

ARI

Copyright © 1993 Soft-Logik Publishing Corporation

COLLABORATORS

	<i>TITLE :</i> ARI		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		March 29, 2025	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	ARI	1
1.1	TYPESMITH2® ALPHABETICAL AREXX INDEX	1
1.2	TYPESMITH2® SUBJECT AREXX INDEX	3
1.3	version	6
1.4	to_front	6
1.5	to_back	6
1.6	set_scale	7
1.7	show_filled	7
1.8	show_template	7
1.9	size_window	7
1.10	goto_char	8
1.11	quit	8
1.12	open_dmf	8
1.13	save_dmf	8
1.14	open_fm	9
1.15	save_fm	9
1.16	import_pfb	9
1.17	export_pfb	9
1.18	import_afm	10
1.19	export_afm	10
1.20	import_pfm	10
1.21	import_if	10
1.22	export_if	11
1.23	export_ps	11
1.24	import_ttf	11
1.25	export_ttf	12
1.26	import_rff	12
1.27	export_rff	12
1.28	import_template	12
1.29	import_dr2d	13

1.30	export_dr2d	13
1.31	new_font	13
1.32	new_metric	14
1.33	next_font	14
1.34	new_view	14
1.35	next_view	14
1.36	get_view	15
1.37	close_font	15
1.38	close_view	15
1.39	move_char	15
1.40	reduce_points	16
1.41	correct_pathdirs	16
1.42	moveto	16
1.43	lineto	16
1.44	curveto	17
1.45	closepath	17
1.46	select_all	17
1.47	unselect_all	18
1.48	first_path	18
1.49	select_path	18
1.50	next_path	18
1.51	previous_path	19
1.52	goto_start	19
1.53	next_point	19
1.54	previous_point	20
1.55	select_area	20
1.56	get_numpaths	20
1.57	get_numpoints	21
1.58	get_pointx	21
1.59	get_pointy	21
1.60	get_prev_bcpX	22
1.61	get_prev_bcpY	22
1.62	get_next_bcpX	22
1.63	get_next_bcpY	23
1.64	get_pathdirection	23
1.65	get_previous_type	23
1.66	get_char_num	24
1.67	get_version	24
1.68	get_width	24

1.69	set_composition	25
1.70	set_point	25
1.71	set_pathdirection	25
1.72	set_fontname	26
1.73	set_family	26
1.74	set_weight	26
1.75	set_fullname	27
1.76	set_fontid	27
1.77	set_notice	27
1.78	set_version	27
1.79	set_width	28
1.80	rotate	28
1.81	skew	28
1.82	scale	29
1.83	flip	29
1.84	move	29
1.85	copy	30
1.86	cut	30
1.87	paste	30
1.88	delete	31
1.89	beepscreen	31
1.90	display_alert	31
1.91	get_string	31
1.92	get_file	32
1.93	help	32
1.94	print_character	32
1.95	print_overview	33
1.96	print_string	33
1.97	t_autotrace	33
1.98	t_edit	34
1.99	t_generate_char	34
1.100	t_get_point	34
1.101	t_get_pos	35
1.102	t_get_charwidth	35
1.103	t_new_bitmapfont	35
1.104	t_open_abf	36
1.105	t_open_amigafont	36
1.106	t_open_iffilbm	36
1.107	t_open_slfont	36

1.108t_clear	37
1.109t_save_abf	37
1.110t_save_amigafont	37
1.111t_save_iffilbm	37
1.112t_save_slfont	38
1.113t_set_charwidth	38
1.114t_set_point	38
1.115t_set_pos	38
1.116t_get_dimensions	39
1.117t_set_dimensions	39
1.118t_shift	39
1.119t_show	40

Chapter 1

ARI

1.1 TYPESMITH2® ALPHABETICAL AREXX INDEX

B beepscreen

C close_font
close_view
closepath
copy
correct_pathdirs
curveto
cut

D delete
display_alert

E export_afm
export_dr2d
export_if
export_pfb
export_ps
export_rff
export_ttf

F first_path
flip

G get_char_num
get_file
get_numpaths
get_numpoints
get_pointx
get_pointy
get_prev_bcpX
get_prev_bcpY
get_next_bcpX
get_next_bcpY
get_pathdirection
get_previous_type
get_string
get_version

get_view
get_width
goto_char
goto_start

H help

I import_afm
import_dr2d
import_if
import_pfb
import_pfm
import_rff
import_ttf
import_template

L lineto

M move
move_char
moveto

N new_font
new_metric
new_view
next_font
next_view
next_path
next_point

O open_dmf
open_fm

P paste
previous_path
previous_point
print_character
print_overview
print_string

Q quit

R reduce_points
rotate

S save_dmf
save_fm
scale
select_all
select_area
select_path
set_composition
set_family
set_fontid
set_fontname
set_fullname
set_notice

set_pathdirection
set_point
set_scale
set_version
set_weight
set_width
show_filled
show_template
size_window
skew

T to_back
t_autotrace
t_clear
t_edit
t_generate_char
t_get_charwidth
t_get_dimensions
t_get_point
t_get_pos
t_new_bitmapfont
t_open_abf
t_open_amigafont
t_open_iffilbm
t_open_slfont
t_save_abf
t_save_amigafont
t_save_iffilbm
t_save_slfont
t_set_charwidth
t_set_dimensions
t_set_point
t_set_pos
t_shift
to_front

U unselect_all

V version

1.2 TYPESMITH2® SUBJECT AREXX INDEX

Screen and Window Commands

close_font
close_view
next_font
new_view
next_view
get_view
set_scale
show_filled
show_template
size_window
goto_char

to_back
to_front
version
quit

File Commands

open_dmf
save_dmf
open_fm
save_fm
import_pfb
export_pfb
import_afm
export_afm
import_pfm
export_ps
import_if
export_if
import_rff
export_rff
import_ttf
export_ttf
import_template
import_dr2d
export_dr2d
new_font
new_metric

Drawing Commands

moveto
lineto
curveto
closepath

Character Commands

correct_pathdirs
get_char_num
get_numpaths
get_numpoints
get_version
get_width
move_char
reduce_points
set_composition
set_pathdirection
set_width

Point and Path Selection Commands

select_all
unselect_all
first_path
select_path
next_path

```
previous_path
goto_start
next_point
previous_point
select_area
```

Point and Path Commands

```
get_pointx
get_pointy
get_prev_bcp_x
get_prev_bcp_y
get_next_bcp_x
get_next_bcp_y
get_pathdirection
get_previous_type
set_point
```

Set Font Name Commands

```
set_fontname
set_family
set_weight
set_fullname
set_fontid
set_notice
set_version
```

Set Point and Path Commands

```
rotate
skew
scale
flip
move
copy
cut
paste
delete
```

Template Commands

```
t_autotrace
t_clear
t_edit
t_generate_char
t_get_charwidth
t_get_dimensions
t_get_point
t_get_pos
t_new_bitmapfont
t_open_abf
t_open_amigafont
t_open_iffilbm
t_open_slfont
t_save_abf
t_save_amigafont
```

```
t_save_iffilbm
t_save_slfont
t_set_charwidth
t_set_dimensions
t_set_point
t_set_pos
t_shift
```

Feedback and Requester Commands

```
beepscreen
display_alert
get_string
get_file
help
print_character
print_overview
print_string
```

1.3 version

Command: VERSION

Purpose: Gets the version number of TypeSmith and stores it in the RESULT variable.

Format: version (no parameters)

Note: The result must be divided by 100 to get the actual version number.

1.4 to_front

Command: TO_FRONT

Purpose: Bring the TypeSmith screen to the front.

Format: to_front (no parameters)

1.5 to_back

Command: TO_BACK

Purpose: Send the TypeSmith screen to the back.

Format: to_back (no parameters)

1.6 set_scale

Command: SET_SCALE

Purpose: Sets the view scale of the active character window.

Format: set_scale <x>

Where: x is the 1:x scale factor (integer).

Errors: RC=5 Value out of range.
RC=10 No font loaded.

1.7 show_filled

Command: SHOW_FILLED

Purpose: Shows characters as filled or outlined.

Format: show_filled <x>

Where: x is 0 (outlined) or 1 (filled).

Errors: RC=10 No font loaded.

1.8 show_template

Command: SHOW_TEMPLATE

Purpose: Shows the bitmap template for the active character window.

Format: show_template <x>

Where: x is 0 (hidden) or 1 (shown).

Errors: RC=5 No template for the current character.
RC=10 No font loaded.

1.9 size_window

Command: SIZE_WINDOW

Purpose: Sets the size of the current character window.

Format: size_window <'x y'>

Where: x is the horizontal size, in pixels (integer).
y is the vertical size, in pixels (integer).

Errors: RC=10 No font loaded.

1.10 goto_char

Command: GOTO_CHAR

Purpose: Changes the character in the active character window to the specified character.

Format: goto_char <x>

Where: x is the decimal character number (integer).

Errors: RC=5 Value out of range.
RC=10 No font loaded.

1.11 quit

Command: QUIT

Purpose: Exits TypeSmith

Format: quit (no parameters)

Errors: <none>

1.12 open_dmf

Command: OPEN_DMF

Purpose: Opens a Soft-Logik DMF file and displays character 65.

Format: open_dmf <'filename'>

Where: filename is the complete file path and name.

Errors: RC=20 Could not open or not a DMF file.

1.13 save_dmf

Command: SAVE_DMF

Purpose: Saves the current font outline as a Soft-Logik DMF file.

Format: save_dmf <'filename'>

Where: filename is the complete file path and name.

Errors: RC=5 Could not save file.
RC=10 Font outline not open.

1.14 open_fm

Command: OPEN_FM

Purpose: Opens a Soft-Logik FM file. A font outline must already be open.

Format: open_fm <'filename'>

Where: filename is the complete file path and name.

Errors: RC=5 Could not open file or not a FM file.
RC=10 No font loaded.

1.15 save_fm

Command: SAVE_FM

Purpose: Saves the current font metric as a Soft-Logik FM file.

Format: save_fm <'filename'>

Where: filename is the complete file path and name.

Errors: RC=5 Could not save file.
RC=10 Font metric not open.

1.16 import_pfb

Command: IMPORT_PFB

Purpose: Imports a PostScript Type 1 PFB file and displays character 65.

Format: import_pfb <'filename'>

Where: filename is the complete file path and name.

Errors: RC=20 Could not find file or not a PFB file.

1.17 export_pfb

Command: EXPORT_PFB

Purpose: Exports the current font outline as a PostScript Type 1 PFB file.

Format: export_pfb <'filename | [1]'>

Where: filename is the complete file path and name.
1 toggles on the generation of hints.

Errors: RC=5 Could not save file.
RC=10 Font outline not open.

1.18 import_afm

Command: IMPORT_AFM

Purpose: Imports a PostScript Type 1 AFM file. A font outline must already be open.

Format: import_afm <'filename'>

Where: filename is the complete file path and name.

Errors: RC=5 Could not find file or not an AFM file.
RC=10 Font outline not open.

1.19 export_afm

Command: EXPORT_AFM

Purpose: Exports the current font metric as a PostScript Type 1 AFM file.

Format: export_afm <'filename'>

Where: filename is the complete file path and name.

Errors: RC=5 Could not save file.
RC=10 Font metric not open.

1.20 import_pfm

Command: IMPORT_PFM

Purpose: Imports a PostScript Type 1 PFM file. A font outline must already be open.

Format: import_pfm <'filename'>

Where: filename is the complete file path and name.

Errors: RC=5 Could not find file or not a PFM file.
RC=10 Font outline not open.

1.21 import_if

Command: IMPORT_IF

Purpose: Imports a Compugraphic Intellifont file and displays character 65.

Format: import_if <'filename'>

Where: filename is the complete file path and name.

Errors: RC=10 Could not find file or not an Intellifont file.

1.22 export_if

Command: EXPORT_IF

Purpose: Exports the current font outline and metric as a Compugraphic Intellifont file.

Format: export_if <'filename1 | [filename2] | [1]'>

Where: filename1 is the complete file path and name for the Intellifont file.
filename2 is the complete file path and name for the Gold Disk files (i.e., .dat and .metric files).
1 toggles on the generation of hints.

Errors: RC=5 Could not save file.
RC=10 Font outline and/or metric not open.

1.23 export_ps

Command: EXPORT_PS

Purpose: Exports the current font outline and metric as a PostScript Type 3 file.

Format: export_ps <'filename'>

Where: filename is the complete file path and name.

Errors: RC=5 Could not save file.
RC=10 Font outline and/or metric not open.

1.24 import_ttf

Command: IMPORT_TTF

Purpose: Imports a TrueType file and displays character 65.

Format: `import_ttf <'filename'>`

Where: `filename` is the complete file path and name.

Errors: RC=10 Could not find file or not a TrueType file.

1.25 export_ttf

Command: `EXPORT_TTF`

Purpose: Exports the current font outline and metric as a TrueType file.

Format: `export_if <'filename | [1]'`

Where: `filename` is the complete file path and name for the TrueType file.
`1` toggles on the generation of instructions (hints).

Errors: RC=5 Could not save file.
RC=10 Font outline and/or metric not open.

1.26 import_rff

Command: `IMPORT_RFF`

Purpose: Imports an IFF RFF file and displays character 65.

Format: `import_rff <'filename'>`

Where: `filename` is the complete file path and name.

Errors: RC=10 Could not find file or not an IFF RFF file.

1.27 export_rff

Command: `EXPORT_RFF`

Purpose: Exports the current font outline and metric as an IFF RFF file.

Format: `export_if <'filename'>`

Where: `filename` is the complete file path and name for the IFF RFF file.

Errors: RC=5 Could not save file.
RC=10 Font outline and/or metric not open.

1.28 import_template

Command: IMPORT_TEMPLATE

Purpose: Imports and displays an IFF ILBM bitmap template for the current character.

Format: import_template <'filename'>

Where: filename is the complete file path and name.

Errors: RC=10 Could not find file or not an IFF ILBM file.

1.29 import_dr2d

Command: IMPORT_DR2D

Purpose: Imports an IFF DR2D drawing and pastes it into the current character.

Format: import_dr2d <'filename'>

Where: filename is the complete file path and name.

Errors: RC=5 Could not find file or not an IFF DR2D file.
RC=10 Font outline not open.

1.30 export_dr2d

Command: EXPORT_DR2D

Purpose: Exports the current character outline as an IFF DR2D file.

Format: export_dr2d <'filename'>

Where: filename is the complete file path and name.

Errors: RC=5 Could not save file.
RC=10 Font outline not open.

1.31 new_font

Command: NEW_FONT

Purpose: Starts a new font with default values, and displays character 65 in a character window.

Format: new_font (no parameters)

Errors: RC=20 Out of memory.
RC=10 Could not open window.

1.32 new_metric

Command: NEW_METRIC

Purpose: Clears the current font metric and generates a new font metric.

Format: new_metric <'track space'>

Where: track is the tracking value as a percentage (integer).
space is the space character width in character units (integer).

Errors: RC=10 Font outline not open.
RC=5 Unable to generate new metrics.

1.33 next_font

Command: NEXT_FONT

Purpose: Activates the first view of the next font. If the last font is reached, the first font will be activated. The name of the font made active will be returned and stored in the RESULT variable if successful.

Format: next_font (no parameters)

Errors: RC=10 Font outline is not open.
RC=5 Only one open font.

1.34 new_view

Command: NEW_VIEW

Purpose: Opens a new view window for the active font showing a specified character. Returns the number of the new view in the RESULT variable if successful.

Format: new_view <char>

Where: char is the ASCII number of the character to view (integer).

Errors: RC=10 Font outline is not open.
RC=5 Out of memory or value out of range.

1.35 next_view

Command: NEXT_VIEW

Purpose: Activates the next view of the same font. If the last view is reached, the first view will be activated. The view number of

the view made active will be returned and stored in the RESULT variable if successful.

Format: next_view (no parameters)

Errors: RC=10 Font outline is not open.
RC=5 Only one open view.

1.36 get_view

Command: GET_VIEW

Purpose: Gets the identification number of the current view and stores it in the RESULT variable.

Format: get_view (no parameters)

Errors: RC=10 No open view.

1.37 close_font

Command: CLOSE_FONT

Purpose: Closes the active font and all its associated views.

Format: close_font (no parameters)

Errors: RC=10 No font to close.

1.38 close_view

Command: CLOSE_VIEW

Purpose: Closes the active view.

Format: close_view (no parameters)

Errors: RC=10 No view to close.

1.39 move_char

Command: MOVE_CHAR

Purpose: Moves a character to another character position. Existing paths in the new character position will be erased. This command also copies metric information and updates kerning pairs.

Format: `move_char <'num1 num2'>`

Where: `num1` is the character to move (integer between 0 and 255).
`num2` is the character position to move to (integer between 0 and 255).

Errors: `RC=10` No outline font loaded.
`RC=5` Value out of range.

1.40 `reduce_points`

Command: `REDUCE_POINTS`

Purpose: Removes unnecessary points from the current character.

Format: `reduce_points` (no parameters)

Errors: `RC=10` No outline font loaded.

1.41 `correct_pathdirs`

Command: `CORRECT_PATHDIRS`

Purpose: Removes unnecessary points from the current character.

Format: `correct_pathdirs` (no parameters)

Errors: `RC=10` No outline font loaded.

1.42 `moveto`

Command: `MOVETO`

Purpose: Move pen to a specified position.

Format: `moveto <'x y'>`

Where: `x` is the horizontal coordinate (integer).
`y` is the vertical coordinate (integer).

Errors: `RC=10` No outline font loaded.

1.43 `lineto`

Command: LINETO

Purpose: Draw a line in the path in progress from the current pen position to the specified position which becomes the new pen position.

Format: lineto <'x y'>

Where: x is the horizontal coordinate (integer).
y is the vertical coordinate (integer).

Errors: RC=10 No outline font loaded.

1.44 curveto

Command: CURVETO

Purpose: Draw a bézier curve in the path in progress from the current pen position to the specified position which becomes the new pen position.

Format: curveto <'bcpx1 bcpy1 bcpx2 bcpy2x y'>

Where: bcpx1 is the horizontal coordinate of control point 1 (integer).
bcpy1 is the vertical coordinate of control point 1 (integer).
bcpx2 is the horizontal coordinate of control point 2 (integer).
bcpy2 is the vertical coordinate of control point 2 (integer).
x is the horizontal coordinate (integer).
y is the vertical coordinate (integer).

Errors: RC=10 No outline font loaded.

1.45 closepath

Command: CLOSEPATH

Purpose: Closes the path in progress. If the pen position is not the same as the startpoint of the path (the moveto), a straight line will be drawn to that position.

Format: closepath (no parameters)

Errors: RC=10 No outline font loaded.

1.46 select_all

Command: SELECT_ALL

Purpose: Selects all points in all paths in the current character.

Format: `select_all` (no parameters)

Errors: RC=10 No font outline loaded.
RC=5 No paths in character.

1.47 `unselect_all`

Command: `UNSELECT_ALL`

Purpose: Deselects all points in all paths.

Format: `unselect_all` (no parameters)

Errors: RC=10 No font outline loaded.

1.48 `first_path`

Command: `FIRST_PATH`

Purpose: Selects the first point of the first path in the current character.

Format: `first_path` (no parameters)

Errors: RC=5 No paths in character.
RC=10 No font outline loaded.

1.49 `select_path`

Command: `SELECT_PATH`

Purpose: Selects all the points in the current path.

Format: `select_path` (no parameters)

Notes: One point must be selected prior to executing this command.

Errors: RC=10 More than one point is selected.
RC=10 No font outline loaded.
RC=10 No point is selected.
RC=10 No paths in character.
RC=5 Activation failed.

1.50 `next_path`

Command: NEXT_PATH

Purpose: Selects the startpoint of the next path in the current character.

Format: next_path (no parameters)

Notes: If the last path is currently selected, this selects the startpoint of the first path. One point must be selected prior to executing this command.

Errors: RC=10 More than one point is selected or no font outline loaded.
RC=10 No point is selected.
RC=5 Only one path.

1.51 previous_path

Command: PREVIOUS_PATH

Purpose: Selects the startpoint of the previous path in the current character.

Format: previous_path (no parameters)

Notes: If the first path is currently selected, this selects the startpoint of the last path. One point must be selected prior to executing this command.

Errors: RC=10 More than one point is selected.
RC=10 No point is selected.
RC=5 Only one path.

1.52 goto_start

Command: GOTO_START

Purpose: Deselects the current point and selects the first point in the current path.

Format: goto_start (no parameters)

Notes: One point must be selected prior to executing this command.

Errors: RC=10 More than one point is selected.
RC=10 No point is selected.
RC=10 No font outline loaded.

1.53 next_point

Command: NEXT_POINT

Purpose: Deselects the current point and selects the next point in the current path.

Format: next_point (no parameters)

Notes: One point must be selected prior to executing this command.

Errors: RC=10 More than one point is selected.
RC=10 No point is selected.
RC=10 No font outline loaded.
RC=5 No next point.

1.54 previous_point

Command: PREVIOUS_POINT

Purpose: Deselects the current point and selects the previous point in the current path.

Format: previous_point (no parameters)

Notes: One point must be selected prior to executing this command.

Errors: RC=10 More than one point is selected.
RC=10 No point is selected.
RC=10 No font outline loaded.
RC=5 No previous point.

1.55 select_area

Command: SELECT_AREA

Purpose: Selects all the points inside of a specified area.

Format: select_area <'x1 y1 x2 y2'>

Where: x1 is the horizontal coordinate of the area's upper left corner.
y1 is the vertical coordinate of the area's upper left corner.
x2 is the horizontal coordinate of the area's bottom right corner.
y2 is the vertical coordinate of the area's bottom right corner.

Errors: RC=10 No font outline loaded.
RC=5 Value out of range or nothing to select.

1.56 get_numpaths

Command: GET_NUMPATHS

Purpose: Gets the number of paths in the current character and stores it in the RESULT variable.

Format: get_numpaths (no parameters)

Errors: RC=10 No font outline loaded.

1.57 get_numpoints

Command: GET_NUMPATHS

Purpose: Gets the number of points in the current path and stores it in the RESULT variable.

Format: get_numpoints (no parameters)

Notes: One point must be selected prior to executing this command.

Errors: RC=10 More than one point is selected.
RC=10 No point is selected.
RC=10 No font outline loaded.
RC=5 Path started but not closed.

1.58 get_pointx

Command: GET_POINTX

Purpose: Gets the horizontal coordinate of the selected point and stores it in the RESULT variable.

Format: get_pointx (no parameters)

Notes: One point must be selected prior to executing this command.

Errors: RC=10 More than one point is selected.
RC=10 No point is selected.
RC=10 No font outline loaded.

1.59 get_pointy

Command: GET_POINTY

Purpose: Gets the vertical coordinate of the selected point and stores it in the RESULT variable.

Format: get_pointy (no parameters)

Notes: One point must be selected prior to executing this command.

Errors: RC=10 More than one point is selected.
RC=10 No point is selected.
RC=10 No font outline loaded.

1.60 get_prev_bcpX

Command: GET_PREV_BCPX

Purpose: Gets the horizontal coordinate of the previous bézier control point (curve handle) for the selected point and stores it in the RESULT variable.

Format: get_prev_bcpX (no parameters)

Notes: One point must be selected prior to executing this command.

Errors: RC=10 More than one point is selected.
RC=10 No point is selected.
RC=10 No font outline loaded.
RC=5 previous segment is not a CURVETO.

1.61 get_prev_bcpY

Command: GET_PREV_BCPY

Purpose: Gets the vertical coordinate of the previous bézier control point (curve handle) for the selected point and stores it in the RESULT variable.

Format: get_prev_bcpY (no parameters)

Notes: One point must be selected prior to executing this command.

Errors: RC=10 More than one point is selected.
RC=10 No point is selected.
RC=10 No font outline loaded.
RC=5 previous segment is not a CURVETO.

1.62 get_next_bcpX

Command: GET_NEXT_BCPX

Purpose: Gets the horizontal coordinate of the next bézier control point (curve handle) for the selected point and stores it in the RESULT variable.

Format: get_next_bcpX (no parameters)

Notes: One point must be selected prior to executing this command.

Errors: RC=10 More than one point is selected.
RC=10 No point is selected.
RC=10 No font outline loaded.
RC=5 next segment is not a CURVETO.

1.63 get_next_bcpy

Command: GET_NEXT_BCPY

Purpose: Gets the vertical coordinate of the next bézier control point (curve handle) for the selected point and stores it in the RESULT variable.

Format: get_next_bcpy (no parameters)

Notes: One point must be selected prior to executing this command.

Errors: RC=10 More than one point is selected.
RC=10 No point is selected.
RC=10 No font outline loaded.
RC=5 next segment is not a CURVETO.

1.64 get_pathdirection

Command: GET_PATHDIRECTION

Purpose: Gets the path direction for the current path and stores it in the RESULT variable.

Format: get_pathdirection (no parameters)

Notes: One point must be selected prior to executing this command.

Results: 0: clockwise
1: counterclockwise

Errors: RC=10 More than one point is selected.
RC=10 No point is selected.
RC=10 No font outline loaded.

1.65 get_previous_type

Command: GET_PREVIOUS_TYPE

Purpose: Gets the type of the segment before the selected point and stores it in the RESULT variable.

Format: get_previous_type (no parameters)

Notes: One point must be selected prior to executing this command.

Results: 'c': curveto
'l': lineto
'm': moveto

Errors: RC=10 More than one point is selected.
RC=10 No point is selected.
RC=10 No font outline loaded.

1.66 get_char_num

Command: GET_CHAR_NUM

Purpose: Gets the ASCII value for the current character and stores it in the RESULT variable.

Format: get_char_num (no parameters)

Errors: RC=10 No font open.

1.67 get_version

Command: GET_VERSION

Purpose: Gets the font version number and stores it in the RESULT variable.

Format: get_version (no parameters)

Note: The result format is %ld.%ld where the first number is the version and the second is the revision.

Errors: RC=10 No font open.

1.68 get_width

Command: GET_WIDTH

Purpose: Gets the width of the current character and stores it in the RESULT variable.

Format: get_width <char>

Where: char is the ASCII character number (integer).

Errors: RC=10 No font open or no metrics data present.
RC=5 Value out of range.

1.69 set_composition

Command: SET_COMPOSITION

Purpose: Sets a composition reference for a character.

Format: set_composition <'char n comp x y'>

Where: char is the ASCII character number (integer).
n is the composition reference number (integer).
comp is the referenced ASCII character number (integer).
x is the horizontal offset (integer).
y is the vertical offset (integer).

Note: If comp=0, then the lookup table will be terminated. The last composition entry MUST be a zero.

Errors: RC=10 No font open.
RC=5 Value out of range.
RC=5 No more compositions to add

1.70 set_point

Command: SET_POINT

Purpose: Changes the coordinates of the current point.

Format: set_point <'x y pbcpx pbcpy nbcpx nbcpy'>

Where: x is the horizontal coordinate (integer).
y is the vertical coordinate (integer).
pbcx is the horizontal coordinate of the previous bézier control point (curve handle) (integer).
pbcy is the vertical coordinate of the previous bézier control point (curve handle) (integer).
nbcx is the horizontal coordinate of the next bézier control point (curve handle) (integer).
nbcy is the vertical coordinate of the next bézier control point (curve handle) (integer).

Notes: One point must be selected prior to executing this command. If either the previous segment or the next segment is a LINETO, specify 0 for the BCP positions. (Out of range values will be ignored.) If both the previous and next segments are LINETO's, the BCP positions may be set to 0, or they may be omitted.

Errors: RC=10 More than one point is selected.
RC=10 No point is selected.
RC=10 No font outline loaded.

1.71 set_pathdirection

Command: SET_PATHDIRECTION

Purpose: Changes the path direction for the current path.

Format: set_pathdirection <x>

Where: x is 0 (clockwise) or 1 (counterclockwise).

Notes: One point must be selected prior to executing this command.

Errors: RC=10 More than one point is selected.

RC=10 No point is selected.

RC=10 No font outline loaded.

1.72 set_fontname

Command: SET_FONTNAME

Purpose: Sets the font name.

Format: set_fontname

Where: font is the font name (string).

Errors: RC=10 No font open.

RC=10 Illegal font name.

1.73 set_family

Command: SET_FAMILY

Purpose: Sets the font family name.

Format: set_family <family>

Where: family is the font family name (string).

Errors: RC=10 No font open.

RC=10 Illegal family name.

1.74 set_weight

Command: SET_WEIGHT

Purpose: Sets the font weight name.

Format: set_weight <weight>

Where: weight is the font weight (string).

Errors: RC=10 No font open.
RC=10 Illegal font weight name.

1.75 set_fullname

Command: SET_FULLNAME

Purpose: Sets the full font name.

Format: set_fullname <name>

Where: name is the full font name (string).

Errors: RC=10 No font open.
RC=10 Illegal full name.

1.76 set_fontid

Command: SET_FONTID

Purpose: Sets the font identification number.

Format: set_fontid <n>

Where: n is the font id (integer).

Errors: RC=10 No font open.

1.77 set_notice

Command: SET_NOTICE

Purpose: Sets the font notice.

Format: set_notice <notice>

Where: notice is the font notice (string).

Errors: RC=10 No font open.
RC=10 Illegal font notice.

1.78 set_version

Command: SET_VERSION

Purpose: Sets the font version number.

Format: set_version <version>

Where: version is the font notice (integer).

Note: Enter a version number with the decimal moved two places to the right. (i.e., Enter 100 instead of 1.00, or 2345 instead of 23.45.)

Errors: RC=10 No font open.

1.79 set_width

Command: SET_WIDTH

Purpose: Sets the width of a character.

Format: set_width <'char width'>

Where: char is the ASCII character number (integer).
width is the character width in character units (integer).

Errors: RC=10 No font open.
RC=10 No metrics data,
RC=5 Value out of range.

1.80 rotate

Command: ROTATE

Purpose: Rotates the selected points and/or paths about the center of their bounding box.

Format: rotate <ra>

Where: ra is the rotation angle in degrees (integer between -360 to 360).

Errors: RC=10 No points are selected (at least 2 points must be selected)
RC=10 No font loaded.
RC=5 Rotation failed or value out of range.

1.81 skew

Command: SKEW

Purpose: Skews the selected points and/or paths about the center of their bounding box.

Format: skew <sa>

Where: sa is the skew angle in degrees (integer between -360 to 360).

Errors: RC=10 No points are selected (at least 2 points must be selected)
RC=10 No font loaded.
RC=5 Skew operation failed or value out of range.

1.82 scale

Command: SCALE

Purpose: Scales the selected points and/or paths about the center of their bounding box.

Format: scales <'x y'>

Where: x is the horizontal scaling factor in percent (integer between 1 and 500).
y is the vertical scaling factor in percent (integer between 1 and 500).

Errors: RC=10 No points are selected (at least 2 points must be selected)
RC=10 No font loaded.
RC=5 Scaling failed or values out of range.

1.83 flip

Command: FLIP

Purpose: Flips the selected points and/or paths about the center of their bounding box.

Format: flip <d>

Where: d is 0 (horizontal flip) or x=1 (vertical flip).

Errors: RC=10 No points are selected (at least 2 points must be selected)
RC=10 No font loaded.
RC=5 Flip operation failed or value out of range.

1.84 move

Command: MOVE

Purpose: Moves the selected points and/or paths the specified horizontal and vertical amounts.

Format: flip <'mx my'>

Where: mx is the horizontal distance to move (integer).

my is the vertical distance to move (integer).

Errors: RC=10 No points are selected.
RC=10 No font loaded.

1.85 copy

Command: COPY

Purpose: Copies the selected points and/or paths to the clipboard.

Format: copy (no parameters)

Errors: RC=10 No points are selected.
RC=10 No font loaded.

1.86 cut

Command: CUT

Purpose: Cuts the selected points and/or paths to the clipboard.

Format: cut (no parameters)

Errors: RC=10 No points are selected.
RC=10 No font loaded.

1.87 paste

Command: PASTE

Purpose: Pastes the clipboard contents to the current character window.

Format: paste ['x y']

Where: x is the horizontal coordinate of the top left corner of the
paths to paste.
y is the vertical coordinate of the top left corner of the
paths to paste.

Notes: If x and y are not specified, the clipboard contents will be
pasted at the same position as they were when copied or cut to
the clipboard.

Errors: RC=10 No font loaded.

1.88 delete

Command: DELETE

Purpose: Removes the selected points and/or paths from the character.

Format: delete (no parameters)

Errors: RC=10 No points are selected.
RC=10 No font loaded.

1.89 beepscreen

Command: BEEPSCREEN

Purpose: Beeps the screen to warn the user that an error has occurred. The screen will be flashed or the speakers will beep depending on the user's Preferences/Sound setting.

Format: beepscreen <x>

Where: x is 0 (beeps the TypeSmith screen only) or x=1 (beeps all screens).

1.90 display_alert

Command: DISPLAY_ALERT

Purpose: Opens an alert requester to notify the user of something.

Format: display_alert <'message|positive[|negative]'\>

Where: message is the message to display in the alert requester (string).
positive is the name of the positive gadget (string).

Options: negative is the name of the negative gadget (string).

Errors: RC=5 if the Negative gadget is selected.
RC=10 Invalid arguments

Example: display_alert 'Select a gadget.|Ok|Cancel'

1.91 get_string

Command: GET_STRING

Purpose: Opens a requester to request input from the user. Gets a string from a text gadget and stores it in the RESULT variable.

Format: `get_string <'message|positive|negative'>`

Where: `message` is the message to display in the requester (string).
`positive` is the name of the positive gadget (string).
`negative` is the name of the negative gadget (string).

Notes: The user input will be stored in RESULT only if the Positive gadget is selected.

Errors: RC=5 if the Negative gadget is selected.
RC=10 Invalid arguments

Example: `get_string 'Enter text:|Ok|Cancel'`

1.92 get_file

Command: `GET_FILE`

Purpose: Opens an ASL open file requester to request that the user select a file path and name, which will be stored in the RESULT variable.

Format: `get_file <'title|positive|path|[file]'`

Where: `title` is the title of the file requester (string).
`positive` is the name of the positive gadget (string).
`path` is the default file path (string).
`file` is the default default filename (string).

Notes: The filepath and name entered will be stored in RESULT only if the Positive gadget is selected.

Errors: RC=5 if the Negative gadget is selected.
RC=10 Invalid arguments

Examples: `get_file 'Select File...|Ok|TypeSmith/TSfonts|Untitled.dmf'`
`get_file 'Select File...|Ok|ram:|'`

1.93 help

Command: `HELP`

Purpose: Lists all supported ARexx commands to the standard output console.

Format: `help` (no parameters)

1.94 print_character

Command: PRINT_CHARACTER

Purpose: Prints a character of the current font to the Preferences printer.

Format: print_character <'density size'>

Where: density is the PrinterGfx density (integer, 1-7).
size is the output size in typographic points.

Errors: RC=10 No font open.
RC=10 Values out of range.
RC=10 Printer error.

1.95 print_overview

Command: PRINT_OVERVIEW

Purpose: Prints a complete character set table of the current font to the Preferences printer.

Format: print_overview <'density size'>

Where: density is the PrinterGfx density (integer, 1-7).
size is the output size in typographic points.

Errors: RC=10 No font open.
RC=10 Values out of range.
RC=10 Printer error.

1.96 print_string

Command: PRINT_STRING

Purpose: Prints a text string in the current font to the Preferences printer.

Format: print_string <'density size'>

Where: density is the PrinterGfx density (integer, 1-7).
size is the output size in typographic points.

Errors: RC=10 No font open.
RC=10 Values out of range.
RC=10 Printer error.

1.97 t_autotrace

Command: T_AUTOTRACE

Purpose: Autotraces the bitmap template for the current character to create additional paths for the character.

Format: `t_autotrace <accuracy>`

Where: `accuracy` is 0 (normal), 1 (medium) or 2 (accurate)

Errors: RC=10 No template, or value out of range.
RC=20 No font open.

1.98 t_edit

Command: `T_EDIT`

Purpose: Toggle the template edit mode.

Format: `t_edit <mode>`

Where: `mode` is 0 (off) or 1 (on).

Errors: RC=5 Already in that mode.
RC=10 No font open.

Notes: RC will be set to 0 if it successfully changed modes but no template was open.

1.99 t_generate_char

Command: `T_GENERATE_CHAR`

Purpose: Rasterize the current character and create bitmap template.

Format: `t_generate_char` (no parameters)

Errors: RC=10 Rasterizer failed.
RC=20 No font open.

Notes: RC will be set to 0 if it successfully changed modes but no template was open.

1.100 t_get_point

Command: `T_GET_POINT`

Purpose: Gets the pixel at coordinate `x y` if the template is set, and stores it in the `RESULT` variable.

Format: `t_get_point <'x y'>`

Where: `x` is the horizontal coordinate (integer).

y is the vertical coordinate (integer).

Errors: RC=10 No font loaded.

Results: 1: if pixel is set
0: if pixel is clear

Errors: RC=5 No template.
RC=10 No font loaded.

1.101 t_get_pos

Command: T_GET_POS

Purpose: Gets the position of the left edge of the current template and stores it as an array in the RESULT variable.

Format: t_get_pos (no parameters)

Errors: RC=5 No template.
RC=10 No font open.

1.102 t_get_charwidth

Command: T_GET_CHARWIDTH

Purpose: Gets the width of bitmap font character in the current window and stores it in the RESULT variable.

Format: t_get_charwidth (no parameters)

Errors: RC=5 No template.
RC=10 No font open.

1.103 t_new_bitmapfont

Command: T_NEW_BITMAPFONT

Purpose: Generates bitmaps for the complete character set and toggles on display of the template.

Format: t_new_bitmapfont <size>

Where: size is the size of the bitmap characters in typographic points, 9 to 48 (integer).

Errors: RC=10 size out of range or out of memory
RC=20 No font open.

1.104 t_open_abf

Command: T_OPEN_ABF

Purpose: Opens a PostScript ABF bitmap font.

Format: t_open_abf <'filename'>

Where: filename is the complete file path and name.

Errors: RC=10 loading error or out of memory
RC=20 No font open.

1.105 t_open_amigafont

Command: T_OPEN_AMIGAFONT

Purpose: Opens an AmigaDOS bitmap font.

Format: t_open_amigafont <'fontname'>

Where: fontname is the font name with '.font' and the size separated with a vertical bar, appended to it. i.e., topaz.font|8

Errors: RC=10 insufficient input, no memory or file not found
RC=20 No font open.

1.106 t_open_iffilbm

Command: T_OPEN_IFFILBM

Same as import_template.

1.107 t_open_slfont

Command: T_OPEN_SLFONT

Purpose: Opens a Soft-Logik bitmap font.

Format: t_open_slfont <'filename'>

Where: filename is the complete file path and name.

Errors: RC=10 loading error or out of memory
RC=20 No font open.

1.108 t_clear

Command: T_CLEAR

Purpose: Clears the template in the current window.

Format: t_clear (no parameters)

Errors: RC=10 No font open.

1.109 t_save_abf

Command: T_SAVE_ABF

Purpose: Saves the current bitmap font in PostScript ABF format.

Format: t_save_abf <'filename'>

Where: filename is the complete file path and name.

Errors: RC=10 error while saving
RC=20 No font open.

1.110 t_save_amigafont

Command: T_SAVE_AMIGAFONT

Purpose: Saves the current bitmap font in AmigaDOS bitmap format.

Format: t_save_amigafont <'filename'>

Where: filename is the complete file path and name of the font
description file (#?.font). Directories for the actual
bitmaps will be created if necessary. i.e.,
'fonts:topaz.font'

Errors: RC=10 error while saving
RC=20 No font open.

1.111 t_save_iffilbm

Command: T_SAVE_IFFILBM

Purpose: Saves the current template in IFF ILBM format.

Format: t_save_iffilbm <'filename'>

Where: filename is the complete file path and name.

Errors: RC=10 error while saving
RC=20 No font open.

1.112 t_save_slfont

Command: T_SAVE_SLFONT

Purpose: Saves the current bitmap font in Soft-Logik format.

Format: t_save_slfont <'filename'>

Where: filename is the complete file path and name.

Errors: RC=10 error while saving
RC=20 No font open.

1.113 t_set_charwidth

Command: T_SET_CHARWIDTH

Purpose: Sets the width of the current bitmap font character.

Format: t_set_charwidth <width>

Where: width is the width in pixels.

Errors: RC=5 No template.
RC=10 No font open.

1.114 t_set_point

Command: T_SET_POINT

Purpose: Sets a pixel in the current template to desired value.

Format: t_set_point <'x y color'>

Where: x is the pixel's horizontal coordinate.
y is the pixel's vertical coordinate.
color is 0 (off) 1 (on).

Errors: RC=5 No template.
RC=10 No font open.

1.115 t_set_pos

Command: T_SET_POS

Purpose: Sets the position of the left edge of the template in the current character.

Format: t_set_pos <'x y'>

Where: x is the horizontal coordinate.
y is the vertical coordinate.

Errors: RC=5 No template.
RC=10 No font open.

1.116 t_get_dimensions

Command: T_GET_DIMENSIONS

Purpose: Gets the dimensions (width and height) of the current template and stores them as an array in the RESULT variable.

Format: t_get_dimensions (no parameters)

Errors: RC=5 No template.
RC=10 No font open.

1.117 t_set_dimensions

Command: T_SET_DIMENSIONS

Purpose: Sets the dimensions (width and height) of the current template.

Format: t_set_dimensions <'x y'>

Where: x is the horizontal coordinate.
y is the vertical coordinate.

Errors: RC=5 No template.
RC=10 No font open.

1.118 t_shift

Command: T_SHIFT

Purpose: Shifts contents of the current template by one pixel.

Format: t_shift <direction>

Where: direction is 0 (up), 1 (right), 2 (down), or 3 (left).

Errors: RC=5 No template.
RC=10 No font open.

1.119 t_show

Command: T_SHOW

Same as show_template.
