


```

MM      MM      CCCCCCCC FFFFFFFFFF 77777777 999999 000000
MM      MM      CCCCCCCC FFFFFFFFFF 77777777 999999 000000
MMMM   MMMM   CC          FF          77 99 99 00 00 00
MMMM   MMMM   CC          FF          77 99 99 00 00 00
MM     MM     MM     CC          FF          77 99 99 00 0000
MM     MM     MM     CC          FF          77 99 99 00 0000
MM     MM     MM     CC          FFFFFFFF 77 99999999 00 00 00
MM     MM     MM     CC          FFFFFFFF 77 99999999 00 00 00
MM     MM     MM     CC          FF          77 99 0000 00
MM     MM     MM     CC          FF          77 99 0000 00
MM     MM     MM     CC          FF          77 99 00 00
MM     MM     MM     CC          FF          77 99 00 00
MM     MM     MM     CCCCCCCC FF          77 999999 000000
MM     MM     MM     CCCCCCCC FF          77 999999 000000

```

```

SSSSSSSS DDDDDDDD LL
SSSSSSSS DDDDDDDD LL
SS        DD        DD LL
SS        DD        DD LL
SS        DD        DD LL
SS        DD        DD LL
SSSSSS   DD        DD LL
SSSSSS   DD        DD LL
          SS       DD LL
          SS       DD LL
          SS       DD LL
          SS       DD LL
SSSSSSSS DDDDDDDD LLLLLLLLLL
SSSSSSSS DDDDDDDD LLLLLLLLLL

```

{
{ Version: 'V04-000'
{

```

*****
{*
{* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
{* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
{* ALL RIGHTS RESERVED.
{*
{* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
{* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
{* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
{* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
{* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
{* TRANSFERRED.
{*
{* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
{* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
{* CORPORATION.
{*
{* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
{* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
{*
*****

```

```

{++
{ FACILITY: VAX/VMS CPU-dependent Code Macro Libraries
{
{ ABSTRACT:
{
{   This file contains the SDL source for 11/790 machine check frame
{   definitions.
{
{ ENVIRONMENT:
{
{   n/a
{
{ --

```

```

{ AUTHOR: Wayne Cardoza      CREATION DATE: 01-Nov-1982
{
{ MODIFIED BY:
{
{   V03-011 WMC0008      Wayne Cardoza      23-Jul-1984
{           Still more spec changes.
{
{   V03-010 WMC0007      Wayne Cardoza      08-Jul-1984
{           Assorted spec changes.
{
{   V03-009 WMC0006      Wayne Cardoza      30-May-1983
{           Minor changes and corrections.
{
{   V03-008 WMC0005      Wayne Cardoza      22-FEB-1983

```

Spec changes to MSTAT1, MSTAT2, MDECC

- V03-007 WMC0004 Wayne Cardoza 08-Feb-1983
Rearrange EHSR
- V03-006 WMC0003 Wayne Cardoza 20-Dec-1982
Separate PAMM code from cache bit
- V03-005 WMC0002 Wayne Cardoza 24-Nov-1982
Add the VMS type code definitions.
- V03-004 WMC0001 Wayne Cardoza 14-Nov-1982
Changes to MDECC, MSTAT1

--

```
module $MCF790DEF;
```

```
aggregate MCF790 structure prefix MCF790$;
```

```

SIZE longword unsigned; /* size in bytes of frame
EHSR_OVERLAY union; /* error handling status register
EHSR longword unsigned; /* entire register
EHSR BITS structure;
EHSR_OVERLAY_1 union;
MCHK_CODE byte unsigned; /* VMS puts a code here
EHSR_BITS_1 structure;
SERV_TYPE bitfield mask length 3; /* VMS service type
FILLTA bitfield length 1 fill prefix MCF790 tag $$;
RSRC_REM bitfield mask; /* Resource removed from service
SBIA bitfield mask; /* full SBIA log follows
SBIA_ERR bitfield mask; /* SBIA error summary included
MBOX_1D bitfield mask; /* MBOX 1D error included
end EHSR_BITS_1;
end EHSR_OVERLAY_1;
TRAP_VEC bitfield mask length 8; /* trap vector
FILLT bitfield length 1 fill prefix MCF790 tag $$;
AUTO_SHUT bitfield mask; /* Severe error flag
MEAR_SAV bitfield mask; /* meaningful to microcode
ICS bitfield mask; /* ICS correction
IDRAM bitfield mask; /* IDRAM correction
FDRAM bitfield mask; /* FDRAM correction
FBACS bitfield mask; /* FBACS correction
FBMCS bitfield mask; /* FBMCS correction
IBOX_GPR bitfield mask; /* IBOX GPR correction
EBOX_SPBA bitfield mask; /* EBOX SP B to A
EBOX_SPAB bitfield mask; /* EBOX SP A to B
FBOX_SP bitfield mask; /* FBOX SP correction
FBOX bitfield mask; /* FBOX service
VMS_ENT bitfield mask; /* VMS entered
EHM_ENT bitfield mask; /* EHM entered
MBOX bitfield mask; /* MBOX service
end EHSR_BITS;
end EHSR_OVERLAY;
EVMQSAV longword unsigned; /* virtual address - EBOX port requests
EBCS_OVERLAY union; /* EBOX control status register
EBCS longword unsigned; /* entire register
EBCS BITS structure;
EBCS_OVERLAY_1 union;
EBCS_BITS_2 structure;
FILL2 bitfield fill prefix MCF790 tag $$;
IO_RD bitfield mask; /* IO read abort
MEM_WRT bitfield mask; /* memory write abort
STA_MOD bitfield mask; /* state modified abort
EB_ABT bitfield mask; /* EBOX abort
FILL3 bitfield length 3 fill prefix MCF790 tag $$;
WBUS_CHK bitfield mask; /* WBUS to EDP error
EDP_PE bitfield mask; /* EBOX data path parity error
USTR_PE bitfield mask; /* EBOX microstack
ECS_PE bitfield mask; /* EBOX control store
EMCR_PE bitfield mask; /* EBOX memory control RAM
IBOX_ERR bitfield mask; /* IBOX hardware error
MBOX_INT bitfield mask; /* MBOX interrupt request

```

```

        MBOX_FE bitfield mask;      /* MBOX fatal error
    end EBCS_BITS_2;
    EBCS_BITS_3 structure;
        FILL2A bitfield fill prefix MCF790 tag $$;
        ABORTS bitfield mask length 4;
        FILL3A bitfield length 3 fill prefix MCF790 tag $$;
        DIAG_ERR bitfield mask;     /* diagnostic error flag
    end EBCS_BITS_3;
    end EBCS_OVERLAY_T;
    FILL4 bitfield length 4 fill prefix MCF790 tag $$;
    PME bitfield mask;              /* performance measurement enable
    FILL5 bitfield length 6 fill prefix MCF790 tag $$;
    ICS_EF bitfield mask;          /* IBOX control store error
    IDRAM_EF bitfield mask;        /* IBOX dispatch RAM error
    FBMCES_EF bitfield mask;       /* FBOX FBM control store error
    FBACS_EF bitfield mask;        /* FBOX FBA control store error
    FDRAM_EF bitfield mask;        /* FBOX dispatch RAM error
    end EBCS_BITS;
    end EBCS_OVERLAY;
    EDPSR_OVERLAY union;           /* EBOX data path status register
    EDPSR longword unsigned; /* entire register
    EDPSR_BITS structure;
        B_RAM_PE bitfield mask;     /* scratchpad to BMUX error
        A_WBUS_PE bitfield mask;    /* WBUS to AMUX error
        A_RAM_PE bitfield mask;     /* scratchpad to AMUX error
        OPER_CHK bitfield mask;     /* operand parity error
        FILL51 bitfield fill prefix MCF790 tag $$;
        RSLT_CHK bitfield mask;     /* result parity error
        B_OPBUS bitfield mask;      /* OPBUS to BMUX error
        B_WBUS bitfield mask;       /* WBUS to BMUX error
        EDP_MISC bitfield mask;     /* misc source parity error
        FILC6 bitfield length 2 fill prefix MCF790 tag $$;
        WREG bitfield mask;         /* W register parity error
        VMO_BYTE bitfield mask length 4; /* VMO byte in error
        FILC7 bitfield length 8 fill prefix MCF790 tag $$;
        AMX_BYTE bitfield mask length 4; /* AMUX byte in error
        BMX_BYTE bitfield mask length 4; /* BMUX byte in error
    end EDPSR_BITS;
    end EDPSR_OVERLAY;
    CSLINT_OVERLAY union;         /* console/interrupt register
    CSLINT longword unsigned; /* entire register
    CSLINT_BITS structure;
        CADR bitfield mask length 6; /* console bus address
        CWRT bitfield mask;         /* console bus write
        CCLK bitfield mask;        /* console bus clock
        CDAT bitfield mask length 8; /* console bus data
        IPR bitfield mask length 4; /* interrupt priority request level
        INT_SRC bitfield mask;      /* IPR due to internal source
        IOA bitfield mask length 2; /* I/O adapter with highest IPR
        CSL_TTX bitfield mask;      /* console terminal transmit
        CSL_TRX bitfield mask;      /* console terminal receive
        CSL_RL bitfield mask;       /* console RL
        INT_TMR bitfield mask;      /* interval timer interrupt
        INT_MBOX bitfield mask;     /* MBOX interrupt
        CPU_PF bitfield mask;       /* CPU powerfail interrupt
        CSL_HP bitfield mask;       /* console halt pending

```

```

end CSLINT BITS;
end CSLINT OVERLAY;
IBESR OVERLAY union; /* IBOX error/status register
  IBESR longword unsigned; /* entire register
  IBESR BITS structure;
    FILL8 bitfield length 8 fill prefix MCF790 tag $$;
    UOP_SEL bitfield mask length 2; /* OP BUS data source
    SRC_IMD bitfield mask; /* OP BUS source was IMD
    UTPR bitfield mask length 3; /* processor port causing microtrap
    FILL9 bitfield length 7 fill prefix MCF790 tag $$;
    ICS_PE bitfield mask; /* IBOX control store parity error
    IDRAM_PE bitfield mask; /* DRAM
    IAMUX_PE bitfield mask; /* AMUX when GPR selected
    RLOG_PE bitfield mask; /* unwinding RLOG
    IBUF_PE bitfield mask; /* error on byte-1, byte-0, or R-mode finder
    IBMUX_PE bitfield mask; /* output of ALU BMUX
    RSV_MODE bitfield mask; /* reserved mode
    IWBUS_PE bitfield mask; /* WBUS error detected by IBOX
    IAMUX_EC bitfield mask length 2; /*
  end IBESR BITS;
end IBESR OVERLAY;
EBXWD1 longword unsigned; /* EBOX write data 1
EBXWD2 longword unsigned; /* EBOX write data 2
IVASAV longword unsigned; /* virtual address for OP port requests
VIBASAV longword unsigned; /* virtual address of next IBUF port request
ESASAV longword unsigned; /* PC during EBOX execution and result storage
ISASAV longword unsigned; /* PC of instruction OP port working on
CPC longword unsigned; /* PC of instruction evaluated in IBUFFER
MSTAT1 OVERLAY union; /* MBOX status register 1
  MSTAT1 longword unsigned; /* entire register
  MSTAT1 BITS structure;
    CSR_DAT_BW bitfield mask; /* datapath parity error on byte write
    ARR_CYCC bitfield mask; /* error detected on array refill to cache
    CSH_ERR bitfield mask; /* indicates which cache had the error
    CSH_DAT_NBW bitfield mask; /* datapath parity error, non byte write
    WRT_DAT_PE bitfield mask length 4; /* MDBUS parity error on write data
    TB_TAG_PE bitfield mask; /* error on address tag
    TB_A_PE bitfield mask; /* error on PTE
    TB_B_PE bitfield mask; /* error on PTE
    TB_VAL_PE bitfield mask; /* error in valid bit
    CSR_HIT bitfield mask length 4; /* cache hit/miss history
    AB_ADPT bitfield mask length 2; /* ABUS adapter in error
    AB_CYCL bitfield mask; /* ABUS cycle in error
    AB_ADR_PE bitfield mask; /* ABUS physical address in error
    AB_CM_PE bitfield mask; /* ABUS cntrl/mask parity error
    AB_DAT_PE bitfield mask; /* ABUS data parity error
    CPR_PE_A bitfield mask; /* cycle parameter RAM error (A)
    CPR_PE_B bitfield mask; /* cycle parameter RAM error (B)
    WDCNT bitfield mask length 2; /* longword in error
    CYCLE_TYP bitfield mask length 4; /* MBOX cycle type
    DEST_CP bitfield mask length 2; /* port being serviced
  end MSTAT1 BITS;
end MSTAT1 OVERLAY;
MSTAT2 OVERLAY union; /* MBOX status register 2
  MSTAT2 longword unsigned; /* entire register
  MSTAT2 BITS structure;

```

```

FILL95 bitfield length 1 fill prefix MCF790 tag $$;
MBOX_LCK bitfield mask; /* error while lock asserted
CP_IO_BUF bitfield mask; /* error on CPU to IO request
NXM bitfield mask; /* non-existent memory
CSH_W bitfield mask; /* selected cache entry was modified
CSH_TAG_W bitfield mask; /* error in cache written bit
CSH_TAG_PE bitfield mask; /* error in cache tag
MUL_ERR bitfield mask; /* multiple MBOX errors
SBIA_STAT bitfield mask length 6; /* SBIA diagnostic status
AB_BAD_DAT bitfield mask; /* ABUS bad data flag received
SBIA_CPBW bitfield mask; /* SBIA error was on CP byte write
PAMM_DATA bitfield mask length 4; /* PAMM code
PAMM_CACHE bitfield mask; /* PAMM cache disable bit
end MSTAT2 BITS;
end MSTAT2 OVERLAY;
MDECC OVERLAY union; /* MBOX data ECC register
MDECC longword unsigned; /* entire register
MDECC BITS structure;
ECC_DIAG bitfield mask length 8; /* force errors
FILC115 bitfield length 1 fill prefix MCF790 tag $$;
SYNDRM bitfield mask length 6; /* error data syndrome
PAR_INV bitfield mask; /* indicates parity is being inverted
FILC11 bitfield length 3 fill prefix MCF790 tag $$;
ADR_PE bitfield mask; /* data address parity error
DBL_BIT bitfield mask; /* double bit error
SNG_ERR bitfield mask; /* single bit error
BAD_DATA bitfield mask; /* bad data flag
DATA_MUL bitfield mask; /* multiple errors
end MDECC BITS;
end MDECC OVERLAY;
MERG longword unsigned; /* MBOX error generator register
CSHCTL OVERLAY union; /* MBOX cache control register
CSRCTL longword unsigned; /* entire register
CSHCTL BITS structure;
CSH_0_ENB bitfield mask; /* cache 0 enable
CSH_1_ENB bitfield mask; /* cache 1 enable
FRC_HIT bitfield mask; /* force cache hit
FRC_MISS bitfield mask; /* force cache miss
end CSHCTL BITS;
end CSHCTL OVERLAY;
MEDR longword unsigned; /* data word used during error
MEAR longword unsigned; /* physical address in latch during error
FBXERR OVERLAY union; /* FBOX error register
FBXERR longword unsigned; /* entire register
FBXERR BITS structure;
FBOX_ERR bitfield; /* There is an error - rest of bits valid
FILLT2 bitfield length 1 fill prefix MCF790 tag $$;
TEST bitfield mask; /* error during self test
FILLT3 bitfield length 11 fill prefix MCF790 tag $$;
DATA_TYP bitfield mask length 2; /* data type during error
FILLT4 bitfield length 1 fill prefix MCF790 tag $$;
FBOX_GPR bitfield mask; /* error reading scratchpad
FBOX_SLF bitfield mask; /* error during self test
FBOX_DRAM bitfield mask; /* DRAM parity error
FBOX_FBA_CS bitfield mask; /* error in adder control store
FBOX_FBM_CS bitfield mask; /* error in multiplier control store

```



```

end FBXERR BITS;
end FBXERR_OVERLAY;
CSES longword unsigned; /* control store error status register
PC longword unsigned;
PSL longword unsigned;

```

```
/* MBOX cycle types
```

```

constant(
  NOP, /* read register
  READ_REG, /* write register
  WRITE_REG, /* write back
  WRITE_BAK, /* ABUS array write
  ABUS_WRT, /* data correction
  DATA_COR, /* clear cache
  CLR_CSH, /* TB probe
  TB_PROBE, /* ABUS
  ABUS, /* CP refill
  CP_REFL, /* invalidate TB
  INVAL_TB, /* TB cycle
  TB_CYCLE, /* CP byte write
  CP_BYT_WRT, /* CP write
  CP_WRT, /* CP read
  CP_READ, /* ABUS refill
  ABUS_REFL
) equals 0 increment 1 prefix MCF790 tag $C;

```

```
/* DEST CP (port) codes
```

```

constant(
  IBF_PORT_0, /* IBUF port
  OP_PORT, /* OP fetch port
  EBOX_PORT, /* EBOX port
  IBF_PORT_3 /* IBUF port
) equals 0 increment 1 prefix MCF790 tag $C;

```

```
/* VMS machine check service codes
```

```

constant(
  FBOX, /* FBOX
  EBOX, /* EBOX
  IBOX, /* IBOX
  MBOX_FE /* MBOX fatal error
) equals 1 increment 1 prefix MCF790 tag $C;

```

```

end MCF790;
end_module $MCF790DEF;

```


001	002	003	004	005	006	007	008	009	010	011	012	013	014	015	016	017	018	019	020	021	022	023	024	025	026	027	028	029	030	031	032	033	034	035	036	037	038	039	040	041	042	043	044	045	046	047	048	049	050	051	052	053	054	055	056	057	058	059	060	061	062	063	064	065	066	067	068	069	070	071	072	073	074	075	076	077	078	079	080	081	082	083	084	085	086	087	088	089	090	091	092	093	094	095	096	097	098	099	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600	601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700	701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800	801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899	900	901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991	992	993	994	995	996	997	998	999	1000
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------

ADPSUB780
LIS

ACKMSG
LIS

MCF790
SQL

MCDEF
MDL

ADPERR750
LIS

ADPSUB730
LIS

CSPDEF
SQL

CLUMBX
SQL

ADPERR780
LIS

ADPSUB750
LIS

CLUSTMAC
MAR

CLUSTER
SQL