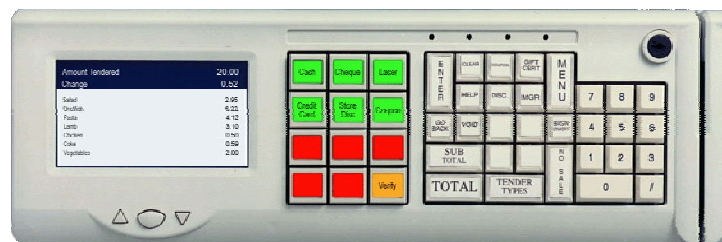




ScreenKey Consoles for the IBM 46xx

MakeDnl
Download Conversion
Utility
User's Guide
Version 1.7

Issue 1.2 March 2006



Purpose

The purpose of this document is to explain how to use MakeDnl software package, to convert SAC files and create and maintain Option files for ScreenKey IBM 46xx keyboards. The package runs on a Windows platform but creates download files for IBM 46xx target systems.

This document is based on MakeDnl release 1.7.

www.ScreenKeys.com

SKI

SK Interfaces Ltd.

MakeDnl – Download Conversion Utility

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.products@ScreenKeys.com

via Web: www.ScreenKeys.com

© 2002-2006 SK Interfaces Ltd.

© 1996-2002 Rapid Technology Interfaces Ltd.

© 1996-2002 Feltscope Ltd.

All rights reserved.

DISCLAIMER:

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

If data file formats are revised, the *FileVersion* field in the HEADER record will be updated. Utility programs written to process the data file should check this field.

Foreword

ScreenKey Keyboards are available for Windows and IBM 46xx target platforms. These operating systems are very different and require different ScreenKey support software and utilities. Please see ScreenKeys website (www.ScreenKeys.com) for further information.

ScreenKey Active Control (SAC) files are used extensively by RTI in providing solutions for both target platforms. The SAC Editor software package is used to create and maintain SAC files for both systems although the Editor itself only runs on Windows platforms. SAC files for IBM 46xx systems must be created on a Windows PC and transferred to the 46xx runtime system.

The SAC Editor is shipped with different installs for each target platform (i.e. SAC Editor and SAC Editor for IBM 46xx).

MakeDnl is used to convert SAC files created with the SAC Editor into a format suitable for download to IBM 46xx ScreenKey Keyboards. It is also used to create POS terminal specific configuration files.

Other helpful documents are:

ScreenKey Planning and Installation Guide for IBM 46xx

Installation instructions and how to get started with ScreenKey keyboards with IBM 46xx systems.

ScreenKey User Exits Integration for IBM 46xx

How to install user exit kernel support with SA and GSA sales applications.

ScreenKey File Download Utility for IBM 46xx

How to install and operate the FDU on IBM 46xx POS terminals.

In this document pointing hands in the left margin are used to indicate differences for certain platforms and keyboard types.



Note



An upward pointing hand indicates that the whole chapter only applies to the description below the pointing hand.



A rightward pointing hand indicates a difference in that section if the hand is pointing to a section or a difference in the paragraph if the hand is pointing to a paragraph.

Most of the information contained in this document is also available via the on-line help.

Table of Contents

1	INTRODUCTION	7
2	MAKEDNL UTILITY INSTALLATION	9
3	FILE DOWNLOADS ON IBM 46XX POS SYSTEMS.....	11
4	THE MAKEDNL UTILITY	13
4.1	MENU SELECTIONS	15
4.2	HOW TO CONVERT A SAC FILE	16
4.3	HOW TO CREATE/EDIT AN OPTIONS FILE	17
4.4	INCLUDE/EXCLUDE INDIVIDUAL OPTION RECORD	18
4.5	HOW TO VIEW A DOWNLOAD FILE HEADER	18
5	OPTIONS RECORD DETAILS.....	19
5.1	SCREENKEY CHARACTER FONTS.....	19
5.2	KEYBOARD ATTRIBUTE TABLE (KAT).....	20
5.3	SPLASH SCREEN FILE	21
5.4	POWER-ON BEEPS	22
5.5	KEY BEEPS	23
5.6	MAGNETIC STRIPE READER EMULATION.....	24
5.6.1	<i>Purpose</i>	24
5.6.2	<i>Options File Setup</i>	24
5.7	BAR CODE READER EMULATION (BUFFER OVERFLOW WORKAROUND).....	26
5.7.1	<i>Purpose</i>	26
5.7.2	<i>Options File Setup</i>	26
5.8	BARCODE/STEMCODE CONVERSION TABLE.....	27
5.9	EXECUTION TIMES.....	28
5.10	OPERATOR DISPLAY AREA CHARACTER FONT	30
5.10.1	<i>Purpose</i>	30
5.10.2	<i>Options File Setup</i>	30
5.11	PRINTER MONITORING.....	32
5.11.1	<i>Purpose</i>	32
5.11.2	<i>Options File Setup</i>	32
5.12	TRANSACTION DISPLAY AREA CHARACTER FONT	34
5.13	API (VIA VIRTUAL DISPLAY EMULATION).....	35
5.13.1	<i>Purpose</i>	35
5.13.2	<i>Options File Setup</i>	35
5.14	CDA\USER MENU	36
5.15	SCREENSAVER.....	37
APPENDIX A	DOCUMENTATION CONTROL.....	39

1 Introduction

ScreenKey Active Control (SAC) files are used extensively by RTI in providing solutions for Windows and IBM 46xx target platforms. SAC files are self-contained data-files containing a set of drill-down menus for the ScreenKeys, the relationship between these menus, and other applicable rules and data.

The SAC Editor software package is used to create and maintain SAC files for both systems although the Editor itself only runs on Windows platforms. SAC files for IBM 46xx systems, created on a Windows PC, must be converted into a format suitable for download to an IBM 46xx ScreenKey Keyboard for use on an IBM 4690OS POS terminal.

MakeDnl is a Windows utility that is used to convert SAC files (.pkf extension) created with the SAC Editor into the SAC download format (.sdl extension).

It is also used to create POS terminal specific configuration files. Each POS terminal can have many different POS peripherals attached. The ScreenKey Keyboard must operate in unison with these devices. It must be setup to ensure that there is no conflict between the device emulations offered by the ScreenKey Keyboard and the other attached peripherals. Downloaded Option files are used to define the configure of each IBM 46xx ScreenKey Keyboard.

MakeDnl Windows utility allows the creation and maintenance of Option files. These files may be downloaded directly to the ScreenKey Keyboard.

Options files also include other keyboard configuration information, such as font files for each of the different displays, screensave options for the large-panel display on the SK-7510, power-up splash screen images, etc.

This document describes the different options available with MakeDnl Download Conversion Utility.

2 MakeDnl Utility Installation

This software will run on Windows 95/98/NT/2000.

MakeDnl may be distributed in a number of ways, either via web download, as one module in a CD software bundle or on a set of floppy disks.

The MakeDnl web download is self-installing. From other distribution media, run SETUP.EXE to begin the installation process. If MakeDnl is distributed as part of a software bundle, SETUP.EXE may reside in a sub-folder off the root. A helper installation utility (e.g. autorun HTML file) may also be provided for handling installation of MakeDnl and other elements in the bundle.

Follow on-screen instructions during the installation.

NOTES:

1. MakeDnl install searches for old "ScreenKey for Windows" installations. These include SAC File Editor and ScreenKey Emulator applications that have been superceded. If found, the user is prompted to uninstall these using Windows Add/Remove Utility before continuing. You may continue the install without removing these if desired.
2. The install will only prompt for a installation folder if this is the first installation of a Windows ScreenKey product using this folder structure. If another Windows ScreenKey product has already been installed, then that destination path will be used without prompting the user for a new destination.
3. Other Windows ScreenKey products that use this folder structure include:
 - SAC Editor
 - SAC Editor for IBM 46xx
 - ScreenKey ActiveX Control
 - SAC Engine Control
 - SAC Controller

Items installed during the installation include...

File Name	Description
MakeDnl.EXE	Utility to edit/create Options file and to convert a SAC file (.pkf) to download format (.sdl)
SFont.PLF	TDA Character set – Code Page 850 Multilingual
LFont.SDF	ODA Character set – Code Page 850 Multilingual
SK_Char.SKF	Character set for ScreenKeys – RTI standard
Splash.BMP	SK-7510 Splash Screen
Kat.KAT	Default KAT file

See the README file that comes with the Install for details of the changes in the latest version and the product revision history.

The purpose of each of the support files is discussed in the relevant subsections of section 5.

Alternative files may be made available by RTI to support different code pages in the future. RTI may be contacted for details of file structures and relevant information required to create or edit them.

3 File Downloads on IBM 46xx POS Systems

Options files are created and maintained using MakeDnl on the Windows PC. When complete, these files must be moved to the 46xx POS controller and then downloaded via the 46xx POS terminals to the attached ScreenKey Keyboards.

Similarly, SAC files created with the SAC Editor must be converted into downloadable format (.sdl) and transferred to the POS 46xx controller.

Once downloaded into the ScreenKey keyboard, these files are held in backed up RAM.

Two methods are available for downloading files into the ScreenKey Keyboard's memory. Both of these run on IBM 4690OS controller platforms and download into ScreenKey Keyboards attached to IBM 46xx POS terminals.

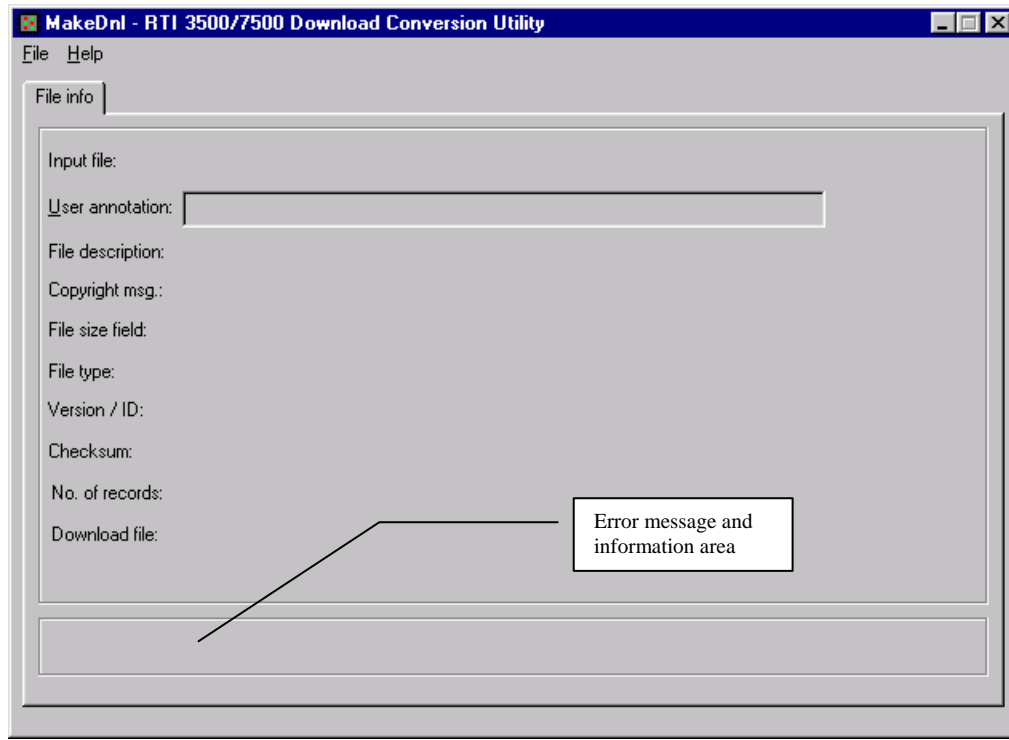
1. File Download Utility (FDU) is a standalone utility that coordinates the download of SAC, Options and Code Update files into the keyboard memory
2. User Exists Integration is a kernel of object modules that can be compiled into a terminal sales application and which offer the ability to tightly integrate the download function into the daily operations of a store. This includes the functionality to download SAC, Options and Code Update files.

The items configured in Options File include...

- Activate API - Device Address
- Activate MSR emulation – Device Address
- Activate Scanner (BCR) Emulation - Device Address
- Activate Model 3/4 or 4610 Thermal Printer Monitoring
- ScreenKey Character Set
- ODA Character Set (Operator Display Area)
- TDA Character Set (Transaction Display Area)
- KAT (Keyboard Attribute Table)
- Splash Screen
- Command & Barcode Execution times
- Barcode to Stemcode Conversion
- Power-On Beeps
- Key Press Beeps
- CDA display format and User Menu enable
- ScreenSaver setup

4 The MakeDnl Utility

When the application is launched, a Windows dialog screen is presented.



The main dialog, the “File Info”-tab, contains information about the current loaded file such as file name and size, but also the file header information needed by the download utility on the 46xx system and information needed by the ScreenKey keyboard for download purposes. This tab is the same for SAC files, Option files and Code update files because the file header contains the same information. Some of the information on this tab is for trouble shooting purposes. The fields are the following:

- **Input file:** Full path name and size of the input file, either the SAC file created by the SAC Editor, an Options file or a Code Update file.
- **User annotation:** When a file is read the user can make his annotation in the file header. This should contain the specifics of the file. The information is not downloaded to the keyboard and is for inspection purposes only. The user annotation can be viewed by using the “View Download file header” menu command.
- **File description:** What the File description shows depends on if it is a SAC file, an Options file or Code Update file.

For a SAC file it shows the date and time of the conversion, the name of the input file, and it’s date and time of last change. It is presented in the following format:

YY.MM.DD HH:MM:SS, Filename.Ext YY.MM.DD HH:MM:SS

For an Options file it shows the date and time of the last change to the file. It is presented on the following format: YY.MM.DD HH:MM:SS.

The format for Code Update files is the same as for SAC files, but the file's date is the actual date of the file's build (retrieved from the file itself), and the time will be set to zero. (Code Update files can only be inspected with this utility.)

Filename.Ext is the name and extension of the input file, not the full path.

- **Copyright msg.:** This shows an RTI copyright message.
- **File size field:** This field displays the file size found in the file header of the downloadable file. This information is used only by the download utility on the 46xx system.
- **File type:** This is the type of the file currently loaded.
"0x1 – Code-file"
"0x2 – SAC-file"
"0x3 – Option-file"
- **Version / ID:** This field shows the version number or ID of the file currently loaded. It is used by the IBM-46xx system to decide if the file needs to be downloaded or not. For SAC files and Option files this is a unique ID, For code update files, it is the actual version release number of the software.
It is presented as an 8 hexadecimal character preceded by '0x'. The Version/ID is displayed on the ScreenKeys during an actual file download. This information can also be accessed from the onboard resident diagnostics in the keyboard.
- **Checksum:** This checksum is calculated for the whole file, for use by the 46xx system to insure the file is correct and complete before it is downloaded to the keyboard. It is presented as an 8 hexadecimal character preceded by '0x'.
- **No. of. Records:** This is the number of download image records in the file. This is for use by the keyboard only. The 46xx system does not use this field. For the moment only the code update files have more than one image record.
For a code update file this line also contains the "Start address" for the code (used for trouble shooting purposes only).
- **Download file:** This is the full name, path and size of the download file, the file to copy to the 46xx terminal.
For a SAC file this is the converted SAC file (.sdl extension).
For an Option file this may be the same as the input file, if the changes were saved back to the same filename (it can be the same as the input file).

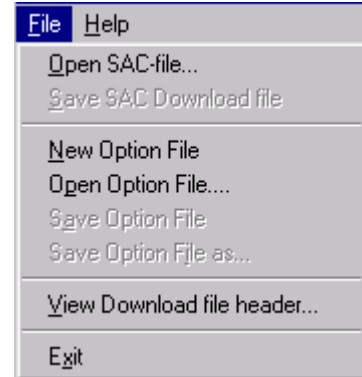
WARNING.

The Download file should always have a maximum of 8 characters in the file name and a maximum of 3 characters in the extension. This is because the IBM 4690OS does not support long file names.

4.1 Menu Selections

The following menu choices are available:

- **File / Open SAC file**
This command opens a specified SAC file for conversion. A standard “Open file” dialog will be presented from which the user can pick an existing SAC file.
- **File / Save SAC Download file**
This command converts and saves a downloadable SAC file. The name will be the same as for the original SAC file, but the file extension will per default be .SDL. An existing SDL file will be backed up and have the same name, but file extension .SDB.
- **File / New Option File**
Creates a default set of Option records, and creates all the tabs needed to define an Option file. The records are not save to a file, this has to be done when editing is finished.
- **File / Open Option File**
This command opens a specified Option file for editing. A standard “Open file” dialog will be presented from which the user can pick an existing Option file.
- **File / Save Option File**
This command saves the Option file back to it original file. Any changes done will be saved back to the file, and any newly excluded records will be lost. An existing ODL file will be backed up and have the same name, but file extension .ODB.
- **File / Save Option File as...**
This command saves the Option file to a new file name. A standard “Save file as” dialog will be presented from which the user can specify a new Option file name. An existing ODL file will be backed up and have the same name, but file extension .ODB.
- **File / View Download file header**
This command is for inspecting the download files headers. It can be useful for inspecting user annotations and for trouble shooting purposes. The command will read downloadable SAC files (SDL), Option files (ODL), and code update files (CDL). Note that the command ONLY reads the header of the files, not the whole files.
- **Help / Help**
Gives a short description of what the program does.
- **Help / About**
Shows the program name, version number and a copyright message.

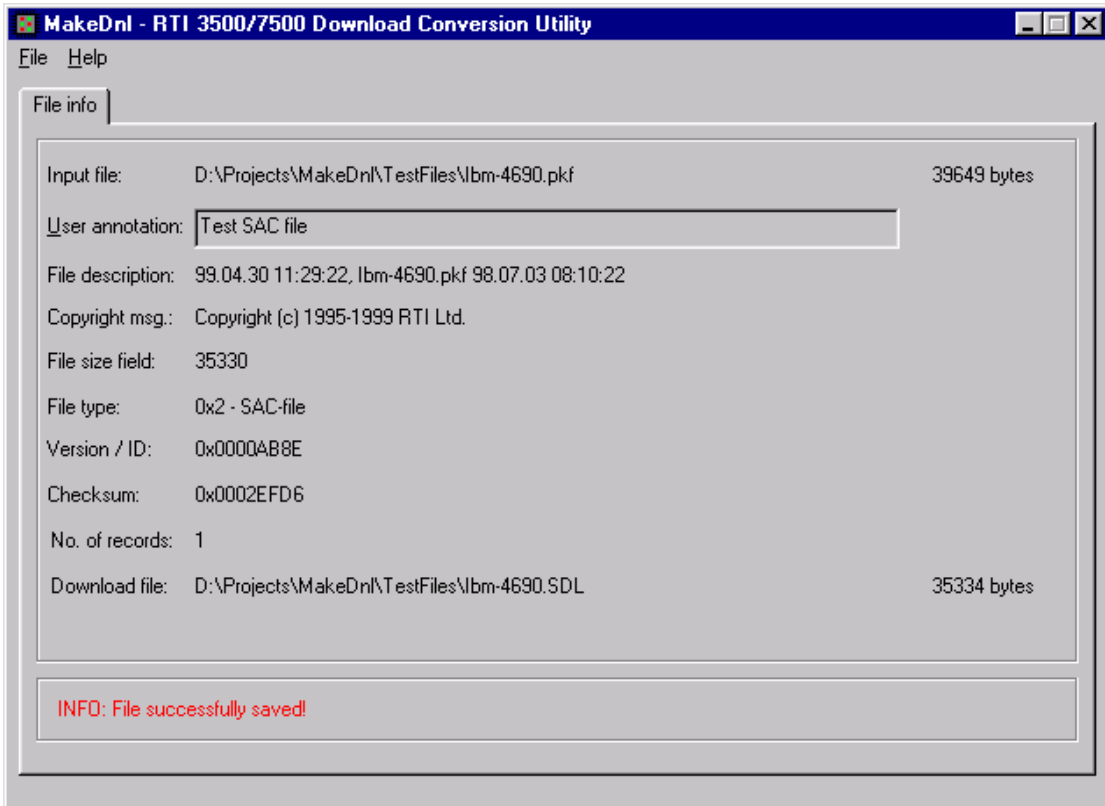


4.2 How to convert a SAC file

A SAC file, created with the SAC Editor, can be converted to a downloadable SAC file using MakeDnl.

1. Open (load) a SAC file using the “Open SAC file” command.
The file will be loaded and the name and size of the file will be displayed in the main dialog.
2. Type your “User annotation” in the edit field. This should describe what is special about the file.
3. Use the command “Save SAC Download file” to perform the conversion and save the file. The file will have the same name as the SAC file, but with the file extension .SDL. This file is the one that has to be copied to the 46xx system.

After a successful SAC file conversion the dialog will look something like this:



4.3 How to create/edit an Options file

A new Options file can be created with the command “New Option File”. A default set of Option records will be generated, and the user can start editing them by selecting the desired tabs. The default data will be the same as the built in factory settings in the SK-7510 keyboard.

An Option file can also be loaded into the utility by using the “Open Option File” command. In this case the file will be read, checksums checked and the contents of the file presented in the tabs.

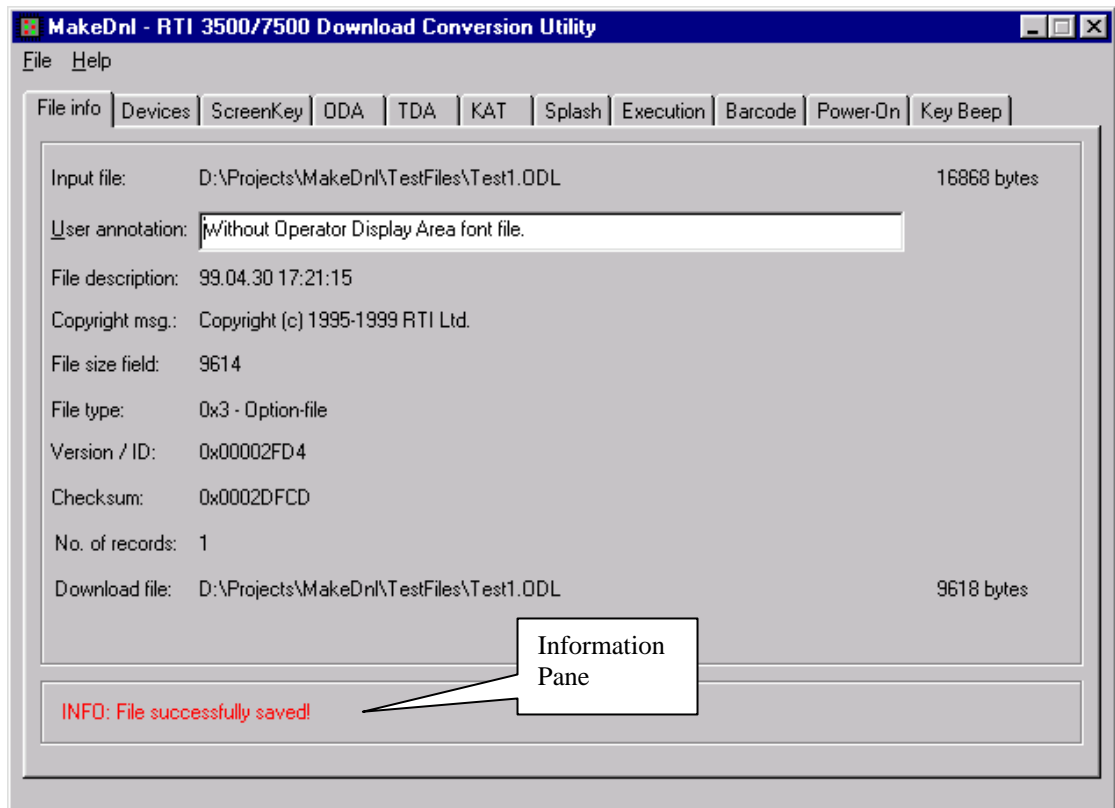
After a file has been created or loaded, a series of tabs will be created and they can be selected and the data can be edited one by one. The different tabs will hold the data for one or more Option records. The tabs (records) are described in detail later in this manual.

When editing is completed the file can either be saved back to the same file, or to a different file name using the “Save Option File” or “Save Option File as” commands.

The application does not check the contents of the tabs for validity until the file is saved. This means you may select any tab at any time, even though the data is incorrect. When saving, the application checks the data for validity, and gives error messages if something is incorrect or missing. Error information is displayed in the *Information Pane*.

The application will give a notice if a changed file is about to be abandoned.

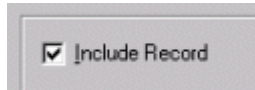
Below is a typical main dialog after an Option file has been saved:



4.4 Include/Exclude Individual Option Record

An Options File is a series of unique records that define the configuration and setup of a ScreenKey Keyboard for use on an IBM 46xx POS terminal.

When editing or creating an Option file using MakeDnl, each 'tab' corresponds to a distinct Options record (except for the *Devices* tab). Note that each tab includes a facility to allow the exclusion of each record using a simple checkbox:



This **must** be checked if the record is to be included in the Options file. If the checkbox is unchecked the data selections will only remain in memory until the program is closed or another file is loaded. When a record is unchecked and the file is saved, the settings previously stored in this record are lost.

Note:

The Devices tab represents a series of device specific records. The include record checkbox on this 'tab' acts on all these device records. However, setting a particular device selection to None causes this record to be ignored by the keyboard.

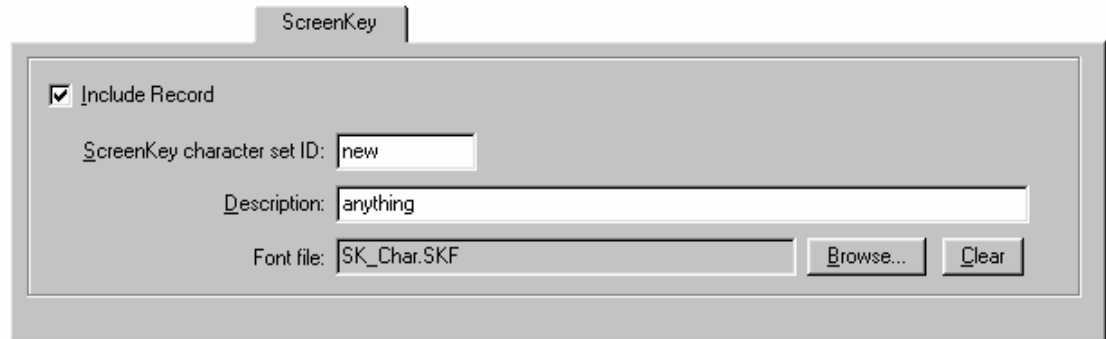
4.5 How to view a download file header

Any downloadable file's header information may be viewed by using the "View Download file header" command. It can be useful to inspect a SAC file (.SDL), e.g. to see the user annotation note, or to inspect the version number of a code update file. This command is primarily implemented for trouble shooting purposes.

5 Options Record Details

5.1 ScreenKey Character Fonts

A ScreenKey character font file may be downloaded to the ScreenKey Keyboard by including it in the Options file using the ScreenKey tab in the MakeDnl utility:



The screenshot shows a dialog box titled "ScreenKey". It has a checked checkbox labeled "Include Record". Below this, there are three text input fields: "ScreenKey character set ID:" containing "new", "Description:" containing "anything", and "Font file:" containing "SK_Char.SKF". To the right of the "Font file:" field are two buttons: "Browse..." and "Clear".

A default file (SK_Char.SKF) is included in the MakeDnl installation. This includes the same contents that are hardcoded in the ScreenKey Keyboard onboard ROM and in code update files.

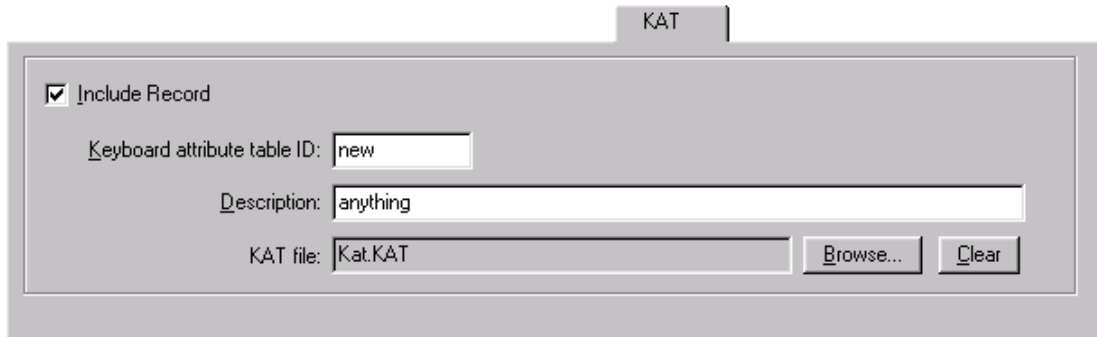
To include a ScreenKey character font file in the Options file, the record has to be included, an ID and a description has to be defined, and a font file has to be specified. This is the ID and description that is displayed on a ScreenKey and on the LCD display respectively during diagnostics. The ID can be up to 10 characters long, two lines of five characters each, and the Description may be up to 64 characters long.

To include a ScreenKey font file, press the Browse button and pick a file with the extension .SKF. To remove a defined ScreenKey font file, press the Clear button. If no ScreenKey font file is wanted, uncheck the 'Include record' box.

5.2 Keyboard Attribute Table (KAT)

The KAT table is a file containing per key attributes, like Key Repeat, Active/Inactive, Key press Beep, Buffer key presses and Send open/close.

The KAT file (Kat.KAT) included with the MakeDnl install is an exact copy of what is hardcoded in the ScreenKey Keyboard, and will therefore not change the behaviour of the keyboard if included.



The image shows a dialog box titled "KAT" with a tab labeled "KAT". Inside the dialog, there is a checked checkbox labeled "Include Record". Below it, there are three text input fields: "Keyboard attribute table ID:" with the value "new", "Description:" with the value "anything", and "KAT file:" with the value "Kat.KAT". To the right of the "KAT file:" field are two buttons: "Browse..." and "Clear".

To include a Keyboard Attribute Table file, define the ID and description and press the Browse button and pick a file with the extension .KAT. To remove a defined KAT file, press the Clear button. If no KAT file is wanted, uncheck the 'Include record' box.

5.3 *Splash Screen file*

The Splash Screen file is a bitmap file that can be displayed on the TDA screen. This file can be customised to contain any picture that can fit on the TDA screen.

The splash file **MUST** be created as a Windows Monochrome Bitmap file.



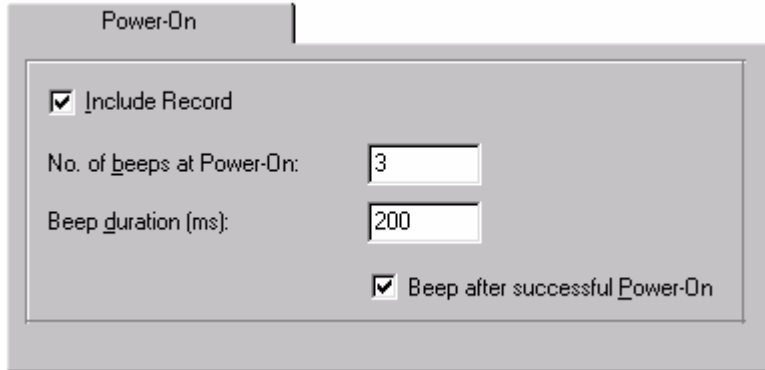
To include a Splash file simply press the Browse button and pick a file with the extension .BMP. To remove a defined Splash file, press the Clear button. If no Splash file is wanted, uncheck the 'Include record' box.

The file can be created with e.g. Microsoft Paint. The bitmap has to be defined as a "Black and White" bitmap, 240 pixels wide and 97 pixels high. Any other configuration will be rejected.

MakeDnl install includes a sample splash bitmap file (Splash.BMP) that displays a "SK-7510" text image.

5.4 Power-On Beeps

The ScreenKey Keyboard can be configured to sound a series of beeps when the power up sequence is completed.



The image shows a dialog box titled "Power-On" with a tab labeled "Power-On". Inside the dialog, there are four settings:

- Include Record
- No. of beeps at Power-On:
- Beep duration (ms):
- Beep after successful Power-On

A certain number of beeps of a given duration can be specified in the Power-On tab. It is also possible to specify if the specified beep sequence is to be used by checking or unchecking the 'Beep after successful Power-On' checkbox. If no Power-On record is wanted, uncheck the 'Include record' box.

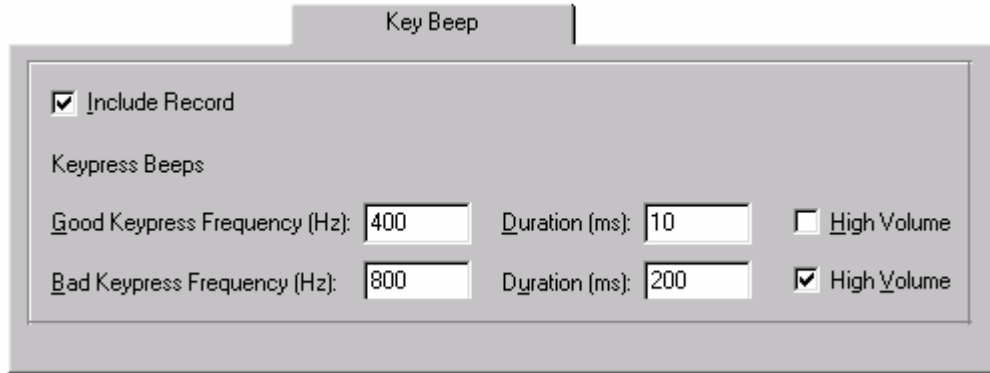
The intention of having a separate checkbox for enabling the Power-On beeps after successful Power-On is that the Option file can contain the setting, but not be in use at the moment. It can then be disabled easily at a later stage.

Note:

This feature is not supported by SK-7510 firmware at the time of writing (version 1.12.00.00).

5.5 Key Beeps

The ScreenKey Keyboard can be configured to beep/click when a key is pressed. It is possible to separate 'Good' key presses from 'Bad' ones, by defining different frequencies, durations and volume levels.



To configure key press beeps select the 'Key Beep' tab. The range for the key press beeps frequency is from 100 Hz to 2000 Hz, and the duration from 1 millisecond to 250 milliseconds. The volume can be either high or low. If no Key press beep record is wanted, uncheck the 'Include record' box.

The keyboard defaults are 2000 Hz at low volume and 10 ms duration for a good key press, and 2000 Hz at low volume and 200 ms duration for a bad key press.

The keyboard interprets keypresses as 'Good' or 'Bad' depending on whether it can act on the keypress or not. For example, if the keyboard is engaged in an end of barcode delay then all keypresses during this delay period are rejected and the 'Bad' keypress beep is sounded. Similarly, if the input keypress buffer becomes full then further keypresses are rejected and the 'Bad' beep is sounded.

5.6 Magnetic Stripe Reader Emulation

5.6.1 Purpose

The IBM 46xx ScreenKey Keyboard assumes a three track reader is physical installed. When a card is swiped the keyboard receives information from 1, 2 or 3 tracks depending on:

- the actual number of tracks the physically installed MSR is capable of reading
- the number of tracks on the card itself

The Terminal Sales Application may enable and disable the MSR. When disabled, card swipes are ignored. When enabled the data is passed to the Terminal. The ScreenKey Keyboard passes only the tracks requested by the terminal. If the keyboard reads three tracks and the terminal requests track 2 only, then the keyboard sends track 2 only and discards the others. On the other hand, if the Terminal requests all three tracks and there is a track 2 only reader installed in the keyboard, then it will respond with track 2 only and null data for the other tracks.

The communication protocol used between the keyboard and the terminal for MSR data is dependant on the type of MSR that the terminal has been told exists in the keyboard (i.e. 1, 2 or 3 track). This is configured in the active Terminal Device Group for that terminal.

The ScreenKey Keyboard must be configured to use the same communication protocol. This is done via the MSR device record in the downloaded Options file.

Note that the POS terminal imposes a restriction on what type of MSR may be used dependent on the type of POS keyboard selected. A three track MSR device may only be chosen if the selected POS keyboard is an IBM N50 emulation. When using IBM 50-Key emulation, only single or double track MSr devices may be chosen.

5.6.2 Options File Setup

To configure the Magnetic Stripe Reader device select the 'Devices' tab. The device address can be selected with the MSR device and Socket dropdown lists. If Dual Track reader is defined, the tracks to read must also be specified.

The screenshot shows a configuration window titled 'Devices'. It contains several fields and dropdown menus for setting up a Magnetic Stripe Reader (MSR) device. The 'Include Record' checkbox is checked. The 'MSR device' dropdown is set to 'Dual track'. The 'Socket' dropdown is set to 'A'. The 'Tracks' dropdown is set to 'Track 1 & 2'. There are also fields for 'Scanner em', 'Printer monit', and 'API device', each with a dropdown menu. A 'Socket' dropdown is also present next to the 'Printer monit' field. A 'TDA lines' field is located at the bottom right. A callout box points to the 'MSR device' dropdown, showing a list of options: 'None', 'Single Track', 'Dual Track', and 'Triple Track'. Another callout box points to the 'Tracks' dropdown, showing a list of options: 'Track 1 & 2' and 'Track 2 & 3'. A third callout box points to the 'TDA lines' field, with the text 'for Printer monitor and/or API device (24-999)'. The 'Printer monit' field has a dropdown menu with options 'A' and 'B'. The 'API device' field has a dropdown menu with options 'A' and 'B'.

The dialog will only allow legal choices. The following combinations are available:

MSR device	Socket	Device address	Tracks
None	N/A	N/A	N/A
Single Track	A	0x40	Fixed to track 2
	B	0x41	
Dual Track	A	0x46	Selectable: track 1 & 2 Or track 2 & 3
	B	0x47	
Triple Track	A	0x48	All tracks
	B	0x49	

5.7 Bar Code Reader Emulation (Buffer Overflow workaround)

5.7.1 Purpose

The ScreenKey Keyboard sends item codes to the terminal as strings of individual digit keystrokes. However, if a number of keys with long item codes assigned to them are pressed quickly then this may trigger a “buffer overflow” in the 46xx terminal. It appears that the keyboard input buffer overflows because the item codes are long and the processing is slow. This is particularly evident on older IBM 468x terminals.

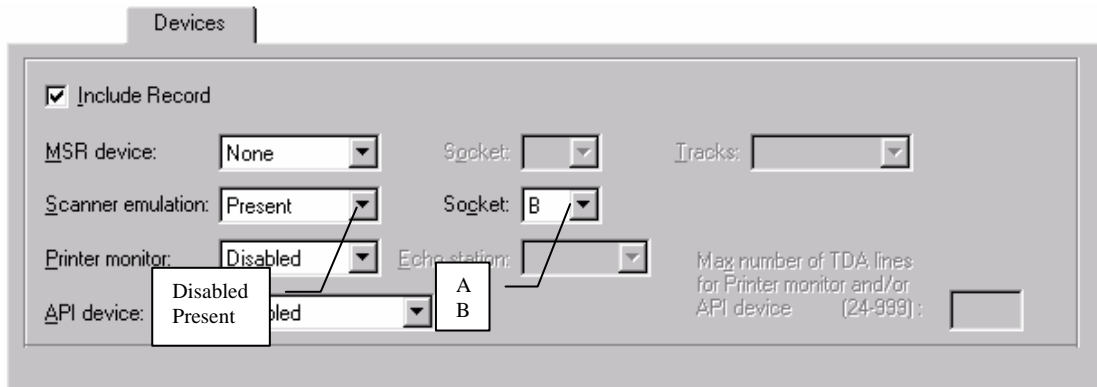
To overcome this, the ScreenKey Keyboard can utilise a bar code reader device channel for sending the item codes to the terminal. This is a much more efficient method and avoids buffer overflow problems.

Note that this feature is only useful for item codes and **not** for sequences incorporating ‘function’ keys (e.g. Sign-On, Cash, etc). As it uses the barcode channel, obviously only data that can be read by an actual barcode unit may be uploaded via this channel.

The use of the bar code channel for item code uploads only works when the SAC file includes such data defined in the “Item Number” field for a key (see SAC Editor User’s Guide).

5.7.2 Options File Setup

To configure the Bar Code Reader Emulation select the ‘Devices’ tab. The device address can be selected with the “Scanner emulation” and socket dropdown lists. Scanner emulation can be either ‘Disabled’ or ‘Present’. If it is present the socket has to be defined, which can be either A or B.



The dialog will only allow legal choices. The following combinations are available:

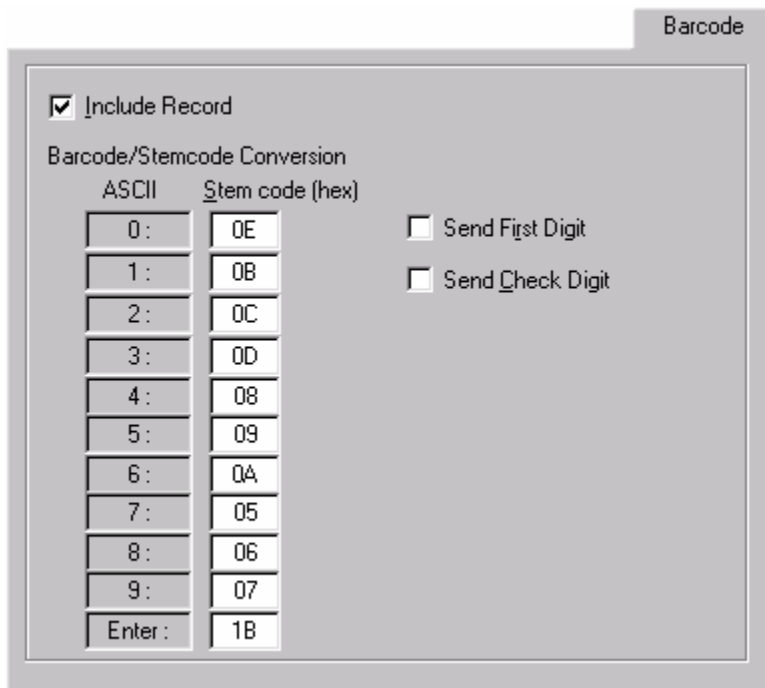
Scanner Emulation	Socket	Device address
Disabled	N/A	N/A
Present	A	0x4A
	B	0x4B

5.8 Barcode/Stemcode Conversion Table

If a bar code reader device channel is not available to the ScreenKey Keyboard and the SAC file references barcode data (in the “Item Number” field), the ScreenKey Keyboard can send the barcodes as ordinary keystrokes via the keyboard channel.

To do this the keyboard needs a Barcode to Stem code conversion table. A barcode is represented (entered) in the editor as a series of ASCII values followed by a Carriage Return (ASCII 0xD), and these have to be replaced by the corresponding Stem codes.

Select the ‘Barcode’ tab to define the conversion table.



ASCII	Stem code (hex)
0 :	0E
1 :	0B
2 :	0C
3 :	0D
4 :	08
5 :	09
6 :	0A
7 :	05
8 :	06
9 :	07
Enter :	1B

In the ‘Barcode’ tab, enter the Stem code for each ASCII value. The corresponding Stem code can be found in the actual keyboard’s documentation, and is NOT dependent on the 46xx terminal setup.

The options to ‘Send First Digit’ and ‘Send Check Digit’ are provided to personalize the ScreenKey Keyboard to suit store specific operations. For example, when an operator encounters an unreadable bar code, he or she will enter the bar code data via the keyboard. Often, this data is entered without the first printed bar code digit or the last printed digit. The last visible bar code digit is referred to as the *check digit*.

These settings **do not** alter bar code handling when a bar code channel is available. They only apply when bar code data is uploaded automatically to the terminal via the keyboard channel.

Note:

The above example matches the IBM 50-Key keyboard.

5.9 Execution Times

When sending barcode data to the terminal as Stem codes, it may be necessary to define Execution Times to slow down the transfer of Stem codes to the terminal. This is because the keyboard can send these much faster than an operator is able to press the numeric keys on the keyboard, which is what the terminal was designed for.

The 'Barcode Interdigit' number is the delay in milliseconds between each digit in a barcode. The 'Barcode transaction' number is the delay in milliseconds the keyboard has to wait for the 46xx terminal to process the sent barcode. Normally the 'Barcode transaction' will be a few seconds, while the 'Barcode Interdigit' will be a few hundred milliseconds. The Execution Time configuration is done in the 'Execution' tab.

Since some terminals will need these delays, and some not, it is possible to enable and disable these with a separate checkbox called 'Enable Execution times' without having to exclude the record and therefore also lose the defined contents.

The 'Execution' tab also contains delay definitions for use with commands other than barcodes. Some commands can be time consuming for the terminal to process. New commands and/or data should **not** be sent while the terminal is working with a previous command.

To help enforce this rule, five categories of execution times are available. These are inserted as 'virtual delays' in the stem code assignment list by the SAC Editor when the SAC file is created. The actual delays used by each keyboard are defined by the *Execution TimesOptions* record.

Execution	
<input checked="" type="checkbox"/> Include Record	
Command Execution Times	Barcode Execution Times
Minimum (ms): <input type="text" value="100"/>	Barcode Interdigit (ms): <input type="text" value="120"/>
Short (ms): <input type="text" value="1000"/>	Barcode transaction (ms): <input type="text" value="2000"/>
Medium (ms): <input type="text" value="2000"/>	
Long (ms): <input type="text" value="5000"/>	<input checked="" type="checkbox"/> Enable Execution times
Maximum (ms): <input type="text" value="10000"/>	

The optimum execution delays will generally be determined on a "trial and error" basis. A 4683 terminal is obviously slower than a 4694 terminal and requires more time to process the stem codes.

The terminal has a limited input buffer size and it seems that it is periodically read by a relatively low priority process and can easily overflow on a 4683 terminal. An example of this is when the printer is working during which time it seems the comm input buffer is not read.

Testing on a 4683 terminal has shown that a good value for the barcode inter-digit delay is 140 milliseconds and 2 seconds for the barcode transaction time. If long sequences of digits (> 6) are associated with menu keys then for slower terminals a delay of 140 milliseconds should be inserted between the digits.

5.10 Operator Display Area Character Font

5.10.1 Purpose

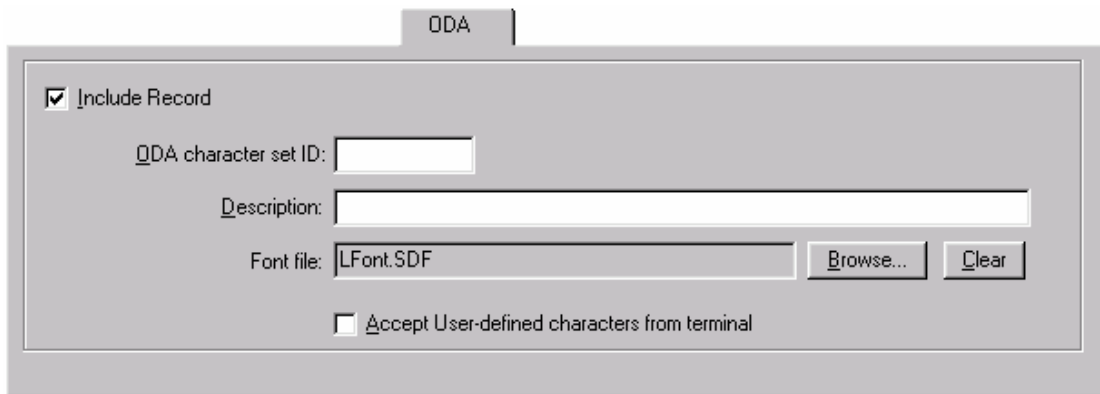
The ODA (Operator/System Display Area) emulates a two-line Vacuum Fluorescent Display Type II or Operator LCD in large characters on the top section of the SK-7510's LCD Panel against a dark background.

The default firmware contains a 64-character ODA font. Full 256-character fonts may be downloaded to backed up RAM via the Options File. Each character is defined on a 12-bit wide by 14-bit high pixel grid (14 words required per character).

The SK-7510 keyboard provides another 'display' device, called the Control Display Area (CDA). This is a pop-up large-font 1x20 display that also uses the ODA character set/font file.

5.10.2 Options File Setup

The Options file can include a replacement 256-character font file for download to the ScreenKey Keyboard to replace the 64 character resident multilingual character set.



The screenshot shows a dialog box titled "ODA" with a tab labeled "ODA". Inside the dialog, there is a checked checkbox labeled "Include Record". Below it are three input fields: "ODA character set ID:" (empty), "Description:" (empty), and "Font file:" (containing "LFont.SDF"). To the right of the "Font file:" field are two buttons: "Browse..." and "Clear". At the bottom of the dialog is an unchecked checkbox labeled "Accept User-defined characters from terminal".

The character set for Code page 850 Multilingual is part of the MakeDnl install (LFont.SDF).

To include an Operator Display Area font file in the Options file, the record has to be included, an ID and a description has to be defined, and a font file has to be specified. The ID and description will be displayed on a ScreenKey and on the LCD display respectively. The ID can be up to 10 characters long, two line of five characters each, and the Description may be up to 64 characters long.

To include an Operator Display Area font file in the record, press the Browse button and pick a file with the extension .SDF. To remove a defined Operator Display Area font file, press the Clear button. If no Operator Display Area font file is wanted, uncheck the 'Include record' box.

The checkbox 'Accept User-defined characters from terminal' specifies if the keyboard should accept characters downloaded from the 46xx terminal. There are 8 User defined characters downloaded from the terminal, characters 0-7. Since the VFD has less resolution than the keyboard's Operator Display Area, these have to be stretched to fit the display. The default Operator Display Area font files contains these characters with a resolution matching

the display. If the user defined characters are not altered on the terminal, the keyboard should be set up not to accept user defined characters because the characters in the font file look better than the stretched downloaded characters.

5.11 Printer Monitoring

5.11.1 Purpose

The SK-7510 can 'listen' to printer data for printer devices attached via the device channel. The SK-7510 supports IBM Model 3/4 and IBM 4610 Thermal printers operating in native modes.

Printer output is displayed in small characters (40 characters / line) on the Transaction Display Area (TDA) of the SK-7510's LCD Panel against a bright background. This is **not** printer emulation. The printer **must** be present and the terminal application must write to it.

This is useful feature for checkout operators as it 'traps' printer data and displays it on an easily reviewed media that includes a simple scroll up and down feature.

With the Model 3/4, the SK-7510 may be configured to 'listen' to either the 'Cash Receipt' station or the 'Summary Journal' station. The 4610 has a 'Cash Receipt' station only.

The Model 3/4 printer supports up to 40 downloadable characters which are stored in the printer. The SK-7510 does not have this level of non-volatile memory to store such information 'semi-permanently'. To overcome this problem, user modified characters should be defined as a new TDA character set and downloaded to the SK-7510 via the Options file. The same principle applies to 4610 printer support.

The SK-7510 ignores logo commands when listening to the cash receipt station. Logo commands are not sent to the journal station.

The printer works as before but with the added feature that either the Journal or Receipt station contents are also echoed on the TDA area of the LCD panel. The 3 buttons below the LCD Panel are used to (from left) Scroll up (Previous), GoTo end (last), Scroll Down (next).

5.11.2 Options File Setup

The Options file must specify which printer and which print station the SK-7510 will listen to, i.e. 4610 or Model 3/4 and summary journal or cash receipt station.

The screenshot shows the 'Devices' configuration window. It includes a 'Close' button in the top right corner. The 'Include Record' checkbox is checked. The 'MSR device' is set to 'None', with 'Socket' and 'Tracks' dropdown menus. 'Scanner emulation' is set to 'Disabled', also with 'Socket' and 'Tracks' dropdown menus. The 'Printer monitor' is set to 'Model 3/4', and the 'Echo station' is set to 'Journal'. The 'API device' is set to 'Disabled'. A text box on the right specifies 'Max number of TDA lines for Printer monitor and/or API device (24-999): 200'. Two callout boxes provide additional context: one for the 'Printer monitor' dropdown menu, which lists 'Disabled', 'Model 3 / 4', and '4610 Thermal'; and another for the 'Echo station' dropdown menu, which lists 'Receipt' and 'Journal'.

Two printers are supported in native mode, Model 3/4 and 4610 Thermal.

If Model 3 / 4 is selected, the station to echo must also be specified, this can be either the Summary Journal or the Cash Receipt station. With the 4610 Thermal printer, only the Cash Receipt station is available.

The maximum number of lines on the TDA also has to be defined. This may be between 24 (which is two full pages) and 999.

The following combinations are available:

Printer Monitor	Echo station	Device address	Max No. of TDA lines
Disabled	N/A	N/A	N/A
Model 3 / 4	Journal	0x34	24-999
	Receipt	0x34	24-999
4610 Thermal	Receipt	0x35	24-999

It should be noted when allocating the number of lines to reserve for usage of the TDA, that each line reserved requires the availability of 40 bytes on RAM from the keyboards onboard memory. Therefore, allocating 999 lines requires the assignment of 39,960 bytes (almost 40Kbytes). This memory is then reserved exclusively for use by the TDA and is not available for other usage (e.g. SAC downloads).

Users should consider carefully the number of lines required and only allocate the minimum number as possible.

5.12 Transaction Display Area Character Font

The Options file can include a replacement 256-character font for download to the ScreenKey Keyboard to replace the firmware resident character set. This facility should also be used to provide downloadable character support in the keyboard.

The screenshot shows a dialog box titled "TDA" with the following elements:

- Include Record
- IDA character set ID:
- Description:
- Font file:

Callout boxes provide additional context:

- Short Name Shown on the ScreenKeys (points to IDA character set ID)
- Long Name Shown on the LCD panel (points to Description)
- The File Name on Disk (points to Font file)

To include a Transaction Display Area (TDA) font file in the Option File, the record has to be included, an ID and a description has to be defined, and a font file has to be specified. The ID and description will be displayed on a ScreenKey and on the LCD display respectively. The ID can be up to 10 characters long, two line of five characters each, and the Description may be up to 64 characters long.

To include a Transaction Display Area font file in the record, press the Browse button and pick a file with the extension .PLF. To remove a defined TDA font file, press the Clear button. If no TDA font file is wanted, uncheck the 'Include record' box.

The MakeDnl install includes a default 256 character set for Code page 850 Multilingual (SFont.PLF).

5.13 API (via Virtual Display Emulation)

5.13.1 Purpose

The keyboard can emulate an Operator Display device in order to facilitate communications between the Terminal Sales Application and the keyboard for the purpose of sending commands to control keyboard operation. This is referred to as the API channel (API – Applications Programming Interface).

The API is used to instruct the keyboard to perform certain actions, e.g. display text on the TDA, show a particular SAC menu on the ScreenKeys, report Manager Key status, display text on the CDA (Control Display Area).

5.13.2 Options File Setup

If API channel is required, an API device has to be specified. It can be 'Operator Display A' or 'Operator Display B', using device address 0x22 and 0x23 respectively. If an API device is configured, the maximum number of lines for the TDA also has to be specified. This may be between 24 (which is two full pages) and 999.

The reason for allocating the number of lines for usage by the TDA when configuring the API is because many of the functions offered by the API relate to management and manipulation of the TDA screen.

The screenshot shows a configuration window titled 'Devices'. It contains several settings:

- Include Record
- MSR device: None (dropdown), Socket: (dropdown), Tracks: (dropdown)
- Scanner emulation: Disabled (dropdown), Socket: (dropdown)
- Printer monitor: Disabled (dropdown), Echo station: (dropdown)
- API device: Operator Display A (dropdown)
- Max number of TDA lines for Printer monitor and/or API device (24-999): 200 (text box)

A callout box points to the 'API device' dropdown, listing 'Operator Display A' and 'Operator Display B'.

The following combinations are available:

API device	Device address	Max No. of TDA lines
Disabled	N/A	N/A
Operator Display A	0x22	24-999
Operator Display B	0x23	24-999

It should be noted when allocating the number of lines to reserve for usage of the TDA, that each line reserved requires the availability of 40 bytes on RAM from the keyboards onboard memory. Therefore, allocating 999 lines requires the assignment of 39,960 bytes (almost 40Kbytes). This memory is then reserved exclusively for use by the TDA and is not available for other usage (e.g. SAC downloads).

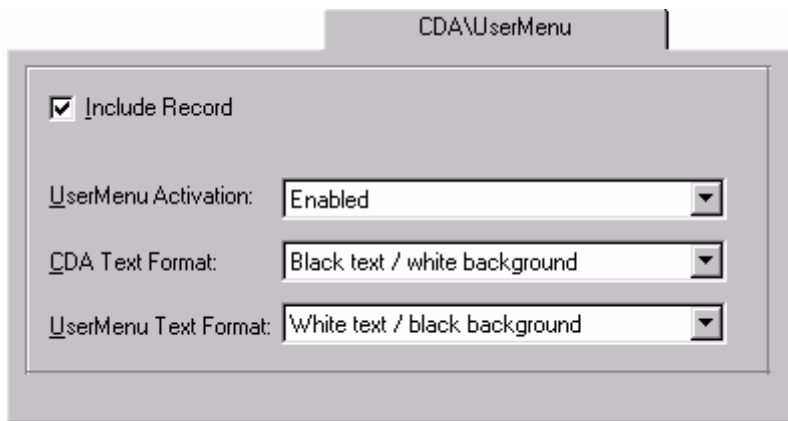
Users should consider carefully the number of lines required and only allocate the minimum number as possible.

5.14 CDA\User Menu

Some ScreenKey Keyboard functions can be accessed from the User Menu. The User Menu uses the CDA (Control Display Area) to present options to the user. The CDA is a pop-up large-font 1x20 display that appears on the bottom of the TDA. The CDA uses the ODA character set/font file.

The CDA is available as a general-purpose display area that may be employed to suit site-specific implementations. For example, the CDA may be used to display running total information during a transaction and then turned off when the transaction completed. Another use for the CDA may be to alert operators of particular information, e.g. watch out for a particular event.

Holding the SELECT button for 5 seconds activates the User Menu. However, access to this function may be disabled for keyboard operators using an Options record (it is allowed by default).



The display format of text on the CDA may be configured differently for both User Menu text and API specified text, as shown in the above example.

5.15 ScreenSaver

The ScreenSaver Options record is provided as a means to protect the lifetime of the bulb backlight used on ScreenKey Keyboards with large-panel LCD displays and as mechanism for displaying a graphic image if the keyboard remains inactive for any long period.

The inactivity timeout may be set from a minimum of 5 minutes to a maximum of 20 minutes. The backlight setting may be set either as NORMAL or OFF. In regular usage, the backlight may be set for to HIGH.

When the ScreenSaver is triggered by keyboard inactivity, it is possible to specify that the splash image (if present in the Options file) will be displayed on the TDA area.

The default values used by the keyboard (when this record is **not** included in the Option file) are:

- 20 minutes timeout
- Brightness NORMAL
- Do NOT show splash

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 SK-7510 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 SK-7510 to alter its mode of operation, i.e. similar to microcode update
EEPROM	Electrically Erasable Read Only Memory, a memory device that retains information indefinitely when power is disconnected
GSA	General Sales Application – sales application running on IBM 46xx sales terminals
LED	Light Emitting Diode, indicator lights on keyboard
MCR	Magnetic Card Reader
ODA	Operator Display Area, upper large text area on the SK-7510 large-panel LCD screen
ODL	Options Download File (created with MakeDnl), configures SK-7510 for specific operations
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 SK-7510 ScreenKeys
SDL	SAC Download file after conversion by MakeDNL utility
Stem Code	Stem codes are the IBM equivalent to scan codes. Stem codes are the actual key values returned to the IBM terminal.
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	11/10/01	M. McDonnell	Initial Release
1.1	06/06/02	M. McDonnell	Correct typo and section cross-reference
1.2	14/03/2006	J. Blomquist	Remove references to discontinued SK-3510

A.4 Change Summary

- 1.1 Correct typo and invalid cross-reference