vComp Pty Ltd
(ABN 39 103 040 311)
PO Box 7356
Cloisters Square
Perth WA 6850
Australia

System Tools Manual

10th November 2015

Overview

System Tools is collection of visual editors that allow the everyday *Surpac* user to leverage tricks usually restricted to the realm of the power user. They provide an easy means to add, edit, and delete system configuration items in *Surpac* that usually have to be done using a text editor like *notepad*.

The various System Tools will ensure correct syntax is maintained in the *Surpac* configuration files. Functions are included to

- Manage logicals definitions
- Define hotkeys (F1 – F12)
- Define command aliases (shortcuts)
- Create *Surpac* range files
- Define portable database definitions

Logical Editor

The logicals editor provides an interface to manage logicals definition's in *Surpac*. Logicals can exist in the standard SSI_ETC:Logicals.ssi file or a user defined logcals file. The editor also provides tools to check for duplicate logical definitions and also to check that the defined directory paths exist.

Hotkey Editor

The hotkey editor provides an easy system to define your keyboard hotkeys F1 through to F12. Hotkeys can be setup to run an internal *Surpac* action or command but can also be setup to run a macro.

Command Alias Editor

The command alias editor allows you to manage up to 10 alias files. A command alias is a shortcut system in *Surpac* that allows the user to define short names (an alias) to a longer command name. For example the shortcut <u>BD</u> maps to the *Surpac* command <u>Bearing and Distance</u>.

Command Alias Integrity Checker

The command alias integrity checker is a tool that performs checks on your defined command aliases. The following three checks can be made

- Check for duplicate alias names
- Check for invalid command names
- Checking for missing macro/script files

Range File Editor

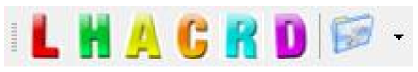
This function allows you to define *Surpac* range files that you can later use in any *Surpac* input field that accepts a range value. Simple specify @myrangefile where myrangefile is name of the range file in the input field

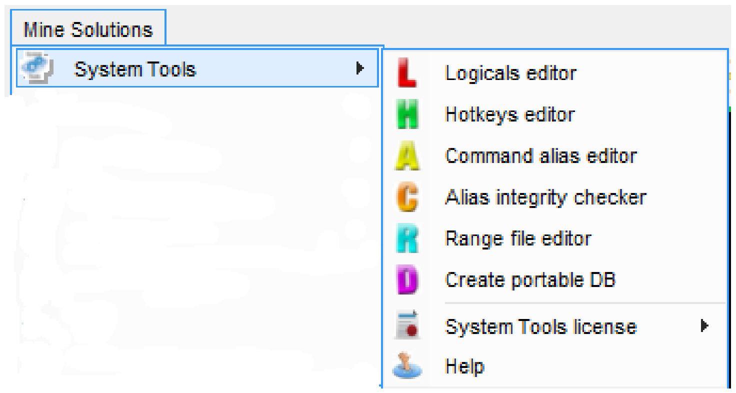
Create Portable Database Definition

In order to access databases from any directory you are required to enter the full directory path into the ddb file as part of the DB_SPECIFIC clause. This function improves on this by inserting a logical definition instead and creating the logical if required.

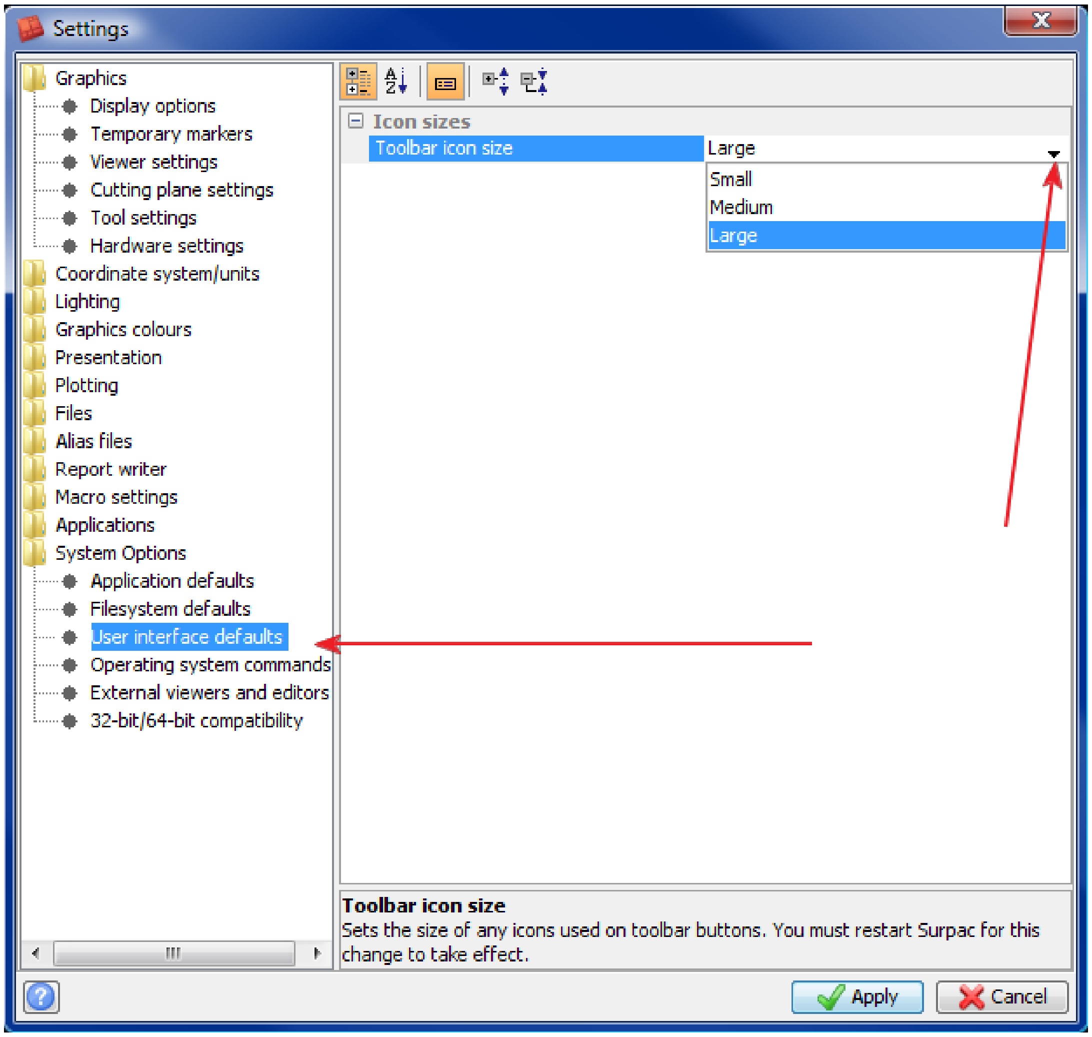
User Interface

All functions are available from the *System Tools* toolbar and menu (found under the Applications menu in *Surpac*)





System Tools supports the 3 different icon sizes that are available in *Surpac V6.3* and later. By default *Surpac* will use medium size icons for the toolbars. You can change this setting in the *Surpac* default preferences editor as shown below using menu option Customise → Default preferences.



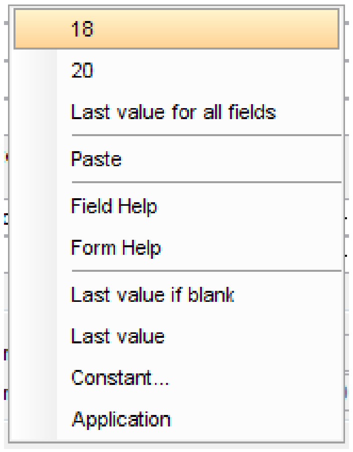
Toolbar icon size: Large

Toolbar icon size
Sets the size of any icons used on toolbar buttons. You must restart Surpac for this change to take effect.

Form Defaults Management

The user interface includes linking into to the *Surpac* defaults management system. This is active on all forms for all *System Tools* functions.

If you right click in the active field on a form you are now presented with the following popup menu.



At the top of this menu you will be presented with the last 5 values that have been previously entered into this form which you can quickly select. (Note the example above only has 2 previous values shown)

You can also select the Last value for all fields option which will set all fields on the form the their last entered values

You also have access to help via the Form Help and Field Help items

The last 4 options on the menu allow you to set the default behaviour for the active field. This will be persistent for the form on all future runs.

- Last value if blank defaults this field to its last value when the input field is blank (ie *System Tools* hasn't put a particular value into this field)
- Last value defaults this field to its last value all the time
- Constant allows you specify a value that will always be the default for this field
- Application has no effect in *System Tools* forms

L Logicals Editor

A logical is a SSI proprietary term that refers to the use of a known name, the logical, that the software maps to some physical directory path. Logicals are used to for a few reasons which are:

- Remove the need to record physical or relative directory paths in the software
- Allows known names to be setup to point to data and macro directories
- Provides a basic system of data centralisation on a network of computers that share data storage
- Provides a mechanism to reduce the length of typing file names by assigning short logical names to a long directory paths
- Allows computer specific paths to be removed from macros to make them generic such that they run on other computers

The Logicals editor allows you to easily create and manage logical definitions with confidence of using the correct syntax and ensuring your directory paths exist. The editor also enforces pseudo standards automatically such as ending each definition with a full colon and ensuring the mandatory slash is present at the end of the path definition. The editor also includes buttons to run checks such as searching for duplicates and checking if defined paths exists. You can also import and export logicals to and from other files.

System, User, & Personal Logicals

Logicals come in three flavors that are defined in three separate files. System logicals are setup by the software installation procedure and are stored into a file known as the translation table, usually *SSI_ETC:translate.ssi*. This file is overwritten each time the software is installed to ensure that the directory paths point to the correct areas for the software to function. This file should not be edited by users.

User logicals are defined in a file called *SSI_ETC:logicals.ssi*. The file is optional and must be created by the user if desired. *System Tools* will create this file automatically when you define logicals through its interface. User logicals are site standard names that are available to each user.

Personal logicals are stored in a file that is specified to the software via a defaults specifier set in the software defaults file *SSI_ETC:defaults.ssi*. A personal logicals file will be created automatically by *System Tools* if desired and the defaults option will be automatically set for you.

It is important to know the order that logicals are loaded into the software. Firstly the system logicals are loaded. Next the user logicals if the *SSI_ETC:logicals.ssi* file exists. Finally the personal logicals file is loaded if one has been set in the system defaults. If you have duplicate logicals defined the last one read by the software is the one that is defined. For example if your redefined SSI_ETC: in your personal logicals file it would override the definition in the system translation map. Duplicate logical names are discussed in full later.

Using Logicals in the Software

To make use of a logical when running the software you simply enter its name as you would a directory name. As an example if you had the following logical definition:

SUPER_PIT: f:/disk1/miningdata/superpit/may2003/

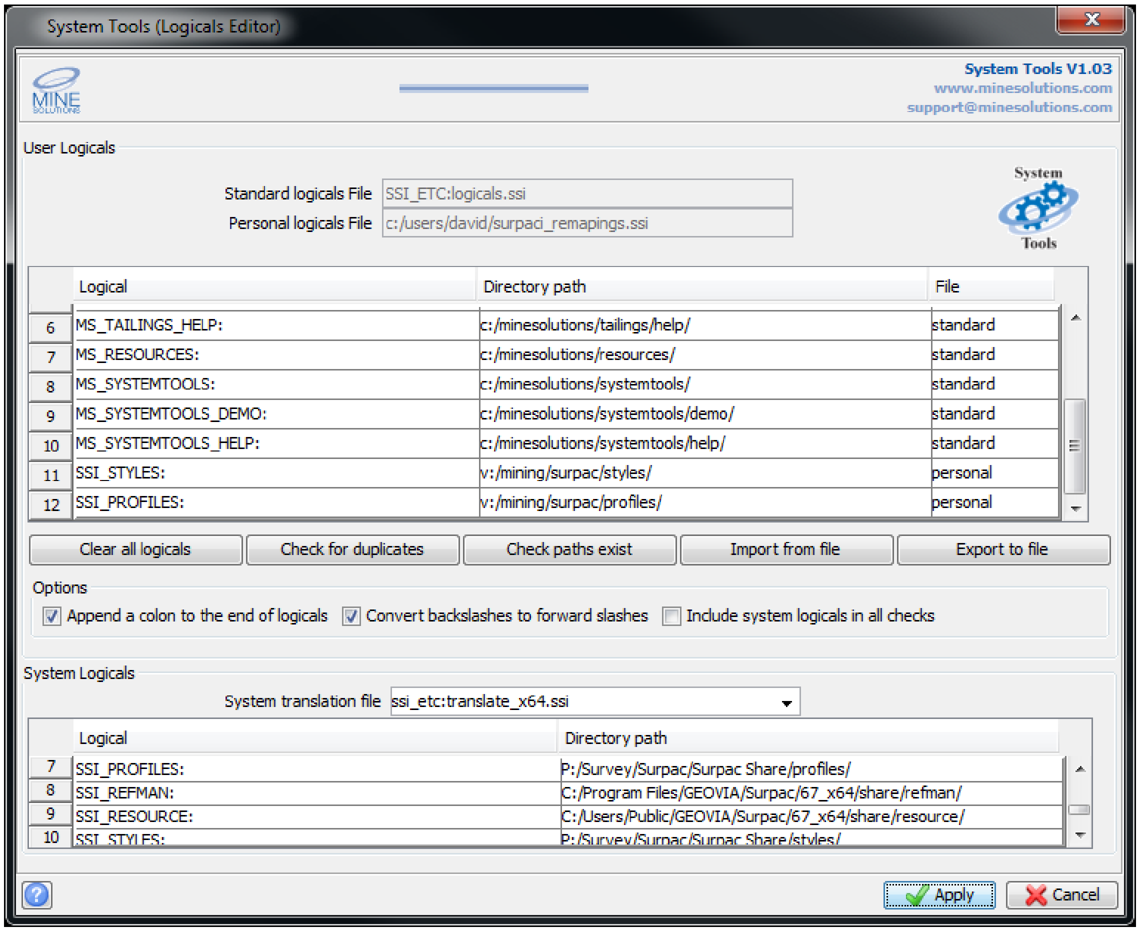
Then to enter a filename into the system you could use either of these two methods:

1. f:/disk1/miningdata/superpit/may2003/pit100.str
2. SUPER_PIT:pit100.str

Obviously method two requires a lot less typing and means that you only have to remember a well know logical name and not a complex directory path. Basically anywhere that you can enter a directory name you can enter a logical name. The software will internally substitute the logical name for its physical directory path.

Duplicate Logical Names

Duplicate logical names are usually an oversight and can sometimes cause undesirable behavior due to fact that you expect that a logical maps to a directory that it doesn't. This is not an error and the software does not report it as one because there can be valid reasons for duplicating logicals. An example might be that you want to map the system logical *SSI_PLOTTING:* to look at a local disk area and not the standard system plotting directory. The Logicals editor provides a duplicate check so that you see what logicals are being overridden.



User Logicals

Standard logicals File: SSI_ETC:logicals.ssi
Personal logicals File: c:/users/david/surpaci_remapings.ssi

	Logical	Directory path	File
6	MS_TAILINGS_HELP:	c:/minesolutions/tailings/help/	standard
7	MS_RESOURCES:	c:/minesolutions/resources/	standard
8	MS_SYSTEMTOOLS:	c:/minesolutions/systemtools/	standard
9	MS_SYSTEMTOOLS_DEMO:	c:/minesolutions/systemtools/demo/	standard
10	MS_SYSTEMTOOLS_HELP:	c:/minesolutions/systemtools/help/	standard
11	SSI_STYLES:	v:/mining/surpac/styles/	personal
12	SSI_PROFILES:	v:/mining/surpac/profiles/	personal

Options
☑ Append a colon to the end of logicals ☑ Convert backslashes to forward slashes ☐ Include system logicals in all checks

System Logicals

System translation file: ssi_etc:translate_x64.ssi

	Logical	Directory path
7	SSI_PROFILES:	P:/Survey/Surpac/Surpac Share/profiles/
8	SSI_REFMAN:	C:/Program Files/GEOVIA/Surpac/67_x64/share/refman/
9	SSI_RESOURCE:	C:/Users/Public/GEOVIA/Surpac/67_x64/share/resource/
10	SSI_STYLES:	P:/Survey/Surpac/Surpac Share/styles/

Standard logicals file

This field is read only. It reports the name of the standard logicals file if it is being used. If blank it signifies that the *SSI_ETC:logicals.ssi* file does not exist.

Personal logicals file

This field is read only. It reports the name of the personal logicals file. If blank it signifies that no personal logicals file has been specified in the system defaults file.

Logical

The logical column in in the user logicals section is where you input your logical name. It is a pseudo standard with logicals to end them with a full colon. Unless you turn this feature off in the options panel a colon is automatically appended if you do not supply one.

To add a new logical definition press the *TAB* key off the last row in the table to create a new blank row. Alternatively you can use the tables popup menu to add or insert a new row by right clicking with the mouse on the tables row headers.

Directory path

The directory path column is were you specify the physical directory path. You can type the path in or use the file browser to select it. It is recommended that forward slashes be used and not back slashes. Unless you turn the feature off in the options panel back slashes are automatically converted to forward slashes.

File

The file column is used to specify which logicals file you want *System Tools* to put the definition into. Your choices are standard for the *SSI_ETC:logicals* file or personal to put them into your own personal logicals file.

If the standard logicals didn't exist when you started the editor it will be created for you. If there was no personal logicals file when the editor started you will be prompted for a file name after applying the form and it will then be created for you and the option to load it set in the system defaults file.

Clear all logicals

This button will clear every logical definition from the user table giving you a clean slate. Use the table pop up menu (right click in the table) to add new rows.

Check for duplicates

The check for duplicates button invokes the duplicate check. Any duplicate names are reported to the message window with the row numbers of the duplicates.

Every logical in the editor is checked even if it has not been saved as yet, so you can check your new definitions before you save them. The system logicals are not included in the duplicate test unless you check the feature to do so in the options panel.

Check paths exist

The check paths button invokes a check of each specified path name to see if it exists. Errors are reported to the message window along with the row number of offending paths. The system logicals are not included in the test unless you check the feature to do so in the options panel.

Logicals that map to non existent physical paths only cause problems if you use them. The fact they exist will not cause errors to be reported if they are not used.

Import from file

The import from file button allows you to locate a file of logical definitions, maybe from an earlier version of the software, and have it loaded into the user table on the form. When you press the button a file browser will appear to allow you to select the file.

Export to file

The Export to file button allows you to save the logicals as defined in the user table to another file. When the button is pressed a file browser will appear that you can either choose an existing file to overwrite or simply enter in the filename in the space provided.

Append a colon to the end of logicals

When this option is checked on, when ever a logical definition is added a colon is automatically appended if one is not supplied. This control when checked can also be doubled clicked with the mouse to perform this function on every logical definition in the user table.

Convert back slashes to forward slashes

When this option is checked on, when ever a path name is added any back slashes are automatically converted to forward slashes. This control when checked can also be doubled clicked with the mouse to perform this function on every path definition in the user table.

Include system logicals in all checks

When this option is checked the system logicals will be included in the duplicate and path checks. The option exists because *System Tools* cannot be sure if it has loaded the correct system translation map. This problem is discussed below.

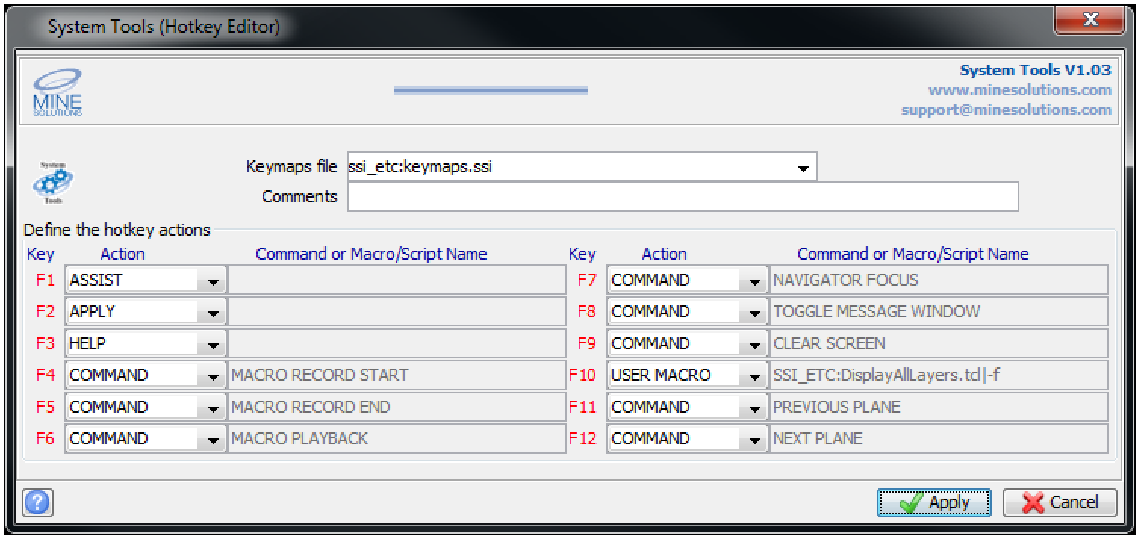
System logicals

The system logicals table is read only as users are not meant to modify this file as it can lead to unexpected problems in the software. The **system**

translation file is user selectable because *System Tools* cannot be sure that it has loaded the correct translation map. The problem is that there is no way of determining the map that the software loaded so an educated guess of *SSI_ETC:translate.ssi is made.*

Hotkey Editor

The hotkey editor provides an easy system to define your keyboard hotkeys F1 through to F12. Hotkeys can be setup to run an internal *Surpac* action or command but can also be setup to run a macro.



Keymaps file: ssi_etc:keymaps.ssi
Comments:

Define the hotkey actions

Key	Action	Command or Macro/Script Name	Key	Action	Command or Macro/Script Name
F1	ASSIST		F7	COMMAND	NAVIGATOR FOCUS
F2	APPLY		F8	COMMAND	TOGGLE MESSAGE WINDOW
F3	HELP		F9	COMMAND	CLEAR SCREEN
F4	COMMAND	MACRO RECORD START	F10	USER MACRO	SSI_ETC:DisplayAllLayers.tcl|-f
F5	COMMAND	MACRO RECORD END	F11	COMMAND	PREVIOUS PLANE
F6	COMMAND	MACRO PLAYBACK	F12	COMMAND	NEXT PLANE

Keymaps filename

Specify the name that you want to store your keymaps definitions to. You can use the file browser to select a previously defined keymaps file to import into the editor. You may for instance wish to maintain a number of keymaps files that access different commands that are relevant to unrelated areas of the software.

Comments

Enter any optional comments that describe the purpose of the keymaps file.

Key

Read only field that describes the function key number.

Action

Specify the action that you wish the key to perform. Possibilities are

ASSIST	This is an editing action. While you are interacting in a field on a any form pressing the assist key will bring up any attached lists or browsers such as file and database browsers or the drop down list of values on a combobox
APPLY	This is an editing action. The apply key will apply any active form. This is largely historical as the keyboard Enter key is usually used to apply forms
HELP	This is an editing action. The help key invokes the online manual.
INSERT	This is an editing action. The insert key places the key entry buffer into insert mode as opposed to overtype mode.
OVERTYPE	This is an editing action. The overtype key places the key entry buffer into overtype mode as opposed to insert mode.
CLEAR	This is an editing action. The clear key will clear the contents of a field.
COMMAND	This attaches the key to an internal software command. When focus moves off the action field a pop up form will appear from which you can choose any valid command.
USER MACRO	This attaches a user Tcl script (macro) to the key. When focus moves off the action field a file browser pop will appear. Either select a macro file using the browser or type the name of the macro file in the space provided.

Command or macro/script file

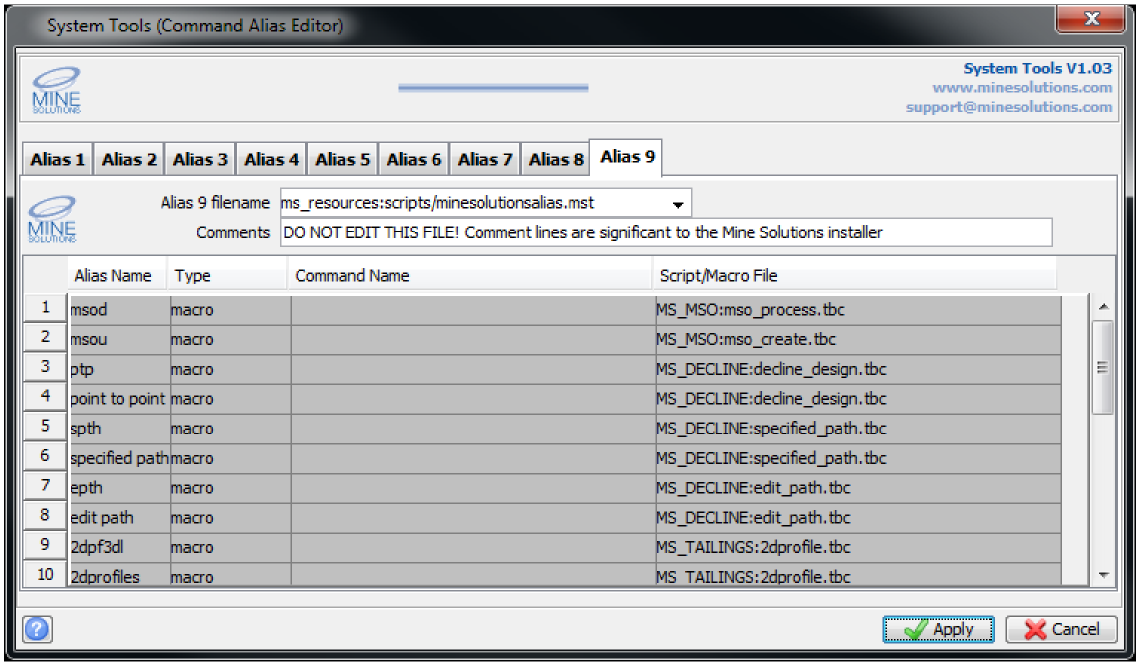
Read only field that lists the software command or macro script that is attached to the key

Command Alias Editor

A Command alias is any name that you can type into the software's command chooser that invokes an internal command or external macro script. Command aliases are defined in files that are specified to the software in the system defaults file. Up to nine different alias files can be specified.

Two alias files are shipped with the software defined as alias1 and alias2. These define the short cut command names and old *Surpac* 1 option mappings to commands. These files are shipped with a '.mst' extension which makes them read only in the editor. Any file with such an extension is read only. This allows users to distribute alias files that can be protected from other people editing them.

Users may want to define their own alias names for two reasons. Firstly names that are more meaningful to user can be selected. Secondly recorded macro scripts can be assigned a command alias name to make them easy to access without having to navigate to a directory to find the macro.



Alias 9 filename ms_resources:scripts/minesolutionsalias.mst
Comments DO NOT EDIT THIS FILE! Comment lines are significant to the Mine Solutions installer

	Alias Name	Type	Command Name	Script/Macro File
1	msod	macro		MS_MSO:mso_process.tbc
2	msou	macro		MS_MSO:mso_create.tbc
3	ptp	macro		MS_DECLINE:decline_design.tbc
4	point to point	macro		MS_DECLINE:decline_design.tbc
5	spth	macro		MS_DECLINE:specified_path.tbc
6	specified path	macro		MS_DECLINE:specified_path.tbc
7	epth	macro		MS_DECLINE:edit_path.tbc
8	edit path	macro		MS_DECLINE:edit_path.tbc
9	2dpf3dl	macro		MS_TAILINGS:2dprofile.tbc
10	2dprofiles	macro		MS_TAILINGS:2dprofile.tbc

The editor contains nine separate tabs for the nine possible alias files. When you create a new alias definition the file is created for you and automatically specified to the software in the system defaults file. The order of the alias numbers is only important in that if duplicate alias names exist the last one read by the software is the one used. The software loads alias files beginning from one though to nine. It does not matter if you miss an alias number and define a file at a higher level.

Alias # filename

Specify the name that you want to store your alias definitions to. You can use the file browser to select a previously defined alias file to import into the editor. Making this field blank will clear the alias tab.

Comments

Enter any optional comments that describe the purpose of the alias file.

Alias name

Enter any name that you want to define as a command alias. Alias names may contain spaces if you wish.

To add a new alias definition press the *TAB* key off the last row in the table to create a new blank row. Alternatively you can use the tables popup menu to add or insert a new row by right clicking with the mouse on the tables row headers.

Type

Alias type can be either *command* or *macro*. A command will allow you to select any internal software command to attach to your alias. The *macro* option will allow you to attach an external cmd or tcl macro to the alias name.

Command name

If the alias type is *command* then this field becomes active and allows you to select from the list a valid software commands.

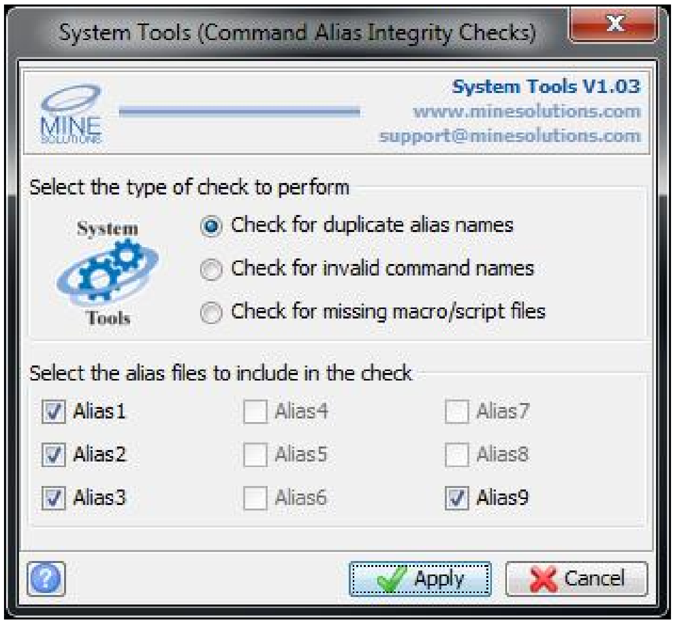
Script/Macro file

If the alias type is 'macro' then this field is active. You can use the file browser to select your macro file or simply type the name. You are encouraged to make use of logicals for directory paths where ever possible.

Command Alias Integrity Checks

This function allows you to perform some integrity checks on your alias definitions. You can determine if a logical is being over written by another. You can ensure that alias definitions only access valid software commands. You can also determine if your alias definitions are trying to access non existent macro files.

The integrity checks are used to pin point problems that you may be experiencing, such as an alias definition that does something other than what you intended. Maybe an alias doesn't appear to do anything because a macro or script path is wrong or has changed.


System Tools (Command Alias Integrity Checks)
System Tools V1.03
www.minesolutions.com
support@minesolutions.com
Apply
Cancel

Select the type of check to perform
[● Check for duplicate alias names]
○ Check for invalid command names
○ Check for missing macro/script files

Select the alias files to include in the check

☑ Alias1	☐ Alias4	☐ Alias7
☑ Alias2	☐ Alias5	☐ Alias8
☑ Alias3	☐ Alias6	☑ Alias9

Select the type of check to perform

Select one of the three alias integrity checks as described above.

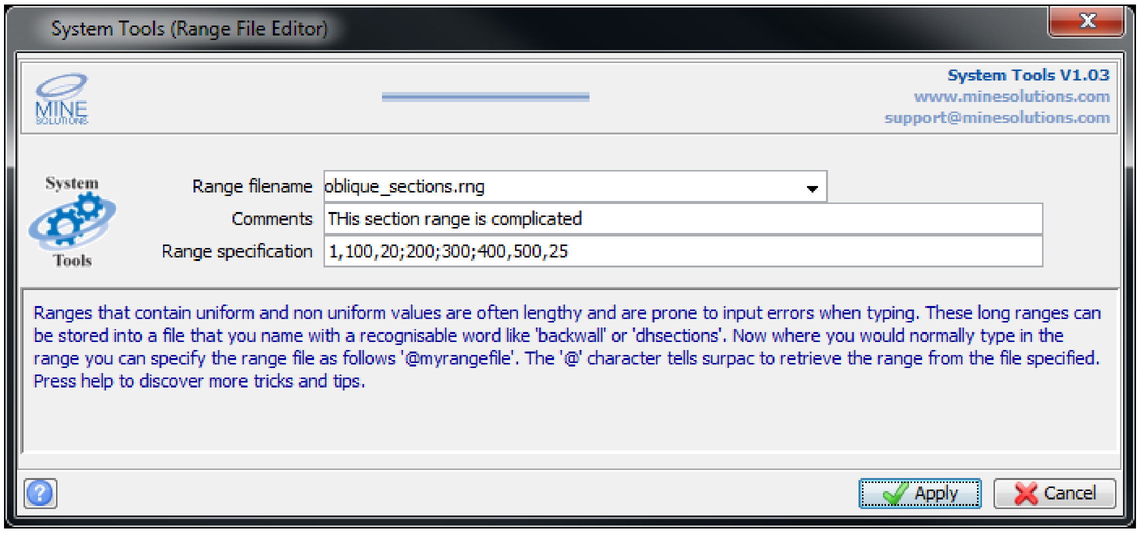
Select the alias files to include in the check

Select the alias files that you wish to include in your integrity checks. Only alias files that are loaded are selectable.

Range File Editor

Ranges in the software can be uniform, non-uniform, or a combination of both. Examples of uniform ranges are 1,200,10, of non-uniform ranges are 100;200;500;1000, and a combination range is 1,100,20;200;400;500,550.

Ranges that contain uniform and non-uniform values are often lengthy and are prone to input errors when typing. These long ranges can be stored into a file that you name with a recognisable word like 'backwall' or 'dhsections'. Now where you would normally type in the range in *Surpac* you can specify the range file as follows '@myrangefile'. The '@' character tells the software to retrieve the range from the file specified. If you had a logical defined to the directory where you store your range file you could enter it using the logical like this example '@RNG:dhsection'.



Range filename: oblique_sections.rng
Comments: THis section range is complicated
Range specification: 1,100,20;200;300;400,500,25

Ranges that contain uniform and non uniform values are often lengthy and are prone to input errors when typing. These long ranges can be stored into a file that you name with a recognisable word like 'backwall' or 'dhsections'. Now where you would normally type in the range you can specify the range file as follows '@myrangefile'. The '@' character tells surpac to retrieve the range from the file specified. Press help to discover more tricks and tips.

Range filename

Specify the name of the range file that you wish to create. If you want to edit an existing range file enter its name or use the file browser to select it. After moving input focus from this field the range details will be retrieved.

Comments

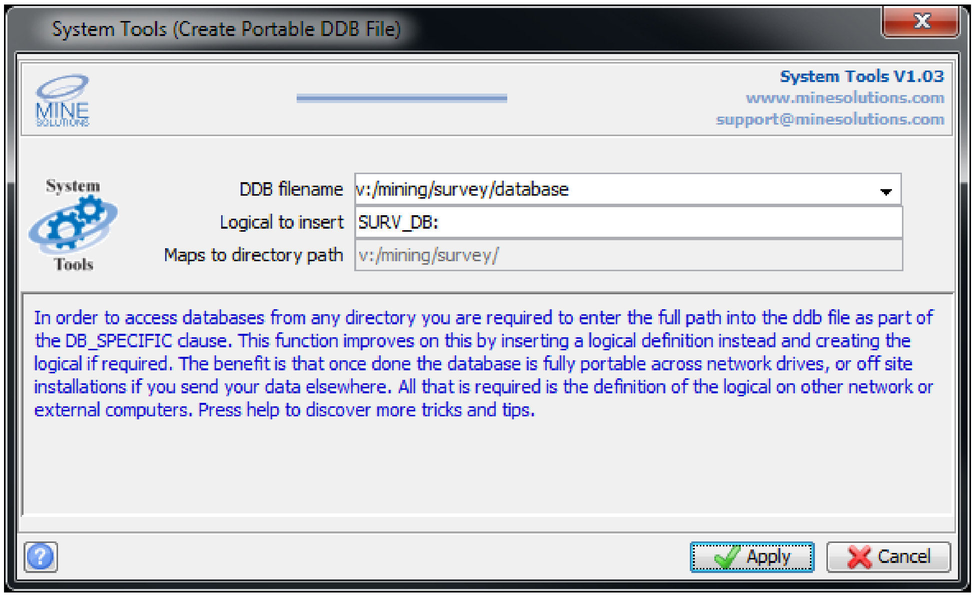
Enter any optional comments that describe the purpose of the range file.

Range specification

Enter the range that you wish to store into the file for later use in the software.

D Create portable database

In order to access databases from any directory you are required to enter the full path into the ddb file as part of the DB_SPECIFIC clause. This function improves on this by inserting a logical definition instead and creating the logical if required. The benefit is that once done the database is fully portable across network drives, or off site installations if you send your data elsewhere. All that is required is the definition of the logical on other network or external computers.


System Tools (Create Portable DDB File)
System Tools V1.03
www.minesolutions.com
support@minesolutions.com
Apply
Cancel

DDB filename: v:/mining/survey/database
Logical to insert: SURV_DB:
Maps to directory path: v:/mining/survey/

In order to access databases from any directory you are required to enter the full path into the ddb file as part of the DB_SPECIFIC clause. This function improves on this by inserting a logical definition instead and creating the logical if required. The benefit is that once done the database is fully portable across network drives, or off site installations if you send your data elsewhere. All that is required is the definition of the logical on other network or external computers. Press help to discover more tricks and tips.

DDB filename

Specify the name of the ddb file that you wish to insert a logical into by typing its name or using the file browser to select it. After moving input focus from this field the function will endeavor to find a logical that maps to the directory that the database is in.

Logical to insert

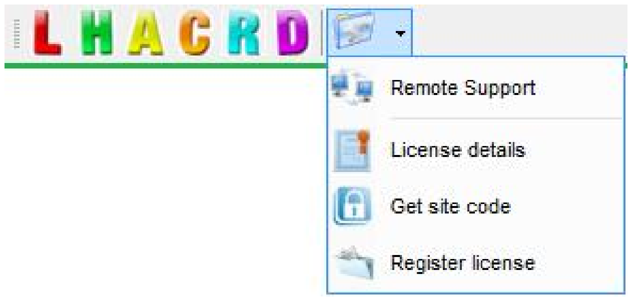
If the system is able to find a logical that maps to the databases directory path it will be displayed. If no logical was then you will need to enter one here. The logical will be automatically inserted into the user logicals file.

Maps to directory path

Read only field that displays the directory path that the logical will or does map to.

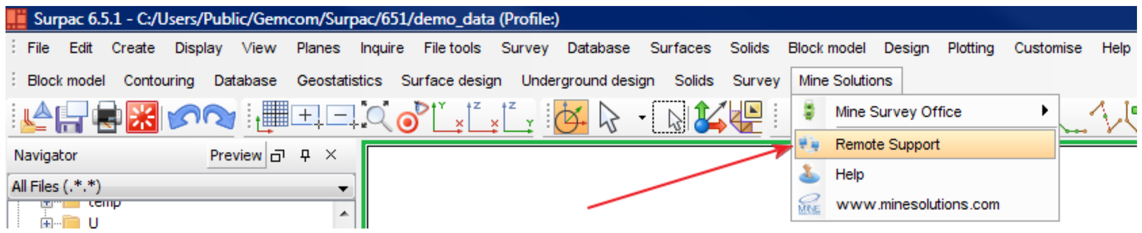
System Tools Administration Options

The License and Utilities menu provides access to *System Tools* administration menu.



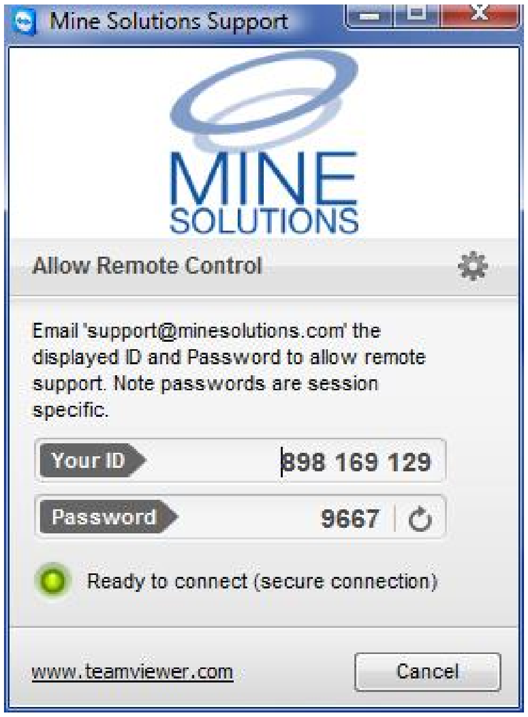
Remote Support Option

You can access the remote support option from the *Mine Solutions* submenu (found on the *Surpac* Applications menu);


Surpac 6.5.1 - C:/Users/Public/Gemcom/Surpac/651/demo_data (Profile:)
File Edit Create Display View Planes Inquire File tools Survey Database Surfaces Solids Block model Design Plotting Customise Help
Block model Contouring Database Geostatistics Surface design Underground design Solids Survey Mine Solutions
Navigator Preview
All Files (.*.*)
Mine Survey Office
Remote Support
Help
www.minesolutions.com

or the *System Tools* toolbar;

Selecting this option will permit *Mine Solutions* support to remotely log onto your computer to assist with problem detection and resolution. As this is an internet link you will need to give the program permission to run. Once you have done this the remote client will display as follows


Mine Solutions Support

Allow Remote Control

Email 'support@minesolutions.com' the displayed ID and Password to allow remote support. Note passwords are session specific.

Your ID 898 169 129

Password 9667

Ready to connect (secure connection)

www.teamviewer.com

All that is required from here is for you to email the 9 digit ID number and 4 digit Password to support@minesolutions.com. Please note the password is session specific and that the remote client must be running to permit access to your computer.

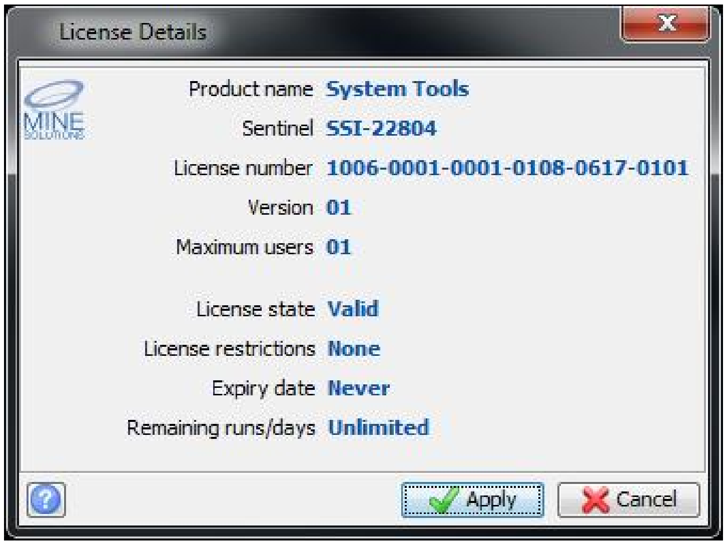
System Tools License System

System Tools uses the standard *Mine Solutions* licensing system which works on both standalone *Surpac* licenses and also the *Surpac License Manager (*SSIlm). When using a *Surpac* network license the *System Tools* token maybe registered on as many computers as required but only the licensed number of users can run at any one instance. For example if it is a 1 user license of *System Tools* then only one user can run (be active) at any given moment, even if the license is registered on 3 computers.

System Tools is offered free of charge but a registration code is still required.

View License Details

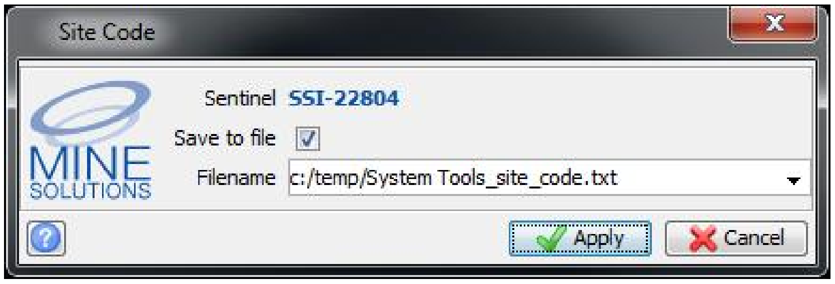
The view license details option displays the current status of your *System Tools* License.



Product name	System Tools
Sentinel	SSI-22804
License number	1006-0001-0001-0108-0617-0101
Version	01
Maximum users	01
License state	Valid
License restrictions	None
Expiry date	Never
Remaining runs/days	Unlimited

Get Site Code

System Tools is licensed to a particular site code which is embedded in the *Surpac* sentinel. This option will retrieve that code. The code can be written to a file that can be emailed to *Mine Solutions* for generation of a license token.

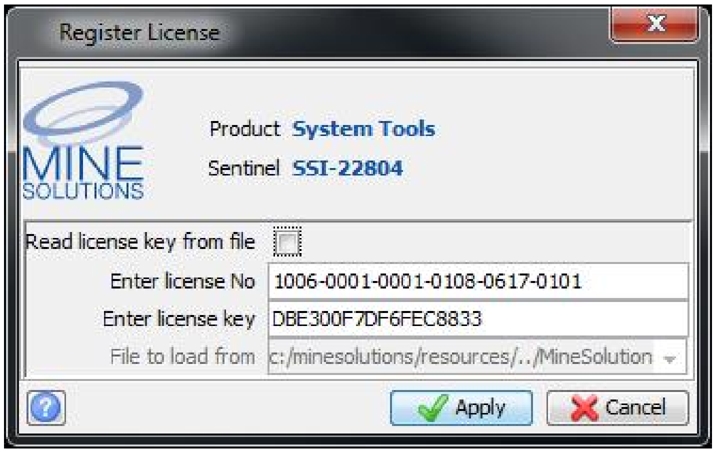


Sentinel	SSI-22804
Save to file	✓
Filename	c:/temp/System Tools_site_code.txt

Register License

This option allows you to register the *System Tools* license token. Normally tokens are distributed in a *Mine Solutions* token file (.tok). If the file is copied into the base *Mine Solutions* directory then the license information will automatically be searched, and the form should be pre-filled.

Note that you need computer administration privileges to register a license. Even if your normal login allows administrator rights you will need to start *Surpac* by right clicking the desktop icon and elect to "Run as administrator"


Register License

Product **System Tools**
Sentinel **SSI-22804**

Read license key from file	☐
Enter license No	1006-0001-0001-0108-0617-0101
Enter license key	DBE300F7DF6FEC8833
File to load from	c:/minesolutions/resources/../MineSolution