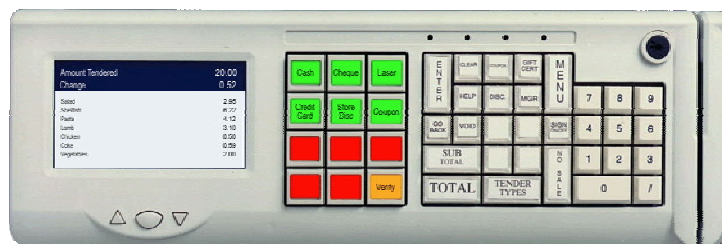




ScreenKey Consoles for the IBM 46xx

User Exit Integration User's Guide Version 3.20

Issue 1.3 March 2006



Purpose

The purpose of this document is to provide instructions on how to install the SK-7510 User Exits Kernel support in IBM 46xx Terminal Sales Applications.

This document is based on User Exits Kernel release 3.20.

www.ScreenKeys.com

SKI

ScreenKey Interfaces Ltd.

User Exits Integration for the SK-7510

Information in this document is subject to change without notice.

The latest revisions of the ScreenKey documents and software can be found on the ScreenKeys web site.

Web: www.ScreenKeys.com

Technical Support is available

via Email: support@ScreenKeys.com

via Web: www.ScreenKeys.com

© 2002-2006 SK Interfaces Ltd.

© 1996-2002 Rapid Technology Interfaces Ltd.

All rights reserved.

DISCLAIMER:

SKI reserves the right to revise data file formats and functionality at any time.

Table of Contents

1	INTRODUCTION	5
1.1	SCREENKEY USER EXITS SUPPORT KERNEL	6
2	DISTRIBUTION FILES	7
3	SUPERMARKET APPLICATION INSTALLATION	9
3.1	RUNTIME MODULE LINKER INCLUDES	9
3.2	USER EXIT INCLUDES	9
3.3	TERMINAL SALES COMPILATION	12
3.4	LINK AND POSTLINK SUPERMARKET APPLICATION	12
3.5	ACTIVATE THE SELECTED USER EXITS IN TERMINAL OPTIONS	12
4	GENERAL SALES APPLICATION INSTALLATION	13
4.1	RUNTIME MODULE LINKER INCLUDES	13
4.2	USER EXIT INCLUDES	13
4.3	TERMINAL SALES COMPILATION	16
4.4	LINK AND POSTLINK GSA APPLICATION	17
5	CHAIN SALES APPLICATION INSTALLATION	19
5.1	RUNTIME MODULE LINKER INCLUDES	19
5.2	USER CODE INCLUDES	19
5.3	TERMINAL SALES COMPILATION	21
5.4	LINK AND POSTLINK CSA APPLICATION	21
6	THE RUNTIME PROCEDURE - DOWNLOADS	22
6.1	DOWNLOAD FILES	22
6.2	TRACEOUTPUT	22
6.3	RTITERMS.DAT TERMINAL LISTING FILE	23
6.4	RTISESSN.DAT SESSION NUMBER ASSIGNMENT	24
6.5	RTIKBNNN TERMINAL SPECIFIC PARAMETERS	25
6.6	RTIMSG.DAT - MESSAGE TEXT FILE	26
6.7	CONTROLLER SET-UP	27
6.7.1	<i>User Logical File Name Set-up</i>	28
6.7.2	<i>File Attributes Set-up</i>	28
7	THE RUNTIME PROCEDURE - API	29
7.1	ESCAPE SEQUENCE SUMMARY	29
7.1.1	<i>TDA\CDA Commands</i>	29
7.1.2	<i>LCD Panel Commands</i>	29
7.1.3	<i>ScreenKey Commands</i>	30
7.1.4	<i>Keyboard Commands</i>	30
8	API CUSTOMISATION EXAMPLES FOR SUPERMARKET APPLICATION	31
8.1	ECHO ELECTRONIC JOURNAL DATA TO TDA IN REAL TIME (POST PRINTING SETUP)	32
8.2	ECHO CASH RECEIPT DATA TO TDA	33
8.3	SHOW SPLASH SCREEN WHEN SIGNED OFF	34
8.4	CLEAR TDA AT START OF NEW TRANSACTION	35
8.5	SHOW FOODSTAMP TOTAL ON THE CDA	36
9	PROBLEM DETERMINATION	39
9.1	APPLICATION EVENT LOG	40
9.1.1	<i>Event Reference and the Application Event Log</i>	40
9.2	TRACEOUTPUT	41

9.2.1	Trace to File	42
9.2.2	Trace to Display	42
9.2.3	Trace to Printer	42
9.3	REQUIRED FILE ERROR HANDLING	43
9.4	DOWNLOAD PROGRESS REPORTING ON SCREENKEYS.....	44
9.4.1	Download Status Reported on ScreenKeys.....	44
10	MESSAGES SUMMARY	45
10.1	TRACELEVEL 0 MESSAGES SUMMARY	45
10.2	TRACELEVEL 1 MESSAGES SUMMARY	46
10.3	TRACELEVEL 2 MESSAGES SUMMARY	47
11	MESSAGE DESCRIPTIONS.....	49
APPENDIX A	DOCUMENTATION CONTROL	61
APPENDIX B	FILE DOWNLOAD UTILITY V USER EXIT INTEGRATION.....	63
USER EXITS INTEGRATION CHECKLIST		65

1 Introduction

The User Exits Kernel is written in CBASIC. Its first purpose is to download operational files to the non-volatile RAM of a SK-7510 ScreenKey keyboard attached to an IBM 46xx terminal. The operational files are the:

Code Update	(.CDL) keyboard firmware
Option file	(.ODL) terminal specific configurations
SAC file	(.SDL) menu navigation and key return codes

See the SK-7510 Technical Reference Manual for full description of the Operational Files.

Its second purpose, referred to as the API Channel, is to allow the Terminal Sales Application (Supermarket Application, General Sales Application or Chain Sales Application) to send commands to the SK-7510. These commands will display information on the ODA or display a ScreenKeys Menu among other things.

The User Exits Kernel is the recommended approach to ensuring that these Operational Files are always up to date in the SK-7510. The alternative approach of using the File Download Utility (FDU) is an efficient way for System Integrators to try out the SK-7510 ScreenKey Keyboard in a Lab. As a standalone application it doesn't involve any changes to the Terminal Sales Application (TSA).

However, using the FDU is not the recommended approach when installing the SK-7510 in a pilot store or in a rollout. The recommended approach is (having proven the concept in the lab using the FDU) to integrate the SK-7510 User Exits Kernel into the Terminal Sales Application. Integration means the download feature is available anytime the TSA requires it whereas to run the FDU you must terminate the TSA, run the FDU and then restart the TSA.

Both the FDU and the User Exits Kernel use the same Configuration files (RTITERMS.DAT, RTISESSN.DAT and RTIKBnnn) and the same Support file (RTIMSG.DAT). So, if these files are set-up to suit the FDU they can be used with the User Exits Kernel. The only word of warning is in relation to the RTISESSN.DAT file. This file contains Session Numbers which may be valid when used by the FDU standalone application but may clash with TSA session numbers when the User Exits Kernel is integrated into the TSA.

See "[APPENDIX B File Download Utility v User Exit Integration](#)" for a comparison of the FDU and User Exits Integration.

1.1 ScreenKey User Exits Support Kernel

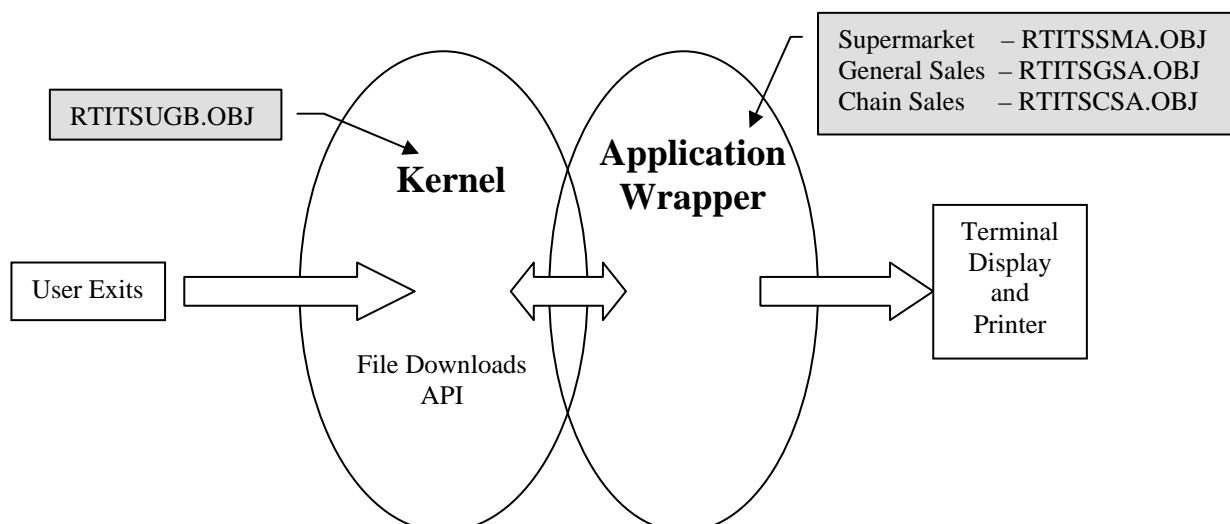
CBASIC runtime modules are available that extend Supermarket Application (SA), General Sales Application (GSA) and Chain Sales Application (CSA) to support the SK-7510 ScreenKey Keyboard:

- ScreenKey Keyboard file downloads (code updates, options and SAC files)
- Applications Programming Interface (API) to control TDA and SAC menu display

The SK-7510 may be tightly integrated into users sales applications using the API. Examples include:

- Real-time display of items sales data on the TDA with or without post-printing
- Only show current transaction on the TDA
- Display splash screen on TDA when operator signed-off
- Show food-stamp running total on the CDA

A generic runtime kernel module is provided that handles the main interaction with the keyboard and performs file downloads. Two application specific 'wrappers' are provided that handle access to the terminal printer and display for kernel functions and control keyboard specific error handling. The following diagram illustrates:



Using a simple function call into the kernel module from an appropriate User Exit (SA or GSA) or from a suitable core module (CSA), enables the sales application to monitor the revision levels of operational files currently active in the ScreenKey console and to compare these against the latest files available for download from the controller. The file revision levels are checked for SAC, Options and Code. If the controller file revision differs from the active downloaded file in keyboard memory the User Exits initiate and control the download process.

The same kernel function also checks for the presence and activation of the keyboard API channel. Application developers/integrators can extend the functionality and effectiveness of the SK-7510 within SA/GSA using the API to control the TDA and to call-up specific SAC menus on the ScreenKeys at appropriate points.

The API is particularly useful when post-printing is enabled and combined with the 4610 thermal printer. The API allows item sales data to be shown immediately on the TDA although nothing is printed to the cash receipt until the transaction is completed. Similarly, electronic journal data may be displayed on the TDA via the API.

Note: Both large and medium memory modules are supplied. Large memory modules have the extension .LBJ and medium memory models have the extension .OBJ.

2 Distribution Files

The following files are provided:

<i>Core Library Functions</i>		
<i>Runtime Module</i>	<i>Purpose</i>	<i>TS Application</i>
RTITSUGB.OBJ	Support kernel routines (medium memory model)	All
RTITSUGB.LBJ	Support kernel routines (large memory model)	All
RTITSUGB.F86	External function definitions to include in User Exit code when kernel functions are called	All
RTITSUGB.V86	Global variable definitions required by RTITSUGB module	All
MGVTSASC.OBJ	Assembler routines for medium memory model	All
MGVTSASB.LBJ	Assembler routines for large memory model	All

<i>Data/Support Files</i>		
<i>Support File</i>	<i>Purpose</i>	<i>TS Application</i>
RTISESSM.C86	Main session number file (%INCLUDE in user exit)	All
RTIMSG.DAT	Error and progress descriptors (fixed format)	All
RTISESSN.DAT	Additional session number setup file	All
RTIKB000	Terminal-specific configuration file	All
RTITERMS.DAT	Terminal numbers to which ScreenKey keyboards are attached	All

<i>Application Wrapper Files</i>		
<i>Wrapper File</i>	<i>Purpose</i>	<i>TS Application</i>
RTITSSMA.OBJ	Supermarket wrapper with SA-specific functions and variables required by RTITSUGB.OBJ (medium memory)	SA
RTITSSMA.LBJ	Supermarket wrapper with SA-specific functions and variables required by RTITSUGB.LBJ (large memory)	SA
RTITSGSA.OBJ	General Sales wrapper with GSA-specific functions and variables required by RTITSUGB.OBJ (medium memory)	GSA
RTITSGSA.LBJ	General Sales wrapper with GSA-specific functions and variables required by RTITSUGB.LBJ (large memory)	GSA

Note:

*Chain Sales Application wrapper files are **not** distributed. These files must be created unique to each CSA installation.*

<i>Supermarket Integration Examples</i>		
<i>Module</i>	<i>Purpose</i>	<i>Applicable SA Module</i>
RTISMA07.C86	%INCLUDE this file in SA terminal sales user exit 7 to call initialisation, file check and download routine Terminal Sales User Exit 07 is activated at IPL and whenever application personalization terminal options are reloaded.	EAMTSU07.J86
EAMTSU07.J86	Sample code demonstrating how to include C86 file in SA user exit 7 for initialisation, file checking and download	EAMTSU07.J86
RTISMA27.C86	%INCLUDE this file in SA terminal sales user exit 27 to call specific error handling Terminal sales user exit 27 is activated at each error occurrence and can be used to trap specific errors for alternative actions	EAMTSU27.J86
EAMTSU27.J86	Sample code demonstrating how to include C86 in SA user exit 27 for specific error handling	EAMTSU27.J86
EAMTSUSU.J86	Sample code demonstrating where to include RTITSUGB.F86 file for SA user exit integration	EAMTSUSU.J86
EAMTSUVA.J86	Sample code demonstrating where to include RTITSUGB.V86 file for SA user exit integration	EAMTSUVA.J86

<i>General Sales Application Integration Examples</i>		
<i>Module</i>	<i>Purpose</i>	<i>Applicable GSA Module</i>
RTIGSA04.C86	%INCLUDE this file in GSA terminal sales user exit 4 to call specific error handling Terminal sales user exit 04 is activated whenever an error is detected	EALTSU04.BAS
EALTSU04.BAS	Sample code demonstrating how to include C86, F86 and V86 files in GSA user exit 4 for specific error handling	EALTSU04.BAS
RTITSGSA.V86	%INCLUDE this file each time RTIGSA04.C86 is included. It defines the local variables used by GSA error handling.	EALTSU04.BAS
RTIGSA16.C86	%INCLUDE this file in GSA terminal sales user exit 16 to call initialisation, file check and download routine Terminal sales user exit 16 is activated after all files have been opened	EALTSU16.BAS
EALTSU16.BAS	Sample code demonstrating how to include C86, F86 and V86 files in GSA user exit 4 for initialisation, file checking and download	EALTSU16.BAS

3 Supermarket Application Installation

First determine if you are using the large or medium memory model application (EAMTS10L.286 is large and EAMTS11L.286 is medium memory application).

3.1 Runtime Module Linker Includes

The runtime core and wrapper modules must be included in the appropriate linker file:

EAMTS10L.INP	Large memory
EAMTS11L.INP	Medium memory

Edit this file to add the following lines:

EAMTS10L.INP (Large)	EAMTS11L.INP (Medium)
RTITSSMA.LBJ[M,LI,LO],	RTITSSMA[M,LI,LO],
RTITSUGB.LBJ[M,LI,LO],	RTITSUGB[M,LI,LO],
MGVTSASB.LBJ[M,LI,LO],	MGVTSASC[M,LI,LO],

Copy the relevant files (LBJ or OBJ extension) from the distribution diskette to the development directory.

3.2 User Exit Includes

Having included the runtime support files, the appropriate user exit must be selected from which to call the SKI function to check file versions and download new files if required. For SA, an appropriate user exit is terminal sales user exit 7. This user exit (UE) is activated at IPL and whenever application personalization of terminal options is reloaded.

Add the highlighted line to EAMTSU07.J86 (an example EAMTSU07.J86 is included on the distribution diskette):

```

\* TIME STAMP BLOCK *****
\* END OF TIME STAMP BLOCK *****/

FUNCTION TSUPEC07 PUBLIC
! TSUPEC07- AFTER IN
%INCLUDE RTISMA07.C86           ! SKI 7510 check and download
END FUNCTION
  
```

The RTISMA07.C86 file is included on the distribution diskette:

```

! Assign initial session number for ScreenKeyboard user exits to access
! external setup files
%INCLUDE RTISESSM.C86

! Initialise SKI keyboard and check if downloads required
CALL RTI.CHECKKEYBOARD           ! Initialization routines

!**** End of Module ****
  
```

The above file demonstrates how to access the core SKI SK-7510 ScreenKey keyboard function (RTI.CHECKKEYBOARD) that interrogates the keyboard via the assigned serial port and downloads new files if required.

This function call may be placed in any appropriate terminal sales user exit suitable to the sales application customisations already made. It must be preceded by the %INCLUDE RTISESSM.C86 which defines the unique session number assigned to the SKI keyboard (see below). This session number is hard-coded (compiled) into the sales application executable module (all other session numbers may be assigned using the RTISESSN.DAT file).

The terminal sales user exit error handling **must** also be modified to add support for handling specific errors.

Terminal Sales user exit 27 is where additional error handling code is inserted.

Add the highlighted line to EAMTSU27.J86 (an example EAMTSU27.J86 is included on the distribution diskette):

```
√* TIME STAMP BLOCK *****  
√** END OF TIME STAMP BLOCK *****/  
  
FUNCTION TSUPEC27 PUBLIC  
! TSUPEC27 - IO ERROR PROCESSING  
  
%INCLUDE RTISMA27.C86           ! SKI 7510 error handling  
  
END FUNCTION
```

The RTISMA27.C86 file is included on the distribution diskette:

```
!*****  
  
CALL RTI.HANDLERR(0,0,0)           ! Handle SKI 7510 errors  
  
!**** End of Module ****
```

The parameters passed to RTI.ERRHANDLE **must** always be set to 0,0,0.

Terminal Sales user exits in SA are separate include modules compiled together in EAMTSUPC.BAS. Global variables and external function declarations are specified in separate include files.

Add the highlighted line to EAMTSUSU.J86 to include the SKI external function declarations (an example EAMTSUSU.J86 is provided):

```
√* TIME STAMP BLOCK *****  
√** END OF TIME STAMP BLOCK *****/  
!  
!THIS INCLUDE FILE PROVIDES A PLACE FOR THE USER TO DEFINE SUBROUTINES  
!WHICH ARE COMMON TO SEVERAL OF HIS USER EXIT FUNCTIONS IN EAMTSUPC.  
!  
  
%INCLUDE RTITSUGB.F86           ! 7510 function declarations
```


3.3 Terminal Sales Compilation

After adding the above code to the user exit include files, compile EAMTSUPC.BAS using the 4680 BASIC compiler:

BASIC EAMTSUPC [fum(b)]	Large memory
BASIC EAMTSUPC [fu]	Medium memory

Note: If large memory is used, after the compile has completed successfully, rename EAMTSUPC.OBJ to EAMTSUPC.LBJ

3.4 Link and Postlink Supermarket Application

After successful compilation of the terminal sales module, the SA application should be linked:

LINK86 EAMTS10L [i]	Large memory
LINK86 EAMTS11L [i]	Medium memory

Note: Successful compilation may require files from multiple directories. Use the linker environment variable to define these paths, for example:

DEFINE LNK86PATH=C:\ADX_UPGM;C:\ADX_SPGM;

Before running on a terminal the 286 file should be postlinked to improve its load time and make memory usage more efficient in a multitasking environment:

POSTLINK EAMTS10L.286	Large memory
POSTLINK EAMTS11L.286	Medium memory

Before running the newly created application, the support and data files must be configured for the ScreenKey keyboard (see below). Additionally, logical filenames must be configured that defines the location of these files.

3.5 Activate the Selected User Exits in Terminal Options

Before loading the terminal with the application, the user exits must be enabled. In terminal options there is a YES/NO flag for each user exit. For example, user exits 7 and 27 mentioned above need to have their flags set to Y for YES.

To enable the user exits do the following:

1. From the *System Main Menu*, select 1 for the primary application and press ENTER
2. From the *Supermarket Application* menu select 6 for *Application Personalization* and press ENTER
3. From the *Application Personalization* menu select 1 for *Terminal Options* and press ENTER.
4. From the *Personalise Terminal Options* menu select 1 for *Change a Set of Options* and press ENTER
5. From the *Change Terminal Options* menu select 20 for *User Exits*, and press ENTER
6. When the User Exits option screen is displayed, TAB to the required user exit number and enter a Y. Repeat until all required user exits have been enabled. Then press ENTER to save the changes. Press F3 Quit to exit *Change a Set of Terminal Options*. The terminal application will use these new settings when it is next loaded.

4 General Sales Application Installation

First determine if you are using the large or medium memory model application (EALTS10L.286 is large and EALTS00L.286 is medium memory application).

4.1 Runtime Module Linker Includes

The runtime core and wrapper modules must be included in the appropriate linker file:

EALTS10L.INP	Large memory
EALTS00L.INP	Medium memory

Edit this file to add the following lines:

EAMTS10L.INP (Large)	EAMTS11L.INP (Medium)
RTITSGSA.LBJ,	RTITSGSA,
RTITSUGB.LBJ,	RTITSUGB,
MGVTSASB.LBJ,	MGVTSASC,

Copy the relevant files (LBJ or OBJ extension) from distribution diskette to the development directory.

4.2 User Exit Includes

Having included the runtime support files, the appropriate user exit must be selected from which to call the SKI function to check file versions and download new files if required. For GSA, one such user exit is terminal sales user exit 16. This user exit (UE) is activated at IPL after opening all files.

As each GSA user exit is a separate compilable module, the SKI global variables (RTITSUGB.V86) and external function definitions (RTITSUGB.F86) must also be included in the user exit BAS module.

Add the highlighted lines to EALTSU16.BAS (an example EALTSU16.BAS is included on the distribution diskette):

```

\* TIME STAMP BLOCK *****
\** END OF TIME STAMP BLOCK *****/

REM EALTSU16
\ REM!!***** OVERVIEW OF FUNCTION *****!
\ REM!! This function receives control after all files used by terminal!
\ REM!! sales are open to allow the user to open user files .      !
\ REM!!*****!

    %ENVIRON T

    %INCLUDE RTITSUGB.V86          ! SKI global variables

INTEGER*1 GLOBAL MICR.ENABLE

    %INCLUDE RTITSUGB.F86          ! SKI function definitions

FUNCTION FILE.OPEN.EXIT PUBLIC
INTEGER*2 FILE.OPEN.EXIT      ! Define function type
MICR.ENABLE = 0

    %INCLUDE RTIGSA16.C86          ! SKI keyboard check and download

END FUNCTION

```

The files RTIGSA16.C86, RTITSUGB.V86 and RTITSUGB.F86 are included on the distribution diskette.

RTIGSA16.C86:

```
!*****  
  
! Assign initial session number for ScreenKeyboard user exits to access  
! external setup files  
%INCLUDE RTISESSM.C86  
  
! Initialise SKI keyboard and check if downloads required  
CALL RTI.CHECKKEYBOARD                   ! Initialization routines  
  
!**** End of Module ****
```

The above file demonstrates how to access the core SKI SK-7510 ScreenKey keyboard function (RTI.CHECKKEYBOARD) that interrogates the keyboard via the assigned serial port and downloads new files if required.

This function call may be placed in any appropriate terminal sales user exit suitable to the sales application customisations already made. It must be preceded by the %INCLUDE RTISESSM.C86 which defines the unique session number assigned to the SKI keyboard (see below). This session number is hard-coded (compiled) into the terminal sales user exit executable module (all other session numbers may be assigned using an external DAT file).

Note: do not forget to %INCLUDE RTITSUGB.V86 and RTITSUGB.F86 in each user exit module that accesses kernel functions.

Another suitable user exit in which the call to RTI.CHECKKEYBOARD might be placed is terminal sales user exit 23 (Operator Selected User-Written Nonsales Transaction) , whereby a key assignment could be created to allow a manager to check the keyboard status and download new files if required from a non-sales keying sequence.

The terminal sales user exit error handling **must** also be modified to add support for handling specific errors.

Terminal Sales user exit 4 is executed on every error detected in GSA and is where additional error handling code is inserted.

Add the highlighted lines to EALTSU04.BAS (an example EALTSU04.BAS is included on the distribution diskette):

```
\* TIME STAMP BLOCK *****
\* END OF TIME STAMP BLOCK *****/

REM=====

%ENVIRON T

%INCLUDE RTITSUGB.V86      ! SKI global variable
%INCLUDE RTITSGSA.V86    ! SKI local variables

STRING GLOBAL TS.TS10WERR$      ! error type
INTEGER*2 GLOBAL TS.ESESSION    ! failing session number
INTEGER*4 GLOBAL TS.TS10ERRN    ! error return code

%INCLUDE RTITSUGB.F86      ! SKI function definitions

REM=====
REM
REM  ERROR.EXIT      called from main on error in EALTS10C
REM
REM=====

SUB ERROR.EXIT( USER.RESUME,USER.RETRY,USER.LOG) PUBLIC
  INTEGER*1 USER.RESUME,      \ If true,resume immediate after exit
    USER.RETRY,              \ If true,resume retry immediate
    USER.LOG                  ! If true, log standard format error
                              ! to application event log.

  %INCLUDE RTIGSA04.C86      ! SKI keyboard error handling

END SUB
END
```

The global variables and global function definitions files (RTITSUGB.V86 and RTITSUGB.F86 respectively) are included on the distribution diskette. RTITSGSA.V86 file defines local variables used by the SKI error handling routine. RTIGSA04.C86 file is included on the distribution diskette:

```
!*****
CALL RTI.HANDLERR(RT.RESUME,RT.RETRY,RT.LOG) ! Handle rt errors

USER.RESUME = RT.RESUME      ! Set GSA variables
USER.RETRY  = RT.RETRY      ! to the values returned
USER.LOG    = RT.LOG        ! by the SKI sub program

!**** End of Module ****
```

Before proceeding to the compilation stage, check the session number assignment in module RTISESSM.C86. This must be unique for your sales application:

```

!*****
!
! RTISESSM.C86 - Assign unique session number for SKI keyboard
!      usage.
!
! This CODE file should be included prior to calling main keyboard
! support routine, i.e. RTI.CHECKKEYBOARD.
!
! This file is distributed with the 7510 kernel.
!
!*****

RTI.SESS.FILE% = 50          ! File access session number

!***** End of RTISESSM.C86 *****

```

Note:
This file is distributed with a default value for RTI.SESS.FILE% set to 50. This may not be appropriate for GSA. A more suitable value for GSA might be 30.

4.3 Terminal Sales Compilation

After adding the above code to the user exit modules, compile each module using the 4680 BASIC compiler, for example:

BASIC EALTSU04 [fum(b)]	Large memory
BASIC EALTSU16 [fum(b)]	
BASIC EALTSU04 [fu]	Medium memory
BASIC EALTSU16 [fu]	

Note: If large memory is used, after the compile has completed successfully rename EALTSUxx.OBJ to EALTSUxx.LBJ

4.4 Link and Postlink GSA Application

After successful compilation of the terminal sales user exit modules, the GSA application should be linked:

LINK86 EALTS10L [i]	Large memory
LINK86 EALTS00L [i]	Medium memory

Note: Successful compilation may require files from multiple directories. Use the linker environment variable to define these paths, for example:

```
DEFINE LNK86PATH=C:\ADX_UPGM;C:\ADX_SPGM;
```

Before running on a terminal the 286 file should be postlinked to improve its load time and make memory usage more efficient in a multitasking environment:

POSTLINK EALTS10L.286	Large memory
POSTLINK EALTS00L.286	Medium memory

Before running the newly created application, the support and data files must be configured for the ScreenKey keyboard (see below). Additionally, logical filenames must be configured that defines the location of these files.

5 Chain Sales Application Installation

Chain Sales Application does not implement a “user-exit model” as provided by Supermarket (SA) and General Sales Applications (GSA). With CSA, the core source modules are directly and uniquely modified in each site, resulting in a CSA installation that is unique to that site.

Integrating support for the SK-7510 console into CSA is very similar to SA and GSA. SKI does not distribute specific CSA wrapper integration files and instead works with each customer to implement the most appropriate wrapper for their CSA installation.

The following is an overview of how a simple integration can be handled.

5.1 Runtime Module Linker Includes

First determine if you are using the large or medium memory model application (EGHTS10L.286 is large and EGHTS00L.286 is medium memory application).

The runtime core and wrapper modules must be included in the appropriate linker file:

EGHTS10L.INP	Large memory
EGHTS00L.INP	Medium memory

Edit this file to add the following lines:

EGHTS10L.INP (Large)	EGHTS11L.INP (Medium)
RTITSCSA.LBJ, RTITSUGB.LBJ, MGVTSASB.LBJ,	RTITSCSA, RTITSUGB, MGVTSASC,

The core SK-7510 library module (RTITSUGB) is as used with SA and GSA. The CSA wrapper module (RTITSCSA) is implemented uniquely for each site.

5.2 User Code Includes

Having included the runtime support files, the appropriate location must be selected from which to call the SKI function to check file versions and download new files if required.

The following code modifications of EIGHTMAIN.BAS module have to be implemented in order to have the SK-7510 ScreenKey console functioning in the Active mode. An appropriate location is between LOADSEQ: and INITFAIL: labels but immediately before the GOTO MAINLOOP statement. Insert the following code:

```
! Assign initial session number for ScreenKey console driver to access
! external setup files
%INCLUDE RTISESSM.C86

! Initialise SKI keyboard and check if downloads required
CALL RTI.CHECKKEYBOARD ! Initialization routines
```

The function RTI.CHECKKEYBOARD interrogates the ScreenKey console via the assigned serial port and downloads new files if required.

This function call may be placed in any appropriate location suitable to the application customisations already made. It must be preceded by the %INCLUDE RTISESSM.C86 which defines the unique

session number assigned to the SKI console. This session number is hard-coded (compiled) into the sales application executable module (all other session numbers may be assigned using the RTISESSN.DAT file). When deciding on the final location of this function call, consideration should be given to the anticipated regularity of console updates (code, option and SAC files) and how these may be triggered. For example, this call could be put into a timer routine that is triggered each night at 3am. Alternatively it could be implemented each time a new user logs on.

To complete the integration the following changes should also be implemented.

In the Global Variables declaration section, insert the following include:

%INCLUDE RTITSUGB.V86 ! SKI keyboard Global Variables

In the External Functions declaration section, insert the following include:

%INCLUDE RTITSUGB.F86 ! SKI keyboard Public Routines

The terminal sales user exit error handling **must** also be modified to add support for handling specific errors. Under TERMERR: label section and immediately after assigning the values for GL.ERRCODE\$, GL.ERRSESN, GL.ERRCODEX variables, insert the following code:

! Handle RT errors
CALL RTI.HANDLERR(0,0,0) ! Handle rt errors

Before proceeding to the compilation stage, check the session number assignment in module RTISESSM.C86. This must be unique for your sales application:

```
!*****
!  

!RTISESSM.C86 - Assign unique session number for SKI keyboard  

!usage.  

!  

!This CODE file should be included prior to calling main keyboard  

!support routine, i.e. RTI.CHECKKEYBOARD.  

!  

!This file is distributed with the 7510 kernel.  

!  

!*****

RTI.SESS.FILE% = 50          ! File access session number

!***** End of RTISESSM.C86 *****
```

5.3 Terminal Sales Compilation

After adding the above code to the user exit modules, compile the module that you have modified, using the 4680 BASIC compiler, for example:

BASIC EGHTMAIN [fum(b)]	Large memory
BASIC EGHTMAIN [fu]	Medium memory

Note: If large memory is used, after the compile has completed successfully rename EGHTSUxx.OBJ to EGHTSUxx.LBJ

5.4 Link and Postlink CSA Application

After successful compilation of the main module, the GSA application should be linked:

LINK86 EGHTS10L [i]	Large memory
LINK86 EGHTS00L [i]	Medium memory

Note: Successful compilation may require files from multiple directories. Use the linker environment variable to define these paths, for example:

```
DEFINE LNK86PATH=C:\ADX_UPGM;C:\ADX_SPGM;
```

Before running on a terminal the 286 file should be postlinked to improve its load time and make memory usage more efficient in a multitasking environment:

POSTLINK EGHTS10L.286	Large memory
POSTLINK EGHTS00L.286	Medium memory

Before running the newly created application, the support and data files must be configured for the 7510 ScreenKey keyboard (see below). Additionally, logical filenames must be configured that defines the location of these files.

6 The Runtime Procedure - Downloads

Since the User Exits Kernel is integrated into the Terminal Sales Application it can be triggered at various points in the application.

When triggered, the User Exit Kernel code that performs the Download function communicates with the SK-7510 ScreenKey Keyboard to ascertain whether the Download files (CDL, SDL and ODL) that are currently in the SK-7510's non-volatile RAM are the same as those named in the ROMFILENAME, SACFILENAME and OPTFILENAME settings in the RTIKBnnn file and stored in the directory on the controller specified by the "RTI:" user logical filename.

If they are then the User Exit Kernel's download function exits without doing anything.

If not then the User Exit Kernel's download function copies one or more download files from the controller to the SK-7510 via the SIO channel serial port.

If the specified file is not present, the User Exit Kernel's download function proceeds to download the next file and an error is recorded as specified by the trace setting.

The User Exit Kernel's download function first checks the CDL file. If it needs to be downloaded then the SK-7510 will clear the current ODL and SDL files from its non-volatile RAM. This means that after a CDL download an ODL and SDL download will be required.

The User Exit Kernel's download function then checks whether an ODL download is required. If it is then the SK-7510 clears the SDL download from its non-volatile RAM. This means that after an ODL download a SDL download will be required.

The User Exit Kernel's download function then checks whether a SDL download is required.

The User Exit Kernel's download function relies on the configuration Files RTITERMS.DAT, RTISESSN.DAT and RTIKBnnn to decide not only the names of the download files but what Serial Port to use and what level of progress reporting.

6.1 Download Files

The Code Download file (.CDL) is supplied by RTI.

The SAC Download File (.SDL) is created by first using the SAC Editor to create a SAC file (.PKF) and then converting the .PKF to a .SDL using the Make Download utility – MAKEDNL.EXE.

The Options Download File (.ODL) is created using the Make Download utility – MAKEDNL.EXE.

6.2 TraceOutput

Progress messages are displayed on the Operator Display and/or sent to the printer cash receipt station and/or saved in a text file. In addition, events are recorded in the Application Event Log. See the discussion of [TraceOutput](#) for details

Three configuration files must be set up:

- RTITERMS.DAT List of terminals that have ScreenKey keyboards attached
- RTISESSN.DAT Session numbers available to user exits kernel
- RTIKB000\RTIKBnnn Terminal specific configuration items

6.3 *RTITERMS.DAT Terminal Listing File*

This file defines the terminal numbers that have a ScreenKey keyboard attached.

Individual terminal numbers may be defined or a number range may be specified. Each item must be specified on a separate line.

The following is an example:

```
" 1 "  
" 5 "  
" 22-35 "  
" 54 "  
" 63-69 "
```

The terminal number **MUST** be listed in RTITERMS.DAT for the User Exits Integration to operate on that terminal.

6.4 RTISESSN.DAT Session Number Assignment

This file defines the session numbers that may be assigned for usage by the User Exits Integration. This file is provided so that assigned session numbers may be changed dynamically. It is critical that the User Exit Kernel integrated into the Terminal Sales Application does not use any Session Numbers that are already in use by the TSA.

This is a Required File. The User Exits Integration will not run without it. The format of RTISESSN.DAT is:

```
"SERIAL=" , 80
"API=" , 51
"TRACE=" , 52
"MESSAGES=" , 53
```

Note:

*In addition to the session numbers defined in RTISESSN.DAT, one other unique session number is hard-coded for SK-7510 user exits usage during the compile stage. This fixed IO session number is used for **all** file accesses (except for the trace file).*

Each line must specify a unique session number. If the associated option is not required then the session number should be set to 0.

The order of items is critical and must be as shown. The Serial IO session number is mandatory whereas all the other items are optional, i.e. may be set to zero.

<i>IO Session</i>	<i>Purpose</i>
-------------------	----------------

SERIAL Access to the terminal SIO channel serial port for downloading information to the keyboard

Do not set to 0.

API Access to the Applications Programming Interface to provide greater control over the actions of the keyboard during operations, e.g. display particular ScreenKey menu, show splash screen on the TDA, etc.

Set to 0 to prevent attempt to open ANDISPLAY3

TRACE The session for use with the RTITRACE.nnn file. TraceOutput is sent to this file (and/or the Display and/or the Printer) depending on the TraceOutput setting in the RTIKBnnn file.

Set to 0 to disable this feature and override the TraceOutput settings in RTIKBnnn.

MESSAGES During operation of the User Exits Integration, TraceOutput is generated and may be displayed, printed or written to a file. The descriptor text for these messages is stored in an external file. This session number must be defined in order to access this file.

Set to 0 to turn off TraceOutput. This overrides the TRACE session number setting above and the TraceOutput settings in the RTIKBnnn file.

If TraceOutput is turned off then TRACELEVEL 1 events will be recorded in the Application Event Log in addition to TRACELEVEL 0 messages.

The User Exits Kernel also uses a number of other Session Numbers. These are...

<i>IO Session</i>	<i>Purpose</i>
-------------------	----------------

50 File access – RTISESSN.DAT, RTITERMS.DAT, RTIKBnnn, .CDL, .SDL and .ODL. This Session number is in RTISESSM.C86 and may be edited.

19 Kernel for GSA assumes and uses session 19 for ANDISPLAY output

21 Kernel for GSA assumes and uses session 21 for CASH RECEIPT station.

6.5 RTIKBnnn Terminal Specific Parameters

The RTIKBnnn file defines the files to be downloaded into the SK-7510, the level of tracing and progress reporting to be employed, the terminal serial port number to use for downloads and the time to wait for the keyboard to come back online after a code (.CDL) download.

RTIKBnnn can be used in a generic mode to define parameters applicable to all terminals by naming it RTIKB000. Unique parameters may be set for one or more specific terminal numbers by creating individual RTIKBnnn files where 'nnn' is replaced with the actual terminal number, e.g. RTIKB012 applies to terminal number 012.

The User Exits Integration searches for a RTIKBnnn file for each terminal number specified in RTITERMS.DAT. If a terminal specific RTIKBnnn file is not found it uses the parameters from RTIKB000.

This is a Required File. The User Exits Integration will not run without it.

It is advisable to always create a RTIKB000 file. Terminal specific files may then be created only as required.

The following is an example of a RTIKBnnn file. The order of the items is critical and MUST be as shown.

```
"ROMFILENAME=" , "code.cdl "  
"SACFILENAME=" , "sac.sdl "  
"OPTFILENAME=" , "options.odl "  
"SERIALPORT#" , 2  
"TRACELEVEL=" , 1  
"TRACETODISPLAY=" , 0  
"TRACETOPRINTER=" , 0  
"TRACETOFILE=" , 1  
"KBDTIMEOUT=" , 30
```

The first three parameters define the filenames to be downloaded to the keyboard. The path by which these files are accessed is created using the User Logical Filename "RTI:".

A null filename may be used to skip a particular download item, e.g. "CDLFILENAME=", "".

The User Exits Integration communicates with the ScreenKey keyboard as a serial device. This is implemented by the keyboard as a Feature Card Emulation. The SERIALPORT# parameter defines the serial port number specified in the terminal back-office configuration (Terminal Device Group). Note that this parameter value is entered as an integer (i.e. it is not enclosed between quotes).

TRACELEVEL, TRACETODISPLAY, TRACETOPRINTER and TRACETOFILE define the level of TraceOutput information and progress reporting presented to the user. The values assigned to each of these trace parameters are entered as integers (i.e. no quotes). See the discussion of [TraceOutput](#) below for details.

After a .CDL and .ODL download the Sk-7510 will be offline for a short time. During this time the SK-7510 will not respond to the USER EXITS INTEGRATION. The value assigned to KBDTIMEOUT is the number of seconds that the USER EXITS INTEGRATION will continue to try to connect to the SK-7510 before timing out.

6.6 RTIMSG.DAT - Message Text File

Message text descriptors for display on ANDISPLAY, printing on the cash receipt station, or for writing to the trace file are taken from an external direct file (RTIMSG.DAT). This file is organised as fixed size records:

```
"- 38 character wide descriptor text -", "nn"
```

where *nn* is the size of usable text within the 38-wide descriptor, i.e. excludes white space at end of descriptor, for example:

```
"DOWNLOAD UTILITY IS STARTING          ", "28"
```

Each record (or line number) in this file is associated with an error or message number used by the User Exits Integration.

This file may be translated for natural language support. There are no other text messages embedded in the compiled code.

Use of the external descriptor file requires a unique session number assignment, set in RTISESSN.DAT ("MESSAGES" parameter). If the MESSAGES session number is set to zero in RTISESSN.DAT, then TRACE parameter destination settings in RTIKBnnn are ignored.

The text in the first line of RTIMSG.DAT file corresponds to message RT01, the text in the second line corresponds to message RT02 and so on. See [Messages Summary](#) for a complete list of all the RTxx messages that should be in RTIMSG.DAT.

6.7 Controller Set-up

Both the master and alternate controllers have to be configured.

- a) create and activate User Logical File Names used by the User Exits Integration
- b) create a suitable directory on both the master and alternate to hold the files used by the User Exits Integration controllers (if one doesn't already exist), copy the files onto the master controller and set the file attributes on the master controller

The User Exits Integration uses four *User Logical Filenames* to find Configuration, Support and Download files.

The file attributes must be set-up such that the Configuration and Support files are automatically managed between the master and alternate controller i.e. file attribute 4 (i.e. compound file, distribute at update). Set the distribution type on the master controller only – no need to set them on the alternate controller.

File Name	Logical File Name used by User Exits Integration to access this file	File Attribute
RTIKB000 RTIKBnnn	RTIK :	4
RTITERMS.DAT	RTITERMS	4
RTISESSN.DAT	RTISESSN	4
RTIMSG.DAT	RTI :	4

where nnn is a 3 digit terminal number e.g. 001

The file attribute does not apply to the Download files or the Trace file.

<name>.CDL <name>.SDL <name>.ODL	RTI :
RTITRACE.nnn	RTI :

where nnn is a 3 digit terminal number e.g. 001

Say we decide that we want to put all the SKI files (RTITERMS.DAT, RTISESSN.DAT, RTIKB000, RTIKBnnn and RTIMSG.DAT) into a directory called C:\RTI. We must...

- a) create a directory called C:\SKI on the master controller
- b) copy the SKI files into C:\RTI on the master controller
- c) create a directory called C:\RTI on the alternate controller (don't copy any files into it)

The corresponding User Logical File Names would then be set-up as follows...

<i>User Logical File Name</i>	<i>Value</i>
RTI :	ADXLXACN : : C : \RTI \
RTIK :	ADXLXACN : : RTI : RTIKB
RTITERMS	ADXLXACN : : RTI : RTITERMS . DAT
RTISESSN	ADXLXACN : : RTI : RTISESSN . DAT

If we decide to place the SKI files in a different directory or spread the files around among a few directories than the values assigned to the User Logical Filenames would have to be changed accordingly.

6.7.1 User Logical File Name Set-up

Create the four User Logical File Names on both the master and alternate controllers. Then activate the changes on both controllers and re-IPL both controllers.

To create a User Logical File Name:

1. From *System Main Menu*, select option 4 *Installation and Update Aids*
2. Select 1 *Change Configuration Data*, then option 2 *Controller Options*
3. Place X beside *User Logical File Names* and press ENTER
4. Select 1 *Define a Logical File Name* and type the new user logical file name to be defined and press ENTER
5. On the next screen type the expanded name (full address of the path and/or filename to use) and press ENTER
6. Press F3 until *Configuration* screen is shown, then select 4 *Activate Configuration*
7. Select 2 *Controller Configuration* and press ENTER
8. Re-IPL controller.

6.7.2 File Attributes Set-up

Set the File Attributes on the files on the master controller only.

File attributes may be set manually:

1. From *System Main Menu*, select option 3 *File Utilities*
2. Select 3 *Distributed File Utilities*
3. Select 3 *Modify File Distribute Type*
4. Type the path and filename and press ENTER
5. Specify the distribution as type 4 (compound file, distribute at update)

The same action may also be performed from the command line using:

```
ADXCSU0L 3 4 path:filename
```

Example:

```
ADXCSU0L 3 4 RTI:RTITERMS.DAT (path defined by user logical file name RTI:)
```

Note: Terminal specific RTIKBnnn files may be created (e.g. RTIKB001 for terminal number 1). Set the file attributes for these terminal-specific RTIKBnnn files to attribute 4 (i.e. compound file, distribute at update).

7 The Runtime Procedure - API

The SK-7510 supports an Application Programming Interface (API), by which a terminal application programmer may send instructions to the SK-7510.

The API is implemented by emulating a ‘virtual’ display device. A terminal application developer can send commands to the SK-7510 using display write commands.

The SK-7510 emulates an Operator display using a device address as specified by the downloaded Options file. The API is disabled if the Options API record is not present or if the device address is set to zero.

7.1 Escape Sequence Summary

The API by default interprets commands as TDA display data where they do not start with an embedded escape sequence. The API uses only alphanumeric data in the escape sequences to overcome non-alphanumeric character translations by the driver.

For details see “ScreenKey for IBM 46xx: Technical Reference Manual: Model SK-7510”.

7.1.1 TDA\CDA Commands

ESC Sequence	Hex Values	Description
<none>	<none>	Defaults to display text on the TDA if there is no ESC sequence
^\SB	5E 5C 53 42	Display Splash Screen on TDA
^\SN	5E 5C 53 4E	Set LCD Panel to Normal Mode – turn OFF CDA, restore back-light, turn OFF splash, restore TDA text
^\CT	5E 5C 43 54	Clear the TDA display and buffer
^\P<xx><yy> <nn>	5E 5C 50 x1 x2 y1 y2 n1 n2	Print (nn) bytes of next line at (xx, yy) co-ordinate on TDA e.g. ^\P0301 means row 3 column 1. Co-ordinates start at 1,1 in the TLH corner. The data is extracted from the next TDA display command
^\LF <nn>	5E 5C 4C 46 n1 n2	Linefeed <nn> lines, e.g. ^\LF03 to print 3 blank lines
^\D<n>	5E 5C 44 n1	Direct all following text messages to TDA (n1 = 0) or CDA (n1 = 1)

7.1.2 LCD Panel Commands

ESC Sequence	Hex Values	Description
^\SD	5E 5C 53 44	Set LCD Panel to Dark (back-light OFF)
^\LPB<n>	5E 5C 4C 50 42 n1	Adjust LCD panel brightness, n1='H' for HIGH, n1='L' for NORMAL
^\LPC<d><nn>	5E 5C 4C 50 43 d1 n1 n2	Adjust LCD panel contrast by <nn> steps: <d> = '+' for forward, <d> = '-' for backward from current setting <d> = 'G' to set absolute value to <nn> (0 < nn < 100)
^\LPS ⁽¹⁾	5E 5C 4C 50 53	Save changes made to LCD panel using API commands to EEPROM
^\LPR	5E 5C 4C 50 52	Abort changes made to LCD panel using API commands and restore previous settings

7.1.3 ScreenKey Commands

ESC Sequence	Hex Values	Description
^\M<name>	5E 5C 4D n1 ... nn	Goto Menu <name> on ScreenKeys <name> is a reference to a menu name defined in the SAC Editor and can be up to 10 characters in length

7.1.4 Keyboard Commands

ESC Sequence	Hex Values	Description
^\QM	5E 5C 51 4D	Request Manager Status return via Keyboard channel

8 API Customisation Examples for Supermarket Application

The user exit kernel provides access to the API channel (Applications Programming Interface). The API may be used to tightly integrate the operation of the ScreenKey keyboard into a stores sales application. Through the API, the actions and operation of the keyboard may be modified through user exit code without manual operator interaction.

The API is accessed via a common function contained in the user exit kernel:

```
RTI.TOAPI(string APIcommand$)
```

The full API command set can be accessed using this kernel function. The passed string parameter can contain an API command otherwise it is treated as display data for the TDA.

Examples

Display data on the TDA:

```
CALL RTI.TOAPI("--- TDA display text ---")      ! text for TDA
```

Issue three linefeeds:

```
CALL RTI.TOAPI("^\\LF03")      ! 3 linefeeds after end trans
```

Goto specific SAC menu:

```
CALL RTI.TOAPI("^\\MMAIN")      ! Display MAIN menu
```

Clear TDA screen and buffer:

```
CALL RTI.TOAPI("^\\CT")      ! Clear TDA screen and buffer
```

This section provides a range of examples that demonstrate how the API may be used to perform different functions that could enhance store performance. All example code is based on Supermarket Application, but the corresponding implementation in General Sales is straightforward.

Note:

*To use the API, the Terminal Device Group (TDG) **must** have ANDISPLAY3 configured and the downloaded Options file **must** also have the API channel enabled, matching the TDG settings. The **SK-7510 Planning and Installation Guide** describes how to do this.*

8.1 Echo Electronic Journal Data to TDA in Real Time (Post Printing Setup)

With the introduction of fast thermal printers (such as the 4610) and the move towards post-printing of receipts, operators can encounter problems determining what has already been entered in a transaction, e.g. if the customer queries if an item has been incorrectly entered twice.

The ScreenKey keyboard provides a method of capturing transaction data in real time and displaying this information on the TDA for ease of access and recall.

The API channel is used to implement this feature.

After installing the user exit kernel, add the highlighted line to call the API in user exit 21 (EAMTSU21.J86):

```
√* TIME STAMP BLOCK *****  
√** END OF TIME STAMP BLOCK *****/  
  
FUNCTION TSUPEC21 PUBLIC  
!CALL SUBSTR(TS.SJDATA$,16,"EC21",0,4)  
  
CALL RTI.TOAPI(TS.PRTBUF$)      ! Send print buffer to API  
  
END FUNCTION
```

TS.PRTBUF\$ contains the text about to be printed to the summary journal (electronic journal for 4610 installations). This data is passed directly to the API which prints the text string on the TDA.

Note:

Remember to activate user exit 21 in terminal options.

8.2 Echo Cash Receipt Data to TDA

In a similar manner to the above example, transaction data printed on the cash receipt station may be directed to the TDA using user exit 20.

After installing the user exit kernel, add the highlighted line to call the API in user exit 20 (EAMTSU20.J86):

```
\/* TIME STAMP BLOCK *****/
\** END OF TIME STAMP BLOCK *****/

FUNCTION TSUPEC20 PUBLIC
!CALL SUBSTR(TS.PRTBUF$,28,"EC20",0,4)
INTEGER*1 TSUPEC20           ! define variable !IR89474

CALL RTI.TOAPI(TS.PRTBUF$)    ! Send print buffer to API

END FUNCTION
```

Notes:

- 1 Remember to activate user exit 20 in terminal options.
- 2 You may want to filter some print lines using *TS.LINETYPE*, e.g. customisations that include printer escape sequences

8.3 Show Splash Screen when Signed Off

The Options file may include a user-generated splash screen. This is a black and white graphic image that covers the TDA area. Stores can use this feature to display some advertising, store logo or other useful message when the POS station is not in use.

This implementation demonstration uses user exit 21 to detect when linetype 23 is sent to the journal printer station to detect sign-off user message.

Add the highlighted lines to call the API in user exit 21 (EAMTSU21.J86):

```
√* TIME STAMP BLOCK *****  
√** END OF TIME STAMP BLOCK *****/  
  
FUNCTION TSUPEC21 PUBLIC  
!CALL SUBSTR(TS.SJDATA$,16,"EC21",0,4)  
  
  IF TS.LINETYPE = 23 THEN BEGIN          ! sign off message  
    CALL RTL.TOAPI("^\\SB")                ! API cmd to show splash  
  ENDIF  
  
END FUNCTION
```

The splash screen must be turned **off** when the operator signs on, i.e. return the TDA to normal mode. One possible solution to this is to use the start of transaction user exit (user exit 1) activated at the start of every new transaction.

Add the highlighted line to call the API in user exit 1 (EAMTSU01.J86):

```
√* TIME STAMP BLOCK *****  
√** END OF TIME STAMP BLOCK *****/  
  
FUNCTION TSUPEC01 PUBLIC  
! TSUPEC01 - BEFORE TX  
  
  CALL RTL.TOAPI("^\\SN")      ! Return TDA to normal mode  
  
END FUNCTION
```

Note:

Remember to activate user exits 1 and 21 in terminal options.

8.4 Clear TDA at Start of New Transaction

For clarity, stores may wish to only show the current transaction on the TDA. This can be done by erasing old text from the TDA at the beginning of every new transaction.

User exit 1 is called at the start of each transaction. Add the highlighted line to call the API in user exit 1 (EAMTSU01.J86):

```
√* TIME STAMP BLOCK *****  
√** END OF TIME STAMP BLOCK *****/  
  
FUNCTION TSUPEC01 PUBLIC  
! TSUPEC01 - BEFORE TX  
  
CALL RTI.TOAPI('^CT') ! Clear TDA  
  
END FUNCTION
```

Note:

Remember to activate user exit 1 in terminal options.

8.5 Show Foodstamp Total on the CDA

This example is included to demonstrate the ability to tightly integrate the ScreenKey keyboard into the sales application using user exits. **It is not intended for use in a live store environment as not all conditions are covered.**

The example demonstrates capturing information within a user exit and displaying this on the CDA, **only** if a foodstamp item is entered, modifying this value as the transaction progresses and removing this display area at the end of the transaction. The foodstamp total is computed specially for this example.

User exit 1 (start of a new transaction) is used to ensure the CDA is off. The foodstamp total is reset to zero at this point. Add the highlighted lines to call the API in user exit 1 (EAMTSU01.J86):

```
√* TIME STAMP BLOCK *****
\** END OF TIME STAMP BLOCK *****/

FUNCTION TSUPEC01 PUBLIC
! TSUPEC01 - BEFORE TX

RTITOTAL% = 0           ! Use var to store computed foodstamp total
CALL RTI.TOAPI("^SN")  ! Turn off CDA (return TDA to normal mode)

END FUNCTION
```

At the end of each transaction, turn off the CDA. Add the highlighted lines to call the API in user exit 2 (EAMTSU02.J86):

```
√* TIME STAMP BLOCK *****
\** END OF TIME STAMP BLOCK *****/

FUNCTION TSUPEC02 PUBLIC
! TSUPEC02 - AFTER TX
INTEGER*1 TSUPEC02           ! define variable !IR89374

CALL RTI.TOAPI("^SN")      ! Turn off CDA (return TDA to normal mode)

END FUNCTION
```

This example uses two variables that should be added to EAMTSUVA.J86:

```
√* TIME STAMP BLOCK *****
\** END OF TIME STAMP BLOCK *****/
!
!THIS INCLUDE FILE PROVIDES A PLACE FOR THE USER TO DEFINE VARIABLES
!WHICH ARE COMMON TO SEVERAL OF HIS USER EXIT FUNCTIONS IN EAMTSUPC.
!
!REAL          RTITOTAL%          ! Stores foodstamp total
STRING        RTITOTAL$          ! Stores string representation of foodstamp total
```

The main processing is handled in user exit 21, called before data is printed on the summary journal. Here the total is updated, increased for new foodstamp item sales and reduced when foodstamps are redeemed against the total.

Add the highlighted lines to call the API in user exit 21 (EAMTSU21.J86):

```

\* TIME STAMP BLOCK *****
\** END OF TIME STAMP BLOCK *****/

FUNCTION TSUPEC21 PUBLIC
!CALL SUBSTR(TS.SJDATA$,16,"EC21",0,4)

IF (TS.LINETYPE=1) THEN BEGIN           ! Item sale
IF ((IR.INDICAT1 AND 08) <> 0)         \ Foodstamp item
THEN BEGIN
RTITOTAL% = RTITOTAL% +             \ Update total
(TS.XPRICE)
ENDIF
ENDIF

IF (TS.LINETYPE=2) THEN BEGIN         ! Tender amount
IF (SL.TE.TENDTYPE=3) AND           \ Foodstamp item
(SL.TE.AMTTENDE>0) THEN BEGIN
RTITOTAL% = RTITOTAL% -           \ Reduce foodstamp
(SL.TE.AMTTENDE)
ENDIF
ENDIF

IF RTITOTAL% > 0 THEN BEGIN           ! Foodstamp total > 0
RTITOTAL$ = STR$(RTITOTAL%)
RTITOTAL$ = "FS TOTAL" +           \
LEFT$(BLANK$,11-LEN(RTITOTAL$)) + \
LEFT$(RTITOTAL$,LEN(RTITOTAL$)-2) + \
"." + RIGHT$(RTITOTAL$,2)

CALL RTI.TOAPI("^D1")               ! Direct output to CDA
CALL RTI.TOAPI(RTITOTAL$)         ! Show Foodstamp total
CALL RTI.TOAPI("^D0")             ! Redirect output to TDA
ENDIF

END FUNCTION

```

Notes:

- 1 Remember to activate user exits 1, 2 and 21 in terminal options.
- 2 Example assumes linetypes 1 and 2 are directed to the summary journal station.

9 Problem Determination

The User Exits Integration can report progress information and error messages in many different ways, configurable to suit each users particular requirement.

Information to base Problem Determination is available from the following sources

- the Application Event Log
- the TraceOutput
- the Required File Handling
- the Progress display on ScreenKeys

The recommended approach to Problem Determination is to use the Application Event Log. Set the MESSAGE Session number in the RTISESSN.DAT file to 0. This will force TRACELEVEL 1 events to be stored in the Application Event Log. TRACELEVEL 0 events are always stored in the Application Event Log. It will also mean that TraceOutput is turned off so there is no progress reporting / error reporting to the Display, Printer or Trace File (RTITRACE.nnn).

If preferred the Display, Printer or Trace File may be used to record progress and error messages.

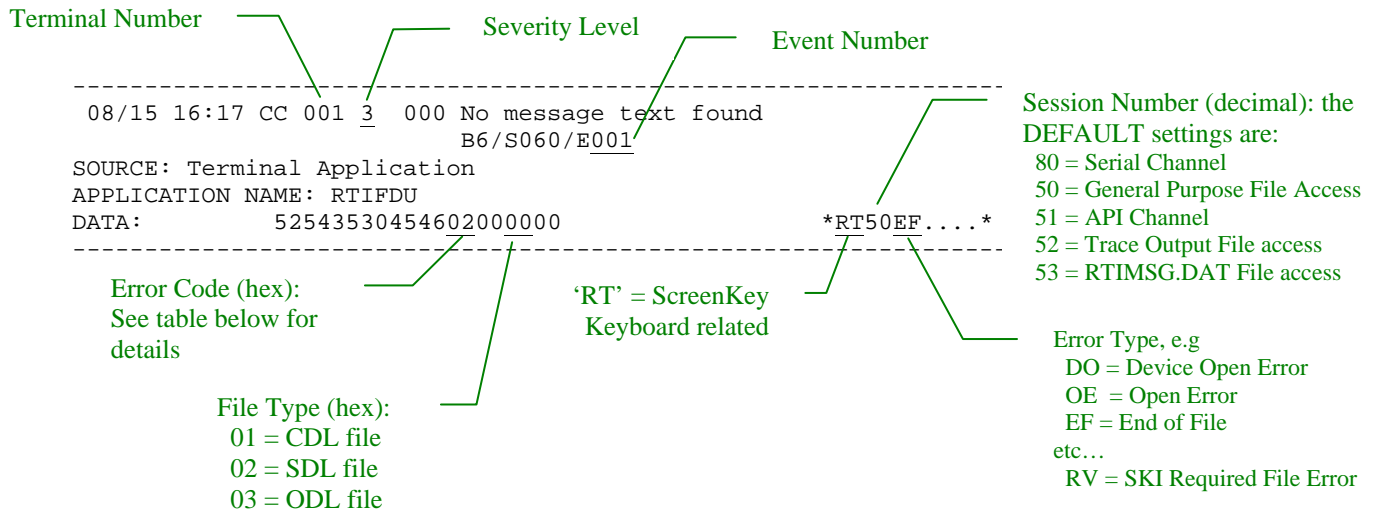
9.1 Application Event Log

All fatal errors (message numbers RT01-RT19) are always recorded in the Application Event Log.

TRACELEVEL 1 messages (message numbers RT20-RT39) may also be recorded in the Application Event Log. If TRACELEVEL flag is set to 1 or 2 but a session number is not assigned for access to the message descriptor file (i.e. "MESSAGES=",0 in RTISESSN.DAT), then all TRACELEVEL 1 messages will be recorded in the Application Event Log.

TRACELEVEL 2 messages (message numbers RT40 and greater) are never recorded in the Application Event Log.

All errors are reported in the Application Event Log with severity level 3 and event number 1.



The Error Code relates directly to the message number described below.

The Error Type reports the error type returned by the 4690 OS, otherwise it will be set to spaces. One SKI specific error code is introduced in the User Exits Integration. Error type "RV" indicates that a required file error occurred, either file could not be accessed or required information in the file was not found.

9.1.1 Event Reference and the Application Event Log

The Event Reference is in the form "RT99". It is used to lookup the explanation. To work out the Event Reference from the Application Event Log simply

- 1) take the "RT"
- 2) take the 2 hex digit Error Code and convert to a two digit decimal number

E.g. if the Application Event Log contained the following

```
DATA: 525435304546OD000000 *RT50EF....*
```

it would equate to Event Reference "RT13" because 0D hex is 13 decimal.

9.2 TraceOutput

TraceOutput may be directed to the terminal's Operator Display device, cash receipt station or to a terminal-specific trace file created on the active controller. It is possible to direct TraceOutput to any, all or none of these three destinations.

TraceOutput is controlled by

- a) the MESSAGES session number in RTISESSN.DAT
- b) access to the RTIMSG.DAT file
- c) the TRACELEVEL setting in RTIKBnnn
- d) the TRACETOFILE, TRACETOPRINTER and TRACETODISPLAY settings in RTIKBnnn
- e) the TRACE session number in RTISESSN.DAT

If the MESSAGES session number is zero or the RTIMSG.DAT file is not accessible then TraceOutput is turned off – no information will be sent to the Display, Printer or Trace File.

Messages are divided into three Levels – Level 0, 1 and 2. The TRACELEVEL flag determines what level of TraceOutput is in operation.

TRACELEVEL 0	RT01 – RT19	Serious or fatal error messages that prevent the User Exits Integration from completing successfully, e.g. RTITERMS file missing, communications error with keyboard
TRACELEVEL 1	RT20 – RT39	Messages that indicate critical steps achieved in the process, e.g. file download complete, or report non-fatal errors
TRACELEVEL 2	RT40 upwards	Additional information that may be useful in debugging situations

The TRACETODISPLAY, TRACETOPRINTER and TRACETOFILE settings in RTIKBnnn are used to control which devices TraceOutput is sent to. Setting the value to '1' turns on TraceOutput to that device.

TRACETODISPLAY	direct messages to the ANDISPLAY: device. This is typically the Operator Display
TRACETOPRINTER	direct messages to printer's Cash Receipt station
TRACETOFILE	direct messages to a trace file called RTITRACE.nnn where nnn is the 3 digit Terminal number in the path specified by the "RTI:" user logical file name using the TRACE session number from RTISESSN.DAT.

9.2.1 Trace to File

When messages are directed to a trace file, a file called RTITRACE.nnn is generated (where 'nnn' is the terminal number) on the controller in the path specified by the "RTI:" user logical file name.

This file is only created when a unique non-zero session number is allocated to the trace file in RTISESSN.DAT and when the MESSAGES session number is non zero and the RTIMSG.DAT file is accessible.

With the TRACE Session number in RTISESSN.DAT set to a valid non-zero Session Number and Trace settings in RTIKBnnn set to "TRACELEVEL=", 0 and "TRACETOFILE=", 1 the trace file output for a successful run will look something like this...

```
27/07/01 @ 01 @ 09:49
VER 3.20
```

With the TRACE Session number in RTISESSN.DAT set to a valid non-zero Session Number and Trace settings in RTIKBnnn set to "TRACELEVEL=", 1 and "TRACETOFILE=", 1 the trace file output for a successful run will look something like this...

Trace Level	Message Code	Message
		27/07/01 @ 01 @ 09:49
		VER 3.20
1	RT20	DOWNLOAD UTILITY IS STARTING
1	RT28	FILE DOWNLOAD IS NOT REQUIRED CDL
1	RT28	FILE DOWNLOAD IS NOT REQUIRED OPT
1	RT27	FILE DOWNLOAD COMPLETED SAC
1	RT31	API NOT ENABLED
1	RT22	DOWNLOAD UTILITY HAS FINISHED

With the TRACE Session number in RTISESSN.DAT set to a valid non-zero Session Number and Trace settings in RTIKBnnn set to "TRACELEVEL=", 2 and "TRACETOFILE=", 1 the trace file output for a successful run will look something like this...

Trace Level	Message Code	Message
		27/07/01 @ 01 @ 09:49
		VER 3.20
1	RT20	DOWNLOAD UTILITY IS STARTING
2	RT43	DOWNLOAD FILE NAME: CDL 01120000.cdl
2	RT43	DOWNLOAD FILE NAME: SAC abcdef.sdl
2	RT43	DOWNLOAD FILE NAME: OPT abcdef.odl
2	RT44	USE SERIAL PORT NUMBER 2
2	RT48	OPENING DOWNLOAD FILE 01120000.cdl
2	RT50	NUMBER OF BYTES/BLKS 93100/389
1	RT28	FILE DOWNLOAD IS NOT REQUIRED CDL
2	RT48	OPENING DOWNLOAD FILE abcdef.odl
2	RT50	NUMBER OF BYTES/BLKS 141/2
1	RT28	FILE DOWNLOAD IS NOT REQUIRED OPT
2	RT48	OPENING DOWNLOAD FILE abcdef.sdl
2	RT50	NUMBER OF BYTES/BLKS 50307/211
1	RT27	FILE DOWNLOAD COMPLETED SAC
1	RT31	API NOT ENABLED
1	RT22	DOWNLOAD UTILITY HAS FINISHED

9.2.2 Trace to Display

The messages are sent to the ANDISPLAY: device.

9.2.3 Trace to Printer

The messages are sent to the printer's Cash Receipt (CR) station

9.3 *Required File Error Handling*

For events RT03, RT04, RT05 and RT06 the response includes additional “Required File” handling. This is intended to follow the procedure for dealing with a B040 message.

If running the Terminal Sales Application with User Exit Integration then the system will display the message:

“B040 FILE ACCESS FAILED session number”

It will then wait for the CLEAR key to be pressed. When the CLEAR key is received the program will retry the operation. If it fails again it will report the error and wait for the CLEAR. If the problem is fixed the application will continue. The Terminal Sales Application will loop forever if the problem is not fixed.

If running the RTIFDU.286 application, instead of the B040 message we display the message

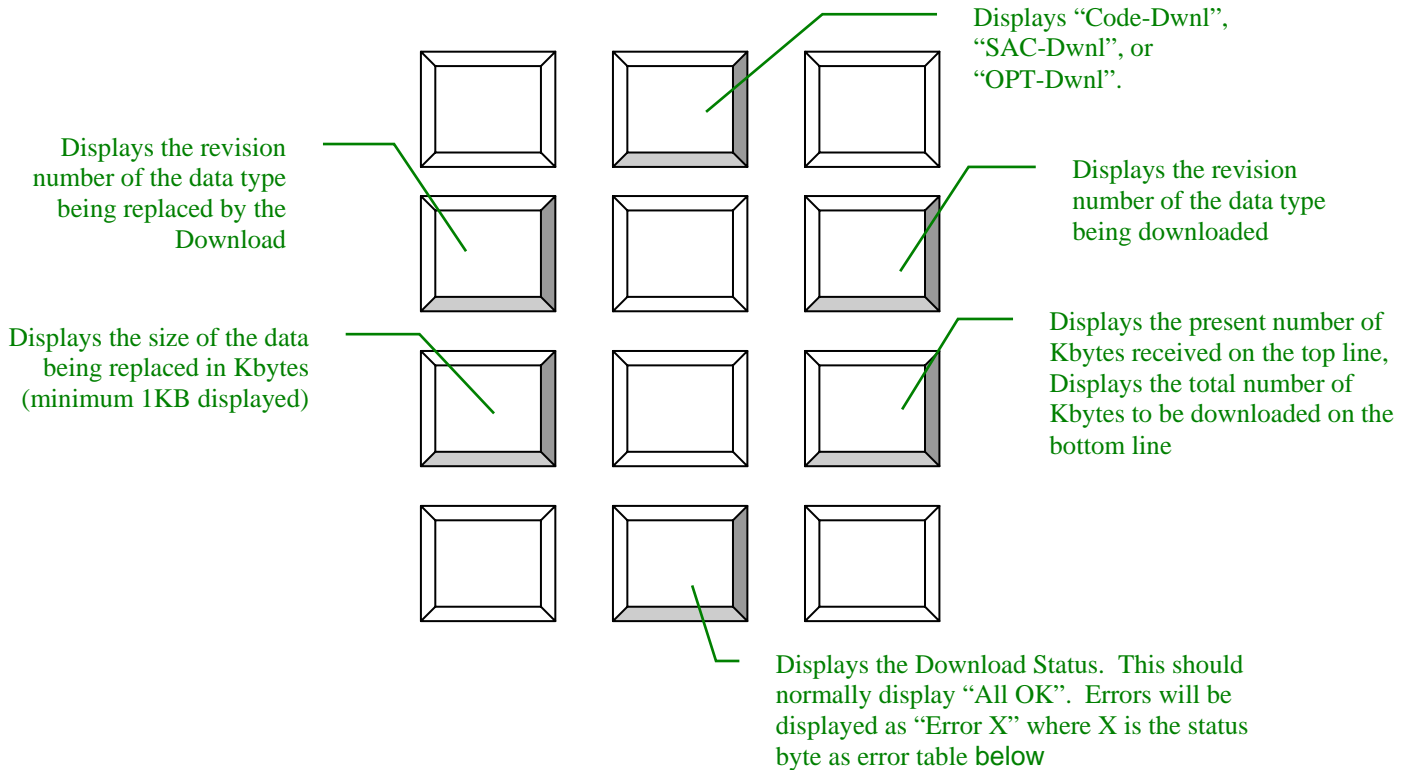
“REQD FILE ERROR session number OPERATION ABORTED”

The Application terminates immediately without attempting to retry the operation.

The required files are RTISESSN.DAT and RTIKB000 or RTIKBnnn.

9.4 Download Progress Reporting on ScreenKeys

The ScreenKey panel is used to give visual indication of the download status. The keys should be interpreted as follows:



9.4.1 Download Status Reported on ScreenKeys

This should normally display "All OK". Errors will be displayed as "Error X" where X is the status byte as error table below. If the download function terminates the SK-7510 will timeout after sixty seconds and reset. During this period "Error No DU" will be displayed.

		Error List Equivalent
All OK	Data received without error	
Error 2	Command not applicable	5011
Error 3	Invalid data type specified (ROM update / Configuration Data file Update)	5012
Error 4	Insufficient memory	5013
Error 5	Invalid placement specifier	5014
Error 6	Data record checksum error	5015
Error 7	Unexpected number of data records received	5020
Error 8	Command ignored, I'm Busy	
Error 9	Data Synchronization error	5021
Error No DU	Lost Communications with the File Download Utility or User Exits Kernel Download Function- wait for 60 seconds before resetting the ScreenKey Keyboard	n/a

The "Error List Equivalent" column lists how these error conditions are recorded in the SK-7510's Error List which is accessible via Diagnostics.

10 Messages Summary

10.1 TRACELEVEL 0 Messages Summary

TRACELEVEL 0 messages include messages from RT01 to RT19.

Serious or fatal error messages that prevent the User Exits Integration from completing successfully, e.g. communications error with keyboard. The TRACELEVEL value in the RTIKBnnn file must be set to at least 0.

		App Event Log	Trace Output
RT01	FILE MISSING - RTITERMS.DAT	4	
RT02	ERROR READING RTITERMS	4	
RT03	RTISESSN.DAT MISSING	4 ⁵	
RT04	FAILED TO GET REQD RTISESSN DATA	4 ⁵	
RT05	FAILED TO GET REQD RTIKB CONFIG DATA	4 ⁵	
RT06	FILE MISSING -- RTIKB000	4 ⁵	
RT07	KEYBOARD IS NOT RESPONDING	4	4 ²
RT08	ERROR READING FILE VERSION ID <code>FileType</code>	4	4 ²
RT09	ERROR WRITING TO SERIAL PORT	4	4 ²
RT10	ERROR READING FROM SERIAL PORT	4	4 ²
RT11	TIMEOUT READING FROM SERIAL PORT	4	4 ²
RT12	COMMUNICATIONS ERROR WITH KEYBOARD	4	4 ²
RT13	ERROR READING FILE <code>FileType</code>	4	4 ²
RT14	BYTE COUNTS DO NOT MATCH	4	4 ²

4² in order for output to be seen the RTIMSG.DAT file must be accessible.

4⁵ In addition to recording in the log the [Required File Error Handling](#) will also be used.

`FileType` is the 3 letter code that identifies the file where
“CDL” is a Code Download file (.CDL)
“OPT” is an Options download File (.ODL) and
“SAC” is a SAC download File (.SDL)

Messages shown **greyed out** are no longer used by this version of the File Download Utility and User Exit Kernel.

10.2 TRACELEVEL 1 Messages Summary

TRACELEVEL 1 messages include all the TRACELEVEL 0 messages as well as messages from RT20 to RT39.

Messages that indicate critical steps achieved in the process, e.g. file download complete, or report non-fatal errors. The TRACELEVEL value in the RTIKBnnn file must be set to at least 1.

		App Event Log	or	Trace Output
RT20	DOWNLOAD UTILITY IS STARTING	4 ¹	or	4 ¹
RT21	KEYBOARD IS NOT SKI 7510 <u>Terminal#</u>	4 ¹	or	4 ¹
RT22	DOWNLOAD UTILITY HAS FINISHED	4 ¹	or	4 ¹
RT23	ERROR READING RTIKB LAST	4 ¹	or	4 ¹
RT24	FILE NOT FOUND <u>FileType</u> <u>FileName</u>	4 ¹	or	4 ¹
RT25	WRONG FILETYPE <u>FileType</u> <u>FileName</u> <u>ByteRead</u>	4 ¹	or	4 ¹
RT26	CHECKSUM ERROR ON FILE <u>FileName</u>	4 ¹	or	4 ¹
RT27	FILE DOWNLOAD COMPLETED <u>FileType</u>	4 ¹	or	4 ¹
RT28	FILE DOWNLOAD IS NOT REQUIRED <u>FileType</u>	4 ¹	or	4 ¹
RT29	ERROR READING TEXT MESSAGES FILE	4		
RT30	DOWNLOADING <u>FileType</u> <u>% Figure</u>			4 ⁴
RT31	API NOT ENABLED	4 ¹	or	4 ¹
RT32	API ACTIVATED	4 ¹	or	4 ¹

4¹ Logged ONLY if the Messages session number in RTISESSN.DAT is zero or the RTIMSG.DAT file is otherwise inaccessible. By definition if the events are logged in the Application Event Log they cannot be sent to the TRACEOUTPUT because there is no RTIMSG.DAT file available.

4⁴ Displayed on the Output Trace display device only

10.3 TRACELEVEL 2 Messages Summary

TRACELEVEL 2 messages include all the TRACELEVEL 0 and TRACELEVEL 1 messages as well as messages from RT40 to RT59.

Additional information that may be useful in debugging situations. The TRACELEVEL value in the RTIKBnnn file must be set to 2.

		App Event Log	Trace Output
RT40	OPENING TERM. LIST FILE- RTITERMS.DAT		4 ³
RT41	TERM. NUMBER NOT IN RTITERMS.DAT		4 ³
RT42	DOWNLOAD FILE NAME: NOT SPECIFIED		4 ³
RT43	DOWNLOAD FILE NAME: <u>FileType</u> <u>FileName</u>		4 ³
RT44	USE SERIAL PORT NUMBER <u>PortNumber</u>		4 ³
RT48	OPENING DOWNLOAD FILE <u>FileName</u>		4 ³
RT49	FILE DOWNLOAD BYPASSED <u>FileType</u>		4 ³
RT50	NUMBER OF BYTES/BLKS <u>Bytes/Blocks</u>		4 ³
RT51	UNABLE TO WRITE TO SERIAL PORT		4 ³
RT52	DATA SYNCHRONIZATION ERROR		4 ³
RT53	WRITE RETRIES TO KEYBOARD FAILED		4 ³
RT54	UNABLE TO RESET KEYBOARD		4 ³
RT55	KEYBOARD RESET DUE TO WRITE ERROR		4 ³
RT56	TEXT MESSAGES FILE OPENED		4 ³

4³ There will be no TraceOutput if the RTIMSG.DAT file is unavailable.

11 Message Descriptions

RT01 "FILE MISSING -- RTITERMS.DAT"

Severity: 3

Reported to: Application Event Log only.

Explanation: The attempt to OPEN the RTITERMS.DAT file failed. The reasons for such a failure include:

- 1) the RTITERMS User Logical File name has not been set-up correctly
- 2) The RTITERMS.DAT file is not stored in the location specified by the RTITERMS User Logical File name

If this error is reported by the User Exits Integration then another reason might be

- 3) There is a clash on the Session Number used to access the RTITERMS.DAT file. The User Exits default to Session Number 50 but this can be changed by modifying RTISESSM.C86 and recompiling/linking the Terminal Sales Application.

If the error is reported by the RTIFDU.286 then the program terminates without attempting to download to the SK-7510

If the error is reported by the User Exits integrated into the Terminal Sales Application then the download function terminates without downloading to the SK-7510 and the Terminal Sales Application continues as normal.

User Response: This is an error only if the terminal has a SK-7510 attached and therefore the Download function required access to the RTITERMS.DAT file. The Download function did not happen and therefore the SK-7510 may not have the correct download files. Correct the problem and re-run the download function.

System Action: Logged in the Application Event Log.

RT02 "ERROR READING RTITERMS"

Severity: 3

Reported to: Application Event Log only.

Explanation: The RTITERMS.DAT file has been opened successfully but there was a problem reading the file or else the Terminal Number was not listed in the file.

The Download function terminates without doing the download. The SK-7510 has not been updated.

User Response: This is an error only if the terminal has a SK-7510 attached and therefore the Download function required access to the RTITERMS.DAT file. Edit the RTITERMS.DAT file to include this terminal number in the list and try again.

System Action: Recorded in the Application Event Log.

RT03 "RTISESSN.DAT MISSING"
and "B040 FILE ACCESS FAILED session number" or "REQD FILE ERROR session number
OPERATION ABORTED"

Severity: 3

Reported to: Application Event Log and System Display.

Explanation: The attempt to OPEN the RTISESSN.DAT file failed. The reasons for such a failure include:

- 1) the "RTISESSN" User Logical File name has not been setup correctly
- 2) The RTISESSN.DAT file is not stored in the location specified by the "RTISESSN" User Logical File name

If this error is reported by the User Exits Integration then another reason might be

- 3) There is a clash on the Session Number used to access the RTISESSN.DAT file. The RTIFDU.286 uses Session Number 50 for this purpose. The SK-7510 User Exits default to session Number 50 but this can be changed by modifying RTISESSM.C86.

If running the RTIFDU.286 application then the program ends. The download was unsuccessful. Fix the problem and retry.

If running the User Exit Integration then the system will display a B040 message and wait for the CLEAR key to be pressed. When the CLEAR key is received the program will retry the operation. If it fails again it will report the error and wait for the CLEAR. If the problem is fixed the application will continue. The Program will loop forever if the problem is not fixed.

The Trace output is displayed on the System Display (if the RTIKBnnn parameters are setup to allow this) and then 5 seconds later the "REQUIRED FILE NOT FOUND" message is displayed on the System Display.

User Response: Correct the problem with the RTISESSN.DAT file

System Action: Recorded in the Application Event Log. Also, if running the User Exit Integration then the "B040 FILE ACCESS FAILED session number" processing will be triggered.

RT04 "FAILED TO GET REQD RTISESSN DATA "
and "B040 FILE ACCESS FAILED session number" or "REQD FILE ERROR session number
OPERATION ABORTED"

Severity: 3

Reported to: Application Event Log and System Display.

Explanation: The RTISESSN.DAT file has been opened successfully but there was a problem reading the file.

If running the RTIFDU.286 application then the User Exits Integration will display the "REQD FILE ERROR session number OPERATION ABORTED" message on ANDISPLAY (where *session number* will be 50 - the session number used to attempt to open the RTISESSN.DAT file). The program ends. The download was unsuccessful.

If running the User Exit Integration then the system will display a B040 message and wait for the CLEAR key to be pressed. When the CLEAR key is received the program will retry the operation. If it fails again it will report the error and wait for the CLEAR. If the problem is fixed the application will continue. The Program will loop forever if the problem is not fixed.

The Trace output is displayed on the System Display (if the RTIKBnnn parameters are setup to allow this) and then 5 seconds later the "REQUIRED FILE NOT FOUND" message is displayed on the System Display.

User Response: Fix the problem and retry.

System Action: Recorded in the Application Event Log. Also, if running the User Exit Integration then the "B040 FILE ACCESS FAILED session number" processing will be triggered.

RT05 "FAILED TO GET REQD RTIKB CONFIG DATA "
and "B040 FILE ACCESS FAILED session number" or "REQD FILE ERROR session number
OPERATION ABORTED"

Severity: 3

Reported to: Application Event Log and System Display.

Explanation: The RTIKBnnn file did not contain the required data. The minimum RTIKBnnn file must contain values for at least..

"ROMFILENAME=", "file_cdl.cdl"

```
"SACFILENAME=", "file_sdl.sdl"  
"CFGFILENAME=", "file_old.odl"  
"SERIALPORT#=", 2
```

If running the RTIFDU.286 application then the User Exits Integration will Display the "REQD FILE ERROR session number OPERATION ABORTED" message on ANDISPLAY (where *session number* will be 50 - the session number used to attempt to open the RTIKBnnn file). The program ends. The download was unsuccessful.

If running the User Exit Integration then the System will display a "B040 FILE ACCESS FAILED session number" message (where session number will be 50 by default or whatever session number has been edited into the RTISESSM.C86 User Exit code). The Program will loop forever if the problem is not fixed.

The Trace output is displayed on the System Display (if the RTIKBnnn parameters are setup to allow this) and then 5 seconds later the "REQUIRED FILE NOT FOUND" message is displayed on the System Display.

User Response:

System Action: Recorded in the Application Event Log.

RT06 "FILE MISSING -- RTIKB000"
and "B040 FILE ACCESS FAILED session number" or "REQD FILE ERROR session number
OPERATION ABORTED"

Severity: 3

Reported to: Application Event Log and System Display and Trace Output.

Explanation: The attempt to OPEN the RTIKBnnn (where nnn is the 3 digit terminal number) file failed and the attempt to open the RTIKB000 file also failed. The reasons for such a failure include:

- 1) the "RTIK:" User Logical File name has not been setup correctly
- 2) The "RTIKB000" or RTIKBnnn" file is not stored in the location specified by the "RTIK:" User Logical File name
If this error is reported by the User Exits Integration then another reason might be
- 3) There is a clash on the Session Number used to access the RTIKB000 and RTIKBnnn file. The RTIFDU.286 uses Session Number 50 for this purpose. The SK-7510 User Exits default to session Number 50 but this can be changed by modifying RTISESSM.C86.

If running the RTIFDU.286 application then the User Exits Integration will display the "REQD FILE ERROR session number OPERATION ABORTED" message on ANDISPLAY (where *session number* will be 50 - the session number used to attempt to open the RTISESSN.DAT file). The program ends. The download was unsuccessful.

If running the User Exit Integration then the system will display a B040 message and wait for the CLEAR key to be pressed. When the CLEAR key is received the program will retry the operation. If it fails again it will report the error and wait for the CLEAR. If the problem is fixed the application will continue. The Program will loop forever if the problem is not fixed.

The Trace output is displayed on the System Display (if the RTIKBnnn parameters are setup to allow this) and then 5 seconds later the "REQUIRED FILE NOT FOUND" message is displayed on the System Display.

User Response:

System Action: Recorded in the Application Event Log.

RT07 "KEYBOARD IS NOT RESPONDING"

Severity: 3

Reported to: TraceOutput and the Application Event Log

Explanation: The Terminal communicates with the SK-7510 via the Serial Port Channel. The Serial Channel is Opened using the session number value specified in the "SERIAL=" setting in the RTISESSN.DAT file and "SERIALPORT#=" value specified in the RTIKBnnn file. If the open fails the code will continue to retry to open the Serial Channel every half second for the number of seconds specified in the "KBDTIMEOUT=" setting in the RTIKBnnn file.

The Download will terminate and not attempt to download this or any more files.

User Response: One reason the Serial channel may not be available is because the SK-7510 may be temporarily offline and not responding. For instance, the SK-7510 will be offline for between 15 and 30 seconds after a code download. So, check that the "KBDTIMEOUT=" setting in RTIKBnnn is sufficient. The maximum value allowed is 49.

There may also be a clash on the Session Number used. Check the value used for the "SERIAL=" setting in the RTISESSN.DAT file.

The serial channel may not be setup correctly in the Terminal Device Group. Either it is configured so that the device address does not match the Serial Channel device address specified in the SK-7510 EEPROM or the port number specified in the TDG configuration does not match the port number in the RTIKBnnn file.

System Action: Recorded in the Application Event Log.

RT08 "ERROR READING FILE VERSION ID *FileType*"

Severity: 3

Reported to: TraceOutput and the Application Event Log

Explanation: The attempt to read the File Type and File ID from the File Header failed. The Download terminates without attempting any remaining downloads.

User Response: Check that this is a valid download file.

System Action: Recorded in the Application Event Log.

RT09 "ERROR WRITING TO SERIAL PORT"

RT10 "ERROR READING FROM SERIAL PORT"

RT11 "TIMEOUT READING FROM SERIAL PORT"

RT12 "COMMUNICATIONS ERROR WITH KEYBOARD"

Severity: 3

Reported to: TraceOutput and the Application Event Log

Explanation: Problem with the Serial Channel. The Download terminates without attempting any remaining downloads.

User Response: Contact Tech Support

System Action: Recorded in the Application Event Log.

RT13 "ERROR READING FILE *FileType*"

Severity: 3

Reported to: TraceOutput and the Application Event Log

Explanation: There was a problem reading the File.

User Response: Contact Technical Support

System Action: Recorded in the Application Event Log.

RT14 "BYTE COUNTS DO NOT MATCH"

Severity: 3

Reported to: TraceOutput and the Application Event Log

Explanation: The number of bytes sent to the SK-7510 by the User Exits Integration does not match the number of bytes that the SK-7510 has received. The User Exits Integration terminates without completing the download that it is currently sending and does not attempt any other downloads.

User Response: Contact Technical Support

System Action: Recorded in the Application Event Log.

RT20 "DOWNLOAD UTILITY IS STARTING"

Severity: 3

Reported to: TraceOutput

Explanation: This progress message indicates that the RTITERMS.DAT, RTISESSN.DAT, RTIMSG.DAT (optional) and RTIKBnnn files have been opened successfully and the Download to the SK-7510 is about to start.

The RTIMSG.DAT file is optional so the Download Utility will run but it will not display messages - not even this one.

User Response: None

System Action: Recorded in the Application Event Log if TraceOutput is off.

RT22 "DOWNLOAD UTILITY HAS FINISHED"

Severity: 3

Reported to: TraceOutput

Explanation: This progress message indicates that the download code is finished processing.

The RTIMSG.DAT file is optional so the Download Utility will run but it will not display messages - not even this one.

User Response: None

System Action: Recorded in the Application Event Log if TraceOutput is off.

RT24 "FILE NOT FOUND *FileType FileName*"

Severity: 3

Reported to: TraceOutput

Explanation: The attempt to open the download file name specified in the RTIKBnnn file using the "RTI:" User Logical File Name and session number(*) failed. The error message is reported to the Trace Output devices and a one and a half second later (to give the user a chance to see the message) the program proceeds to try the next download file.

(*) the Session number is 50 in the FDU and by default in the User Exit Kernel. The User Exit Kernel may be changed by editing the RTISESSM.C86 file.

User Response: None

System Action: Recorded in the Application Event Log if TraceOutput is off.

RT25 "WRONG FILETYPE *FileType* *FileName* *ByteValue* "

Severity: 3

Reported to: TraceOutput

Explanation: The *FileType* is part of the first block of data in the file. When we compare the File Type we are looking for (*FileType*) with the file type in the file itself (*ByteValue*) they did not match. The attempt to download this file is abandoned and the application moves on to the next download file.

FileType is the 3 letter code that identifies the file where

"CDL" is a Code Download file (.CDL)
"OPT" is an Options download File (.ODL) and
"SAC" is a SAC download File (.SDL)

ByteValue is a value read from the file. It should be 1 for .CDL files, 2 for .SAC files and 3 for .ODL files.

User Response: Correct the problem and rerun the Download

System Action: Recorded in the Application Event Log if TraceOutput is off.

RT26 "CHECKSUM ERROR ON FILE *FileName*"

Severity: 3

Reported to: TraceOutput

Explanation: The Checksum is calculated by reading through the whole file. If it's incorrect then the attempt to download this file is abandoned and the application moves on to the next download file.

User Response: Correct the problem and rerun the Download.

System Action: Recorded in the Application Event Log if TraceOutput is off.

RT27 "FILE DOWNLOAD COMPLETED *FileType*"

Severity: 3

Reported to: TraceOutput

Explanation: This file has been downloaded successfully.

User Response: None

System Action: Recorded in the Application Event Log if TraceOutput is off.

RT28 "FILE DOWNLOAD IS NOT REQUIRED *FileType*"

Severity: 3

Reported to: TraceOutput

Explanation: The file that is already in the SK-7510 is the same as the file on the Controller in the location specified by the "RTI:" User Logical Filename. The File Name is taken from the RTIKBnnn settings.

User Response: None

System Action: Recorded in the Application Event Log if TraceOutput is off.

RT29 "ERROR READING TEXT MESSAGES FILE"

Severity: 3

Reported to: Application Event Log only - regardless of TRACELEVEL setting

Explanation: The RTIMSG.DAT file is opened only if the "MESSAGES=" session number taken from the RTISESSN.DAT file is not zero. An attempt is made to open the RTIMSG.DAT file using the Session number from the RTISESSN.DAT file and the "RTI:" user Logical File Name.

This is not a fatal error. The download procedure will continue. However, there will be no TraceOutput. TRACELEVEL 0 messages will continue to be logged to the Application Event Log. In Addition TRACELEVEL 1 messages will also be logged to the Application Event Log even if the TRACELEVEL setting is 0.

User Response: None

System Action: None.

RT30 " DOWNLOADING *FileType %figure* "

Severity: 3

Reported to: Display device only if TraceOutput is active and TRACELEVEL is greater than zero.

Explanation: Progress message showing the percentage complete figure. The progress reporting frequency for large files (≥ 20 blocks) is 20 times, medium (≥ 5 blocks)report 5 times small report every block - at most 4 times.

Shown on the Display device only. Never sent to the Application Event Log.

User Response: None

System Action: None.

RT31 "API NOT ENABLED"

Severity: 3

Reported to: TraceOutput if the TRACELEVEL is set to 1 or 2

Explanation: This progress message indicates that API channel is not available because the "API=" session number in RTISESSN.DAT is missing/zero OR because we tried and failed to OPEN "ANDISPLAY3:".

Opening the API Channel is a separate job from the Download and will still be attempted even if the Download fails.

User Response: No response is required if the user has chosen to disable the API. However, if the user selected to activate the API and it's reported as "NOT ENABLED" then remedial action is required. There may be a Session Number clash on session number used for the API channel. This is 51 by default but is set in the "API=" value of RTISESSN.DAT.

System Action: Recorded in the Application Even Log if TraceOutput is off.

RT32 "API ACTIVATED"

Severity: 3

Reported to: Trace Output if the TRACELEVEL is set to 1 or 2

Explanation: This progress message indicates that API channel is available because "ANDISPLAY3:" has been opened successfully.

Opening the API Channel is a separate job from the Download and will still be done even if the Download fails.

User Response: No response is required if the user has chosen to activate the API Channel by assigning a non-zero session number to the "API=" setting in the RTISESSN.DAT file. However, if the API is Active and the intention was to disable the API then edit the session numbers file - RTISESSN.DAT - and set the API session number to zero.

System Action: Recorded in the Application Even Log if TraceOutput is off.

RT42 "DOWNLOAD FILE NAME: NOT SPECIFIED"

Severity: 3

Reported to: Trace Output

Explanation: This progress message indicates that the Download File Names read from the RTIKBnnn file was specified as "" (empty) or set to all spaces so a download will not be attempted for this file.

User Response: None

System Action: None.

RT43 "DOWNLOAD FILE NAME: *FileName*"

Severity: 3

Reported to: TraceOutput

Explanation: This progress message reports the Download File Names read from the RTIKBnnn file.

User Response: None

System Action: None.

RT44 "USE SERIAL PORT NUMBER *PortNumber*"

Severity: 3

Reported to: Trace Output if the TRACELEVEL is set to 2

Explanation: This progress message shows the Serial Port Number read from the RTIKBnnn file.

User Response: None

System Action: None.

RT48 "OPENING DOWNLOAD FILE *FileName*"

Severity: 3

Reported to: TraceOutput if TRACELEVEL is set to 2

Explanation: This progress message shows the Download File Name before the attempt to open the file. *FileName* is the name taken from the RTIKBnnn file.

User Response: None

System Action: None.

RT49 "FILE DOWNLOAD BYPASSED *FileType*"

Severity: 3

Reported to: TraceOutput if TRACELEVEL is set to 2

Explanation: See RT42.

User Response: None

System Action: None.

RT50 "NUMBER OF BYTES/BLKS *FileSize/NumBlocks*"

Severity: 3

Reported to: TraceOutput if TRACELEVEL is set to 2

Explanation: This progress message displays the size of the file that is about to be downloaded in bytes and also in 240 byte blocks.

User Response: n/a

System Action: None.

RT51 "UNABLE TO WRITE TO SERIAL PORT"

Severity: 3

Reported to: TraceOutput if TRACELEVEL is set to 2

Explanation: This message indicates a problem with the Serial Communications.

User Response: Contact Tech Support

System Action: None.

RT52 "DATA SYNCHRONIZATION ERROR"

Severity: 3

Reported to: TraceOutput if TRACELEVEL is set to 2

Explanation: This message corresponds to an "Error 9" message on the ScreenKey download Status display. It indicates that the number of blocks sent by the application to the SK-7510 does not match the number of Blocks received by the SK-7510.

User Response: Contact Tech Support

System Action: None.

RT53 "WRITE RETRIES TO KEYBOARD FAILED"

Severity: 3

Reported to: TraceOutput if TRACELEVEL is set to 2

Explanation: This message indicates a problem with the Serial Communications.

User Response: Contact Tech Support

System Action: None.

RT54 "UNABLE TO RESET KEYBOARD"

Severity: 3

Reported to: TraceOutput if TRACELEVEL is set to 2

Explanation: This message indicates a problem with the Serial Communications.

User Response: Contact Tech Support

System Action: None.

APPENDIX A Documentation Control

A.1 Change Control

This document is the responsibility of the author and is subject to formal change control after the initial approved release (i.e. issue 1.0).

A.2 Abbreviations Used/Terms of Reference

API	Applications Programming Interface
Backed-up RAM	Internal RAM in the keyboard which retains information for a period of time dependant upon the charged state of the keyboard. The keyboard is continuously charged while connected to a terminal
BCR	Bar Code Reader
CDA	Control Display Area, pop-up 1x20 large font display on bottom of large-panel LCD screen
CDL	Code Update – new runtime application file that can be downloaded into keyboard to alter its mode of operation, i.e. similar to microcode update
GSA	General Sales Application – sales application running on IBM 46xx sales terminals
MCR	Magnetic Card Reader
ODA	Operator Display Area, upper large text area on the SK-7510 large-panel LCD screen
ODL	Options DownLoad file.
OPT	Options download file
PC	Personal Computer.
POS	Point of Sale—the cash register in a shop.
SA	Supermarket Application – sales application running on IBM 46xx sales terminals
SAC	ScreenKey Active Control, file that defines the set of drill-down menus to be implemented on the ScreenKeys
SDL	SAC Download file after conversion by MakeDNL utility
TDA	Transaction Display Area, lower small text section on the SK-7510 large-panel LCD screen
TDG	Terminal Device Group, specifies the devices attached to the terminal and to which sockets they are attached

A.3 Historical Change Reference

Issue	Date	Author	Changes Made
1.0	08/02/01	M. McDonnell	Initial Release
1.1	01/09/01	U Carroll	Updated based on feedback from the field.
1.2	10/05/05	J Blomquist	Updated SKI references
1.3	3/14/06	J Blomquist	Removed references to discontinued SK-3510

A.4 Change Summary

- 1.1 Information reorganised and expanded. Added to the “Problem Determination” section.
Added the Checklist. Added a comparison of User Exits V File Download Utility...
- 1.2 Updated RTI references to SKI
- 1.3 Removed references to discontinued SK-3510

APPENDIX B File Download Utility v User Exit Integration

The following table compares and contrasts the SK-7510 File Download Utility and the SK-7510 User Exit Kernel.

File Download Utility	User Exit Kernel
<p>What does it do?</p> <p>The purpose of the File Download Utility (RTIFDU.286) is to ensure that the correct Code (.CDL), SAC (.SLD) and Options (.ODL) files are installed in the SK-7510 non-volatile memory.</p> <p>It compares the versions of these files stored in the SK-7510 with the version stored on the Controller. If they are different it downloads the files from the Controller to the SK-7510.</p>	<p>The same.</p> <p>However, this is not the only purpose of the User Exit kernel. The User Exit kernel also enables the Terminal Sales Application to control additional SK-7510 features using the API. For example, the application can display data on the SK-7510's LCD Panel. The Download is a core part of the User Exit Kernel. The API features are optional.</p>
<p>When is it Run?</p> <p>The File Download Utility is a standalone .286 program – separate from the Terminal Sales Application. If you want it to check/download to the SK-7510 you must explicitly run it on the terminal. Once it does its job it terminates. Typically the sequence of actions would be to a) stop the Terminal Sales Application, b) Start the File Download Utility, c) when the FDU terminates start the Terminal Sales application again.</p>	<p>The User Exits Kernel must be integrated into the Terminal Sales Application (TSA).</p> <p>Integration is the recommended way to handle the Downloads as it enables the Terminal Sales Application, at its discretion, to trigger a download thus ensuring that the latest files are always in the SK-7510. The code to check / download can be run every time the TSA is started and also every time it is personalised.</p> <p>The additional features, if any, which were coded at User Exit Integration time will be active as well.</p>
<p>What Configuration Files does it use?</p> <p>The File Download Utility uses the RTITERMS.DAT, RTIKBnnn, RTISESSN.DAT and RTIMSG.DAT files. These files must be available in the directory specified by the Logical File Names</p>	<p>The same.</p>
<p>What about Session Numbers?</p> <p>Since this is a standalone program it is free to use whatever session numbers it likes apart from those hardcoded numbers used in the Download Utility itself (see page 24). However, since the File Download utility is often a prelude to using the User Exits it would make sense to use session numbers that are acceptable to the User Exits.</p>	<p>RTISESSN.DAT specifies which Session Numbers the User Exit code is allowed to use. It is crucial that those chosen don't clash with Session Numbers already in use by the Terminal Sales application.</p> <p>There is also a Session Number in RTISESSM.C86 than can be changed.</p>
<p>How should the Terminal be Configured?</p> <p>The Terminal Device Group must be configured to include a Serial Port. The File Download Utility needs this serial port in order to communicate with the SK-7510 for the purpose of the download.</p>	<p>The same.</p> <p>In addition, the Terminal Device Group must be configured to include an ANDISPLAY3 device if the API features of the User Exit Kernel are to be used.</p>
<p>How should the Controller be Configured?</p> <p>The File Download Utility uses a number of Logical Files Names to locate the SKI Files.</p>	<p>The same.</p>
<p>How should the SK-7510 be Configured?</p> <p>The SK-7510 must be configured (EEPROM) to support the same serial port (a device address rather than a Port number) as specified in the Terminal Device Group</p>	<p>The same.</p> <p>Also, the Options Download (.ODL) should enable the API device (API channel) if required.</p>

User Exits Integration Checklist

Installation Site:

Terminal Number :

Terminal Device Group:

Date :

Use this checklist to ensure that all the items are correctly configured to enable the SK-7510 Kernel to be integrated into the Terminal Sales Application via the User Exits.

SK-7510 Keyboard Configuration

- Configure the EEPROM (SK-7510 Diagnostics menu) to set the Serial Port device address (Typically 64) the Keyboard device address (typically 10) and in the case of the SK-7510, the Operator Display Device address (Typically 22)

Terminal Configuration

Terminal Device Group Setup

- Set the Serial device address to match that selected in EEPROM and set the Port number to match that selected in the RTIKBxxx file (see below)
- Set the Keyboard device to match that selected in EEPROM
- Set the Operator Display device to match that selected in EEPROM
- Set other Devices to match the content of the Options Download File (see .ODL below)

Keyboard Layout Setup

- Create a Keyboard layout for a 4683 keyboard ("old 50Key"), if one doesn't already exist

Terminal Load Definition Setup

- Set the Terminal Device Group and Keyboard layout to those that were created to support the SK-7510

Activate Terminal Configuration

- Activate the Terminal Configuration
- Re-IPL the Terminal or "Load Terminal Configuration" from the terminal commands menu

Download Files Configuration

Options Download (.ODL)

- Use MAKEDNL to create an options (.ODL) file that is used to further configure the SK-7510 – including Magnetic Stripe reader, Printer Monitoring etc...

SAC Download (.SDL)

- Use The SAC Editor (46XX version) to create a legend file (.LEG) that matches the Keyboard layout selected in the Terminal Device Group (see above)
- Use The SAC Editor (46XX version) to create a SAC File (.PKF)
- use MAKEDNL to convert the SAC file into a downloadable SAC file (.SDL)

Code Download (.CDL)

- Ensure that you have the latest Code Download File (.CDL) – it's supplied as part of the SK-7510 Kernel
-

Controller Configuration

Set Logical File Names

- Set the logical File Names

Install SKI Files

- Copy the SKI files into the directories specified in the Logical File Names (RTITERMS.DAT, RTIKBxxx, RTISESSN.DAT, RTIMSG.DAT, the Code file (.CDL), the Options File (.ODL) and the ScreenKeys file (SDL))

Configure SKI Files

- configure the RTITERMS.DAT file (specify which terminals have SK-7510 keyboards)
- configure the RTIKBxxx file (to specify the Serial Port Number – must match the port number assigned in the terminal device Group – the names of the .CDL, .ODL and .SDL files to download, and the level of progress reporting (Tracing) required)
- configure the RTISESSN file to avoid Session Number Clashes

Activate Controller Configuration

- Activate the Controller Configuration
- Reboot the Controller

Terminal Sales Application Configuration

User Exit Source Code Modifications

- Modify the existing Terminal sales application User Exits to include the SK-7510 User Exit Code
- The default session number is 50. Choose a different Session Number if this clashes

Compile, Link and PostLink

- After compilation remember to rename the .OBJ to .LBJ as appropriate (suggested naming convention for large / medium memory model objects)
- Before Linking make sure that the appropriate Object modules are included in the Linker .INP file

Application Personalisation

- Activate User Exits as appropriate
- Configure Journal / Receipt output as appropriate (if printer monitoring is selected in the Options file)
-