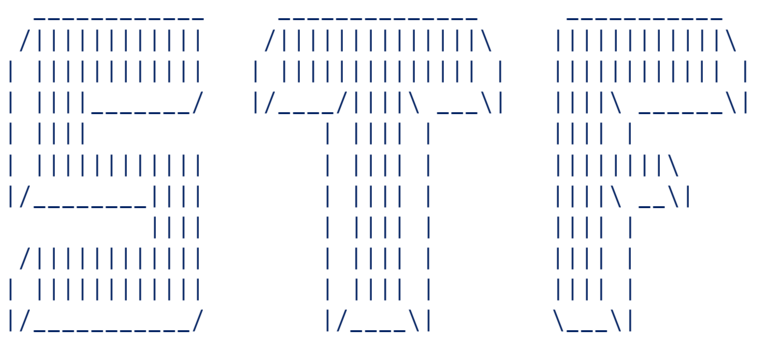

Skybird Test Facility
Version 5, Release 0.2
Customization Guide

Contents

List of figures

About this guide

The customization guide describes how to modify STF Help files and STF Exits to your personal needs.

This document was last updated in January, 2026. The latest version of this guide will also be available at the STF website: https://www.skybird.net/stf/

Introduction

The customization guide is intended for those who wish to play around with the provided STF source code.

Customize STF

With STF Version-5, Rel. 0.2, no customization is required, but if you wish you can add or update the Help information. Also, there are four exits available for accounting and security.

STF Help files

The supplied Help files are stored in the PDS allocated to STFHELP DD statement (default: STF.HELP). All help members are plain text files with LRECL=80. The maximum number of lines for one member is 133. Although each line contains 80 characters, only the first 72 characters are displayed by STF. So the last 8 characters in each line should be blank.

Help members can be added or updated using the EDIT command within STF:

EDIT DSN=STF.HELP,M=helptext, where M= specifies the Help member name.

If the member name supplied does not exist, a new member is created: within EDIT enter:

I *,n (where n= number of lines to insert) to insert a few blank lines.

Enter Help DIR for a list of existing Help members.

I suggest you update the Help member: NEWS with your own text.

STF Exits

The supplied exits are all dummy exits except for STFEX01 which is used to verify userid/password during logon. If the (supplied) exits are not used, dummy exits that give a return code of 0000 must be installed to prevent an 806-ABEND.

STFEX00 is invoked by STFMAIN during STF initialization,

STFEX01 is invoked by STFLGN during LOGON,

STFEX02 is invoked by STFMAIN after LOGOFF,

STFEX03 is invoked by STFMAIN after STF initialization/termination,

STFEX04 is invoked by STFCMD before a command is executed.

STFEX00 - The exit is invoked during STF initialization. The exit may be used to control STF execution. If a return code of 0008 is returned, STF will not initialize. Instead STF terminates with an error message (supplied by the exit). The MVT can be used to check how STF was invoked, MVTTSK contains either the STC name, the JOB name or the TSU name, depending on how STF was started.

Upon entry register 1 contains the address of the following parmlist:

```
AL4(STFMVT)
AL4(msgarea) 80-bytes
```

Before returning the exit should load register 15 with one of the following return codes:

```
RC= 0000 - ACCESS ALLOWED, NO MESSAGE,
RC= 0004 - ACCESS ALLOWED, MESSAGE,
RC= 0008 - ACCESS NOT ALLOWED, MESSAGE.
```

The message area can be used to pass a message to STFMAIN, if used (rc= 0004 or 0008) the message is sent to the STF operator.

STFEX01 - This exit is invoked by STFLGN during LOGON of an STF user. The exit may be used to check if the user is allowed to use STF. If a return code of 0008 is passed back, the LOGON will be rejected and access to STF is denied. The fields from the TVB and TIB may be used to verify the LOGON.

Upon entry register 1 contains the address of the following parameter list:

```
AL4(STFTVB)
AL4(msgarea1) 80-bytes (SAF message)
AL4(msgarea2) 80-bytes (STF message)
```

From the TVB the TVBTOAA field points to the task output area. At offset +8 in the output area is the address of the first entry in the output area. This entry contains the old password (8 bytes) and optionally the new password (8 bytes) and the verification password (8 bytes). If no new password was specified, this field contains blanks or hex-nulls. As soon as the exit returns control to STF, these fields are cleared by STF.

Before returning the exit should load register 15 with one of the following return codes:

```
RC= 0000 - ACCESS ALLOWED, NO MESSAGE,

RC= 0004 - ACCESS ALLOWED, MESSAGE,

RC= 0008 - ACCESS NOT ALLOWED, MESSAGE.
```

The message areas can be used to pass a message back to the STF user who tries

to LOGON.

STFEX02 - This exit is invoked when a task terminates. It may be used for accounting purposes and/or task cleanup. Return codes passed back to STF are not checked, but should always be zero.

Upon entry register 1 contains the address of the following parmlist:

```
AL4(STFTVB)
```

STFEX03 - This exit is invoked after STF has successfully initialized and before STF terminates. The exit may be used for accounting purposes. Return codes passed back to STF are not checked, but should always be zero.

Upon entry register 1 contains the address of the following parmlist:

```
AL4(STFMVT)
```

STFEX04 - This exit is invoked before a command is executed. The exit can be used for command authorization. If a non-zero return code is passed back to STF the command will not be executed.

Upon entry register 1 contains the address of the following parmlist:

```
AL4(STFTVB)
AL4(buffer) 86-bytes
```

The BUFFER contains the command in bytes 1-8 (padded with blanks) and operands (if any) in bytes 9-86.

STF Screen files

STF screen files are used with STFACQ, STFISC and STFTEST. The screens are located in STFLOAD and contain 3270 test screen definitions. The naming convention for screen files is SCRN@xxx where xxx is A-Z, 0-9 and $, # or @. The following screens are supplied with STF 5.02:

```
SCRN@ATR, SCRN@CHR, SCRN@EDS, SCRN@STF and SCRN@SYM.
```

SCRN@ is a sample user defined screen. The source code can be modified as needed. Because STFACQ, STFISC and STFTEST add a header line, row 01 should not be used in user screen definitions. STFTEST (TEST command) also adds a message, status and command line, therefor row 02, 22, 23 and 24 should not be used also. So, to be on the safe side: only use row 03 through 21 in user screen definitions.

STF Control Blocks

Control Block overview.

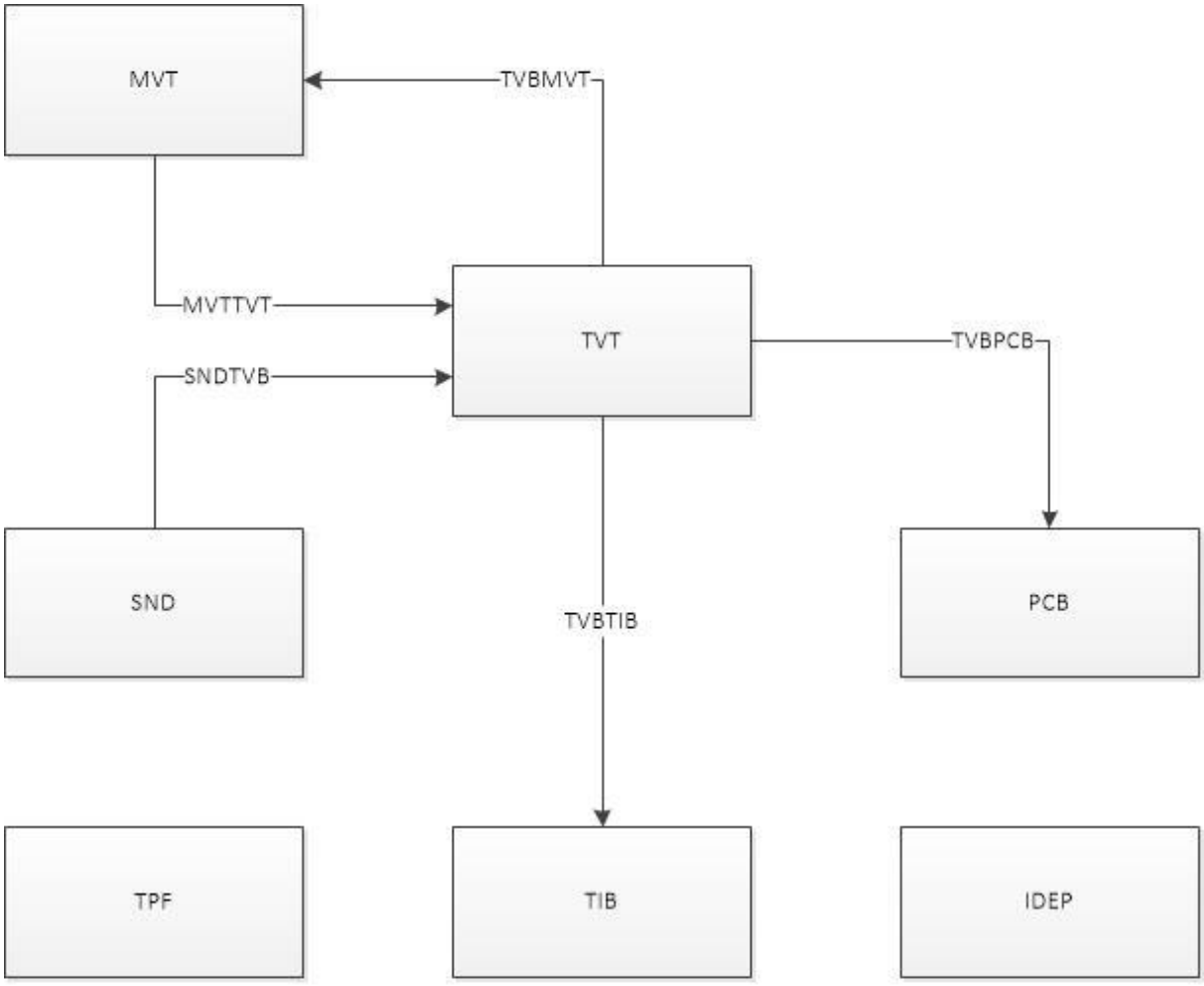

Figure 1 - Control block overview

Control block layout

The STF Common area, MVT and TVT are allocated in contiguous storage, located in STFMAIN.

The other control blocks STFPCB, STFSND, STFREQ, STFTPF and STFIDEP are used (and allocated) by individual modules.

The common area is not a separate control block, but instead it is located at the

beginning of the MVT. In addition to the fields specified below, the common area also contains equates that specify the BINDSIZE, LOGNSIZE, TSKISIZE and TSKOSIZE. The MVT macro variable &NUMBER specifies the maximum number of tasks. This number must match the number of task ECB entries defined in the ECBLIST in STFMAIN.

Common area lay-out:

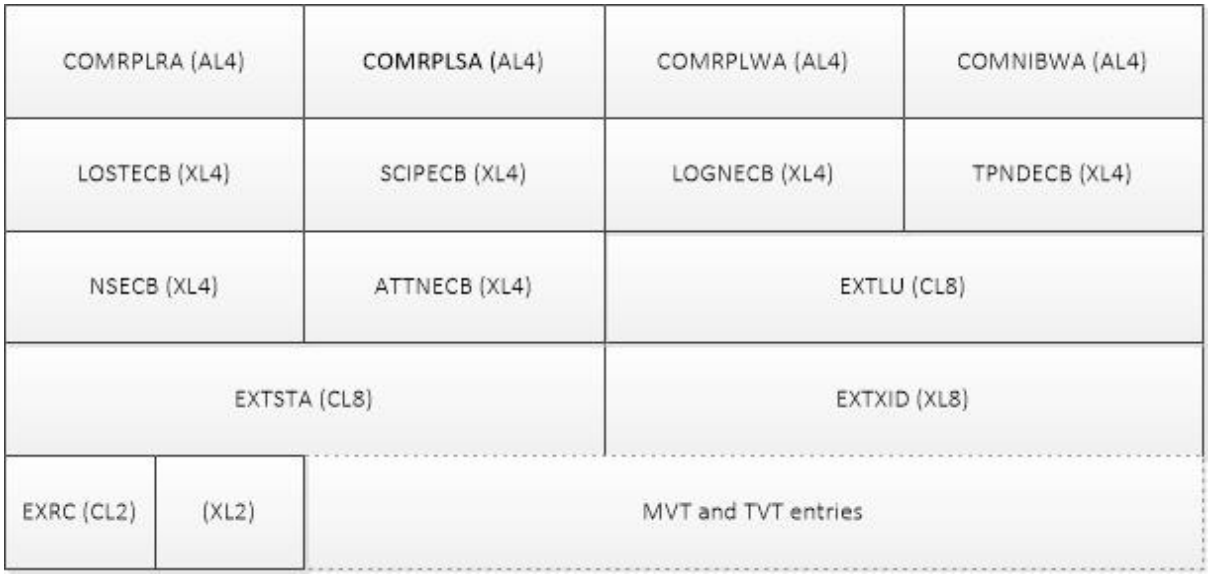

Figure 2 - Common area

The control blocks are mapped by the following macros:

STF$IDEP maps the ID-Entry Point table (start/stop a task)

STF$MVT maps the Main Vector Table (main pointers)

STF$TVB maps the Task Vector Table entries (ECB's)

STF$TIB maps the Task Information Block (task related pointers)

STF$PCB maps the Parse Control Block (communication between tasks)

STF$SND maps the Send Control Block (Vtam Send macro)

STF$TPF maps the Text Processing Facility Control Block used by STFEDT

Control block usage

Control Block:	Usage:
STFMVT	Contains global fields, equates and a pointer to TVT. The MVT is created at start-up by STFMAIN. The MVT is freed at closedown by STFMAIN.
STFTVT	Contains a TVT entry for every (sub)task. Each TVT entry is called Task Vector Block and is mapped by the TVB (STF$TVB). The TVT is created at start-up by STFMAIN. The TVT is freed at closedown by STFMAIN.
STFTVB	The Task Vector Block is used by ALL modules to map their corresponding TVT entry. The TVB is used to pass REQUESTS and RESPONSES between a sub-TASK and the MAIN module. REQUESTS are sent from a TASK to MAIN, RESPONSES are sent from MAIN to a TASK. The TVB also contains all ECB's and pointers to MVT, TIB and PCB. For communication between tasks two area's are used: INPUT (TIBTIAA) and OUTPUT (TIBOAA) area. The INPUT area contains input data for the TASK (this is output from the command module) and the OUTPUT area contains output from the TASK (which is input to the command module). When a command is entered all device dependent control characters are removed and the command is placed in the OUTPUT area. Then the TASK issues a LINK to the command module (STFCMD) which formats the command, scans its command table, checks the authority and then issues a LINK to the module that actually processes the command and places the requested data into the INPUT area. The TVT is created at start-up by STFMAIN. The TVT is freed at closedown. Each TVT entry (TVB) is (re-)initialized by STFMAIN every time a TASK using the TVB is started or stopped.
STFTIB	The Task Information Block contains the Identification and Status data of a TASK. The TIB is created by STFMAIN when a TASK is started. The TIB is freed by STFMAIN when a TASK is stopped.
STFPCB	The PCB is used to pass start-up parameters when a task is started and by tasks or command modules to pass data to STFPRS (parse

routine). The Parse Control Block is mapped by STF$PCB.
The PCB is created by STFMAIN at start-up time.
The PCB is freed by STFMAIN when STF is stopped.

STFSND — Control block, supplied by calling task to pass SEND parameters to and from STF#SND.

IDEP — The ID-EntryPoint table is mapped by STF$IDEP and contains a list of STF (sub)tasks and their attributes. It is used when a task is started.

STFTPF — The Text Processing Facility control block is used internally by STFEDIT.

Control block fields.

Each storage area (Task Input Area, and Task Output Area) has a 20-byte prefix describing the storage.

Storage area prefix:

```
+0   total length storage area
+4   indicators;
     byte 0  1... ....  send clear screen
             .1.. ....  full screen message
             ..xx ....  reserved
             .... xxxx  reserved
     byte 1  .... ....  copy of SWFUNC
             1... ....  SEND
             .1.. ....  INT_MSG
             ..1. ....  MODIFY (F)
             ...1 ....  INT_RSP
             .... 1...  INT_LOG
             .... .1..  ROUTE
             .... ..Xx  reserved
     byte 2  xxxx xxxx  reserved
     byte 3  xxxx xxxx  reserved
+8   address 1st entry
+12  length 1 entry
+16  address last entry
+20  start usable storage (1st entry)
```

STFPRS (parse) uses a keyword table to parse a command. The keyword table is supplied by the module calling the parse routine.

Keyword table prefix:

```
+0   AL4  Address first keyword entry
+4   AL4  Length one entry
+8   AL4  Address last keyword entry
```

Keyword table entry lay-out:

```
+0   CL8  keyword
+8   CL1  A/N/X    indicator (A=alphanumeric, N=numeric, X=hex)
+9   CL1  R        indicator (R=required field)
+10  XL2  Flags:   0000= CAPS, 0100= asis
+12  AL4  address operand entry
```

Operand entry lay-out:

```
+0   XL2  Indicators;
          byte 0  1... .... operand supplied
                  .xxx xxxx reserved
          byte 1  xxxx xxxx reserved
+2   AL4  Length operand
+6   C    Operand (variable length)
```

STF#FMT (format) uses a format table to move data from input area to output area. The address of this table must be supplied by the calling program.

Format table prefix:

```
+0   AL4  Address first format entry
+4   AL4  Length one entry
+8   AL4  Address last format entry
```

Format table entry lay-out:

```
+0   XL2  SBA or LABEL
+2   XL2  Flags: 0000= CAPS, 0100= asis
+4   XL2  length output field
+6   AL4  address output field
```

Message queue prefix lay-out:

```
+0   AL4  start of free entry chain
+4   XL4  lockword (= TVB address of owning task)
+8   AL4  start of msg entry chain
```

Message queue element lay-out:

```
+0   AL4  address next element
+4   XL4  indicators;
          byte 0  1... ....  active entry
                  .xxx xxxx  reserved
          byte 1  0000 0000  unknown message type
```

```
                  1... ....  SEND
                  .1.. ....  INT_MSG
                  ..1. ....  MODIFY
                  ...1 ....  INT_RSP
                  .... 1...  INT_LOG
                  .... .1..  ROUTE
                  .... ..xx  reserved
          byte 2  xxxx xxxx  reserved
          byte 3  xxxx xxxx  reserved
+8   CL8  destination task
+16  CL8  origin task
+24  CL80 message text
```

TVBUSE	x'..'	Entry usage
	x'00'	Entry free
	x'80'	Entry active
	x'06'	APPL entry (STF to STF communication task)
	x'03'	Entry in use by START processing
	x'02'	Entry in use by user LOGGED-ON
	x'01'	Entry in use by task ATTACHED
	x'FF'	MAIN entry
TVBREQ/TVBRSP	x'..'	Request/response codes
	x'01'	TVB@STA - start
	x'02'	TVB@LGN - logon
	x'03'	TVB@STO - stop
	x'04'	TVB@LGF - logoff
	x'05'	TVB@CNL - cancel
	x'07'	TVB@UNB - unbind
	x'08'	TVB@CLS - close
	x'10'	TVB@PCMD- exit, pass command
	x'12'	TVB@DMP - dump
	x'13'	TVB@SIM - simlogon
	x'20'	TVB@FSCN- reset after fullscreen command
	x'31'	TVB@RLS - release
	x'33'	TVB@HLD - hold
	x'C0'	TVB@RQS - request session
	x'C1'	TVB@BND - bind
	x'C2'	TVB@SDT - start data traffic
	x'C3'	TVB@CLN - NSEXIT: cleanup
	x'C4'	TVB@NSP - NSEXIT: NSPE
	x'C5'	TVB@NFY - NSEXIT: Notify
	x'C6'	TVB@BDF - SCIP: Bind Failure (OPNSEC failed)
	x'C7'	TVB@STS - STSN
	x'C8'	TVB@LST - LOSTTERM

```
                x'EF'    TVB@RTY - retry

TIBSTA          x'....'  Task status indicator
                x'00..'  Task not active
                x'01..'  Task active
                x'03..'  Task HELD
                x'04..'  Task LOCKED
                x'..01'  Starting - ATTACH (Start command)
                x'..02'  Starting - LOGON  (Vtam logon)
                x'..03'  Stopping - TIB@STO - STOP (Main requested
                                             stop)
                x'..04'  Stopping - TIB@LGF - LOGOFF (Task
                                             requested stop)
                x'..05'  Stopping – TIB@UKN - Unknown response
                                             received
                x'..06'  Stopping - TIB@SES - Session parameters
                                             not supported
                x'..07'  Stopping - TIB@ERR - Unrecoverable RPL
                                             error occurred
                x'..09'  Stopping - TIB@TRM - Terminated (with
                                             errors)
                x'..10'  Stopping - Abend/cancel
                x'..B0'  Logon - okee
                x'..B1'  Logon - okee, message
                x'..B2'  Logon - not allowed, message

TIBTFLG         x'..'    Task flags
                x'80'    TIB@RST - Restart after abend
                x'40'    TIB@SYS - System task
                x'20'    TIB@MSG - Message receiver
                x'01'    TIB@VFY - User has been verified
                x'02'    TIB@DUP - Duplicate USERID allowed

TIBVFLG         x'..'    VTAM flags
                x'80'    TIB@ACQ - Terminal acquired (ACB open)
                x'40'    TIB@PND - session pending (logon)
                x'20'    TIB@SDT - SDT (start data traffic
                                   received)
                x'10'    TIB@RCV - Receive pending
                x'01'    TIB@SIM - Simlogon pending
```

Contact information

For the latest news and documentation updates visit the STF Homepage:
http://www.skybird.net/stf/

Consider joining the following MVS user group:

https://groups.io/g/turnkey-mvs/