


```

      AAAAAA  NN      NN      AAAAAA  LL      YY      YY  ZZZZZZZZZZ  EEEEEEEEEEE
      AAAAAA  NN      NN      AAAAAA  LL      YY      YY  ZZZZZZZZZZ  EEEEEEEEEEE
AA      AA  NN      NN      AA      AA  LL      YY      YY  ZZ      EE
AA      AA  NN      NN      AA      AA  LL      YY      YY  ZZ      EE
AA      AA  NNNN     NN      AA      AA  LL      YY      YY  ZZ      EE
AA      AA  NNNN     NN      AA      AA  LL      YY      YY  ZZ      EE
AA      AA  NN      NN      AA      AA  LL      YY      YY  ZZ      EE
AA      AA  NN      NN      AA      AA  LL      YY      YY  ZZ      EE
AAAAAAAAAA  NN      NNNN  AAAAAAAAAA  LL      YY      YY  ZZ      EEEEEEEEE
AAAAAAAAAA  NN      NNNN  AAAAAAAAAA  LL      YY      YY  ZZ      EEEEEEEEE
AA      AA  NN      NN      AA      AA  LL      YY      YY  ZZ      EE
AA      AA  NN      NN      AA      AA  LL      YY      YY  ZZ      EE
AA      AA  NN      NN      AA      AA  LL      YY      YY  ZZ      EE
AA      AA  NN      NN      AA      AA  LLLLLLLLLL  YY      YY  ZZZZZZZZZZ  EEEEEEEEEEE
AA      AA  NN      NN      AA      AA  LLLLLLLLLL  YY      YY  ZZZZZZZZZZ  EEEEEEEEEEE

```

```

LL      IIIIII  SSSSSSSS
LL      IIIIII  SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLLLL  IIIIII  SSSSSSSS

```

```

1 0001 0 MODULE ANALYZE (%TITLE 'Analyze a save set'
2 0002 0 IDENT = 'V04-000'
3 0003 0 ) =
4 0004 1 BEGIN
5 0005 1
6 0006 1
7 0007 1 *****
8 0008 1 *
9 0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
10 0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
11 0011 1 * ALL RIGHTS RESERVED. *
12 0012 1 *
13 0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
14 0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
15 0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
16 0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
17 0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
18 0018 1 * TRANSFERRED. *
19 0019 1 *
20 0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
21 0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
22 0022 1 * CORPORATION. *
23 0023 1 *
24 0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
25 0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
26 0026 1 *
27 0027 1 *
28 0028 1 *****
29 0029 1
30 0030 1
31 0031 1 ++
32 0032 1 FACILITY:
33 0033 1 Backup/Restore
34 0034 1
35 0035 1 ABSTRACT:
36 0036 1 This module contains the routines that analyze a save set.
37 0037 1
38 0038 1 ENVIRONMENT:
39 0039 1 VAX/VMS user mode.
40 0040 1 --
41 0041 1
42 0042 1 AUTHOR: M. Jack, CREATION DATE: 03-Sep-1980
43 0043 1
44 0044 1 MODIFIED BY:
45 0045 1
46 0046 1 V03-009 LY0510 Larry Yetto 19-JUL-1984 08:44
47 0047 1 Add support for the new longword devtyp in the physical
48 0048 1 volume attributes record. The format of this longword is
49 0049 1 the same as UCBSL_MEDIA_ID
50 0050 1
51 0051 1 V03-008 LY0485 Larry Yetto 27-APR-1984 08:42
52 0052 1 FT1 QAR # 2088 - If the saveset being read is encrypted
53 0053 1 and /ENCRYPT not specified then report an error
54 0054 1
55 0055 1 V03-007 LMP0176 L. Mark Pilant, 6-Dec-1983 10:41
56 0056 1 Use the correct width in the call to $FORMAT_ACL.
57 0057 1

```

58	0058	1	V03-006	JWT0137	Jim Teague	19-Sep-1983	08:22
59	0059	1		CRYPTO_INIDEC and CRYPTO_DECR_BLOCK need to be WEAKened.			
60	0060	1					
61	0061	1	V03-005	JEP0003	J. Eric Pollack,	23-Apr-1983	10:53
62	0062	1		Add support for encrypted savesets.			
63	0063	1					
64	0064	1	V03-004	ACG0332	Andrew C. Goldstein,	19-Apr-1983	18:10
65	0065	1		Add support for file highwater mark and RMS journal flags			
66	0066	1					
67	0067	1	V03-003	LMP0100	L. Mark Pilant,	14-Apr-1983	13:17
68	0068	1		Add te \$FORMAT_ACL system service.			
69	0069	1					
70	0070	1	V03-002	ACG0313	Andrew C. Goldstein,	12-Feb-1983	16:01
71	0071	1		Add routine subtitles			
72	0072	1					
73	0073	1	V03-001	LMP0044	L. Mark Pilant,	21-Oct-1982	15:10
74	0074	1		Add support for ACL's.			
75	0075	1					
76	0076	1	V02-006	MLJ0081	Martin L. Jack,	26-Feb-1982	16:16
77	0077	1		Add RETAINMIN and RETAINMAX attributes to support new home block fields.			
78	0078	1					
79	0079	1					
80	0080	1	V02-005	MLJ0075	Martin L. Jack,	28-Jan-1982	20:02
81	0081	1		Add VERLIMIT and DIR_VERLIM attributes to support version limit handling.			
82	0082	1					
83	0083	1					
84	0084	1	V02-004	MLJ0062	Martin L. Jack,	3-Dec-1981	12:16
85	0085	1		Add DIR_STATUS attribute to support /INCREMENTAL.			
86	0086	1					
87	0087	1	V02-003	MLJ0036	Martin L. Jack,	28-Aug-1981	17:09
88	0088	1		Implement parent directory attributes and reel restart.			
89	0089	1					
90	0090	1	V02-002	MLJ0023	Martin L. Jack,	23-Apr-1981	11:36
91	0091	1		Implement placement attribute.			
92	0092	1					
93	0093	1	V02-001	MLJ0010	Martin L. Jack,	25-Mar-1981	14:58
94	0094	1		Add new attributes for image restore. Make some routines common with LIST module. Replace OWN storage with LOCAL. Change !SL directives to !UL.			
95	0095	1					
96	0096	1					
97	0097	1					
98	0098	1					**

```

100 0099 1 REQUIRE 'SRCS:COMMON';
101 1205 1 LIBRARY 'SYSS$LIBRARY:STARLET';
102 1206 1 REQUIRE 'LIBS:BACKDEF';
103 1656 1
104 1657 1
105 1658 1 LINKAGE
106 1659 1     L_PS =          CALL: GLOBAL(PS=11);
107 1660 1
108 1661 1
109 1662 1 MACRO
110 1663 1     L_DECL =        EXTERNAL REGISTER PS = 11: REF VECTOR %;
111 1664 1
112 1665 1
113 1666 1 FORWARD ROUTINE
114 1667 1     ANALYZE_ONE_ATTRIBUTE:
115 1668 1         L_PS NOVALUE,      ! Format an attribute record
116 1669 1     ANALYZE_ONE_RECORD:
117 1670 1         L_PS NOVALUE,      ! Format a record
118 1671 1     ANALYZE_ONE_BUFFER:
119 1672 1         L_PS NOVALUE,      ! Format a block
120 1673 1     ANALYZE:          NOVALUE;      ! Driver for save set analysis
121 1674 1
122 1675 1
123 1676 1 EXTERNAL ROUTINE
124 1677 1     DEBLOCK:          L_PS NOVALUE,      ! Deblock a save set buffer
125 1678 1     DEBLOCK_ATTR:    L_PS NOVALUE,      ! Deblock an attribute record
126 1679 1     DECODE_DEVTYP:  NOVALUE,          ! Convert DEVTYP to ASCII string
127 1680 1     FIN_IN_SAVE:    NOVALUE,          ! Finish input save set processing
128 1681 1     INIT_IN_SAVE:   NOVALUE,          ! Initialize input save set processing
129 1682 1     LIST_FAO:        L_PS NOVALUE,      ! Add information to line buffer
130 1683 1     LIST_EOL:        L_PS NOVALUE,      ! Write line buffer to listing file
131 1684 1     LIST_PROTECTION:L_PS NOVALUE,      ! List protection code
132 1685 1     READ_BUFFER,     ! Get a save set buffer
133 1686 1     RESTORE_HANDLER, ! Handler for RESTORE, LIST, ANALYZE
134 1687 1     CRYPTO_INIDEC:  WEAK,              ! Initialize for reading encrypted saveset
135 1688 1     CRYPTO_CHKSAV,  ! Check if saveset is encrypted
136 1689 1     CRYPTO_DECR_BLOCK: NOVALUE
137 1690 1         WEAK;          ! Decrypt one block
138 1691 1
139 1692 1
140 1693 1 G$DEFINE();          ! Define global common area
141 1694 1
142 1695 1
143 1696 1 BIND
144 1697 1     FALSETRUE = UPLIT (
145 1698 1         UPLIT BYTE (%ASCIC 'False'),
146 1699 1         UPLIT BYTE (%ASCIC 'True'))
147 1700 1     : VECTOR;
148 1701 1
149 1702 1
150 1703 1 EXTERNAL LITERAL
151 1704 1     BACKUP$_BACNOTENC,
152 1705 1     BACKUP$_ENC$SAVSET;
153 1706 1
154 1707 1 MACRO
155 M 1708 1     FAO_(A)=
156 M 1709 1     _LIST_FAO(

```

```
: 157      M 1710 1      UPLIT BYTE (%ASCIC A)
: 158      1711 1      %IF NOT %NULL(%REMAINING) %THEN , %FI %REMAINING) %,
: 159      1712 1
: 160      1713 1
: 161      M 1714 1      EOL_(A)=
: 162      1715 1      _LIST_EOL() %;
: 163      1716 1
: 164      1717 1
: 165      1718 1  MACRO
: 166      1719 1      _LIST_DESC=      PS[0] %,      ! Descriptor for buffer
: 167      1720 1      _LIST_BUFFER=    PS[2] %,      ! Listing buffer
: 168      1721 1      [IST_DESC(N)=    VECTOR[_LIST_DESC,N] %;
: 169      1722 1
: 170      1723 1
: 171      1724 1  LITERAL
: 172      1725 1      PSSIZE=          2 + CH$ALLOCATION(LIST_SIZE);
```

```

174 1726 1 %SBTTL 'ANALYZE_ONE_ATTRIBUTE - analyze contents of attribute record'
175 1727 1 ROUTINE ANALYZE_ONE_ATTRIBUTE(ATT): L_PS NOVALUE=
176 1728 1
177 1729 1 !++
178 1730 1
179 1731 1 FUNCTIONAL DESCRIPTION:
180 1732 1 This routine analyzes the contents of one attribute record.
181 1733 1
182 1734 1 INPUT PARAMETERS:
183 1735 1 ATT - Pointer to attribute record.
184 1736 1
185 1737 1 IMPLICIT INPUTS:
186 1738 1 NONE
187 1739 1
188 1740 1 OUTPUT PARAMETERS:
189 1741 1 NONE
190 1742 1
191 1743 1 IMPLICIT OUTPUTS:
192 1744 1 NONE
193 1745 1
194 1746 1 ROUTINE VALUE:
195 1747 1 NONE
196 1748 1
197 1749 1 SIDE EFFECTS:
198 1750 1 The listing is produced.
199 1751 1
200 1752 1 --
201 1753 1
202 1754 2 BEGIN
203 1755 2
204 1756 2 LITERAL
205 1757 2 DEVTYP_BUF_LEN = 5 ;
206 1758 2
207 1759 2 MAP
208 1760 2 ATT: REF BBLOCK; ! Pointer to attribute record
209 1761 2
210 1762 2 BIND
211 1763 2 ATTRS = UPLIT (
212 1764 2 UPLIT BYTE (%ASCIC 'SSNAME'),
213 1765 2 UPLIT BYTE (%ASCIC 'COMMAND'),
214 1766 2 UPLIT BYTE (%ASCIC 'COMMENT'),
215 1767 2 UPLIT BYTE (%ASCIC 'USERNAME'),
216 1768 2 UPLIT BYTE (%ASCIC 'USERUIC'),
217 1769 2 UPLIT BYTE (%ASCIC 'DATE'),
218 1770 2 UPLIT BYTE (%ASCIC 'OPSYS'),
219 1771 2 UPLIT BYTE (%ASCIC 'SYSVER'),
220 1772 2 UPLIT BYTE (%ASCIC 'NODENAME'),
221 1773 2 UPLIT BYTE (%ASCIC 'SIR'),
222 1774 2 UPLIT BYTE (%ASCIC 'DPIVEID'),
223 1775 2 UPLIT BYTE (%ASCIC 'BACKVER'),
224 1776 2 UPLIT BYTE (%ASCIC 'BLOCKSIZE'),
225 1777 2 UPLIT BYTE (%ASCIC 'XORSIZE'),
226 1778 2 UPLIT BYTE (%ASCIC 'BUFFERS'),
227 1779 2 UPLIT BYTE (%ASCIC 'VOLSETNAM'),
228 1780 2 UPLIT BYTE (%ASCIC 'NVOLS'),
229 1781 2 UPLIT BYTE (%ASCIC 'BACKSIZE'),
230 1782 2 UPLIT BYTE (%ASCIC 'BACKFILES'),

```

231	1783	2	UPLIT BYTE (%ASCIC 'VOLSTRUCT'),
232	1784	2	UPLIT BYTE (%ASCIC 'VOLNAME'),
233	1785	2	UPLIT BYTE (%ASCIC 'OWNERNAME'),
234	1786	2	UPLIT BYTE (%ASCIC 'FORMAT'),
235	1787	2	UPLIT BYTE (%ASCIC 'RVN'),
236	1788	2	UPLIT BYTE (%ASCIC 'VOLOWNER'),
237	1789	2	UPLIT BYTE (%ASCIC 'PROTECT'),
238	1790	2	UPLIT BYTE (%ASCIC 'FILEPROT'),
239	1791	2	UPLIT BYTE (%ASCIC 'RECPROT'),
240	1792	2	UPLIT BYTE (%ASCIC 'VOLCHAR'),
241	1793	2	UPLIT BYTE (%ASCIC 'VOLDATE'),
242	1794	2	UPLIT BYTE (%ASCIC 'WINDOW'),
243	1795	2	UPLIT BYTE (%ASCIC 'LRU LIM'),
244	1796	2	UPLIT BYTE (%ASCIC 'EXTEND'),
245	1797	2	UPLIT BYTE (%ASCIC 'CLUSTER'),
246	1798	2	UPLIT BYTE (%ASCIC 'RESFILES'),
247	1799	2	UPLIT BYTE (%ASCIC 'VOLSIZE'),
248	1800	2	UPLIT BYTE (%ASCIC 'TOTSIZI'),
249	1801	2	UPLIT BYTE (%ASCIC 'TOTFILES'),
250	1802	2	UPLIT BYTE (%ASCIC 'MAXFILES'),
251	1803	2	UPLIT BYTE (%ASCIC 'MAXFILNUM'),
252	1804	2	UPLIT BYTE (%ASCIC 'SERIALNUM'),
253	1805	2	UPLIT BYTE (%ASCIC 'FILENAME'),
254	1806	2	UPLIT BYTE (%ASCIC 'STRUCLEV'),
255	1807	2	UPLIT BYTE (%ASCIC 'FID'),
256	1808	2	UPLIT BYTE (%ASCIC 'BACKLINK'),
257	1809	2	UPLIT BYTE (%ASCIC 'FILESIZE'),
258	1810	2	UPLIT BYTE (%ASCIC 'UIC'),
259	1811	2	UPLIT BYTE (%ASCIC 'FPRO'),
260	1812	2	UPLIT BYTE (%ASCIC 'RPRO'),
261	1813	2	UPLIT BYTE (%ASCIC 'ACLEVEL'),
262	1814	2	UPLIT BYTE (%ASCIC 'UCHAR'),
263	1815	2	UPLIT BYTE (%ASCIC 'RECATTR'),
264	1816	2	UPLIT BYTE (%ASCIC 'REVISION'),
265	1817	2	UPLIT BYTE (%ASCIC 'CREDATE'),
266	1818	2	UPLIT BYTE (%ASCIC 'REVDATI'),
267	1819	2	UPLIT BYTE (%ASCIC 'EXPDATE'),
268	1820	2	UPLIT BYTE (%ASCIC 'BAKDATE'),
269	1821	2	UPLIT BYTE (%ASCIC 'SECTORS'),
270	1822	2	UPLIT BYTE (%ASCIC 'TRACKS'),
271	1823	2	UPLIT BYTE (%ASCIC 'CYLINDERS'),
272	1824	2	UPLIT BYTE (%ASCIC 'MAXBLOCK'),
273	1825	2	UPLIT BYTE (%ASCIC 'DEVTYP'),
274	1826	2	UPLIT BYTE (%ASCIC 'SERIAL'),
275	1827	2	UPLIT BYTE (%ASCIC 'DEVNAM'),
276	1828	2	UPLIT BYTE (%ASCIC 'LABEL'),
277	1829	2	UPLIT BYTE (%ASCIC 'BADBLOCK'),
278	1830	2	UPLIT BYTE (%ASCIC 'INDEXLBN'),
279	1831	2	UPLIT BYTE (%ASCIC 'BOOTBLOCK'),
280	1832	2	UPLIT BYTE (%ASCIC 'BOOTVBN'),
281	1833	2	UPLIT BYTE (%ASCIC 'PLACEMENT'),
282	1834	2	UPLIT BYTE (%ASCIC 'DIR_UIC'),
283	1835	2	UPLIT BYTE (%ASCIC 'DIR_FPRO'),
284	1836	2	UPLIT BYTE (%ASCIC 'DIR_STATUS'),
285	1837	2	UPLIT BYTE (%ASCIC 'DIR_VERLIM'),
286	1838	2	UPLIT BYTE (%ASCIC 'VERLIMIT'),
287	1839	2	UPLIT BYTE (%ASCIC 'RETAINMIN'),


```

288      1840      2      UPLIT BYTE (%ASCIC 'RETAINMAX'),
289      1841      2      UPLIT BYTE (%ASCIC 'ACLSEGMENT'),
290      1842      2      UPLIT BYTE (%ASCIC 'HIGHWATER'),
291      1843      2      UPLIT BYTE (%ASCIC 'JNL_FLAGS'),
292      1844      2      UPLIT BYTE (%ASCIC 'CRYPDATKEY'))
293      1845      2      : VECTOR;
294      1846      2      L_DECL;
295      1847      2
296      1848      2
297      1849      2      ! List the attribute name.
298      1850      2      !
299      P 1851      2      FAO ('      SIZE = !3SL, TYPE = !AC',
300      1852      2      .ATT[BSASW_SIZE], .ATTRS[.ATT[BSASW_TYPE]-1]);
301      1853      2      EOL_();
302      1854      2
303      1855      2
304      1856      2      ! List the attribute value in an appropriate format.
305      1857      2      !
306      1858      2      FAO ('      ');
307      1859      2      CASE .ATT[BSASW_TYPE] FROM BSASK_SSNAME TO BSASK_NUM_ATTRS-1 OF
308      1860      2      SET
309      1861      2
310      1862      2      [BSASK_SSNAME, BSASK_COMMAND, BSASK_COMMENT, BSASK_USERNAME, BSASK_SYSVAR,
311      1863      2      BSASK_BACKVER, BSASK_NODENAME, BSASK_DRIVEID, BSASK_VOLSETNAM,
312      1864      2      BSASK_VOLNAME, BSASK_OWNERNAME, BSASK_FORMAT, BSASK_FILENAME,
313      1865      2      BSASK_DEVNAM, BSASK_LABEL]:
314      P 1866      2      FAO ('!AF',
315      1867      2      .ATT[BSASW_SIZE], ATT[BSASC_LENGTH,0,0,0]);
316      1868      2
317      1869      2      [BSASK_USERUIC, BSASK_VOLOWNER, BSASK_UIC, BSASK_DIR_UIC]:
318      P 1870      2      FAO ('!XU',
319      1871      2      .ATT[BSASC_LENGTH,0,32,0]);
320      1872      2
321      1873      2      [BSASK_DATE, BSASK_VOLDATE, BSASK_CREDATE, BSASK_REVDATE, BSASK_EXPDATE,
322      1874      2      BSASK_BAKDATE, BSASK_RETAINMIN, BSASK_RETAINMAX]:
323      P 1875      2      FAO ('!XD',
324      1876      2      .ATT[BSASC_LENGTH,0,0,0]);
325      1877      2
326      1878      2      [BSASK_BLOCKSIZE, BSASK_XORSIZE, BSASK_BUFFERS, BSASK_NVOLS,
327      1879      2      BSASK_BACKFILES, BSASK_RVN, BSASK_WINDOW, BSASK_LRU_LIM, BSASK_EXTEND,
328      1880      2      BSASK_CLUSTER, BSASK_RESFILES, BSASK_VOLSIZE, BSASK_TOTFILES,
329      1881      2      BSASK_MAXFILES, BSASK_MAXFILNUM, BSASK_FILESIZE, BSASK_REVISION,
330      1882      2      BSASK_SECTORS, BSASK_TRACKS, BSASK_CYLINDERS, BSASK_MAXBLOCK,
331      1883      2      BSASK_INDEXLBN, BSASK_BOOTVBN, BSASK_DIR_VERIFY,
332      1884      2      BSASK_VERLIMIT, BSASK_HIGHWATER]:
333      P 1885      2      FAO ('!UL',
334      1886      2      .ATT[BSASC_LENGTH,0,.ATT[BSASW_SIZE]*8,1]);
335      1887      2
336      1888      2      [BSASK_DEVTYP] :
337      1889      2      IF .ATT[BSASW_SIZE] EQL 1
338      1890      2      THEN
339      1891      2      ! Old format DEVTYP attribute. This is the DEVTYP from the UCB
340      1892      2      !
341      P 1893      2      FAO ('!UL',
342      1894      2      .ATT[BSASC_LENGTH,0,.ATT[BSASW_SIZE]*8,1])
343      1895      2
344      1896      2      ELSE

```

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

1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953

```

BEGIN
    ! New format DEVTYP. We now use a longword and store
    ! the MEDIA_ID from the UCB. We use the nondecoded
    ! form of the MEDIA_ID so we must now pull the ASCII out.
LOCAL
    NAME_LENGTH      : LONG   INITIAL (DEVTYP_BUF_LEN),
    TYPE_LENGTH      : LONG   INITIAL (DEVTYP_BUF_LEN),
    NAME_BUFFER      : VECTOR[DEVTYP_BUF_LEN, BYTE],
    TYPE_BUFFER      : VECTOR[DEVTYP_BUF_LEN, BYTE];

DECODE_DEVTYP ( .ATT[BSASC_LENGTH, 0, 32, 0],
                NAME_LENGTH, NAME_BUFFER,
                TYPE_LENGTH, TYPE_BUFFER );

FAO_ ('!XL (!AF, !AF)',
      .ATT[BSASC_LENGTH, 0, .ATT[BSASW_SIZE]*8, 0],
      .TYPE_LENGTH, TYPE_BUFFER,
      .NAME_LENGTH, NAME_BUFFER );

END ;

[BSASK_FID, BSASK_BACKLINK]:
FAO_ ('!UL, !UL, !UL',
      .ATT[BSASC_LENGTH, 0, 16, 0] + .ATT[BSASC_LENGTH+5, 0, 8, 0] ^ 16,
      .ATT[BSASC_LENGTH+2, 0, 16, 0],
      .ATT[BSASC_LENGTH+4, 0, 8, 0]);

[BSASK_PROTECT]:
LIST_PROTECTION(.ATT[BSASC_LENGTH, 0, 16, 0], 'RWCD');

[BSASK_FILEPROT, BSASK_FPRO, BSASK_DIR_FPRO]:
LIST_PROTECTION(.ATT[BSASC_LENGTH, 0, 16, 0], 'RWED');

[BSASK_RECPROT, BSASK_RPRO]:
LIST_PROTECTION(.ATT[BSASC_LENGTH, 0, 16, 0], 'RWUD');

[BSASK_BADBLOCK]:
BEGIN LOCAL P: REF VECTOR;
P = ATT[BSASC_LENGTH, 0, 0, 0];
WHILE .P LSSA ATT[BSASC_LENGTH, 0, 0, 0] + .ATT[BSASW_SIZE] DO
    BEGIN
        FAO_ ('!UL: !UL ', .P[0], .P[1]);
        P = .P + 8;
    END;
END;

[BSASK_SERIAL, BSASK_SERIALNUM]:
FAO_ ('!OL',
      .ATT[BSASC_LENGTH, 0, 32, 0]);

[BSASK_ACLSEGMENT]:
BEGIN
LOCAL
    ACE_POINTER : REF BBLOCK,          ! Address of the current ACE

```

```

: 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

```

```

1954 3 ACE_BINDESC : BBLOCK [8], ! ACE binary descriptor
1955 3 ACE_TXTDESC : BBLOCK [8], ! ACE text descriptor
1956 3 ACE_TEXT : BBLOCK [512]; ! Converted ACE storage
1957 3 ACE_POINTER = ATT[BSASC_LENGTH,0,0,0]; ! Start of ACE's
1958 3 CHSFILL (0, 8, ACE_BINDESC);
1959 3 CHSFILL (0, 8, ACE_TXTDESC);
1960 3 UNTIL .ACE_POINTER GEQA ATT[BSASC_LENGTH,0,0,0] + .ATT[BSASW_SIZE]
1961 3 DO
1962 4 BEGIN
1963 4 ACE_BINDESC[DSCSW_LENGTH] = .ACE_POINTER[ACESB_SIZE];
1964 4 ACE_BINDESC[DSCSA_POINTER] = .ACE_POINTER;
1965 4 ACE_TXTDESC[DSCSW_LENGTH] = 512;
1966 4 ACE_TXTDESC[DSCSA_POINTER] = ACE_TEXT;
1967 4 $FORMAT_ACL (ACLENT = ACE_BINDESC,
1968 4 ACLEN = ACE_TXTDESC[DSCSW_LENGTH],
1969 4 ACLSTR = ACE_TXTDESC,
1970 4 WIDTH = %REF(80),
1971 4 TRMDSC = $DESCRIPTOR (%CHAR(13), %CHAR(10)),
1972 4 INDENT = %REF(6));
1973 4
1974 4 ! Shave off the first 6 blanks if this is the first ACE being
1975 4 ! output as they have already been output.
1976 4
1977 4
1978 4 IF .ACE_POINTER EQLA ATT[BSASC_LENGTH,0,0,0]
1979 4 THEN
1980 5 BEGIN
1981 5 ACE_TXTDESC[DSCSW_LENGTH] = .ACE_TXTDESC[DSCSW_LENGTH] - 6;
1982 5 ACE_TXTDESC[DSCSA_POINTER] = .ACE_TXTDESC[DSCSA_POINTER] + 6;
1983 4 END;
1984 4 FAO_ ('!AS', ACE_TXTDESC);
1985 4 EOL_ ();
1986 4 ACE_POINTER = .ACE_POINTER + .ACE_POINTER[ACESB_SIZE];
1987 4 END;
1988 2 END;
1989 2
1990 2 [BSASK_CRYPTKEY] :
1991 2 BEGIN
1992 3 FAO_ ('Encrypted with algorithm: !XB', .ATT[BSASB_CRYPTYP]);
1993 3 EOL_ ();
1994 3 IF .ATT[BSASB_CRYPTYP] NEQU 0
1995 3 THEN
1996 4 BEGIN
1997 4 FAO_ ('Key: !XL !XL, Iv: !XL !XL',
1998 4 .ATT[$BYTEOFFSET(BSASQ_CRYPKKEY),0,32,0],
1999 4 .ATT[$BYTEOFFSET(BSASQ_CRYPKKEY)+4,0,32,0],
2000 4 .ATT[$BYTEOFFSET(BSASQ_CRYPIV),0,32,0],
2001 4 .ATT[$BYTEOFFSET(BSASQ_CRYPIV)+4,0,32,0]);
2002 4 EOL_ ();
2003 4 END;
2004 2 END;
2005 2
2006 2 [INRANGE, OTRANGE]:
2007 2 BEGIN
2008 2 DECR I FROM .ATT[BSASW_SIZE]-1 TO 0 DO
2009 2 FAO_ ('!XB', .ATT[.I+BSASC_LENGTH,0,8,0]);
2010 2 END;

```

P
P
P
P
P

P
P
P
P

:	459	2011	2	
:	460	2012	2	TES;
:	461	2013	2	EOL ();
:	462	2014	1	END;

```

.TITLE ANALYZE Analyze a save set
.IDENT \V04-000\
.PSECT COMMON,NOEXE, (OVR,2

00000 GLOBAL_BASE:
      .BLKB 0
00000 FREE_LIST:
      .BLKB 8
00008 INPUT_WAIT:
      .BLKB 8
00010 REREAD_WAIT:
      .BLKB 8
00018 OUTPUT_WAIT:
      .BLKB 8
00020 JPI_UIC: .BLKB 4
00024 JPI_USERNAME:
      .BLKB 12
00030 JPI_DATE:
      .BLKB 8
00038 JPI_NODE_DESC:
      .BLKB 8
00040 JPI_CURPRIV:
      .BLKB 8
00048 SYI_VERSION:
      .BLKB 4
0004C SYI_SID: .BLKB 4
00050 RWSV_HOLD_LIST:
      .BLKB 8
00058 RWSV_CRC16:
      .BLKB 64
00098 RWSV_AUTODIN:
      .BLKB 64
000D8 RWSV_FILESET_ID:
      .BLKB 8
000E0 RWSV_VOLUME_ID:
      .BLKB 12
000EC RWSV_VOL_NUMBER:
      .BLKB 2
000EE RWSV_SEG_NUMBER:
      .BLKB 2
000F0 RWSV_FILE_NUMBER:
      .BLKB 4
000F4 RWSV_SAVE_QUAL:
      .BLKB 4
000F8 RWSV_SAVE_FAB:
      .BLKB 4
000FC RWSV_CHAN:
      .BLKB 4
00100 RWSV_XOR_BCB:
      .BLKB 4

```

00104 RWSV_IN_SEQ:
 .BLKB 4
00108 RWSV_IN_SEQ 0:
 .BLKB 4
0010C RWSV_IN_XOR_SEG:
 .BLKB 4
00110 RWSV_IN_XOR_RFA:
 .BLKB 6
00116 RWSV_LOOKAHEAD:
 .BLKB 1
00117 RWSV_XOR_SIZE:
 .BLKB 1
00118 RWSV_IN_GROUP_SIZE:
 .BLKB 4
0011C RWSV_IN_ERRORS:
 .BLKB 2
0011E RWSV_IN_XORUSE:
 .BLKB 2
00120 RWSV_IN_ORGERR:
 .BLKB 8
00128 RWSV_IN_VBN:
 .BLKB 4
0012C RWSV_IN_VBN 0:
 .BLKB 4
00130 RWSV_ALLOC:
 .BLKB 4
00134 RWSV_EOF:
 .BLKB 4
00138 RWSV_OUT_SEQ:
 .BLKB 4
0013C RWSV_OUT_VBN:
 .BLKB 4
00140 RWSV_OUT_BLOCK_COUNT:
 .BLKB 4
00144 RWSV_OUT_ERRORS:
 .BLKB 2
00146 RWSV_SEQ_ERRORS:
 .BLKB 2
00148 RWSV_OUT_GROUP_COUNT:
 .BLKB 1
00149 RWSV_PADDING:
 .BLKB 3
0014C QUAL: .BLKB 112
001BC COM_SSNAME:
 .BLKB 8
001C4 COM_VALID_TYPES:
 .BLKB 2
001C6 COM_FLAGS:
 .BLKB 2
001C8 COM_PADDING:
 .BLKB 1
001C9 COM_BUFF_COUNT:
 .BLKB 1
001CA COM_I_SETCOUNT:
 .BLKB 1
001CB COM_O_SETCOUNT:
 .BLKB 1

001CC COM_I_STRUCNAME:
 .BLKB 12
001DB COM_O_STRUCNAME:
 .BLKB 12
001E4 COM_O_BSRDATE:
 .BLKB 8
001EC ALT_SSNAME:
 .BLKB 32
0020C INPUT_FUNC:
 .BLKB 1
0020D INPUT_RTYPE:
 .BLKB 1
0020E OUTPUT_FUNC:
 .BLKB 1
0020F FAST_STRUCLEV:
 .BLKB 1
00210 INPUT_BEG:
 .BLKB 0
00210 INPUT_CHAN:
 .BLKB 4
00214 INPUT_FLAGS:
 .BLKB 2
00216 INPUT_PADDING:
 .BLKB 2
00218 INPUT_FAB:
 .BLKB 4
0021C INPUT_NAM:
 .BLKB 4
00220 INPUT_BCB:
 .BLKB 4
00224 INPUT_QUAL:
 .BLKB 4
00228 INPUT_BAD:
 .BLKB 4
0022C INPUT_BLOCK:
 .BLKB 4
00230 INPUT_MAXBLOCK:
 .BLKB 4
00234 INPUT_MEDIA_ID:
 .BLKB 4
00238 INPUT_NAMEDESC:
 .BLKB 8
00240 INPUT_STATBLK:
 .BLKB 8
00248 INPUT_HDR_BEG:
 .BLKB 0
00248 INPUT_CREDATE:
 .BLKB 8
00250 INPUT_REVDATE:
 .BLKB 8
00258 INPUT_EXPDATE:
 .BLKB 8
00260 INPUT_BAKDATE:
 .BLKB 8
00268 INPUT_FILEOWNER:
 .BLKB 4
0026C INPUT_FILECHAR:

00270	INPUT_RECATTR:	.BLKB	4
00290	INPUT_HDR_END:	.BLKB	32
00290	INPUT_END:	.BLKB	0
00290	INPUT_PROC_LIST:	.BLKB	0
00294	INPUT_PLACEMENT:	.BLKB	4
0029C	INPUT_VBN_LIST:	.BLKB	8
002A4	INPUT_PLACE_LEN:	.BLKB	8
002A6	INPUT_PADDING_2:	.BLKB	2
002A8	OUTPUT_BEG:	.BLKB	2
002A8	OUTPUT_CHAN:	.BLKB	0
002AC	OUTPUT_FLAGS:	.BLKB	4
002AE	OUTPUT_PADDING:	.BLKB	2
002B0	OUTPUT_FAB:	.BLKB	2
002B4	OUTPUT_NAM:	.BLKB	4
002B8	OUTPUT_BCB:	.BLKB	4
002BC	OUTPUT_QUAL:	.BLKB	4
002C0	OUTPUT_BAD:	.BLKB	4
002C4	OUTPUT_BLOCK:	.BLKB	4
002C8	OUTPUT_MAXBLOCK:	.BLKB	4
002CC	OUTPUT_DEVGEO:	.BLKB	4
002D4	OUTPUT_ATTBUF:	.BLKB	8
00364	OUTPUT_END:	.BLKB	144
00364	LIST_TOTFILES:	.BLKB	0
00368	LIST_TOTSIZE:	.BLKB	4
0036C	VERIFY_FAB:	.BLKB	4
00370	VERIFY_USE_COUNT:	.BLKB	4
00374	VERIFY_QUAL:	.BLKB	4
00378	COMPARE_BCB:	.BLKB	4

0037C FAST_BUFFER:
 .BLKB 4
00380 FAST_BUFFER_SIZE:
 .BLRB 4
00384 FAST_RVN:
 .BLKB 1
00385 FAST_PADDING:
 .BLKB 1
00386 DIR_VERLIMIT:
 .BLKB 2
00388 FAST_VOL_BEG:
 .BLKB 0
00388 FAST_IMAP_SIZE:
 .BLKB 4
0038C FAST_IMAP:
 .BLKB 4
00390 FAST_HDR_OFFSET:
 .BLKB 4
00394 FAST_BOOT_LBN:
 .BLKB 4
00398 FAST_VOL_END:
 .BLKB 0
00398 JOUR_BUFFER:
 .BLKB 4
0039C JOUR_DIR:
 .BLKB 4
003A0 JOUR_HIBLK:
 .BLKB 4
003A4 JOUR_EFBLK:
 .BLKB 4
003A8 JOUR_INBLK:
 .BLKB 4
003AC JOUR_FFBYTE:
 .BLKB 2
003AE JOUR_INBYTE:
 .BLKB 2
003B0 JOUR_STRUCT_LEV:
 .BLRB 2
003B2 JOUR_COUNT:
 .BLKB 1
003B3 JOUR_REVERSE:
 .BLKB 1
003B4 JOUR_EXSZ:
 .BLKB 2
003B6 JOUR_PADDING:
 .BLKB 2
003B8 CHKPT_HIGH_SP:
 .B[KB 4
003BC CHKPT_LOW_SP:
 .BLKB 4
003C0 CHKPT_STACK:
 .BLKB 4
003C4 CHKPT_VARS:
 .BLKB 4
003C8 CHKPT_STATUS:
 .BLKB 4
003CC DIR_BEG:.BLKB 0

003CC DIR_CHAN:
 .BLKB 4
003D0 DIR_NAM:.BLKB 4
003D4 DIR_DEV_DESC:
 .BLKB 4
003D8 DIR_SEL_DIR:
 .BLKB 8
003E0 DIR_SEL_NTV:
 .BLKB 8
003E8 DIR_STRUCLEV:
 .BLKB 1
003E9 DIR_LEVELS:
 .BLKB 1
003EA DIR_FLAGS:
 .BLKB 1
003EB DIR_STATUS:
 .BLKB 1
003EC DIR_STRING:
 .BLKB 320
0052C DIR_STACK:
 .BLKB 612
00790 DIR_SP:.BLKB 4
00794 DIR_SEL_LATEST:
 .BLKB 4
00798 DIR_END:.BLKB 0
00798 DIR_SCANLIMIT:
 .BLKB 36
007BC INPUT_MTL:
 .BLKB 4
007C0 OUTPUT_MTL:
 .BLKB 4
007C4 CURRENT_MTL:
 .BLKB 4
007C8 CURRENT_VCB:
 .BLKB 4
007CC CURRENT_WCB:
 .BLKB 4
007D0 ACL_FIB_DESCR:
 .BLKB 8
007D8 ACL_FIB:.BLKB 64
00818 ACL_LENGTH:
 .BLKB 4
0081C ACL_BUFFER:
 .BLKB 4
00820 CRYP_IN_CONTEXT:
 .BLKB 4
00824 CRYP_OU_CONTEXT:
 .BLKB 4
00928 CRYP_DA_CONTEXT:
 .BLKB 4
0082C CRYP_DATA_ENCIV:
 .BLKB 8
00834 CRYP_DATA_CODE:
 .BLKB 4
00838 CRYP_DATA_KEY:
 .BLKB 8
00840 CRYP_DATA_IV:

00848 CRYP_DATA CKSM:
.BLKB 8
.BLKB 4

.PSECT CODE,NOWRT,2

			65	73	6C	61	46	05	00000	P.AAB:	.ASCII	<5>\False\			
			65	75	72	54	04	04	00006	P.AAC:	.ASCII	<4>\True\			
									00008		.BLKB	1			
									0000C	P.AAA:	.ADDRESS	P.AAB, P.AAC			
		45	4D	41	4E	53	53	06	00014	P.AAE:	.ASCII	<6>\SSNAME\			
	44	4E	41	4D	4D	4F	43	07	0001B	P.AAF:	.ASCII	<7>\COMMAND\			
	54	4E	45	4D	4D	4F	43	07	00023	P.AAG:	.ASCII	<7>\COMMENT\			
45	4D	41	4E	52	45	53	55	08	0002B	P.AAH:	.ASCII	<8>\USERNAME\			
	43	49	55	52	45	53	55	07	00034	P.AAI:	.ASCII	<7>\USERUIC\			
				45	54	41	44	04	0003C	P.AAJ:	.ASCII	<4>\DATE\			
				53	59	53	50	4F	00041	P.AAK:	.ASCII	<5>\OPSYS\			
				52	45	56	53	59	00047	P.AAL:	.ASCII	<6>\SYSVER\			
45	4D	41	4E	45	44	4F	4E	08	0004E	P.AAM:	.ASCII	<8>\NODENAME\			
					52	49	53	03	00057	P.AAN:	.ASCII	<3>\SIR\			
		44	49	45	56	49	52	44	07	0005B	P.AAO:	.ASCII	<7>\DRIVEID\		
		52	45	56	4B	43	41	42	07	00063	P.AAP:	.ASCII	<7>\BACKVER\		
45	5A	49	53	4B	43	4F	4C	42	09	0006B	P.AAQ:	.ASCII	<9>\BLOCKSIZE\		
		45	5A	49	53	52	4F	58	07	00075	P.AAR:	.ASCII	<7>\XORSIZE\		
		53	52	45	46	46	55	42	07	0007D	P.AAS:	.ASCII	<7>\BUFFERS\		
4D	41	4E	54	45	53	4C	4F	56	09	00085	P.AAT:	.ASCII	<9>\VOLSETNAM\		
				53	4C	4F	56	4E	05	0008F	P.AAU:	.ASCII	<5>\NVOLS\		
		45	5A	49	53	4B	43	41	42	08	00095	P.AAV:	.ASCII	<8>\BACKSIZE\	
53	45	4C	49	46	4B	43	41	42	09	0009E	P.AAW:	.ASCII	<9>\BACKFILES\		
54	43	55	52	54	53	4C	4F	56	09	000A8	P.AAX:	.ASCII	<9>\VOLSTRUCT\		
		45	4D	41	4E	4C	4F	56	07	000B2	P.AAY:	.ASCII	<7>\VOLNAME\		
45	4D	41	4E	52	45	4E	57	4F	09	000BA	P.AAZ:	.ASCII	<9>\OWNERNAME\		
			54	41	4D	52	4F	46	06	000C4	P.ABA:	.ASCII	<6>\FORMAT\		
						4E	56	52	03	000CB	P.ABB:	.ASCII	<3>\RVN\		
		52	45	4E	57	4F	4C	4F	56	08	000CF	P.ABC:	.ASCII	<8>\VOLOWNER\	
		54	4F	52	50	45	4C	49	46	08	000D8	P.ABD:	.ASCII	<7>\PROTECT\	
			54	4F	52	50	43	45	52	07	000E0	P.ABE:	.ASCII	<8>\FILEPROT\	
				52	41	48	43	4C	4F	56	07	000E9	P.ABF:	.ASCII	<7>\RECPROT\
			45	54	41	44	4C	4F	56	07	000F1	P.ABG:	.ASCII	<7>\VOLCHAR\	
				57	4F	44	4E	49	57	06	000F9	P.ABH:	.ASCII	<7>\VOLDATE\	
			4D	49	4C	5F	55	52	4C	07	00101	P.ABI:	.ASCII	<6>\WINDOW\	
				44	4E	45	54	58	45	06	00108	P.ABJ:	.ASCII	<7>\LRU LIM\	
				52	45	54	53	55	4C	43	07	00110	P.ABK:	.ASCII	<6>\EXTEND\
		53	45	4C	49	46	53	45	52	08	00117	P.ABL:	.ASCII	<7>\CLUSTER\	
			45	5A	49	53	4C	4F	56	07	0011F	P.ABM:	.ASCII	<8>\RESFILES\	
			45	5A	49	53	54	4F	54	07	00128	P.ABN:	.ASCII	<7>\VOLSIZE\	
		53	45	4C	49	46	54	4F	54	08	00130	P.ABO:	.ASCII	<7>\TOTSIZE\	
		53	45	4C	49	46	58	41	4D	08	00138	P.ABP:	.ASCII	<8>\TOTFILES\	
4D	55	4E	4C	49	46	58	41	4D	09	00141	P.ABQ:	.ASCII	<8>\MAXFILES\		
	55	4E	4C	49	46	58	41	4D	09	0014A	P.ABR:	.ASCII	<9>\MAXFILNUM\		
	45	4D	41	4E	45	4C	49	46	08	00154	P.ABS:	.ASCII	<9>\SERIALNUM\		
	56	45	4C	43	55	52	54	53	08	0015E	P.ABT:	.ASCII	<8>\FILENAME\		
						44	49	46	03	00167	P.ABU:	.ASCII	<8>\STRUCLEV\		
						44	49	46	03	00170	P.ABV:	.ASCII	<3>\FID\		
		4B	4E	49	4C	4B	43	41	42	08	00174	P.ABW:	.ASCII	<8>\BACKLINK\	
		45	5A	49	53	45	4C	49	46	08	0017D	P.ABX:	.ASCII	<8>\FILESIZE\	
						43	49	55	03	00186	P.ABY:	.ASCII	<3>\UIC\		

.....

					4F	52	50	46	04	0018A	P.ABZ:	.ASCII	<4>\FPRO\					
					4F	52	50	52	04	0018F	P.ACA:	.ASCII	<4>\RPRO\					
		4C	45	56	45	4C	43	41	07	00194	P.ACB:	.ASCII	<7>\ACLEVEL\					
				52	41	48	43	55	05	0019C	P.ACC:	.ASCII	<5>\UCHAR\					
		52	54	54	41	43	45	52	07	001A2	P.ACD:	.ASCII	<7>\RECATTR\					
	4E	4F	49	53	49	56	45	52	08	001AA	P.ACE:	.ASCII	<8>\REVISION\					
		45	54	41	44	45	52	43	07	001B3	P.ACF:	.ASCII	<7>\CREDATE\					
		45	54	41	44	56	45	52	07	001BB	P.ACG:	.ASCII	<7>\REVDATE\					
		45	54	41	44	50	58	45	07	001C3	P.ACH:	.ASCII	<7>\EXPDATE\					
		45	54	41	44	4B	41	42	07	001CB	P.ACI:	.ASCII	<7>\BAKDATE\					
		53	52	4F	54	43	45	53	07	001D3	P.ACJ:	.ASCII	<7>\SECTORS\					
			53	4B	43	41	52	54	06	001DB	P.ACK:	.ASCII	<6>\TRACKS\					
	53	52	45	44	4E	49	4C	59	43	09	001E2	P.ACL:	.ASCII	<9>\CYLINDERS\				
		4B	43	4F	4C	42	58	41	4D	08	001EC	P.ACM:	.ASCII	<8>\MAXBLOCK\				
				50	59	54	56	45	44	06	001F5	P.ACN:	.ASCII	<6>\DEVTYPE\				
				4C	41	49	52	45	53	06	001FC	P.ACO:	.ASCII	<6>\SERIAL\				
				4D	41	4E	56	45	44	06	00203	P.ACP:	.ASCII	<6>\DEVNAM\				
		4B	43	4F	4C	42	44	41	4C	05	0020A	P.ACQ:	.ASCII	<5>\LABEL\				
		4E	42	4C	58	45	44	4E	49	08	00210	P.ACR:	.ASCII	<8>\BADBLOCK\				
	4B	43	4F	4C	42	54	4F	4F	42	09	00219	P.ACS:	.ASCII	<8>\INDEXLBN\				
				4E	42	56	54	4F	42	07	00222	P.ACT:	.ASCII	<9>\BOOTBLOCK\				
	54	4E	45	4D	45	43	41	4C	50	09	0022C	P.ACU:	.ASCII	<7>\BOOTVBN\				
			43	49	55	5F	52	49	44	07	00234	P.ACV:	.ASCII	<9>\PLACEMENT\				
		4F	52	50	46	5F	52	49	44	08	0023E	P.ACW:	.ASCII	<7>\DIR_UIC\				
	53	55	54	41	54	53	5F	52	49	44	0A	00246	P.ACX:	.ASCII	<8>\DIR_FPRO\			
	4D	49	4C	52	45	56	5F	52	49	44	0A	0024F	P.ACY:	.ASCII	<10>\DIR_STATUS\			
			54	49	4D	49	4C	52	45	56	08	0025A	P.ACZ:	.ASCII	<10>\DIR_VERLIM\			
		4E	49	4D	4E	49	41	54	45	52	09	00265	P.ADA:	.ASCII	<8>\VERLIMIT\			
		58	41	4D	4E	49	41	54	45	52	09	0026E	P.ADB:	.ASCII	<9>\RETAINMIN\			
	54	4E	45	4D	47	45	53	4C	43	41	0A	00278	P.ADC:	.ASCII	<9>\RETAINMAX\			
		52	45	54	41	57	48	47	49	48	09	00282	P.ADD:	.ASCII	<10>\ACLSEGMENT\			
		53	47	41	4C	46	5F	4C	4E	4A	09	0028D	P.ADE:	.ASCII	<9>\HIGHWATER\			
		59	45	4B	54	41	44	50	59	52	43	0A	00288	P.ADF:	.ASCII	<9>\JNL_FLAGS\		
													002A1	P.ADG:	.ASCII	<10>\CRYPDATKEY\		
00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	002AC	P.AAD:	.ADDRESS	P.AAE, P.AAF, P.AAG, P.AAH, P.AAI, -					
00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	002C4			P.AAJ, P.AAK, P.AAL, P.AAM, P.AAN, P.AAO, -					
00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	002DC			P.AAP, P.AAQ, P.AAR, P.AAS, P.AAT, P.AAU, -					
00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	002F4			P.AAV, P.AAW, P.AAX, P.AAY, P.AAZ, P.ABA, -					
00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	0030C			P.ABB, P.ABC, P.ABD, P.ABE, P.ABF, P.ABG, -					
00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00324			P.ABH, P.ABI, P.ABJ, P.ABK, P.ABL, P.ABM, -					
00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	0033C			P.ABN, P.ABO, P.ABP, P.ABQ, P.ABR, P.ABS, -					
00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00354			P.ABT, P.ABU, P.ABV, P.ABW, P.ABX, P.ABY, -					
00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	0036C			P.ABZ, P.ACA, P.ACB, P.ACC, P.ACD, P.ACE, -					
00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00384			P.ACF, P.ACG, P.ACH, P.ACI, P.ACJ, P.ACK, -					
00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	0039C			P.ACL, P.ACM, P.ACN, P.ACO, P.ACP, P.ACQ, -					
00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	003B4			P.ACR, P.ACS, P.ACT, P.ACU, P.ACV, P.ACW, -					
00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	003CC			P.ACX, P.ACY, P.ACZ, P.ADA, P.ADB, P.ADC, -					
00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	00000000'	003E4			P.ADD, P.ADE, P.ADF, P.ADG					
53	33	21	20	3D	20	45	5A	49	53	20	20	20	1B	003F0	P.ADH:	.ASCII	<27>\ SIZE = !SSL, TYPE = !AC\	
		43	41	21	20	3D	20	45	50	59	54	20	2C	4C	003FF			
								20	20	20	20	20	06	0040C	P.ADI:	.ASCII	<6>\	
								20	20	20	22	05	00413	P.ADJ:	.ASCII	<5>\!AF\		
								46	41	21	22	03	00419	P.ADK:	.ASCII	<3>\!ZU\		
									55	25	21	03	0041D	P.ADL:	.ASCII	<3>\!XD\		
									44	25	21	03	00421	P.ADM:	.ASCII	<3>\!UL\		
									4C	55	21	03	00425	P.ADN:	.ASCII	<3>\!UL\		
									4C	55	21	03	00429	P.ADO:	.ASCII	<13>\!XL (!AF,!AF)\		
29	46	41	21	2C	46	41	21	28	20	4C	58	21	0D					

ANALYZE
V04-000

Analyze a save set
ANALYZE_ONE_ATTRIBUTE - analyze contents of att

K 3
15-Sep-1984 23:40:04
14-Sep-1984 11:53:45

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]ANALYZE.B32;1

00BA	019D	00EA	00DA	000B4	5\$-1\$,-
00A4	00DA	0180	00BA	000BC	9\$-1\$,-
0165	00C8	00A4	00DA	000C4	2\$-1\$,-
00D1	00DA	00DA	00A4	000CC	9\$-1\$,-
00A4	00DA	01AA	00D1	000D4	2\$-1\$,-
		00A4	0227	000DC	5\$-1\$,-
					5\$-1\$,-
					5\$-1\$,-
					9\$-1\$,-
					7\$-1\$,-
					15\$-1\$,-
					16\$-1\$,-
					17\$-1\$,-
					2\$-1\$,-
					8\$-1\$,-
					9\$-1\$,-
					9\$-1\$,-
					9\$-1\$,-
					9\$-1\$,-
					9\$-1\$,-
					2\$-1\$,-
					9\$-1\$,-
					9\$-1\$,-
					9\$-1\$,-
					21\$-1\$,-
					5\$-1\$,-
					2\$-1\$,-
					14\$-1\$,-
					14\$-1\$,-
					9\$-1\$,-
					7\$-1\$,-
					16\$-1\$,-
					17\$-1\$,-
					2\$-1\$,-
					2\$-1\$,-
					2\$-1\$,-
					9\$-1\$,-
					8\$-1\$,-
					8\$-1\$,-
					8\$-1\$,-
					8\$-1\$,-
					9\$-1\$,-
					9\$-1\$,-
					9\$-1\$,-
					9\$-1\$,-
					10\$-1\$,-
					21\$-1\$,-
					5\$-1\$,-
					5\$-1\$,-
					19\$-1\$,-
					9\$-1\$,-
					2\$-1\$,-
					9\$-1\$,-
					2\$-1\$,-
					7\$-1\$,-
					16\$-1\$,-

.....

	50	04	A6	3C	00181	MOVZWL	4(R6), R0		
	52	09	A6	9A	00185	MOVZBL	9(R6), R2		
52	52		10	78	00189	ASHL	#16, R2, R2		
			6240	9F	0018D	PUSHAB	(R2)[R0]		
		FE09	CF	9F	00190	PUSHAB	P.ADP		
	6A		04	FB	00194	CALLS	#4, LIST_FAO		
			4A	11	00197	BRB	23\$		
	44435752		8F	DD	00199	15\$: PUSHL	#1145263954	1928	
			0E	11	0019F	BRB	18\$		
	44455752		8F	DD	001A1	16\$: PUSHL	#1145395026	1931	
			06	11	001A7	BRB	18\$		
	44555752		8F	DD	001A9	17\$: PUSHL	#1146443602	1934	
	7E	04	A6	3C	001AF	18\$: MOVZWL	4(R6), -(SP)		
00000000G	00		02	FB	001B3	CALLS	#2, LIST_PROTECTION		
			27	11	001BA	BRB	23\$		
	52	04	A6	9E	001BC	19\$: MOVAB	4(R6), P	1938	
	50	04	A746	9E	001C0	20\$: MOVAB	4(R7)[R6], R0	1939	
	50		52	D1	001C5	CMPL	P, R0		
			39	1E	001C8	BGEQU	26\$		
	7E		62	7D	001CA	MOVQ	(P), -(SP)	1941	
		FDD8	CF	9F	001CD	PUSHAB	P.ADQ		
	6A		03	FB	001D1	CALLS	#3, LIST_FAO		
	52		08	CO	001D4	ADDL2	#8, P	1942	
			E7	11	001D7	BRB	20\$	1939	
		04	A6	DD	001D9	21\$: PUSHL	4(R6)	1948	
		FDD2	CF	9F	001DC	PUSHAB	P.ADR		
	6A		02	FB	001E0	22\$: CALLS	#2, LIST_FAO		
			00A8	31	001E3	23\$: BRW	29\$		
	59	04	A6	9E	001E6	24\$: MOVAB	4(R6), R9	1957	
	58		59	D0	001EA	MOVL	R9, ACE_POINTER		
08	00		00	2C	001ED	MOVCS	#0, (SP), #0, #8, ACE_BINDESC	1958	
			F8	AD	001F2				
08	00		00	2C	001F4	MOVCS	#0, (SP), #0, #8, ACE_TXTDESC	1959	
			F0	AD	001F9				
	50	04	A746	9E	001FB	25\$: MOVAB	4(R7)[R6], R0	1960	
	50		58	D1	00200	CMPL	ACE_POINTER, R0		
			DE	1E	00203	26\$: BGEQU	23\$		
	F8	AD	68	9B	00205	MOVZBW	(ACE_POINTER), ACE_BINDESC	1963	
	FC	AD	58	D0	00209	MOVL	ACE_POINTER, ACE_BINDESC+4	1964	
	F0	AD	0200	8F	B0	0020D	MOVW	#512, ACE_TXTDESC	1965
	F4	AD	10	AE	9E	00213	MOVAB	ACE_TEXT, ACE_TXTDESC+4	1966
			7E	D4	00218	CLRL	-(SP)	1972	
	08	AE	06	D0	0021A	MOVL	#6, 8(SP)		
		08	AE	9F	0021E	PUSHAB	8(SP)		
		FD95	CF	9F	00221	PUSHAB	P.ADS		
	0C	AE	50	8F	9A	00225	MOVZBL	#80, 12(SP)	
			0C	AE	9F	0022A	PUSHAB	12(SP)	
			F0	AD	9F	0022D	PUSHAB	ACE_TXTDESC	
			F0	AD	9F	00230	PUSHAB	ACE_TXTDESC	
			F8	AD	9F	00233	PUSHAB	ACE_BINDESC	
00000000G	00		07	FB	00236	CALLS	#7, SYSSFORMAT_ACL		
	59		58	D1	0023D	CMPL	ACE_POINTER, R9	1978	
			0A	12	00240	BNEQ	27\$		
	F0	AD	06	A2	00242	SUBW2	#6, ACE_TXTDESC	1981	
	F4	AD	06	CO	00246	ADDL2	#6, ACE_TXTDESC+4	1982	
			F0	AD	9F	0024A	27\$: PUSHAB	ACE_TXTDESC	1984
		FD71	CF	9F	0024D	PUSHAB	P.ADU		

ANALYZE
V04-000

Analyze a save set
ANALYZE_ONE_ATTRIBUTE - analyze contents of att

N 3
15-Sep-1984 23:40:04
14-Sep-1984 11:53:45

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]ANALYZE.B32;1

Page 22
(3)

00000000G	6A		02	FB	00251	CALLS	#2, LIST_FAO		
	00		00	FB	00254	CALLS	#0, LIST_EOL		1985
	50		68	9A	0025B	MOVZBL	(ACE_POINTER), R0		1986
	58		50	C0	0025E	ADDL2	R0, ACE_POINTER		
			98	11	00261	BRB	25\$		1960
	7E		66	9A	00263	MOVZBL	(R6), -(SP)		1992
		FD5C	CF	9F	00266	PUSHAB	P.ADV		
00000000G	6A		02	FB	0026A	CALLS	#2, LIST_FAO		
	00		00	FB	0026D	CALLS	#0, LIST_EOL		1993
			66	95	00274	TSTB	(R6)		1994
			16	13	00276	BEQL	29\$		
	7E		A6	7D	00278	MOVQ	12(R6), -(SP)		2001
	7E		A6	7D	0027C	MOVQ	4(R6), -(SP)		
		FD60	CF	9F	00280	PUSHAB	P.ADW		
00000000G	6A		05	FB	00284	CALLS	#5, LIST_FAO		
	00		00	FB	00287	CALLS	#0, LIST_EOL		2002
00000000G	00		00	FB	0028E	CALLS	#0, LIST_EOL		2013
			04	00295	RET				2014

; Routine Size: 662 bytes, Routine Base: CODE + 049A


```

: 464 2015 1 %SBTTL 'ANALYZE_ONE_RECORD - analyze save set record'
: 465 2016 1 ROUTINE ANALYZE_ONE_RECORD(REC): L_PS NOVALUE=
: 466 2017 1
: 467 2018 1 !++
: 468 2019 1
: 469 2020 1 FUNCTIONAL DESCRIPTION:
: 470 2021 1 This routine analyzes the contents of one save set record.
: 471 2022 1
: 472 2023 1 INPUT PARAMETERS:
: 473 2024 1 REC - Pointer to record.
: 474 2025 1
: 475 2026 1 IMPLICIT INPUTS:
: 476 2027 1 NONE
: 477 2028 1
: 478 2029 1 OUTPUT PARAMETERS:
: 479 2030 1 NONE
: 480 2031 1
: 481 2032 1 IMPLICIT OUTPUTS:
: 482 2033 1 NONE
: 483 2034 1
: 484 2035 1 ROUTINE VALUE:
: 485 2036 1 NONE
: 486 2037 1
: 487 2038 1 SIDE EFFECTS:
: 488 2039 1 The listing is produced.
: 489 2040 1
: 490 2041 1 --
: 491 2042 1
: 492 2043 2 BEGIN
: 493 2044 2 MAP
: 494 2045 2 REC: REF BBLOCK; ! Pointer to record
: 495 2046 2 BIND
: 496 2047 2 RTYPES = UPLIT (
: 497 2048 2 UPLIT BYTE (%ASCIC 'NULL'),
: 498 2049 2 UPLIT BYTE (%ASCIC 'SUMMARY'),
: 499 2050 2 UPLIT BYTE (%ASCIC 'VOLUME'),
: 500 2051 2 UPLIT BYTE (%ASCIC 'FILE'),
: 501 2052 2 UPLIT BYTE (%ASCIC 'VBN'),
: 502 2053 2 UPLIT BYTE (%ASCIC 'PHYSVOL'),
: 503 2054 2 UPLIT BYTE (%ASCIC 'LBN'),
: 504 2055 2 UPLIT BYTE (%ASCIC 'FID'),
: 505 2056 2 UPLIT BYTE (%ASCIC 'FILE_EXT'))
: 506 2057 2 : VECTOR;
: 507 2058 2 L_DECL:
: 508 2059 2
: 509 2060 2
: 510 2061 2 ! Format the record header.
: 511 2062 2
: 512 2063 2 FAO_('Record header');
: 513 2064 2 EOL_();
: 514 2065 2 FAO_(' RSIZE = !UL!- = %X''!XW''', .REC[BRH$W_RSIZE]);
: 515 2066 2 EOL_();
: 516 2067 2 FAO_(' RTYPE = !AC', .RTYPES[.REC[BRH$W_RTYPE]]);
: 517 2068 2 EOL_();
: 518 2069 2 FAO_(' BADATA = !AC', .FALSETRUE[.REC[BRH$V_BADATA]]);
: 519 2070 2 EOL_();
: 520 2071 2 FAO_(' DIRECTORY = !AC', .FALSETRUE[.REC[BRH$V_DIRECTORY]]);

```

```

2072 2 EOL_():
2073 3 FAO_(' ADDRESS = !UL', .REC[BRH$L_ADDRESS]);
2074 2 EOL_():
2075 2 EOL_():
2076 2
2077 2
2078 2 ! Format the record contents.
2079 2
2080 2 ! CASE .REC[BRH$W_RTYPE] FROM BRH$K_NULL TO BRH$K_FILE_EXT OF
2081 2 SET
2082 2
2083 2 [BRH$K_NULL, BRH$K_VBN, BRH$K_LBN, OUTRANGE]:
2084 2 0;
2085 2
2086 2 [BRH$K_SUMMARY, BRH$K_VOLUME, BRH$K_FILE, BRH$K_FILE_EXT, BRH$K_PHYSVOL]:
2087 2 BEGIN
2088 P FAO_(' STRUCLEV = !XW',
2089 2 .BBLOCK[REC[BRH$C_LENGTH,0,0,0], BSASW_STRUCLEV]);
2090 2 EOL_();
2091 2 DEBLOCK_ATTR(.REC, 0, ANALYZE_ONE_ATTRIBUTE);
2092 2 EOL_();
2093 2 END;
2094 2
2095 2 [BRH$K_FID]:
2096 2 BEGIN
2097 P FAO_(' STRUCLEV = !XW',
2098 2 .BBLOCK[REC[BRH$C_LENGTH,0,0,0], BSASW_STRUCLEV]);
2099 2 EOL_();
2100 P FAO_(' FID COUNT = !UL',
2101 2 .BBLOCK[REC[BRH$C_LENGTH,0,0,0], BSASW_FID_COUNT]);
2102 2 EOL_();
2103 2 INCR I FROM 0 TO .BBLOCK[REC[BRH$C_LENGTH,0,0,0], BSASW_FID_COUNT]-1 DO
2104 2 BEGIN
2105 P FAO_(' FID = (!UL,!UL,!UL)',
2106 P .BBLOCK[REC[BRH$C_LENGTH,0,0,0], BSASW_FID_NUM] +
2107 P .BBLOCK[REC[BRH$C_LENGTH,0,0,0], BSASW_FID_NUM] ^ 16 + .I,
2108 P .BBLOCK[REC[BRH$C_LENGTH+.1*2,0,0,0], BSASW_FID_SEQ],
2109 P .BBLOCK[REC[BRH$C_LENGTH,0,0,0], BSASW_FID_RVN]);
2110 2 EOL_();
2111 2 END;
2112 2 EOL_();
2113 2 END;
2114 2
2115 2 TES;
2116 1 END;

```

59	52	4:	4C	4C	55	4E	04	00730	P.ADZ:	.ASCII	<4>\NULL\
	45	4D	55	4C	4F	56	06	00735	P.AEA:	.ASCII	<7>\SUMMARY\
		45	4C	49	46	04	0073D	P.AEB:	.ASCII	<6>\VOLUME\	
			4E	42	56	03	00744	P.AEC:	.ASCII	<4>\FILE\	
	4C	4F	56	53	59	48	07	00749	P.AED:	.ASCII	<3>\VBN\
			4E	42	4C	03	0074D	P.AEE:	.ASCII	<7>\PHYSVOL\	
			44	49	46	03	00755	P.AEF:	.ASCII	<3>\LBN\	
							00759	P.AEG:	.ASCII	<3>\FID\	
54	58	45	5F	45	4C	49	46	08	0075D	P.AEH:	<8>\FILE_EXT\

```

00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00766
00000000' 00000000' 00000000' 00768 P.ADY: .BLKB 2
00000000' 00000000' 00000000' 00780 P.AEJ: .ADDRESS P.ADZ, P.AEA, P.AEB, P.AEC, P.AED, -
0078C P.AEI: .ASCII <13>\Record header\
0079A P.AEJ: .ASCII <29>\ RSIZE = !UL!- = %X%!XW%\
007A9 P.AEK: .ASCII <17>\ RTYPE = !AC\
007B8 P.AEL: .ASCII <17>\ BADATA = !AC\
007C7 P.AEM: .ASCII <17>\ DIRECTORY = !AC\
007CA P.AEN: .ASCII <17>\ ADDRESS = !UL\
007D9 P.AEO: .ASCII <18>\ STRUCLEV = !XW\
007DC P.AEP: .ASCII <18>\ STRUCLEV = !XW\
007EB P.AEQ: .ASCII <19>\ FID_COUNT = !UL\
007EE P.AER: .ASCII <23>\ FID = (!UL,!UL,!UL)\
007FD
00800
0080F
00813
00822
00826
00835
0083A
00849

```

RTYPES= P.ADY

03FC 0000 ANALYZE_ONE_RECORD:

59	F7B4	CF	9E	00002	.WORD	Save R2,R3,R4,R5,R6,R7,R8,R9	2016
58	00000000G	00	9E	00007	MOVAB	FALSETRUE, R9	
57	00000000G	00	9E	0000E	MOVAB	LIST_FAO, R8	
	0780	C9	9F	00015	MOVAB	LIST_EOL, R7	
68		01	FB	00019	PUSHAB	P.AEI	2063
67		00	FB	0001C	CALLS	#1, LIST_FAO	
54	04	AC	D0	0001F	CALLS	#0, LIST_EOL	2064
7E		64	3C	00023	MOVL	REC, R4	2065
	078E	C9	9F	00026	MOVZWL	(R4), -(SP)	
68		02	FB	0002A	PUSHAB	P.AEJ	
67		00	FB	0002D	CALLS	#2, LIST_FAO	
52	02	A4	3C	00030	CALLS	#0, LIST_EOL	2066
	075C	C942	DD	00034	MOVZWL	2(R4), R2	2067
	07AC	C9	9F	00039	PUSHL	RTYPES[R2]	
68		02	FB	0003D	PUSHAB	P.AEK	
67		00	FB	00040	CALLS	#2, LIST_FAO	
50	04	A4	00	EF	CALLS	#0, LIST_EOL	2068
01		00	EF	00043	EXTZV	#0, #1, 4(R4), R0	2069
	07BE	C9	9F	0004C	PUSHL	FALSETRUE[R0]	
68		02	FB	00050	PUSHAB	P.AEL	
67		00	FB	00053	CALLS	#2, LIST_FAO	
50	04	A4	01	EF	CALLS	#0, LIST_EOL	2070
01		01	EF	00056	EXTZV	#1, #1, 4(R4), R0	2071
	07D0	C9	9F	0005F	PUSHL	FALSETRUE[R0]	
68		02	FB	00063	PUSHAB	P.AEM	
67		00	FB	00066	CALLS	#2, LIST_FAO	
	07E2	C9	9F	0006C	CALLS	#0, LIST_EOL	2072
68		08	DD	00069	PUSHL	8(R4)	2073
		C9	9F	0006C	PUSHAB	P.AEN	
		02	FB	00070	CALLS	#2, LIST_FAO	

0013	0013	0013	0089	00	FB	00073	1\$:	CALLS	#0, LIST_EOL	2074
0032	0089	0013	0089	00	FB	00076		CALLS	#0, LIST_EOL	2075
			0013	52	AF	00079		CASEW	R2, #0, #8	2080
				0089		0007D		.WORD	7\$-1\$,-	
				0013		00085			2\$-1\$,-	
						0008D			2\$-1\$,-	
									2\$-1\$,-	
									7\$-1\$,-	
									2\$-1\$,-	
									7\$-1\$,-	
									3\$-1\$,-	
									2\$-1\$	
		7E	10	A4	04	0008F	2\$:	RET		
			07F4	C9	3C	00090		MOVZWL	16(R4), -(SP)	2089
		68		9F	9F	00094		PUSHAB	P.AEO	
		67		02	FB	00098		CALLS	#2, LIST_FAO	
			FBA6	00	FB	0009B		CALLS	#0, LIST_EOL	2090
				CF	9F	0009E		PUSHAB	ANALYZE_ONE_ATTRIBUTE	2091
				7E	D4	000A2		CLRL	-(SP)	
				54	DD	000A4		PUSHL	R4	
	00000000G	00		03	FB	000A6		CALLS	#3, DEBLOCK_ATTR	
				54	11	000AD		BRB	6\$	2092
		52	10	A4	9E	000AF	3\$:	MOVAB	16(R4), R2	2098
		7E		62	3C	000B3		MOVZWL	(R2), -(SP)	
			0807	C9	9F	000B6		PUSHAB	P.AEP	
		68		02	FB	000BA		CALLS	#2, LIST_FAO	
		67		00	FB	000BD		CALLS	#0, LIST_EOL	2099
		7E		A2	3C	000C0		MOVZWL	6(R2), -(SP)	2101
			081A	C9	9F	000C4		PUSHAB	P.AEQ	
		68		02	FB	000C8		CALLS	#2, LIST_FAO	
		67		00	FB	000CB		CALLS	#0, LIST_EOL	2102
		56	06	A2	3C	000CE		MOVZWL	6(R2), R6	2103
		55	02	A2	9E	000D2		MOVAB	2(R2), R5	2109
		53		01	CE	000D6		MNEGL	#1, 1	
				24	11	000D9		BRB	5\$	
		7E	04	A2	9A	000DB	4\$:	MOVZBL	4(R2), -(SP)	
		7E	18	A443	3C	000DF		MOVZWL	24(R4)[I], -(SP)	
		50		65	3C	000E4		MOVZWL	(R5), R0	
		51	05	A2	9A	000E7		MOVZBL	5(R2), R1	
	51	51		10	78	000EB		ASHL	#16, R1, R1	
		50		51	CO	000EF		ADDL2	R1, R0	
				6340	9F	000F2		PUSHAB	(I)[R0]	
			082E	C9	9F	000F5		PUSHAB	P.AER	
		68		04	FB	000F9		CALLS	#4, LIST_FAO	
		67		00	FB	000FC		CALLS	#0, LIST_EOL	2110
	D8	53		56	F2	000FF	5\$:	AOBLSS	R6, 1, 4\$	2103
		67		00	FB	00103	6\$:	CALLS	#0, LIST_EOL	2112
				04	04	00106	7\$:	RET		2116

```
567 2117 1 %SBTTL 'ANALYZE_ONE_BUFFER - analyze save set buffer'
568 2118 1 ROUTINE ANALYZE_ONE_BUFFER(BCB): [_PS NOVALUE=
569 2119 1
570 2120 1 !++
571 2121 1
572 2122 1 FUNCTIONAL DESCRIPTION:
573 2123 1 This routine analyzes the contents of one save set buffer.
574 2124 1
575 2125 1 INPUT PARAMETERS:
576 2126 1 BCB - Pointer to buffer control block.
577 2127 1
578 2128 1 IMPLICIT INPUTS:
579 2129 1 NONE
580 2130 1
581 2131 1 OUTPUT PARAMETERS:
582 2132 1 NONE
583 2133 1
584 2134 1 IMPLICIT OUTPUTS:
585 2135 1 NONE
586 2136 1
587 2137 1 ROUTINE VALUE:
588 2138 1 NONE
589 2139 1
590 2140 1 SIDE EFFECTS:
591 2141 1 The listing is produced.
592 2142 1 The buffer is released.
593 2143 1
594 2144 1 --
595 2145 1
596 2146 2 BEGIN
597 2147 2 MAP
598 2148 2 BCB: REF BBLOCK; ! Pointer to buffer control block
599 2149 2 LOCAL
600 2150 2 BUF: REF BBLOCK; ! Pointer to buffer
601 2151 2 L_DECL;
602 2152 2
603 2153 2
604 2154 2 ! Point to buffer.
605 2155 2
606 2156 2 BUF = .BCB[BCB_BUFFER];
607 2157 2
608 2158 2
609 2159 2 ! Format the block header.
610 2160 2
611 2161 2 FAO_('Block header');
612 2162 2 EOL_();
613 2163 2 FAO_(' SIZE = !UL', .BUF[BBH$W_SIZE]);
614 2164 2 EOL_();
615 2165 2 FAO_(' OPSYS = !UL', .BUF[BBH$W_OPYSYS]);
616 2166 2 EOL_();
617 2167 2 FAO_(' SUBSYS = !UL', .BUF[BBH$W_SUBSYS]);
618 2168 2 EOL_();
619 2169 2 FAO_(' APPLIC = !UL', .BUF[BBH$W_APPLIC]);
620 2170 2 EOL_();
621 2171 2 FAO_(' NUMBER = !UL', .BUF[BBH$L_NUMBER]);
622 2172 2 EOL_();
623 2173 2 FAO_(' STRUCLEV = !XW', .BUF[BBH$W_STRUCLEV]);
```

```

: 624 2174 2 EOL_(:);
: 625 2175 2 FAO_(: VOLNUM = !UL', .BUF[BBH$W_VOLNUM]);
: 626 2176 2 EOL_(:);
: 627 2177 2 FAO_(: CRC = !XL', .BUF[BBH$L_CRC]);
: 628 2178 2 EOL_(:);
: 629 2179 2 FAO_(: BLOCKSIZE = !UL', .BUF[BBH$L_BLOCKSIZE]);
: 630 2180 2 EOL_(:);
: 631 2181 2 FAO_(: NOCRC = !AC', .FALSETRUE(.BUF[BBH$V_NOCRC]));
: 632 2182 2 EOL_(:);
: 633 2183 2 FAO_(: SSNAME = '!AC''', BUF[BBH$T_SSNAME]);
: 634 2184 2 EOL_(:);
: 635 2185 2 FAO_(: FID = !UL', !UL', !UL',
P 2186 2 .BUF[BBH$W_FID_NUM] + .BUF[BBH$B_FID_NMX] ^ 16,
P 2187 2 .BUF[BBH$W_FID_SEQ],
P 2188 2 .BUF[BBH$B_FID_RVN]);
: 639 2189 2 EOL_(:);
: 640 2190 2 FAO_(: DID = !UL', !UL', !UL',
P 2191 2 .BUF[BBH$W_DID_NUM] + .BUF[BBH$B_DID_NMX] ^ 16,
P 2192 2 .BUF[BBH$W_DID_SEQ],
P 2193 2 .BUF[BBH$B_DID_RVN]);
: 644 2194 2 EOL_(:);
: 645 2195 2 FAO_(: FILENAME = '!AC''', BUF[BBH$T_FILENAME]);
: 646 2196 2 EOL_(:);
: 647 2197 2 FAO_(: ATTRIB = !XL!XL', .(BUF[BBH$B_BKTSIZE]), .(BUF[BBH$B_RTYPE]));
: 648 2198 2 EOL_(:);
: 649 2199 2 FAO_(: FILESIZE = !UL', .BUF[BBH$L_FILESIZE]);
: 650 2200 2 EOL_(:);
: 651 2201 2 EOL_(:);
: 652 2202 2 EOL_(:);
: 653 2203 2 EOL_(:);
: 654 2204 2 ! Format the records contained in this buffer.
: 655 2205 2 !
: 656 2206 2 DEBLOCK(.BCB, ANALYZE_ONE_RECORD);
: 657 2207 1 END;

```

20	3D	20	20	20	20	20	20	45	5A	49	53	20	20	11	00959	P.AES:	.ASCII	<12>\Block header\	
												4C	55	21	00966	P.AET:	.ASCII	<17>\ SIZE	= !UL\
20	3D	20	20	20	20	20	53	59	53	50	4F	20	20	11	00975				
												4C	55	21	00978	P.AEU:	.ASCII	<17>\ OPSYS	= !UL\
20	3D	20	20	20	20	53	59	53	42	55	53	20	20	11	00987				
												4C	55	21	0098A	P.AEV:	.ASCII	<17>\ SUBSYS	= !UL\
20	3D	20	20	20	20	43	49	4C	50	50	41	20	20	11	00999				
												4C	55	21	0099C	P.AEW:	.ASCII	<17>\ APPLIC	= !UL\
20	3D	20	20	20	20	52	45	42	4D	55	4E	20	20	11	009AB				
												4C	55	21	009AE	P.AEX:	.ASCII	<17>\ NUMBER	= !UL\
20	3D	20	20	56	45	4C	43	55	52	54	53	20	20	11	009BD				
												4C	55	21	009C0	P.AEY:	.ASCII	<17>\ STRUCLEV	= !XW\
20	3D	20	20	20	20	4D	55	4E	4C	4F	56	20	20	11	009CF				
												4C	55	21	009D2	P.AEZ:	.ASCII	<17>\ VOLNUM	= !UL\
20	3D	20	20	20	20	20	20	20	43	52	43	20	20	11	009E1				
												4C	58	21	009E4	P.AFA:	.ASCII	<17>\ CRC	= !XL\
20	3D	20	45	5A	49	53	4B	43	4F	4C	42	20	20	11	009F3				
												4C	55	21	009F6	P.AFB:	.ASCII	<17>\ BLOCKSIZE	= !UL\
20	3D	20	20	20	20	20	43	52	43	4F	4E	20	20	11	00A05				
												4C	55	21	00A08	P.AFC:	.ASCII	<17>\ NOCRC	= !AC\

			64		00	FB	00088	CALLS	#0, LIST_EOL	:	2178	
				28	A2	DD	0008B	PUSHL	40(BUF)	:	2179	
				009D	C6	9F	0008E	PUSHAB	P.AFB	:		
			65		02	FB	00092	CALLS	#2, LIST_FAO	:		
			64		00	FB	00095	CALLS	#0, LIST_EOL	:	2180	
50	2C	A2	01		00	EF	00098	EXTZV	#^, #1, 24(BUF), R0	:	2181	
				F6B3	C640	DD	0009E	PUSHL	FALSETRUE[R0]	:		
				00AF	C6	9F	000A3	PUSHAB	P.AFC	:		
			65		02	FB	000A7	CALLS	#2, LIST_FAO	:		
			64		00	FB	000AA	CALLS	#0, LIST_EOL	:	2182	
				30	A2	9F	000AD	PUSHAB	48(BUF)	:	2183	
				00C1	C6	9F	000B0	PUSHAB	P.AFD	:		
			65		02	FB	000B4	CALLS	#2, LIST_FAO	:		
			64		00	FB	000B7	CALLS	#0, LIST_EOL	:	2184	
			7E	54	A2	9A	000BA	MOVZBL	84(BUF), -(SP)	:	2188	
			7E	52	A2	3C	000BE	MOVZWL	82(BUF), -(SP)	:		
			50	50	A2	3C	000C2	MOVZWL	80(BUF), R0	:		
			51	55	A2	9A	000C6	MOVZBL	85(BUF), R1	:		
	51		51		10	78	000CA	ASHL	#16, R1, R1	:		
					6140	9F	000CE	PUSHAB	(R1)[R0]	:		
				00D5	C6	9F	000D1	PUSHAB	P.AFE	:		
			65		04	FB	000D5	CALLS	#4, LIST_FAO	:		
			64		00	FB	000D8	CALLS	#0, LIST_EOL	:	2189	
			7E	5A	A2	9A	000DB	MOVZBL	90(BUF), -(SP)	:	2193	
			7E	58	A2	3C	000DF	MOVZWL	88(BUF), -(SP)	:		
			50	56	A2	3C	000E3	MOVZWL	86(BUF), R0	:		
			51	5B	A2	9A	000E7	MOVZBL	91(BUF), R1	:		
	51		51		10	78	000EB	ASHL	#16, R1, R1	:		
					6140	9F	000EF	PUSHAB	(R1)[R0]	:		
				00EF	C6	9F	000F2	PUSHAB	P.AFF	:		
			65		04	FB	000F6	CALLS	#4, LIST_FAO	:		
			64		00	FB	000F9	CALLS	#0, LIST_EOL	:	2194	
				5C	A2	9F	000FC	PUSHAB	92(BUF)	:	2195	
				0109	C6	9F	000FF	PUSHAB	P.AFG	:		
			65		02	FB	00103	CALLS	#2, LIST_FAO	:		
			64		00	FB	00106	CALLS	#0, LIST_EOL	:	2196	
				00DC	C2	DD	00109	PUSHL	220(BUF)	:	2197	
				00E0	C2	DD	0010D	PUSHL	224(BUF)	:		
				011D	C6	9F	00111	PUSHAB	P.AFH	:		
			65		03	FB	00115	CALLS	#3, LIST_FAO	:		
			64		00	FB	00118	CALLS	#0, LIST_EOL	:	2198	
				00E4	C2	DD	0011B	PUSHL	228(BUF)	:	2199	
				0132	C6	9F	0011F	PUSHAB	P.AFI	:		
			65		02	FB	00123	CALLS	#2, LIST_FAO	:		
			64		00	FB	00126	CALLS	#0, LIST_EOL	:	2200	
			64		00	FB	00129	CALLS	#0, LIST_EOL	:	2201	
				FC85	CF	9F	0012C	PUSHAB	ANALYZE_ONE_RECORD	:	2206	
					53	DD	00130	PUSHL	R3	:		
				00000000G	00	02	FB	00132	CALLS	#2, DEBLOCK	:	
						04	00139	RET		:	2207	

; Routine Size: 314 bytes, Routine Base: CODE + 0A9D


```

: 659      2208 1 %SBTTL 'ANALYZE - main analyze routine'
: 660      2209 1 GLOBAL ROUTINE ANALYZE: NOVALUE=
: 661      2210 1
: 662      2211 1 !++
: 663      2212 1
: 664      2213 1 ! FUNCTIONAL DESCRIPTION:
: 665      2214 1 !   This routine is the driver for analysis generation.
: 666      2215 1
: 667      2216 1 ! INPUT PARAMETERS:
: 668      2217 1 !   NONE
: 669      2218 1
: 670      2219 1 ! IMPLICIT INPUTS:
: 671      2220 1 !   NONE
: 672      2221 1
: 673      2222 1 ! OUTPUT PARAMETERS:
: 674      2223 1 !   NONE
: 675      2224 1
: 676      2225 1 ! IMPLICIT OUTPUTS:
: 677      2226 1 !   NONE
: 678      2227 1
: 679      2228 1 ! ROUTINE VALUE:
: 680      2229 1 !   NONE
: 681      2230 1
: 682      2231 1 ! SIDE EFFECTS:
: 683      2232 1 !   NONE
: 684      2233 1
: 685      2234 1 !--
: 686      2235 1
: 687      2236 2 BEGIN
: 688      2237 2
: 689      2238 2 LOCAL
: 690      2239 2     BCB,           ! Pointer to buffer control block
: 691      2240 2     CHK_SAVESET:  BYTE,           ! Check save set encryption on frst pass
: 692      2241 2     PSAREA:      VECTOR[P$SIZE]; ! Impure area
: 693      2242 2
: 694      2243 2 GLOBAL REGISTER
: 695      2244 2     P$ = 11:      REF VECTOR;      ! Impure area base register
: 696      2245 2
: 697      2246 2 BUILTIN
: 698      2247 2     FP;
: 699      2248 2
: 700      2249 2
: 701      2250 2 ! Establish the handler.
: 702      2251 2 !
: 703      2252 2 .FP = RESTORE_HANDLER;
: 704      2253 2
: 705      2254 2
: 706      2255 2 ! Initialize impure area.
: 707      2256 2 !
: 708      2257 2 P$ = PSAREA;
: 709      2258 2 LIST_DESC[0] = LIST_SIZE;
: 710      2259 2 LIST_DESC[1] = _LIST_BUFFER;
: 711      2260 2 CHK_SAVESET = 1;
: 712      2261 2
: 713      2262 2
: 714      2263 2 ! Do the listing.
: 715      2264 2 !

```

```

: 716 2265 2 INIT_IN_SAVE(FALSE);
: 717 2266 2 WHILE (BCB = READ_BUFFER()) NEQ 0 DO
: 718 2267 3 BEGIN
: 719 2268 3
: 720 2269 3 IF .QUAL[QUAL_SS_ENCRYP]
: 721 2270 3 THEN
: 722 2271 4 BEGIN
: 723 2272 4 | If we are decrypting an encrypted saveset and we don't yet have
: 724 2273 4 | the required decrypt context, do a special scan of the first block
: 725 2274 4 | to locate the backup summary record, extract the datakey information
: 726 2275 4 | and initialize the decryption context.
: 727 2276 4 |
: 728 2277 4 IF .CRYP DATA CODE EQLU 0
: 729 2278 5 THEN IF NOT (QUAL[QUAL_SS_ENCRYP] = CRYPTO_INIDEC(.BCB))
: 730 2279 4 THEN SIGNAL(BACKUP$_BACNOTENC);
: 731 2280 4
: 732 2281 4 | Decrypt the buffer
: 733 2282 4
: 734 2283 4 CRYPTO_DECR_BLOCK(.BCB);
: 735 2284 4
: 736 2285 4 END
: 737 2286 3 ELSE
: 738 2287 4 BEGIN
: 739 2288 4 |
: 740 2289 4 | Make sure that the save set is not encrypted.
: 741 2290 4 |
: 742 2291 4 IF .CHK_SAVESET
: 743 2292 4 THEN IF CRYPTO_CHKSAV(.BCB)
: 744 2293 4 THEN SIGNAL(BACKUP$_ENCSAVSET) ;
: 745 2294 4 CHK_SAVESET = 0 ;
: 746 2295 3 END;
: 747 2296 3
: 748 2297 3 ANALYZE_ONE_BUFFER(.BCB);
: 749 2298 3
: 750 2299 2 END;
: 751 2300 2
: 752 2301 2 FIN_IN_SAVE(FALSE);
: 753 2302 1 END;

```

		083C 00000	.ENTRY ANALYZE, Save R2,R3,R4,R5,R11	: 2209
	55 00000000G	00 9E 00002	MOVAB LIB\$SIGNAL, R5	:
	54 00000000'	EF 9E 00009	MOVAB QUAL+12, R4	:
	5E FEF8	CE 9E 00010	MOVAB -264(SP), SP	:
	6D 00000000G	00 9E 00015	MOVAB RESTORE_HANDLER, (FP)	: 2252
	5B	6E 9E 0001C	MOVAB PSAREA, PS	: 2257
	6B 0100	8F 3C 0001F	MOVZWL #256, (PS)	: 2258
04	AB 08	AB 9E 00024	MOVAB 8(R11), 4(PS)	: 2259
	53	01 90 00029	MOVB #1, CHK_SAVESET	: 2260
		7E D4 0002C	CLRL -(SP)	: 2265
	00000000G 00	01 FB 0002E	CALLS #1, INIT_IN_SAVE	:
	00000000G 00	00 FB 00035 1\$:	CALLS #0, READ_BUFFER	: 2266
	52	50 D0 0003C	MOVL R0, BCB	:
		54 13 0003F	BEQL 6\$:

ANALYZE
V04-000

Analyze a save set
ANALYZE - main analyze routine

L 4
15-Sep-1984 23:40:04
14-Sep-1984 11:53:45

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]ANALYZE.B32;1

Page 33
(6)

	2C	02	A4		04	E1	00041	BBC	#4, QUAL+14, 3\$:	2269
				06DC	C4	D5	00046	TSTL	CRYP_DATA_CODE	:	2277
					1B	12	0004A	BNEQ	2\$:	
					52	DD	0004C	PUSHL	BCB	:	2278
02	A4	01	00000000G		01	FB	0004E	CALLS	#1, CRYPTO_INIDEC	:	
					50	F0	00055	INSV	RO, #4, #1, QUAL+14	:	
					50	E8	0005B	BLBS	RO, 2\$:	
			00000000G		8F	DD	0005E	PUSHL	#BACKUP\$ BACNOTENC	:	2279
				65	01	FB	00064	CALLS	#1, LIB\$SIGNAL	:	
			00000000G		52	DD	00067	PUSHL	BCB	:	2283
				00	01	FB	00069	CALLS	#1, CRYPTO_DECR_BLOCK	:	
				15	1A	11	00070	BRB	5\$:	2269
					53	E9	00072	BLBC	CHK_SAVESET, 4\$:	2291
					52	DD	00075	PUSHL	BCB	:	2292
			00000000G		01	FB	00077	CALLS	#1, CRYPTO_CHKSAV	:	
				09	50	E9	0007E	BLBC	RO, 4\$:	
			00000000G		8F	DD	00081	PUSHL	#BACKUP\$ ENCSAVSET	:	2293
				65	01	FB	00087	CALLS	#1, LIB\$SIGNAL	:	
					53	94	0008A	CLRB	CHK_SAVESET	:	2294
					52	DD	0008C	PUSHL	BCB	:	2297
			FE33	CF	01	FB	0008E	CALLS	#1, ANALYZE_ONE_BUFFER	:	
					A0	11	00093	BRB	1\$:	2266
					7E	D4	00095	CLRL	-(SP)	:	2301
			00000000G		01	FB	00097	CALLS	#1, FIN_IN_SAVE	:	
					04	0009E		RET		:	2302

; Routine Size: 159 bytes, Routine Base: CODE + 0BD7

ANALYZE
V04-000

Analyze a save set
ANALYZE - main analyze routine

M 4
15-Sep-1984 23:40:04
14-Sep-1984 11:53:45

VAX-11 Blis-32 V4.0-742
[BACKUP.SRC]ANALYZE.B32;1

Page 34
(7)

: 755 2303 1 END
: 756 2304 0 ELUDOM

.EXTRN LIB\$SIGNAL

PSECT SUMMARY

Name	Bytes	Attributes
COMMON	2124	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, OVR, NOPIC, ALIGN(2)
CODE	3190	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	9	0	581	00:01.2

COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:ANALYZE/OBJ=OBJ\$:ANALYZE MSRC\$:ANALYZE/UPDATE=(ENH\$:ANALYZE)

: Size: 1398 code + 3916 data bytes
: Run Time: 00:37.7
: Elapsed Time: 01:44.5
: Lines/CPU Min: 3662
: Lexemes/CPU-Min: 38872
: Memory Used: 382 pages
: Compilation Complete

0010 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

Grid of 100 terminal windows (10x10) showing various system utilities and data. Visible titles include:

- BACKUPMSG LIS
- ANALYZE LIS
- BUFFERS LIS
- CREATEDIR LIS
- BADBLOCK LIS
- BACKUPCMD LIS
- COMMAND LIS

Each window contains text-based data, including lists, tables, and status reports.