

# •Beginners Mac Assembly•



## Appendices



### Appendix A - Fantasm errors

#### A1 - Fatal errors.

**Out of memory!** - Exactly what it says - allocate more memory to Fantasm in the "Finders" get info box.

**File too big, allocate more memory...** - Fantasm needs more memory to assemble this program, again allocate more memory.

**Address not recognised by dispose\_ptr.** - You should not get this error message. It indicates the amount of memory taken by Fantasm does not match up with the memory it wants to return to the system. Contact Lightsoft (unless you've been hacking the program).

**Name of file - can't find this file!** - Fantasm can not find an include file. Check the file physically exists in the right folder, also check the spelling of the filename.

**Can't find the main source file** - Fantasm cannot find the main source file. You should not get this unless you've hand typed the filename into the selector and mistyped.

**Open error** - Fantasm could not open a file to read from. The file may be open from another application or physically damaged.

**Create error** - Fantasm could not create a file, possibly because the disk is locked, or damaged.

**Cant open file after creating it** - You should not get this error - it is included just in case something really strange happens, such as a power glitch during assembly.

#### A2 - Pass 1 and 2 errors.

**/ expected in register list** - Generated by the MOVEM instruction. Data registers and address registers must be separated by a / - i.e. d0,d3,d5/a0,a6

**( expected , or ) expected** - basic syntax error - Fantasm expected a parenthesis.

**Addressing mode not available in this version** - Registered users should not see this error message. On some unregistered versions certain addressing modes are disabled for the MOVEM instruction.

**Bad register in list** - Generated by the MOVEM instruction when it cant decipher the register lists

- valid syntaxes are dn-dn,dn/an-an,an

**Bad register number** - valid register numbers are d0-d7, a0-a7, usp, sp, sr and ccr.

**Bytes expected** - End of line was encountered before the number of bytes to be reserved - Eg  
ds.b <CR>

**Byte size not allowed with MOVEP instruction** - The movep instruction only allows .w and .l sizes.

**Cant extend .b** - The 68000 only supports .w and .l extensions.

**Can't use an address register for source op for AND instruction** - The AND instruction will not allow an address register to be used as a source operand - I.e.  
AND.L A0,D0 is illegal.

**Comma expected** - The operands must be separated by a comma - E.g. d0,15(a0)

**Comma expected in operand field** - A comma was expected, but didn't appear. Eg  
move.w 15(a0d0.w) should be 15(a0,d0.w)

**Data too big/small** - Produced if the data is too big or small to fit into the size allocated - i.e. move.b #1234,fred will produce this error as \$1234 would be truncated to \$34 which may not be what you wanted!

**<LABEL> defined twice** - A label has been defined twice, possibly once as an equate and once as a label.

**Destination addressing mode illegal** - You have used the wrong addressing mode for the destination operand for this type of instruction.

**Destination must be a data register** - For this instruction, only a data register is allowed as the destination operand.

**Destination operand must be an address register** - This error is only produced for the LEA instruction, where the destination operand must be an address register.

**Displacement greater than (16 or 8) bits** - The branch offset is too great for this type of branch instruction - for a 68000 branch the displacement must fit into a signed word (+32767/-32768), and for short branches must fit into a signed byte - +127/-128.

**Displacement limited to 16 bits** - This error is generated by the LINK instruction when the stack displacement is greater than +32767 or -32768

**Error in number** - A general error thrown up when the evaluator has a problem converting a number. Eg. \$fa9s is an invalid hex number, and %1021 is an invalid binary number.

**Error with immediate conversion** - The evaluator failed to convert an immediate number correctly - E.g. move.b #1x3,d0 would cause this error.

**Expression too complicated** - Fantasm v1.00 can cope with upto five operators in an expression - any more than this and this error is generated. Eg. 5\*fred+stu/20+1-20\*10 will generate this error.

**Invalid address** - Generated when an address is not terminated correctly in absolute long

addressing mode - for example move.l \$12345678.d0 - the comma has been mistyped as a full stop.

**Invalid instruction** - Fantasm cannot find the op code in its dictionary. Generally a syntactical error - Eg. Movw.l

**Invalid size** - Generated when an incorrect size character is found - i.e. move.c  
Valid size specifiers for vl.00 are .b .w .l or .s for short branches.

**Invalid string - quotes expected** - Fantasm found the end of line before the end quotes in a string. i.e. "Fred

**Longs expected** - See bytes expected.

**Macros not implemented** - V1.00 does not support macro functions.

**Number expected** - fantasm was expecting a number - Eg Move.w #,d0

**Number too big to fit in a byte** - The largest number that will fit in a byte is 255 unsigned or +128/-127 signed. Eg dc.b 300 will produce this error.

**Number too big to fit in a word** - The largest number that will fit in a word is 65536 unsigned or +32767/-32768 signed. Eg dc.w 300000 will produce this error.

**Offset too great at nnnn bytes** - A supplementary error notifying the user of the branch offset when a short branch fails.

**Only 0-8 allowed for addq/subq** - The immediate data must fall in this range for these instructions.

**Only TRAP 0-15 allowed** - The 68000 has 16 trap vectors referenced 0 to 15. Any number outside this range with the trap instruction forces this error.

**Quotes expected** - Speech marks missing from a long string - Eg. "DEAD <CR>

**Register list backwards** - Generated by the MOVEM instruction - registers in a list must start with the lowest i.e. d0-d5 NOT d5-d0.

**Source addressing mode illegal** - If an addressing mode is wrong for a particular type of instruction, Fantasm will tell you.

**Undefined label** - A label that has not been defined was found in an expression. Eg if FRED had not been defined then 5\*fred will generate this error.

**Unrecognised option** - An invalid character has been type after the "opt" directive.  
E.g. opt z

**Words expected** - See bytes expected.

ABusVars	02D8	ACount	0A9A	ADDBase	0CF8
AGBHandle	0D1C	AlarmState	021F	ANumber	0A98
ApFontID	0984	App2Packs	0BC8	ApplLimit	0130
ApplScratch	0A78	ApplZone	02AA	AppPacks	0AB8
AppParmHandle	0AEC	ASCBASE	0CC0	AtalkHk1	0B14
AtalkHk2	0B18	AtMenuBottom	0A0C	AuxCtlHead	0CD4
AuxWinHead	0CD0	BNMQHd	0B60	BootDrive	0210
BootMask	0B0E	BootTmp8	0B36	BtDskRfn	0B34
BufPtr	010C	BufTgDate	0304	BufTgFBkNum	0302
BufTgFFlg	0300	BufTgFNum	02FC	CaretTime	02F4
ChooserBits	0946	ChunkyDepth	0D60	CkdDB	0340
CloseOrnHook	0A88	ColLines	0C22	CoreEditVars	0954
CPUFlag	012F	CQDGlobals	0CCC	CrsrAddr	0888
CrsrBase	0898	CrsrBusy	08CD	CrsrCouple	08CF
CrsrDevice	089C	CrsrNew	08CE	CrsrObscure	08D2
CrsrPin	0834	CrsrPtr	0D62	CrsrRect	083C
CrsrRow	08AC	CrsrSave	088C	CrsrScale	08D3
CrsrState	08D0	CrsrThresh	08EC	CrsrVis	08CC
CurActivate	0A64	CurApName	0910	CurApRefNum	0900
CurDeactive	0A68	CurDeKind	0A22	CurDirStore	0398
CurDragAction	0A46	CurFMDenom	0994	CurFMDevice	098E
CurFMFace	098C	CurFMFamily	0988	CurFMInput	0988
CurFMNeedBits	098D	CurFMNumer	0990	CurFMSize	098A
CurJTOffset	0934	CurMap	0A5A	CurPageOption	0936
CurPitch	0280	CurrentA5	0904	CurStackBase	0908
DABeeper	0A9C	DAStrings	0AA0	DeflStack	0322
DefVCBPtr	0352	DeskCPat	0CD8		
DeskHook	0A6C	SetOSDefKey	0CDC	DeskPattern	0A3C
DeskPort	09E2	DeviceList	08A8	DiskVars	0222
DlgFont	0AFA	DoubleTime	02F0	DragFlag	0A44
DragHook	09F6	DragPattern	0A34	DrMstrBlk	034C
DrvQHdr	0308	DSAlertRect	03F8	DSAlertTab	02BA
DSCtrAdj	0DA8	DSDrawProc	0334	DSErrCode	0AF0
DskErr	0142	DskRtnAdr	0124	DskSwTchHook	03EA
DskVerify	012C	DskWrll	012F	DSWndUpdate	015D
DTQFlags	0D92	DTQueue	0D92	DTskQHdr	0D94
DTskQTail	0D98	EjectNotify	0338	EndSRTPtr	0DB4
ErCode	03A2	EventQueue	014A	EvtBufCnt	0154
ExpandMem	02B6	ExtFSHook	03E6	ExtStsDT	02BE
FCBSPtr	034E	FDevDisable	0BB3	FileVars	0340
Finder	0261	FinderName	02E0	FLckUnlck	0348
FlEvtMask	025E	FlushOnly	0346	FMDefaultSize	0987
FMDotsPerInch	09B2	FMExist	0D42	FMgrOutRec	0998
FMStyleTab	09B6	FondID	0BC6	FondState	0903
FontFlag	015E	FOutAscent	09A5	FOutBold	099E
FOutDenom	09AE	FOutDescent	09A6	FOutError	0998
FOutExtra	09A4	FOutFontHandle	099A	FOutItalic	099F
FOutLeading	09A8	FOutNumer	09AA	FOutRec	0998
FOutShadow	09A3	FOutULOffset	09A0	FOutULShadow	09A1
FOutULThick	09A2	FOutUnused	09A9	FOutWidMax	09A7
FPState	0A4A	FractEnable	0BF4	FrcSync	0349
FSBusy	0360	FScaleDisable	0A63	FScaleHFact	0BF6
FScaleVFact	0BFA	FSFCBLen	03F6	FSQHdr	0360
FSQHead	0362	FSQTail	0366	FSQueueHook	03E2
FSTemp4	03DE	FSTemp8	03D6	FSVarEnd	03F6

GetParam	01E4	GhostWindow	0A84	GotStrike	0986
GrafBegin	0800	GrafVar	0824	GrayRgn	09EE
GZMoveHnd	0330	GZRootHnd	0328	GZRootPtr	032C
HeapEnd	0114	HFSFlags	0376	HiHeapMark	0BAE
HiKeyLast	0216	HiliteMode	0938	HiliteRGB	0DA0
HpChk	0316	HWCfgFlags	0B22	IAZNotify	033C
IconBitmap	0A0E	IconTLAddr	0DAC	IntFlag	015F
IntlSpec	0BA0	IWM	01E0	JAdrDisk	0252
JAllocCrsr	088C	JControl	0242	JCrsrObscure	081C
JCrsrTask	08EE	JDCDReset	0B48	JDiskPrime	0226
JDiskSel	0B40	jDTInstall	0D9C	JFetch	08F4
JFigTrkSpd	0222	JFontInfo	08E4	JGNEFilter	029A
JHideCursor	0800	JInitCrsr	0814	JIODone	08FC
JKybdTask	021A	JMakeSpdTbl	024E	JOpcodeProc	0894
JournalFlag	08DE	JournalRef	08E8	JRdAddr	022A
JRdData	022E	JRecal	023E	JReSeek	024A
JScrnAddr	080C	JScrnSize	0810	JSeek	0236
JSendCmd	0B44	JSetCCrsr	0890	JSetCrsr	0818
JSetSpeed	0256	JSetUpPoll	023A	JShell	0212
JShieldCursor	0808	JShowCursor	0804	JStash	08F8
JSwapFont	08E0	jSwapMMU	0DBC	JUpdateProc	0820
JVBLTask	0D28	JWakeUp	0246	JWrData	0232
KbdLast	0218	KbdType	021E	KbdVars	0216
KeylTrans	029E	Key2Trans	02A2	KeyLast	0184
KeyMap	0174	KeyMVars	0B04	KeypadMap	017C
KeyRepThresh	0190	KeyRepTime	018A	KeyThresh	018E
KeyTime	0186	LastDepth	0D40	LastFond	0BC2
LastFore	0D36	LastLGlobal	0944	LastMode	0D3E
LastPGlobal	0954	LastSPEXtra	0B4C	LastTGGlobal	0AFC
LastTxGDevice	0DC4	LaunchFlag	0902	LGrafJump	0824
LoaderPBlock	093A	LoadFiller	090C	LoadTrap	012D
LoadVars	0900	Lvl1DT	0192	Lvl2DT	01B2
MacJump	0120	MacPgm	0316	MAErrProc	0BE8
MainDevice	08A4	MaskBC	031A	MaskHandle	031A
MaskPtr	031A	MASuperTab	0BEC	MaxDB	0344
MBarEnable	0A20	MBarHeight	0BAA	MBarHook	0A2C
MBDFHndl	0B58	MBSaveLoc	0B5C	MBState	0172
MBTicks	016E	MemErr	0220	MemTop	0108
MenuCInfo	0D50	MenuDisable	0B54	MenuFlash	0A24
MenuHook	0A30	MenuList	0A1C	MickeyBytes	0D6A
MinStack	031E	MMDefFlags	0326	MmInOK	012E
MMU32bit	0CB2	MMUFlags	0CB0	MMUFluff	0CB3
MMUTbl	0CB4	MMUTblSize	0CB8	MMUType	0CB1
MonkeyLives	0100	Mouse	0830	MouseMask	08D6
MouseOffset	08DA	MrMacHook	0A2C	MTemp	0828
NewCrsrJTbl	088C	NewMount	034A	NewUnused	0BC0
NiblTbl	025A	NMIFlag	0C2C	NxtDB	0342
OldContent	09EA	OldStructure	09E6	PaintWhite	09DC
Params	03A4	pCDeskPat	020B	PmgrHandle	0DC8
PollProc	013E	PollRtnAddr	0128	PollStack	013A
PortAUse	0290	PortBUse	0291	PortList	0D66
PrintErr	0944	PrintVars	0944	PWMBuf1	0B0A
PWMBuf2	0312	PWMValue	0138	QDColors	08B0
QDErr	0D6E	QDExist	08F3	RAMBase	02B2
RawMouse	082C	RegRsrc	0347	ReqstVol	03EE
ResErr	0A60	ResErrProc	0AF2	ResLoad	0A5E

ResReadOnly	0A5C	RestProc	0A8C	ResumeProc	0A8C
RGBBlack	0C10	RGBWhite	0C16	RgSvArea	036A
RMgrHiVars	0B80	RMgrPerm	0BA4	RndSeed	0156
ROM85	028E	ROMBase	02AE	RomFont0	0980
ROMMapHndl	0B06	RomMapInsert	0B9E	RowBits	0C20
RSDHndl	028A	SavedHandle	0A28	SavedHilite	0D43
SaveFondFlags	0986	SaveProc	0A90	SaveSegHandle	0930
SaveSP	0A94	SaveUpdate	09DA	SaveVisRgn	09F2
SCCASts	02CE	SCCBSts	02CF	SCCRd	01D8
SCCWrr	01DC	ScrapCount	0968	ScrapHandle	0964
ScrapInfo	0960	ScrapName	096C	ScrapSize	0960
ScrapState	096A	ScrapTag	0970	ScrapVars	0960
Scratch20	01E4	Scratch8	09FA	ScrDmpEnb	02F8
ScrDmpType	02F9	ScreenBytes	0C24	ScreenRow	0106
ScreenVars	0292	ScrHRes	0104	ScrnBase	0824
ScrnVBLPtr	0D10	ScrVRes	0102	SCSIBase	0C00
SCSIDMA	0C04	SCSIDrvrs	0B2E	SCSIFlag	0B22
SCSIGlobals	0C0C	SCSIHsk	0C08	SCSIPoll	0C2F
SdEnable	0261	SDMBusErr	0DC0	SDMJumpTblPtr	0DB8
SdVolume	0260	SegHiEnable	0BB2	SerialVars	02D0
SEVarBase	0C30	SEvtEnb	015C	SFSaveDisk	0214
SInfoPtr	0CBC	SInitFlags	0D90	SlotPrTbl	0D08
SlotQDT	0D04	SlotTICKS	0D14	SlotVBLQ	0D0C
SMGlobals	0CC4	SmgrCore	0BA0	SonyVars	0134
SoundActive	027E	SoundBase	0266	SoundDCE	027A
SoundGlue	0AE8	SoundLevel	027F	SoundPtr	0262
SoundVars	0262	SoundVBL	026A	SPAlarm	0200
SPATalkA	01F9	SPATalkB	01FA	SPClikCaret	0209
SPConfig	01FB	SPFont	0204	SPKbd	0206
SPMisc1	020A	SPMisc2	020B	SPPortA	01FC
SPPortB	01FE	SPPrint	0207	SPValid	01F8
SPVolCtl	0208	SrcDevice	08A0	SRsrcTblPtr	0D24
StkLowPt	0110	Switcher	0282	SwitcherTPtr	0286
SynListHandle	0D32	SysCom	0100	SysEvtBuf	0146
SysEvtMask	0144	SysFontFam	0BA6	SysFontSize	0BA8
SysMap	0A58	SysMapHndl	0A54	SysParam	01F8
SysResName	0AD8	SysVersion	015A	SysZone	02A6
TlArbitrate	0B3F	TableSeed	0D20	TagData	02FA
TaskLock	0A62	TEDoText	0A70	TempRect	09FA
TERecal	0A74	TEScrpHandle	0AB4	TEScrpLength	0AB0
TESysJust	0BAC	TEWdBreak	0AF6	TheCrsr	0844
TheGDevice	0CC8	TheMenu	0A26	TheZone	0118
Ticks	016A	Time	020C	TimeDBRA	0D00
TimeSCCDB	0D02	TimeSCSIDB	0DA6	TimeVars	0B30
TmpResLoad	0B9F	Tocks	0173	ToExtFS	03F2
ToolScratch	09CE	TopMapHndl	0A50	TopMenuItem	0A0A
TrapAgain	0B00	Twitcher1	0B50	Twitcher2	0B7C
UnitNtryCnt	01D2	UsedFWidths	0BF5	UTableBase	011C
VBLQueue	0160	VCBQHdr	0356	VertRRate	0D30
VIA	01D4	VIA2DT	0D70	VideoInfoOK	0DB0
VidMode	0C2E	VidType	0C2D	WarmStart	0CFC
WidthListHand	08E4	WidthPtr	0B10	WidthTabHandle	0B2A
WindowList	09D6	WMgrCPort	0D2C	WMgrPort	09DE
WordRedraw	0BA5	WWExist	08F2		

**\*\*WARNING\*\*** Aimlessly playing with the system variables, whilst being very educational, can

seriously crash your computer! Do not have any unsaved data in memory if poking about in the system variables.

### Appendix C - Ascii codes.

Dec	Hex	Character	Dec	Hex	Character	Dec	Hex	Character
000	00	CTL-@ NULL	001	01	CTL-A SOH	002	02	CTL-B STX
003	03	CTL-C ETX	004	04	CTL-D EOT	005	05	CTL-E ENQ
006	06	CTL-F ACK	007	07	CTL-G BELL	008	08	CTL-H BS
009	09	CTL-I TAB	010	0A	CTL-J LF	011	0B	CTL-K VT
012	0C	CTL-L FF	013	0D	CTL-M CR	014	0E	CTL-N SO
015	0F	CTL-O SI	016	10	CTL-P DLE	017	11	CTL-Q DC1
018	12	CTL-R DC2	019	13	CTL-S DC3	020	14	CTL-T DC4
021	15	CTL-U NAK	022	16	CTL-V SYN	023	17	CTL-W ETB
024	18	CTL-X CAN	025	19	CTL-Y EM	026	1A	CTL-Z SUB
027	1B	CTL-[ ESC	028	1C	CURS LEFT	029	1D	CUR RIGHT
030	1E	CURS UP	031	1F	CURS DOWN	032	20	SPACE
033	21	!	034	22	"	035	23	# (OPTION 3)
036	24	\$	037	25	%	038	26	&
039	27	`	040	28	(	041	29	)
042	2A	*	043	2B	+	044	2C	,
045	2D	-	046	2E	.	047	2F	/
048	30	0	049	31	1	050	32	2
051	33	3	052	34	4	053	35	5
054	36	6	055	37	7	056	38	8
057	39	9	058	3A	:	059	3B	;
060	3C	<	061	3D	=	062	3E	>
063	3F	?	064	40	@	065	41	A
066	42	B	067	43	C	068	44	D
069	45	E	070	46	F	071	47	G
072	48	H	073	49	I	074	4A	J
075	4B	K	076	4C	L	077	4D	M
078	4E	N	079	4F	O	080	50	P
081	51	Q	082	52	R	083	53	S
084	54	T	085	55	U	086	56	V
087	57	W	088	58	X	089	59	Y
090	5A	Z	091	5B	[	092	5C	\
093	5D	]	094	5E	^	095	5F	_
096	60	NOT DEFINED	097	61	a	098	62	b
099	63	c	100	64	d	101	65	e
102	66	f	103	67	g	104	68	h
105	69	i	106	6A	j	107	6B	k
108	6C	l	109	6D	m	110	6E	n
111	6F	o	112	70	p	113	71	q
114	72	r	115	73	s	116	74	t
117	75	u	118	76	v	119	77	w
120	78	x	121	79	y	122	7A	z
123	7B	{	124	7C		125	7D	}
126	7E	~	127	7F	DELETE			

Note - from the above it can be seen that to convert from one case to another - ie A to a is simply a case of adding or subtracting 32 or \$20 to/from the ASCII code.