

# **PROGED.MANUAL**

<b>COLLABORATORS</b>
----------------------

	<i>TITLE :</i> PROGED.MANUAL	
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>
WRITTEN BY		January 18, 2023
		<i>SIGNATURE</i>

<b>REVISION HISTORY</b>
-------------------------

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>PROGED.MANUAL</b>	<b>1</b>
1.1	ProgED V2.1 ©1995-96 Giovanni Lambiase . . . . .	1
1.2	Triton . . . . .	2
1.3	Thanks . . . . .	2
1.4	Author . . . . .	3
1.5	Known bugs . . . . .	4
1.6	SAS/C Support . . . . .	4
1.7	Collaboration . . . . .	4
1.8	Distribution . . . . .	5
1.9	Requirements . . . . .	6
1.10	Introduction . . . . .	6
1.11	Features . . . . .	7
1.12	Programmable keyword colors . . . . .	8
1.13	Keyword indentation . . . . .	8
1.14	Automatic phrase completion . . . . .	8
1.15	Automatic case correction . . . . .	8
1.16	Block folding . . . . .	9
1.17	Programmable keyboard and menu . . . . .	9
1.18	Integrated compiler frontend for SAS/C V6.x . . . . .	9
1.19	Programmable quick function search . . . . .	9
1.20	Templates for structure creation . . . . .	9
1.21	Project management . . . . .	9
1.22	Search for static references between files . . . . .	10
1.23	Search for dynamic referemences between project files . . . . .	10
1.24	File hunter . . . . .	10
1.25	Integreted API (APplication Interface) . . . . .	10
1.26	Autosave & Backup . . . . .	10
1.27	Auto decrunch of PowerPacker crunched files . . . . .	10
1.28	XPk libraries support . . . . .	11
1.29	Fast mode for slow processors . . . . .	11

---

---

1.30	On line macro creation . . . . .	11
1.31	Window freeze/unfreeze . . . . .	11
1.32	Iconify . . . . .	11
1.33	Columnar block marking . . . . .	11
1.34	Indentation & block layout . . . . .	12
1.35	Multilevel Undo/Redo . . . . .	12
1.36	Print texts on two colomns . . . . .	12
1.37	Handler to read files in memory . . . . .	12
1.38	Handler to read files in project . . . . .	12
1.39	Window splitting . . . . .	13
1.40	On line macro creation . . . . .	13
1.41	File-types management . . . . .	13
1.42	Static markers management . . . . .	13
1.43	Template/Tooltypes . . . . .	14
1.44	FILE/M, . . . . .	14
1.45	CX_POPKEY/K, . . . . .	14
1.46	CX_PRIORITY/K, . . . . .	14
1.47	CX_POPUP/K, . . . . .	15
1.48	HANDLERPED/K, . . . . .	15
1.49	HANDLERPRJ/K, . . . . .	15
1.50	CONFIG/K, . . . . .	15
1.51	SCREEN/K . . . . .	15
1.52	Default settings . . . . .	15
1.53	Menu: Project . . . . .	25
1.54	Menu: Project/New . . . . .	25
1.55	Menu: Project/Open . . . . .	25
1.56	Menù: Project/Load . . . . .	25
1.57	Menu: Project/Save . . . . .	25
1.58	Menu: Project/Save as... . . . . .	25
1.59	Menu: Project/Save as... (XPK) . . . . .	25
1.60	Menu: Project/Save all changes . . . . .	26
1.61	Menu: Project/Save & close . . . . .	26
1.62	Menu: Project/Close . . . . .	26
1.63	Menu: Project/----- . . . . .	26
1.64	Menu: Project/Revert . . . . .	26
1.65	Menu: Project/Clear . . . . .	26
1.66	Menu: Project/Insert . . . . .	26
1.67	Menu: Project/Append . . . . .	27
1.68	Menu: Project/New name . . . . .	27

---

---

1.69 Menu: Project/-----	27
1.70 Menu: Project/Current dir	27
1.71 Menu: Project/-----	27
1.72 Menu: Project/Print	27
1.73 Menu: Project/-----	27
1.74 Menu: Project/About	27
1.75 Menu: Project/-----	28
1.76 Menu: Project/Save all & quit	28
1.77 Menu: Project/Quit	28
1.78 Menu: Edit	28
1.79 Menu: Edit/Mark	28
1.80 Menu: Edit/Mark columnar	28
1.81 Menu: Edit/Mark lines	28
1.82 Menu: Edit/Select all	29
1.83 Menu: Edit/-----	29
1.84 Menu: Edit/Cut	29
1.85 Menu: Edit/Cut/To std clip	29
1.86 Menu: Edit/Cut/To clip 1	29
1.87 Menu: Edit/Cut/To clip 2	29
1.88 Menu: Edit/Cut/To clip 3	29
1.89 Menu: Edit/Copy	30
1.90 Menu: Edit/Copy/To std clip	30
1.91 Menu: Edit/Copy/To clip 1	30
1.92 Menu: Edit/Copy/To clip 2	30
1.93 Menu: Edit/Copy/To clip 3	30
1.94 Menu: Edit/Insert	30
1.95 Menu: Edit/Insert/From std clip	31
1.96 Menu: Edit/Insert/From clip 1	31
1.97 Menu: Edit/Insert/From clip 2	31
1.98 Menu: Edit/Insert/From clip 3	31
1.99 Menu: Edit/Clear	31
1.100Menu: Edit/-----	32
1.101Menu: Edit/Cut line	32
1.102Menu: Edit/Insert line	32
1.103Menu: Edit/Swap line	32
1.104Menu: Edit/Double line	32
1.105Menu: Edit/Delete to EOL	32
1.106Menu: Edit/-----	33
1.107Menu: Edit/Undo	33

---

---

1.108Menu: Edit/Redo . . . . .	33
1.109Menu: Edit/----- . . . . .	33
1.110Menu: Edit/Layout block . . . . .	33
1.111Menu: Edit/Layout block/Left . . . . .	33
1.112Menu: Edit/Layout block/Right . . . . .	34
1.113Menu: Edit/Layout block/Center . . . . .	34
1.114Menu: Edit/Layout block/Full . . . . .	34
1.115Menu: Edit/Move block . . . . .	35
1.116Menu: Edit/Move block/Left . . . . .	35
1.117Menu: Edit/Move block/Right . . . . .	35
1.118Menu: Edit/Move block/Center . . . . .	36
1.119Menu: Edit/Indent block . . . . .	36
1.120Menu: Edit/Save block as... . . . . .	36
1.121Menu: Search&Replace . . . . .	36
1.122Menu: Search&Replace/Jump to line . . . . .	36
1.123Menu: Search&Replace/Unfold . . . . .	37
1.124Menu: Search&Replace/----- . . . . .	37
1.125Menu: Search&Replace/Jump to byte . . . . .	37
1.126Menu: Search&Replace/Unfold . . . . .	37
1.127Menu: Search&Replace/----- . . . . .	37
1.128Menu: Search&Replace/Set marker . . . . .	37
1.129Menu: Search&Replace/Set marker/1 . . . . .	38
1.130Menu: Search&Replace/Set marker/2 . . . . .	38
1.131Menu: Search&Replace/Set marker/3 . . . . .	38
1.132Menu: Search&Replace/Set marker/4 . . . . .	38
1.133Menu: Search&Replace/Set marker/5 . . . . .	38
1.134Menu: Search&Replace/Set marker/6 . . . . .	39
1.135Menu: Search&Replace/Set marker/7 . . . . .	39
1.136Menu: Search&Replace/Set marker/8 . . . . .	39
1.137Menu: Search&Replace/Set marker/9 . . . . .	39
1.138Menu: Search&Replace/Set marker/10 . . . . .	39
1.139Menu: Search&Replace/Goto marker . . . . .	40
1.140Menu: Search&Replace/Goto marker/1 . . . . .	40
1.141Menu: Search&Replace/Goto marker/2 . . . . .	40
1.142Menu: Search&Replace/Goto marker/3 . . . . .	40
1.143Menu: Search&Replace/Goto marker/4 . . . . .	41
1.144Menu: Search&Replace/Goto marker/5 . . . . .	41
1.145Menu: Search&Replace/Goto marker/6 . . . . .	41
1.146Menu: Search&Replace/Goto marker/7 . . . . .	41

---

---

1.147Menu: Search&Replace/Goto marker/8 . . . . .	42
1.148Menu: Search&Replace/Goto marker/9 . . . . .	42
1.149Menu: Search&Replace/Goto marker/10 . . . . .	42
1.150Menu: Search&Replace/Set marker . . . . .	43
1.151Menu: Search&Replace/Jump to ... . . . .	43
1.152Menu: Search&Replace/Unfold . . . . .	43
1.153Menu: Search&Replace/----- . . . . .	43
1.154Menu: Search&Replace/Search for... . . . .	43
1.155Menu: Search&Replace/Repeat search fwd . . . . .	44
1.156Menu: Search&Replace/Repeat search bkw . . . . .	44
1.157Menu: Search&Replace/----- . . . . .	44
1.158Menu: Search&Replace/Replace . . . . .	44
1.159Menu: Search&Replace/Repeat replace . . . . .	45
1.160Menu: Search&Replace/----- . . . . .	45
1.161Menu: Search&Replace/Matching bracket . . . . .	45
1.162Menu: Search&Replace/Check bracket . . . . .	45
1.163Menu: Special . . . . .	46
1.164Menu: Special/Fold . . . . .	46
1.165Menu: Special/Unfold . . . . .	46
1.166Menu: Special/Fold All . . . . .	46
1.167Menu: Special/Unfold All . . . . .	46
1.168Menu: Special/----- . . . . .	46
1.169Menu: Special/Window . . . . .	46
1.170Menu: Special/Windows/Arrange ver . . . . .	47
1.171Menu: Special/Windows/Arrange hor . . . . .	47
1.172Menu: Special/Windows/Zip . . . . .	47
1.173Menu: Special/Windows/Enlarge . . . . .	47
1.174Menu: Special/Windows/Center . . . . .	48
1.175Menu: Special/Extend active win. . . . .	48
1.176Menu: Special/----- . . . . .	48
1.177Menu: Special/Case letter . . . . .	48
1.178Menu: Special/Case letter/Upper . . . . .	48
1.179Menu: Special/Case letter/Lower . . . . .	48
1.180Menu: Special/Case letter/Toggle . . . . .	48
1.181Menu: Special/Case block . . . . .	49
1.182Menu: Special/Case block/Upper . . . . .	49
1.183Menu: Special/Case block/Lower . . . . .	49
1.184Menu: Special/Case block/Swap . . . . .	49
1.185Menu: Special/----- . . . . .	49

---

---

1.186Menu: Special/ASCII . . . . .	49
1.187Menu: Special/ASCII/Show . . . . .	49
1.188Menu: Special/ASCII/Insert . . . . .	50
1.189Menu: Special/ASCII/Table . . . . .	50
1.190Menu: Special/----- . . . . .	50
1.191Menu: Special/Window . . . . .	50
1.192Menu: Special/Window/Close . . . . .	50
1.193Menu: Special/Window/Open . . . . .	50
1.194Menu: Special/Window/Swap . . . . .	51
1.195Menu: Special/----- . . . . .	51
1.196Menu: Special/Window next . . . . .	51
1.197Menu: Special/Window prev . . . . .	51
1.198Menu: Special/----- . . . . .	51
1.199Menu: Special/Iconify . . . . .	51
1.200Menu: Special/----- . . . . .	51
1.201Menu: Special/Split window . . . . .	51
1.202Menu: Special/----- . . . . .	52
1.203Menu: Special/Change filetype . . . . .	52
1.204Menu: Miscellaneous . . . . .	52
1.205Menu: Miscellaneous/DOS . . . . .	52
1.206Menu: Miscellaneous/DOS/Delete file . . . . .	52
1.207Menu: Miscellaneous/DOS/Rename file . . . . .	52
1.208Menu: Miscellaneous/DOS/----- . . . . .	52
1.209Menu: Miscellaneous/DOS/Create new dir . . . . .	52
1.210Menu: Miscellaneous/----- . . . . .	53
1.211Menu: Miscellaneous/Shell . . . . .	53
1.212Menu: Miscellaneous/Calculator . . . . .	53
1.213Menu: Miscellaneous/----- . . . . .	53
1.214Menu: Miscellaneous/Insert . . . . .	53
1.215Menu: Miscellaneous/Insert/Path . . . . .	53
1.216Menu: Miscellaneous/Insert/Date . . . . .	53
1.217Menu: Miscellaneous/Insert/Time . . . . .	54
1.218Menu: Miscellaneous/----- . . . . .	54
1.219Menu: Miscellaneous/Scan . . . . .	54
1.220Menu: Miscellaneous/Unfold . . . . .	54
1.221Menu: Miscellaneous/----- . . . . .	54
1.222Menu: Miscellaneous/Template . . . . .	54
1.223Menu: Miscellaneous/----- . . . . .	55
1.224Menu: Miscellaneous/Find reference . . . . .	55

---



---

1.225Menu: Miscellaneous/Ask for reference . . . . .	55
1.226Menu: Miscellaneous/----- . . . . .	56
1.227Menu: Miscellaneous/Find project refer. . . . .	56
1.228Menu: Miscellaneous/Ask for project ref. . . . .	56
1.229Menu: Miscellaneous/----- . . . . .	57
1.230Menu: Miscellaneous/File hunter . . . . .	57
1.231Menu: Miscellaneous/Ask for hunted file . . . . .	57
1.232Menu: Miscellaneous/----- . . . . .	58
1.233Menu: Miscellaneous/Complete . . . . .	58
1.234Menu: Miscellaneous/Autocase . . . . .	58
1.235Menu: Miscellaneous/----- . . . . .	58
1.236Menu: Miscellaneous/Project files . . . . .	58
1.237Menu: Miscellaneous/Project search . . . . .	59
1.238Menu: Macro . . . . .	59
1.239Menu: Macro/Edit macro . . . . .	59
1.240Menu: Macro/Save & run as macro . . . . .	59
1.241Menu: Macro/----- . . . . .	59
1.242Menu: Macro/Start recording . . . . .	59
1.243Menu: Macro/Stop recording . . . . .	60
1.244Menu: Macro/Continue recording . . . . .	60
1.245Menu: Macro/----- . . . . .	60
1.246Menu: Macro/Macro play . . . . .	60
1.247Menu: Macro/Macro play times... . . . .	60
1.248Menu: Macro/----- . . . . .	60
1.249Menu: Macro/Save macro . . . . .	60
1.250Menu: SAS/C . . . . .	60
1.251Menu: SAS/C/SAS/C options . . . . .	61
1.252Menu: SAS/C/Arguments . . . . .	61
1.253Menu: SAS/C/Create SMakeFile . . . . .	62
1.254Menu: SAS/C/----- . . . . .	62
1.255Menu: SAS/C/Build all . . . . .	62
1.256Menu: SAS/C/Build . . . . .	62
1.257Menu: SAS/C/Link . . . . .	62
1.258Menu: SAS/C/----- . . . . .	62
1.259Menu: SAS/C/Run WB . . . . .	62
1.260Menu: SAS/C/Esegui CLI . . . . .	63
1.261Menu: SAS/C/----- . . . . .	63
1.262Menu: SAS/C/First error . . . . .	63
1.263Menu: SAS/C/Next error . . . . .	63

---

---

1.264Menu: SAS/C/Last error . . . . .	63
1.265Menu: SAS/C/----- . . . . .	63
1.266Menu: SAS/C/Compile file . . . . .	63
1.267Menu: Preference . . . . .	63
1.268Menu: Preference/Screen . . . . .	64
1.269Menu: Preference/Screen/Mode . . . . .	64
1.270Menu: Preference/Screen/Colors . . . . .	64
1.271Menu: Preference/Screen/Fonts . . . . .	64
1.272Menu: Preference/Screen/DrawInfo . . . . .	65
1.273Menu: Preference/Menu . . . . .	66
1.274Menu: Preference/Keyboard . . . . .	66
1.275Menu: Preference/Mouse . . . . .	67
1.276Menu: Preference/Reference . . . . .	67
1.277Menu: Preference/Scanners . . . . .	67
1.278Menu: Preference/File hunter . . . . .	67
1.279Menu: Preference/API clients . . . . .	67
1.280Menu: Preference/Startup REXX macro . . . . .	68
1.281Menu: Preference/Miscellaneous (glb) . . . . .	68
1.282Menu: Preference/----- . . . . .	72
1.283Menu: Preference/Filetypes . . . . .	72
1.284Menu: Preference/----- . . . . .	73
1.285Menu: Preference/Keyword colors . . . . .	73
1.286Menu: Preference/Comment colors . . . . .	73
1.287Menu: Preference/Indentation . . . . .	74
1.288Menu: Preference/Templates . . . . .	75
1.289Menu:Preference/Dictionary . . . . .	75
1.290Menu: Preference/Fold function . . . . .	75
1.291Menu: Preference/Bracket . . . . .	76
1.292Menu: Preference/Saved files . . . . .	76
1.293Menu: Preference/Miscellaneous (ft) . . . . .	76
1.294Menu: Preference/----- . . . . .	79
1.295Menu: Preference/I/O . . . . .	79
1.296Menu: Preference/I/O/Load from... . . . .	79
1.297Menu: Preference/I/O/Save as... . . . .	79
1.298Menu: Preference/I/O----- . . . . .	80
1.299Menu: Preference/I/O/Load default . . . . .	80
1.300Menu: Preference/I/O/Save default . . . . .	80
1.301Menu: Preference/I/O----- . . . . .	80
1.302Menu: Preference/I/O/Use . . . . .	80

---

---

1.303	Adapt ProgED . . . . .	80
1.304	Registering . . . . .	81
1.305	Updating from 1.x . . . . .	81
1.306	Updating from 2.0 . . . . .	82
1.307	ARexx commands . . . . .	82
1.308	Info: ADDPROJECTFILES . . . . .	86
1.309	Info: ALTSTEP . . . . .	87
1.310	Info: AUTOARRANGEVER . . . . .	87
1.311	Info: AUTOARRANGEHOR . . . . .	87
1.312	Info: AUTORELOADPRJ . . . . .	87
1.313	Info: BLANKBORDERS . . . . .	87
1.314	Info: BOTTO MBORDER . . . . .	87
1.315	Info: BOTTOMEXTRASPACE . . . . .	87
1.316	Info: CLIPBOARD . . . . .	88
1.317	Info: EXTENDACTIVEMWIN . . . . .	88
1.318	Info: FASTLOAD . . . . .	88
1.319	Info: FASTMODE . . . . .	88
1.320	Info: FILE . . . . .	88
1.321	Info: FILETYPE LABEL . . . . .	89
1.322	Info: FILETYPE PATTERN . . . . .	89
1.323	Info: LEFTBORDER . . . . .	89
1.324	Info: LEFTEXTRASPACE . . . . .	89
1.325	Info: MAXCOL . . . . .	89
1.326	Info: MOUSEBLANKER . . . . .	89
1.327	Info: NUMFILES . . . . .	90
1.328	Info: NUMFILETYPES . . . . .	90
1.329	Info: NUMPROJECTFILES . . . . .	90
1.330	Info: NUMWINDOWS . . . . .	90
1.331	Info: PROJECTFILES . . . . .	90
1.332	Info: PUBSCREENNAME . . . . .	91
1.333	Info: RIGHTBORDER . . . . .	91
1.334	Info: RIGHTEXTRASPACE . . . . .	91
1.335	Info: SETDEF PUBSCREEN . . . . .	91
1.336	Info: SAFESAVE . . . . .	91
1.337	Info: SCREENADDRESS . . . . .	91
1.338	Info: SEPARATEITEMS . . . . .	92
1.339	Info: SHOWCURSORPOS . . . . .	92
1.340	Info: TOPBORDER . . . . .	92
1.341	Info: TOPEXTRASPACE . . . . .	92

---

---

1.342Info: UNFOLDONGOTOBYTE . . . . .	92
1.343Info: UNFOLDONGOTOLINE . . . . .	92
1.344Info: UNFOLDONGOTOMARKER . . . . .	92
1.345Info: UNFOLDONGOTOSCAN . . . . .	93
1.346Info: USEAPPICON . . . . .	93
1.347Info: USEAPPITEM . . . . .	93
1.348Info: USEBOTTOMGADGET . . . . .	93
1.349Info: USECLOCK . . . . .	93
1.350Info: USERIGHTGADGET . . . . .	93
1.351Info: USERTFILEREQ . . . . .	93
1.352Info: USERTFONTREQ . . . . .	94
1.353Info: USERTSCREENREQ . . . . .	94
1.354Info: USERTSYSREQ . . . . .	94
1.355Info: XPKLIB . . . . .	94
1.356Info: XPKMODE . . . . .	94
1.357Info: XPKPASSWORD . . . . .	94
1.358Info: AUTOFOLD . . . . .	94
1.359Info: AUTORELOAD . . . . .	95
1.360Info: BACKUP . . . . .	95
1.361Info: BACKUPDIR . . . . .	95
1.362Info: CHECKBRACKETS . . . . .	95
1.363Info: CREATEICONS . . . . .	95
1.364Info: ERASERIGHT . . . . .	95
1.365Info: INDENTJUMP . . . . .	95
1.366Info: LEFTMARGIN . . . . .	96
1.367Info: MARKERSDIR . . . . .	96
1.368Info: SHOWPATHNAME . . . . .	96
1.369Info: MINAUTOSAVE . . . . .	96
1.370Info: REFSDIR . . . . .	96
1.371Info: RIGHTMARGIN . . . . .	96
1.372Info: RIGHTWW . . . . .	97
1.373Info: SAVEMARKERS . . . . .	97
1.374Info: SAVEREFS . . . . .	97
1.375Info: SEPARATORSSTRING . . . . .	97
1.376Info: TAB . . . . .	97
1.377Info: TABEQSPACES . . . . .	97
1.378Info: USEAUTOCASE . . . . .	97
1.379Info: USEAUTOSAVE . . . . .	98
1.380Info: USECOMMENT1COLORS . . . . .	98

---

---

1.381Info: USECOMMENT2COLORS . . . . .	98
1.382Info: USEKEYWORDCOLORS . . . . .	98
1.383Info: USEKEYWORDINDENT . . . . .	98
1.384Info: USETEMPLATE . . . . .	98
1.385Info: WORDWRAPPING . . . . .	98
1.386Info: ABSCURSORLINE . . . . .	99
1.387Info: ABSNUMLINES . . . . .	99
1.388Info: CHANGES . . . . .	99
1.389Info: CURSORCOL . . . . .	99
1.390Info: CURSORLINE . . . . .	99
1.391Info: FILENAME . . . . .	99
1.392Info: FROZEN . . . . .	100
1.393Info: LINE . . . . .	100
1.394Info: LINECHARS . . . . .	100
1.395Info: MARKCOL . . . . .	100
1.396Info: MARKING . . . . .	100
1.397Info: MARKLINE . . . . .	100
1.398Info: MEMCURSORCOL . . . . .	101
1.399Info: MEMLINECHARS . . . . .	101
1.400Info: NUMLINES . . . . .	101
1.401Info: SIZE . . . . .	101
1.402Info: TYPEMODE . . . . .	101
1.403Info: WINDOWPOSX . . . . .	101
1.404Info: WINDOWPOSY . . . . .	102
1.405Internal commands . . . . .	102
1.406Internal command: ABOUT . . . . .	105
1.407Internal command: APICLIENTSPREFS . . . . .	106
1.408Internal command: APPEND . . . . .	106
1.409Internal command: ARRANGE . . . . .	106
1.410Internal command: ASCII . . . . .	106
1.411Internal command: BACK . . . . .	107
1.412Internal command: BLOCKCASE . . . . .	107
1.413Internal command: BRACKET . . . . .	107
1.414Internal command: CENTER . . . . .	107
1.415Internal command: CHARCASE . . . . .	108
1.416Internal command: CLEAR . . . . .	108
1.417Internal command: CLEARTEXT . . . . .	108
1.418Internal command: CLOSE . . . . .	108
1.419Internal command: COMMENTSCOLORSPREFS . . . . .	109

---

---

1.420	Internal command: COMPLETE . . . . .	109
1.421	Internal command: COPY . . . . .	109
1.422	Internal command: CR . . . . .	109
1.423	Internal command: CUT . . . . .	110
1.424	Internal command: DEL . . . . .	110
1.425	Internal command: DELFILE . . . . .	110
1.426	Internal command: DICTIONARYPREFS . . . . .	110
1.427	Internal command: DOWN . . . . .	111
1.428	Internal command: DRAWINFOPREFS . . . . .	111
1.429	Internal command: ENLARGE . . . . .	111
1.430	Internal command: FILETYPEPREFS . . . . .	111
1.431	Internal command: FILETYPE . . . . .	112
1.432	Internal command: FOLD . . . . .	112
1.433	Internal command: FOLDPREFS . . . . .	112
1.434	Internal command: FONTSPREFS . . . . .	112
1.435	Internal command: FREEZE . . . . .	113
1.436	Internal command: GOTO . . . . .	113
1.437	Internal command: GUI . . . . .	113
1.438	Internal command: HUNTER . . . . .	114
1.439	Internal command: HUNTERPREFS . . . . .	114
1.440	Internal command: ICONIFY . . . . .	114
1.441	Internal command: INDENTBLOCK . . . . .	115
1.442	Internal command: INDENTPREFS . . . . .	115
1.443	Internal command: INS . . . . .	115
1.444	Internal command: INSERT . . . . .	115
1.445	Internal command: JUMPTOBYTE . . . . .	116
1.446	Internal command: JUMPTOLINE . . . . .	116
1.447	Internal command: KEYBOARDPREFS . . . . .	116
1.448	Internal command: KWCOLORSPREFS . . . . .	117
1.449	Internal command: LAYOUTBLOCK . . . . .	117
1.450	Internal command: LEFT . . . . .	118
1.451	Internal command: LINE . . . . .	118
1.452	Internal command: LOAD . . . . .	118
1.453	Internal command: MACRO . . . . .	118
1.454	Internal command: MAKEDIR . . . . .	119
1.455	Internal command: MARK . . . . .	119
1.456	Internal command: MENU . . . . .	119
1.457	Internal command: MENUPREFS . . . . .	120
1.458	Internal command: MISC1PREFS . . . . .	120

---

---

1.459	Internal command: MISC2PREFS . . . . .	120
1.460	Internal command: MOUSE . . . . .	120
1.461	Internal command: MOUSEPREFS . . . . .	121
1.462	Internal command: MOVECURSOR . . . . .	121
1.463	Internal command: NEW . . . . .	121
1.464	Internal command: NEWCURDIR . . . . .	121
1.465	Internal command: NEWNAME . . . . .	121
1.466	Internal command: OPEN . . . . .	122
1.467	Internal command: PASTE . . . . .	122
1.468	Internal command: PREFS . . . . .	122
1.469	Internal command: PRJREFERENCEFIND . . . . .	123
1.470	Internal command: PROJECT . . . . .	123
1.471	Internal command: PROJECTSEARCH . . . . .	123
1.472	Internal command: QUIT . . . . .	124
1.473	Internal command: REDO . . . . .	124
1.474	Internal command: REFERENCEFIND . . . . .	124
1.475	Internal command: REFERENCEPREFS . . . . .	125
1.476	Internal command: RENAME . . . . .	125
1.477	Internal command: REPLACE . . . . .	125
1.478	Internal command: REVERT . . . . .	126
1.479	Internal command: REQUEST . . . . .	126
1.480	Internal command: RIGHT . . . . .	126
1.481	Internal command: SAVE . . . . .	126
1.482	Internal command: SAVEAS . . . . .	127
1.483	Internal command: SAVEBLOCKAS . . . . .	127
1.484	Internal command: SAVEDFILEPREFS . . . . .	127
1.485	Internal command: SCAN . . . . .	127
1.486	Internal command: SCANPREFS . . . . .	128
1.487	Internal command: SCREENCOLORSPREFS . . . . .	128
1.488	Internal command: SCREENMODEPREFS . . . . .	128
1.489	Internal command: SEARCH . . . . .	128
1.490	Internal command: SPLIT . . . . .	129
1.491	Internal command: STARTUPMACROPREFS . . . . .	129
1.492	Internal command: STORE . . . . .	129
1.493	Internal command: SWAP . . . . .	129
1.494	Internal command: TEMPLATEPREFS . . . . .	130
1.495	Internal command: TYPE . . . . .	130
1.496	Internal command: UNDO . . . . .	130
1.497	Internal command: UNFOLD . . . . .	130

---

---

1.498Internal command: UNFREEZE . . . . .	131
1.499Internal command: UNICONIFY . . . . .	131
1.500Internal command: UP . . . . .	131
1.501Internal command: WINDOW . . . . .	132
1.502Internal command: WSELECT . . . . .	132
1.503Internal command: ZIP . . . . .	132
1.504Edit program... . . . .	132

---



## Chapter 1

# PROGED.MANUAL

### 1.1 ProgED V2.1 ©1995-96 Giovanni Lambiase

```

#####                               ##### ←
      ##
    ##  ##
  ##  ## #####  #####  #####  #####  #####
##### ##  ## ##  ## ##  ## ## ##  ##  ##  ##
##      ##      ##  ## ##  ##      ##      ##  ##
##      ##      #####  #####  #####  #####
                                ##
                                ##  ##
                                ####

```

ProgED V2.1 - Programmers' Editor

Copyright ©1995-96 Giovanni Lambiase, June 1996

Reqtools.library - Copyright Nico François  
 Powerpacker.library - Copyright Nico François  
 XpkMaster.library - Copyright Urban Dominik Müller & Bryan Ford  
 Triton.library - Copyright Stefan Zeiger

\* \* S H A R E W A R E \* \*

Distribution

Requirements

Introduction

Features

Template/Tooltypes

Default settings

Arexx commands

Internal commands

Adapt ProgED

Collaboration

Registering

Updating from 1.x

Updating from 2.0

SAS/C Support

Known Bugs

Triton

Author

Thanks

```
+-----+
|
| NOTE: Excuse me for my English. Yes, I know, it's terrible,
|       but I had to translate it or you couldn't read it.
|       If you want correct this file and send it to me I'll
|       be very gratefull.
|
+-----+
```

## 1.2 Triton

Triton is a library from Stefan Zeiger. ProgED 2.1 requires it so, as Stefan wrote, I included a simple script to install a general release of this library for all 2.0 and above systems. Anyway Triton has a preferences editor and an optimized 3.0+ release. Search for the archive "TRI14USR.LHA" on your local bbs or Aminet to get full distribution. Triton.library is shareware but it's fully working even in unregistered release.

To help you I've implemented a ProgED internal snapshot of Triton windows (available only in registered Triton). As soon as you save preferences ProgED stores position & size of all Triton windows. This allows you to position & resize your windows and to get them even equal.

## 1.3 Thanks

A big thanks to following people:

---

```

Enrico Altavilla      \
Bruno Barbera        | Beta testers
Mario Bianchi        /

Alberto Longo        - For his useful suggests

Mario Mure'          - For all time he losed for me

Felice Murolo        - The (old) boss!

All registered users - (:-)

Bruno Barbera        \
Mario Bianchi        |
Enrico Altavilla     |
Luca Forlizzi        |
Paolo Menichetti     |
Celestino Amoroso    |
Costa Tsaousis       | For all suggests
Daniel Stenberg      |
Tim Stack            |
Matthias Moeller     |
Timo Westen          |
Johan Sandgren       |
Stefan Ruppert       |
Marcel Offermans     /

Stefan Zeiger        - For his Triton GUI

The Commodore        - For Amiga's born

The Escom            - For Amiga's re-born

The Viscorp (?)      - We'll see...

Claudio Baglioni     - For his music, which helped me during
                    all long programming sessions.

```

## 1.4 Author

Giovanni Lambiase  
 Via Capo Di Piazza , 6  
 84010 Minori (SA) - Italy

FOR FIDONET USERS: (Giovanni Lambiase) 2:335/253.1@fidonet

FOR INTERNET USERS: Giovanni.Lambiase@p1.f253.n335.z2.fidonet.org

PER AMIGANET USERS: (Giovanni Lambiase) 39:102/114.1@amiganet

These addresses belong to the BBS "3D On Line BBS". I'm the cossyp of this BBS. Anyway I've not a modem, so I can't pool regularly. Anyway you can use these addresses because I regularly go to sysop's home where I can copy all messages arrived. I'm sorry, but it's all I can do.

If you want register ProgED, please send me your e-mail addresses, too.

## 1.5 Known bugs

I don't know any bugs, at moment. If you find a bug send me a message reporting situation causing it. For my addresses see

Author  
. It's preferable use of e-mails, if you can.

## 1.6 SAS/C Support

Default settings was written so that SCMSG utility must close its window if there aren't errors, after a compilation. If you want that SCMSG doesn't close its window, then comment out the 'HIDE' command in the ARexx script:

```
"ProgED:SASC_Support/FirstError.script"
```

set "OpenSCMSG.rexx" as Post-iconify & Startup macro, and "CloseSCMSG.rexx" as Pre-iconify macro. It is also useful to reserve some space in screen to make room for the SCMSG window. If you want to do so, use the relative gadgets in preference window

```
Miscellaneous (glb)
```

```
.
```

At last, if you want to position the SCMSG window on screen, edit file "ProgED:SASC\_Support/SCMSG.options". This file contains SCMSG options used with ProgED.

## 1.7 Collaboration

If someone wishes to write a new interface menu with a ←  
compiler

(DICE, GNU, ...) or assembler (DEVPAC, ASM-ONE, ...) and he wants to contribute to ProgED development, he can send me his preference file and all relative Rexx macro. I'm also searching for foreign language catalogs (in "sources" drawer you can find the files "PED.cd" and "PED\_ita.ct"), preferably as CatComp files (.ct extension).

At last, since noone is perfect, I like receive your suggests and bugs-reports. To do this is preferably to use E-mail.

---

For my address see  
Author  
.

## 1.8 Distribution

### LICENCE

-----

This license applies to the product called 'ProgED', a program for the Amiga computer, published by Giovanni Lambiase under the concepts of SHAREWARE, and the accompanying documentation, example files and anything else that comes with the original distribution. The terms 'program' and 'ProgED' below, refer to this product. The licensee is addressed as 'you'.

- o You may copy and distribute verbatim copies of the program's executable code and documentation as you receive it, in any medium, provided that you conspicuously and appropriately publish only the original, unmodified program, with all copyright notices and disclaimers of warranty intact and including all the accompanying documentation, example files and anything else that came with the original.
  - o You may not copy and/or distribute this program without the accompanying documentation and other additional files that came with the original. You may not copy and/or distribute modified versions of this program.
  - o You may not copy, modify, sublicense, distribute or transfer the program except as expressly provided under this license. Any attempt otherwise to copy, modify, sublicense, distribute or transfer the program is void, and will automatically terminate your rights to use the program under this license. However, parties who have received copies, or rights to use copies, from you under this license will not have their licenses terminated so long as such parties remain in full compliance.
  - o By copying, distributing and/or using the program you indicate your acceptance of this license to do so, and all its terms and conditions.
  - o Each time you redistribute the program, the recipient automatically receives a license from the original licensor to copy, distribute and/or use the program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein.
  - o You may not disassemble, decompile, re-source or otherwise reverse engineer the program.
  - o You may use the programs for a period of up to 2 weeks for evaluation. After that, you have to register or quit using the program.
-

- o You agree to cease distributing the program and data involved if requested to do so by the author.
- o You may charge a fee to recover distribution costs. The fee for diskette distribution may not be more than the cost to obtain a public domain diskette from Fred Fish.
- o Above listed conditions not applies for file 'keyfile' obtained registering. You may not redistribute this file.

#### WARRANTIES

-----

THERE IS NO WARRANTY FOR THE PROGRAM, EITHER EXPRESSED OR IMPLIED.  
THE AUTHOR IS NOT RESPONSABLE FOR ANY DAMAGE YOU GET FROM THE PROGRAM.

USE IT AT YOUR OWN RISK!

## 1.9 Requirements

#### REQUIREMENTS

-----

ProgED needs O.S. 2.1, reqtools.library V38 (Copyright Nico François), a 68020 processor and 1Mb of RAM. Notwithstanding this I suggest a fast processor (68030-40-60), 2-4-6 Mb di memoria and O.S. 3.0 or above. A minimal Triton.library distribution is included in this archive. Read the related document to more informations about Triton.library registration.

Optionally you can use powerpacker.library (Copyright Nico François) and XPK libraries.

## 1.10 Introduction

#### INTRODUCTION

-----

ProgED is a new programmers editor inspired from two greit editors: CygnusEd and GoldED. In this new editor I have plugged in all I found useful for a programmer without useless frills. Somebody will found many features in common with CygnusED and GoldED. I did this choice to get an editor sufficiently flexible (like GoldED) and fast and simple like CygnusED.

Starting from 2.0 release I've added a new preferences system which allows you to define various "filetypes". This new system allows you to define different preferences for each type of file managed.

---

## 1.11 Features

### Features

- - Programmable keyword colors
    - Keyword indentation
    - Automatic phrase completion
    - Automatic case correction
    - Block folding
    - Programmable keyboard and menu
    - Integrated compiler frontend for SAS/C V6.x
    - Programmable quick function search
    - Templates for structure creation
    - Project management
    - Search for static references between files
    - Search for dynamic references between project files
    - File hunter
    - Integrated API (APplication Interface)
    - Autosave & Backup
    - Auto decrunch of PowerPacker crunched files
    - XPK libraries support
    - Fast mode for slow processors
    - On line macro creation
    - Window freeze/unfreeze
    - Iconify
    - Columnar block marking
    - Indentation & block layout
    - Multilevel Undo/Redo
    - Print texts on two colomns

- Handler to read files in memory
  - o
- Handler to read files in project
  - o
- Window splitting
  - o
- On line help
  - o
- File-types management
  - o
- Static markers management

## 1.12 Programmable keyword colors

ProgED allows user to program keywords of the used language and the relative colours used in rendering. This feature is, probably, the mainly cause pushed me to write ProgED. In fact, I couldn't think that no Amiga editor (except GoldED3) allows it, while ALL pc editors allow it!

Starting from 2.0 release the keywords coloration system is improved. Now you can use two "prefix" words (two words are "prefix" if the first word is a prefix of the second). Furthermore you can insert constructions containing spaces, such as "PTR TO LONG" of E language (thanks, Enrico!).

## 1.13 Keyword indentation

When you press RETURN ProgED analyses actual line. If it finds out a keyword you specified, it moves line at a lower, equal or greater indentation level. Furthermore cursor jumps to next line at a lower, equal or greater level.

## 1.14 Automatic phrase completion

ProgED stores a programmable phrases dictionary. A special command allows you to complete the phrase at left side of cursor and to expand it. Same dictionary allows ProgED to correct the case of letters in words written.

## 1.15 Automatic case correction

Using the same dictionary that allow to complete phrases, ProgED can correct case of letters in words written. The modification is accomplished when the cursor change line.

---



## 1.16 Block folding

Using an external function ProgED allows recognition and folding of text blocks. These external functions allow user to add new folder types. Function sources are included to allow adjustment to user preferences.

## 1.17 Programmable keyboard and menu

ProgED allows the TOTAL modification of keyboard and menu. User can modify the supplied configuration to adjust it to his/her preferences.

## 1.18 Integrated compiler frontend for SAS/C V6.x

The ProgED standard preferences allows creation of a special menu capable to interface ProgED and SAS/C. This menu supplies items useful to compile the actual project (using a SAS/C compiler V6.0 and above). Menu itself allows to modify the compilation options and to execute project (via WB or CLI).

## 1.19 Programmable quick function search

ProgED allows searches for a text line (AmigaGuide nodes, C functions, etc.) and jumping to wished line with a single mouse click. This research type is accomplished using external functions. User, thus, can modify (using included sources) or write own search function. This search functions will be called "Scanners".

## 1.20 Templates for structure creation

Templates allows user to get a complete structure typing only a configurable string. For example, using standard preferences, you can type "while{" to get a complete while cycle. Cursor, also, will be positioned ready to write cycle condition.

## 1.21 Project management

ProgED allows user to store and manage a project (declaring all files in project). Some external programs (executables or ARexx) can read project file list to compile project using your favourite compiler/assembler.

---

## 1.22 Search for static references between files

Static references allows searches between some files (user declared) for all symbols gotten via scanners. This references are called "static" because ProgED creates a map reference and saves it in a special file. So, your references must be static! (isn't it ? (-X ). If you want references between project files you must have a look at dynamic references.

## 1.23 Search for dynamic referemences between project files

Since static references can't be used for project management ProgED supplys the DYNAMIC references. ProgED allows to save, together with the text file, a special file (containing all references found in text file). This special files, thus, stores an updated map, at each save, of sources so that you get a valid references map for your project files.

## 1.24 File hunter

Using a special command ProgED allows to open a new window and to load in it a file. Path of this file will be "extracted" from text under cursor. This search type is useful especially to open immediately an "include" file.

## 1.25 Integreted API (APplication Interface)

ProgED can hook-up with external executables using an API (APplication Interface) message port. These executables are called "clients". Clients can open their windows on ProgED screen and, generally, add new features.

## 1.26 Autosave & Backup

At regular intervals ProgED can save, in a temporary file, text in windows. This allows, if your Amiga guru, to save work done. Furthermore, at each save, ProgED can create a backup file renaming old file.

## 1.27 Auto decrunch of PowerPacker crunched files

If you have powerpacker library (Copyright Nico François) and XpkMaster library (Copyright Urban Dominik Müller & Bryan Ford) in your LIBS: drawer, ProgED can decrunch automatically files crunched using these crunchers.

## 1.28 XPK libraries support

ProgED allows to save files using XPK libraries to crunch or encode data. At loading ProgED recognizes automatically crunched files. To use this feature you need XpkMaster library (Copyright Urban Dominik Müller & Bryan Ford) and some XPK libraries (I use xpkNUKE.library, it's greit! :- )).

## 1.29 Fast mode for slow processors

Using keyword colors you can slow down ProgED, especially using slow processors. If you have a slow Amiga or if you don't want keyword colors then you can use this mode. It disables colors but speeds up text refresh.

## 1.30 On line macro creation

ProgED, in its default configuration, creates a special menu dedicated to macros creation. You can create an ARexx macro simply doing what you want to insert in it. When you have finished macro can be saved in an ASCII file. You can, later, modify or use it.

## 1.31 Window freeze/unfreeze

ProgED can, eventually, freeze a window. Freeze a window means that you can close it but the relative text remains in memory. Using special commands ProgED can show a frozen window list and clicking in list you can reopen the window (unfreeze). A particular mode allows you to open, at same time, only a window. This feature is useful if you haven't a multiscan monitor (no interlaced screens).

## 1.32 Iconify

Using a hotkey (default ralt rshift return) you can close ProgED screen. It will show (if you want so) an AppIcon and an AppItem on WB screen. You can configure ProgED so that you can get AppIcon without AppItem, or vice versa or nothing. However you can open ProgED screen using the same hotkey used to iconify it.

## 1.33 Columnar block marking

ProgED can manage columnar blocks. All programmers needs this feature.

---

## 1.34 Indentation & block layout

If you want write docs you can use the indentation and block layout features of ProgED. It can format a block justifying it to left, right, center or full. Furthermore you can move text blocks to left, right or center to adjust wrong indentation.

## 1.35 Multilevel Undo/Redo

Each good editor needs this feature. Don't say me that you not know what it means (:-)

## 1.36 Print texts on two colomns

Using an external program ('PED\_Print') ProgED allows you to print your sources using one or two columns on ANY printer. Using two columns allows you to save paper.

## 1.37 Handler to read files in memory

ProgED can mount two particular handlers. One of them allows you to read, externally, files actually in memory using a new device (PED:). If you are not interesting in this feature then you can unmount PED device (move PED icon from Devs/DosDrivers to Storage/DosDrivers).

If you are using the demo release of ProgED then you can read ONLY the first 8192 bytes of texts loaded into memory.

NOTE: While you are reading a file using this handler the relative window is LOCKED (in title will appear the word 'LOCKED'). This is necessary because you'd modify "on the fly" text data. When a window is locked you can't select its menu. If you want to select a menu click on a window not locked or on ProgED screen background. Window will be unlocked when the external program terminates reading.

## 1.38 Handler to read files in project

2nd handler supplied together with ProgED, allows world to read files actually in project. This handler was done so that you can read a file in memory if it is actually in memory, or to read it from disk if it isn't actually in memory. Using this handler is possible to compile temporary projects modifying some files (without saving them) and to compile project. Modified file will be read from memory while others from disk. If the modification is right you can save, later, the modified files. As for first handler if you are not interested in this feature

---

you can move PRJ icon from Devs/DosDrivers to Storage/DosDrivers.

If you are using the demo release of ProgED then you can read ONLY the first 8192 bytes of texts loaded into memory.

NOTE: While you are reading a file using this handler the relative window is LOCKED (in title will appear the word 'LOCKED'). This is necessary because you'd modify "on the fly" text data. When a window is locked you can't select its menu. If you want to select a menu click on a window not locked or on ProgED screen background. Window will be unlocked when the external program terminates reading.

### 1.39 Window splitting

ProgED allows user to "split" a window, creating a new window containing same text. So, you can edit your text in both windows. This feature, furthermore, allows you to edit line 2367 of text while you are wiewing (on other window) the first text lines.

### 1.40 On line macro creation

If you move your mouse over a menu and press HELP key, ProgED will show you a brief help regarding selected menu. This feature is active only if you have Kickstart 3.0 or above.

### 1.41 File-types management

ProgED 2.0 has a new feature: major part of preferences, now, depend upon the "type" of file actually loaded. Other preferences remain "global", thus, they are equal for all files. The user can create wished "filetypes" specifying the AmigaDOS pattern which match the file-name. For each filetype you can change all related preferences. At file loading ProgED searches for the first filetype which match the name of the file and use it for the new window opened. When the user changes window ProgED switch to related preferences. So, writing in a C source, ProgED will use some preferences, while writing in a classic text it will use other preferences.

This is the major innovation of ProgED V2.0. I hope this new feature will be welcome.

### 1.42 Static markers management

From 2.0 release you can use the "static" markers. At internal command "STORE CURRENT" execution ProgED stores, in a special list, the current position of the cursor. Using the "GOTO STORED" command is

---

possible, later, to jump to a position stored previously in any file. The main feature of the static markers, anyway, is the possibility to save them together the file. This allows you, after a reset or a putting off, to reload all markers stored yesterday.

### 1.43 Template/Tooltypes

ProgED's template is the following:

```
FILE/M,  
CX_POPKEY/K,  
CX_PRIORITY/N,  
CX_POPUP/K,  
HANDLERPED/K,  
HANDLERPRJ/K,  
CONFIG/K,  
SCREEN/K
```

Same strings can be used as tooltypes if you start ProgED from ↵  
WB.

### 1.44 FILE/M,

Specify file(s) to load at start. If ProgED is already in memory new ProgED task will send to old ProgED task a special message signaling it to open your file(s).

NOTE: You can load ONLY one instance of ProgED.

### 1.45 CX\_POPKEY/K,

It specifies hotkey used to iconify and uniconify ProgED. Default is "rshift ralt return".

### 1.46 CX\_PRIORITY/K,

It specifies commodity priority. ProgED, in fact, can be used ↵  
as a  
commodity. For example you can insert it in WbStartup drawer to start  
it at each reset. Thus is useful the  
CX\_POPUP  
tooltype that allows  
you to start ProgED iconified.

## 1.47 CX\_POPUP/K,

It specifies if ProgED must open or not its windows at start.  
To open, later, its screen use relative hotkey (see  
CX\_POPKEY  
).

## 1.48 HANDLERPED/K,

It allows you to specify name of PED device. Default PED:. ↔  
Generally  
you need not this option, but you'll find it very useful if you would  
change handler's name. See also  
Handler to read files in memory  
.

## 1.49 HANDLERPRJ/K,

It allows you to specify name of PRJ device. Default PRJ:. ↔  
Generally  
you need not this option, but you'll find it very useful if you would  
change handler's name. See also  
Handler to read files in project  
.

## 1.50 CONFIG/K,

It allows you to specify initial config file. Generally ProgED  
use default file "ENV:PED/PED.prefs". Sometimes you'd change its  
behaviour to use special configurations.

## 1.51 SCREEN/K

It allows you to specify name of public screen opened from  
ProgED. Default "PED\_SCREEN".

## 1.52 Default settings

---

Since ProgED is fully configurable, supplied preferences are only one of the thousands possibly configurations. User can modify menu and keyboard to adjust them. In this doc, thus, I'll describe you only default settings. If you modified them you'll find, thus, some differences.

Menu:

```
Project
New
Open
Load
Save
Save as...
Save as... (XPK)
Save all changes
Save & close
Close
-----
Revert
Clear
Insert
Append
New name
-----
Current dir
-----
Print
-----
About
-----
Save all & quit
```

---



Quit

Edit

Mark

Mark columnar

Mark lines

Select all

-----

Cut

To std clip

To clip 1

To clip 2

To clip 3

Copy

To std clip

To clip 1

To clip 2

To clip 3

Paste

From std clip

From clip 1

From clip 2

From clip 3

Clear

-----

Cut line

Paste line

Swap line

Double line

Delete to EOL

-----

Undo

Redo

-----

Layout block

Left

Right

Center

Full

Move block

Left

Right

Center

Indent block

Save block as...

Search/Replace

Jump to line

Unfold

-----

Jump to byte

Unfold

-----

Set marker

1

2

3

4

5

6

7

8

9

10

Goto marker

1

2

3

4

5

6

7

8

9

10

Set marker

Jump to...

Unfold

-----

Search for...

Repeat search fwd

Repeat search bkw

-----

Replace

Repeat replace

-----

Matching bracket

---

Check bracket  
Special  
Fold  
Unfold  
Fold all  
Unfold all  
-----  
Windows  
Arrange ver  
Arrange hor  
Zip  
Enlarge  
Center  
Extend active win  
-----  
Case letter  
Upper  
Lower  
Toggle  
Case block  
Upper  
Lower  
Toggle  
-----  
ASCII  
Show  
Insert  
Table  
-----

---

Window

Close

Open

Swap

-----

Window next

Window prev

-----

Iconify

-----

Split window

-----

Change filetype

Misc

DOS

Delete file

Rename file

-----

Create new dir

-----

Shell

Calculator

-----

Insert

Path

Date

Time

-----

Scan

Unfold

-----

Template

-----

Find reference

Ask for reference

-----

Find project refer.

Ask for project ref.

-----

Hunt file

Ask for hunted file

-----

Complete

Autocase

-----

Project files

Project search

Macro

Edit macro

Save & run as macro

-----

Start recording

Stop recording

Continue recording

-----

Macro play

Macro play times...

---

-----

Macro save

SAS/C

SAS/C Options

Arguments

Create SMakeFile

-----

Build all

Make

Link

-----

Run WB

Run CLI

-----

First error

Next error

Last errore

-----

Compile file

Preferences

Screen

Mode

Colors

Fonts

Drawinfo

Menu

Keyboard

Mouse

---

Reference

Scanners

Hunter

Startup API clients

Startup REXX macro

Miscellaneous (glb)

-----

Filetypes

-----

Keyword colors

Comment colors

Indentation

Templates

Dictionary

Fold function

Brackets

Saved files

Miscellaneous (ft)

-----

I/O

Load from...

Save as...

-----

Load default

Save default

-----

Use

---



## 1.53 Menu: Project

This menu contains all classic operations you can apply on files: open new window, load, save, close window, reload file from disk, insert a file, print. Furthermore you can find here classic options as "About" to show infos, and "Quit" to exit.

## 1.54 Menu: Project/New

This menu opens a new window containing an empty text. It's useful to start a new text.

## 1.55 Menu: Project/Open

It shows a file-requester that you can use to open one or more files. To load more files multiselect them using shift key. For each selected file ProgED will open a new window containing relative text.

## 1.56 Menu: Project/Load

It shows a file-requester that you can use to load a file into actually active window. New file will substitute the old one.

## 1.57 Menu: Project/Save

It Saves on disk text contained in currently active window. ProgED will use the filename showed in window title bar.

## 1.58 Menu: Project/Save as...

It saves on disk text contained in currently active window. ProgED will open a file-requester to ask you for name to use.

## 1.59 Menu: Project/Save as... (XPK)

This menu operates as previous (see Save as... but this menu saves files using the XPK library chosen in Miscellaneous (glb) preference window.

## 1.60 Menu: Project/Save all changes

This menu saves all changed texts. Each text will be saved using filename showed on relative window title bar.

## 1.61 Menu: Project/Save & close

This menu operates as  
Save  
. Unique difference is that this  
menu closes the window after save.

## 1.62 Menu: Project/Close

It allows you to close a window freeing memory allocated for the text. You can, however, close the window using the Intuition close gadget. If you changed text then ProGED will ask you to confirm the operation.

## 1.63 Menu: Project/-----

## 1.64 Menu: Project/Revert

Selecting this menu the window will be refreshed reloading the text from disk. This option is useful to undo all operation done on text and to return to the original text on disk. If text was changed then ProGED will ask you to confirm the operation.

## 1.65 Menu: Project/Clear

It clears the active window, freeing the memory allocated for text. After this operation the window will be empty. If text was changed then ProGED will ask you to confirm the operation.

## 1.66 Menu: Project/Insert

It allows you to insert a file (at cursor position) in actually active window. You can specify filename of new text using a file requester.

---

### **1.67 Menu: Project/Append**

It allows you to append a file at end of window. You can specify filename of new text using a file-requester.

### **1.68 Menu: Project/New name**

It modifies filename of text loaded in currently active window. New name will be supplied using a file-requester.

### **1.69 Menu: Project/-----**

### **1.70 Menu: Project/Current dir**

It allows you to modify the current directory of ProGED. Generally it is the program directory ('ProGED:'), but sometimes you have to change it.

### **1.71 Menu: Project/-----**

### **1.72 Menu: Project/Print**

It starts external print program 'PED\_Print'. This program allows you to print your sources on one or two columns.

### **1.73 Menu: Project/-----**

### **1.74 Menu: Project/About**

Selecting this menu you will get a message containing the version number, my address and names for ARexx port, public screen and API port.

In same message you can identify your copy as demo or registered. If your copy is demo you'll get a message as 'UNREGISTERED VERSION', while if it is registered you will get a message as 'REGISTERED VERSION' followed by your name and serial number.

---

## 1.75 Menu: Project/-----

## 1.76 Menu: Project/Save all & quit

Selecting this menu ProgED will save all changed files. When it will finish it will quit.

## 1.77 Menu: Project/Quit

It quits ProgED.

## 1.78 Menu: Edit

This menu contains all options useful to edit text blocks: block marking, block cutting, block pasting, line cutting, etc.

## 1.79 Menu: Edit/Mark

Selecting this menu you can mark start of a text block. However, you can do same operation double clicking on the start position. Furthermore, if you want a "line" block you can hold down shift key. A "line" block is a block which includes all text from start line to end line.

Reselecting this menu you can unmark the block.

## 1.80 Menu: Edit/Mark columnar

This menu allows you to select a "columnar" block. A columnar block is a "rectangular" block. The rectangle extremes are the start position of the block and the cursor position. Columnar blocks are very useful sometimes. You can, furthermore, mark a columnar block double-clicking on block start position while holding down an ALT key.

## 1.81 Menu: Edit/Mark lines

Selecting this menu allows you to select a "line" block. They are text blocks which include all text from start line to end line. You can also mark a line block double clicking on start position while holding down shift key.

---

## 1.82 Menu: Edit/Select all

It selects all text in active window. It's equivalent to jump to start, to mark block start and to jump to end.

## 1.83 Menu: Edit/-----

## 1.84 Menu: Edit/Cut

Cut the text block currently selected and insert it in a clipboard. You can specify which clipboard have to be used using subitems:

Std clip

Clip 1

Clip 2

Clip 3

## 1.85 Menu: Edit/Cut/To std clip

It cuts out selected text block and inserts it in standard clipboard. You can specify standard clipboard number using preference's window

```
miscellaneous (glb)
. Default 0.
```

## 1.86 Menu: Edit/Cut/To clip 1

It cuts out selected text block and inserts it in clipboard #1.

## 1.87 Menu: Edit/Cut/To clip 2

It cuts out selected text block and inserts it in clipboard #2.

## 1.88 Menu: Edit/Cut/To clip 3

It cuts out selected text block and inserts it in clipboard #3.

---

## 1.89 Menu: Edit/Copy

It copys selected text block to a clipboard. You can specify wished clipboard using subitems:

Std clip

Clip 1

Clip 2

Clip 3

## 1.90 Menu: Edit/Copy/To std clip

It copys selected text block to a clipboard. Selecting this subitem you'll insert it in standard clipboard. You can change the standard clipboard using preference's window

miscellaneous (glb)

Default 0.

## 1.91 Menu: Edit/Copy/To clip 1

It copys selected text block to the clipboard # 1.

## 1.92 Menu: Edit/Copy/To clip 2

It copys selected text block to the clipboard # 2.

## 1.93 Menu: Edit/Copy/To clip 3

It copys selected text block to the clipboard # 3.

## 1.94 Menu: Edit/Insert

It inserts text block from a clipboard (choosed using subitems  $\leftrightarrow$ )  
)  
to actually active window at cursor position:

---

Std clip

Clip 1

Clip 2

Clip 3

### 1.95 Menu: Edit/Insert/From std clip

It inserts text block from standard clipboard to actually active window at cursor position. The standard clipboard is the clipboard specified in the preference's window  
Miscellaneous (glb)  
. Default 0.

NOTE: if text block was columnar it will be inserted in columnar mode, inserting, thus, as a rectangular block of text in the window.

### 1.96 Menu: Edit/Insert/From clip 1

It inserts text block from clipboard # 1 to the actually active window at cursor position.

NOTE: if text block was columnar it will be inserted in columnar mode, inserting, thus, as a rectangular block of text in the window.

### 1.97 Menu: Edit/Insert/From clip 2

It inserts text block from clipboard # 2 to the actually active window at cursor position.

NOTE: if text block was columnar it will be inserted in columnar mode, inserting, thus, as a rectangular block of text in the window.

### 1.98 Menu: Edit/Insert/From clip 3

It inserts text block from clipboard # 3 to the actually active window at cursor position.

NOTE: if text block was columnar it will be inserted in columnar mode, inserting, thus, as a rectangular block of text in the window.

### 1.99 Menu: Edit/Clear

It clears from window the actually selected text block. It will be NOT inserted in a clipboard. To copy a block to a clipboard see

Cut

.

### **1.100 Menu: Edit/-----**

#### **1.101 Menu: Edit/Cut line**

It cuts the cursor line and inserts it in a special internal clipboard. To insert this line around the text use

Insert line

.

#### **1.102 Menu: Edit/Insert line**

It inserts a text line from the line internal clipboard at cursor line.

To insert a line in this special clipboard use

Cut line

.

#### **1.103 Menu: Edit/Swap line**

It swaps cursor line & next line.

#### **1.104 Menu: Edit/Double line**

It creates a copy of the line cursor. This copy will be inserted next to cursor line itself.

#### **1.105 Menu: Edit/Delete to EOL**

Delete all chars at right side of the cursor.

---



### 1.106 Menu: Edit/-----

### 1.107 Menu: Edit/Undo

It does an UNDO operation. If you wrong something, selecting  $\leftrightarrow$  this menu you can get from the undo queue all previous data. To return to actual data you'll can select the menu

- Redo
- .

### 1.108 Menu: Edit/Redo

It does a REDO operation. If you did an undo operation and you  $\leftrightarrow$  wronged doing so, you can return to original data selecting this menu. To return to previous data select

- Undo
- menu.

### 1.109 Menu: Edit/-----

### 1.110 Menu: Edit/Layout block

It layouts actually selected block formatting it using left  $\leftrightarrow$  and right margin you'll find in preference's window

- Miscellaneous (ft)
- . Layout can

be done in four ways:

- Left

- Right

- Center

- Full

NOTE: Text block must be marked using the menu

- Mark lines

- .

### 1.111 Menu: Edit/Layout block/Left

---

It layouts actually selected block to fill in all lines. New ↔  
 lines  
 will be moved to fit (if possible) left side of text. ProgED will use  
 left and right margin specified in the preference's window

Miscellaneous (ft)

.

NOTE: Text block must be marked using the menu  
 Mark lines

.

### 1.112 Menu: Edit/Layout block/Right

It layouts actually selected block to fill in all lines. New ↔  
 lines  
 will be moved to fit (if possible) right side of text. ProgED will use  
 left and right margin specified in the preference's window

Miscellaneous (ft)

.

NOTE: Text block must be marked using the menu  
 Mark lines

.

### 1.113 Menu: Edit/Layout block/Center

It layouts actually selected block to fill in all lines. New ↔  
 lines  
 will be moved to center (if possible) of text. ProgED will use left and  
 right margin specified in the preference's window

Miscellaneous (ft)

.

NOTE: Text block must be marked using the menu  
 Mark lines

.

### 1.114 Menu: Edit/Layout block/Full

It layouts actually selected block to fill in all lines. New ↔  
 lines  
 will be moved to fit (if possible) whole text. ProgED will use left and  
 right margin specified in the preference's window

Miscellaneous (ft)

.

NOTE: Text block must be marked using the menu  
Mark lines

.

### 1.115 Menu: Edit/Move block

It moves each line of actually selected text block in three ←  
ways:

Left

Right

Center

NOTE: Text block must be marked using the menu  
Mark lines

.

### 1.116 Menu: Edit/Move block/Left

It moves each line of actually selected text block to left ←  
margin  
specified in preference's window  
Miscellaneous (ft)

.

NOTE: Text block must be marked using the menu  
Mark lines

.

### 1.117 Menu: Edit/Move block/Right

It moves each line of actually selected text block to right ←  
margin  
specified in preference's window  
Miscellaneous (ft)

.

NOTE: Text block must be marked using the menu  
Mark lines

.

### 1.118 Menu: Edit/Move block/Center

It moves each line of actually selected text block to center ↔  
it  
between left and right margins specified in preference's window

Miscellaneous (ft)

.

NOTE: Text block must be marked using the menu  
Mark lines

.

### 1.119 Menu: Edit/Indent block

It moves a text block one char (or more) to left (or right). ↔  
You'll  
can choose offset using a special dialog window.

NOTE: Text block must be marked using the menu  
Mark lines

.

### 1.120 Menu: Edit/Save block as...

It saves actually selected block. A file-requester allows you to  
specify filename to use.

### 1.121 Menu: Search&Replace

This menu contains all line/string search functions. It contains also  
the replace function.

### 1.122 Menu: Search&Replace/Jump to line

This menu allows you to jump to wished line specifying the line  
number. If line itself is in a fold you can tell ProGED to open  
automatically the fold or not. This is possible using next menu

Unfold

.

### 1.123 Menu: Search&Replace/Unfold

If this checkmark is enabled then, if you search for a line by  $\leftrightarrow$  number and line itself is in a fold (see Jump to line ), then ProgED will unfold automatically the fold to position cursor exactly. Otherwise ProgED will position cursor on the fold.

### 1.124 Menu: Search&Replace/-----

### 1.125 Menu: Search&Replace/Jump to byte

This menu allows you to jump to the wished byte in the text  $\leftrightarrow$  giving the byte number (from start of text). If it is in a fold you can tell ProgED to open automatically the fold. It's possible using next menu

Unfold

.

### 1.126 Menu: Search&Replace/Unfold

If this checkmark is enabled then, if you search for a byte by  $\leftrightarrow$  number and the byte itself is in a fold (see Jump to byte ), then ProgED will unfold automatically the fold to position cursor exactly. Otherwise ProgED will position cursor on the fold.

### 1.127 Menu: Search&Replace/-----

### 1.128 Menu: Search&Replace/Set marker

Using the relative subitem you can store in a marker the  $\leftrightarrow$  current cursor position. Using menu Goto marker you can jump to the stored position. Ten markers are available.

### 1.129 Menu: Search&Replace/Set marker/1

It stores actual cursor position in the marker # 1. Later, you ↔  
can  
jump to the stored position using menu  
Goto marker  
.

### 1.130 Menu: Search&Replace/Set marker/2

It stores actual cursor position in the marker # 2. Later, you ↔  
can  
jump to the stored position using menu  
Goto marker  
.

### 1.131 Menu: Search&Replace/Set marker/3

It stores actual cursor position in the marker # 3. Later, you ↔  
can  
jump to the stored position using menu  
Goto marker  
.

### 1.132 Menu: Search&Replace/Set marker/4

It stores actual cursor position in the marker # 4. Later, you ↔  
can  
jump to the stored position using menu  
Goto marker  
.

### 1.133 Menu: Search&Replace/Set marker/5

It stores actual cursor position in the marker # 5. Later, you ↔  
can  
jump to the stored position using menu  
Goto marker  
.

---

### 1.134 Menu: Search&Replace/Set marker/6

It stores actual cursor position in the marker # 6. Later, you ↔  
can  
jump to the stored position using menu  
Goto marker  
.

### 1.135 Menu: Search&Replace/Set marker/7

It stores actual cursor position in the marker # 7. Later, you ↔  
can  
jump to the stored position using menu  
Goto marker  
.

### 1.136 Menu: Search&Replace/Set marker/8

It stores actual cursor position in the marker # 8. Later, you ↔  
can  
jump to the stored position using menu  
Goto marker  
.

### 1.137 Menu: Search&Replace/Set marker/9

It stores actual cursor position in the marker # 9. Later, you ↔  
can  
jump to the stored position using menu  
Goto marker  
.

### 1.138 Menu: Search&Replace/Set marker/10

It stores actual cursor position in the marker # 10. Later, ↔  
you can  
jump to the stored position using menu  
Goto marker  
.

---

### 1.139 Menu: Search&Replace/Goto marker

It jumps to cursor position stored in a marker. You can choose marker using subitems. To store a cursor position use menu

```
Set marker
```

```
.
```

### 1.140 Menu: Search&Replace/Goto marker/1

It jumps to cursor position stored in marker #1. To store a  $\leftrightarrow$  cursor position see

```
Set marker
```

```
. If checkmark of next menu (
```

```
Unfold
```

```
) is
```

enabled and the stored position is closed in a fold, then ProgED will open the fold to position exactly. Otherwise ProgED will position cursor on fold itself.

### 1.141 Menu: Search&Replace/Goto marker/2

It jumps to cursor position stored in marker #2. To store a  $\leftrightarrow$  cursor position see

```
Set marker
```

```
. If checkmark of next menu (
```

```
Unfold
```

```
) is
```

enabled and the stored position is closed in a fold, then ProgED will open the fold to position exactly. Otherwise ProgED will position cursor on fold itself.

### 1.142 Menu: Search&Replace/Goto marker/3

It jumps to cursor position stored in marker #3. To store a  $\leftrightarrow$  cursor position see

```
Set marker
```

```
. If checkmark of next menu (
```

```
Unfold
```

```
) is
```

enabled and the stored position is closed in a fold, then ProgED will open the fold to position exactly. Otherwise ProgED will position cursor on fold itself.

---



**1.143 Menu: Search&Replace/Goto marker/4**

It jumps to cursor position stored in marker #4. To store a ↔  
 cursor

position see

```
Set marker
. If checkmark of next menu (
  Unfold
) is
```

enabled and the stored position is closed in a fold, then ProgED will open the fold to position exactly. Otherwise ProgED will position cursor on fold itself.

**1.144 Menu: Search&Replace/Goto marker/5**

It jumps to cursor position stored in marker #5. To store a ↔  
 cursor

position see

```
Set marker
. If checkmark of next menu (
  Unfold
) is
```

enabled and the stored position is closed in a fold, then ProgED will open the fold to position exactly. Otherwise ProgED will position cursor on fold itself.

**1.145 Menu: Search&Replace/Goto marker/6**

It jumps to cursor position stored in marker #6. To store a ↔  
 cursor

position see

```
Set marker
. If checkmark of next menu (
  Unfold
) is
```

enabled and the stored position is closed in a fold, then ProgED will open the fold to position exactly. Otherwise ProgED will position cursor on fold itself.

**1.146 Menu: Search&Replace/Goto marker/7**

It jumps to cursor position stored in marker #7. To store a ↔  
 cursor

position see

```
Set marker
. If checkmark of next menu (
```

Unfold  
) is

enabled and the stored position is closed in a fold, then ProgED will open the fold to position exactly. Otherwise ProgED will position cursor on fold itself.

### 1.147 Menu: Search&Replace/Goto marker/8

It jumps to cursor position stored in marker #8. To store a ↔  
cursor

position see

Set marker  
. If checkmark of next menu (  
Unfold  
) is

enabled and the stored position is closed in a fold, then ProgED will open the fold to position exactly. Otherwise ProgED will position cursor on fold itself.

### 1.148 Menu: Search&Replace/Goto marker/9

It jumps to cursor position stored in marker #9. To store a ↔  
cursor

position see

Set marker  
. If checkmark of next menu (  
Unfold  
) is

enabled and the stored position is closed in a fold, then ProgED will open the fold to position exactly. Otherwise ProgED will position cursor on fold itself.

### 1.149 Menu: Search&Replace/Goto marker/10

It jumps to cursor position stored in marker #10. To store a ↔  
cursor

position see

Set marker  
. If checkmark of next menu (  
Unfold  
) is

enabled and the stored position is closed in a fold, then ProgED will open the fold to position exactly. Otherwise ProgED will position cursor on fold itself.

### 1.150 Menu: Search&Replace/Set marker

Using this menu you can set a "static" marker which can be saved together the file itself.

### 1.151 Menu: Search&Replace/Jump to ...

Using this menu you can jump to a previously defined "static" marker. A special window will appear to let you choose which marker ProgED have to jump to.

### 1.152 Menu: Search&Replace/Unfold

If this checkmark is enabled and you are jumping to cursor ← position (stored in a marker) in a fold (see Goto marker ) ProgED will open the fold to position cursor exactly. Otherwise ProgED will position cursor on the fold.

### 1.153 Menu: Search&Replace/-----

### 1.154 Menu: Search&Replace/Search for...

Questo menù consente di ricercare determinate stringhe nel testo. A partire dalla versione 2.0, ProgED consente di utilizzare o meno il sistema "wildcards". Non utilizzando tale sistema ProgED cerca semplicemente la stringa data nel testo. Utilizzando il sistema

This menu allows you to search for strings in the text. From 2.0 release ProgED let you choose between a "wildcards" or a normal search. The normal search simply searches for given string in the text. The "wildcards" search allows you to search for an AmigaDOS pattern in the text. Using two checkboxes you can select a 'case sensitive' research and a reverse research. Generally ProgED start research from cursor position and go forward to end of text. Using reverse research you can start research from cursor position and go backward to begin of text.

Using AmigaDOS patterns allows you to search for a wide range of strings. For example the pattern:

```
struct#?amigaguide#?
```

allows you to search for the following strings:

```
struct AmigaGuideMsg
```

```

struct NewAmigaGuide
struct AmigaGuideHost

        etc. etc.

```

Using the cursor keys when the string gadget is enabled, you can get last strings written to reuse them.

### 1.155 Menu: Search&Replace/Repeat search fwd

Selecting this menu ProgED begins a new research. This new ←  
 research  
 will use all settings used last time you used menu  
 Search for...

It's useful to position cursor on next occurrence of AmigaDOS pattern without rewrite it.

### 1.156 Menu: Search&Replace/Repeat search bkw

Selecting this menu ProgED begins a new research. This new ←  
 research  
 will use all settings used last time you used menu  
 Search for...

It's useful to position cursor on previous occurrence of AmigaDOS pattern without rewrite it.

### 1.157 Menu: Search&Replace/-----

### 1.158 Menu: Search&Replace/Replace

This menu allows you to replace an AmigaDOS pattern for a ←  
 string.

Pattern and string are entered using a window like that used for the search window (see

```

Search for...
). When you start replace you

```

can have:

- 1) the pattern occurrence doesn't exists
- 2) the pattern occurrence exists

In former case ProgED will show you a message reporting search failure. In latter case ProgED will move cursor on occurrence and will show you a requester reporting following question: "Do you want to replace it ?", and the answers:

- a) Yes
- b) No
- c) All
- d) Last
- e) Stop

If you hit "Yes" ProgED will replace occurrence with the given string and will start search for next occurrence. If you hit "No" ProgED will start search for next occurrence without replacing previous occurrence. If you hit "All" ProgED will no ask you more. It will replace ALL occurrence so more found. If you hit "Last" ProgED will replace this occurrence and will stop. Hitting "Stop" ProgED will stop without replacing the previous occurrence.

### 1.159 Menu: Search&Replace/Repeat replace

ProgED will start a new replace research. ProgED will use all settings used last time you used menu  
 Replace  
 . It's useful because, so, you can start a repeat research without reentering pattern and string.

### 1.160 Menu: Search&Replace/-----

### 1.161 Menu: Search&Replace/Matching bracket

This menu should be selected only if cursor is on a bracket defined using preference window  
 Brackets  
 . If it's true then ProgED will search for matching bracket in text. If ProgED finds it then it will move cursor on matching bracket. Otherwise it will show a requester to notify you.

### 1.162 Menu: Search&Replace/Check bracket

This menu checks the cursor line to verify if number of opened bracket "(" is equal to number of closed bracket ")". If it is so, ProgED will do nothing, else it will show you a message.

---

### 1.163 Menu: Special

This menu contains special functions such as: fold, window management, iconify.

### 1.164 Menu: Special/Fold

Selecting this menu you'll activate selected external fold ↔  
function.

According to standard preferences you'll activate the C fold function. This function will fold the function containing cursor. Later you can see a single line on window showing ">FOLD" string followed from function name (string returned from external fold function).

To open a fold move cursor on it and select menu  
Unfold

.

### 1.165 Menu: Special/Unfold

This menu allows you to open a fold. To do so, move cursor on fold and select it. The fold itself will be opened and cursor will move to its position at fold time.

### 1.166 Menu: Special/Fold All

Using this menu you can fold all C functions in active window.

See also

Fold

.

### 1.167 Menu: Special/Unfold All

It opens all folds in active window. See also  
UnFold

.

### 1.168 Menu: Special/-----

### 1.169 Menu: Special/Window

This menu controls window sizing. You have five use ways:

Arrange ver

Arrange hor

Zip

Enlarge

Center

### 1.170 Menu: Special/Windows/Arrange ver

It arranges all opened windows vertically on screen. If  special

checkmark of menu

Extend active win.

is enabled then ProgED will

extend the active window to fit half screen. The remaining windows will share rest of screen.

### 1.171 Menu: Special/Windows/Arrange hor

It arranges all opened windows horizontally on screen. If  special

checkmark of menu

Extend active win.

is enabled then ProgED will

extend the active window to fit half screen. The remaining windows will share rest of screen.

### 1.172 Menu: Special/Windows/Zip

It zips a window. You can get same function using the Intuition Gadget.

### 1.173 Menu: Special/Windows/Enlarge

It enlarges active window to fit all screen area.

---

### 1.174 Menu: Special/Windows/Center

It centers active window on the screen.

### 1.175 Menu: Special/Extend active win.

If you arrange windows (see  
Arrange ver  
or  
Arrange hor  
and this checkmark is enabled then ProgED will extend the active  
window to fit half screen area.

### 1.176 Menu: Special/-----

### 1.177 Menu: Special/Case letter

This menu allows you to modify case of letter in text in three  
ways:

Upper

Lower

toggle

### 1.178 Menu: Special/Case letter/Upper

It modifies letter at cursor position so that it will be upper.  
If it was already upper ProgED will do nothing.

### 1.179 Menu: Special/Case letter/Lower

It modifies letter at cursor position so that it will be lower.  
If it was already lower ProgED will do nothing.

### 1.180 Menu: Special/Case letter/Toggle

It modifies letter at cursor position so that if it was lower  
it will be upper and vice versa.

---



### 1.181 Menu: Special/Case block

This menu allows you to modify case of letters in a text block ↔  
in  
three ways:

Upper

Lower

Swap

### 1.182 Menu: Special/Case block/Upper

All selected letter will be upper.

### 1.183 Menu: Special/Case block/Lower

All selected letter will be lower.

### 1.184 Menu: Special/Case block/Swap

It modifies letters in text block so that if it was lower it will be upper and vice versa.

### 1.185 Menu: Special/-----

### 1.186 Menu: Special/ASCII

This menu allows you some operation on ASCII characters:

Show

Insert

Table

### 1.187 Menu: Special/ASCII/Show

It shows ASCII code of char at cursor position.

---

### 1.188 Menu: Special/ASCII/Insert

It allows you to specify an ASCII code to insert in text (1-255).

### 1.189 Menu: Special/ASCII/Table

It shows a complete ASCII table. Clicking on a char you'll insert it in text.

### 1.190 Menu: Special/-----

### 1.191 Menu: Special/Window

This menu allows you to open and close windows on screen ↔  
without  
freeing their text.

Close

Open

Swap

### 1.192 Menu: Special/Window/Close

The actually active window will be close. The relative text ↔  
will  
be not freed. To reopen window use menu

Open

or

Swap

.

### 1.193 Menu: Special/Window/Open

This menu shows a special list showing all closed windows. To open any of them click on it.

---

**1.194 Menu: Special/Window/Swap**

The actually active window will be closed (see  
Close  
).

Later, ProgED will show you a special list containing all closed window. Clicking on any of them you can open it. This menu is very useful if you haven't a monitor, because so you can open only a window a time. At each swap ProgED will close actual window and will open another.

**1.195 Menu: Special/-----****1.196 Menu: Special/Window next**

It cycles on all window. At each selection ProgED will active next window. Using an hotkey you haven't to use mouse!

**1.197 Menu: Special/Window prev**

It cycles on all window. At each selection ProgED will active previous window. Using an hotkey you haven't to use mouse!

**1.198 Menu: Special/-----****1.199 Menu: Special/Iconify**

Selecting this menu ProgED will close all window and its ↵  
screen to  
iconify. To reopen ProgED you can use the commodity hotkey (default:  
rshift ralt return), AppIcon or AppItem on WB screen. AppIcon & AppItem  
must be activated using the preference window  
Miscellaneous (glb)  
.

**1.200 Menu: Special/-----****1.201 Menu: Special/Split window**

Using this menu ProgED will split actually active window. So, you'll get two views of same text. In each view you'll can edit text. Furthermore it's possible to split a splitted window so that you can have three views, four views, and so on, on same text.

### **1.202 Menu: Special/-----**

### **1.203 Menu: Special/Change filetype**

Using this menu you can change the filetype related to the text of the currently active window.

### **1.204 Menu: Miscellaneous**

In this menu you'll find all miscellaneous functions dedicated to file management, dynamic & static references, etc.

### **1.205 Menu: Miscellaneous/DOS**

Here you can do three basic DOS operation:

Delete file

Rename file

Create new dir

### **1.206 Menu: Miscellaneous/DOS/Delete file**

Selecting this menu you can delete a file. A file-requester allows you to select wished file.

### **1.207 Menu: Miscellaneous/DOS/Rename file**

Selecting this menu you can rename a file. A file-requester allows you to select wished file. A special windows allows you to enter new name of file.

### **1.208 Menu: Miscellaneous/DOS/-----**

### **1.209 Menu: Miscellaneous/DOS/Create new dir**

Selecting this menu you'll can create a new drawer. A file-requester will allow you to select wished name.

---

### 1.210 Menu: Miscellaneous/-----

### 1.211 Menu: Miscellaneous/Shell

It creates a new shell process. The relative window will be open on  
ProgED screen only if you enabled the 'Set DefPubScreen' flag in

Miscellaneous (glb)  
window.

### 1.212 Menu: Miscellaneous/Calculator

Execute "Calculator" program (supplied on WB3.0) on ProgED screen.

### 1.213 Menu: Miscellaneous/-----

### 1.214 Menu: Miscellaneous/Insert

Using this menu you can insert three objects in text:

Path

Date

Time

### 1.215 Menu: Miscellaneous/Insert/Path

Selecting this menu you can insert in text a file path (at cursor position). A file-requester will allow you to choose wished path.

### 1.216 Menu: Miscellaneous/Insert/Date

Selecting this menu you'll can insert actual date in text (at cursor position). Date will be write according to locale preferences of your system.

---

### 1.217 Menu: Miscellaneous/Insert/Time

This menu allows you to insert actual time in text (at cursor position).

### 1.218 Menu: Miscellaneous/-----

### 1.219 Menu: Miscellaneous/Scan

This menu is one of more important menu. Selecting it you'll get a window containing two lists. At right list you'll see all scanners actually resident (see Scanners), at left list you'll see a list of symbols. These symbols are computed using selected scanner on the right window. Changing scanner ProgED will recompute all symbols.

Using supplied scanners it's possible to get start of C functions, C structures (struct), AmigaGuide nodes etc. Selecting a symbol in the left list ProgED will position cursor exactly on scanner reported position (function start point, structure start point, node start).

Using next menu (see Unfold) you can specify if, at jump time, ProgED must, or not, open a fold, if any.

### 1.220 Menu: Miscellaneous/Unfold

This checkmark says ProgED if, at a cursor jump using scan function (see Scan), it have to open the fold found, if any.

### 1.221 Menu: Miscellaneous/-----

### 1.222 Menu: Miscellaneous/Template

This checkmark activates template's management (see Templates).

A template allows you to activate a ProgED internal command (see

ProgED internal commands  
 ) each time user use a defined key sequence.

Templates allow cycle construction (while,for,do-while,etc.) without effort!  
 Try typing "while{" in a window. While body will be expanded and the cursor  
 will be positioned ready to write cycle condition.

### 1.223 Menu: Miscellaneous/-----

### 1.224 Menu: Miscellaneous/Find reference

Using this menu you can find, in a few seconds, definition of  
 a static reference. A static reference is a reference (#defines,  
 C structures, etc.) independent from actual project. A reference,  
 thus, is simply the occurrence in a "static" file independent from  
 project and that, thus, change never. These files are, generally,  
 include files that change never, of course, but are useful because  
 they contains many definitions.

Symbols from "static" files are extracted using scanners  
 (see

Scanners

). Before use this menu you have to build a  
 references map (and store it in a file) using the special preference  
 menu

Reference

.

When you have a references map you can select this menu to  
 find any symbol written under cursor. Wished string, so, will be  
 extracted automatically from text at cursor position. It's possible  
 to specify any other string using the menu

Ask for reference

.

If reference was found, ProgED will open a new window containing  
 file with searched symbol and will move cursor at exact definition  
 point.

### 1.225 Menu: Miscellaneous/Ask for reference

Using this menu you can find, in a few seconds, definition  
 of a static reference. A static reference is a reference (#defines,  
 C structures, etc.) independent from actual project. A reference,  
 thus, is simply the occurrence in a "static" file independent from  
 project and that, thus, change never. These files are, generally,  
 include files that change never, of course, but are useful because  
 they contains many definitions.

Symbols from "static" files are extracted using scanners

---

(see

Scanners

). Before use this menu you have to build

a references map (and store it in a file) using the special preference menu

Reference

.

When you have a references map you can select this menu to find any symbol. You'll give ProGED an AmigaDOS pattern. It will search for all matches of pattern and symbol in references map.

If reference was found, ProGED will open a new window containing file with searched symbol and will move cursor at exact definition point.

## 1.226 Menu: Miscellaneous/-----

### 1.227 Menu: Miscellaneous/Find project refer.

Using this menu, in a few seconds, definition of a dynamic reference. A dynamic reference is a reference (#defines, C structures, etc.) dependent from actual project. A reference, thus, is definition of symbols contained in actual project.

Symbols from project files are extracted using scanners (see

Scanners

). To use this menu you MUST activate the checkmark

Save related refs

. Thus, at each save ProGED will write, together the text file, a new file (with extension ".refs"). This file will contain all symbol defined in text and found via scanners. IT'S VERY IMPORTANT TO REMEMBER THAT YOU MUST NOT DEACTIVATE THE CHECKMARK, SAVE A SOURCE AND REACTIVATE THE CHECKMARK, in fact, this way you'll have wrong references in the related file! If you like this feature activate checkmark and deactivate it NEVER! If this feature looks useless deactivate it.

At menu selection time, ProGED will extract from text under cursor the searched string. To supply any string see

Ask for project ref.

.

If ProGED found the reference it will open a new window containing text defined symbol (if it's not opened already) and it will move cursor on symbol definition.

### 1.228 Menu: Miscellaneous/Ask for project ref.



Using this menu, in a few seconds, definition of a dynamic reference. A dynamic reference is a reference (#defines, C structures, etc.) dependent from actual project. A reference, thus, is definition of symbols contained in actual project.

Symbols from project files are extracted using scanners (see

Scanners  
). To use this menu you MUST activate the checkmark

Save related refs

. Thus, at each save ProGED will write, together the text file, a new file (with extension ".refs"). This file will contain all symbol defined in text and found via scanners. IT'S VERY IMPORTANT TO REMEMBER THAT YOU MUST NOT DEACTIVATE THE CHECKMARK, SAVE A SOURCE AND REACTIVATE THE CHECKMARK, in fact, this way you'll have wrong references in the related file! If you like this feature activate checkmark and deactivate it NEVER! If this feature looks useless deactivate it.

At menu selection time ProGED will ask you for an AmigaDOS pattern. All symbols matching this pattern will be showed.

If ProGED found the reference it will open a new window containing text defined symbol (if it's not opened already) and it will move cursor on symbol definition.

## 1.229 Menu: Miscellaneous/-----

## 1.230 Menu: Miscellaneous/File hunter

This menu allows you to load, simply and immediately, wished ↔  
file.

At menu selection time ProGED will extract file name from text under cursor (it will search for <, >, ", ', etc.). File will be searched in a list of user defined paths (see

File hunter  
).

If file was found, it is loaded in memory.

To supply, via keyboard, wished filename use menu

Ask for hunted file  
.

## 1.231 Menu: Miscellaneous/Ask for hunted file

This menu allows you to load, simply and immediately, wished ↔  
file.

At menu selection time ProgED will ask you for wished file name.  
File will be searched in a list of user defined paths (see  
File hunter  
).

If file was found, it will be loaded in memory.

### 1.232 Menu: Miscellaneous/-----

### 1.233 Menu: Miscellaneous/Complete

ProgED allows you to use shortcuts while typing more used ↔  
phrases.

Insert in dictionary (see  
Dictionary  
) phrases more used. Later,  
write only few chars and select this menu. ProgED will search for  
first phrase matching your chars, and it will complete phrase for you.

### 1.234 Menu: Miscellaneous/Autocase

ProgED allows to use phrases contained in the dictionary  
(see

Dictionary  
) to correct automatically char case (how many  
times you wrote "Openlibrary" instead of "OpenLibrary" ?). Activating  
this checkmark ProgED will check for words in dictionary on the cursor  
line (each time cursor leave line). ProgED will correct wrong case, so.

### 1.235 Menu: Miscellaneous/-----

### 1.236 Menu: Miscellaneous/Project files

ProgED manage internally projects. Using this menu you can specify  
sources of your project. It's important to use this menu to say ProgED  
what you are doing. Use it also to open, quickly, a file in the actual  
project.

---

### 1.237 Menu: Miscellaneous/Project search

Selecting this menu you'll see a search window. This window allows you to search for an AmigaDOS pattern in ALL files of your project (how many times you want know where do you used "x" variable ?). When you press return ProgED will start search for pattern occurrences. During search ProgED will select all files where an occurrence was found. Click on "Edit" button to open them.

NOTE: During search ProgED will read files from memory, if they are loaded. Otherwise it will read them from disk.

### 1.238 Menu: Macro

This menu allows you to create and to use ARexx macros. Using it you can create a macro "on line" and save it for later use.

### 1.239 Menu: Macro/Edit macro

It allows, simply, to open a window loading an ARexx macro.

### 1.240 Menu: Macro/Save & run as macro

It saves text in active window and, later, executes it (from disk) as an ARexx macro.

### 1.241 Menu: Macro/-----

### 1.242 Menu: Macro/Start recording

It starts on-line recording of an ARexx macro. If ProgED have  
 a ←  
 previous recorded macro, ProgED ask you to confirm operation.  
 After recording start ProgED will follow all you do (menu select,  
 pressed keys, etc.). To stop recording use  
 Stop recording  
 .

NOTE: If a file-requester pops up ProgED will ask you if you want store file-requester contents or open file-requester during macro execution, later.

### 1.243 Menu: Macro/Stop recording

This menu stops macro recording. Later, if you want to ←  
continue  
recording you can restart recording selecting  
Continue recording  
.

### 1.244 Menu: Macro/Continue recording

If you recorded a macro but you want add more commands use this menu. It will continue recording, appending new commands to old macro.

### 1.245 Menu: Macro/-----

### 1.246 Menu: Macro/Macro play

Execute macro actually in memory. All operation recorded will be played back.

### 1.247 Menu: Macro/Macro play times...

Execute macro actually in memory more times. All operation recorded will be played back. At macro end ProgED will reexecute it. Before start you can specify number of play-back times.

### 1.248 Menu: Macro/-----

### 1.249 Menu: Macro/Save macro

It allows you to save macro actually in memory. This macro can be reused, later.

### 1.250 Menu: SAS/C

This menu supplies a practical interface between ProgED and SAS compiler (V6.0 and above). Using this menu you can build project, execute it or, if an error occurs, move cursor automatically on errors. If you haven't SAS compiler delete this menu.

To use this menu you must:

- 1) allow SAS utility windows (mainly SCMSG, but it's the only one) to open on ProGED screen. You can do so setting flag "SetDefPubScr" in preference window  
Miscellaneous (glb)  
.
- 2) have all project files in a single directory.
- 3) mount PRJ handler.
- 4) supply an executable name in the special gadget of scopts window.

All Arexx macro, scripts and executable used, are available in "SASC\_Support" drawer. Here you can found also sources of executable used.

This menu was written so that if a smakefile was given in project directory it will be used. Otherwise it will compile all files and link them together using the options written using scopts utility.

NOTE: I apologise, but if you want to add assembly sources in your project (and assembly them using asm SAS utility), you MUST create a smakefile using  
Create SNAkefile  
menu. Unfortunately this is due to SAS "asm" utility.

Write a simple asm source in 'RAM:foo.a'. Try to compile it using following command:

```
SC OBJNAME T: RAM:foo.a
```

You'll get an error (:-0). Why ? asm utility simply cats objname string (here, 'T:') and file name (here 'RAM:foo.a') getting 'T:RAM:foo.a'. What a silly bug, isn't it?  
If on your system you haven't this behaviour then you can assembly without using a SMakeFile.

## 1.251 Menu: SAS/C/SAS/C options

This menu opens the scopts window. Using it you can modify compile options regarding actually loaded project. Remember that you MUST supply an executable name!

## 1.252 Menu: SAS/C/Arguments

Selecting this menu you can specify executable arguments. ↔  
These  
arguments will be used if you select menu  
Run CLI  
.

### 1.253 Menu: SAS/C/Create SMakeFile

This menu creates, in project dir, a file for SMake utility. You can use this file as base for your makefile. Before to use this menu you have to say ProGED all files in project creating it.

### 1.254 Menu: SAS/C/-----

### 1.255 Menu: SAS/C/Build all

This menu executes a "build all" operation, rebuilding, thus, all objects containing in actual project and linking them to rebuild executable. If an error occurs cursor will be positioned on first error.

### 1.256 Menu: SAS/C/Build

This menu executes a "build" operation, rebuilding, thus, all objects modified from last compilation. Later, if necessary, linker will be called to create new executable.

### 1.257 Menu: SAS/C/Link

Selecting this menu you can call linker to rebuild your executable.

### 1.258 Menu: SAS/C/-----

### 1.259 Menu: SAS/C/Run WB

This menu allows you to run your project. Execution is made using WB procedures. ProGED will not use arguments specified using the special menu

Arguments

.

## 1.260 Menu: SAS/C/Esegui CLI

This menu allows you to run your project. It simply creates a new AmigaDOS process to run your executable as a CLI program. ProgED, here, will use arguments you specified in menu Arguments.

## 1.261 Menu: SAS/C/-----

## 1.262 Menu: SAS/C/First error

If, during a compilation, errors occurs, this menu allows you to position cursor on first error of list contained in scmsg utility window (it automatically will be opened).

## 1.263 Menu: SAS/C/Next error

If, during a compilation, errors occurs, this menu allows you to position cursor on next error of list contained in scmsg utility window (it automatically will be opened).

## 1.264 Menu: SAS/C/Last error

If, during a compilation, errors occurs, this menu allows you to position cursor on last error of list contained in scmsg utility window (it automatically will be opened).

## 1.265 Menu: SAS/C/-----

## 1.266 Menu: SAS/C/Compile file

This menu was added to compile small sources (without save them). Selecting it, ProgED will compile source contained in active window and will put relative object file (.o) in RAM:.. Using this menu you can avoid to use project management for small sources.

## 1.267 Menu: Preference

Menu Preference allows you to configure ProgED according your preferences.

---

## 1.268 Menu: Preference/Screen

This menu allows you to choose type and properties of ProgED screen. It uses four subitems:

Mode  
 Colors  
 Fonts  
 Drawinfo

## 1.269 Menu: Preference/Screen/Mode

Selecting this menu you can choose DisplayID of ProgED screen. If you are using `asl.library` (see `Miscellaneous (glb)`) you'll can

choose to open ProgED on a screen cloned from WB ("Same as Workbench") or open it on a given public screen. In latter case you must choose wished public screen selection the virtual DisplayID named "Open on:..." (followed by screen name). Note that if you open a WB clone screen you'll get a screen using same width,height and display-id. ProgED will NOT copy WB screen-depth.

NOTE: If you are using `ReqTools.library` you CANNOT choose these special modes. I apologise, but the `ReqTools.library` can't shows special display id. ):-(

## 1.270 Menu: Preference/Screen/Colors

Using this menu you'll can modify ProgED screen palette. Of course, this palette isn't used if you open ProgED on a public screen.

## 1.271 Menu: Preference/Screen/Fonts

Selecting this menu you can choose fonts used for ProgED screen. You can select:

a) Screen Font:

Used for screen bar and for all Intuition window opened on ProgED screen. This font will NOT be used if you open ProgED on a public screen (the public screen font will be used).



## b) Menu Font:

Used to render menu.

## c) Text Font:

Used to redraw texts in ProgED windows. This font MUST be size fixed.

## d) Gadget Font:

Used for all remaining ProgED window, and relative gadgets. This font should be tall 8 pixel. ProgED, in fact, doesn't scale windows to adjust them to used font. This menu give you only a possibility to adjust ProgED look at your preferences. So, DON'T USE A 24 PIXEL FONT !

## 1.272 Menu: Preference/Screen/DrawInfo

Selecting this menu you can choose all pens used to render ProgED screen. You can see, here, all AmigaDOS standard pens, and four special pens: ←

## a) Background pen:

Used to render background of ProgED screen.

## b) Paper pen:

Used to render background of text windows.

## c) Text pen:

Used to render text in ProgED windows.

## d) Mark pen:

Used to highlight marked blocks.

(b), (c) and (d) pens will be used ONLY you aren't using fast mode (see

Fast mode

). Otherwise pen (b) is 0, pen (c) is (1) and pen (d) is 2.

NOTA: Nella versione 1.1 sono presenti anche due penne chiamate "Penna cursore" e "Penna cursore (marcaggio)" non ancora implementate. Il loro scopo dovrebbe essere il cambiamento di colore del cursore.

NOTE: In release 1.1 you'll find two pens called "Cursor pen" and "Marking cursor pen". The aim should be the change of cursor color. THIS PENS AREN'T YET IMPLEMENTED.

### 1.273 Menu: Preference/Menu

Using this menu you can modify ProgED menu itself. BEWARE! Modifying menu you can loose this item. If you do so, you'll can never remodify a menu. This function, thus, must be used using brain.

In window opened at selection time, you'll see three lists. In left one you'll see all menu titles, in center one you'll see all item titles in menu actually selected in first list, in right one you'll see all subitems of item actually selected in center list. Using gadget relative at each list you can add a menu/item/subitem, delete an entry, or move an entry up or down.

Double clicking on any item or subitem you'll see a new window called

```

Edit program...
used to change action
executed at item/subitem selection.

```

A string gadget allows you to enter an hotkey used for each items/subitems (ProgED allows you to specify same char in two menu such that in one you use it lower, in other upper). If you are running ProgED on a 3.0 machine, it's possible to specify an entire string (not only a key) such as "F1". Unfortunately, so O.S. will not activate menu at F1 activation. You must, thus, modify keyboard prefs to create an hotkey for F1 key. I apologise, but 3.0 kick don't activate menu if you are using a string...

The gadget "CHECK" allows you to modify item/subitem actually selected so to create a checkmark. If you create a checkmark and you change checkmark using menu, you modify preferences flags. A checkmark selection, furthermore, allows you to execute a normal internal program. But generally you'll use only checkmarks without internal commands. To do so delete all commands in "Edit program" window.

At last, gadget "Separate menu" allows you to place a barlabel, each menu you left empty. Otherwise they will be showed as empty labels.

### 1.274 Menu: Preference/Keyboard

Using this menu you can modify keyboard behaviour. Clicking on a key in list ProgED will show you

```

Edit program...
that allows
you to modify action executed at each pressed key (or hotkey).

```

## 1.275 Menu: Preference/Mouse

Using this menu you can modify mouse behaviour. You can specify a program to be executed at each mouse button pressure (with qualifier or not). If you have a three buttons mouse please let me know if the management of middle button works because I couldn't test it. Thanks.

## 1.276 Menu: Preference/Reference

This menu allows you to create a file containing the static reference map. A special list contains all directory (or AmigaDOS patterns) of files/dirs that ProgED must read during map creation. A string gadget allows you to specify map filename. Clicking on "Create" gadget ProgED will read all dirs and files matching patterns, on each file readed ProgED will apply ALL possible scanners actually mounted (possible because a file must match scanner pattern or it will be not scanned), and all references found will be written on output file that, at end, will contain all references found applying scanners mounted.

NOTE: Due to high number of disk I/O this function will be completed in one or more minutes.

A special gadget allows you to load and keep reference map in RAM. I suggest you to set this gadget only if you have at least 6Mb.

## 1.277 Menu: Preference/Scanners

This menu allows you to insert or remove scanners used to search for symbols, in reference map creation and during a scan operation. A string gadget allows you to specify an AmigaDOS pattern. This pattern specifies, for each scanner, file that ProgED will scan. For example, a C-function scanner should use a pattern as "#?.c".

## 1.278 Menu: Preference/File hunter

ProgED can search for and load automatically files. To do so, However, you must say ProgED dirs to scan during research. Using window opened at selection time you can supply these directories. The research follows specified order.

## 1.279 Menu: Preference/API clients

Using window opened at selection time, you can specify filenames of all API clients you want run at ProgED execution. To activate a new client, thus, you must add it in the list, save preferences, quit and reload ProgED.

---

NOTE: Since an API client is runned as a CLI program, you can use this window to run, at each ProgED load, any CLI program.

## 1.280 Menu: Preference/Startup REXX macro

Using this menu you can specify REXX macro which ProgED must execute at:

- a) Start
- b) Before close its screen (iconify)
- c) After open its screen (uniconify)

## 1.281 Menu: Preference/Miscellaneous (glb)

This menu allows you to modify many flags of ProgED GLOBAL preferences. GLOBAL because the Miscellaneous (ft) changes misc flags of the active filetypes.

- 1) Clipboard number

number of standard clipboard used as default value during execution of internal command such as CUT,COPY,PASTE.

- 2) Max chars in line

Max bytes in line. You must supply an high value here (at least 300-400). If your line lenght is greater than this value you could get a GURU!

- 3) Alt step

If you use ALT during a scroll operation (ALT+cursor UP,etc) you'll jump over a number of lines/columns here specified.

- 4) Left scroll border

if cursor exceed this value (at left side) ProgED will scroll window.

- 5) Right scroll border

if cursor exceed this value (at right side) ProgED will scroll window.

- 6) Bottom scroll border

if cursor exceed this value (at bottom side) ProgED will scroll window.

- 7) Top scroll border

if cursor exceed this value (at top side) ProgED will scroll window.

---

## 8) Extra left

It specifies pixel number to be left at left side of screen. This option allows you to open other window on ProgED screen without overlapping all windows.

## 9) Extra Right

It specifies pixel number to be left at right side of screen.

## 10) Extra Top

It specifies pixel number to be left at top side of screen.

## 11) Extra bottom

It specifies pixel number to be left at bottom side of screen.

## 12) UNDO levels

It specifies UNDO queue length. Each new window opened will have this undo queue length (don't change opened windows!).

## 13) EOL char

It specifies the ascii value of the char used to render a CR just at right of lines. Generally you can use a value such as 1 to get a simply rectangle. Some fonts, on the contrary, let you to set value 13 and to see a small "CR", such as XCourier/13 (my favourite). Even the rendered char is dependent from used font. The value 0 disables EOL render.

## 14) % Arrange

It allows you to set the percentual of screen height used to show the active window, if the favourite mouse is active.

## 15) Use AppIcon

Activating this checkbox you say ProgED that, at iconify time, it have to create an AppIcon on WB screen. The icon file used is "ProgED:Icons/def\_iconify.info". Double clicking on this AppIcon will reopen ProgED screen.

## 16) Use AppItem

Activating this checkbox you say ProgED that, at iconify time, it must create an item in the 'Tools' menu of WB screen. This item allows you to reopen ProgED screen.

## 17) Ask filetype set

At load time ProgED must choose a filetype for loaded text. If this checkbox is active then ProgED will ask you if chosen filetype is correct or you want change it. Disabling this checkbox you'll get even the filetype chosen by ProgED.

---

## 18) Ask filetype change

When you change file name you could change filetype. So, if this checkbox is active, ProgED will ask you if you want switch to new filetype or keep old filetype. If it's disabled you'll get even the new filetype chosen by ProgED.

## 19) ReqTools system-reqs

It activates ReqTools.library system-requester. Otherwise ProgED will use Intuition EasyRequest.

## 20) ReqTools file-reqs

It activates ReqTools.library file-requester. Otherwise ProgED will use Asl.library requester.

## 21) ReqTools font-reqs

It activates ReqTools.library font-requester. Otherwise ProgED will use Asl.library requester.

## 22) ReqTools screen-reqs

It activates ReqTools.library screenmode-requester. Otherwise ProgED will use Asl.library requester.

## 23) Auto arrange (VER)

It arranges, if active, all window at each window operation, such that windows shares screen vertically.

## 24) Auto arrange (HOR)

It arranges, if active, all window at each window operation, such that windows shares screen horizontally.

## 25) Extend active win.

Using this flag ProgED will left most of screen height to active the window each time it arranges windows according to the percentual chosen.

## 26) Show cursor pos

If activated, in window title bar, ProgED will show you cursor position (column & line number). Line number will be showed considering folds and not considering folds.

NOTE: If you activate this option you can slow down window refresh.

## 27) Show path

It shows, if activated, the complete path of file in window title bar. Otherwise ProgED will show you only file name (without path).

---

## 28) Fast load

ProgED, generally, reads all files using a small I/O buffer. This, however, slows down I/O operation. To speed up I/O activate this checkbox. If so, ProgED will allocate a big buffer to load (in a single read) all file. I suggest you to use always this flag because if ProgED cannot allocate requested memory it will try loading using "slow" mode.

## 29) Safe save

System guru (or other) can trouble to you loosing you files. To avoid this activate this flag. So, ProgED save each file using a different file name. If save is right ProgED will delete old file and rename new. If save isn't right ProgED will delete new file and report failure. If your disk is about to be full you can, anyway, save your file, now!

## 30) Use Right gadget

It activates right scroller gadget on ProgED text windows.

## 31) Use bottom gadget

It activates bottom scroller gadget on ProgED text windows.

## 32) Auto reload

If you enable this checkbox, then ProgED, on quit, will save all opened files, and latest loaded project file name, in "ProgED:OldFiles". When you reload ProgED it will reload all files and project.

## 33) Use clock

It activates a clock on ProgED screen title bar.

## 34) Unfold (scan)

It enables or disables unfold during a scan search.

## 35) Unfold (marker)

It enables or disables unfold during a jump to marker.

## 36) Unfold (linea)

It enables or disables unfold during a jump to line.

## 37) Unfold (byte)

It enables or disables unfold during a jump to byte.

## 38) Blank border

It allows you to blank the ProgED screen borders.

---

## 39) Mouse blanker

This flag activate mouse blanker. It allows you to remove mouse pointer (mouse arrow) from screen when user start typing. At first mouse move, ProgED will bring back mouse pointer.

## 40) Set DefPubScreen

Enabling this flag you set ProgED screen as default public screen. So all windows will open on ProgED screen. Use it if you don't want to swap screens using lamiga+m.

## 41) FFFAAASSSTTT!!!

If you activate this mode, then ProgED will deactivate all color management to speed up windows refresh.

## 42) XPK packers

It allows you to choose a XPK library (you'll see here all XPK libraries in 'LIBS:compressors'). The choosed library will be used when you save a file using the special menu

Save as... (XPK)

.

## 43) XPK mode

It allows you to specify XPK library mode to be used in a

Save as... (XPK)  
operation.

## 44) XPK Password

It allows you to specify the XPK password. This password will be used if you save a file using an XPK encode library, or if you want to load an XPK encoded file.

## 1.282 Menu: Preference/-----

## 1.283 Menu: Preference/Filetypes

Using the window which is opened using this menu you can modify, create, delete or duplicate a filetype. Most of preferences flags of ProgED, starting from 2.0 release, are depending from filetype loaded in the window. A filetype is identified using the AmigaDOS pattern specified in this window. All texts which name matches this pattern belong to the filetype. The filetypes list is readed from top to bottom. First matching pattern specify the filetype of the text. If none of the pattern matches the filename ProgED uses an internal

---



filetype which is useful mainly for plain texts. If you want specify an own plain text filetype simply add a filetype AT THE END of the list which has pattern "#?". Doing so, all text will match the own created filetypes.

All item of the preferences menu below this item are depending from the the active filetype or, in other words, from the filetype which is related to the text contained in the currently active window. Each time you activate an item (below this!) you'll get preferences regarding the filetype related to the active window. if you want modify an other filetype simply click on the screen background. You'll get a window to let you choose the wished filetype to modify.

NOTE: The label of the filetype related to the text in windows is shown in the title window, just before the filename.

### **1.284 Menu: Preference/-----**

### **1.285 Menu: Preference/Keyword colors**

WARNING: The preferences shown if this window are related to the filetype of the active window! See Filetypes

Using this menu you can modify colors used to render some words. In this list ProgED will show you all keywords actually stored. At each word you can associate a color. Two string gadgets allows you to specify chars which enable and disable color management. An integer gadget allows you to specify, for each word in list, number of following words using same color. For example the word "struct" shows 1 in this gadget such that string "struct Library" will be rendered all using color specified for "struct" word.

NOTE: You can select ONLY colors between 1 and 127. So, DON'T SELECT COLOR 0 OR GREATER THAN 127 ! IT'S ILLEGAL !

### **1.286 Menu: Preference/Comment colors**

WARNING: The preferences shown if this window are related to the filetype of the active window! See Filetypes

Using this menu you can modify colors used in comment rendering. ProgED manages two comment types:

- a) From a string to end of line (C++ "//" )

- b) From a string to another string across lines  
(C `"/*"- "*/"`).

In window, opened at selection time, it's possible to specify comment start and stop strings and relative colors. An integer gadget allows you to specify max lines searched for a comment during refresh. If you put here 0, you want no limit. However, I suggest you to put here a normal limit (such us 30-100). If you put a high limit you force ProgED to search for comments in wide range of text. This can, thus, show down screen refresh (especially for long texts). A special flag "Color all line" allows you to color all line of a C++ `"/"` style comment. This allows you to color a quoted fidonet/internet message specifying the string `>` as C++ `"/"` style comment. Doing so all line will be colored.

## 1.287 Menu: Preference/Indentation

WARNING: The preferences shown if this window are related to the filetype of the active window! See  
Filetypes

Using indentation, each time you press return, ProgED analyses line just entered. If it ends with a string specified in this list, ProgED executes the action specified via mx gadgets. These actions are:

- a) `->`  
Cursor, on new line, will be moved to a higher indentation level.
- b) `<-`  
Cursor, on new line, will be moved to a lower indentation level.
- c) `>>>`  
Line just entered will be moved to a higher indentation level.
- d) `<<<`  
Line just entered will be moved to a lower indentation level.
- e) `>>> ->`  
Combines (c) and (a) effects.
- f) `<<< ->`  
Combines (d) and (a) effects.
- g) `>>> <-`  
Combines (c) and (b) effects.
-

h) <<< <-

Combines (d) and (b) effects.

Number of spaces of each indentation level can be specified using preference window

Miscellaneous (ft)

. Note that ProgED will substitute spaces for tabs, if possible.

## 1.288 Menu: Preference/Templates

WARNING: The preferences shown if this window are related to the filetype of the active window! See Filetypes

.

Window opened using this menu allows you to specify a template list. A "template" cause a ProgED program activation each time user type a key sequence. In window list you can insert these sequences. Double clicking on each sequence you'll see the Edit program... window.

Using this window you can specify a ProgED program to activate at sequence typing.

## 1.289 Menu:Preference/Dictionary

WARNING: The preferences shown if this window are related to the filetype of the active window! See Filetypes

.

Using this menu opened window, you can specify a word (or phrase) list. These words will be used in two case: phrase AutoCase and phrase Completing. In former case, if autocase is enabled, each time cursor moves to new line ProgED will check if on old line exists an occurrence of a word of dictionary and if it's true It will modify the word case to match word in dictionary. In latter case, at a special activation command, ProgED will complete phrase at left size of cursor using dictionary. Starting from 2.0 release ProgED will allow you to specify a set of enable/disable chars.

## 1.290 Menu: Preference/Fold function

WARNING: The preferences shown if this window are related to the filetype of the active window! See Filetypes

This menu allows you to choose external fold function to use in fold scan. Starting from 2.0 release ProgED allows you to enter two strings: start and end string of the fold block. Note that the use of this strings is a folder job so, if the folder doesn't use them, they aren't useful.

### 1.291 Menu: Preference/Bracket

WARNING: The preferences shown if this window are related to the filetype of the active window! See Filetypes

Selecting this menu ProgED will show you a list of brackets. All brackets here inserted will be searched any time you select menu

Matching bracket  
 . To do so, you must insert in list a string

such:

<start bracket>,<end bracket>

inserting, so, a comma between start bracket string and end bracket string. If you not insert a comma ProgED will not use this string.

### 1.292 Menu: Preference/Saved files

WARNING: The preferences shown if this window are related to the filetype of the active window! See Filetypes

This menu allows you to modify all protection bits to apply at save time. It's possible, furthermore, add a comment to files. A special gadget allows you to specify if ProgED must, or no, save an icon for each file. Icon used have to be in the "ProgED:Icons" drawer. Its name must be "def\_Icon.info".

### 1.293 Menu: Preference/Miscellaneous (ft)

WARNING: The preferences shown if this window are related to the filetype of the active window! See Filetypes

This menu allows you to modify other ProgED features:

1) Left margin

This integer gadget contains left margin using during block layout operations.

2) Right margin

This integer gadget contains right margin using during block layout operations.

3) TAB size

Size of tabs in characters (generally 8).

4) Indentation jump

Modify the indentation jump from level to level. This value is given in characters. You can supply any value here but I suggest you to put here same value as TAB size, so each indentation level will be a single TAB.

5) Erase right

At each return presson, ProgED will delete all spaces and tab forgotten at line end. This flag may trouble to you writing text (not programs). Disable it, if necessary.

6) Check ( )

If you activate this flag ProgED will check, each time cursor move to a new line, the brackets "(" and ")". Sometimes, however, this flag can trouble you...

7) Fold all on load

When ProgED ends loading a text, if this flag is enabled, then it will execute a "fold all" operation, closing all found folds.

8) Ins mode

Activating this gadget you set INS mode as starting mode (just after opening) in a window. If it's disabled you'll get OVR mode at window opening.

9) Tabs=Spaces

If you activate this checkbox as soon as you press TAB button you'll get spaces. This allows you to simulate TABs in a

text which CAN'T contain tabs (such as fidonet messages).

10) Backup on save

If you activate this flag you'll get, at each save, a backup file. This file (with ".bak" extension") will be created renaming old file.

11) Backup dir

If you left blank this string gadget ProgED will save backups in the same directory used to save the text. On the contrary, if you specify here a directory all backups will be written here. It's useful if you don't want see all backups if your project directory \*:-)

12) Save related refs

To keep updated project references map, ProgED must write, at each save, a new file with ".refs" extension. This checkbox enable refs-file save. If you don't want to use project references disable it. REMEMBER: if you disable this checkbox and you save a file, the references map will not be updated! To build a new map simply delete all ".refs" file. When needed ProgED will build all them.

13) Refs dir

It's job is equal to the "backup dir" checkbox. Of course it's used to specify the directory which must be used to write .refs files.

14) Save markers

Starting from 2.0 release ProgED allows you to save "static" markers together the text. This checkbox actives this feature.

15) Marker dir

It's job is equal to the "backup dir" checkbox. Of course it's used to specify the directory which must be used to write .mark files.

16) Activate autosave

Activate autosave. Autosaved files will have an ".auts" extension.

17) Autosave every (min.)

Specify how many minutes ProgED must wait between each autosave.

18) Word wrapping

This checkbox actives the word wrapping feature.

19) WW margin

---

This string gadget can be used to specify the right margin used by word wrapping engine. As soon as cursor reaches this margin the line is cutted.

#### 20) Separators string

Here you can specify all separators to be used. All of them will be used to identify words using "LEFT WORD" or "RIGHT WORD" internal command (CTRL+LEFT,CTRL+RIGHT keyboard equivalent).

You can specify ascii value comma separated, such as "0,8,13". You can also define an ascii value range using - char, such as "0-32,128-255". Using " char you can specify real chars, such as " ()|:,. ".

### 1.294 Menu: Preference/-----

### 1.295 Menu: Preference/I/O

This menu allows you to load and save ProgED preferences using following subitems:

```

Load from...
Save as...
-----
Load default
Save default
-----
Use

```

### 1.296 Menu: Preference/I/O/Load from...

It allows you to load a preference file suppling its file name. You'll choose wished file name using a file-requester.

### 1.297 Menu: Preference/I/O/Save as...

It allows you to save a preference file suppling file name. You'll choose wished file name using a file-requester.

---

### 1.298 Menu: Preference/I/O-----

### 1.299 Menu: Preference/I/O/Load default

It allows you to load the standard preference file. This is the "ENV:PED/PED.prefs" file.

### 1.300 Menu: Preference/I/O/Save default

It allows you to save the standard preference file. ProgED, so, will save preferences in "ENV:PED/PED.prefs" and "ENVARC:PED/PED.prefs" files.

### 1.301 Menu: Preference/I/O-----

### 1.302 Menu: Preference/I/O/Use

It allows you to save the standard preferences file. ProgED will save prefs in "ENV:PED/PED.prefs". Thus, this preferences will be used until next reset.

### 1.303 Adapt ProgED

#### Adapt ProgED

-----

ProgED philosophy, as for GoldED, allows user to modify completely the used menu and keyboard. In ProgED I tried to supply the same user-interface coming with CygnusED (before GoldED, the BEST amiga editor!). Many CygnusED users, thus, will find same menu shortcuts and managements. But if you want modify ProgED look, you have only to rewrite its preference file.

In 'icons' drawer you can save some icons used from ProgED in some cases. These are their names:

- o def\_Arexx: ARexx macro icons created by using ProgED.
- o def\_Icon: Icon supplied for saved text files.
- o def\_Iconify: AppIcon showed on WB screen at iconify time.
- o def\_Project: Icon for project files.

Furthermore, ProgED will come with three icons, but it not use them automatically:

- o def\_Client
  - o def\_FuncFolder
  - o def\_Scanner
-



ProgED don't use these icons, but they will come in ProgED distribution because, so, you will be able to apply them to new folders, scanners or clients.

At last ProgED allows you language localization. Now, I supplied only the internal English catalog and an Italian external catalog. If you want write a catalog in your language have a look in 'sources' drawer. Here you can find the 'PED.cd' file used in catalogs creation. If you write the relative 'PED.ct' file send me it and I'll use it in future ProgED release.

### 1.304 Registering

This ProgED release is a DEMO version. This demo CANNOT save files. Using this version you can value if you need it. From this release ProgED limits reading from PED and PRJ handlers. If your copy isn't registered you'll get a max of 8192 bytes from files loaded in a ProgED window.

To register your ProgED copy, send 20 US\$, or 33000 Italian lire, or 30DM, or equivalent via registered mail specifying that you want register your ProgED copy, your name and your address. You'll receive latest ProgED available release and the special file "keyfile". This file ables your demo version to save files. In keyfile I'll code your name & address. DO NOT DISTRIBUTE THIS FILE!

I'll accept only cash. Italian users can send me a postal money-order. I can't accept eurocheck & bank-draft (I haven't bank accounts!). However, specify ALWAYS your name, surname & address, and e-mail/fidonet addresses.

If you are a registered user you can upgrade your copy simply downloading new releases of ProgED and putting your keyfile in ProgED: drawer.

To get my address see  
 Author

```
***** !!! NOTE FOR ALL HACKERS !!! *****
*
*      IN THE INCLUDED EXECUTABLE THERE ISN'T ANY SAVE ROUTINE...
*
*****
```

### 1.305 Updating from 1.x

If you are a 1.0/1.1 registered user you haven't to register again your copy. Look into ProgED: drawer, you'll find the command UpdateKeyFile which can be used to get a new 2.0 keyfile starting from your 1.x keyfile.

Follow these steps:

- o MAKE A BACKUP OF YOUR OLD 1.x KEYFILE! #;-)
- o Use the command: UpdateKeyFile <old keyfile> <new keyfile>  
For example:  
  
UpdateKeyFile Keyfile Keyfile20
- o Copy new keyfile (Keyfile20) into ProgED: drawer
- o Rename it as "Keyfile"

That's all. If you have troubles updating your keyfile let me know (see Author you shouldn't have problems.

### 1.306 Updating from 2.0

If you are a 2.0 registered user you haven't to register again. Your keyfile will work for 2.1, too. Simply install ProgED 2.1 and use your 2.0 keyfile. During installation the script will rename your old preferences file, but you can use it for ProgED 2.1.

### 1.307 ARexx commands

ProgED have an ARexx port called "PED\_AREXX". Commands ProgED ← recognizes  
are equal to the Internal commands  
.

If you write an ARexx macro, however, you must lock events input from keyboard, menu and mouse. This is necessary to avoid user to modify macro behaviour. To do so, insert at macro start the command:

```
GUI LOCK
```

and at the end the command:

```
GUI UNLOCK
```

You must remember that, if you don't unlock your window, you'll can do no operation on window till the GUI UNLOCK command.

Furthermore, ProgED can recognize an additional ARexx command (not available in internal commands) called "QUERY". It allows you to obtain some infos regard ProgED status and actually active window status.

You may use QUERY command in following way:

QUERY <info>

where <info> is one of the following strings. Each string specifies an info type. The QUERY command, so, will return this value to the AREXX macro. You can ask for three types of info: global, filetype, and window dependent. The former regard ProGED global status, the latter regard active window status. Filetype infos regard all infos defined into filetypes. These infos are active window dependent, too.

Global infos:

ADDPROJECTFILES  
ALTSTEP  
AUTOARRANGEHOR  
AUTOARRANGEVER  
AUTORELOADPRJ  
BLANKBORDERS  
BOTTOMBORDER  
BOTTOMEXTRASPACE  
CLIPBOARD  
    EXTENDACTIVIEWINDOW  
FASTLOAD  
FASTMODE  
FILE  
FILETYPELABEL  
FILETYPEPATTERN  
LEFTBORDER  
LEFTEXTRASPACE  
MAXCOL  
MOUSEBLANKER  
NUMFILES  
NUMFILETYPES  
NUMPROJECTFILES  
NUMWINDOWS

---

---

PROJECTFILES  
PUBSCREENNAME  
RIGHTBORDER  
RIGHTEXTRASPACE  
SETDEFPUBSCREEN  
SAFESAVE  
SCREENADDRESS  
SEPARATEITEMS  
SHOWCURSORPOS  
SHOWPATHNAME  
TOPBORDER  
TOPEXTRASPACE  
UNFOLDONGOTOBYTE  
UNFOLDONGOTOLINE  
UNFOLDONGOTOMARKER  
UNFOLDONGOTOSCAN  
USEAPPICON  
USEAPPITEM  
USEBOTTOMGADGET  
USECLOCK  
USERIGHTGADGET  
USERTFILEREQ  
USERTFONTREQ  
USERTSCREENREQ  
USERTSYSREQ  
XPCLIB  
XPKMODE  
XPKPASSWORD  
Filetype infos:

---

AUTOFOLD  
AUTORELOAD  
BACKUP  
BACKUPDIR  
CHECKBRACKETS  
CREATEICONS  
ERASERIGHT  
INDENTJUMP  
LEFTMARGIN  
MARKERSDIR  
MINAUTOSAVE  
REFSDIR  
RIGHTMARGIN  
RIGHTWW  
SAVEMARKERS  
SAVEREFS  
SEPARATORSSTRING  
TAB  
TABEQSPACES  
USEAUTOCASE  
USEAUTOSAVE  
USECOMMENT1COLORS  
USECOMMENT2COLORS  
USEKEYWORDCOLORS  
USEKEYWORDINDENT  
USETEMPLATE  
WORDWRAPPING  
Active window infos:

---

ABSCURSORLINE  
ABSNUMLINES  
CHANGES  
CURSORCOL  
CURSORLINE  
FILENAME  
FROZEN  
LINE  
LINECHARS  
MARKCOL  
MARKING  
MARKLINE  
MEMCURSORCOL  
MEMLINECHARS  
NUMLINES  
SIZE  
TYPEMODE  
WINDOWPOSX  
WINDOWPOSY

### 1.308 Info: ADDPROJECTFILES

It returns a "char \*" array address. Each pointer in the  $\leftrightarrow$  array points to a filename in the actual project. This command is very useful to external C programs who want read the loaded project. Remember to use the NUMPROJECTFILES to get project file number.

Result type: integer

---

### 1.309 Info: ALTSTEP

It returns line number scroll at alt+cursor keys scroll.

Result type: integer

### 1.310 Info: AUTOARRANGEVER

It signals if the auto arrange (ver) mode is activated.

Result type: YES/NO

### 1.311 Info: AUTOARRANGEHOR

It signals if the auto arrange (hor) mode is activated.

Result type: YES/NO

### 1.312 Info: AUTORELOADPRJ

It signals if the auto reload mode of the project previously opened is activated.

Result type: YES/NO

### 1.313 Info: BLANKBORDERS

It signals if ProgED must blank screen borders.

Result type: YES/NO

### 1.314 Info: BOTTOMBORDER

It returns the bottom scroll border.

Result type: integer

### 1.315 Info: BOTTOMEXTRASPACE

It returns the bottom extra space that ProgED leaves on bottom side of screen at each window arrange operation.

Result type: integer

---

### 1.316 Info: CLIPBOARD

It returns standard clipboard number.

Result type: integer

### 1.317 Info: EXTENDACTIVIEWIN

It signals if the extend active window mode is activated.

Result type: YES/NO

### 1.318 Info: FASTLOAD

It signals if the fast load mode is activated.

Result type: YES/NO

### 1.319 Info: FASTMODE

It signals if the fast mode is activated.

Result type: YES/NO

### 1.320 Info: FILE

It shows up a simple asl file-requester. It's useful writing ARexx scripts. The syntax is:

```
QUERY FILE <req. title> <path> <filename> <pattern>
```

If <pattern> is an empty string the requester doesn't show a pattern gadget.

For example:

```
QUERY FILE "Choose a file" "ProgED:" "" "#?.c"
```

allows you to choose for a file matching #?.c in the directory ProgED: If you don't want show the pattern gadget substitute "" (empty string) for "#?.c".

Result type: Complete path name of choosen file. An empty string if user has selected "cancel".

---



### 1.321 Info: FILETYPELABEL

It returns the label related to the filetype of the text loaded in the active window.

Result type: string

### 1.322 Info: FILETYPEPATTERN

It returns the AmigaDOS pattern related to the filetype of the text loaded in the active window.

Result type: string

### 1.323 Info: LEFTBORDER

It returns the left scroll border.

Result type: integer

### 1.324 Info: LEFTEXTRASPACE

It returns the left extra space that ProgED leaves on left side of screen at each window arrange operation.

Result type: integer

### 1.325 Info: MAXCOL

It returns the max number of columns that ProgED have to use.

Result type: integer

### 1.326 Info: MOUSEBLANKER

It signals if the mouse blanker is activated.

Result type: YES/NO

---

### 1.327 Info: NUMFILES

It returns number of file actually loaded. Note that this ↔ doesn't include twin windows. So if you have a window and you slip it you'll get "1" using this QUERY. If you want to get the number of opened window see

```
NUMWINDOWS
```

.

Result type: integer

### 1.328 Info: NUMFILETYPES

It returns number of filetypes actually loaded in preferences.

Result type: integer

### 1.329 Info: NUMPROJECTFILES

It returns number of file actually present in project.

Result type: integer

### 1.330 Info: NUMWINDOWS

It returns number of windows actually opened on ProgED screen.

Result type: integer

### 1.331 Info: PROJECTFILES

This command must be used in following way:

```
QUERY PROJECTFILES <i>
```

It returns the *i*-th filename in actual project. If *<i>* is greather than number of files in project, it will return an empty string.

Result type: string

---

### 1.332 Info: PUBSCREENNAME

Returns ProgED screen name. If ProgED opened on a public screen you'll get public screen name. If you get an empty string, then ProgED is actually iconified.

Result type: string

### 1.333 Info: RIGHTBORDER

It returns the right scroll border.

Result type: integer

### 1.334 Info: RIGHTEXTRASPACE

It returns the right extra space that ProgED leaves on right side of screen at each window arrange operation.

Result type: integer

### 1.335 Info: SETDEFPUBSCREEN

It signals if ProgED must sets its screen as default public screen.

Result type: YES/NO

### 1.336 Info: SAFESAVE

It signals if the safe save mode is activated.

Result type: YES/NO

### 1.337 Info: SCREENADDRESS

Returns ProgED screen address. If ProgED opened on a public screen you'll get public screen address. If you get zero, then ProgED is actually iconified.

Result type: integer

---

### 1.338 Info: SEPARATEITEMS

It signals if ProGED will substitute barlabel for empty menu labels.

Result type: YES/NO

### 1.339 Info: SHOWCURSORPOS

It signals if ProGED are showing the cursor position on windows title bars.

Result type: YES/NO

### 1.340 Info: TOPBORDER

It returns the top scroll border.

Result type: integer

### 1.341 Info: TOPEXTRASPACE

It returns the top extra space that ProGED leaves on top side of screen at each window arrange operation.

Result type: integer

### 1.342 Info: UNFOLDONGOTOBYTE

It signals if ProGED must unfold a fold on a jump to byte.

Result type: YES/NO

### 1.343 Info: UNFOLDONGOTOLINE

It signals if ProGED must unfold a fold on a jump to line.

Result type: YES/NO

### 1.344 Info: UNFOLDONGOTOMARKER

It signals if ProGED must unfold a fold on a jump to marker.

Result type: YES/NO

---

### **1.345 Info: UNFOLDONGOTOSCAN**

It signals if ProgED must unfold a fold on a scan search.

Result type: YES/NO

### **1.346 Info: USEAPPICON**

It signals if ProgED will use AppIcon on WB screen at iconify time.

Result type: YES/NO

### **1.347 Info: USEAPPITEM**

It signals if ProgED will use AppItem on WB screen at iconify time.

Result type: YES/NO

### **1.348 Info: USEBOTTOMGADGET**

It signals if ProgED are using the bottom scroller gadget for its windows.

Result type: YES/NO

### **1.349 Info: USECLOCK**

It signals if the clock window is activated.

Result type: YES/NO

### **1.350 Info: USERIGHTGADGET**

It signals if ProgED are using the right scroller gadget for its windows.

Result type: YES/NO

### **1.351 Info: USERTFILEREQ**

It signals if ProgED are using the reqtools file-requesters.

Result type: YES/NO

---

### 1.352 Info: USERTFONTREQ

It signals if ProGED are using the reqtools font-requesters.

Result type: YES/NO

### 1.353 Info: USERTSCREENREQ

It signals if ProGED are using the reqtools screenmode-requesters.

Result type: YES/NO

### 1.354 Info: USERTSYSREQ

It signals if ProGED are using the reqtools system-requesters.

Result type: YES/NO

### 1.355 Info: XPKLIB

It returns the used XPK library name.

Result type: string

### 1.356 Info: XPKMODE

It returns the XPK crunch library mode (0-100).

Result type: integer

### 1.357 Info: XPKPASSWORD

It returns the password used during XPK library encode/decode.

Result type: string

### 1.358 Info: AUTOFOLD

It signals if auto fold-all mode is activated.

Result type: YES/NO

---

### 1.359 Info: AUTORELOAD

It signals if the auto reload mode is activated.

Result type: YES/NO

### 1.360 Info: BACKUP

It signals if the backup mode is activated.

Result type: YES/NO

### 1.361 Info: BACKUPDIR

It returns the directory used to save backups. If it's an empty string ProGED saves backups into same directory used to save the text.

Tipo di risultato: string

### 1.362 Info: CHECKBRACKETS

It signals if the automatic brackets "(""," check is activated.

Result type: YES/NO

### 1.363 Info: CREATEICONS

It signals if ProGED will save icons at each text save.

Result type: YES/NO

### 1.364 Info: ERASERIGHT

It signals if the erase right mode is activated.

Result type: YES/NO

### 1.365 Info: INDENTJUMP

It returns the indentation jump (column difference between levels).

Result type: integer

---

### 1.366 Info: LEFTMARGIN

It returns the left margin that ProGED will use during text block layout operation.

Result type: integer

### 1.367 Info: MARKERSDIR

It returns the directory used to save static markers files. If it's an empty string ProGED saves markers files into same directory used to save the text.

Tipo di risultato: string

### 1.368 Info: SHOWPATHNAME

It signals if ProGED are showing the paths on window titles bars.

Result type: YES/NO

### 1.369 Info: MINAUTOSAVE

It returns the autosave interval between autosaves.

Result type: integer

### 1.370 Info: REFSDIR

It returns the directory used to save referements files. If it's an empty string ProGED saves refs files into same directory used to save the text.

Tipo di risultato: string

### 1.371 Info: RIGHTMARGIN

It returns the right margin that ProGED will use during text block layout operation.

Result type: integer

---



**1.372 Info: RIGHTWW**

It returns the wordwrapping right border.

Result type: integer

**1.373 Info: SAVEMARKERS**

It returns YES if ProGED saves markers together texts.

Result type: YES/NO

**1.374 Info: SAVEREFS**

It returns YES if ProGED saves referements file together texts.

Result type: YES/NO

**1.375 Info: SEPARATORSSTRING**

It returns a string containing all used defined separators.

Result type: string

**1.376 Info: TAB**

It returns tab size in characters.

Result type: integer

**1.377 Info: TABEQSPACES**

It signals if the "tab=spaces" mode is activated.

Result type: YES/NO

**1.378 Info: USEAUTOCASE**

It signals if the autocase mode is activated.

Result type: YES/NO

---

**1.379 Info: USEAUTOSAVE**

It signals if the autosave mode is activated.

Result type: YES/NO

**1.380 Info: USECOMMENT1COLORS**

It signals if the multiple lines comment color management is activated.

Result type: YES/NO

**1.381 Info: USECOMMENT2COLORS**

It signals if the single line comment color management is activated.

Result type: YES/NO

**1.382 Info: USEKEYWORDCOLORS**

It signals if the keyword color management is activated.

Result type: YES/NO

**1.383 Info: USEKEYWORDINDENT**

It signals if indentation mode is activated.

Result type: YES/NO

**1.384 Info: USETEMPLATE**

It signals if the templates are enabled.

Result type: YES/NO

**1.385 Info: WORDWRAPPING**

It signals if the "wordwrapping" mode is activated.

Tipo di risultato: YES/NO

---

**1.386 Info: ABSCURSORLINE**

It returns the ABSOLUTE cursor line column number. This value is computed considering all folded lines, too!

Result type: integer

**1.387 Info: ABSNUMLINES**

It returns the ABSOLUTE number of lines in the text. This value is computed considering folded lines, too!

Result type: integer

**1.388 Info: CHANGES**

It returns the number of changes from last save.

Result type: integer

**1.389 Info: CURSORCOL**

It returns the cursor column number.

Result type: integer

**1.390 Info: CURSORLINE**

It returns the cursor line number.

Result type: integer

**1.391 Info: FILENAME**

It returns the complete file path of file actually loaded in the active window.

Result type: string

---

### 1.392 Info: FROZEN

It signals if the active window is locked due to external reads via handlers.

Result type: YES/NO

### 1.393 Info: LINE

It returns the cursor line.

Result type: string

### 1.394 Info: LINECHARS

It returns the column number of the cursor line.

Result type: integer

### 1.395 Info: MARKCOL

It returns the start column number of marked block. You'd use this only if exists a marked block!

Result type: integer

### 1.396 Info: MARKING

It signals if you have marked start of a block.

Result type: YES/NO

### 1.397 Info: MARKLINE

It returns the start line number of marked block. You'd use this only if exists a marked block!

Result type: integer

---

### 1.398 Info: MEMCURSORCOL

It returns the column number where cursor is. This number, however, don't count tab, so each tab is computed as a single char. Thus, this values is simply number of bytes at left side of cursor.

Result type: integer

### 1.399 Info: MEMLINECHARS

It returns the line bytes lenght. Computing this value you must remember that a tab is a single byte!

Result type: integer

### 1.400 Info: NUMLINES

It returns the number of lines in the text.

Result type: integer

### 1.401 Info: SIZE

It returns the text size (in bytes).

Result type: integer

### 1.402 Info: TYPEMODE

It signals if you are using insert or overwrite mode.

Result type: INS/OVW

### 1.403 Info: WINDOWPOSX

It returns the X-position (in characters) of char at upper left corner of window inside text.

Result type: integer

---

## 1.404 Info: WINDOWPOSY

It returns the Y-position (in characters) of char at upper left corner of window inside text.

Result type: integer

## 1.405 Internal commands

ProgED is highly configurable due to its internal commands. ↔

The

internal commands are commands that ProgED executes each time user selects a given menu or press a hotkey. During menu and keyboard customizing, when ProgED shows you the

Edit program...

window, user

can "program" a sequence of internal commands or Shell/Arexx external programs. Internal commands supply all internal capabilities.

NOTE: Each command can be specified in many ways. In the following list you'll see commands and relative use mode.

ABOUT

APICLIENTSPREFS

APPEND

ARRANGE

ASCII

BACK

BLOCKCASE

BRACKET

CENTER

CHARCASE

CLEAR

CLEARTEXT

CLOSE

COMMENTSCOLORSPREFS

COMPLETE

COPY

---

---

CR  
CUT  
DEL  
DELFILE  
DICTIONARYPREFS  
DOWN  
DRAWINFOPREFS  
ENLARGE  
FILETYPEPREFS  
FILETYPE  
FOLD  
FOLDPREFS  
FONTSPREFS  
FREEZE  
GOTO  
GUI  
HUNTER  
HUNTERPREFS  
ICONIFY  
INDENTBLOCK  
INDENTPREFS  
INS  
INSERT  
JUMPTOBYTE  
JUMPTOLINE  
KEYBOARDPREFS  
KWCOLORSPREFS  
LAYOUTBLOCK

---

---

LEFT  
LINE  
LOAD  
MACRO  
MAKEDIR  
MARK  
MENU  
MENUPREFS  
MISC1PREFS  
MISC2PREFS  
MOUSE  
MOUSEPREFS  
MOVECURSOR  
NEW  
NEWCURDIR  
NEWNAME  
OPEN  
PASTE  
PREFS  
    PRJREFERENCEFIND  
PROJECT  
PROJECTSEARCH  
QUIT  
REDO  
REFERENCEFIND  
REFERENCEPREFS  
RENAME  
REPLACE  
REVERT

---



REQUEST  
RIGHT  
SAVE  
SAVEAS  
SAVEBLOCKAS  
SAVEDFILEPREFS  
SCAN  
SCANPREFS  
SCREENCOLORSPREFS  
SCREENMODEPREFS  
SEARCH  
SPLIT  
STARTUPMACROPREFS  
STORE  
SWAP  
TEMPLATEPREFS  
TYPE  
UNDO  
UNFOLD  
UNFREEZE  
UNICONIFY  
UP  
WINDOW  
WSELECT  
ZIP

#### **1.406 Internal command: ABOUT**

ABOUT command haven't arguments. It shows classic info window containing, however, user name (if you own a registered copy).

---

- a) ABOUT

### 1.407 Internal command: APICLIENTSPREFS

This command must be given using the ASK option:

```
APICLIENTSPREFS ASK
```

It shows the preference window regarding API clients.

See

```
API clients
```

.

### 1.408 Internal command: APPEND

APPEND allows you to append to active window text a new file. This command can be given in two ways:

- a) APPEND ASK
- b) APPEND FILE <name>

In former case ProGED shows a file-requester to allow user to select wished file. In latter case ProGED append given file (using <name> as file name).

### 1.409 Internal command: ARRANGE

ARRANGE allows you to reorganize all windows on screen in two different ways:

- a) ARRANGE HOR [FAVOURITE]
- b) ARRANGE VER [FAVOURITE]

In former case all windows will be arranged horizontally on screen, in latter case they will be arranged vertically. The FAVOURITE flag allows you to specify if ProGED must, or not, favourite the active window giving it half screen. If you don't specify it then ProGED will use value given in actual preferences.

### 1.410 Internal command: ASCII

ASCII command can be given in three ways:

- a) ASCII TABLE
  - b) ASCII SHOW
-

c) ASCII ASK

In (a) case ProgED shows you an ASCII symbol table. Clicking on any char you'll insert it in text. In (b) case, on the contrary, ProgED shows you, using a special window, the ascii value of char under cursor. At last, in (c) case ProgED shows you a window. You'll can, thus, insert wished ascii code.

### 1.411 Internal command: BACK

Backspace a char or a word (according user defined separators). The optional TIMES arguments allows you to specify how many backspaces ProgED must do (default 1).

- a) BACK
- b) BACK TIMES <number>
- c) BACK WORD TIMES <number>

### 1.412 Internal command: BLOCKCASE

This command allows you to modify all letters belonging actually marked text block. Using the following three flags you can modify command behaviour:

- a) BLOCKCASE UPPER
- b) BLOCKCASE LOWER
- c) BLOCKCASE TOGGLE

In (a) case all letters will become upper. In (b) case all letters will become lower. At last, in (c) case, all upper letters will become lower and vice versa.

### 1.413 Internal command: BRACKET

You can specify it in two ways:

- a) BRACKET MATCH
- b) BRACKET CHECK

In former case ProgED searches for bracket matching bracket under cursor (you can search for any bracket defined in special preference window). In latter case ProgED will check the number of "(" and ")" brackets on cursor line. If they are not equal (and ONLY if they aren't equal) ProgED will report you.

### 1.414 Internal command: CENTER

---

This command center the active window on screen.

- a) CENTER

### 1.415 Internal command: CHARCASE

This command modify the letter at cursor position so that they will become lower, upper, or toggle its case. Three ways are possible:

- a) CHARCASE UPPER
- b) CHARCASE LOWER
- c) CHARCASE TOGGLE

First way modifies a lower letter to upper, second from upper to lower and third toggles from lower to upper or vice versa.

### 1.416 Internal command: CLEAR

This command does a "clear" operation. It, substantially, ←  
delete  
marked text block from text. Deleted block ISN'T inserted in a  
clipboard (however you can get it using an  
UNDO  
operation).

### 1.417 Internal command: CLEARTEXT

CLEARTEXT clears active window deleting all text. The FORCE flag allows you to force operation if text was changed (in this case, if you don't use FORCE, ProGED will ask you to confirm operation).

- a) CLEARTEXT [FORCE]

### 1.418 Internal command: CLOSE

CLOSE closes the active window freeing all text. The FORCE flag allows you to force operation if text was changed (in this case, if you don't use FORCE, ProGED will ask you to confirm operation).

- a) CLOSE [FORCE]
-

### 1.419 Internal command: COMMENTSCOLORSPREFS

This command can be given only this way:

- a) COMMENTSCOLORSPREFS ASK

This command opens a preference window that allows you to change colors used in comment render. See

Comment colors

.

### 1.420 Internal command: COMPLETE

This command says ProgED to complete phrase wrote at left side of cursor. As we saw ProgED stores a phrase dictionary and it will be used to get phrase. See

Dictionary

.

- a) COMPLETE

### 1.421 Internal command: COPY

COPY allows you to copy to a clipboard (standard Amiga) ←↔ contents of actually marked block. Two ways are possible:

- a) COPY
- b) COPY CLIP <number>

In former case the block will be copied in the standard clipboard (the standard clipboard is the clipboard defined in Miscellaneous (glb)

.

The latter case, on the contrary, allows you to specify wished clipboard number (0-255).

### 1.422 Internal command: CR

CR is the command ProgED executes each time user presses return key. An optional argument TIMES allows you to specify how many times ProgED must press return (default 1).

- a) CR
  - b) CR TIMES <number>
-

### 1.423 Internal command: CUT

CUT allows you to cut the actually marked text block to a ↔ clipboard (standard Amiga). You can use this command in two ways:

- a) CUT
- b) CUT CLIP <number>

In former case text block will be deleted from text and copied in standard clipboard (standard clipboard is the clipboard defined in

Miscellaneous (glb)

. In latter case, on the contrary, ProgED allows you to specify clipboard number (0-255).

### 1.424 Internal command: DEL

DEL is executed each time you press del key. It deletes a char or a word (according to user defined separators). The optional TIMES argument allows you to specify how many times ProgED must press del key. Last usage way allows you to delete right side of line just at right of cursor.

- a) DEL
- b) DEL TIMES <number>
- c) DEL WORD TIMES <number>
- d) DEL EOL

### 1.425 Internal command: DELFILE

DELFILE command allows you, without quitting ProgED, to delete a file. Two ways are possible:

- a) DELFILE NAME <name>
- b) DELFILE ASK

In former case ProgED will delete named file, in latter it will ask user (using a file-requester) for wished file name to delete.

### 1.426 Internal command: DICTIONARYPREFS

DICTIONARYPREFS allows you to open preference window regarding the dictionary (see

Dictionary

). It can be given in a single

way:

---

- a) DICTIONARYPREFS ASK

### 1.427 Internal command: DOWN

DOWN moves cursor down. It allows you five ways:

- a) DOWN MAX
- b) DOWN STEP
- c) DOWN PAGE
- d) DOWN
- e) DOWN TIMES <number>

In (a) case the cursor will be moved to bottom of text (last line).  
In (b) case the cursor will be moved down some lines. Scroll step is  
the 'Alt step' from preference window

Miscellaneous (glb)

In (c) case cursor jumps to bottom of page or to bottom of next page.  
In (d) case ProGED will move cursor simply down one line. At last,  
in (e) case, the command allows you to specify how many lines do you want  
to move cursor down.

### 1.428 Internal command: DRAWINFOPREFS

You can use this command only in a single way:

- a) DRAWINFOPREFS ASK

This command opens a preference window that allows you to choose  
pens used to render ProGED screen. See

Drawinfo

### 1.429 Internal command: ENLARGE

ENLARGE command enlarges the active window such that it will  
fit all screen area.

- a) ENLARGE

### 1.430 Internal command: FILETYPEPREFS

You can use this command only in a single way:

- a) FILETYPEPREFS ASK

This command opens a preference window that allows you to modify all filetypes recognized from ProgED.

### 1.431 Internal command: FILETYPE

You can use this command in the following ways:

- a) FILETYPE ASK
- b) FILETYPE FORCE <label>

Former way opens a window to let user choose favourite filetype. Latter way simply force filetype which user defined label is <label>. Both ways change filetype related to the active window.

### 1.432 Internal command: FOLD

FOLD allows you to activate the fold function. You can use this command in three ways:

- a) FOLD
- b) FOLD ALL
- c) FOLD TOGGLE

In (a) case the block (returned from external fold function) will be folded, if any. If cursor was on a fold nothing happen. In (b) case ProgED will close all block found in text. In (c) case, finally, if cursor is on a fold this will be open, otherwise the command behaviour is equal to first case.

### 1.433 Internal command: FOLDPREFS

This command can be given in a single way:

- a) FOLDPREFS ASK

It allows you to open the preference window regarding external fold function. See

Fold function

.

### 1.434 Internal command: FONTSPREFS

---



FONTSPREFS can be gives in a single way:

- a) FONTSPREFS ASK

This command opens a preference window regarding the ProgED screen fonts. See

```
Font
.
```

### 1.435 Internal command: FREEZE

FREEZE command closes a window without free its text. The internal commands

```
SWAP
and
UNFREEZE
allows you to
```

reopen the window. You can use it in two ways:

- a) FREEZE FILE <name>
- b) FREEZE CURRENT

In former case you have to specify the (complete) file name of wished window to close. In latter case, on the contrary, you'll close the actually active window.

### 1.436 Internal command: GOTO

GOTO allows you to jump to wished text marker. Using the ← command

```
STORE
```

you can mark up to 10 text position plus several "static" markers. Each position is identified using a number from 1 to 10. Using GOTO (first way, see below), specifying the relative number, you can jump to one of this markers. Regard "static" markers you must use second way which opens a window to let user choose wished marker to jump to.

- a) GOTO POSITION <number>
- b) GOTO STORED

(where <number> is an integer from 1 to 10).

### 1.437 Internal command: GUI

You can use the GUI command in two ways:

- a) GUI LOCK
- b) GUI UNLOCK

GUI should be used ONLY in an ARexx program (or it haven't effect). The former case stops mouse, keyboard and menu input of active window. The latter case enables it. This allows you to avoid user to modify macro behaviour.

### 1.438 Internal command: HUNTER

HUNTER allows you to search for (and load) wished file using ↔ its file name and searching for it in a special paths list (see File hunter ).

Three possible ways are:

- a) HUNTER ASK
- b) HUNTER STRING <name>
- c) HUNTER EXTRACT

In (a) case Proged will open a window to ask you filename. In (b) case the given string (name) will be used directly as filename. In last case, finally, Proged "extract" filename from text reading chars under cursor (Proged manages <, >, ", ' chars).

### 1.439 Internal command: HUNTERPREFS

HUNTERPREFS command opens a preference window regarding file hunter. See

File hunter  
. Unique admissible way is:

- a) HUNTERPREFS ASK

### 1.440 Internal command: ICONIFY

ICONIFY command iconifys Proged, closes its screen, and shows ↔ if user asked for them, an AppIcon and an AppItem on WB screen. Using these icon and item (or the special hotkey) user can reopen Proged screen. This effect can be obtained using an ARexx macro and the command:

UNICONIFY

.

### 1.441 Internal command: INDENTBLOCK

INDENTBLOCK allows you to "move" actually marked block text one char left or right. A special window allows you to do it using some gadgets. Admissible ways are:

- a) INDENTBLOCK LEFT
- b) INDENTBLOCK RIGHT
- c) INDENTBLOCK ASK

In (a) case block will be moved one char left. In (b) case it will be moved one char right. At last, in (c) case, ProgED opens the special dialog window.

### 1.442 Internal command: INDENTPREFS

INDENTPREFS command opens preference window regarding indentation. ↔

See

Indentation  
. Unique admissible way to use it

is:

- a) INDENTPREFS ASK

### 1.443 Internal command: INS

INS command allows you to toggle between insert and overwrite modes. Three ways are admissible:

- a) INS ON
- b) INS OFF
- c) INS TOGGLE

In (a) case you'll activate insert mode. In (b) case you'll activate overwrite mode. In last case you will toggle between insert and overwrite modes.

### 1.444 Internal command: INSERT

This command allows you to insert some objects in text. Here you can see five admissible ways:

- a) INSERT FILE <name>
  - b) INSERT PATH
  - c) INSERT DATE
  - d) INSERT TIME
  - e) INSERT ASK
-

In (a) case ProgED will insert, at cursor position, file which name is <name>. In (b) case ProgED will show user a file requester to choose a path. Path itself will be inserted in text at cursor position. The (c) case allows you to insert actual date. The (d) case, of course, allows you to insert actual time. At last, the (e) case allows you to insert a file at cursor position. You'll asked for wished file using a file requester.

### 1.445 Internal command: JUMPTOBYTE

JUMPTOBYTE allows you to jump to any text byte. You can use this command in two different ways:

- a) JUMPTOBYTE NUM <number> [HIGHLIGHT]
- b) JUMPTOBYTE ASK [HIGHLIGHT]

In former case ProgED will jump to text byte which number is <number>. In latter case ProgED will ask you for wished byte number. Anyway you can always specify the HIGHLIGHT flag. If you use it ProgED will highlight searched position.

### 1.446 Internal command: JUMPTOLINE

JUMPTOLINE command allows you to jump to any line in text. Admissible ways are:

- a) JUMPTOLINE NUM <number> [HIGHLIGHT]
- b) JUMPTOLINE ASK [HIGHLIGHT]

In (a) case ProgED will jump to text line which number is <number>. In (b) case ProgED will open a window to ask you wished line number. Anyway you can always specify the HIGHLIGHT flag. if you use it ProgED will highlight searched position.

### 1.447 Internal command: KEYBOARDPREFS

This command opens a preference window regarding keyboard preferences. ←

See

keyboard  
. You can use this command in this way:

- a) KEYBOARDPREFS ASK

## 1.448 Internal command: KWCOLORSPREFS

KWCOLORSPREFS opens a preference window regarding language ↔  
keyword  
and relative colors used to render them. See  
Keyword colors  
. You can  
use a single way:

- a) KWCOLORSPREFS ASK

## 1.449 Internal command: LAYOUTBLOCK

This command allows you to layout actually marked text block.  
ProgED can do it in seven different ways:

- a) LAYOUTBLOCK LEFT
- b) LAYOUTBLOCK RIGHT
- c) LAYOUTBLOCK CENTER
- d) LAYOUTBLOCK FULL
- e) LAYOUTBLOCK LEFT MOVE
- f) LAYOUTBLOCK RIGHT MOVE
- g) LAYOUTBLOCK CENTER MOVE

In (a) case text block will be reorganized to fit all columns  
between left and right margins (specified in  
Miscellaneous (ft)  
).  
Each line will start at left margin.

In (b) case text block will be reorganized to fit all columns  
between left and right margins (specified in  
Miscellaneous (ft)  
).  
Each line will be right-justified.

In (c) case text block will be reorganized to fit all columns  
between left and right margins (specified in  
Miscellaneous (ft)  
).  
Each line will be centered between margins.

In (d) case text block will be reorganized to fit all columns  
between left and right margins (specified in  
Miscellaneous (ft)  
).  
Each line will be filled using spaces to fit all area.

In (e) case all lines belonging to text block will be moved to  
left margin.

In (f) case all lines belonging to text block will be moved to  
right margin.

---

In (e) case all lines belonging to text block will be centered between left and right margins.

### 1.450 Internal command: LEFT

LEFT command is equal to the DOWN command. It has two differences: LEFT doesn't supply PAGE argument and LEFT supplies a WORD argument to jump left a word.

### 1.451 Internal command: LINE

LINE command allows you some operation on cursor line:

- a) LINE CUT
- b) LINE PASTE
- c) LINE SWAP
- d) LINE DOUBLE

In case (a) cursor line will be deleted from text and it will be copied in a special internal clipboard. In (b) case you can insert a previously cutted line into the text. In (c) case cursor line and next line will be swapped. At last, in (d) case, cursor line will be doubled.

### 1.452 Internal command: LOAD

LOAD command is similar to OPEN but it loads a text into a previously opened window. INE command allows you some operation on ← cursor line:

### 1.453 Internal command: MACRO

MACRO command manages on-line ARexx macro recording. You can use it in following ways:

- a) MACRO RECSTART
  - b) MACRO RECSTOP
  - c) MACRO RECCONT
  - d) MACRO PLAY [TIMES <number>]
-

- e) MACRO SAVE
- f) MACRO ASKTIMES

In (a) case ProgED will start macro recording. All executed internal commands will be stored automatically. In (b) case you'll stop recording. Using (c) case you can continue recording starting from last recorded macro end. In (d) case you can play macro. Optional TIMES argument allows you to specify how many times ProgED will play macro. In (e) case ProgED will pop up a file requester to save stored macro. In last case ProgED will ask you how many times it must play macro.

### 1.454 Internal command: MAKEDIR

MAKEDIR command, without quitting ProgED, allows you to create a drawer. You can use this ways:

- a) MAKEDIR DIR <name>
- b) MAKEDIR ASK

In former case ProgED will create a drawer using <name> string. In latter case ProgED will ask you for name using a file requester.

### 1.455 Internal command: MARK

MARK allows you to mark start of a text block. You can use this command modes:

- a) MARK
- b) MARK COLUMNAR
- c) MARK ALL
- d) MARK LINE
- e) MARK WORD

In (a) case ProgED will mark start of block at cursor position. In (b) case ProgED will mark start of a columnar block at cursor position. In (c) case you'll mark all text in window. At last, in (d) case, you'll mark start of a "line" text block. A line text block is a block starting at first column of first line and ending at last column of last line. This marking is required to do some special operations. In (e) case ProgED marks the word below cursor. Of course ProgED uses user defined separators to identify the word itself.

### 1.456 Internal command: MENU

This command activates the specified menu such if you select it using mouse. This is command syntax:

---

- a) MENU MENU <menu number> ITEM <item number> SUB <sub number>

### 1.457 Internal command: MENUPREFS

This command opens preference window regarding menu. See  
Menu  
. Unique admissible way is:

- a) MENUPREFS ASK

### 1.458 Internal command: MISC1PREFS

This command opens preference window regarding miscellaneous global preferences. See  
Miscellaneous (glb)  
. You can use this command in a single way:

- a) MISC1PREFS ASK

### 1.459 Internal command: MISC2PREFS

This command opens preference window regarding miscellaneous filetypes preferences. See  
Miscellaneous (ft)  
. You can use this command in a single way:

- a) MISC2PREFS ASK

### 1.460 Internal command: MOUSE

It manages the cursor. Four ways are admissible:

- a) MOUSE SET
- b) MOUSE MARK
- c) MOUSE MARKCOL
- d) MOUSE MARKLINE

In (a) case ProGED will move cursor to mouse pointer position. In (b) case ProGED will start a block marking. (c) case starts a columnar marking and (d) starts a line block marking.

---



### 1.461 Internal command: MOUSEPREFS

MOUSEPREFS command opens a preference window regarding mouse management (new for 2.0). Unique admissible way is:

- a) MOUSEPREFS ASK

### 1.462 Internal command: MOVECURSOR

It moves cursor to another position. Three ways are admissible:

- a) MOVECURSOR LINE <number>
- b) MOVECURSOR COL <number>
- c) MOVECURSOR LINE <number> COL <number>

In (a) case ProGED will jump to given line (via <number>). In (b) case ProGED will jump to given column (via <number>). In (c) case ProGED will jump to wished position (gived via line and column numbers).

### 1.463 Internal command: NEW

It opens a new empty window.

- a) NEW

### 1.464 Internal command: NEWCURDIR

NEWCURDIR allows you to modify ProGED current directory. Two ways are admissible:

- a) NEWCURDIR DIR <name>
- b) NEWCURDIR ASK

In former case ProGED will create a new drawer using given name. In latter case ProGED will ask you drawer's name using a file requester.

### 1.465 Internal command: NEWNAME

NEWNAME allows you to modify active window filename. It's possible to use in following two ways:

- a) NEWNAME FORCE <name>
- b) NEWNAME ASK

In former case ProGED will set as new name the <name> string. In latter case ProGED will ask you for new name using a file-requester.

---

## 1.466 Internal command: OPEN

It opens a new window to load a text. You can use this command using these four ways:

- a) OPEN ASK
- b) OPEN ASK PATH <name>
- c) OPEN FILE <name> [FORCECHANGE]
- d) OPEN PRJFILE <name> [FORCECHANGE]

In (a) case ProgED will ask you for filename to open (using a file requester). In (b) case you can specify path gadget contents. In (c) case ProgED loads file given (via <name>). In (d) case ProgED will search for wished file (ProgED will search it WITHOUT path, checking only for file name!) among the opened windows. If it finds it ProgED will activate its window. Otherwise ProgED will start research among project files. If, in (c) and (d) case, you use the FORCECHANGE parameter and you are loading an already loaded file, ProgED will no ask you if you want to reload same file.

NOTE: The (d) case was been created to allow SCMSG utility, supplied with SAS compiler, to search for wished files. For an example you can see file 'SASC\_Support/SCMSG.options' containing options used for SCMSG utility.

## 1.467 Internal command: PASTE

PASTE command inserts your clipboard contents in text at cursor position. Two ways are admissible:

- a) PASTE
- b) PASTE CLIP <number>

In former case ProgED will insert text block from standard clipboard (see

Miscellaneous (glb)

. In latter case ProgED will use given clipboard number.

## 1.468 Internal command: PREFS

PREFS command manages prefs loading and saving. You can use it in the following ways:

- a) PREFS LOAD <name>
  - b) PREFS SAVE <name>
  - c) PREFS DEFLOAD
  - d) PREFS DEFSAVE
  - e) PREFS USE
  - f) PREFS ASKLOAD
-

## g) PREFS ASKSAVE

In (a) case ProGED will load wished preferences using file name <name>. In (b) case ProGED will save actual preferences using file name <name>. In (c) case ProGED will load default preferences from 'ENV:PED/PED.prefs' file. In (d) case ProGED will save actual preferences as default preferences in files 'ENV:PED/PED.prefs' and 'ENVARC:PED/PED.prefs'. In (e) case ProGED will save actual preferences as default preferences in file 'ENV:PED/PED.prefs'. In (f) case ProGED will ask you for wished preferences using a file-requester. In (g) case proGED will save preferences asking you for file name using a file-requester.

## 1.469 Internal command: PRJREFERENCEFIND

PRJREFERENCEFIND command allows you to search for dynamic symbols in actual project:

- a) PRJREFERENCEFIND STRING <reference> [CASE]
- b) PRJREFERENCEFIND EXTRACT [CASE]
- c) PRJREFERENCEFIND ASK [CASE]

In (a) case ProGED will search for given reference in string <reference>. In (b) case ProGED will extract string from text under cursor. In (c) case ProGED will ask you for wished string using a special window. At last optional CASE parameter allows you to activate case-sensitive research (default no case-sensitive).

## 1.470 Internal command: PROJECT

PROJECT command allows you to manage actual ProGED project. Five ways are admissible:

- a) PROJECT ASK
- b) PROJECT ASKLOAD
- c) PROJECT ASKSAVE
- d) PROJECT LOAD <name>
- e) PROJECT SAVE <name>

In (a) case ProGED will show you a special window. This special window allows you to modify actual project, to load new project, to save project, etc. In (b) case ProGED will load a project asking user for filename using a file-requester. In (c) case ProGED will save actual project asking user for filename using a file-requester. In (d) case ProGED will load given project using <name> file name. In (e) case ProGED will save actual project using <name> file name.

## 1.471 Internal command: PROJECTSEARCH

PROJECTSEARCH allows you to search for a given pattern in project's files. It searches occurrences as

```
SEARCH
command
```

does.

- a) PROJECTSEARCH ASK
- b) PROJECTSEARCH FOR <string> [CASE]

In latter case ProGED will ask user for wished string. In former case ProGED will search for <string>. Optional CASE parameter forces case-sensitive research.

### 1.472 Internal command: QUIT

QUIT command can be used this way:

- a) QUIT

QUIT execution forces ProGED quitting (:-O) !

### 1.473 Internal command: REDO

```
REDO command is symmetric of
UNDO
command. You can
```

use it so:

- a) REDO
- b) REDO TIMES <number>

In former case ProGED will redo a single operation. In latter case ProGED will redo <number> operations (default 1).

### 1.474 Internal command: REFERENCEFIND

REFERENCEFIND command allows you to search, in active window for static symbols in actual references map. Three ways are admissible:

- a) REFERENCEFIND ASK
- b) REFERENCEFIND STRING <string> [CASE]
- c) REFERENCEFIND EXTRACT [CASE]

In (a) case ProGED will open a search window. In this window you'll can specify wished search pattern. In (b) case, on the contrary, ProGED will search for <string> string. In (c) case ProGED will extract

---

searched string from text under cursor. Optional CASE parameter forces a case-sensitive research.

### 1.475 Internal command: REFERENCEPREFS

REFERENCEPREFS command opens a preference window regarding static references map. Before to use this command is necessary to mount at least a scanner using the SCANNERPREFS command. Unique admissible way is:

- a) REFERENCEPREFS ASK

### 1.476 Internal command: RENAME

RENAME command allows you to rename a file without quitting ProgED. Two ways are possible:

- a) RENAME ASK
- b) RENAME FILE <file1> AS <file2>

In (a) case ProgED will open a file-requester to ask you for old file, and a window to ask you for new file name. In (b) case ProgED will rename <file1> as <file2>.

### 1.477 Internal command: REPLACE

REPLACE command allows you to replace all occurrences of an AmigaDOS pattern for a string. For an extended overview see

Search for...  
. Three ways are admissible:

- a) REPLACE STRING <pattern> BY <string> [CASE] [REVERSE] [WILDCARDS]
- b) REPLACE NEXT [CASE] [REVERSE] [WILDCARDS]
- c) REPLACE ASK

In (a) case ProgED will search for first pattern occurrence and it will replace it for <string>. CASE and REVERSE (optional parameters) specify if you want a case-sensitive research and/or a reverse research. In (b) case, which is useful after use of an (a) command, ProgED will search for next pattern occurrence. In (c) case ProgED will open a standard research window. In (a) and (b) cases WILDCARDS argument allows you to force a search using AmigaDOS wildcards. If you omit this argument you'll get a simple search.

---

### 1.478 Internal command: REVERT

REVERT command reload from disk text contained in active window. You can use it so:

a) REVERT [FORCE]

Optional FORCE parameter allows you to force operation if file was changed. If file was changed, in fact, ProGED will ask you to confirm operation. FORCE disables this request.

### 1.479 Internal command: REQUEST

REQUEST command is, generally, used only in ARexx scripts. It supplies a standard requester. Unique admissible way is:

a) REQUEST TITLE <title> BODY <body> FORMAT <format>

<title> is requester title, <body> requester body and <format> format gadget string. You can create gadgets using a '|' char in <format> string (see intuition EasyRequest). This command returns EasyRequest return code.

### 1.480 Internal command: RIGHT

RIGHT command moves cursor to right. Its use is equal to use ←  
of

DOWN

command. It has two differences: RIGHT doesn't supply PAGE argument and WORD argument can be used to jump right a word.

### 1.481 Internal command: SAVE

SAVE command saves actually active window text to disk. You can use it so:

- a) SAVE ALLCHANGES [XPK]
- b) SAVE [XPK]

In former case ProGED will save all changed files. In latter case ProGED will save active window text. Anyway the optional XPK parameter allows you to save xpk-packed files, according to actual preferences.

---

### 1.482 Internal command: SAVEAS

SAVEAS command allows you to save text from active windows to disk, using a new file name. Two ways are possible:

- a) SAVEAS FILE <name> [XPK]
- b) SAVEAS ASK [XPK]

In former case ProgED will save actual window text using given filename <name>. In latter case ProgED will save actual window text asking user for file name using a file-requester. Anyway optional XPK parameter allows you to save xpk-packed files, according to actual preferences.

### 1.483 Internal command: SAVEBLOCKAS

SAVEBLOCKAS command saves actually marked text block to disk. You can use this command in following ways:

- a) SAVEBLOCKAS FILE <name>
- b) SAVEBLOCKAS ASK

In former case ProgED will save text block using given filename <name>. In latter case ProgED will ask you for file name using a file requester.

### 1.484 Internal command: SAVEDFILEPREFS

SAVEDFILEPREFS command allows you to open a preference window regarding file protection bits, file comments, etc.

See also

Saved files

. You can use this command in a

single way:

- a) SAVEDFILEPREFS ASK

### 1.485 Internal command: SCAN

SCAN command allows you to jump, using any of mounted ↔  
scanners,

to a symbol definition. For a wide overview see

Scan

. You can

use this command in a single way:

- a) SCAN ASK
-

## 1.486 Internal command: SCANPREFS

SCANPREFS command allows you to open a preference window regarding mounted scanners, used during a research started with a

SCAN

command. You can use this command in following way:

- a) SCANPREFS ASK

## 1.487 Internal command: SCREENCOLORSPREFS

SCREENCOLORSPREFS command allows you to open a preference window regarding screen colors. You can use this command in following way:

- a) SCREENCOLORSPREFS ASK

## 1.488 Internal command: SCREENMODEPREFS

SCREENMODEPREFS command allows you to open a preference window regarding screen mode. You can use this command in following way:

- a) SCREENMODEPREFS ASK

## 1.489 Internal command: SEARCH

SEARCH command allows you to search for any occurrence of an AmigaDOS pattern in text. For a wide overview see Search for...

You can use this command in following ways:

- a) SEARCH FOR <pattern> [CASE] [REVERSE] [WILDCARDS]
- b) SEARCH ASK
- c) SEARCH NEXT [CASE] [NOCASE] [REVERSE] [FORWARD] [WILDCARDS]

In case (a) ProGED will search for given pattern. Optional CASE and REVERSE parameters select case-sensitive research and reverse-research. In (b) case ProGED will open a search window. In (c) case, useful if you given an (a) or (b) command, you can search for next pattern occurrence. CASE force a case-sensitive search. NOCASE force a no case-sensitive search. If you omit both of them then you use search specified into (last opened) search window. The couple FORWARD/REVERSE works similarly. Last, the WILDCARDS arguments allows to force a search using AmigaDOS wildcards. If you omit this argument you'll get a simple

---



search.

### 1.490 Internal command: SPLIT

SPLIT command allows you to split active window. Later, you'll have two equal window. So you'll can edit text in both windows. SPLIT command haven't arguments:

- a) SPLIT

### 1.491 Internal command: STARTUPMACROPREFS

STARTUPMACROPREFS command allows you to specify some ARexx macro. ProgED will use these macros at start, iconify and uniconify time. You can use only a single way:

- a) STARTUPMACROPREFS ASK

### 1.492 Internal command: STORE

STORE command allows you to store in any of 10 markers cursor position. Using

GOTO

command, later, you'll can jump to stored position. Second admissible way allows you to store current cursor position into a "static" marker.

- a) STORE POSITION <number>
- b) STORE CURRENT

<number> is an integer from 1 to 10.

### 1.493 Internal command: SWAP

SWAP allows you to close active window (without freeing its text) and reopen a window closed using

FREEZE

command. This

allows you to open, on screen, a single window a time.

- a) SWAP ASK
- b) SWAP ASK CLOSE <file>
- c) SWAP CLOSE <file1> OPEN <file2>

In (a) case ProgED will close active window and it will open a

---

special window to ask user which window have to be opened. In (b) case ProGED will close window containing <file> and it will open the special window to ask you wished window. In (c) case ProGED will close <file1> window and it will open <file2> window.

### 1.494 Internal command: TEMPLATEPREFS

TEMPLATEPREFS opens a preference window regarding templates. For a wide overview on templates see also

Template

. Unique

admissible way is:

- a) TEMPLATEPREFS ASK

### 1.495 Internal command: TYPE

TYPE allows you to type any text in active window at cursor position. You can use TYPE in following ways:

- a) TYPE ASCII <number>
- b) TYPE TEXT <text> [STAY]

In former case ProGED will insert char whose ASCII code was given using <number>. In latter case ProGED will simply insert <text> at cursor position. If you specify STAY parameter cursor will not move.

### 1.496 Internal command: UNDO

UNDO command allows you to repair if your typing was wrong. At each UNDO invocation ProGED repairs to a user change. You can use this command in following ways:

- a) UNDO
- b) UNDO TIMES <number>

In latter case you can specify how many undo operations ProGED must do.

### 1.497 Internal command: UNFOLD

UNFOLD allows you to open a text block folded using the FOLD command. Two ways are possible:

- a) UNFOLD
- b) UNFOLD ALL

In (a) case ProgED will unfold text block under cursor (if any).  
In (b) case ProgED will unfold all folded text blocks in text.

## 1.498 Internal command: UNFREEZE

UNFREEZE command allows you to open any frozen window closed ←  
using

FREEZE  
or  
SWAP  
commands. You can use two ways:

- a) UNFREEZE FILE <name>
- b) UNFREEZE ASK

In former case ProgED will unfreeze window containing file named <name>. In latter case ProgED will open a special window to ask user wished window.

## 1.499 Internal command: UNICONIFY

UNICONIFY reopens ProgED screen after an  
ICONIFY  
command.

This command is useful only in ARexx macro (naturally..). It haven't arguments:

- a) UNICONIFY

## 1.500 Internal command: UP

UP command is equal to DOWN command, so see  
DOWN

- a) UP MAX
  - b) UP STEP
  - c) UP PAGE
  - d) UP
  - e) UP TIMES <number>
-

### 1.501 Internal command: WINDOW

WINDOW command allows you to activate, in a cycle, all window on ProGED screen. This command, thus, allows you to use only keyboard, without jumping to mouse. It have two ways:

- a) WINDOW NEXT
- b) WINDOW PREV

In former case ProGED will activate next window, while in former ProGED will active previous window.

### 1.502 Internal command: WSELECT

WSELECT command allows you to select a window during an ARexx macro. You can use it so:

- a) WSELECT ACTIVE
- b) WSELECT NEXT
- c) WSELECT FILE <name>

In (a) case ProGED will say ProGED that we wish send our ARexx command to active window. In (b) case ProGED will select next window. In (c) case ProGED will activate window whose file is <name>.

NOTE: If you want to write an ARexx macro you MUST insert a WSELECT command above all commands (generally WSELECT ACTIVE). Otherwise ProGED can loose your commands!

WSELECT use is useless during menu or keyboard programmation. In these cases ProGED executes (automatically) a WSELECT ACTIVE command.

### 1.503 Internal command: ZIP

ZIP command zips your window as Intuition gadget does. It haven't arguments:

- a) ZIP

### 1.504 Edit program...

ProGED opens the "Edit program" window each time user wants ←  
modify  
a menu or a keyboard hotkey. Using this window you can specify an instruction list executed in list order. Instruction types can be anyone of:

---

- 1) dummy (-----)
- 2) internal
- 3) Shell
- 4) Arexx
- 5) Text

See all of them:

- 1) ProgED ignores completely "dummy" instruction. They are useful if you want to comment out some block.
- 2) Internal instructions are, simply,  
Internal commands  
.
- 3) Shell instructions allows you to execute normal AmigaDOS programs. For each command you can specify current directory, output file (default "con:") and sync/async execution. It's important to say that, generally, you have to use async execution. This happens because if your program run sync and it sends an ARExx msg to ProgED you'll hang-up your Amiga! This happens because if ProgED are executing your program it can't wait for messages...
- 4) ARExx instructions allows you to execute any ARExx script. This script types are ALWAYS async to avoid problems as for shell instructions. You can specify current directory and output file, too.
- 5) Text instruction simply inserts given string in active window.

At last, in Arexx macro string, Shell command string or internal command string you can use following modifiers:

- 1) \n

it will be replaced with COMPLETE file name contained in active window.

- 2) \f

it will be replaced with file name (with no path) contained in active window.

- 3) \s

it will be replaced with public screen name on which ProgED was opened.

- 4) \xxx

it will be replaced with ASCII char which code is 'xxx'.

- 5) \

it will be replace with a single \ char.

Starting from 2.0 release ProgED allows you to force the "Shanghai"

---

mode which is a special mode used to force the ProgED public screen to become (for one or two seconds) the default public screen. This special mode allows you to force Intuition to open windows on ProgED public screen. Simply activate Shanghai mode and set how many seconds the ProgED screen must remain the default public screen. That's all.