FILLER                   PIC X(18)    VALUE 'LAST RETURN CODE:'.
    05  ERROR-CODE PIC X.
01  FMT-LINE.
    05  FILLER                   PIC X(10)    VALUE 'ARGUMENT:'.
    05  PRINT-ARGUMENT           PIC X(2).
    05  FILLER                   PIC X(5)     VALUE SPACE.
    05  FILLER                   PIC X(10)   VALUE 'OBJECT:'
    05  PRINT-OBJECT             PIC X(20).
01  ARG-OBJ.
    05  ARG                      PIC X(2).
    05  OBJ                      PIC X(20).
01  LOC-EQUAL                   PIC X        VALUE '0'.
01  LOC-NEXT                    PIC X        VALUE '1'.
01  LOC-PRIOR                   PIC X        VALUE '2'.
01  ACT-VERIFY                  PIC X        VALUE '0'.
01  ACT-LOAD                    PIC X        VALUE '1'.
01  ACT-DELOAD                 PIC X        VALUE '2'.
01  ACT-LOAD-DELOAD            PIC X        VALUE '3'.
01  ACT-SEARCH                  PIC X        VALUE '4'.
01  ACT-EXIT                    PIC X        VALUE '8'.
01  ACT-LOAD-SEARCH            PIC X        VALUE '5'.
01  ACT-SEARCH-DELOAD          PIC X        VALUE '6'.

01  TABLE-TEST                PIC X        VALUE 'T'.
    * TABLE USER INTERFACE CONTROL BLOCK

01  USERMAP COPY USERMAP.

01  TUI-CONTROL-BLOCK COPY TUICBL.

PROCEDURE DIVISION.

```

```

HOUSEKEEPING.
  MOVE 'STATES'           TO TUI-DIS-TABLE-NAME.
  MOVE ACT-LOAD           TO TUI-ACTION-CODE.
  MOVE TABLE-TEST       TO TUI-TABLE-TYPE.

PERFORM CALL-GTTABLE-PROCESSOR THRU CALL-GTTABLE-PROCESSOR-EXIT.

  MOVE ACT-SEARCH        TO TUI-ACTION-CODE.
  MOVE LOC-NEXT          TO TUI-OBJECT-LOCATE-CODE.

PERFORM CALL-GTTABLE-PROCESSOR THRU PRINT-ARG-EXIT UNTIL
  NOT TUI-RTN-OK.

EXEC CICS SENDMAP('USERMAP') FROM(USERMAP) ERASE FREEKB.

  MOVE ACT-EXIT          TO TUI-ACTION CODE.
PERFORM CALL-GTTABLE-PROCESSOR THRU CALL-GTTABLE-PROCESSOR-EXIT.
EXEC CICS RETURN.

CALL-GTTABLE-PROCESSOR.
EXEC CICS LINK PROG('GTBTUIO') COMMAREA(TUI-CONTROL-BLOCK).
CALL-GTTABLE-PROCESSOR-EXIT. EXIT.

PRINT-ARG.
  IF NOT TUI-RTN-OK GO TO PRINT-ARG-EXIT.
  MOVE TUI-DIS-ARG-OBJ TO ARG-OBJ.
  MOVE ARG              TO PRINT-ARGUMENT.
  MOVE OBJ              TO PRINT-OBJECT.
  MOVE FMT-LINE        TO PRINT-LINE.

* USERS ADD CODE HERE FOR USE OF DATA ON LINE TO BE MOVED TO MAP
PRINT-ARG-EXIT. EXIT.

```

Contents:

- [Understanding Demos](#)
- [Demo Menu Options](#)

Related topics:

- [Design a Demo](#)
- [Redesign \(Edit\) a Demo](#)
- [Run a Demo](#)
- [List Demos](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Working with Demos

A demo is a working model of an application (a set of related maps).

You can use BMS/TS demos to simulate your application in order to test and display all appropriate maps before the application goes into production.

Understanding Demos

You design a demo by designating the maps you want BMS/TS to display in sequence, so that you or an end-user can run the demo. You can also assign each map in your demo to a different function key, so that you can display a map other than the next one in the sequence. This gives the demo a variety of ways to simulate your finished application.

Note: The operations normally performed using function keys in BMS/TS are unavailable when running a demo. When running a demo, the function keys only allow you to skip to individual maps.

Chaining Demos

While BMS/TS limits you to 10 maps for display when designing a demo, you can actually display an infinite number of maps by having one demo call another demo. You do this by designating another demo as one of the maps for display in your demo.

Demos and Input Rules

Input rules function when running a demo. If you have added input rules to your maps, your demo not only displays how your application looks, but also how it truly works. BMS/TS validates information keyed by the user, field by field, before proceeding to the next screen.

Demo Menu Options

When you select Demos from the main menu, the Demo Options Menu displays (view [sample](#)). The Demo Options Menu offers the following options:

- Design** Create a demo:
- Name the demo.
 - List the maps in the order you want them displayed.
 - Indicate whether you want BMS/TS to clear the screen with each map display, or display one map over another.
 - Create alternate work flows in your demo by assigning a function key to each map.
- Redesign** Change (edit) your demo by adding new maps, or rearranging the display sequence of existing maps.
- Run** Select the demo you want to run, as well as enter data as you move through the maps. Depending on how you design the demo, you can also press function keys to display specific maps.
- List** Display a list of demos stored in the current BMS/TS library.
-

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Select \(Name\) a Demo](#)
- [Design a Demo](#)

Related topics:

- [Working with Demos](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Design a Demo

When you design (create) a demo, you will:

- Name the demo.
- List the maps in the order you want them displayed.
- Indicate whether you want BMS/TS to clear the screen with each map display, or display one map over another.
- Create alternate work flows in your demo by assigning a function key to each map.

Select (Name) a Demo

Selecting a demo lets you name the new demo, and (optionally) protect the demo with a password. Perform the following steps to select a demo.

1. From the Main Menu, select the Maps option and press Enter. The Maps Menu displays (view [sample](#)).
2. From the Maps Menu, select the Demos option and press Enter. The Demo Options Menu displays (view [sample](#)).
3. From the Demo Options Menu, select the Design option and press Enter. The Select a Demo screen displays (view [sample](#)).
4. Type a demo name (1–8 characters) in the Demo name field.

Note: To display a list of demos in the current BMS/TS library, press F2.

5. If you want to password-protect the demo, type the password in the Password field. Remember that if you assign a password to your demo, you must enter the password each time you edit the demo.
 6. Press Enter. BMS/TS displays the Design a Demo screen (view [sample](#)).
-

Design a Demo

Designing a demo lets you select the maps you want to include in your demo and list the mapsets and map names for your demo in the sequence you want them displayed. You can tell BMS/TS to clear the screen with each display, or overlay your maps, during the actual demo. You can (optionally) assign function keys to the maps.

After you [select](#) (name) the demo, the Design a Demo screen appears (view [sample](#)).

1. The cursor is in the Mapset Name field of Step No. 0. Press Tab to move to the Description field.
2. Type a brief description (up to 35 characters) of the demo, then press Tab to return to the Mapset Name field of Step No. 0.
3. Type the mapset name of the first map you want displayed by the demo, then press Tab to move to the Member Name field.

If all maps in the demo have the same mapset name, you only need to enter the mapset name for the first entry. If you leave a Mapset Name field blank, BMS/TS defaults to the last mapset name field containing an entry.

4. Type the name of the first map you want displayed by your demo, then press Tab to move to the Erase field.
5. Indicate whether you want BMS/TS to clear the demo screen after each map displays (enter Y), or display one map over another on the demo screen (enter N). The cursor moves to the Func Key field.

The following two steps are **optional**. They allow you to customize your demo so that specific maps display when function keys are pressed during the actual demo. Unless you want the demo to display maps in an order other than that indicated on this screen, skip to step 8.

6. (*optional*) Type the 2-digit number of the function key that you want associated with this map. (For example, specify F1 as "01".) By entering a number in this field, you can automatically advance to another map during the demo when the appropriate function key is pressed.
7. (*optional*) To advance to another map when the function key is pressed, type the step number (0–9) of the map to which you want to advance.

For example, if you define the function key as 4 and the step as 5, BMS/TS automatically advances to the map defined in the Step No. 5 field when you press F4 from this map during the actual demo.

Note: The Errors/Comments field displays error or information messages about your demo. You cannot enter data in this field.

8. Repeat steps 1–7 to define the other maps you want to display in the demo.

You can define up to 10 maps (using the ten Step No. fields) for the demo. Remember, though, that you can chain an infinite number of maps for the demo by entering the name of another demo in a Mapset Name field.

9. After defining all maps for your demo, press F3 to save your demo and return to the Select a Demo screen.

See [Run a Demo](#) for information on running the demo.

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Select a Demo](#)
- [Redesign \(Edit\) a Demo](#)

Related topics:

- [Working with Demos](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Redesign (Edit) a Demo

Redesigning a demo lets you modify the demo's design. You can:

- Add maps to the demo,
- Delete maps from the demo,
- Change the sequence of maps displayed,
- Change function key assignments.

Select a Demo

Perform the following steps to select a demo for redesign.

1. From the Main Menu, select the Maps option and press Enter to display the Maps Menu (view [sample](#)).
2. From the Maps Menu, select the Demos option and press Enter to display the Demo Options Menu (view [sample](#)).
3. From the Demo Options Menu, select the Redesign option and press Enter. The Select a Demo screen displays (view [sample](#)).
4. Type the name of the demo (1–8 characters) you want to redesign in the Demo name field. To display a list of demos in the current BMS/TS library, press F2.
5. If you password-protected the demo, type the password in the Password field.

Otherwise, skip this field.

6. Press Enter. BMS/TS displays the Design a Demo screen (view [sample](#)).

Redesign (Edit) a Demo

Redesigning a demo lets you modify any or all components of the demo, edit the current list of mapsets and maps for the demo in the sequence you want them displayed. You can tell BMS/TS to clear the screen with each display, or overlay your maps, during the demo. Optionally, you can assign function keys to the maps.

After you [select](#) a demo, the Design a Demo screen appears (view [sample](#)).

1. The cursor is in the Description field. If desired, edit a description (up to 35 characters) of the demo.
2. Press Tab to move to all other fields on screen that you want to update, then edit the existing entries.

Refer to the instructions in [Design a Demo](#) for information on the fields displayed on this screen.

You can define up to 10 maps (using the ten Step No. fields) for the demo. Remember, though, that you can chain an infinite number of maps for the demo by entering the name of another demo in a Mapset Name field.

3. When you are finished editing the demo, press F3 to save the changes and return to the Select a Demo screen.

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Run a Demo](#)

Related topics:

- [Working with Demos](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Run a Demo

When you run a demo, BMS/TS displays the Select a Demo screen so that you can indicate the demo you want to run. BMS/TS then begins the demo.

Select and Run a Demo

1. From the Main Menu, select the Maps option and press Enter. The Maps Menu displays (view [sample](#)).
2. From the Maps Menu, select the Demos option and press Enter. The Demo Options Menu displays (view [sample](#)).
3. From the Demo Options Menu, select the Run option and press Enter. The Select a Demo screen displays (view [sample](#)).
4. Type the name of the demo you want to run in the Demo name field. To display a list of demos in the current BMS/TS library, press F2.
5. If you password-protected the demo, type the password in the Password field.

Otherwise, skip this field.
6. Press Enter. BMS/TS displays the first map in your demo.
7. If desired, enter data in the displayed fields and press Enter. Or, if you defined a function key for this map, press it now to jump to the specified map in the demo.

Note: If you have added input rules to the maps in your demo, BMS/TS responds to your entries according to your input rules. You must therefore enter valid data for BMS/TS to continue to the next map in the demo.
8. Repeat step 7 to move through all of the maps of your demo.
9. When you have advanced through your demo, press Clear. BMS/TS returns to the Select a Demo screen.

[Top of page](#)

Contents:

- [List Demos in the Current Library](#)
- [Field Descriptions](#)

Related topics:

- [Working with Demos](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

List Demos in the Current Library

You can view a list of all demos stored in your current CICS library.

Note: You can also use the [Listing of Members](#) screen to perform file maintenance on a table in the library.

List Demos in the Current Library

To list all current demos in the current CICS library:

1. From the Main Menu, select the Maps option and press Enter.
2. From the Maps Menu, select the Demos option and press Enter.
3. From the Demo Options Menu, select the List option and press Enter. The Listing of Members screen displays.
4. Review the [displayed information](#) (described below). You can use the [function keys](#) listed at the bottom of the screen to help you review the list of members.
5. If you want to advance the list to a specific demo, type the demo name at the command prompt.
6. Press Enter. The list scrolls to the desired demo name.
7. To exit this screen, press F3 to return to the Demo Options Menu.

The Listing of Members screen includes the fields described in the table below.

Field	Description
?	Entry field for single character command. Press F1 for details. See Review the List of Members for valid commands .
Set/Type	Always *DEMO.
Member	Demo name.
Flags	Status code(s) for the demo:

* Demo is password-protected.

P Demo is in production status (locked); cannot be updated.

U Demo is currently being updated.

Date Date the demo was last updated. By default, the date is displayed in *mmddccyy* format; if European dates are used, the date is formatted as *ddmmccyy*.

Time Time of day that the demo was last updated.

Description Description of demo as entered in Design a Demo screen.

The following fields are displayed when you press F2 for the alternate display.

OPID Terminal or operator ID (operator ID used whenever present).

SIZE *Not applicable.*

START *Not applicable.*

MAP *Not applicable.*

DSECT *Not applicable.*

You can use the function keys described in the following table to navigate the list.

Fkey	Description
F1	Display screen level help for this screen.
F2	View the alternate display for this screen. Press F2 to toggle between these two displays.
F3	Exit -- cancel this option and return to the Demo Options menu.
F7	Scroll backward one screen (if applicable).
F8	Scroll forward one screen (if applicable).

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- Working with Help Screens

Related topics:

- [Create Help Screens](#)
- [Chained Help Screens](#)
- [Branched Help Screens](#)
- [Add Help Screens to Transaction Monitor](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Working with Help Screens

Help screens are maps that display informative text to assist users with problems, error conditions, data entry, etc. You create help screens for your application in the same way that you create a map.

You can define fields in a help screen as input fields, output fields, or input/output fields (though help screens primarily contain output fields). These output fields display the help text read by your users.

After creating your help screens, there are two steps you must take to access the help information. First, you must give each application you create a CICS transaction code. Next, you make an entry in the Transaction Monitor table for each transaction that has an associated help screen. BMS/TS uses your entries in the Transaction Monitor as a reference to know which applications have help screens and when you want those help screens displayed.

Designing Help Screens

Designing help screens requires you to give some thought as to how you want the finished help screens to look, and the naming method you want to use so that BMS/TS can quickly locate the screens. Help screens can be as simple as a single displayed message, or as complex as a series of interconnected screens that provide extensive data to the user.

As you can imagine, this gives you a powerful tool for creating intelligent help displays that can make any online application more user-friendly. With BMS/TS, you can get new or inexperienced users up and running quickly by providing a wealth of valuable online information.

Display Formats

You can choose a number of ways to present a help screen to the user. The format you choose depends upon the scope and nature of the information you are presenting. BMS/TS can display help screens in one of four formats:

- | | |
|-------------------------|--|
| Single Display | Single help screen displays per transaction give you a quick and simple way to increase the effectiveness of your CICS application. BMS/TS presents the user with a single display of information. After displaying the help text, BMS/TS returns control of the program to the application. |
| Multiple Display | Multiple help screen displays per transaction (or "chained" help screens) allow you to create numerous help displays that the user can review by pressing one or more function keys. You can display both general and specific information in the help text before BMS/TS returns control of the program to the application. |

- Selection Display** Selection (or "branching") help screens that start from a menu let the user select the exact help information to review. Each selection points to an additional help display, or a group of additional help displays. The user can then access the desired help information without delay.
- Tutorials** Tutorials refer to help displays called outside the scope of a CICS transaction. Normally, the user launches a tutorial by pressing a function key that is reserved for tutorials. Tutorials can include any or all of the above formats.
-

Naming Conventions

When you create a help screen with BMS/TS, the first thing you do is define the name for the screen. Then, when the screen is designed, you also add an entry to the Transaction Monitor so that BMS/TS can locate the help screen. This entry includes the mapset and map name of the help screen.

You can name help screens for easy identification when you are ready to add them to the BMS/TS Transaction Monitor.

Help Display Related to Transaction Code

For a standard help display, define your mapset using a format of GTHxxxx, where xxxx is replaced by the appropriate CICS transaction code. Using this type of help display, you can define the map name of the help screen to anything that you wish.

For example, if you were creating help screens for an Accounts Receivable function, you might want to name the mapset GTHACCT to correspond with your ACCT transaction code. If you were creating a help screen for entry of Customer Account Adjustment information, you might name the map HELPAMT.

When using this naming convention with chained or branching help displays, you only specify the first help screen to the Transaction Monitor. BMS/TS then presents all other related help screens based on the information received from the first display.

Help Display Related to Cursor Row Position

You can name help screens by linking the name of the help screen with a particular row on the displayed screen. This allows BMS/TS to display a specific help screen depending on the position of the cursor when the user requests help.

Using this method, you should define your mapset using the format GTHxxxx, where xxxx is the appropriate CICS transaction code. The map name of the help screen is in a format of RWrr, where rr represents the appropriate row number of the cursor.

For example, a map name of RW05 causes a specific help screen to display when the cursor is anywhere in row 5.

Note: If you use this method, you must enter *ROW as the map name when you update the Transaction Monitor to include this help screen.

Help Display Related to Cursor Field Position

You can also name help screens by linking the name of the help screen with a particular field on the displayed screen. This allows a specific help screen to display depending on which field the cursor is positioned in when the user requests help.

Using this method, you should define your mapset using a format of GTHxxxx, where xxxx is replaced by the appropriate CICS transaction code. The map name of the help screen is in a format of RWrrcc, where rrcc represents the appropriate row and column number of the field.

For example, a map name of RW0580 causes a specific help screen to display when the cursor is anywhere from row 1/column 1 to row 5/column 80. Keep in mind that you should define the column number to include all possible positions of the field so that the same help screen displays regardless of where the cursor is positioned in the field.

Note: If you use this method, you must enter *ROWCOL as the map name when you update the Transaction Monitor to include this help screen.

Help Display Related to Mapset Name and Cursor Field Position

Lastly, you can name help screens by combining the actual mapset name of the displayed screen with a particular field on the displayed screen. This allows you to use information from your own application screen to easily identify the help screen.

Using this method, you should define your mapset using a format of Hxxxxxx, where xxxxxx is the mapset used in your application program. The map name of the help screen is in a format of RWrrcc, where rrcc represents the appropriate row and column number of the field.

For example, if your program's mapset is ACCT, you would use HACCT as the help screen's mapset name. Also, a map name of RW0580 causes a specific help screen to display when the cursor is anywhere from row 1/column 1 to row 5/column 80 on your screen. Keep in mind that you should define the column number to include all possible positions of the field so that the same help screen displays regardless of where the cursor is positioned in the field.

Note: If you use this method, you must enter *ROWSET as the map name when you update the Transaction Monitor to include this help screen.

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Select a Help Screen \(Map\)](#)
- [Define Help Screen \(Map\) Options](#)
- [Design a Help Screen \(Map\)](#)
- [Define Help Screen \(Map\) Fields](#)
- [Verify the Help Screen \(Map\)](#)

Related topics:

- [Working with Help Screens](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Create a Help Screen

You create a help screen in the same way that you would design a map.

You can design a basic help screen to display text, or you can create a more advanced help screen which offers the users options to display more information ([chained helps](#) or [selection helps](#)).

Select a Help Screen (Map)

Selecting a map lets you name your new help screen, and select or create a mapset name for your help screen.

1. From the Main Menu, select the Maps option and press Enter. The Maps Menu displays (view [sample](#)).
2. From the Maps Menu, select the Maps option and press Enter. The Map Options Menu displays (view [sample](#)).
3. From the Map Options Menu, select the Design option and press Enter. The Select a Map screen displays (view [sample](#)).
4. Type a mapset name (1–7 characters) in the Mapset field. To display a list of mapsets in the current BMS/TS library, press F2.
5. Type a map name (1–7 characters) in the Map name field. To display a list of maps in the current BMS/TS library, press F2.
6. If you want to protect the help screen, type a password in the Password field. Otherwise, skip this field.

Remember that if you assign a password to your help screen, you must enter the password each time you edit the help screen.

7. Press Enter. BMS/TS displays the Map Definition Options screen (view [sample](#)).

Define Help Screen (Map) Options

Defining map options lets you specify an individual help screen's characteristics.

After you [select](#) a help screen, the Map Definition Options screen appears (view [sample](#)).

1. Press Tab to move to the entry fields you want to modify, then type over the currently displayed value.
 2. Repeat the previous step until you make all of your modifications.
 3. Press Enter. BMS/TS displays the Design screen (view [sample](#)).
-

Design a Help Screen (Map)

Designing a map lets you design the physical help screen, and indicate the start of input and/or output fields.

After you complete the Map Definition Options screen, the Design screen appears (view [sample](#)).

1. Press ErEOF to remove the displayed instructions and list of function key assignments.
2. Use the Arrow keys to move the cursor to the location on screen where you want to begin designing your help screen.
3. Type the help text (view [sample](#)).

Note: Change Field Characters indicate input or output fields on the help screen. The default Change Field Character is a plus sign (+). Type a Change Field Character to mark the first position of each input or output field.

4. Place your cursor where you want it positioned when the help screen appears to the end-user.
5. Press Enter. BMS/TS displays the Field Definition screen for the first field marked with a Change Field Character (view [sample](#)).

Note: BMS/TS skips the Field Definition screen if your help screen contains no Change Field Characters. Remember, Change Field Characters indicate input or output fields on screen. If a help screen contains no Change Field Characters — that is, if the help screen simply displays information — then there is no need for you to define the characteristics of the displayed fields.

You can use the following function keys to design your help screen:

Function Key	Description
F1	Help. Displays help text for a specific field. Press F1 again for help concerning the screen you are using.
F2	Dup Line. Duplicates the line where the cursor is positioned.

- F3** Skip Fld. Assigns default values to all fields and advances directly to the Map Verification screen.
 - F4** Options. Returns to the Map Definition Options screen.
 - F5** Check. Tells BMS/TS that you have modified current fields (usually during an update) using Insert and/or Delete. For this reason, BMS/TS then realigns the symbolic and physical maps so that no information is lost when you exit the screen.
 - F6** Hold. Saves your work up to this point.
 - F7** Del line. Deletes the line where the cursor is positioned.
 - F8** Undo. Refreshes the screen from the previously saved image.
 - F9** Add Line. Inserts a blank line where the cursor is positioned.
 - F10** Action. Goes to the [Copy/Move](#) functions screen.
-

Define Help Screen (Map) Fields

Defining fields lets you select various field characteristics and attributes for your help screen. (Remember: BMS/TS skips the Field Definition screen if your help screen contains no Change Field Characters.)

After you complete the Design screen, the Field Definition screen appears (view [sample](#)).

Note: BMS/TS displays an altered version of the Field Definition screen if you use the Assembler language; the Edit Pattern fields do not display, and the name field is 8 bytes long.

The top section of the Field Definition screen contains the help screen you have created, with the first field highlighted. BMS/TS places a "?" at the highlighted field to identify the field you are currently defining.

1. Type a descriptive name in the field name field.
2. Press Tab to move to the parameters you want to select or change, then type over the currently displayed value.
3. After you define all of the parameters for this field, press Enter. BMS/TS moves to the next field containing a Change Field Character -- marking it with a "?" -- so you can begin defining its parameters.
4. Repeat steps 1–3 until you have defined the parameters for all of the fields on this help screen.
5. Press Enter. BMS/TS displays the Map Verification screen.

Note: BMS/TS displays the Input Rule Editor screen when you press Enter if you selected Y (Yes) for input validation rules on the Map Definition Options screen.

Verify the Help Screen (Map)

Verifying a map lets you display your newly created help screen as it will appear in your application. You can then visually verify the layout of your help screen, as well as type in sample data to test edit patterns and data type restrictions.

After you complete the Field Definition screen, the Map Verification screen appears (view [sample](#)).

Note: When present, protected fields appear with a V at every fifth byte, a number at every tenth byte and four asterisks between, for the length of your defined field:
(****V****1****V****2 etc.)

1. Review the displayed data to verify the appearance of the help screen.
2. If desired, you can enter sample data into the fields displayed on the screen.

Note: BMS/TS does not validate the sample data unless you have written and implemented input rules for the map. BMS/TS limits only the length of the field according to the definition you have specified.

3. After keying sample data into all desired fields, press Enter. BMS/TS displays the sample data as it would appear when your map is being used.
4. Review the displayed data to verify the appearance of the help screen. At this point, you have the following options:
 - ◆ If you want to save the help screen, continue at step 5.
 - ◆ If the help screen's appearance is acceptable but you want to reposition the cursor on screen, continue at step 6.
 - ◆ If you want to display all spaces you placed in the help screen as question marks (?), continue at step 7.
 - ◆ If you want to correct a field prompt, label, or change a field definition, continue at step 8.
5. If the help screen's appearance is acceptable, press Enter. BMS/TS saves the help screen and redisplay the Select a Map screen.

Note: BMS/TS displays the Input Rule Editor screen when you press Enter if you selected Y (Yes) for input validation rules on the Map Definition Options screen.

6. If the help screen's appearance is acceptable but you want to reposition the cursor on screen, press F2, then place your cursor where you want it positioned for the end-user when the help screen appears. Press F2 again to accept this cursor position, then continue at step 5.
7. If you want to display all spaces you placed in the help screen as question marks (?), press F5. To redisplay the screen without the question marks, press F5 again. Then continue at step 5.

Note: This shows you which bytes of the help screen are true spaces (hexadecimal 40) instead of nulls (hexadecimal/binary 00).

8. If you want to correct a field prompt, label, or change a field definition, press F4. The Map Verification screen redisplay.

Notice that any fields on the help screen are now preceded with an @. The @ character is the

Start Field Character. It precedes all input and/or output fields.

9. If you want to correct a literal, perform the following steps. If you want to correct a field definition, continue at step 10.
 - a. Use the Arrow keys to move the cursor to the field prompt or label you want to change.
 - b. Enter the correct information over the existing literal.
 - c. Press Enter. BMS/TS redisplays the Map Verification screen with the new information.
 - d. Repeat steps 1–5 to verify the help screen with the new information.

10. To correct a field definition, perform the following steps:
 - a. Press Tab to move to the field you want to redefine.
 - b. Type a Change Field Character (default is +) after all of the fields you want to change, then press Enter.
 - c. BMS/TS displays the Field Definition screen for each field marked with a Change Field Character. (Keep in mind that the displayed fields are still preceded by @ characters.)
 - d. After redefining all desired fields, press Enter. BMS/TS returns to the Map Verification screen. Repeat steps 1–5 to verify the help screen with the new information.

After you create the help screen, you need to tell BMS/TS when to display it by [adding an entry to the Transaction Monitor](#).

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Create the First Help Screen](#)
- [Create Additional Help Screens](#)
- [Create the Return Help Screen](#)

Related topics:

- [Working with Help Screens](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Create Chained Help Screens

You can allow a help screen to call another help screen by defining input/output fields. That is, by reading data sent from a help screen as a map name, BMS/TS can look in the current library for a help screen with a matching map name, then retrieve and display the next help screen.

You create chained help screens in the same way that you design a single help screen. However, you must also define a **Call field** on each screen. You place this output only field in the upper left corner of the map so that the currently displayed help screen can call another help screen.

Chained help screens use the same mapset name, but individual screens have separate map names. You create the initial help screen, any intermediate screens, and a return screen that calls the initial help screen and closes the chain.

Create the First Help Screen

You will create the first help screen in the chain, following the instructions for [creating a help screen](#).

Remember that you must define a Call field in the upper left-hand corner of the screen which specifies the name of the next help screen in the chain. Define the Call field with these attributes:

Parameter	Setting	Description
Field name	1–29 characters	Description of field. This field is required, although you can enter any field name you wish. Note: Length of this field varies according to your programming language.
Field intensity	D (Dark)	Indicates brightness of the displayed field. Because the output field is used to display another help screen and is not needed by the end-user, set this field to D (dark) so it does not display.
Field type	A (Askip)	Indicates whether user can enter data into this field. By setting this field to A (automatic skip), BMS/TS automatically skips this field and moves to the next unprotected field when the help screen displays.
Field size	1–8	Indicates the length of the next help screen's name (1–8 characters). That is, the name of the current Call field tells BMS/TS the name of the help screen that should

Mod. data tag Y (Yes)

display next.

Forces the terminal to return a field to BMS/TS, even if the field does not contain data, so the Call field name can represent entered data.

Be sure that the help map tells users to press Enter to display the next screen in the chain.

Create Additional Help Screens

Next you create the intermediate help screens in a series of chained screens, following the instructions to [create a help screen](#). Remember, your chained help screens use the same mapset name, but individual screens have unique map names.

When creating your additional help screens, you must define the Call field with the [attributes](#) noted above. Be sure that each map tells users to press Enter to display the next screen in the chain.

Create the Return Help Screen

The last screen in a series of chained helps calls the first help screen, which closes the chain.

[Create the return help screen](#). Remember that chained help screens use the same mapset name, but individual screens have separate map names.

The [attributes](#) of the Call field on the return screen are the same as the other help screens.

Be sure that the return map tells users to press Enter to close the chain.

Note: After you have created the help screen chain, you need to tell BMS/TS when to display it. See [Add a Help Screen to the Transaction Monitor](#) for more information.

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Create the Selection Screen](#)
- [Create Additional Help Screens](#)
- [Create the Return Help Screen](#)

Related topics:

- [Working with Help Screens](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Create Selection (Branched) Help Screens

[Chained help screens](#) allow users to view a pre-set sequence of helps. Another option is to offer a selection screen, which enables the user to *choose* the help screens to view.

A selection help screen is the same as any other help screen, but with two additions: a call field and a selection field.

- **Call Field**
Place this field in the upper left corner of the map to identify the first part of a help screen's map name. BMS/TS combines the call field with the selection field to display another help screen.
- **Selection Field**
Place this field in a selection list in the body of the actual help screen. It identifies the second part of a help screen's map name. BMS/TS combines the selection field with the call field to display another help screen.

You will create the initial (selection) help screen, the associated intermediate screens, and a return screen that calls the initial help screen and closes the chain. Remember that selection (branching) help screens use the same mapset name, but each screen has a unique map name.

Create the Selection Screen

First, you [create the initial help screen](#) which offers the selections to the user.

You must define the [Call field](#) with the following attributes:

Parameter	Setting	Description
Field name	1–29 characters	Description of field. This field is required, although you can enter any field name you wish. Note: Length of this field varies according to your programming language.
Field intensity	D (Dark)	Indicates brightness of the displayed field. Because the CALL field simply enables BMS/TS to display another help screen and therefore is of no use to the end-user, set this field to D (dark) so it

does not display.

Field type	A (Askip)	Indicates whether user can enter data into this field. By setting this field to A (automatic skip), BMS/TS automatically skips this field and moves to the next unprotected field when the help screen displays.
Field size	1–8	Indicates the length of the next help screen's name (1–8 characters). That is, the name of the current Call field tells BMS/TS the name of the help screen that should display next.
Mod. data tag	Y (Yes)	Forces the terminal to return a field to BMS/TS, even if the field does not contain data. This allows the Call field name to represent entered data.

You must define the [Selection field](#) with the following attributes:

Parameter	Setting	Description
Field name	1–29 characters	Description of field. This field is required, although you can enter any field name you wish. Note: Length of this field varies according to your programming language.
Field type	U (Unprotected)	Indicates whether user can enter data into this field. By setting this field to U (unprotected), the user can enter data into this field when the help screen displays.
Field size	1–8	Indicates the maximum number of characters (up to 8) the user can enter at this field. BMS/TS combines the current Selection field with the Call field to find the help screen that should display next.
Mod. Data tag	Y (Yes)	Forces the terminal to return a field to BMS/TS, even if the field does not contain data. This allows the output field name to represent entered data.

Note: Be sure that the help text tells users to how to select another help for display.

Create Additional Help Screens

Next you create the intermediate help screens — the screens the user can select from the initial help screen. Remember, each of these screens must the same mapset name, but individual screens have unique map names.

When creating your additional help screens, you must define the Call field with the [attributes](#) noted above. Be sure that the text of each help map tells users to how proceed.

Create the Return Help Screen

Finally, you create the last (return) help screen. This help screen calls the first help screen, which then closes the chain.

When creating the return help screen, remember that the Call field must appear on screen as the name of your first help screen (in order to close the chain).

Note: After you have created the help screen chain, you need to tell BMS/TS when to display it. See [Add a Help Screen to the Transaction Monitor](#) for more information.

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Select a Transaction Monitor](#)
- [Edit the Transaction Monitor](#)
- [Reload Transaction Monitor](#)

Related topics:

- [Working with Help Screens](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Add a Help Screen to the Transaction Monitor

You will add an entry to the BMS/TS Transaction Monitor to identify all maps in your application program for which you have designed help screens. After you make this entry, BMS/TS displays the proper help screens for a given map (based on user input) before passing the map's data to your application program.

To add a help screen to the Transaction Monitor, you must also replace your program name in the Program Control Table (PCT) with the BMS/TS program name, GTBTHLP.

If you need assistance, contact your Product Administrator.

Select a Transaction Monitor

You can select the Transaction Monitor to which you want to add the entry so that BMS/TS can locate the help screen.

1. From the Main Menu, select the System option and press Enter. The System Functions Menu displays (view [sample](#)).
2. From the System Functions Menu, select the Transaction Monitor option and press Enter. The Transaction Monitor Options Menu displays (view [sample](#)).
3. Select the Edit Monitor option and press Enter. The Select a Transaction Monitor screen displays (view [sample](#)).
4. Type the name of the Transaction Monitor to which you want to add an entry in the Transaction Monitor Name field.

Remember, in almost all cases the Transaction Monitor you will use is called **Monitor**, which is the default entry.

5. If you password-protected the Transaction Monitor, type the password in the Password field. Otherwise skip this field.
 6. Press Enter. BMS/TS displays the Transaction Monitor Control screen (view [sample](#)).
-

Edit the Transaction Monitor

Editing the Transaction Monitor lets you add an entry that defines the transactions which BMS/TS should monitor for help screens. You can also use this option to modify an existing entry in the Transaction Monitor.

After you [select](#) a Transaction Monitor, the Transaction Monitor Control screen appears (view [sample](#)).

1. Press Tab to move to any LC (Line Command) field.
2. Type I (Insert) and the number of lines you want to insert.

Note: BMS/TS automatically sorts all entries when you exit the Transaction Monitor. This means that you can enter your information in any order using a line command to insert blank lines. See Appendix A, [Using the Line Editor](#), for a description of the line commands you can use for this screen.

3. Press Enter. BMS/TS inserts the desired number of lines, then moves the cursor to the Tran field of the first blank line.
4. Type the transaction code (as it appears on the PCT) that you want BMS/TS to monitor for help screens.

Note: For [chained](#) and [selection](#) (branching) help screens, the only help screen you need to add to the Transaction Monitor is the first one in the series.

5. Press Tab to move to the Fkey field, then type the appropriate function key number that calls the help screen. When this key is pressed, BMS/TS displays the appropriate help screens for the maps associated with the transaction code you entered in step 4.
6. Press Tab to move to the File field, then type the prefix of the library where the transaction is stored. You can also leave this field blank to accept the default, **GT**.
7. Press Tab to move to the Set/Type field, then type the mapset name of the help screen.
8. Press Tab to move to the Member field, then type the map name of the help screen.

If you are using a help screen naming convention, you must enter a specific variable in the Member field:

If you want to:

Enter this variable:

Display help related to cursor row position

*ROW

Display help related to cursor field position

*ROWCOL

Display help related to mapset name and cursor field position

*ROWSET

9. Press Tab to move to the Description field, then type any comments you want regarding the entry.
10. Repeat steps 4–9 until you have completed your entries.

11. Press F3. BMS/TS displays the message SORT Processing. BMS/TS then clears the screen and displays a screen prompting you to load the Transaction Monitor.
 12. Press Enter if you want to load the Transaction Monitor now. Press Clear to exit this screen and return to the Select a Transaction Monitor screen.
 - If you press Enter, BMS/TS loads the Transaction Monitor (which immediately implements the help screens), then returns to the Select a Transaction Monitor screen.
 - If you press Clear, you'll need to [reload the Transaction Monitor table](#).
-

Reload Transaction Monitor

You can reload the Transaction Monitor with your updates.

1. From the Transaction Monitor Options Menu, select the Refresh Monitor option and press Enter. The Reload Monitor screen displays (view [sample](#)).
2. Press Enter to reload the Transaction Monitor.

Note: BMS/TS can only load a Transaction Monitor into CICS memory if the monitor has a set name of *SYSTEM and a member name of MONITOR. You can create a monitor table with another name, but you must rename it *SYSTEM/MONITOR or merge it with *SYSTEM/MONITOR before you can load it.

BMS/TS reloads the Transaction Monitor, then displays the message System monitor initialization complete.

3. Press Enter. BMS/TS returns to the Select a Transaction Monitor screen.
-

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Special Sets](#)
- [Library Functions](#)

Related topics:

- [Manage Members in a Library](#)
- [Transfer Members Between Libraries](#)
- [Link Libraries](#)
- [List Libraries](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Performing Library Functions

A library is a collection of programs or data files (known as "members") grouped for specific purposes. Regardless of how your system is set up, you will have at least one CICS library and one BMS/TS library. In most cases, you will have multiple CICS and BMS/TS libraries.

BMS/TS uses a two-tier identification system to organize its libraries. You distinguish each library member using a set name and a member name. In general, you can define both names in any way you wish, although member names must be unique. After you name the members, you can group them together into sets.

For example, if you were designing a group of maps related to an Accounts Receivable function, you might name the mapset CUSTACC, then assign each related map a different name: SCREEN1, SCREEN2, etc.

Special Sets

Although you can assign set names for maps, BMS/TS uses predefined mapset names for other members. These members are known as special sets. The following table describes these special sets:

Special Set Name	Description
*ADMIN	Contains administration records created when you install BMS/TS.
*DEMO	Contains demonstrations created with BMS/TS.
*JCL	Contains JCL that you write and store in GT libraries to submit your tables and maps.
*MAPS	Contains all maps you create with BMS/TS. Note: Although you can assign names for mapsets, BMS/TS stores all maps under this special set name so that you can view a directory list of all mapsets within a library.
*OPID	Contains security records for users of BMS/TS.
*SYSTEM	Contains Transaction Monitor table records.
*TABLE	Contains tables of data created with BMS/TS.

Library Functions

You have the following options for managing, transferring, and listing library members with BMS/TS:

Manage Library Members

Perform various options, including:

- Copy, display, erase, or rename existing library members.
- Reset the flag of an existing set and/or member which tells BMS/TS that the set and/or member is currently being updated.
- Add, change, or delete passwords from existing sets and members.
- Link to a region of another library.

Transfer Between Libraries

Copy, move, or retrieve a member from one library to another.

Link Libraries

Link to a region in another library without leaving your current library.

List Library Members

Review a special set's contents, as well as perform maintenance on sets and/or members.

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Copy a Library Member](#)
- [Display a Member](#)
- [Erase a Library Member](#)
- [Rename a Library Member](#)
- [Reset a Library Member](#)
- [Add, Change, or Delete Password](#)

Related topics:

- [Performing Library Functions](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Manage Library Members

You can perform file management functions within a library.

Note: The BMS/TS security options control the availability of the above maintenance functions. As a result, you may not have access to all of these functions.

For assistance, contact your Product Administrator.

Copy a Library Member

Copying a library member lets you make a duplicate of the original set and/or member, giving a new unique name to the duplicate. You can copy:

- A complete mapset,
- A single map,
- A demo,
- A JCL submit member,
- A system record,
- A table,
- An operator ID record.

Perform the following steps to copy a library member.

1. From the Main Menu, select the Library option and press Enter. The Library Functions Menu displays (view [sample](#)).
2. From the Library Functions Menu, select the Manage Library Members option and press Enter. The Manage Library Members screen displays (view [sample](#)).
3. Type the name of the existing set in the Set Name field. To display a list of sets in the current BMS/TS library, press F2.

4. Type the name of the existing member in the Member Name field. To display a list of members in the current BMS/TS library, press F2.
Note: If you are copying a complete mapset, type an asterisk (*) in this field.
 5. If you password-protected the set or member, type the password in the Password field. Otherwise, skip this field.
 6. Type the name of the new set (1–8 characters) in the New Set Name field.
Note: If you are copying a map, you can leave this field blank and BMS/TS will automatically use the existing mapset name.
 7. Type the name of the new member (1–8 characters) in the New Member Name field.
Note: If you are copying a complete mapset, type an asterisk (*) in this field.
 8. If you want to add or modify a password for the new set or member, type the new password in the New Password field. Otherwise, skip this field.
Note: If the existing set or member has a password and you do not enter a password in this field, BMS/TS automatically assigns the original password to the new set or member.
 9. Press F5 (or type COPY at the Command line and press Enter). BMS/TS copies the desired set or member, then displays the message Operation Complete.
-

Display a Member

Displaying a library member lets you review a map or help screen exactly as it will appear to the end-user. Perform the following steps to display a library member.

1. From the Main Menu, select the Library option and press Enter. The Library Functions Menu displays (view [sample](#)).
 2. From the Library Functions Menu, select the Manage Library Members option and press Enter. The Manage Library Members screen displays (view [sample](#)).
 3. Type the name of the existing set in the Set Name field. To display a list of sets in the current BMS/TS library, press F2.
 4. Type the name of the member that you want to display in the Member Name field. To display a list of members in the current BMS/TS library, press F2.
 5. If you protected the set or member with a password, type the password in the Password field. Otherwise, skip this field.
 6. Press Tab to move to the Command line.
 7. Type DISPLAY at the Command line and press Enter. BMS/TS displays the map or help screen as it will appear to the end-user.
Note: See [Display a Map](#) for more information.
 8. Review the member, then press F3 to return to the Manage Library Members screen.
-

Erase a Library Member

Erasing a library member lets you remove a set and/or member from the current library. You can erase:

- A complete mapset,
- A single map,
- A demo,
- A JCL submit member,
- A system record,
- A table,
- An operator ID record.

Perform the following steps to erase a library member.

1. From the Main Menu, select the Library option and press Enter. The Library Functions Menu displays (view [sample](#)).
 2. From the Library Functions Menu, select the Manage Library Members option and press Enter. The Manage Library Members screen displays (view [sample](#)).
 3. Type the name of the existing set in the Set Name field. To display a list of sets in the current BMS/TS library, press F2.
 4. Type the name of the existing member in the Member Name field. To display a list of members in the current BMS/TS library, press F2.
Note: If you are erasing a complete mapset, type an asterisk (*) in this field.
 5. If you password-protected the set or member, type the password in the Password field. Otherwise skip this field.
 6. Press F6 (or type ERASE at the Command line and press Enter). BMS/TS erases the set or member, then displays the message Operation Complete.
-

Rename a Library Member

Renaming a library member lets you change the name of a set and/or member without changing the actual set and/or member. You can rename:

- A complete mapset,
- A single map,
- A demo,
- A JCL submit member,
- A system record,
- A table,
- An operator ID record.

Perform the following steps to rename a library member.

1. From the Main Menu, select the Library option and press Enter. The Library Functions Menu displays (view [sample](#)).

2. From the Library Functions Menu, select the Manage Library Members option and press Enter. The Manage Library Members screen displays (view [sample](#)).
 3. Type the name of the existing set in the Set Name field. To display a list of sets in the current BMS/TS library, press F2.
 4. Type the name of the existing member in the Member Name field. To display a list of members in the current BMS/TS library, press F2.
Note: If you are renaming a complete mapset, type an asterisk (*) in this field.
 5. If you password-protected the set or member, type the password in the Password field. Otherwise skip this field.
 6. Type the name of the new set (1–8 characters) in the New Set Name field. If you are renaming a map, you can leave this field blank and BMS/TS will automatically use the existing mapset name.
 7. Type the name of the new member (1–8 characters) in the New Member Name field.
Note: If you are renaming a complete mapset, type an asterisk (*) in this field.
 8. If you want to add or modify a password for the new set or member, type the new password in the New Password field. Otherwise, skip this field.
Note: If the existing set or member has a password and you do not enter a password in this field, BMS/TS automatically assigns the original password to the new set or member.
 9. Press F7 (or type RENAME at the Command line and press Enter). BMS/TS renames the desired set or member, then displays the message Operation Complete.
-

Reset a Library Member

Resetting a library member lets you remove the flag of an existing set and/or member which tells BMS/TS that the set and/or member is currently being updated.

If the CICS system abnormally terminates when you are using BMS/TS, a library member may be flagged as being in an update mode. You can remove the update flag for any of the following:

- A single map,
- A demo,
- A JCL submit member,
- A system record,
- A table,
- An operator ID record.

Perform the following steps to reset a library member.

1. From the Main Menu, select the Library option and press Enter. The Library Functions Menu displays (view [sample](#)).
2. From the Library Functions Menu, select the Manage Library Members option and press Enter. The Manage Library Members screen displays (view [sample](#)).

3. Type the name of the existing set in the Set Name field. To display a list of sets in the current BMS/TS library, press F2.
 4. Type the name of the existing member in the Member Name field. To display a list of members in the current BMS/TS library, press F2.
 5. If you password-protected the member, type the password in the Password field. Otherwise, skip this field.
 6. Press F9 (or type RESET at the Command line and press Enter). BMS/TS resets the member, then displays the message Operation Complete.
-

Add, Change, or Delete Password

You can add, change, or delete a password for an existing set and/or member for any of the following:

- A complete mapset,
- A single map,
- A demo,
- A JCL submit member,
- A system record,
- A table,
- An operator ID record.

Perform the following steps to add, change, or delete a password.

1. From the Main Menu, select the Library option and press Enter. The Library Functions Menu displays (view [sample](#)).
2. From the Library Functions Menu, select the Manage Library Members option and press Enter. The Manage Library Members screen displays (view [sample](#)).
3. Type the name of the existing set in the Set Name field.
Note: To display a list of sets in the current BMS/TS library, press F2.
4. Type the name of the existing member in the Member Name field. To display a list of members in the current BMS/TS library, press F2.
5. If the set or member is password-protected, type the password in the Password field. Otherwise, skip this field.
6. Press Tab to skip the New Set Name and New Member Name fields. You do not use these fields for password functions.
7. If you want to add or modify a password for the new set or member, type the new password in the New Password field.

If you want to delete the password, type NONE in this field.

8. Press F8 (or type PASSWORD on the Command line and press Enter.) BMS/TS adds, changes, or deletes the password, then displays the Operation Complete message.

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Copy Member](#)
- [Move Member](#)
- [Fetch Member](#)

Related topics:

- [Performing Library Functions](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Transfer Members Between Libraries

You can perform the following functions to transfer members between libraries:

- **Copy** -- make a copy of a set or member and place it in another library.
- **Move** -- move a member to another library.
- **Fetch** -- retrieve a BMS map from a system library and place it in another library for updating with BMS/TS.

Note: The BMS/TS security options control the availability of the above maintenance functions. As a result, you may not have access to all of these functions. For assistance, contact your Product Administrator.

Copy Library Member

Copying a library member lets you take an entire mapset, or a single member of a special set, and make a duplicate copy in another CICS or BMS/TS library.

1. From the Main Menu, select the Library option and press Enter to display the Library Functions Menu (view [sample](#)).
2. From the Library Functions Menu, select the Transfer Between Libraries option and press Enter. The Select a Set or Member screen displays (view [sample](#)).
3. If you are copying an entire mapset, type the name of the existing mapset in the Set Name field. To display a list of mapsets in the current BMS/TS library, press F2.

If you are copying a single member of a [special set](#), type the name of the special set in this field.

Note: You can only copy single members from the following special sets: *DEMO, *JCL, *OPID, *SYSTEM, *TABLE.

4. If you are copying an entire mapset, press Tab to skip the Member Name field.

If you are copying a single member of a special set, type the name of the existing member in this field. To display a list of members in the current BMS/TS library, press F2.

5. If you protected the set or member with a password, type the password in the Password field. Otherwise skip this field.

6. Press Enter. BMS/TS displays the Transfer a Member or Set screen (view [sample](#)). Notice that the Current Library field displays for information only. It lists the library in which you are currently working.
 7. Type C (Copy) in the Transfer type field.
Note: If you are copying a single member of a special set, BMS/TS automatically displays the special set name in the Special set name field.
 8. In the Set/Member name field, BMS/TS automatically displays the mapset or special set member you previously entered. If the displayed information is correct, press Tab to skip this field. Otherwise, enter the correct mapset or special set member name (1–8 characters).
 9. If you want BMS/TS to copy over a member in the receiving library with a matching member from the sending library, type Y (Yes) in the Replace field.

Otherwise, press Tab to accept the default (No) and skip this field.
 10. Type the 2–letter prefix of the sending library in the FROM Library prefix field.
 11. Type the 2–letter prefix of the receiving library in the TO Library prefix field, or simply press Tab to accept the default (**GT**) and skip this field.
 12. Press Enter. BMS/TS copies the desired set or member, then displays the Operation Complete message.
-

Move Library Member

Moving a library member lets you transfer a member from one BMS/TS or CICS library to another.

1. From the Main Menu, select the Library option and press Enter to display the Library Functions Menu (view [sample](#)).
2. From the Library Functions Menu, select the Transfer Between Libraries option and press Enter. The Select a Set or Member screen displays (view [sample](#)).
3. If you are moving an entire mapset, type the name of the existing mapset in the Set Name field. To display a list of mapsets in the current BMS/TS library, press F2.

If you are moving a single member of a [special set](#), type the name of the special set in this field.

Note: You can only move single members from the following special sets: *DEMO, *JCL, *OPID, *SYSTEM, *TABLE.

4. If you are moving an entire mapset, press Tab to skip the Member Name field.

If you are moving a single member of a special set, type the name of the existing member in this field. To display a list of members in the current BMS/TS library, press F2.

5. If you password–protected the set or member, type the password in the Password field. Otherwise, skip this field.

6. Press Enter. BMS/TS displays the Transfer a Member or Set screen. Notice that the Current Library field displays for information purposes only. It simply lists the library in which you are currently working.
7. Type M (Move) in the Transfer type field.

Note: If you are moving a single member of a special set, BMS/TS automatically displays the special set name in the Special set name field.

8. In the Set/Member name field, BMS/TS automatically displays the mapset or special set member you previously entered. If the displayed information is correct, press Tab to skip this field. Otherwise, enter the correct mapset or special set member name (1–8 characters).
9. If you want BMS/TS to copy over a member in the receiving library with a matching member from the sending library, type Y (Yes) in the Replace field.

Otherwise, press Tab to accept the default (**No**) and skip this field.

10. Type the 2–letter prefix of the sending library in the FROM Library prefix field.
 11. Type the 2–letter prefix of the receiving library in the TO Library prefix field, or simply press Tab to accept the default (GT) and skip this field.
 12. Press Enter. BMS/TS moves the desired set or member, then displays the Operation Complete message.
-

Fetch BMS Map

Fetching a BMS map lets you retrieve a BMS map (physical map only) from a system library and place it in another library. You can then update the map just as if it had been created with BMS/TS.

1. From the Main Menu, select the Library option and press Enter to display the Library Functions Menu (view [sample](#)).
2. From the Library Functions Menu, select the Transfer Between Libraries option and press Enter. The Select a Set or Member screen displays (view [sample](#)).
3. Type the name of the desired mapset in the Set Name field. To display a list of mapsets in the current BMS/TS library, press F2.
4. Type the name of the desired BMS map in the Member Name field. To display a list of members in the current BMS/TS library, press F2.
5. If you password–protected the BMS map, type the password in the Password field. Otherwise skip this field.
6. Press Enter. BMS/TS displays the Transfer a Member or Set screen. Notice that the Current Library field displays for information purposes only. It simply lists the library in which you are currently working.
7. Type F (Fetch) in the Transfer type field.

8. In the Set/Member name field, BMS/TS automatically displays the mapset you previously entered. If the displayed information is correct, press Tab to skip this field. Otherwise, enter the correct mapset name (up to 8 characters).
 9. If you want BMS/TS to copy over a member in the receiving library with a matching BMS map from the sending library, type Y (Yes) in the Replace field.

Otherwise, press Tab to accept the default (**No**) and skip this field.
 10. Type the 2–letter prefix of the sending library in the FROM Library prefix field.
 11. Type the 2–letter prefix of the receiving library in the TO Library prefix field, or simply press Tab to accept the default (**GT**) and skip this field.
 12. Press Enter. BMS/TS retrieves the desired BMS map, then displays the Operation Complete message.
-

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- Link Libraries

Related topics:

- [Performing Library Functions](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Link Libraries

You can use a region in another library without leaving your current library.

The BMS/TS Product Administrator (or someone with the equivalent security rights) assigns a library to be the current library. At times, however, you may need to work on sets or members stored in another library. While you cannot assign yourself a new current library, you can access a section of another library using the BMS/TS link function.

Each BMS/TS library actually consists of a number of sections, each of which holds members of a different special set. The link function essentially swaps a section of your current library with the corresponding section of another library. For example, if you need to work on a demo stored in another library, you can swap the demo section of your current library for the demo section of the other library. After you have established the link, you can update, run, or even create a demo.

Keep in mind that the link is only temporary: it disappears when you exit BMS/TS.

Note: BMS/TS security options control the availability of the above function. As a result, you may not have access to this function. For assistance, contact your Product Administrator.

Link a Library

1. From the Main Menu, select the Library option and press Enter. The Library Functions Menu displays (view [sample](#)).
2. From the Library Functions Menu, select the Manage Library Members option and press Enter. The Manage Library Members screen appears (view [sample](#)).
3. Type the name of the [special set](#) in the Set Name field. You can link to any one of the following special sets in the other library: *DEMO, *JCL, *MAPS, *OPID, *SYSTEM, *TABLE.
4. Type the 2-letter prefix of the library you want to link to in the Member Name field.
5. Press F10 (or type LINK at the Command line and press Enter). BMS/TS links to the desired library, then displays the Operation Complete message.

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Define the Directory List](#)
- [Review the List of Members](#)
- [Modify a Member in the List](#)

Related topics:

- [Performing Library Functions](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

List Members in a Library

You can view a list of all members stored in a library.

You can also perform certain file maintenance functions on list members.

Define the Directory List

You can define where the directory list begins within the current library.

1. From the Main Menu, select the Library option and press Enter. The Library Functions Menu displays.
 2. From the Library Functions Menu, select the List Members option and press Enter. The General Directory List screen appears (view [sample](#)).
 3. Specify where to begin the directory list:
 - ◆ To display a specific mapset or special set, type the name of the set (1–8 characters) in the Set Name field, and then continue with step 4. You can review a list of members for the following special sets: *DEMO, *JCL, *MAPS, *OPID, *SYSTEM, and *TABLE.
 - ◆ To begin the display with a specific alphabetic character(s), type the appropriate alphabetic character(s) in the Set Name field, and then continue with step 5. BMS/TS defaults to the *MAPS special set when you use this option.
 - ◆ To display the beginning of the mapset directory in the current library, continue with step 5.
 4. If you want the list to begin at a specific member name within the set, type the name of the member (1–8 characters) in the Member Name field. Otherwise, skip this field.
 5. Press Enter. BMS/TS displays the Listing of Members screen (view [sample](#)).
-

Review the List of Members

After you [define a directory list](#), the Listing of Members screen displays (view [sample](#)).

Note that the screen will display members based on what you specified on the General Directory List screen, such as a list of administration records, demonstrations, JCL, security records, Transaction Monitor records, or tables of data contained in the current library.

1. Review the [displayed information](#) (described below). You can use the [function keys](#) listed at the bottom of the screen to help you review the list of members.
2. If you want to advance the list to a specific member within the listing, type the member name at the command prompt and press Enter. The list scrolls to the desired member name.

You can perform certain [file maintenance functions](#) on list members. Or, press F3 to exit this screen.

The Listing of Members screen contains the following columns:

Field	Description
?	Entry field for single character command. See below for valid commands .
Set/Type	The 1–8 character mapset name. The current set name or special set type. Special set types include: *DEMO, *JCL, *MAPS, *OPID, *SYSTEM, and *TABLE.
Member	Map name. Name of the map, administration record, demonstration, JCL, security record, Transaction Monitor record, or table.
Flags	Status code(s) for map: <ul style="list-style-type: none">* Map is password-protected.P Map is in production status (locked); cannot be updated.U Map is currently being updated.C,A,P,R Language indicator: C (COBOL), A (Assembler), P (PL/1), R (RPG).E Map has extended attributes (not SOSI).S Map has extended attributes with SOSI.H,T Map is header (H) or trailer (T) of page-built screen.F,L Map is justified as first or last on a page-built screen.V Map has input validation rules.
Date	Date the map was last updated. By default, the date is displayed in <i>mmddccyy</i> format; if European dates are used, the date is formatted as <i>ddmmccyy</i> .
Time	Time of day that map was last updated.
Description	Comments on member as entered in Map Definition Options screen.

The following fields are displayed when you press F2 for the alternate display.

OPID	Terminal or operator ID (operator ID used whenever present).
SIZE	Size of map in row/column format.
START	Starting position of map on screen in row/column format. Or, may be described using the codes: N (next row or column) or S (same row or column).
MAP	Length of the physical map.
DSECT	Length of the symbolic map.

You can use the function keys described in the following table to navigate the list.

Fkey	Description
F1	Display screen level help for this screen.
F2	View the alternate display for this screen. Press F2 to toggle between these two displays.
F3	Exit — cancel this option and return to the General Directory List screen.
F7	Scroll backward one screen (if applicable).
F8	Scroll forward one screen (if applicable).

Note: After you reach the end of a list of members by scrolling forward, BMS/TS starts over at the beginning of the list.

Modify a Member in the List

You can perform file maintenance functions on the members listed on your screen.

1. Select the desired member by pressing Tab to move to the ? column of the member.
2. Identify the function you want to perform:

Type:	If you want to:
C	Copy a member.
D	Display a member.
E	Erase a member.
F	Reset a member's "in-use" flag.
P	Change, add, or delete member password.
R	Rename a member.
S, U, X, \ (forward slash)	Select a member to update.

3. Press Enter.
 - ◆ If you are updating a member, the appropriate update member screen displays.
 - ◆ Otherwise, the Manage Library Members screen displays (view [sample](#)).

4. After you have performed the desired function on the member, press F3. BMS/TS returns to the General Directory List screen.

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [How BMS/TS Works with the IBM 3270 Bridge](#)
- [Configure BMS/TS for Template Generation](#)

Related topics:

- [3270 Bridge Web Services](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

BMS/TS Support for IBM 3270 Bridge

BMS/TS supports the IBM CICS Transaction Server 3270 Web Bridge (IBM 3270 bridge).

The IBM CICS Transaction Server 3270 Web Bridge (introduced in Transaction Server 1.2 and enhanced in TS 1.3) provides a method of accessing existing CICS applications written for 3270 input and output. It enables access to 3270-based CICS applications without the use of 3270 terminals. You can access them from the Web, MQSeries, CICS Business Transaction Services (BTS), or any other non-3270 application.

Refer to the IBM Redbook SG24-5480-00 "CICS Transaction Server for OS/390 Version 1 Release 3: Web Support and 3270 Bridge" for detailed information about the IBM 3270 bridge.

How BMS/TS Works with the IBM 3270 Bridge

BMS/TS support for the IBM 3270 bridge includes building "Application Data Structure" (ADS) in the BMS mapset load module. The ADS is a structure that contains additional information about the 3270 application that is accessing the BMS mapset. In addition to this structure, BMS/TS provides for the base generation of the templates used by the Transaction Server Web Interface.

Support for the the IBM 3270 bridge is based on the URM (User Replaceable Module). However it may not be necessary for you to write anything. CICS TS 1.3 provides three bridge (URM) programs; MQSeries version 2.1 provides an additional exit. One of the supplied exits should support most client needs.

These exits are based on the information supplied via the ADS generated by BMS/TS.

How does BMS/TS support differ from base CICS TS macro support? BMS/TS removed the need to run three different assemble steps to build the required map structures, as these are built directly at map submit time.

Because CICS TS macros do not support line occurs and expanded field occurs, often the ADS structure does not reflect the true description of the data. BMS/TS will use the occurs support defined in the base ADS to allow the definition of line occurs and field occurs, so you have a clearer picture of how the data is used in the target 3270 application.

BMS/TS will support the older forms of HTML template generation as well as the newer forms; there are advantages and disadvantages to both forms. Choose the form that best suits your needs:

- The older form of HTML template generation (before IBM CRI94F) does not support various features (such as multi-send operations and dynamic attribute switching by applications) that may be used in CICS applications. If your application does not use these features, then the newer form is not a requirement. Your control over the final document generated at run time is greater, because dynamic tag generation by the 3270 bridge is not used. This form is suitable for creating documents which are focused toward data requests or situations where the desired output is not a facsimile of the 3270 display. In BMS/TS, we refer to this older form as Type 1.
- The newer form (with the IBM CRI94F updates) allows the 3270 bridge to present HTML that will function more like a terminal interface. This is beneficial if you want to use the 3270 bridge as a gateway for client access to the CICS applications. We refer to this newer form as Type 2.

Configure BMS/TS for Template Generation

Perform these steps to configure BMS/TS for template generation:

- [Configure CICS Transaction Server for 3270 Bridge Operation](#),
- [Configure a *JCL Member for CICS Template Submission](#),
- [Configure a GTBUTIL Job for Batch Template Submission](#).

Configure CICS Transaction Server for 3270 Bridge Operation

You must install and configure the IBM-supplied 3270 Bridge exits in the CICS region. Refer to the appropriate IBM documentation for details about 3270 Bridge exit definition.

Keep in mind:

- Allocate a DFHHTML PDS, if it will be used. Remember to update the startup JCL for the CICS region to include a DD statement referring to this dataset.
- Configure the TCPIP SERVICES which will provide the HTTP interfaces for the 3270 bridge.

Configure a *JCL Member for CICS Template Submission

To create the templates to be used by the 3270 bridge, you need to create a SUBMIT member which includes the new @HTML parameter, as shown in the following sample.

```
//IEBUPDTE EXEC      PGM=IEBUPDTE, PARM='NEW'
//SYSPRINT DD      SYSOUT=A
//SYSUT1  DD      DSN=YOUR.HTML.PDS, DISP=SHR
//SYSUT2  DD      DSN=YOUR.HTML.PDS, DISP=SHR
//SYSIN   DD      *
@HTML    3270 BRIDGE HTML INSERTED HERE
/*
```

There are several options that you can use with the @HTML parameter:

BRCOMP Indicates whether you want to compress the HTML code.

- BRCOMP=Y compresses the HTML; this is the default value.
- BRCOMP=N does not compress the HTML.

Note: HTML is generated, like the BMS macros, with each tag on its own line. This creates requirements for large buffer sizes used by the CICS Bridge. To reduce these buffer requirements, use BRCOMP=Y to combine tags on lines where room allows.

BRLOOK Indicates how you want to the HTML screens to display.

- BRLOOK=IBM produces a light gray background similar to that of the BMS macro template generation.
- BRLOOK=GT produces a black background similar to a 3270–style display. This is the default value.

Remember, you can edit the resulting HTML to customize the displays.

BRTYPE Indicates the type of field names.

- BRTYPE=1 represents simple field names as implemented in Transaction Server version 1.
Type 1 example: &FLDNAME (uses full field name).
- BRTYPE=2 represents the field naming conventions used in Transaction Server version 2 in *rrccclll* (row, column, length) format.
Type 2 example: &F010020032_FLDNAME (uses up to 21 characters of name).

If you are unsure which to use, BRTYPE=1 is the only option supported by all versions of Transaction Server. Default is 1.

If you choose to use any of the options, separate them with commas. For example:

```
@HTML BRCOMP=N,BRLOOK=IBM
```

Configure a GTBUTIL Job for Batch Template Submission

You can generate templates using the GTBUTIL batch utility. This procedure uses the BMS/TS [CATAL](#) command in a manner similar to physical (and symbolic) mapset generation. The CATAL command now includes several new options for 3270 bridge support.

A new output DD for GTBUTIL has also been added "GT\$PCHH" to hold the generated template output prior to writing it to its final destination. GT\$PCHH is identical in definition to the GT\$PCHD and GT\$PCHS datasets.

Sample JCL for Bridge Template Generation Using GTBUTIL

```
//STEP1OF1 EXEC PGM=GTBUTIL
//SYSLST DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//*                                     ===== PLEASE MODIFY
//*                                     |||
//STEPLIB DD DISP=SHR,DSN=your.GTBR8000.TEST.LOADLIB
//GT$FILE DD DISP=SHR,DSN=your.GTBR8000.GT$FILE
```

```
//GT$PCHS DD SYSOUT=*
//GT$PCHP DD SYSOUT=*
//GT$PCHH DD SYSOUT=*
//SYSIN DD *
      CATAL MAPSET=ACCT, BRIDGE=YES, ADS=YES, SOSI=NO
/*
```

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.



Contents:

- [Overview](#)
- [BWS Analyzer](#)
- [GTBSHTM Service](#)

Related topics:

- [BWS Server Administration](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

BMS/TS Bridge Web Services (BWS)

BMS/TS Bridge Web Services (BWS) provide Transaction Server CICS Web Service (CWS) developers with several tools to assist in developing and managing Web applications.

There are several paths that a client application program or a Web browser can use to communicate with Transaction Server application services. This document explains how BMS/TS can enhance the interfaces to two of the major access solutions for Transaction Server (CWS) and the IBM 3270 Bridge.

You need to be familiar with the following abbreviations as you use this document.

BWS	BMS/TS B ridge W eb S ervices
CWI	C ICS W eb I nterface
CWS	C ICS W eb S ervice
TS	C ICS T ransaction S erver
TS	T ransaction S erver

Overview

The IBM 3270 Bridge provides access to 3270 applications via various interfaces. With BMS/TS Bridge Web Services (BWS), the focus is on HTML delivery and testing. With the introduction of BMS/TS and the support for the 3270 Bridge, BMS/TS now provides a set of tools and utilities to extend the tool set provided by Transaction Server.

Transaction Server CICS Web Services (CWS) with its early introduction as CICS Web Interface (CWI) in CICS/ESA 4.1 has grown from a simple HTTP interface into a Web application interface. Transaction Server (TS) 1.2 introduced the 3270 Bridge to allow existing 3270-based transactions to be driven from a Web browser or Web applications.

The majority of the CWS are still modules that must be modified by the client to extend their function or customized to meet client requirements.

With the introduction of TS 1.3, Secure Socket Layer was provided for secure access to applications making TS an even more promising Web Application platform. TS version 2 is becoming the accepted standard for Web Service applications. The need for tools to bridge the gap between the 3270 and Web Services created the customer demand for BMS/TS Bridge Web Services.

TS provides a set of new EXEC CICS commands for the Document API to allow applications to interface with HTML in a more programmer friendly environment. Even with these advances and the fact that CWS is easier to use, it still does not provide all the functions of a HTTP Web Server. BMS/TS BWS was designed to fill some of the openings left in the IBM–provided User Replacable Modules of CWS.

Enhancing the CWS with the BMS/TS Bridge Web Services (BWS)

The BMS/TS Bridge Web Services (BWS) provide the CWS with a rules–based analyzer program and a set of CWS service applications to enhance the process of building Web applications. The CWS service applications include:

- HTTP Server Services
 - ◆ URL Aliasing
 - ◆ URL Redirect
 - ◆ Basic Authentication
 - ◆ BMS/TS Server Process
 - ◆ Server Side Includes
- File system for text and graphic files
- FTP processing for file system
- Common ZIP file processing

The BWS will bridge the gap between the Transaction Server CICS Web Services and the Web application designer and developer.

The BWS provides a robust set of APIs to speed the development of CICS Web–based applications. Applications can be written to the BMS/TS HTTP Server method to allow the choice of running under the CWS or the BMS/TS HTTP Server.

Because the BWS provides functions previously found only in an HTTP server, now functions that normally have to be performed by another Web server can be performed in CICS. All of the static HTML and binary graphics files can be serviced by the BWS, providing a central location for the Web applications written for CICS. This makes the HTML coding simpler for the Web designer and the Web application developer.

The BMS/TS file system supports zip file formats and can support the directory structures found in Windows 9x, NT, Me, 2000, or XP. BWS also supports an FTP server for CICS, so that Web designers can use their PC tools to build the HTML and graphics that are needed to support any Web designs.

BWS with CICS brings a powerful application server to the Web, with all the CICS benefits of dynamic transaction backout and recovery and scalability.

BWS Analyzer

The BWS Analyzer for the CICS Web Services TCPIP SERVICE is a rules–based analyzer. The BWS Analyzer uses information from the incoming URL request and the analyzer rules base to decide what CICS resources are needed to process the request. The [BWS Analyzer rules](#) can specify:

- The name of the CICS program that is to process the request.
- The name of the converter that is to process the request.

- The name of the alias transaction that is to process the request.
- A user ID or terminal to be associated with the alias transaction.
- Any code page conversion that is needed for user data.

To use the CICS Web Services, you must first ensure that all of the installation requirements have been completed. When the CWS is ready, you can install the BWS.

You must install one or more TCPIP SERVICE resources, because they are required for HTTP requests to be submitted directly to CICS. The TCPIP SERVICE definition allows you to define which TCP/IP services are to use CICS internal sockets support. The TCPIP SERVICE definition also allows you to manage these internal CICS interfaces, with CICS listening on multiple ports, and different CICS Web Services on each port. You must install and open a TCPIP SERVICE definition for each port on which CICS is to listen for incoming HTTP requests.

Create your own TCPIP SERVICE definition for the BWS, or copy the HTTPNSSL or HTTPSSL definitions from the IBM supplied DFH\$SOT group into your own group and modify them to meet the BMS/TS requirements.

The important parameters for a Web TCPIP SERVICE are:

- The Status must be OPEN.
- The Transaction to be attached.
- The Portnumber on which CICS is to listen.
- The Backlog of requests to be processed which OS/390 TCP/IP is to allow
- The name of the analyzer program GTBANALZ.
- An IP address which CICS will listen to for incoming requests. If none is specified, CICS listens on all addresses used by TCP/IP in the OS/390 region where CICS is running.
- A TS queue name. This is the 6-character prefix of TS queue names generated by CICS Web support when writing inbound and outbound data to temporary storage. This prefix should correspond to the prefix defined by an installed TS model definition. If no prefix is supplied on the definitions, the default name of DFHWEB is used to generate TS queue names.

Define and install a TCPIP SERVICE definition to specify the port on which you want CICS and the browser to communicate. For example:

```

TCpipservice ==> BMSTSWS
Group ==> BMSTSWS
DEscription ==> Provides port number to display template on Web
URM ==> GTBANALZ
Portnumber ==> 8080
Certificate ==>
SStatus ==> Open
SSI ==> No
TTransaction ==> GTXN
Backlog ==> 00001
TSqprefix ==>
Ipaddress ==>

```

For more information on defining Web TCPIP SERVICES, see the *CICS Resource Definition Guide*.

CICS Transactions

The BWS requires extra alias transactions in order to define which file will be used to locate rules, static HTML and graphics.

Two CICS transactions are provided with CICS Web Support:

- Web attach transaction **GTXN**. This transaction invokes the analyzer program. It establishes the context in which the alias transaction **GTWA** will run, and issues the appropriate ATTACH command. When **GTXN** is defined as the Transaction on the TCPIP SERVICE definition, it is started by the sockets listener task **CSOL** when a new connection request is received on the port specified on the TCPIP SERVICE definition. If the HTTP 1.0 Keep-Alive header has been sent by the Web browser, **GTXN** remains in the system after the alias has been attached, and attaches new alias transactions to process further HTTP requests received from browser. If Keep-Alive has not been specified, **GTXN** terminates after the alias has been attached.
- Alias transaction **GTWA**. An alias transaction is a transaction that is started by the Web attach transaction (**GTXN**) to process a single request. Many instances of the alias transaction can be active in a CICS system at the same time, each processing a different request. The alias transaction runs the CICS-supplied alias program that calls the CICS program. If you wish, you can set up additional transaction definitions for alias transactions, each using the CICS-supplied alias program.

When BMS/TS was installed, the group BMSTSWs was created. It contains the definition of **GTWA** alias transaction as follows:

```
DEFINE TRANSACTION(GTWA) GROUP(BMSTSWs)
PROGRAM(DFHWBA) TWASIZE(0)
PROFILE(DFHCICST) STATUS(ENABLED)
TASKDATALOC(ANY) TASKDATAKEY(USER)
RUNAWAY(SYSTEM) SHUTDOWN(ENABLED)
PRIORITY(1) TRANCLASS(DFHTCL00)
DTIMOUT(NO) INDOUBT(BACKOUT)
SPURGE(YES) TPURGE(NO)
RESSEC(NO) CMDSEC(NO)
```

At the heart of the BWS Analyzer are the [rules](#) that control the decoding of the incoming URL. Without BWS, the CICS programmer must rely on the default model that is supplied by IBM or else must spend time additional writing modules to make the CWS do what is needed.

The following is an example of an incoming URL for the CWS 3270 Bridge using the default analyzer method.

```
http://n.n.n.n/cics/cwba/dfhwbtta/CEMT+I+TAS+ALL
```

Here is the same URL when using the BWS Analyzer for the CWS 3270 Bridge.

```
http://n.n.n.n/anyname/CEMT+I+TAS+ALL
```

As you can see, the BWS Analyzer makes the URL simpler and hides all the information from the browser user. The following entries are required in the BWS rules file ([bms_bws.ini](#)) to perform the

above URL decode operation.

```
{METHOD}
; URL Method for 3270 Bridge
/anyname*=BRIDGE
...
...
{BRIDGE}
; The service program for the bridge is DFHWBTTA
SERVICE=DFHWBTTA
USERID=*
```

The following is an example of an incoming URL for the an user written CWS Web Application.

```
http://n.n.n.n/cics/cwba/dfh$wb1a
```

Now the same URL needed when using the BMS/TS BWS Analyzer for the any CWS user-written Web application.

```
http://n.n.n.n/any-path-required/dfh$wb1a.BWS
```

Again a log of information is hidden from the user, but one hidden benefit is that the any <href> links in the HTML generated from the dfh\$wb1a program will come from the directory entered. Thus if you want two versions of HTML and graphics supporting the dfh\$wb1a module, it's a simple URL change; in fact, both copies exist and still function properly. The following entries are required in the BWS rules file ([bms_bws.ini](#)) to perform the above URL decode operation.

```
{METHOD}
; URL Method for any CICS Service
*.BWS*=STANDARD
...
...
{STANDARD}
; The service program comes from the URL
SERVICE=*
USERID=*
```

BWS Analyzer INI File

The BWS Analyzer is controlled by the GT:/ETC/BMS_BWS.INI file. This file contains all the information to customize the Analyzer program. Refer to [BWS Analyzer INI File](#) for details.

BMS/TS GTBSHTM Service

The BWS GTBSHTM Service provides HTTP access to all the files in the BMS/TS Hierarchical File System (HFS). This CWS program will serve out any static HTML and/or binary graphic files. It also provides date and time checking so that the files are not downloaded to the browser unless the files have changed.

This in conjunction with the BWS Analyzer program allows the CWS application designers have easy access to static and binary graphic files to support their Web application efforts. This service eliminates the need to write CWS service programs to do nothing more that serve out static files. It also makes for easier transportation of HTML files from other servers.

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Line Editor Functions](#)
- [Entering a Line Command](#)
- [Using the Command Prompt](#)
- [Using Function Keys](#)

[User's Guide Home](#)

[Administrator's Guide Home](#)

[BMS/TS Doc. Home](#)

Appendix A, Using the Line Editor

A line editor enables you to create, review, and update files containing lines of text. These textual lines perform a variety of functions: they set parameters, define operating procedures, protect data with passwords, etc. With a line editor such as the one included with BMS/TS, you work with one line of text at a time.

BMS/TS provides a line editor so you can perform a number of tasks. Although the body of the editor may vary depending on the option you are performing, the format of the different screens is basically the same.

This document describes how to perform options using the editor's command line, line commands, and function keys.

Line Editor Functions

You will use the BMS/TS line editor to:

- Edit input rules,
- Create and update tables,
- Edit the BMS/TS Transaction Monitor,
- Edit JCL,
- Update Operator Security.

The BMS/TS line editor provides three ways to process information on the editor screen:

- **Line Command Fields:** Enter commands which insert, copy, move, and/or delete lines of text from the editor.
- **Command Prompt:** Perform specific operations by entering commands from a Command line.
- **Function Keys:** Perform specific operations by pressing a function key.

Both the Command Prompt and the Function Keys enable you to locate and change data, display help, move around the editor screen, save information, and exit the editor screen. (View [sample](#) editor screen.)

With the exception of the JCL Editor, each of the editor screens has input fields for specific entries relating to the option. The JCL Editor has a blank body that allows you to write and review JCL

members.

Entering a Line Command

You can enter a line command to insert, copy, move, and/or delete lines of text from the editor.

1. Press Tab to move to the desired line in the LC column.
2. Perform the desired function:

If you want to...	Then...
Insert a line	<ol style="list-style-type: none">1. Type I (Insert).2. Press Enter. BMS/TS inserts a blank line immediately after the line in which the cursor is positioned.
Insert multiple lines	<ol style="list-style-type: none">1. Type <i>I</i><i>nn</i> (where <i>nn</i> indicates the number of lines you want to insert, from 1–99).2. Press Enter. BMS/TS inserts the desired number of blank lines immediately after the line in which the cursor is positioned.
Copy a line	<ol style="list-style-type: none">1. Type R (Replicate) or " (double quotation mark).2. Press Enter. BMS/TS copies the line on which the cursor is positioned.
Copy multiple lines	<ol style="list-style-type: none">1. Type <i>R</i><i>nn</i> or <i>nn</i> (where <i>nn</i> indicates the number of lines you want to copy, from 1–99).2. Press Enter. BMS/TS copies the desired number of lines beginning from the position of the cursor.
Copy a block of lines	<ol style="list-style-type: none">1. Type CC in front of the first line you want to copy.2. Type CC again in front of the last line in the block you want to copy.3. Position the cursor where you want to copy the marked block of lines.4. Type A (After) or B (Before) to indicate whether you want to insert the block after or before the current line.5. Press Enter. BMS/TS copies the block of lines to the desired location.
Move a line	<ol style="list-style-type: none">1. Type M (Move) in front of the line you want to move.2. Position the cursor where you want to move the line.3. Type A (After) or B (Before) to indicate whether you want to insert the line after or before the current line.4. Press Enter. BMS/TS moves the line to the desired location.
Move a block of lines	<ol style="list-style-type: none">1. Type MM (Move) in front of the first line you want to move.2. Type MM again in front of the last line in the block you want to move.

	<ol style="list-style-type: none"> 3. Position the cursor where you want to copy the marked block of lines. 4. Type A (After) or B (Before) to indicate whether you want to insert the block after or before the current line. 5. Press Enter. BMS/TS moves the block of lines to the desired location.
Move current line to top of screen	<ol style="list-style-type: none"> 1. Type / (forward slash). BMS/TS redisplay the screen with current line at the top of the editor screen. Keep in mind that this is to help text entry only; the screen arrangement actually remains unchanged.
Delete a line	<ol style="list-style-type: none"> 1. Type D (Delete). 2. Press Enter. BMS/TS deletes the line where your cursor is positioned.
Delete multiple lines	<ol style="list-style-type: none"> 1. Type <i>Dnn</i> (where <i>nn</i> indicates the number of lines you want to delete, from 1–99). 2. Press Enter. BMS/TS deletes the desired number of lines beginning from where your cursor is positioned.

Using the Command Prompt

You can perform specific functions (such as repositioning the cursor, changing data, etc.) by entering processing commands from the Command line.

1. Press Tab to move to the Command line at the bottom of the screen.
2. Perform the desired function:

If you want to...	Then...
Move to the bottom of the editor screen	<ol style="list-style-type: none"> 1. Type BOT or B (Bottom). 2. Press Enter. BMS/TS moves the cursor to the last line of the screen.
Move to the top of the editor screen	<ol style="list-style-type: none"> 1. Type TOP or T (Top). 2. Press Enter. BMS/TS moves the cursor to the first line of the screen.
Move down the editor screen a certain number of lines	<ol style="list-style-type: none"> 1. Type <i>Dnn</i> (where <i>nn</i> indicates the number of lines you want to move down, from 1–99). 2. Press Enter. BMS/TS moves the cursor down the desired number of lines.
Move up the editor screen a certain number of lines	<ol style="list-style-type: none"> 1. Type <i>Unn</i> (where <i>nn</i> indicates the number of lines you want to move up, from 1–99). 2. Press Enter. BMS/TS moves the cursor up the desired number of lines.
Scroll the display forward	<ol style="list-style-type: none"> 1. Type SF (Scroll Forward). 2. Press Enter. BMS/TS moves the display forward one screen (if applicable).

Scroll the display backward	<ol style="list-style-type: none"> 1. Type SB (Scroll Backward). 2. Press Enter. BMS/TS moves the display back one screen (if applicable).
Switch between half and full screen scrolling	<ol style="list-style-type: none"> 1. Type SC (Scroll). 2. Press Enter. BMS/TS switches the type of screen scrolling (from half page to full page, or vice versa).
Switch between regular and alternate display	<ol style="list-style-type: none"> 1. Type FS (Format Switch). 2. Press Enter. BMS/TS switches the type of screen display (from regular to alternate display, or vice versa, if applicable). <p>Note: Alternate display lets you review information such as the current operator ID, length of a physical map, etc.</p>
Display the Help information for the screen	<ol style="list-style-type: none"> 1. Type HELP or H (Help). 2. Press Enter. BMS/TS displays help information for the appropriate editor screen.
Locate a specific line number	<ol style="list-style-type: none"> 1. Type Lnn (where <i>nn</i> indicates the number of the line you want to locate, from 1–99). 2. Press Enter. BMS/TS moves the cursor to the desired line.
Locate a specific string of data	<ol style="list-style-type: none"> 1. Type L /<i>data</i> or L '<i>data</i>' (where <i>data</i> represents the data you want to search for in the editor screen). 2. Press Enter. BMS/TS searches for and moves the cursor to the first occurrence of the desired data. <p>Note: BMS/TS searches forward from the position of the cursor.</p>
Repeat the last locate data command	<ol style="list-style-type: none"> 1. Type RL (Repeat Locate). 2. Press Enter. BMS/TS searches for and moves the cursor to the next occurrence of the desired data. <p>Note: BMS/TS searches forward from the position of the cursor.</p>
Change data	<ol style="list-style-type: none"> 1. Type CHG /<i>olddata/newdata</i>/ or CHG '<i>olddata/newdata</i>' (where <i>olddata</i> represents the existing string of data that you want to change, and <i>newdata</i> represents the new string of data to use to replace the old data). 2. Press Enter. BMS/TS searches for the first occurrence of the <i>olddata</i>, and replaces it with the <i>newdata</i>.
Repeat change of data	<ol style="list-style-type: none"> 1. Type RC (Repeat Change). 2. Press Enter. BMS/TS repeats the last change data command.
Change all occurrences of data	<ol style="list-style-type: none"> 1. Type CHG /<i>olddata/newdata</i>**/ or CHG '<i>olddata/newdata</i>'*** (where <i>olddata</i> represents the existing string of data that you want to change, and <i>newdata</i> represents the new string of data to use to replace the old

	<p>data).</p> <p>2. Press Enter. BMS/TS searches for all occurrences of the <i>olddata</i>, and replaces them with the <i>newdata</i>.</p> <p>Note: To ensure that you change all desired data, move the cursor to the top of the editor screen before using this command.</p>
Sort transaction codes or table entries in ascending order	<p>1. Type SORT.</p> <p>2. Press Enter. BMS/TS sorts the transaction codes on Transaction Monitor Control screen, or table entries, in ascending order.</p>
Cancel the edit and return to the previous menu	<p>1. Type CANCEL or CAN (Cancel).</p> <p>2. Press Enter. BMS/TS returns to the previous menu without saving the current changes to the data on the editor screen.</p>
Save data on screen and return to previous menu	<p>1. Type END or E (End), or FILE or F (File).</p> <p>2. Press Enter. BMS/TS saves the updates to the editor screen and returns to the previous menu.</p>

Using Function Keys

You can perform specific functions (such as displaying help information, repositioning the cursor, etc.) by pressing a function key. The following table describes the available function keys. Note that some function keys are not available on all screens.

If you want to...	Then...
Display the Help information for the screen	Press F1 or F13. BMS/TS displays help information for the appropriate editor screen.
Switch between regular and alternate display	<p>Press F2 or F14. BMS/TS switches the type of screen display (from regular to alternate display, or vice versa, if applicable).</p> <p>Note: Alternate display lets you review information such as the current operator ID, length of a physical map, etc.</p>
Scroll right to display all 80 columns of Job Control Language	<p>Press F2 or F14. BMS/TS scrolls right so you can see all 80 columns of the screen.</p> <p>Note: This only applies to the JCL Editor.</p>
Save data on screen and return to previous menu, or exit help screen	Type F3 or F15. BMS/TS saves the updates to the editor screen and returns to the previous menu, or exits the help screen and returns to the editor screen.
Repeat the last locate data command	Press F5 or F17. BMS/TS repeats the last locate data command.

	Note: This key option does not appear on the Transaction Monitor Control screen.
Repeat the last change of data command	Press F6 or F18. BMS/TS repeats the last change data command. Note: This key option does not appear on the Transaction Monitor Control screen.
Scroll the display backward	Press F7 or F19. BMS/TS moves the display back one screen (if applicable).
Scroll the display forward	Press F8 or F20. BMS/TS moves the display forward one screen (if applicable).
Switch between half and full screen scrolling	Press F9 or F21. BMS/TS switches the type of screen scrolling (from half page to full page, or vice versa).
Move cursor to the command line	Press F12 or F24. BMS/TS moves the cursor to the Command line at the bottom of the editor screen.
Process data keyed on screen and/or move to next screen	Press Enter. BMS/TS processes the data you entered on the editor screen and either lets you key more data or continues to the next screen.
Cancel the edit and return to previous menu	Press Clear. BMS/TS returns to the previous menu without saving the current changes on the editor screen.

[Top of page](#)

[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.



Contents:

- Batch Functions

[User's Guide Home](#)

[Administrator's Guide Home](#)

[BMS/TS Doc. Home](#)

Appendix B, Batch Functions

This appendix provides basic information about the BMS/TS batch functions. It describes how to use the batch functions, with examples of JCL showing how to run maintenance jobs, conversions, and how to generate BMS macro source code.

This appendix provides instructions for using the following BMS/TS batch programs:

- [GTBCDMS](#) -- import/convert a DMS panel,
- [GTBCONV](#) -- import a map to BMS/TS,
- [GTBPUNCH](#) -- generate BMS source code ,
- [GTBUTIL](#) -- perform batch file utility operations.

Copyright GT Software, Inc.

Contents:

- [Convert DMS Panels](#)
- [Sample JCL](#)

Related topics:

- [Batch Functions](#)

[User's Guide Home](#)

[Administrator's Guide Home](#)

[BMS/TS Doc. Home](#)

Using GTBCDMS to Convert a DMS Panel

You can bring a DMS panel into the BMS/TS system library so that you can perform maintenance on the map and use it with standard CICS programs. You convert DMS panels using the GTBCDMS panel conversion program.

BMS/TS converts your DMS panels according to the following routine.

1. GTBCDMS reads the working panel the DMS card images as input.
2. GTBCDMS builds the symbolic mapname from the type "E" record of the DMS panel.
3. GTBCDMS builds a BMS/TS map of the panel. The conversion function also produces an audit trail to help you locate problems in a conversion.

Note: Keep in mind that DMS allows some parameters and conventions such as duplicate field names that are invalid for regular programming languages. Any errors associated with these invalid entries appear on the audit trail.

4. After BMS/TS converts the panel, the PRINT batch function lets you print an image of the map with the position of every field, field name, and attributes.

Convert DMS Panels

You can convert your DMS panel into a BMS/TS map so that you can perform maintenance on the map.

1. Identify the DMS panels you want to convert.
2. Log-on to your editor.
3. Write JCL to convert your DMS panel into your BMS/TS system library so that you can perform maintenance on the map.

Note: In order for the GTBCDMS program to work correctly, your JCL must include a DLBL or DD statement pointing to GT\$FILE. Refer to the JCL samples below.

4. Run the job.
 5. Review the audit trail to ensure that the panel successfully converted.
 6. Review the converted map online to verify that you can access and update it with BMS/TS. See [Working With Maps](#) for information on how to review and/or update a map.
-

Sample JCL for GTBCDMS

Sample JCL (OS/390) for GTBCDMS

```
//BMSTS    JOB. . . .
//JOB CAT  DD   DSN=usercat,DISP=SHR
//GTBCDMS  EXEC PGM=GTBCDMS
//STEPLIB  DD   DSN=BMSTS.LOADLIB,DISP=SHR
//SYSLST   DD   SYSOUT=A
//GT$FILE  DD   DSN=GT.MASTER.FILE,DISP=SHR
//SYSIN    DD   *
DMS001                                C  004A
                                        2  A
1210                                    1  02B
TEST DMS PANEL                        2  B
1          DMS PANEL dms001
2LINE2 : _____FIELD1=====>
                                        C
                                        D
FLD1      21010      FLD2      23515A  E
/*
/ &
```

Sample JCL (VSE) for GTBCDMS

```
// JOB. . . .
// DLBL GT$FILE, 'GT.MASTER.FILE', , VSAM
// EXEC PROC=GTBLIBS
// EXEC GTBCDMS, SIZE=AUTO
DMS001                                C  004A
                                        2  A
1210                                    1  02B
TEST DMS PANEL                        2  B
1          DMS PANEL dms001
2LINE2 : _____FIELD1=====>
                                        C
                                        D
FLD1      21010      FLD2      23515A  E
/*
/ &
```

[Top of page](#)

[User's Guide Home](#)

[Administrator's Guide Home](#)

[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Import Process](#)
- [Importing BMS Maps – OS/390](#)
- [Importing BMS Maps – VSE](#)
- [Importing BMS Maps – VSE/SP](#)
- [Review the Audit Trail](#)
- [Review the Imported Map](#)

Related topics:

- [Batch Functions](#)
- [Appendix D, Support for Multi-Byte Character Sets](#)

[User's Guide Home](#)

[Administrator's Guide Home](#)

[BMS/TS Doc. Home](#)

Using GTBCONV to Convert/Import BMS Maps

You can import a BMS map to the BMS/TS system library using OS/390 and VSE operating systems. Then you can review the audit trail and the converted map so that you can perform maintenance on the map.

You convert/import BMS maps using the GTBCONV map conversion program. The parameters used with GTBCONV vary depending on your operating system.

The conversion program will convert most BMS macro-produced maps and copybooks. SDF or maps produced by other screen generators may require minor manual edits.

Note: Through the rest of this document, we will use the term “import” in place of “import/convert”.

Caution: You must close the GT\$FILE in order to use GTBCONV.

GTBCONV Import Process

BMS/TS imports your BMS maps according to the following routine.

1. GTBCONV first retrieves the desired physical map from the designated load library, then assigns "dummy" field names to the map. Because the conversion function does not require any BMS source macros, you can bring any map used by CICS into the BMS/TS system library.
2. GTBCONV extracts the symbolic map from the specified object or copybook library.

Note: If the symbolic map is an input/output map, the mapname must include an I (Input) or O (Output) suffix. All fields within the symbolic map must also have the appropriate suffixes in order for BMS/TS to correctly identify field types during the conversion.

3. GTBCONV compares the field specifications of both the physical and symbolic maps to ensure that they correctly match.
4. If the field names match, GTBCONV places the field names from the symbolic map into the physical map. The resulting map is a BMS/TS map.

Importing BMS Maps – OS/390

Unlike the other BMS/TS batch functions, the parameters associated with GTBCONV do not immediately follow the GTBCONV command in the actual JCL. Instead, they are passed at appropriate points in the job stream so that BMS/TS can locate the desired physical and symbolic BMS maps that you want to import.

The following table describes the valid parameters when using GTBCONV with an OS/390 system.

Parameters	Description
DSN	<p>Required --- Indicates the VSAM library where your BMS map is cataloged.</p> <p>Use the format: DSN=xxxxxxx where xxxxxxx is the library containing the BMS map.</p>
DSN	<p>Required --- Indicates the name of the library containing the physical BMS map you want to import.</p> <p>Use the format DSN=yyyyyyy where yyyyyy is the library containing the physical map.</p>
DCB	<p>Required --- Defines the imported BMS map as having the same block size as the original. This avoids errors during the conversion.</p> <p>Use the format: DCB=BLKSIZE=xxxx where xxxx is the map block size.</p>
CONVERT	<p>Required --- Indicates 1) name of the map you want to import, and 2) the programming language in which map is written and field prefix of map if applicable.</p> <p>Use the format CONVERT MAPSET=aaaaaaa,LANG=b,PREFIX=c where:</p> <ul style="list-style-type: none"> • <i>a</i> is the 1–8 character map name • <i>b</i> is the name of the programming language you are using. Valid entries are: C (COBOL), A (Assembler), P (PL/1), and R (RPG). • <i>c</i> is the field prefix, if applicable <p>Note: If your map includes a field prefix, you must use the CONVERT statement and the PREFIX parameter. By doing this, BMS/TS includes the prefix on the symbolic map, which you can then alter when you update the map with BMS/TS.</p>
DSN	<p>Required --- Indicates the name of the library containing the symbolic BMS map you want to import.</p> <p>Use the format DSN=zzzzzzzz where zzzzzzz is the library containing the symbolic map.</p>

Procedure -- Import BMS Maps (OS/390)

1. Identify the BMS maps you want to import.
2. Log-on to your editor.
3. Write JCL to import your BMS map into the BMS/TS system library so that you can perform maintenance on the map with your OS/390 system. Refer to the JCL sample below.

Note: In order for the GTBCONV program to work correctly, your JCL must include a DD statement pointing to GT\$FILE.

4. Run the job.
5. [Review the audit trail](#) to ensure that the map was successfully imported.
6. [Review the imported map](#) online to verify that you can access and update it with BMS/TS.

This JCL is only an example; your JCL must be specific to the files and libraries used at your company.

```
//..... JOB. . . .
//GTBCONV EXEC PGM=GTBCONV
//STEPLIB DD DSN=BMSTS.LOADLIB,DISP=SHR
//          DD DSN=loadlib.of.executable.map,DISP=SHR
//SYSLST DD SYSOUT=A
//GT$FILE DD DSN=GT.MASTER.FILE,DISP=SHR
//SYSIN DD *
          CONVERT MAPSET=CUSTACC,LANG=C,PREFIX=ACCT-
/*
//          DD DSN=copybook.pds(copybook),DISP=SHR
```

Importing BMS Maps – VSE

Unlike the other BMS/TS batch functions, the parameters associated with GTBCONV do not immediately follow the GTBCONV command in the actual JCL. Instead, they are passed at appropriate points in the job stream so that BMS/TS can locate the desired physical and symbolic BMS maps that you want to import.

The following table describes the valid parameters when using GTBCONV with a VSE system.

Parameter	Description
FROM	Required -- Indicates the name of the library containing the symbolic BMS map you want to import. Use the format FROM=xxxxxxx where xxxxxxx is the library containing the symbolic map.
PUNCH	

	<p>Required -- Indicates the programming language in which the BMS map is written, and the name of the BMS map you want to import.</p> <p>Use the format PUNCH x.yyyyyyyy where:</p> <ul style="list-style-type: none"> • x – The name of the programming language you are using. Valid entries are: C (COBOL), A (Assembler), P (PL/1), R (RPG). • y – The 1–8 character map name.
SEARCH	<p>Required -- Indicates the name of the library or libraries containing the physical BMS map you want to import.</p> <p>Use the format SEARCH=xxxxxxx,yyyyyyy where:</p> <ul style="list-style-type: none"> • xxxxxxx – the first library you want BMS/TS to search for the physical map, • yyyyyyy – the second library you want BMS/TS to search for the physical map. <p>Note: BMS/TS will search through all indicated libraries for the physical map, so include all desired libraries in the SEARCH parameter. When including multiple libraries, be sure to separate each library with a comma.</p>

Procedure -- Import BMS Maps (VSE)

1. Identify the BMS maps you want to import.
2. Log-on to your editor.
3. Write JCL to import your BMS map into your BMS/TS system library so that you can perform maintenance on the map with your VSE system. Refer to the sample JCL below.

Note: In order for GTBCONV to work correctly, your JCL must include a DLBL statement pointing to GT\$FILE.

4. Run the job.
5. [Review the audit trail](#) to ensure that the map was successfully imported.
6. [Review the imported map](#) online to verify that you can access and update it with BMS/TS.

This is only an example; your JCL must be specific to the files and libraries used at your company.

```
// JOB. . . .
// DLBL IJSYSPH, 'PUNCH.FILE', 69/001
// EXTENT SYSPCH, DOSRES, . . . .
//   ASSGN SYSPCH, DISK, . . . .
// LIBDEF SL, FROM=ATSDEMO
// EXEC SSERV
// PUNCH C.CUSTACC
/*
// CLOSE SYSPCH, PUNCH
```

```

// DLBL IJSYSIN, 'PUNCH.FILE', 69/001
// EXTENT SYSIPT, DOSRES, . . . .
  ASSGN SYSIPT, DISK, . . . .
// DLBL GT$FILE, 'GT.MASTER.FILE', , VSAM, CAT=CUST
// EXEC PROC=GTBLIBS
// LIBDEF CL, SEARCH=ATSDEMO, ATXDEMO
*
// EXEC GTBCONV, SIZE=AUTO
  CLOSE SYSIPT, SYSRDR
/&

```

Importing BMS Maps – VSE/SP

Unlike the other BMS/TS batch functions, the parameters associated with GTBCONV do not immediately follow the GTBCONV command in the actual JCL. Instead, they are passed at appropriate points in the job stream so that BMS/TS can locate the desired physical and symbolic BMS maps that you want to import.

The following table describes the valid parameters when using GTBCONV with a VSE/SP system.

Parameter	Description
ACCESS	<p>Required -- Indicates the name of the library containing the symbolic BMS map you want to import.</p> <p>Use the format: ACCESS S=xxxxxxx, where xxxxxxx is the library containing the symbolic map.</p>
FORMAT	<p>Required -- Indicates that the format of the PUNCH parameter places the programming language indicator last. Always FORMAT=OLD.</p>
PUNCH	<p>Required -- Indicates the name of the BMS map you want to import, and the programming language in which the BMS map is written.</p> <p>Use the format: PUNCH yyyyyyyy.x, where:</p> <ul style="list-style-type: none"> • y – the 1–8 character map name, • x – the name of the programming language you are using. Valid entries are: C (COBOL), A (Assembler), P (PL/1), and R (RPG).

Procedure – Import BMS Maps (VSE/SP)

1. Identify the BMS maps you want to import.
2. Log-on to your editor.
3. Write JCL to import your BMS map to the BMS/TS system library so that you can perform maintenance on the map with your VSE/SP system. Refer to the sample JCL below.

Note: In order for the GTBCONV program to work correctly, your JCL must include a DLBL statement pointing to GT\$FILE.

4. Run the job.
5. [Review the audit trail](#) to ensure that the map was successfully imported.
6. [Review the imported map](#) online to verify that you can access and update it with BMS/TS.

This JCL is only an example; your JCL must be specific to the files and libraries used at your company.

```
// JOB. . . .
// DLBL IJSYSPH, 'PUNCH.FILE', 69/001
// EXTENT SYSPCH, DOSRES, . . . .
  ASSGN SYSPCH, DISK, . . . .
// EXEC SSERV
  ACCESS S=ATSDEMO
  PUNCH CUSTACC.C, FORMAT=OLD
/*
  CLOSE SYSPCH, PUNCH
// DLBL IJSYSIN, 'PUNCH.FILE', 69/001
// EXTENT SYSIPT, DOSRES, . . . .
  ASSGN SYSIPT, DISK, . . . .
// DLBL GT$FILE, 'GT.MASTER.FILE', , VSAM, CAT=CUST
// EXEC PROC=GTBLIBS
// EXEC GTBCONV, SIZE=AUTO
  CLOSE SYSIPT, SYSRDR
/&
```

Review the Audit Trail

You can review the audit trail to verify that the BMS maps were successfully imported.

The BMS/TS map conversion utility produces an audit trail that you can use to verify that the BMS maps were successfully imported, and to quickly locate problems. A sample audit trail appears below:

```
mmdyyyy          BMS/TS CONVERSION

      CONVERT MAPSET=COB3, LANG=C, PREFIX=ACCT-
B538 NEW MAPSET NAME ACCEPTED
          01 COB3I
B546 AUTOMATIC RETRIEVE OF MAPSET : COB3 COB3

B539 MAP FOUND
          02 ACCT-FLDTEST1I
B541 EXISTING FIELD NAME : FLD001
B535 NEW FIELD NAME ACCEPTED      : FLDTEST1

          02 ACCT-FLDTEST2I.
B587 GROUP FIELD FOUND

B541 EXISTING FIELD NAME          : FLD002
B535 NEW FIELD NAME ACCEPTED     : FLDTEST2

          03 ACCT-GRPFLD3I

B541 EXISTING FIELD NAME          : FLD002
B535 NEW FIELD NAME ACCEPTED     : GRPFLD1

          03 ACCT-GRPFLD2I
```

```

B541 EXISTING FIELD NAME          : FLD002
B535 NEW FIELD NAME ACCEPTED     : GRPFLD2

      03 ACCT-FLDOOCCSI
B586 FIELD OCCURS FOUND

B541 EXISTING FIELD NAME          : FLD003
B535 NEW FIELD NAME ACCEPTED     : FLDOCCS

B543 MAP REWRITTEN                : COB3 COB3
B510 ** OPERATION COMPLETE ** VM: 6.3.0

```

Audit Trail Error Messages

If the map import process does not successfully complete, the audit trail will include one or more error messages. These error messages identify the problem encountered during the conversion process.

The following table describes some of the error messages you are most likely to receive. (Refer to [System Messages](#) for a complete list of BMS/TS messages.)

Error Message	Description	Action
B530 Too many fields found	Occurs if the symbolic and physical maps do not match. Either: <ul style="list-style-type: none"> • the symbolic map has more fields than the physical map, or • the physical map has more fields than the symbolic map, or • you edited a physical map containing arrays but did not update the symbolic map. 	Verify that the fields of the symbolic map and physical map match, then retry the conversion.
B544 Field size does not match	Occurs if the symbolic and physical maps do not match. Either: <ul style="list-style-type: none"> • you edited the size of a field for one map but did not update the other, or • a map contains a group field and the size of the members does not add up to the total number of bytes in the group. 	Verify that the symbolic map is current and that it matches the physical map, then retry the conversion.
\$302 Record not found	Occurs if BMS/TS cannot locate the physical map in the specified library.	Verify that the specified library contains the physical map, then retry the conversion.

Hint: If a mapset contains one map that is causing conversion errors, you can delete that map from the copybook and run the conversion again. Then you will need to manually convert only the map that caused the error.

Arrays and Audit Trail Error Messages

If you encounter an error on the audit trail that is related to an array problem, you must perform the following steps.

1. Remove the partially–imported mapset from GT\$FILE. This is necessary because BMS/TS begins creating the imported file even if you subsequently encounter errors. The imported file therefore may not be complete.
 2. Check your JCL, then retry the import program.
 3. If the problem persists, update the partially–imported file. This enables you to correctly define all field, line, column, or user–defined arrays on the map using BMS/TS. Refer to [Working With Maps](#) for more information.
 4. Retry the import program.
-

Review the Imported Map

Review the imported map to verify that all fields were defined correctly. Update the map as needed. Refer to [Working with Maps](#) for instructions.

Note: Maps that contain occurs will return a condition code of 4; be sure to review these maps.

[Top of page](#)

[User's Guide Home](#)

[Administrator's Guide Home](#)

[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [PUNCH Format](#)
- [Parameters](#)
- [Examples](#)
- [JCL Samples](#)
- [Pass Information to the Output File](#)

Related topics:

- [Batch Functions](#)

[User's Guide Home](#)

[Administrator's Guide Home](#)

[BMS/TS Doc. Home](#)

Using GTBPUNCH to Generate BMS Source Code

You generate BMS source code by using the GTBPUNCH batch macro program and the PUNCH function.

PUNCH Format

Use the following format for the PUNCH function:

```
PUNCH MAPSET=aaaaaaa
```

Note: Type a comma to separate each parameter from any following parameter.

Also, in order for the PUNCH function to work correctly, your JCL must include a DLBL or DD statement pointing to GT\$FILE.

PUNCH Parameters

The following table describes the PUNCH parameters.

PUNCH Parameters	Description
MAPSET	<p>Required -- Indicates the name of the mapset for which you want to generate BMS source code.</p> <p>Note: The mapsets for which you want to generate macro source code must already reside in the BMS/TS system library.</p>
The following parameters are optional.	
BASE	Indicates whether you want to specify a base name for your COBOL or PL/1 programs. This allows symbolic maps of more than one mapset to use the same storage base.

	Use the format <code>BASE=aaaaaaaa</code> where <code>aaaaaaaa</code> represents the desired 1–8 character base name.
CNTRL	Indicates whether you want BMS/TS to generate the CATAL and BKEND for VSE or IEBUPDTE for OS/390 control cards. Valid entries are YES or NO. The default is YES .
LANG	Indicates the name of the programming language you are using. Valid entries are: C (COBOL), A (Assembler), P (PL/1), and R (RPG). The default is C (COBOL).
LDC	Indicates whether you want to use the CICS mnemonic when determining the logical device code to be used for BMS. Use the format <code>LDC=aa</code> where <code>AA</code> represents the CICS mnemonic.
MODE	Indicates whether you want to create a symbolic map for input or output use. Valid entries are: IN (create symbolic map for input use only), OUT (create symbolic map for output use only), and INOUT (create symbolic map for input & output). The default is INOUT .
OBFMT	Indicates whether you want to enable the BMS OBFMT option. Valid entries are YES or NO. The default is NO . Note: The BMS OBFMT option indicates that all maps within this mapset are eligible for use in outboard formatting.
PRINT	Indicates whether you want to enable the BMS print option for the map. Valid entries are YES or NO. The default is NO . Keep in mind that you must set this parameter to YES if you want to use the printer with this map.
SHORTNAME	Indicates that you want to use the short field name convention in the BMS source code. Include the SHORTNAME parameter to enable the short–naming convention. Otherwise, the long–naming convention is the default.
STORAGE	Indicates whether you want to set up each map within its own storage area. Valid entries are AUTO (Yes) or NO. The default is NO .
SUFFIX	Indicates whether BMS/TS should attach a single alphabetic character to the field name of the symbolic map. This allows you to use a suffix to describe the field. Valid entries are: <ul style="list-style-type: none"> • I -- Input • O -- Output • L -- Length • A -- Attribute • F -- Flag byte, if the field has been modified Note: BMS conventions require a suffix on field names. BMS/TS field name lengths are therefore short by one byte from standard field name lengths to accommodate the BMS requirement. In most cases, you should use one of the above suffixes so that the field name on your symbolic map

	conforms to CICS conventions.
TIOAPFX	Indicates whether you want to insert a terminal input/output area prefix for each map. Valid entries are YES or NO. The default is YES . Note: This parameter is required for command-level programming languages.
TYPE	Indicates whether you want to generate physical and/or symbolic maps during the assembly process. Valid entries are: <ul style="list-style-type: none"> • MAP (generate physical map only), • DSECT (generate symbolic map only), • &SYSPARM (generate physical & symbolic maps). The default is &SYSPARM .
ZEROLEN	Indicates whether you want to enable the BMS/TS zero field length capability. This allows you to set ending field attributes to a length of either 1 or 0. Valid entries are: YES (length of zero 0 for ending field attributes), and NO (length of one 1 for ending field attributes). The default is YES (zero).

PUNCH Examples

The following sample PUNCH function generates BMS macro source code for all members of the GTBDEMO mapset.

```
PUNCH MAPSET=GTBDEMO
```

The following sample PUNCH function generates BMS macro source code for all members of the GTBDEMO mapset. Notice that it generates code using the long naming convention but without using the BMS/TS zero length capability.

```
PUNCH MAPSET=GTBDEMO,LONGNAME,ZEROLEN=NO
```

PUNCH JCL Samples

The following examples show JCL statements that use the PUNCH function. Keep in mind that these are only examples; your JCL must be specific to the files and libraries used at your company.

Sample JCL (OS/390) for GTBPUNCH

```
//BMSTS    JOB.    . . .
//JOB CAT  DD     DSN=usercat,DISP=SHR
//GTBPUNCH EXEC  PGM=GTBPUNCH
//STEPLIB  DD     DSN=BMSTS.LOADLIB,DISP=SHR
//SYS LST  DD     SYSOUT=A
//GT$FILE  DD     DSN=GT.MASTER.FILE,DISP=SHR
//GT$PCHS  DD     DSN=&PCHS,UNIT=SYSDA,SPACE=CYL,1,1,
                   DISP=,PASS,DCB=BLKSIZE=80
```

```

//SYSIN DD *
PUNCH MAPSET=GTBDEMO
/*
//UPDATE EXEC PGM=IEBUPDTE
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSN=users.objlib,DISP=SHR
//SYSUT2 DD DSN=users.objlib,DISP=SHR
//SYSIN DD DSN=*.GTBPUNCH.GT$PCHS,
// DISP=OLD,DELETE,DELETE
/*

```

Sample JCL (VSE) for GTBPUNCH

```

// JOB. . . .
// DLBL GT$FILE, 'GT.MASTER.FILE', , VSAM
// DLBL IJSYSPH, 'PUNCH.FILE'
// EXTENT. . . .
ASSGN SYSPCH,DISK, . . . .
// EXEC PROC=GTBLIBS
// EXEC GTBPUNCH, SIZE=AUTO
PUNCH MAPSET=GTBDEMO
/*
CLOSE SYSPCH, PUNCH
// DLBL IJSYSIN, 'PUNCH.FILE'
// EXTENT. . . .
ASSGN SYSIPT,DISK, . . . .
* ASSGN A PVT SOURCE LIBRARY IF NEEDED
// EXEC MAINT
CLOSE SYSIPT, SYSRDR
/&

```

Sample JCL (VSE/SP) for GTBPUNCH

```

// JOB. . . .
// EXEC PROC=GTBLIBS
// DLBL GT$FILE, 'GT.MASTER.FILE', , VSAM
// DLBL IJSYSPH, 'PUNCH.FILE'.0
// EXTENT SYSPCH, . . . .
ASSGN SYSPCH,DISK, . . . .
// EXEC GTBPUNCH, SIZE=AUTO
PUNCH MAPSET=GTBDEMO
/*
CLOSE SYSPCH, PUNCH
// DLBL IJSYSIN, 'PUNCH.FILE'.0
// EXTENT SYSIPT. . . .
ASSGN SYSIPT,DISK, . . . .
// EXEC LIBR, PARM='A S=user.source'
CLOSE SYSIPT, SYSRDR
/&

```

Using PUNCH to Pass Information to the Output File

If desired, you can use PUNCH to pass information to the output file. This lets you add special comments, date cards, etc., to the BMS source code.

To pass a card to the VSE punch file or OS/390 GT\$PCHS file, type a % (percent sign) in the first position. This shifts the card image one position to the left and writes to the output file. For example, the following sample JCL adds two comments indicated by asterisks * to the source code:

```
%* ASSEMBLY DATE=@DATE  
% PUNCH MAPSET=GTBDEMO  
%* MAPS FOR GENERAL DEMO
```

[Top of page](#)

[User's Guide Home](#)

[Administrator's Guide Home](#)

[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- GTBUTIL

Related topics:

- [Batch Functions](#)

[User's Guide Home](#)

[Administrator's Guide Home](#)

[BMS/TS Doc. Home](#)

Using the GTBUTIL Batch Utility Program

The GTBUTIL utility program is used to perform the batch operations described below.

GTBUTIL Batch Functions

You use the GTBUTIL utility program to perform the following batch operations.

Function	Description
ALTER *	Changes the options of the existing *ADMIN record; directs processing to alternate CICS libraries. (See also CREATE , MOVE and SETFILE .)
BACKUP **	Backs up the entire BMS/TS system library. (See also DISKSEL .)
CATAL	Moves a specific map into the BMS/TS system library. (See also PROD and PUNCH .)
COPY *	Makes a duplicate of a member in the current BMS/TS system library. (See also RENAME .)
CREATE *	Initializes the BMS/TS system library and creates the *ADMIN record. (See also ALTER .)
DELETE *	Removes a member from the current BMS/TS system library.
DISKSEL **	Selectively backs up the BMS/TS system library. (See also BACKUP .)
DLIST	Prints a list of the entire BMS/TS system library. (See also PRINT .)
MOVE *	Transfers a member from one CICS library to another. (See also ALTER and SETFILE .)
PASSWRD *	Adds/changes password of a member in the current system library.
PRINT	Selectively prints a list of the BMS/TS system library. (See also DLIST .)
PROD *	Changes the status of a member in a library from test to production. (See also CATAL and PUNCH .)
PUNCH	Moves a specific table into the BMS/TS system library. (See also CATAL and PROD .)
RENAME *	Changes name of a member in the current BMS/TS system library. (See also COPY .)
RESET *	

	Changes the update flag of a set and/or member in the current BMS/TS system library so that you can edit the set/member.
RESTORE *	Restores the previous BMS/TS system library backup. (See also TAPESEL .)
SETFILE	Redirects all input and output requests to a specific CICS library. (See also ALTER and MOVE .)
TAPESEL *	Selectively restores the previous BMS/TS system library backup. (See also RESTORE .)
*You must close the xx\$FILE in order to use this command.	
**You must close the xx\$FILE to updates to ensure that you get an accurate backup copy of the file.	

[Top of page](#)

[User's Guide Home](#)

[Administrator's Guide Home](#)

[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Command Format](#)
- [Input Parameters](#)
- [Example](#)

Related topics:

- [GTBUTIL Batch Utility](#)

[User's Guide Home](#)
[Administrator's Guide Home](#)
[BMS/TS Doc. Home](#)

GTBUTIL ALTER

Use the ALTER function to change the parameters of the existing *ADMIN record.

Caution: You must close the GT\$FILE to use the ALTER command.

You can modify some of these parameters using the [online *ADMIN functions](#).

The ALTER function enables you to direct processing to CICS libraries other than GT\$FILE.

ALTER Format

Use the following format for the ALTER function.

ALTER *parameter(s)*

Note: Type a comma to separate each parameter from any following parameter.

Caution: If the *ADMIN record is protected by a master password (MastPass), then you must supply the MastPass every time you run the ALTER command.

In order for the ALTER function to work correctly, your JCL must include a DLBL or DD statement pointing to GT\$FILE.

ALTER Parameters

The following table describes the ALTER parameters. All of these parameters are optional.

ALTER Parameter	Description
ALIGNED	Indicates whether the maps produced by BMS/TS should be aligned or unaligned. Valid entries are YES (aligned) or NO (unaligned). The default is NO .
CSSF	Identifies the transaction used for your sign-off support. Use this parameter if you need BMS/TS to interface with your SIGNON transaction to perform storage cleanup and termination after a forced log-off. The default is NONE (no sign-off cleanup). Note: If you use this parameter, you must change the PCT entry for your sign-off transaction to point to GTBCSSN.

CSSF	<p>Identifies your CICS sign-off program name. The default is DFHSNP.</p> <p>Use this parameter to tell BMS/TS if you are using a sign-off program name other than a standard IBM sign-off program. Without this information GT Software cannot service your CICS region after a CICS ABEND or a forced log-off.</p>
CSSN	<p>Identifies the transaction used for your sign-on support. The default is NONE (no signon cleanup).</p> <p>Use this parameter if you need BMS/TS to interface with your SIGNON transaction to perform storage clean-up and termination after a forced log-off.</p>
CSSNP	<p>Identifies your CICS sign-on program name. Use this parameter to tell BMS/TS if you are using a sign-on program name different from the standard IBM sign-on program. Without this information GT Software cannot service your CICS region after a CICS ABEND or a forced log-off. The default is DFHSNP.</p>
DEMOLIB	<p>Indicates the 2-letter prefix of the *DEMO library, which contains your demo members. There is no default.</p>
DKEY	<p>Indicates the function key F1 through F24 that you want to act as the secondary help key for field level help requests. This option affects all entries on the Transaction Monitor and is system wide. There is no default.</p>
EUROPE	<p>Indicates whether to enable the European options. These options include date as well as numeric editing. Valid entries are Y (Yes) or N (No). The default is N.</p>
HELPMODEL	<p>Indicates the format of all BMS/TS help transactions for the TOR Terminal Owning Region and AOR Application Owning Region. The format of this parameter is X**H, where:</p> <ul style="list-style-type: none"> • X = any alpha or numeric character, • * = wildcard: any character is valid, • H = Always H. <p>For example:</p> <p>G**H -- Use all BMS/TS transaction codes starting with G and ending with H in the TOR/AOR help process.</p> <p>GT*H -- Use all BMS/TS transaction codes starting with GT and ending with H in the TOR/AOR help process.</p>
HELPTRN	<p>Indicates the help transaction code if you plan to run BMS/TS in a MRO multi-region option environment and have BMS/TS installed in the AOR only. Use a 3-character transaction code. The default is GTB.</p>
INTLIB	<p>Indicates the internal library. The default is GT, which represents the DDNAME that points to the internal library containing the internal library members BMS/TS help screens and menus.</p> <p>Note: When you use a prefix other than GT, you must change the internal library prefix to reflect the new DDNAME prefix.</p>

ITERM	Indicates the number of terminal work areas BMS/TS should acquire at system initialization. The value for this parameter should be an average for the number of active terminals using BMS/TS at the same time. There is no default.
JCLLIB	Indicates the 2–letter prefix of the library that contains your JCL members. The default is GT .
MAPLIB	Indicates the 2–letter prefix of the *MAPS library, which contains your maps. The default is GT .
MASTPASS	Allows you to enter the 1–8 character master password that protects various system maintenance functions. This also overrides any member passwords. Note: Do not include this parameter if you do not have a current master password.
MONITOR	Indicates the transaction code of the product that controls the Transaction Monitor. The default is GTB .
MRO	Required only if you are running BMS/TS in a MRO environment. Valid entries are: <ul style="list-style-type: none"> • NONE – You are running BMS/TS in a standalone environment. This is the default value. • AOR – You are running BMS/TS in an AOR environment; BMS/TS is not initialized in the TOR. You must also specify the parameter HELPMODEL=NONE. • TORAOR – BMS/TS will be initialized in both the TOR and AORs. You must also specify the HELPMODEL parameter. See the HELPMODEL parameter in this table for more information.
NAME	Indicates the name you want to assign to your library. The name can include up to 30 characters, but must begin and end with the same special character which cannot be used in the name itself. For example, if you wanted to call the library the Production Library, you would enter: +PRODUCTION LIBRARY+
NEWMAST	Allows you to replace the existing master password. Note: Use the word NONE to completely remove password security from your library.
PKEY	Indicates the function key F1 through F24 that tells BMS/TS to interpret the next function key pressed as a help request. This expands the number of function keys available for help. There is no default.
PRILIB	Indicates the primary library. The default is GT , which represents the DDNAME that is built based on the first two characters of the CICS transaction code for online functions.
SIGNON	Indicates whether to enable the security function. Valid entries are Y (Yes) or N (No). The default is N .
TCTUA	Required only if you are running BMS/TS in a MRO environment and installing BMS/TS in the TOR.

The format is <i>nnn</i> , where <i>nnn</i> represents an offset into the TCTUA of a 8 byte area. BMS/TS uses the area specified to pass information between CICS regions. The default is NONE .

ALTER Examples

The following sample ALTER function defines a master password (MASTPASS) to the *ADMIN record.

```
ALTER MASTPASS=password
```

The following sample ALTER function defines the library name.

```
ALTER NAME=+production library+
```

The following sample ALTER function enables security for the TR\$FILE library.

```
ALTER SIGNON=y,PRILIB=tr
```

[Top of page](#)

[User's Guide Home](#)

[Administrator's Guide Home](#)

[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Command Format](#)
- [Input Parameters](#)
- [Example](#)
- [JCL Samples](#)

Related topics:

- [GTBUTIL Batch Utility](#)

[User's Guide Home](#)

[Administrator's Guide Home](#)

[BMS/TS Doc. Home](#)

GTBUTIL BACKUP

Use the BACKUP function to make a backup copy of the entire BMS/TS system library.

Caution: You must close the xx\$FILE to updates in order to ensure that you get an accurate backup copy of the file.

BACKUP Format

Use the following format for the BACKUP function.

```
BACKUP LIST=xxx
```

Note: Type a comma to separate each parameter from any following parameter.

In order for the BACKUP function to work correctly, your JCL must include a DLBL or DD statement pointing to GT\$FILE.

BACKUP Parameters

The following table describes the BACKUP parameters. All of these parameters are optional.

BACKUP Parameter	Description
FROMDATE	Backup members created on/after the specified date. Enter the date in <i>ccyyymmdd</i> format; this format does not change for European dates.
LIST	Produces a list of the members written to the backup tape. Valid entries are YES or NO.
REWIND	Indicates whether you want the tape rewound following the backup. Valid entries are YES or NO. The default is NO .
TODATE	Backup members created on/before the specified date. Enter the date in <i>ccyyymmdd</i> format; this format does not change for European dates.

BACKUP Example

The following sample BACKUP function backs up the BMS/TS system library and instructs BMS/TS to include a list of members written to the tape.

```
BACKUP LIST=YES
```

BACKUP JCL Samples

The following examples show JCL statements that use the BACKUP function. Keep in mind that these are only examples; your JCL must be specific to the files and libraries used at your company.

BACKUP: Sample JCL (OS/390)

```
//BMSTS    JOB. . . .  
//JOB CAT  DD   DSN=usercat,DISP=SHR  
//GTBUTIL  EXEC PGM=GTBUTIL  
//STEPLIB  DD   DSN=BMSTS.LOADLIB,DISP=SHR  
//SYS LST  DD   SYSOUT=A  
//GT$FILE  DD   DSN=GT.MASTER.FILE,DISP=SHR  
//TAPOUT   DD   DSN=xxxxxxx, UNIT=TAPE,. . . . .  
//SYS IN   DD   *  
BACKUP  
/*
```

Note: OS/390 can also back up to disk.

BACKUP: Sample JCL (VSE)

```
// JOB. . . .  
// DLBL GT$FILE, 'GT.MASTER.FILE', , VSAM  
// TLBL TAPOUT, . . . . .  
// ASSGN SYS008, cuu  
// EXEC PROC=GTBLIBS  
// EXEC GTBUTIL, SIZE=AUTO  
BACKUP  
/*  
/&
```

[Top of page](#)

[User's Guide Home](#)

[Administrator's Guide Home](#)

[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Command Format](#)
- [Input Parameters](#)
- [Mapset Restrictions](#)
- [Example](#)
- [JCL Samples](#)

Related topics:

- [GTBUTIL Batch Utility](#)

[User's Guide Home](#)
[Administrator's Guide Home](#)
[BMS/TS Doc. Home](#)

GTBUTIL CATAL

Use the CATAL function to build a symbolic and/or physical map and catalog it into the BMS/TS system library.

Note: Output of CATAL is two files unless otherwise stated in the function: one for the physical map, one for the symbolic map.

CATAL Format

Use the following format for the CATAL function.

CATAL MAPSET=aaaaaaa

Note: Type a comma to separate each parameter from any following parameter.

In order for the CATAL function to work correctly, your JCL must include a DLBL or DD statement pointing to GT\$FILE.

CATAL Parameters

The following table describes the CATAL parameters.

CATAL Parameter	Description
MAPSET	Indicates the name of the mapset that you want to move to the BMS/TS system library. This is the only required CATAL parameter.
The parameters listed below are optional.	
ADS	Indicates whether you want to create ADS for the 3270 bridge. Valid entries are Y (Yes, create ADS for 3270 bridge) or N (No, do not create ADS). The default value is N .
BASE	Indicates whether you want to specify a base name for your COBOL or PL/1 programs. The format is <i>aaaaaaa</i> , where <i>aaaaaaa</i> represents the desired base name.

BRCOMP	<p>Indicates whether you want to compress the HTML code. Valid values are Y (Yes, compress the HTML) or N (No, do not compress). The default value is Y.</p> <p>Note: HTML is generated, like the BMS macros, with each tag on its own line. This creates requirements for large buffer sizes used by the CICS Bridge. To reduce these buffer requirements, use BRCOMP=Y to combine tags on lines where room allows.</p>
BRIDGE	<p>Currently the only valid value is bridge=html (or bridge=h). This forces the ADS parameter to Y.</p>
BRLOOK	<p>Indicates how you want to the HTML screens to display.</p> <ul style="list-style-type: none"> • BRLOOK=IBM produces a light gray background similar to that of the BMS macro template generation. • BRLOOK=GT produces a black background similar to a 3270–style display. This is the default value.
BRTYPE	<p>Indicates the type of field names.</p> <ul style="list-style-type: none"> • BRTYPE=1 represents simple field names as implemented in Transaction Server version 1. Type 1 example: &FLDNAME (uses full field name). • BRTYPE=2 represents the field naming conventions used in Transaction Server version 2 in <i>rrcccllll</i> (row, column, length) format. Type 2 example: &F010020032_FLDNAME (uses up to 21 characters of name). <p>If you are unsure which to use, BRTYPE=1 is the only option supported by all versions of Transaction Server. Default is 1.</p>
CNTRL	<p>Indicates whether you want BMS/TS to generate the CATAL and BKEND for VSE or IEBUPDTE for OS/390 control cards. Valid entries are YES or NO. The default is YES.</p>
ONLY	<p>Indicates whether you want to suppress the building of the physical map. Valid entries are YES or NO. The default is NO.</p>
LANG	<p>Indicates the name of the programming language you are using. Valid entries are:</p> <ul style="list-style-type: none"> • C (COBOL) -- this is the default. • A (Assembler) • P (PL/1) • R (RPG) <p>Note: You must set this parameter to “A” (Assembler) if you are requesting a resident map.</p>
LDC	<p>Indicates whether you want BMS/TS to use the CICS mnemonic when determining the logical device code to be used for BMS. Use the format LDC=aa, where aa represents the CICS mnemonic.</p>
MAP	<p>Indicates the name of the map that you want to move to the BMS/TS system library.</p>

	Note: This parameter is only valid for Assembler programs.
MAPONLY	Indicates whether you want to suppress the building of the symbolic map. Valid entries are YES or NO. The default is NO .
MODE	Indicates whether you want to create a symbolic map for input or output use. Valid entries are: <ul style="list-style-type: none"> • IN -- Create symbolic map for input use only. • OUT -- Create symbolic map for output use only. • INOUT -- Create symbolic map for input & output use; this is the default.
OBFMT	Indicates whether you want to enable the BMS OBFMT option. Valid entries are YES or NO. The default is NO . Note: The BMS OBFMT option indicates that all maps within this mapset are eligible for use in outboard formatting.
PRINT	Indicates whether you want to enable the BMS print option for the map. Valid entries are YES or NO. The default is NO . You must set this parameter to YES if you want to use the printer with this map.
RES	Indicates whether the physical map is resident in the ALC program. Valid entries are YES or NO. The default is NO . If you set this parameter to YES, you must also use the MAP parameter. Note: This parameter is only valid for Assembler programs.
SOSI	Indicates whether you want all of the maps in the mapset to be MBCS/SOSI-enabled. Valid entries are Y (Yes, make all maps in mapset MBCS/SOSI-enabled) or N (No, do not make the maps in the mapset MBCS/SOSI-enabled). The default value is N .
STORAGE	Indicates whether you want to set up each map within its own storage area. Valid entries are AUTO (Yes) or NO. The default is NO .
SUFFIX	Specifies a single alphabetic character for BMS/TS to attach to the end of the name of the physical and symbolic maps.
SUPP01	Indicates whether you want to change the 01 levels in the symbolic map to 02 to accommodate the new compiler restrictions for COBOL copybooks. Valid entries are YES or NO. The default is NO .
TIOAPFX	Indicates whether BMS/TS should insert a terminal input/output area prefix for each map. Valid entries are YES or NO. The default is YES . Note: This parameter is required for command-level programming languages.

Mapset Restrictions

The IBM 3270 Bridge requires a suffix which is appended to the mapset name. If your mapset name is 7 characters, the suffix is limited to one additional character (A–Z, 0–9). This means that you can have no more than 36 maps in the mapset.

If your mapset name is 6 (or fewer) characters, the suffix can be two additional characters (A–Z, 0–9, AA–ZZ, 00–99). You can have as many as 72 maps in the mapset.

CATAL Example

The following sample CATAL function submits all members of the GTBDEMO mapset to the BMS/TS system library.

```
CATAL MAPSET=GTBDEMO
```

CATAL JCL Samples

The following figures show examples of JCL statements that use the CATAL function. Keep in mind that these are only examples; your JCL must be specific to the files and libraries used at your company.

Note: The SAMPLIB member CATAL includes samples showing the Bridge Web Services parameters.

CATAL: Sample JCL (OS/390)

```
//BMSTS    JOB. . . .
//JOB CAT  DD   DSN=usercat,DISP=SHR
//GTBUTIL  EXEC PGM=GTBUTIL
//STEPLIB  DD   DSN=BMSTS.LOADLIB,DISP=SHR
//SYSLST   DD   SYSOUT=A
//GT$FILE  DD   DSN=GT.MASTER.FILE,DISP=SHR
//GT$PCHP  DD   DSN=&&PCHP,UNIT=SYSDA,SPACE=CYL,1,1,
              DISP=,PASS,DCB=BLKSIZE=80
//GT$PCHS  DD   DSN=&&PCHS,UNIT=SYSDA,SPACE=CYL,1,1,
              DISP=,PASS,DCB=BLKSIZE=80
//SYSIN    DD   *
CATAL MAPSET=GTBDEMO
/*
//UPDATE   EXEC PGM=IEBUPDTE
//SYSPRINT DD   SYSOUT=A
//SYSUT1   DD   DSN=users.copylib,DISP=SHR
//SYSUT2   DD   DSN=users.copylib,DISP=SHR
//SYSIN    DD   DSN=*.GTBUTIL.GT$PCHS,
//
              DISP=OLD,DELETE,DELETE
//LINK     EXEC PGM=IEWL,PARM='LIST,NCAL,LET, XREF'
//SYSPRINT DD   SYSOUT=A
//SYSLMOD  DD   DSN=BMSTS.LOADLIB,DISP=SHR
//SYSUT1   DD   UNIT=SYSDA,SPACE 1024, 100,10
//SYSLIN   DD   DSN=*.GTBUTIL.GT$PCHP,
//
              DISP=OLD,DELETE,DELETE
/*
```

CATAL: Sample JCL (VSE)

```
// JOB. . . .
// DLBL GT$FILE, 'GT.MASTER.FILE', , VSAM
// DLBL IJSYSPH, 'PUNCH.FILE'
// EXTENT. . . .
  ASSGN SYSPCH, DISK, . . . .
// EXEC PROC=GTBLIBS
// EXEC GTBUTIL, SIZE=AUTO
  CATAL MAPSET=GTBDEMO
/*
  CLOSE SYSPCH, PUNCH
// DLBL IJSYSIN, 'PUNCH.FILE'
// EXTENT. . . .
  ASSGN SYSIPT, DISK, . . . .
// EXEC PROC=GTBLIBS
* CATALOG PHYSICAL MAP
// OPTION CATAL
  INCLUDE
// EXEC LNKEDT
* CATALOG SYMBOLIC MAP
// EXEC MAINT
  CLOSE SYSIPT, SYSRDR
/&
```

CATAL: Sample JCL (VSE/SP)

```
// JOB. . . .
// DLBL GT$FILE, 'GT.MASTER.FILE', , VSAM
// DLBL IJSYSPH, 'PUNCH.FILE'
// EXTENT. . . .
  ASSGN SYSPCH, DISK, . . . .
// EXEC PROC=GTBLIBS
// EXEC GTBUTIL, SIZE=AUTO
  CATAL MAPSET=GTBDEMO
/*
  CLOSE SYSPCH, PUNCH
// DLBL IJSYSIN, 'PUNCH.FILE'
// EXTENT. . . .
  ASSGN SYSIPT, DISK, . . . .
// EXEC PROC=GTBLIBS
* CATALOG PHYSICAL MAP
// OPTION CATAL
  INCLUDE
// EXEC LNKEDT
* CATALOG SYMBOLIC MAP
// EXEC LIBR, PARM='A S=user.source'
  CLOSE SYSIPT, SYSRDR
/&
```

[Top of page](#)

[User's Guide Home](#)

[Administrator's Guide Home](#)

[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Command Format](#)
- [Input Parameters](#)
- [Example](#)

Related topics:

- [GTBUTIL Batch Utility](#)

[User's Guide Home](#)
[Administrator's Guide Home](#)
[BMS/TS Doc. Home](#)

GTBUTIL COPY

Use the COPY function to make a duplicate of an existing member in the current BMS/TS system library.

Caution: You must close the GT\$FILE to use the COPY command.

COPY Format

Use the following format for the COPY function.

```
COPY SET=aaaaaaa,MEMBER=bbbbbbb,NSET=ccccccc,NMEMBER=ddddddd
```

Note: Type a comma to separate each parameter from any following parameter.

In order for the COPY function to work correctly, your JCL must include a DLBL or DD statement pointing to GT\$FILE.

COPY Parameters

The following table describes the COPY parameters.

COPY Parameter	Description
MEMBER or MAP	<p>Required -- Indicates the 1–8 character name of the map or member that you want to copy.</p> <p>Note: You can use a * wild card if necessary. For example, to delete all members beginning with GT, you would enter GT*.</p>
NMEMBER or NMAP	<p>Required -- Indicates the 1–8 character name of the duplicate map or member.</p> <p>Note: You can use a * wild card if necessary. For example, to delete all members beginning with GT, you would enter GT*.</p>
NSET or NTYPE	<p>Required -- Indicates name of the duplicate mapset or special set. Mapsets are 1–8 characters. Valid special set entries are *DEMO, *JCL,</p>

	*MAPS, *OPID, *SYSTEM, *TABLE.
SET or TYPE or MAPSET	Required -- Indicates the name of the mapset or special set for the member you want to copy. Mapset names are 1–8 characters. Valid special set entries are *DEMO, *JCL, *MAPS, *OPID, *SYSTEM, *TABLE.
The following parameters are optional.	
NPASS	Indicates the password for the duplicate set or member.
PASS	Indicates the password of the set or member if you password protected the set/member. Note: If members of a set/type do not have the same password and you use a wildcard '*', you must supply the master password or BMS/TS displays the message Previous Function Abnormally Terminated. This message also displays if any update flags are set for a member.

Note: If you leave out a new name for a set/type, member/map, or password, BMS/TS uses the existing name or password.

COPY Example

The following sample COPY function duplicates a specific map in the BMS/TS system library.

```
COPY TYPE=GTBDEMO,MAP=MAP1,NTYPE=GTBDEMO,NMAP=MAP2
```

[Top of page](#)

[User's Guide Home](#)

[Administrator's Guide Home](#)

[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Command Format](#)
- [Input Parameters](#)
- [Examples](#)

Related topics:

- [GTBUTIL Batch Utility](#)

[User's Guide Home](#)
[Administrator's Guide Home](#)
[BMS/TS Doc. Home](#)

GTBUTIL CREATE

The CREATE function initializes the BMS/TS system library and creates the *ADMIN record.

Caution: You must use the CREATE function before the online version of *ADMIN can operate.

You must close the xx\$FILE in order to use the CREATE command.

CREATE Format

Use the following format for the CREATE function.

CREATE parameter(s)

Note: Type a comma to separate each parameter from any following parameter.

In order for the CREATE function to work correctly, your JCL must include a DLBL or DD statement pointing to GT\$FILE.

CREATE Parameters

The following table describes the unique CREATE parameters. All of these parameters are **optional**.

CREATE Parameter	Description
ALIGNED	Indicates whether the maps produced by BMS/TS should be aligned or unaligned. Valid entries are YES (aligned) or NO (unaligned). The default is NO .
MASTPASS	Allows you to define a 1–8 character master password that protects various system maintenance functions. This also overrides any member passwords.
NAME	Indicates the name you want to assign to your BMS/TS system library. The name can include up to 30 characters, but must begin and end with the same special character which cannot be used in the name itself. For example, if you wanted to call the library the Production Library, you would enter: +PRODUCTION LIBRARY+

Note: The remaining optional CREATE parameters are identical to those listed in the discussion of the [ALTER](#) function.

CREATE Examples

The following sample CREATE function initializes the *ADMIN record and adds a master password.

```
CREATE MASTPASS=password
```

The following sample CREATE function turns off the security signon feature.

```
CREATE SIGNON=n
```

The following sample CREATE function specifies the file containing the map and JCL libraries.

```
CREATE MAPLIB=zt,JCLLIB=zt
```

The following sample CREATE function specifies that the internal library containing the internal library members BMS/TS help screens and menus will be located in the ZT\$FILE.

```
CREATE INTLIB=zt
```

[Top of page](#)

[User's Guide Home](#)

[Administrator's Guide Home](#)

[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Command Format](#)
- [Input Parameters](#)
- [Example](#)

Related topics:

- [GTBUTIL Batch Utility](#)

[User's Guide Home](#)
[Administrator's Guide Home](#)
[BMS/TS Doc. Home](#)

GTBUTIL DELETE

Use the DELETE function to remove an existing member from the current BMS/TS system library.

Caution: You must close the xx\$FILE in order to use this command.

DELETE Format

Use the following format for the DELETE function.

```
DELETE SET=aaaaaaa,MEMBER=bbbbbbb
```

Note: Type a comma to separate each parameter from any following parameter.

In order for the DELETE function to work correctly, your JCL must include a DLBL or DD statement pointing to GT\$FILE.

DELETE Parameters

The following table describes the DELETE parameters.

DELETE Parameter	Description
MEMBER or MAP	<p>Required -- Indicates the name of the map or member 1–8 characters that you want to delete.</p> <p>Note: You can use a * wild card if necessary. For example, to delete all members beginning with GT, you would enter GT*.</p>
SET or TYPE or MAPSET	<p>Required -- Indicates the name of the mapset or special set for the member you want to delete.</p> <p>Mapsets are 1–8 characters. Valid special set names are *DEMO, *JCL, *MAPS, *OPID, *SYSTEM, *TABLE.</p>
PASS	<p>Indicates the password of the set or member that you want to delete. This parameter is required only if you password-protected the member.</p>

Note: If members of a set/type do not have the same password and you use a wildcard '*', you must supply the master password or BMS/TS displays the message Previous Function Abnormally Terminated. This message also displays if any update flags are set for a member.

DELETE Example

The following sample DELETE function deletes a specific map in the BMS/TS system library.

```
DELETE SET=GTBDEMO,MEMBER=MAP1
```

[Top of page](#)

[User's Guide Home](#)

[Administrator's Guide Home](#)

[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Command Format](#)
- [Input Parameters](#)
- [Example](#)
- [JCL Samples](#)

Related topics:

- [GTBUTIL Batch Utility](#)

[User's Guide Home](#)
[Administrator's Guide Home](#)
[BMS/TS Doc. Home](#)

GTBUTIL DISKSEL

Use the DISKSEL function to back up specific members of the BMS/TS system library. DISKSEL can read a previous backup or history file and then let you select members from the current BMS/TS system library to be merged, producing a new history file.

Caution: You must close the xx\$FILE to updates to ensure that you get an accurate backup copy of the file.

DISKSEL Format

Use one of the following formats for the DISKSEL function:

```
DISKSEL MAPSET=aaaaaaa,DELETE=xxx,TYPE=bbbb
```

or

```
DISKSEL SET=*aaaaaaa,MEMBER=*
```

Note: Type a comma to separate each parameter from any following parameter.

In order for the DISKSEL function to work correctly, your JCL must include a DLBL or DD statement pointing to GT\$FILE.

BMS/TS sets the dataset control block information dynamically so you do not need to code the block. However, if you accidentally code the block, BMS/TS automatically overrides coded control blocks.

DISKSEL Parameters

The following table describes the DISKSEL parameters. All of these parameters are **optional**.

DISKSEL Parameter	Description
ALL	Indicates that you want to back up all members from the current BMS/TS system library to the backup tape. Valid entries are ALL (enable global backup) or blank (disable global backup).
DELETE	

	Indicates whether you want BMS/TS to move all selected mapsets in production status to the history file, then delete the mapsets from the BMS/TS system library. This lets you move a new set of maps into production status. Valid entries are YES or NO. Note: Use this parameter with extreme caution.
FROMDATE	Backup members created on/after the specified date. Enter the date in <i>ccyyymmdd</i> format; this format does not change for European dates.
MAPSET	Indicates the name of the mapset you want to back up. You can select as many mapset names as desired. Note: You can use a * wild card if necessary. For example, to back up all mapsets beginning with GT, you would enter GT*.
MEMBER	Indicates the name of the member you want to back up. Required only if you enter a special set name. Keep in mind that members from the BMS/TS library replace the same named member in the history file.
REWIND	Indicates whether you want the tape rewound following the backup. Valid entries are YES or NO. The default is NO .
SET	Indicates the name of the special set you want to back up. Valid entries are *DEMO, *JCL, *MAPS, *OPID, *SYSTEM, *TABLE.
TODATE	Backup members created on/after the specified date. Enter the date in <i>ccyyymmdd</i> format; this format does not change for European dates.
TYPE	Moves maps with a specific status code to the History file. Valid codes are TEST (test status maps) or PROD (production status maps).

DISKSEL Example

The following sample DISKSEL function moves all mapsets with specific names to the history file.

```
DISKSEL MAPSET=CUST*,MAPSET=GT*
```

DISKSEL JCL Samples

The following examples show JCL statements that use the DISKSEL function. Keep in mind that these are only examples; your JCL must be specific to the files and libraries used at your company.

DISKSEL Sample JCL (OS/390)

```
//BMSTS   JOB. . . .
//JOB CAT DD   DSN=usercat,DISP=SHR
//GTBUTIL EXEC PGM=GTBUTIL
//STEPLIB DD   DSN=BMSTS.LOADLIB,DISP=SHR
//SYSLST  DD   SYSOUT=A
//GT$FILE DD   DSN=GT.MASTER.FILE,DISP=SHR
//TAPIN   DD   DSN=xxxxxx, UNIT=TAPE,. . . . .
```

```
//TAPOUT DD DSN=xxxxxx, UNIT=TAPE,. . . . .
//SYSIN DD *
DISKSEL SET=*JCL, MEMBER=SUBMIT
/*
```

Note: In the above example, //TAPIN DD DSN=xxxxxx, UNIT=TAPE,. . . indicates the tape unit address. If you want to create an output tape only, replace this statement with //TAPIN DD DUMMY. You can then use the output tape file with either the RESTORE or TAPESEL functions.

OS/390 can also back up to disk.

DISKSEL Sample JCL (VSE)

```
// JOB. . . .
// DLBL GT$FILE, 'GT.MASTER.FILE', , VSAM
// TLBL TAPIN, . . . . .
// TLBL TAPOUT, . . . . .
// ASSGN SYS007, cuu
// ASSGN SYS008, cuu
// EXEC PROC=GTBLIBS
// EXEC GTBUTIL, SIZE=AUTO
DISKSEL SET=*JCL, MEMBER=SUBMIT
/*
/&
```

Note: In the above example, ASSGN SYS007,cuu and ASSGN SYS008,cuu indicate the tape unit address. If you want to create an output tape only, replace cuu with IGN: for example, ASSIGN SYS007,IGN. You can then use the output tape file with either the RESTORE or TAPESEL functions.

[Top of page](#)

[User's Guide Home](#)

[Administrator's Guide Home](#)

[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Command Format](#)
- [Input Parameters](#)

Related topics:

- [GTBUTIL Batch Utility](#)

[User's Guide Home](#)
[Administrator's Guide Home](#)
[BMS/TS Doc. Home](#)

GTBUTIL DLIST

Use the DLIST function to produce a printed list of the entire BMS/TS system library.

DLIST Format

Use the following format for the DLIST function.

```
DLIST
```

DLIST Parameters

DLIST does not use any parameters. Simply include the DLIST command in your JCL.

[Top of page](#)

[User's Guide Home](#)
[Administrator's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Command Format](#)
- [Input Parameters](#)
- [Example](#)
- [JCL Samples](#)

Related topics:

- [GTBUTIL Batch Utility](#)

[User's Guide Home](#)
[Administrator's Guide Home](#)
[BMS/TS Doc. Home](#)

GTBUTIL MOVE

Use the MOVE function to transfer a member from one CICS library to another.

Caution: You must close the xx\$FILE in order to use this command.

MOVE Format

Use the following format for the MOVE function.

```
MOVE FFILE=aa,TFILE=bb,SET=xxxxxxx,MEMBER=yyyyyyy
```

Note: Type a comma to separate each parameter from any following parameter.

In order for the MOVE function to work correctly, your JCL must include a DLBL or DD statement pointing to GT\$FILE.

MOVE Parameters

The following table describes the MOVE parameters.

MOVE Parameter	Description
FFILE	Required --- Indicates the 2-character prefix of the library from which you are transferring a members.
MEMBER or MAP	Required --- Indicates the 1-8 character name of the map or member that you want to move. Note: You can use a * wild card if necessary. For example, to move all members beginning with GT, you would enter GT*.
SET or TYPE or MAPSET	Required --- Indicates the 1-8 character name of the mapset or special set for the member you want to move. Valid special set names are *DEMO, *JCL, *OPID, *SYSTEM, *TABLE.

	Note: If you are moving a member from a special set other than *MAPS, you must use the SET parameter.
TFILE	Required -- Indicates the 2-character prefix of the library to which you are transferring a members.
The following parameters are optional.	
DELETE	Indicates whether BMS/TS should delete the mapset from the sending library after the transfer. Valid entries are YES or NO. The default is NO .
REPLACE	Indicates whether BMS/TS should replace duplicate mapsets in the receiving library with the transferred mapsets. Valid entries are YES or NO. The default is NO .

MOVE Example

The following sample MOVE function transfers all maps in the GTBDEMO mapset of the GT library to the AT library.

```
MOVE FFILE=GT,TFILE=AT,SET=GTBDEMO,MEMBER=*
```

MOVE JCL Samples

The following examples show JCL statements that use the MOVE function. Keep in mind that these are only examples; your JCL must be specific to the files and libraries used at your company.

MOVE Sample JCL (OS/390)

```
//BMSTS   JOB. . . .
//JOB CAT DD   DSN=usercat,DISP=SHR
//GTBUTIL EXEC PGM=GTBUTIL
//STEPLIB DD   DSN=BMSTS.LOADLIB,DISP=SHR
//SYSLST  DD   SYSOUT=A
//GT$FILE DD   DSN=GT.MASTER.FILE,DISP=SHR
//AT$FILE DD   DSN=AT.MASTER.FILE,DISP=SHR
//SYSIN   DD   *
MOVE FFILE=GT,TFILE=AT,SET=GTBDEMO,MEMBER=*
/*
```

MOVE Sample JCL (VSE and VSE/SP)

```
// JOB. . . .
// DLBL GT$FILE,'GT.MASTER.FILE',,VSAM
// DLBL AT$FILE,'AT.MASTER.FILE',,VSAM
// EXEC PROC=GTBLIBS
// EXEC GTBUTIL,SIZE=AUTO
MOVE FFILE=GT,TFILE=AT,SET=GTBDEMO,MEMBER=*
/*
```

[Top of page](#)

[User's Guide Home](#)

[Administrator's Guide Home](#)

[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Command Format](#)
- [Input Parameters](#)
- [Example](#)

Related topics:

- [GTBUTIL Batch Utility](#)

[User's Guide Home](#)
[Administrator's Guide Home](#)
[BMS/TS Doc. Home](#)

GTBUTIL PASSWRD

Use the PASSWRD function to add or change the password of a member in the current BMS/TS system library.

Caution: You must close the xx\$FILE in order to use this command.

PASSWRD Format

Use the following format for the PASSWRD function.

```
PASSWRD SET=aaaaaaa,MEMBER=bbbbbbb
```

Note: Type a comma to separate each parameter from any following parameter.

In order for the PASSWRD function to work correctly, your JCL must include a DLBL or DD statement pointing to GT\$FILE.

PASSWRD Parameters

The following table describes the PASSWRD parameters.

PASSWRD Parameters	Description
MEMBER or MAP	<p>Required Indicates the 1–8 character name of the map or member for which you want to add or update a password.</p> <p>Note: You can use a * wild card if necessary. For example, to add or update passwords for all members beginning with GT, you would enter GT*.</p>
SET or TYPE or MAPSET	<p>Required Indicates the 1–8 character name of the mapset or special set for the member for which you want to add or update a password.</p> <p>Valid special set names are *DEMO, *JCL, *MAPS, *OPID, *SYSTEM, *TABLE.</p>

The following parameters are optional.	
NPASS	Indicates the new password for the set or member.
PASS	Indicates the password of the set or member that you want to delete if you password protected the set/member. Note: If members of a set/type do not have the same password and you use a wildcard '*', you must supply the master password or BMS/TS displays the message Previous Function Abnormally Terminated. This message also displays if any update flags are set for a member.

PASSWRD Example

The following sample PASSWRD function changes the password for a map in the BMS/TS system library.

```
PASSWRD SET=GTBDEMO,MEMBER=MAP1,PASS=1234,NPASS=4321
```

[Top of page](#)

[User's Guide Home](#)

[Administrator's Guide Home](#)

[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Command Format](#)
- [Input Parameters](#)
- [Examples](#)

Related topics:

- [GTBUTIL Batch Utility](#)

[User's Guide Home](#)
[Administrator's Guide Home](#)
[BMS/TS Doc. Home](#)

GTBUTIL PRINT

Use the PRINT function to print a display of a requested map, along with a list of all fields defined for the map. The report can also include all system defaults.

PRINT Format

Use one of the following formats for the PRINT function:

```
PRINT MAPSET=aaaaaaa,MAP=bbbbbbb
or
PRINT SET=*aaaaaaa,MEMBER=bbbbbbb
```

Note: Type a comma to separate each parameter from any following parameter.

In order for the PRINT function to work correctly, your JCL must include a DLBL or DD statement pointing to GT\$FILE.

PRINT Parameters

The following table describes the PRINT parameters. All of these parameters are **optional**.

PRINT Parameters	Description
*ADMIN	Allows you to print the contents of the *ADMIN record, which contains the BMS/TS system defaults. If the *ADMIN record has been protected with the Master Password, you must also include the MASTPASS=xxxxxxx parameter, where xxxxxxxx is the Master Password.
DSECT	Indicates whether to print the symbolic map. Valid entries are YES or NO. The default is YES .
FILLER	Indicates whether to print a special pattern wherever a name field display only exists on the physical map. Valid entries are YES or NO. The default is YES .

MAP	<p>Indicates the name of the map whose display and defined fields you want to print. Required only if you enter a mapset name. Valid entries are:</p> <ul style="list-style-type: none"> • * -- Print all members in a mapset, • aaaaaaa -- Where aaaaaaa represents the name of the map or member you want to print.
MAPprt	<p>Indicates whether to print the physical map. Valid entries are YES or NO. The default is YES.</p>
MAPSET	<p>Indicates the name of the mapset whose display and defined fields you want to print. You can select as many mapset names as desired.</p> <p>Note: You can use a * wild card if necessary. For example, to print the display and defined fields of all mapsets beginning with GT, you would enter GT*.</p>
MASTPASS	<p>Indicates the BMS/TS master password. This parameter is required if you are printing the *ADMIN record and it has been protected with the master password. There is no default.</p>
MEMBER	<p>Indicates the name of the map or member whose display and defined fields you want to print. Required only if you enter a special set name.</p> <p>Note: You can use a * wild card if necessary. For example, to print the display and defined fields of all members beginning with GT, you would enter GT*.</p>
SET	<p>Indicates the name of the special set whose maps you want to print.</p> <p>Valid entries are *JCL, *MAPS, *OPID, *SYSTEM, *TABLE.</p>
SF	<p>Indicates whether to print a single special character wherever an attribute byte exists on the physical map. Valid entries are:</p> <ul style="list-style-type: none"> • NONE -- The attribute byte does not print, • NO -- The attribute byte does not print, • a -- Where "a" represents the character you want to print in place of all attribute bytes. <p>The default is @.</p>
SPACE	<p>Indicates whether to print a single special character wherever a true space exists on the physical map. This shows you which bytes of the map are true spaces hexadecimal 40 instead of nulls hexadecimal/binary 00. Valid entries are:</p> <ul style="list-style-type: none"> • NO -- Special character does not print. • a -- Where "a" represents the character you want to print in place of all true spaces. <p>The default is "?".</p>
VALID	<p>Indicates whether to print any input rules associated with the map. Valid entries are YES or NO. The default is NO.</p>

PRINT Examples

The following sample PRINT function prints a specific map. The report includes special characters in place of both attribute bytes and true spaces on the physical map.

```
PRINT MAPSET=GTBDBST,MAP=MAP1,SF=#,SPACE=&,FILLER=NO
```

The following sample PRINT function prints the *ADMIN record. Notice that the MASTPASS parameter is required if *ADMIN has been protected with a master password.

```
PRINT *ADMIN,MASTPASS=xxxxxxx
```

[Top of page](#)

[User's Guide Home](#)

[Administrator's Guide Home](#)

[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Command Format](#)
- [Input Parameters](#)
- [Example](#)

Related topics:

- [GTBUTIL Batch Utility](#)

[User's Guide Home](#)
[Administrator's Guide Home](#)
[BMS/TS Doc. Home](#)

GTBUTIL PROD

Use the PROD function to change the status of a member in the BMS/TS system library from test to production.

Caution: You must close the xx\$FILE in order to use this command.

PROD Format

Use the following format for the PROD function.

```
PROD MAPSET=aaaaaaa,MAP=bbbbbbb
```

Note: Type a comma to separate each parameter from any following parameter.

In order for the PROD function to work correctly, your JCL must include a DLBL or DD statement pointing to GT\$FILE.

PROD Parameters

The following table describes the PROD parameters.

PROD Parameters	Description
MAP	<p>Required -- Indicates the name of the map whose status you want to change.</p> <p>Note: You can use a * wild card if necessary. For example, to delete all members beginning with GT, you would enter GT*.</p>
MAPSET	<p>Required -- Indicates the name of the mapset for the member whose status you want to change. Mapset names are 1–8 characters.</p>
The following parameter is optional.	
PASS	<p>Indicates the password of the set or member if you password protected the set/member.</p> <p>Note: If members of a set/type do not have the same password and you use a wildcard '*', you must supply the master password or</p>

BMS/TS displays the message Previous Function Abnormally Terminated. This message also displays if any update flags are set for a member.

PROD Example

The following sample PROD function modifies a specific map from test to production status.

```
PROD MAPSET=GTBDEMO,MEMBER=MAP1
```

[Top of page](#)

[User's Guide Home](#)

[Administrator's Guide Home](#)

[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Command Format](#)
- [Input Parameters](#)
- [Example](#)
- [JCL Samples](#)

Related topics:

- [GTBUTIL Batch Utility](#)

[User's Guide Home](#)

[Administrator's Guide Home](#)

[BMS/TS Doc. Home](#)

GTBUTIL PUNCH

Use the PUNCH function to move a specific table into the BMS/TS system library.

PUNCH Format

Use the following format for the PUNCH function.

```
PUNCH SET=*TABLE,MEMBER=aaaaaaa
```

Note: Type a comma to separate each parameter from any following parameter.

In order for the PUNCH function to work correctly, your JCL must include a DLBL or DD statement pointing to GT\$FILE.

PUNCH Parameters

The following table describes the PUNCH parameters.

PUNCH Parameters	Description
SET	Required -- Indicates the name of the special set for the table you want to move into the BMS/TS system library. Always *TABLE .
MEMBER	Required -- Indicates the name of the table you want to move into the BMS/TS system library.

PUNCH Example

The following sample PUNCH function moves a specific table to the BMS/TS system library.

```
PUNCH SET=*TABLE,MEMBER=TABLE1
```

PUNCH JCL Samples

The following examples show JCL statements that use the PUNCH function. Keep in mind that these are only examples; your JCL must be specific to the files and libraries used at your company.

PUNCH Sample JCL (OS/390)

```
//BMSTS JOB. . . .
//JOB CAT DD DSN=usercat,DISP=SHR
//GTBUTIL EXEC PGM=GTBUTIL
//STEPLIB DD DSN=BMSTS.LOADLIB,DISP=SHR
//SYSLST DD SYSOUT=A
//GT$FILE DD DSN=GT.MASTER.FILE,DISP=SHR
//GT$PCHP DD DSN=&&PCHP,UNIT=SYSDA,SPACE=CYL,1,
           DISP=,PASS,DCB=BLKSIZE=80
//SYSIN DD *
PUNCH SET=*TABLE,MEMBER=TABLE1
/*
//IEBUPDTE EXEC PGM=IEBUPDTE
//SYS PRINT DD SYSOUT=A
//SYSUT1 DD DSN=users.objlib,DISP=SHR
//SYSUT2 DD DSN=users.objlib,DISP=SHR
//SYSIN DD DSN=*.GTBUTIL.GT$PCHP,
           DISP=OLD,DELETE,DELETE
//LINK EXEC PGM=IEWL,PARM='LIST,NCAL,LET, XREF
//SYS PRINT DD SYSOUT=A
//SYSLMOD DD DSN=BMSTS.LOADLIB,DISP=SHR
//SYSUT1 DD UNIT=SYSDA,SPACE 1024, 100,10
//SYSLIB DD DSN=users.objlib,DISP=SHR
//SYSLIN DD *
INCLUDE SYSLIBmmmmmmmmmm
NAME mmmmmmmmmR
/*
```

PUNCH Sample JCL (VSE and VSE/SP)

```
// JOB. . . .
// DLBL GT$FILE,'GT.MASTER.FILE',,VSAM
// DLBL IJSYSPH,'PUNCH.FILE'
// EXTENT. . . .
ASSGN SYSPCH,DISK,. . . .
// EXEC PROC=GTBLIBS
// EXEC GTBUTIL,SIZE=AUTO
PUNCH SET=*TABLE,MEMBER=TABLE1
/*
CLOSE SYSPCH,PUNCH
// DLBL IJSYSIN,'PUNCH.FILE'
// EXTENT. . . .
ASSGN SYSIPT,DISK,. . . .
// EXEC PROC=GTBLIBS
* CATALOG PHYSICAL MAP
// OPTION CATAL
PHASE mmmmmmmmm,*
INCLUDE
// EXEC LNKEDT
CLOSE SYSIPT,SYSRDR
/&
```

[Top of page](#)

[User's Guide Home](#)

[Administrator's Guide Home](#)

[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Command Format](#)
- [Input Parameters](#)
- [Example](#)

Related topics:

- [GTBUTIL Batch Utility](#)

[User's Guide Home](#)
[Administrator's Guide Home](#)
[BMS/TS Doc. Home](#)

GTBUTIL RENAME

Use the RENAME function to change the name of a member in the current BMS/TS system library.

Caution: You must close the xx\$FILE in order to use this command.

RENAME Format

Use the following format for the RENAME function.

```
RENAME SET=aaaaaaa,MEMBER=bbbbbbb,NSET=xxxxxxx,NMEMBER=yyyyyyy
```

Note: Type a comma to separate each parameter from any following parameter.

In order for the RENAME function to work correctly, your JCL must include a DLBL or DD statement pointing to GT\$FILE.

RENAME Parameters

The following table describes the RENAME parameters.

RENAME Parameters	Description
MEMBER or MAP	Required --- Indicates the name of the map or member that you want to rename. Note: You can use a * wild card if necessary. For example, to rename all members beginning with GT, you would enter GT*.
NMEMBER or NMAP *	Required --- Indicates the new name of the map or member.
NSET or TYPE *	Required --- Indicates the new name of the mapset or special set for the member you want to rename. Mapsets are 1–8 characters. Valid special set names are *DEMO, *JCL, *MAPS, *OPID, *SYSTEM, *TABLE.

SET or TYPE or MAPSET	<p>Required --- Indicates the name of the mapset or special set for the member you want to rename.</p> <p>Mapsets are 1–8 characters. Valid special set names are *DEMO, *JCL, *MAPS, *OPID, *SYSTEM, *TABLE.</p>
The following parameters are optional.	
NPASS *	Indicates the new password for the set or member.
PASS	<p>Indicates the password of the set or member that you want to rename if you password protected the set/member.</p> <p>Note: If members of a set/type do not have the same password and you use a wildcard '*', you must supply the master password or BMS/TS displays the message Previous Function Abnormally Terminated. This message also displays if any update flags are set for a member.</p>
* Note: If you do not specify a new name for a set/type, member/map, or password, BMS/TS uses the existing name or password.	

RENAME Example

The following sample RENAME function changes the name of a specific map in the BMS/TS system library.

```
RENAME SET=GTBDEMO,MEMBER=MAP1,NSET=GTBDEMO,NMEMBER=MAP2
```

[Top of page](#)

[User's Guide Home](#)

[Administrator's Guide Home](#)

[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Command Format](#)
- [Input Parameters](#)
- [Example](#)

Related topics:

- [GTBUTIL Batch Utility](#)

[User's Guide Home](#)
[Administrator's Guide Home](#)
[BMS/TS Doc. Home](#)

GTBUTIL RESET

Use the RESET function to change the update flag of a set and/or member in the current BMS/TS system library so that you can edit the set/member.

Note: You only need to use this function when CICS abnormally terminates, leaving a member in an "update" status.

Caution: You must close the xx\$FILE in order to use this command.

RESET Format

Use the following format for the RESET function.

```
RESET SET=aaaaaaa,MEMBER=bbbbbbb
```

Note: Type a comma to separate each parameter from any following parameter.

In order for the RESET function to work correctly, your JCL must include a DLBL or DD statement pointing to GT\$FILE.

RESET Parameters

The following table describes the RESET parameters.

RESET Parameters	Description
MEMBER or MAP	<p>Required -- Indicates the name of the map or member whose update flag you want to reset.</p> <p>Note: You can use a * wild card if necessary. For example, to reset the update flags for all members beginning with GT, you would enter GT*.</p>
SET or TYPE or MAPSET	<p>Required -- Indicates the name of the mapset or special set for the member whose update flag you want to reset.</p> <p>Mapsets are 1–8 characters. Valid special set names are *DEMO, *JCL, *MAPS, *OPID, *SYSTEM, *TABLE.</p>
The following parameter is optional.	
PASS	Indicates the password of the set or member whose update flag you want to reset if you password protected the set/member.

Note: If members of a set/type do not have the same password and you use a wildcard '*', you must supply the master password or BMS/TS displays the message Previous Function Abnormally Terminated. This message also displays if any update flags are set for a member.

RESET Example

The following sample RESET function resets the update flag for a specific map in the BMS/TS system library.

```
RESET SET=GTBDEMO,MEMBER=MAP1
```

[Top of page](#)

[User's Guide Home](#)

[Administrator's Guide Home](#)

[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Command Format](#)
- [Input Parameters](#)
- [Example](#)
- [JCL Samples](#)

Related topics:

- [GTBUTIL Batch Utility](#)

[User's Guide Home](#)
[Administrator's Guide Home](#)
[BMS/TS Doc. Home](#)

GTBUTIL RESTORE

Use the RESTORE function to recover a previous BMS/TS system library backup, including the password and expiration date of the software.

Caution: The BMS/TS system library must have an empty status before you perform this function.

You must close the xx\$FILE in order to use this command.

RESTORE Format

Use the following format for the RESTORE function.

```
RESTORE LIST=xxx,FROMDATE=yyyymmdd,TODATE=yyyymmdd
```

Note: In order for the RESTORE function to work correctly, your JCL must include a DLBL or DD statement pointing to GT\$FILE.

Separate multiple parameters with a comma.

RESTORE Parameters

The following table describes the RESTORE parameters. All of these parameters are **optional**.

RESTORE Parameters	Description
FROMDATE	Restore members created on/after the specified date. Enter the date in <i>ccyyymmdd</i> format; this format does not change for European dates.
LIST	Produces a list of the members written from the backup dataset. Valid entries are YES or NO.
TODATE	Restore members created on/before the specified date. Enter the date in <i>ccyyymmdd</i> format; this format does not change for European dates.

RESTORE Example

The following sample RESTORE function restores a previous BMS/TS system library backup, and instructs BMS/TS to include a list of members written to the tape.

RESTORE JCL Samples

The following examples show JCL statements that use the RESTORE function. Keep in mind that these are only examples; your JCL must be specific to the files and libraries used at your company.

RESTORE Sample JCL (OS/390)

```
//BMSTS    JOB. . . .
//JOB CAT  DD   DSN=usercat,DISP=SHR
//GTBUTIL  EXEC PGM=GTBUTIL
//STEPLIB  DD   DSN=BMSTS.LOADLIB,DISP=SHR
//SYS LST  DD   SYSOUT=A
//GT$FILE  DD   DSN=GT.MASTER.FILE,DISP=SHR
//TAPIN    DD   DSN=xxxxxxx, UNIT=TAPE,. . . . .
//SYSIN    DD   *
RESTORE
/*
```

RESTORE Sample JCL (VSE)

```
// JOB. . . .
// DLBL GT$FILE, 'GT.MASTER.FILE', , VSAM
// TLBL TAPIN, . . . . .
// ASSGN SYS007, cuu
// EXEC PROC=GTBLIBS
// EXEC GTBUTIL, SIZE=AUTO
RESTORE LIST=YES
/*
/&
```

[Top of page](#)

[User's Guide Home](#)

[Administrator's Guide Home](#)

[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Command Format](#)
- [Input Parameters](#)
- [Example](#)
- [JCL Samples](#)

Related topics:

- [GTBUTIL Batch Utility](#)

[User's Guide Home](#)

[Administrator's Guide Home](#)

[BMS/TS Doc. Home](#)

GTBUTIL SETFILE

Use the SETFILE function to redirect all input and output requests to a specific CICS library. This enables you to access multiple CICS libraries.

SETFILE Format

Use the following format for the SETFILE function.

```
SETFILE PREFIX=aa
```

Note: In order for the SETFILE function to work correctly, your JCL must include a DLBL or DD statement pointing to GT\$FILE.

SETFILE Parameters

The following table describes the SETFILE parameter.

SETFILE Parameter	Description
PREFIX	<p>Required -- Indicates the 2-character prefix of the library to which you want to redirect input and output.</p> <p>Note: BMS/TS uses the library you specify here until you issue another SETFILE function.</p>

SETFILE Example

The following sample SETFILE function redirects all input and output to the AT library.

```
SETFILE PREFIX=AT
```

SETFILE JCL Samples

The following examples show JCL statements that use the SETFILE function. Keep in mind that these are only examples; your JCL must be specific to the files and libraries used at your company.

SETFILE Sample JCL (OS/390)

```
//BMSTS    JOB.    . . .
//JOB CAT  DD     DSN=usercat,DISP=SHR
//GTBUTIL  EXEC   PGM=GTBUTIL
//STEPLIB  DD     DSN=BMSTS.LOADLIB,DISP=SHR
//SYSLST   DD     SYSOUT=A
//GT$FILE  DD     DSN=GT.MASTER.FILE,DISP=SHR
//AT$FILE  DD     DSN=AT.MASTER.FILE,DISP=SHR
//SYSIN    DD     *
DLIST
SETFILE PREFIX=AT
DLIST
/*
```

SETFILE Sample JCL (VSE and VSE/SP)

```
// JOB.    . . .
// DLBL GT$FILE, 'GT.MASTER.FILE', , VSAM
// DLBL AT$FILE, 'AT.MASTER.FILE', , VSAM
// EXEC PROC=GTBLIBS
// EXEC GTBUTIL, SIZE=AUTO
DLIST
SETFILE PREFIX=AT
DLIST
/*
/&
```

[Top of page](#)

[User's Guide Home](#)

[Administrator's Guide Home](#)

[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Command Format](#)
- [Input Parameters](#)
- [Example](#)
- [JCL Samples](#)

Related topics:

- [GTBUTIL Batch Utility](#)

[User's Guide Home](#)
[Administrator's Guide Home](#)
[BMS/TS Doc. Home](#)

GTBUTIL TAPESEL

Use the TAPESEL function to read a previous backup or history file and copy selected members back into the current BMS/TS system library.

Caution: You must close the xx\$FILE in CICS in order to use the TAPESEL function.

Separate multiple parameters with a comma.

You can only use one TAPESEL in a single execution of GTBUTIL.

TAPESEL Format

Use one of the following formats for the TAPESEL function:

```
TAPESEL MAPSET=aaaaaaa,REPLACE=xxx
or
TAPESEL ALL,REPLACE=xxx
or
TAPESEL SET=*aaaaaaa,MEMBER=aaaaaaa
```

Note: Type a comma to separate each parameter from any following parameter.

In order for the TAPESEL function to work correctly, your JCL must include a DLBL or DD statement pointing to GT\$FILE.

TAPESEL Parameters

The following table describes the TAPESEL parameters. All of these parameters are **optional**.

TAPESEL Parameters	Description
ALL	<p>Indicates that all members from the backup tape should merge into the current BMS/TS system library.</p> <p>Valid entries are ALL (enable global restore) or blank (disable global restore).</p> <p>Note: If duplicate members appear in the library, BMS/TS does not overwrite them.</p>
FROMDATE	

	Copy members created on/after the specified date. Enter the date in <i>ccyyymmdd</i> format; this format does not change for European dates.
MAPSET	Indicates the name of the mapset you want to restore. You can select as many mapset names as desired. Note: You can use a * wild card if necessary. For example, to restore all mapsets beginning with GT, you would enter GT*.
MEMBER	Indicates the name of the member you want to restore. Required only if you enter a special set name. Keep in mind that BMS/TS does not overwrite a member if a duplicate appears in the current library.
REPLACE	Indicates whether you want members from the backup tape to replace members in the BMS/TS system library with the same name. Valid entries are YES or NO. The default is NO .
SET	Indicates the name of the special set you want to restore. Valid entries are *DEMO, *JCL, *MAPS, *OPID, *SYSTEM, *TABLE.
TODATE	Copy members created on/before the specified date. Enter the date in <i>ccyyymmdd</i> format; this format does not change for European dates.

TAPESEL Example

The following sample TAPESEL function moves all maps on the backup tape to the BMS/TS system library.

```
TAPESEL ALL
```

TAPESEL JCL Samples

The following examples show JCL statements that use the TAPESEL function. Keep in mind that these are only examples; your JCL must be specific to the files and libraries used at your company.

TAPESEL Sample JCL (OS/390)

```
//BMSTS    JOB. . . .
//JOB CAT  DD   DSN=usercat,DISP=SHR
//GTBUTIL  EXEC PGM=GTBUTIL
//STEPLIB  DD   DSN=BMSTS.LOADLIB,DISP=SHR
//SYSLST   DD   SYSOUT=A
//GT$FILE  DD   DSN=GT.MASTER.FILE,DISP=SHR
//TAPIN    DD   DSN=xxxxxx, UNIT=TAPE,. . . .
//SYSIN    DD  *
           TAPESEL ALL
/*
```

TAPESEL Sample JCL (VSE)

```
// JOB. . . .  
// DLBL GT$FILE, 'GT.MASTER.FILE', , VSAM  
// TLBL TAPIN, . . . . .  
// ASSGN SYS007, cuu  
// EXEC PROC=GTBLIBS  
// EXEC GTBUTIL, SIZE=AUTO  
TAPESEL ALL  
/*  
/&
```

[Top of page](#)

[User's Guide Home](#)

[Administrator's Guide Home](#)

[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- BMS/TS Transaction Codes

[User's Guide Home](#)
[Administrator's Guide Home](#)
[BMS/TS Doc. Home](#)

Appendix C, BMS/TS Transaction Codes

The table below lists and describes BMS/TS transaction codes.

Code	Description
GTB/GTBA	Displays the main system menu. Using the GTB transaction, you can add, update, or display members on the product library. GTBA is the same as GTB, except that it supports alternate screen sizes.
GTBH/GTBI	Activates and deactivates the product by loading and unloading (GTBH Deload) the Transaction Monitor table. It is the transaction used during user help screen displays, and Application Interface processing. GTBI is the same as GTBH, except that it supports alternate screen sizes.
GTBS	Initiates the Installation Status Review screen. The status screen displays information about the installation; it is used for troubleshooting as well as during the install checkout process.
GTC/GTCA	HFS Command Line Processor. GTCA is the same as GTC, except that it supports alternate screen sizes.
GTIB	Password Administration function.
GTLB	NLS (National Language Support) administration.
GTP	Printer configuration.
GTRV	FTP Listener and Server.
GTTT	3270 Bridge Test Tool.

[Top of page](#)

[User's Guide Home](#)

[Administrator's Guide Home](#)

[ASSIST/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Basic Mapping Support](#)
- [BMS/TS MBCS Considerations](#)
 - ◆ [Design](#)
 - ◆ [Redesign](#)
 - ◆ [Submit Options](#)
 - ◆ [Map Conversion](#)

[User's Guide Home](#)
[Administrator's Guide Home](#)
[BMS/TS Doc. Home](#)

Appendix D, Support for Multi-Byte Character Sets

DBCS (Double-Byte Character Set) or MBCS (Multi-Byte Character Set) refer to character sets that require more than one character to compose a display character. Latin based languages use SBCS (Single byte Character Set) to define characters.

BMS/TS provides support for multi-byte character set (MBCS) languages such as Chinese, Japanese, Thai, etc. during map design and map creation. You can migrate BMS screens which were not previously designed for MBCS by simply recreating them during the SUBMIT function. This adds the BMS "SOSI" (Shift-Out/Shift-In) support to the map and DSECT copybooks.

SOSI refers to the method of using two special character values to allow switching from SBCS to DBCS or MBCS languages. Languages such as Chinese and Japanese rely on SOSI. Languages such as Thai, which is a true multi-byte language, rely on rules to define the order of characters and do not use SOSI.

For more refined or detailed requirements, you can design screens to include MBCS characters as screen constants and/or the individual control of each screen field to determine whether MBCS characters are allowed.

Caution: In order to develop maps using MBCS, you must have the Program Symbol turned **on** for your terminal. See your CICS Systems Programmer for assistance.

Basic Mapping Support

IBM has implemented basic Mapping Support (BMS) for MBCS support using the following procedures:

- **SOSI** – Allows the application to send/receive data that may or may not contain MBCS characters. The determining factor is the presence of the SO character (x'0E') preceding MBCS characters and the SI character (x'0F') indicating that any data following will be SBCS (Single-Byte Character Set). For additional information, refer to the appropriate CICS reference manuals.
- **Programmed Symbols** – Acts as an indicator that all data within the field is treated as MBCS data. Currently, the allowed values are x'00' or x'F8'. Programmed symbol '8' (x'F8') indicates the field contents are to be treated as MBCS data. For compatibility with ASCII versions of CICS, the ASCII '8' (x'38') is also supported and provides the same functionality. For additional

information, refer to the appropriate CICS reference manuals.

BMS/TS MBCS Considerations

You may need to consider BMS/TS support for MBCS when you:

- [Design a map](#),
 - [Redesign a map](#),
 - [Submit a map](#),
 - [Convert/import an existing map](#).
-

MBCS Support in BMS/TS Map Design

During the map options display, you can specify SOSI=YES to allow MBCS data to be entered while designing the screen. Then, [design the map](#). For MBCS fields, use the Field Mark key to represent any MBCS initial value. During field definition, you will be prompted to enter the initial value desired. In this way, BMS/TS can help the designer in defining the appropriate field size and attributes.

After you press Enter from the design screen, BMS/TS proceeds to field definitions. Aside from the typical field data, you may choose to enter the hex value F8 into the Programmed Symbol field, or type Y in the SOSI option field. Either way, the user will be presented with a panel to enter any initial value desired. BMS/TS will display a field of the appropriate size and attributes in which to key this data. This helps ensure the data and length attributes are properly set.

As before, when all fields have been defined, BMS/TS will present a verification display of the newly created/updated map (with the MBCS data embedded).

MBCS Support in BMS/TS Map Redesign

Screen redesign works the same as for non-MBCS maps. During the initial screen representations, MBCS data is signified using the Field Mark character. If you wish to change and examine initial MBCS data, use the Change field character to mark the field you wish to examine.

MBCS Support in BMS/TS Map Submission Options

If the map was designed with SOSI, you do not need to select options at submit time. If the map was not designed for SOSI and you want to do a quick conversion, select the SOSI option to enable SOSI for all Named fields.

During the map build (submit) process online or batch, you can specify SOSI=Yes to indicate that the map being submitted will be created for MBCS using SOSI. This allows the application to request a field be enabled for SOSI and send/receive MBCS data within these selected fields.

MBCS Support in BMS/TS Map Conversion (Import)

The batch import utility ([GTBCONV](#)) supports maps using SOSI and/or PS=8 MBCS support.

[Top of page](#)

[User's Guide Home](#)
[Administrator's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.

Contents:

- [Product Documentation](#)
- [Technical Support Staff](#)
- [Contact GT Software](#)

[BMS/TS Doc. Home](#)

[User's Guide Home](#)

[Administrator's Guide Home](#)

Getting Technical Support

This document explains how to get technical assistance if you have questions about BMS/TS.

Product Documentation

BMS/TS is supported by online help. The help system is the first place you should look when you need information about the product.

If the online help does not answer your question, refer to the appropriate sections of the product documentation for more detailed information.

You can also **search the product support database** on the GT Web site at <http://www.gtsoftware.com>.

Technical Support Staff

If you cannot resolve the problem using the product documentation, the GT Software Technical Support staff is available for telephone consultation. Before you call, have the following information available:

- BMS/TS release number,
- Transaction Server release number,
- The number and text of the system message (if any),
- Complete description of the problem.

In order to resolve a problem, it is important to tell the Technical Support representative exactly what was being done when the problem occurred. Details are important. It may be helpful to make notes about what happened before you call.

You can also try to re-create the problem, and tell the Technical Support representative whether you got the same results.

Contacting GT Software

GT Software Technical Support representatives are available Monday through Friday between the hours of 8:30 AM and 5:15 PM Eastern time. After business hours, you can leave a message. Your call will be returned the next business day.

To call GT Software, dial 404–253–1300.

To send a facsimile (fax), dial 404–253–1314.

Send Internet e–mail to sales@gtsoftware.com or support@gtsoftware.com.

Visit the GT Software Web site at [<http://www.gtsoftware.com>](http://www.gtsoftware.com).

[Top of page](#)

[BMS/TS Doc. Home](#)

[User's Guide Home](#)

[Administrator's Guide Home](#)

Copyright GT Software, Inc.

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

User's Guide Index

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [K](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#)
[X](#) [Y](#) [Z](#)

[~symbols
and
numbers~](#)

*[ADMIN](#) (Administration) record
 [generate](#) (GTBUTIL CREATE)
 [modify](#) (GTBUTIL ALTER)
 *[ADMIN](#) special set
 *[DEMO](#) special set
 *[JCL](#) special set
 *[MAPS](#) special set
 *[OPID](#) special set
 *[ROW](#) member name
 *[ROWCOL](#) member name
 *[ROWSET](#) member name
 *[SYSTEM](#) special set
 *[TABLE](#) special set
[3270 Bridge support](#)
[@HTML parameter](#) (online SUBMIT)

[~A~](#)

[ABEND messages](#)
 Administration record (*[ADMIN](#))
 [generate](#) (GTBUTIL CREATE)
 [modify](#) (GTBUTIL ALTER)
[ADS structure](#)
[ALTER](#) (GTBUTIL command)
 arrays
 [column arrays](#)
 [field arrays](#)
 [line arrays](#)
 [user-defined arrays](#)
[Assembler call or link to BMS/TS table processor](#)
[ATTRIB](#) (input rules command)
[audit trail, GTBCONV](#)
 [error messages](#)

[~B~](#)

[backup BMS system library](#)
[BACKUP](#) (GTBUTIL command)
[batch commands, GTBUTIL](#)
 [ALTER](#) (modify *[ADMIN](#) record)
 [BACKUP](#) (backup BMS library)
 [CATAL](#) (build/catalog symbolic or physical map)
 [COPY](#) (copy member in current library)

- CREATE (generate new *ADMIN record)
- DELETE (delete member from current library)
- DISKSEL (backup specific members)
- DLIST (print directory list)
- MOVE (move member to another library)
- PASSWRD (add/change member password)
- PRINT (print member)
- PROD (move member from test to production)
- PUNCH (move table into the BMS library)
- RENAME (rename member)
- RESET (reset update flag for member or set)
- RESTORE (restore system library backup)
- SETFILE (redirect input/output requests)
- TAPESEL (restore selected members)

batch functions

- GTBCDMS (convert DMS panels)
- GTBCONV (convert BMS maps)
- GTBPUNCH (generate BMS source code)
- GTBUTIL (batch utility functions)

batch job submission

- maps
- tables

BMS map conversion (GTBCONV)

- arrays and audit trail error messages
- audit trail messages
- audit trail review
- import/convert process
- OS/390
- review the converted map
- VSE

BMS source code generation (GTBPUNCH)

- branched help screens
- Bridge Web Services (BWS)
- build a page
- BWS Analyzer

~C~

- CALL (input rules command)
- Call field (branched help screens)
- CATAL (GTBUTIL command)
- chained demos
- chained help screens
- Change Field Character
 - help screen design
 - map design
- character sets, double-byte and multi-byte
- COBOL call or link to BMS/TS table processor
- column arrays
- Command line (Command prompt) on line editor screen
- contact GT Software
- conversational programs and input rules
- convert BMS maps (GTBCONV)
- convert DMS panels (GTBCDMS)

- copy library member
 - copy between multiple libraries
 - copy members in single library
- copy/move a field or block of fields (map redesign)
- COPY (GTBUTIL command)
- copybook
- CPU ID
 - add a CPU ID
 - change SYSPASS or FEATURE
 - delete CPU ID
 - online password administration
- CREATE (GTBUTIL command)
- CURS (input rules command)
- custom error messages (input rules)

~D~

- data types and lengths (in tables)
 - define
 - update
- DBCS (double-byte character sets)
- DELETE (GTBUTIL command)
- demos
 - chain demos
 - design a demo
 - input rules
 - list demos in the current library
 - menu options
 - redesign a demo
 - run demo
- DISKSEL (GTBUTIL command)
- display library member
- DLIST (GTBUTIL command)
- DMS panel conversion (GTBCDMS)
- double-byte character sets
- DSECT

~E~

- EDIT (input rules command)
- editor screens
- erase library member
- error messages

~F~

- fetch BMS map
- field arrays
- FILL (input rules command)
- flags (directory list)
- force complete entry of a field (input rules)
- force entry of data in a field (input rules)
- force field values (input rules)
- format dates (input rules)
- format numeric fields (input rules)

~G~

Generate BMS source code (GTBPUNCH)
GT Software contact information
GTBCDMS, convert DMS panels
GTBCONV, convert BMS maps
GTBPUNCH, generate BMS source code
GTBSHTM service
GTBUTIL batch utility functions
 ALTER (modify *ADMIN record)
 BACKUP (backup BMS library)
 CATAL (build/catalog symbolic or physical map)
 COPY (copy member in current library)
 CREATE (generate new *ADMIN record)
 DELETE (delete member from current library)
 DISKSEL (backup specific members)
 DLIST (print directory list)
 MOVE (move member to another library)
 PASSWRD (add/change member password)
 PRINT (print member)
 PROD (move member from test to production)
 PUNCH (move table into the BMS library)
 RENAME (rename member)
 RESET (reset update flag for member or set)
 RESTORE (restore system library backup)
 SETFILE (redirect input/output requests)
 TAPESEL (restore selected members)

~H~

help screens
 branched
 chained
 create
 design a map
 display formats
 map fields (define)
 map options
 naming conventions
Transaction Monitor, edit to add a help screen
verify the map

~I~

IBM 3270 Bridge support
IF, ELSE, ENDIF (input rules command)
initialize empty fields (input rules)
input rules
 and demos
 ATTRIB command
 CALL command
 commands
 common uses
 conversational programs, using with
 create
 CURS command
 custom error messages

- define
- EDIT command
- editor
- FILL command
- force complete entry of a field
- force entry of data in a field
- force field values
- format dates
- format numeric fields
- IF, ELSE, ENDIF commands
- implement
- initialize empty fields
- JUST command
- LET command
- OPTION command
- options
- PAD command
- reserved words
- RETN command
- return custom error messages
- SEL command
- STRIP command
- syntax
- table look-ups
- test after creation
- test after modification
- Transaction Monitor entry
- TROFF command
- TRON command

~J~

JUST (input rules command)

~L~

- LET (input rules command)
- library functions
 - add, change, or delete member password
 - copy member in library
 - copy member between libraries
 - display member
 - erase member
 - fetch BMS map
 - link libraries
 - list members
 - move member
 - rename member
 - reset member
 - special sets
 - transfer members between libraries
- line arrays
- line command field (line editor screen)
- line editor screens
 - Command prompt

- line command field
- function keys
- link libraries
- list
 - demos in the current library
 - maps in the current directory
 - members in a library
 - tables in the current library

~M~

- maps
 - arrays in a map
 - batch job submission
 - build a page
 - column arrays
 - copy/move a field or block of fields
 - define fields
 - define map options
 - design a map
 - display map
 - display fields
 - edit (redesign) map
 - field arrays
 - fields (define)
 - line arrays
 - list maps in the current library
 - menu options
 - multi-byte character set support
 - options (define)
 - page build map
 - protected fields on verification screen
 - protected redesign
 - redesign (modify) map
 - redesign (modify) using protected redesign
 - resident maps
 - review a map
 - submit a map
 - submit a map with ADS
 - symbolic maps
 - user-defined arrays
 - verify a map
- MBCS (multi-byte character sets)
- messages
 - abend messages
 - audit trail, GTBCONV
 - error (\$*nnn*) messages
 - Password Administration messages
 - system (*Bnnn*) messages
- move a field or block of fields (map redesign)
- MOVE (GTBUTIL command)
- move library member
- multi-byte character sets

- ~N~
 - naming conventions
 - *ROW member name
 - *ROWCOL member name
 - *ROWSET member name

- ~O~
 - one-column tables
 - OPTION (input rules command)

- ~P~
 - PAD (input rules command)
 - page build map
 - password, set or member (add, change or delete)
 - passwords, system
 - add CPU ID
 - change SYSPASS or FEATURE
 - delete CPU ID
 - online password administration
 - PASSWRD (GTBUTIL command)
 - PRINT (GTBUTIL command)
 - print
 - directory list (DLIST)
 - member (PRINT)
 - PROD (GTBUTIL command)
 - protected fields on Map Verification screen
 - protected redesign of a map
 - PUNCH (GTBUTIL command)

- ~R~
 - range checks (input rules)
 - redefine fields on a map
 - redesign
 - protected redesign
 - REJ (input rules command)
 - RENAME (GTBUTIL command)
 - rename library member
 - reserved words used with input rules
 - with EDIT command
 - with format dates input rule
 - RESET (GTBUTIL command)
 - reset library member
 - resident maps
 - RESTORE (GTBUTIL command)
 - RETN (input rules command)
 - return custom error messages (input rules)

- ~S~
 - SEL (input rules command)
 - selection field (branched help screens)
 - SETFILE (GTBUTIL command)
 - special sets
 - STRIP (input rules command)
 - submit
 - map

- map with ADS
- table
- symbolic maps
- SYSPASS (online password administration)
- system messages

~T~

- table lookups (input rules)
- tables
 - argument-only
 - argument/object
 - Assembler call or link to BMS/TS table processor
 - batch job submission
 - COBOL call or link to BMS/TS table processor
 - data types and lengths (in tables)
 - define
 - update
 - design
 - list tables in the current library
 - load module tables
 - one-column tables
 - online tables
 - menu options
 - parameters
 - Assembler programs
 - COBOL programs
 - redesign (update)
 - sample tables
 - Assembler
 - COBOL
 - submit
 - test a table
 - two-column tables
 - using tables in your application
- TAPESEL (GTBUTIL command)
- Technical support, GT Software
- template generation
- transaction codes for BMS/TS
- Transaction Monitor, edit
 - add help screens
 - implement input rules
- Transfer members between libraries
 - copy member
 - fetch BMS map
 - move member
- TROFF (input rules command)
- TRON (input rules command)
- two-column tables

~U~

User-defined arrays

~V~

[verify a map](#)
[design](#)
[help screen](#)

[Top of page](#)

[User's Guide Home](#)
[BMS/TS Doc. Home](#)

Copyright GT Software, Inc.