


```

CCCCCCCC  SSSSSSSS  PPPPPPPP  DDDDDDDD  EEEEEEEEE  FFFFFFFF
CCCCCCCC  SSSSSSSS  PPPPPPPP  DDDDDDDD  EEEEEEEEE  FFFFFFFF
CC        SS        PP        PP  DD        DD  EE        FF
CC        SS        PP        PP  DD        DD  EE        FF
CC        SS        PP        PP  DD        DD  EE        FF
CC        SS        PP        PP  DD        DD  EE        FF
CC        SSSSSS  PPPPPPPP  DD        DD  EEEEEEEE  FFFFFFFF
CC        SSSSSS  PPPPPPPP  DD        DD  EEEEEEEE  FFFFFFFF
CC        SS        PP        DD        DD  EE        FF
CC        SS        PP        DD        DD  EE        FF
CC        SS        PP        DD        DD  EE        FF
CCCCCCCC  SSSSSSSS  PP        DD        DD  EE        FF
CCCCCCCC  SSSSSSSS  PP        DDDDDDDD  EEEEEEEEE  FF

```

```

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

```

```
{ module      %CSPDEF;
{ 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 Cluster Server Process
{* ABSTRACT:    CSP Internal Data Structure Definitions
{* AUTHOR:      Paul R. Beck
{* DATE:        3-MAR-1983      Last Edit: 29-JUL-1983 17:47:11
```

```
{* REVISION HISTORY:
```

```
{* V03-007 ADE0002      Alan D. Eldridge      28-Feb-1983
{*       Add CSP%_LOCAL and CSD%K_QUORUM.
{*
{* V03-006 ADE0001      Alan D. Eldridge      7-Dec-1983
{*       Add CLX$A_PO_CSD and CLX$A_SO_CSD.  Removed CLX$W_NETCHAN.
{*       Add CLX$L_P1 thru P4 for use by client code.
{*
{*       Add CSD$W_STS, remove CSD$W_NET_CHAN, and rearrange fields
{*       for proper alignment.
{*
{*       Add CLX$V_RESUME_REQ and changed CLX$V_MUTEX semantics.
{*       Add %CSPDEF codes.
{*
{* V03-005 PRB0249      Paul Beck              9-Sep-1983 18:11
{*       Add CSD$L_P1 : CSD$L_P8 default offset definitions.
{*
{* V03-004 CWH3001      CW Hobbs                30-Jul-1983
```

Add OPCOM and MOUNT clients.

V03-003 PRB0232 Paul Beck 14-JUL-1983 11:36
Add support for builtin stack in CLX structure.

V03-002 PRB0202 Paul Beck 19-MAY-1983 22:39.49
Add client constant for RCP control.
Change "CTX" to "CLX".

V03-001 JLV0248 Jake VanNoy 29-APR-1983
Add offsets for BRKTHRU. Add client constants.

```
MODULE $CSPDEF ;
/*
/* CSP - Codes used for communication between the CSP process and the loadable
/*   Exec code
/*
CONSTANT
(nop          /* No-op. Should never be used
,init        /* Initialize loadable code and data
,abort       /* CSP process is gone or going away
,badcscd     /* Something wrong with CSD structure
,done        /* Block transfer exchange has finished
,reply       /* Send a block transfer and terminate
,reject      /* Don't engage in block transfer
,local      /* Request to pass local CSD to CSP
) EQUALS 0 INCREMENT 1
  PREFIX csp TAG $ ;

CONSTANT csp$k_max_flwctl EQUALS 8 ; /* Maximum allowed active requests
CONSTANT csp$k_max_csdlnq EQUALS 4096 ; /* Maximum allowed CSD length

END_MODULE ;
```

```

MODULE $CLXDEF;          /*
/**
/* CLX - Scheduling context for CSP
/*
/* Registers are saved by dint of the CALL to the routine which
/* creates the CLX block, and are restored by the RET. R0 and R1
/* must be saved and restored explicitly.
/*
/*-
AGGREGATE $clxdef STRUCTURE PREFIX clx$:

  flink      ADDRESS;          /* forward link
  blink      ADDRESS;          /* reverse link
  size       WORD UNSIGNED;    /* size of structure
  type       BYTE UNSIGNED;    /* structure type (DYN$C_XXX)

  flags_overlay UNION;
    flags     BYTE UNSIGNED;    /* Flags
    flags_bits STRUCTURE;      /*
      queued  BITFIELD MASK;    /* CLX queue linkage in used
      mutex   BITFIELD MASK;    /* mutex between RESUME and WAIT routine
      resume_req BITFIELD MASK; /* signal flag between RESUME and WAIT
      local_stack BITFIELD MASK; /* local stack in use.
      filler  BITFIELD LENGTH 8-^ FILL;
  END        flags_bits;
END          flags_overlay;

  s0_csd     ADDRESS;          /* Pointer to S0 space CSD
  p0_csd     ADDRESS;          /* Pointer to P0 space CSD
  P1         LONGWORD UNSIGNED; /* Parameter for use by client code
  P2         LONGWORD UNSIGNED; /* Parameter for use by client code
  P3         LONGWORD UNSIGNED; /* Parameter for use by client code
  P4         LONGWORD UNSIGNED; /* Parameter for use by client code
  index      LONGWORD UNSIGNED; /* context ID for debugging purposes
  r0         LONGWORD UNSIGNED; /* saved R0
  r1         LONGWORD UNSIGNED; /* saved R1

  iosb_overlay UNION;
    iosb      QUADWORD UNSIGNED; /* I/O status block
    iosb_structure STRUCTURE;
      iosb_stat WORD UNSIGNED;    /* IOSB completion status
      iosb_size WORD UNSIGNED;    /* IOSB size of completed operation
      iosb_devdata LONGWORD UNSIGNED; /* IOSB device-dependent data
  END        iosb_structure;
END          iosb_overlay;

  stacksize  LONGWORD UNSIGNED; /* size of saved stack + registers
  stack      ADDRESS;          /* addr of saved stack area
  stackblock STRUCTURE;        /* allow local stack of 64 longwords
    local_stack BYTE UNSIGNED
              DIMENSION 256 TAG B;
END stackblock;

CONSTANT local_stack EQUALS 256;
CONSTANT "length" EQUALS .;    /* length of data structure

```



```

MODULE %CSDDEF;          /*
/**
/* CSD - Cluster Server Data
/*
/* Defines client data to send to the cluster server in the indicated node.
/*-
AGGREGATE %csddef STRUCTURE PREFIX csd$:

  messages      UNION;

  maincsd STRUCTURE;

  flink          ADDRESS;          /* forward link
  blink          ADDRESS;          /* reverse link
  size           WORD UNSIGNED;    /* size of structure
  type           BYTE UNSIGNED;    /* structure type (DYN$C-CLU)
  subtype        BYTE UNSIGNED;    /* structure subtype (DYN$C-CSD)
  constant       DYN$C-CSD equals 100 prefix ' ' tag ' '; /* *** TEMPORARY ***

/*
/* Caller-supplied data
/*
  code           WORD UNSIGNED;    /* client code - who gets the msg in CSP
  csid_overlay   UNION;
    csid          LONGWORD UNSIGNED; /* Cluster ID
    csid_structure STRUCTURE;
      csid_index  WORD UNSIGNED;    /* index into CLU$GL-CLUSVEC
      csid_seq     WORD UNSIGNED;    /* sequence number
    END          csid_structure;
  END csid_overlay;

  sendlen        LONGWORD UNSIGNED; /* length of send buffer
  sendoff        LONGWORD UNSIGNED; /* offset (self relative) to send buffer
  recvlen        LONGWORD UNSIGNED; /* length of rcv buffer
  rcrvoff        LONGWORD UNSIGNED; /* offset (self relative) to rcv buffer
  *astadr        ADDRESS;          /* address of caller AST
  **astadr        ADDRESS;          /* address of caller's caller's AST
  user_astprm    LONGWORD UNSIGNED; /* param...
  user_iosb      ADDRESS;          /* caller's caller's IOSB
  user_efn       BYTE UNSIGNED;    /*
  efn            BYTE UNSIGNED;

/*
/* Internal data (filled in by EXE$ALLOC-CSD)
/*
  sts_overlay    UNION;
    sts          WORD UNSIGNED;    /* Status flags
    sts_bits     STRUCTURE;
      sts_rem    BITFIELD MASK ; /* Request is from remote node
    END          sts_bits;
  END            sts_overlay;

  ipid_overlay   UNION;
    ipid         LONGWORD UNSIGNED; /* Local Process ID
    ipid_structure STRUCTURE;
      ipid_index  WORD UNSIGNED;    /* index
      ipid_seq     WORD UNSIGNED;    /* sequence number
    END          ipid_structure;

```



```

END      ipid_overlay;

iosb_overlay      UNION;
  int_iosb      QUADWORD UNSIGNED; /* Internal I/O status block
  iosb_structure      STRUCTURE;
  iosb_stat      WORD UNSIGNED; /* IOSB completion status
  iosb_size      WORD UNSIGNED; /* IOSB size of completed operation
  iosb_devdata      LONGWORD UNSIGNED; /* IOSB device-dependent data
END      iosb_structure;
END      iosb_overlay;

procpriv      QUADWORD UNSIGNED; /* process privileges of caller
procuic      LONGWORD UNSIGNED; /* UIC of caller
imgcnt      LONGWORD UNSIGNED; /* used to detect new image

/*
/* Start of caller-supplied message
/*
CONSTANT      length EQUALS .; /* start of buffers
/*
/* Define some default of sts for client-supplied data
/*
client_data UNION;
  data      BYTE UNSIGNED TAG 'AB'; /* ...
  default_data STRUCTURE; /* allow clients to specify...
    p1      LONGWORD UNSIGNED; /* ... CSD$P1 - CSD$P8
    p2      LONGWORD UNSIGNED; /* ... so they don't need to edit this
    p3      LONGWORD UNSIGNED; /* ... file.
    p4      LONGWORD UNSIGNED;
    p5      LONGWORD UNSIGNED;
    p6      LONGWORD UNSIGNED;
    p7      LONGWORD UNSIGNED;
    p8      LONGWORD UNSIGNED;
  END      default_data;
END      client_data;

END maincsd;

/*
/* Client codes
/*
CONSTANT ( /* caller codes
  test,      { test vector
  brkthru,   { break though write
  rcpc,      { RCP control
  jnl_master, { Journal failover (remastering)
  opc0m,     { Operator communications
  mount,     { Cluster mount
  quorum     { Quorum disk support
) EQUALS 1 INCREMENT 1;

/*
/* $BRKTHRU message definitions
/*
brkthru STRUCTURE;
  brk_carcon      LONGWORD UNSIGNED; /* carriage control

```

```

:++
:
:
:
:
: 0
:
:--

```

```
brk_flags      LONGWORD UNSIGNED; /* flags
brk_sendto     CHARACTER LENGTH 16; /* counted string
brk_sndtyp     WORD UNSIGNED; /* send to value
brk_reqid      WORD UNSIGNED; /* class request id
brk_timeout    WORD UNSIGNED; /* seconds of timeout
brk_msglen     WORD UNSIGNED; /* message length
brk_msgbuf     CHARACTER LENGTH 0; /* text string
```

```
END brkthru;
```

```
/*
/* ... next caller def goes here...
/*
```

```
END messages; /* end of main UNION
```

```
END %csddef;
```

```
END_MODULE;
```


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