

0 (IT-OV - OVERLAY PREVIX
1 (IT-OV - IFIELDS, AFIELDS
2 (IT-OV - CLRPAD PAD+INDENT box constants
3
4 (IT-OV - IDENT-BOX\$ IDENT-SHIP\$ IDENT-LIFE\$
5 (IT-OV - IDENT-VESSEL\$ U)\$
6 (IT-OV - VAL,VOL>PAD
7 (IT-OV - IDENT-ELEMENT\$ DATE\$>ADR
8 (IT-OV - IDENT-MESSAGE\$ IDENT-ARTIFACTS\$
9 (IT-OV - IDENT-SPECIMEN\$ IDENT-BIO-DATA\$
10 (IT-OV - IDENT-ITEM\$
11 (IT-OV - ITEM>PAD
12 (IT-OV - TEXT>PAD
13 (IT-OV - >SCROLL-LIST
14 (IT-OV - BOX>LIST GET-BOXES
15 (IT-OV - MAKE-SCROLL-BOX DELETE-SCROLL-BOX
16 (IT-OV - CLASS>BOX-SPEC
17 (IT-OV - MAKE-CLASS-BOX BOX>TOCS
18 (IT-OV - >BOX
19 (IT-OV - ?ELEMENT {BOX>}
20 (IT-OV - BOX>
21 (IT-OV - OVERLAY SUFFIX -----

0

3

0 (IT-OV - OVERLAY PREVIX
 1
 2 VOCABULARY IT-VOC IMMEDIATE
 3
 4 131 OPEN-OVERLAY
 5
 6 IT-VOC DEFINITIONS
 7
 8 1600 TRANS-ALLOT
 9
 10 NEWT-DP
 11
 12
 13
 14
 15

10-1-85)

1

4

0 (IT-OV - IFIELDS, AFIELDS
 1 EXIT
 2 0 17 2 IFIELD: INST-VAL 68 4 3 AFIELD: SHAPE
 3 0 19 2 IFIELD: INST-DATE 68 146 3 AFIELD: RESEMBLES
 4 11 0 16 AFIELD: BOX-NAME
 5 20 52 15 IFIELD: XNAME
 6 27 0 16 AFIELD: ORIG-NAME
 7 40 0 16 AFIELD: SPEC-NAME
 8 43 0 16 AFIELD: BD-NAME
 9 26 0 16 AFIELD: ELEM-NAME
 10 26 16 2 AFIELD: ELEM-VAL
 11 28 0 24 AFIELD: ART-NAME 28 29 1 AFIELD: ART-ANLYZ
 12 28 27 2 AFIELD: ART-VAL
 13 28 25 2 AFIELD: ART-VOL
 14 48 11 1 IFIELD: PHR-CNT
 15 48 12 254 IFIELD: PHRASE

10-1-85) (IT-OV - IDENT-BOX\$ IDENT-SHIP\$ IDENT-LIFE\$

10-1-85)

HEAD: IDENT-BOX\$ (-- \ build box string)
 T: (identifier at PAD)
 CLRPAD (clear line leaders) BOX-NAME PAD+INDENT 16 CMOVE T;
 \ HEAD: IDENT-SHIP\$ (-- \ build ship string)
 \ T: (identifier at PAD)
 \ " ISS " PAD+INDENT SWAP CMOVE XNAME PAD+INDENT 4 + 15 CMOVE
 \ T;
 \ HEAD: IDENT-RUIN\$
 \ T: @INST-SPECIES 2 MOD IF " RECENT" ELSE " ANCIENT"
 \ THEN PAD+INDENT SWAP CMOVE T;
 HEAD: IDENT-LIFE\$ T: RESEMBLES 1.5@ 2DUP OR 0=
 IF 2DROP SHAPE 1.5@ THEN >C+S PHRASE PHR-CNT C@
 PAD+INDENT SWAP CMOVE ICLOSE T;

2

5

0 (IT-OV - CLRPAD PAD+INDENT box constants
 1 28 29 1 AFIELD: ART-ANLYZ
 2
 3 10 C: B.ELEM 15 C: B.ART 14 C: B.SPEC
 4 17 C: B.MSG 11 C: B.LIFE 38 C: B.BIO
 5 16 C: B.RUINS
 6
 7
 8
 9 HEAD: CLRPAD (-- \ blank line buffer)
 10 T: PAD 1+ 38 BL FILL T;
 11
 12 HEAD: PAD+INDENT (-- u \ pad+depth indent)
 13 T: SRDEPTH 0 MAX 2* PAD + 1+ T;
 14
 15

10-1-85) (IT-OV - IDENT-VESSEL\$ U)\$

10-1-85)

EXIT
 \ HEAD: IDENT-VESSEL\$ (-- \ build vessel)
 \ T: (identifier at PAD - *** REVISE ***)
 \ CLRPAD
 \ " ALIEN VESSEL" PAD+INDENT SWAP
 \ CMOVE T;
 : U)\$ (u -- adr u \ convert single)
 (length number to string)
 0 <# #S #>;

6

```

0 ( IT-OV - VAL,VOL>PAD
1
2 HEAD: VAL,VOL>PAD ( val vol -- \ convert)
3 T: ( to strings and move to pad using)
4 ( identifier format)
5 10 /MOD SWAP ( 10ths & 1s )
6 U>$ DROP ( 10ths addr -- )
7 C@ PAD 32 + C! ( fraction)
8 ASCII . PAD 31 + C! ( decimal point)
9 U>$ ( 1s addr u -- )
10 PAD 31 + OVER - SWAP CMOVE
11 XVAL @ U* 100 U/MOD SWAP DROP ( value adjustment)
12 U>$ ( adr u -- )
13 PAD 38 + OVER 1- - SWAP CMOVE T;
14
15

```

7

```

0 ( IT-OV - IDENT-ELEMENT$ DATE$>ADR
1
2 HEAD: IDENT-ELEMENT$ ( -- \build element)
3 T: ( identifier at PAD)
4 ELEM-NAME PAD+INDENT 16 CMOVE
5 INST-QTY @ ELEM-VAL @ 10 * ( vol val)
6 SWAP VAL,VOL>PAD T;
7
8 : DATE$>ADR ( date adr -- \ build a )
9 ( date string at adr w/o count byte)
10 " 00-00-" 3 PICK SWAP CMOVE
11 SWAP 300 /MOD 4620 + ( adr rem yr --)
12 U>$ 4 PICK 6 + SWAP CMOVE
13 30 /MOD 1+ U>$ ( adr day-1 adr u --)
14 4 PICK 4 + OVER 1- - SWAP CMOVE
15 1+ U>$ ROT 1+ OVER 1- - SWAP CMOVE ;

```

8

```

0 ( IT-OV - IDENT-MESSAGE$ IDENT-ARTIFACTS$
1
2 HEAD: IDENT-MESSAGE$ ( -- \ build message)
3 T: ( identifier string at PAD)
4 CLRPAD ( clear line leaders)
5 INST-DATE @ PAD+INDENT DATE$>ADR
6 ORIG-NAME PAD+INDENT 11 + 16 CMOVE T;
7
8 HEAD: UKN-ART T: PAD 36 + 3 ASCII ? FILL T;
9
10 HEAD: IDENT-ARTIFACT$ ( -- \ build relic)
11 T: ( identifier at PAD)
12 ART-NAME PAD+INDENT 24 CMOVE
13 ART-ANLYZ C@ ???IT @ OR 0=
14 ART-VAL @ OVER NOT * ART-VOL @ VAL,VOL>PAD
15 IF UKN-ART THEN T;

```

9

```

10-1-85) ( IT-OV - IDENT-SPECIMEN$ IDENT-BIO-DATA$
10-1-85)
HEAD: IDENT-SPECIMEN$ ( -- \build specimen)
T: ( identifier at PAD)
SPEC-NAME PAD+INDENT 16 CMOVE
INST-VAL @ INST-QTY @ VAL,VOL>PAD T;
HEAD: IDENT-BIO-DATA$ ( --\build biodata)
T: ( identifier at PAD)
BD-NAME PAD+INDENT 16 CMOVE
INST-VAL @ INST-QTY @ VAL,VOL>PAD T;

```

10

```

10-1-85) ( IT-OV - IDENT-ITEM$
10-1-85)
CASE IDENT-ITEM$ ( class -- )
  11 IS IDENT-BOX$
  \ 20 IS IDENT-SHIP$
  \ 25 IS IDENT-VESSEL$
  26 IS IDENT-ELEMENT$
  27 IS IDENT-MESSAGE$
  28 IS IDENT-ARTIFACT$
  40 IS IDENT-SPECIMEN$
  \ 41 IS IDENT-RUIN$
  68 IS IDENT-LIFE$
  43 IS IDENT-BIO-DATA$
  OTHERS UNRAVEL

```

11

10/22/85) (IT-OV - ITEM>PAD

10-1-85)

```

HEAD: ITEM>PAD ( --u\ build the identifier )
T: ( for the current instance in the )
  ( format {name} {vol} {val}, indent)
  ( and adjust value by %val. at PAD.)
  CLRPAD ( blank fill buffer)
  PAD+INDENT PAD 30 + OVER - ASCII .
  FILL ( line leaders)
  38 PAD C! ( string count)
  @INST-CLASS
  IDENT-ITEM$ T;

```

12

```

0 ( IT-OV - TEXT)PAD
1 EXIT
2 ( Text Instances for Window & Scroll
3
4
5 FILE: SCROLL-TEXT 11 3 IFIELD TEXT-CONT
6 FILE: SCROLL-TEXT 14 3 IFIELD TEXT-INST
7 FILE: SCROLL-TEXT 17 38 IFIELD TEXT-TEXT
8
9 : TEXT)PAD ( -- )
10 TEXT-TEXT PAD 1+ 38 CMOVE 38 PAD C! ;
11
12
13
14
15

```

13

```

0 ( IT-OV - >SCROLL-LIST
1 ( Instance Text to Scroll Box >SCROLL-LIST
2
3 HEAD: >SCROLL-LIST ( iaddr iaddr' $nopack ( -- ) )
4 T: (SCROLL-BOX) 1.5@ >C+S IOPEN
5 FILE: SCROLL-TEXT LITERAL 0 1 *CREATE
6 TEXT-TEXT 38 CMOVE
7 TEXT-INST 1.5! TEXT-CONT 1.5!
8 ICLOSE 1 SCROLL-LEN +! ICLOSE T;
9
10 \ appends a SCROLL-TEXT instance to the contents of the box
11 \ whose iaddress is in (SCROLL-BOX).
12
13
14
15

```

14

```

0 ( IT-OV - BOX>LIST GET-BOXES
1 ( Instance Text to Scroll Box
2 : BOX>LIST ( ( iaddr' iaddr -- iaddr' iaddr ) )
3 SUBROOT CI
4 BEGIN CI' CI ITEM)PAD PAD 1+ >SCROLL-LIST
5 NEXT-NODE 2DUP CI D=
6 UNTIL 2DROP ;
7
8 \ appends the box iaddr and its contents to the contents of
9 \ the box whose iaddress is in (SCROLL-BOX).
10
11 : GET-BOXES ( iaddr -- )
12 >C+S IOPEN
13 BEGIN BOX>LIST INEXT ?FIRST UNTIL CDROP ICLOSE ;
14
15 \ does BOX>LIST for all instances in super-box iaddr.

```

15

```

10-1-85) ( IT-OV - MAKE-SCROLL-BOX DELETE-SCROLL-BOX
      ( Text Instances for Window & Scroll
      AWK )
      : MAKE-SCROLL-BOX ( ( -- ) )
      FRAGMENT >C+S IOPEN
      11 56 1 *CREATE CI (SCROLL-BOX) 1.5!
      CI' (SCROLL-CONT) 1.5! SCROLL-LEN OFF CDROP ICLOSE ;

      \ creates a box to hold SCROLL-TEXT; box is in FRAGMENT;
      \ (SCROLL-CONT) holds box's iaddr; CAUTION:
      \ kills last iaddr in (SCROLL-CONT) (see DELETE-SCROLL-BOX).
      : DELETE-SCROLL-BOX ( ( -- ) )
      (SCROLL-CONT) 1.5@ >C (SCROLL-BOX) 1.5@ >C+
      SET-CURRENT IDELETE CDROP ICLOSE ;

      \ deletes SCROLL-TEXT box whose iaddr is in (SCROLL-CONT);

```

16

```

10-1-85) ( IT-OV - CLASS>BOX-SPEC
      AWK )
      CASE CLASS>BOX-SPEC
      26 IS B.ELEM ( elements )
      28 IS B.ART ( artifacts )
      40 IS B.SPEC ( specimens )
      41 IS B.RUINS ( ruins )
      43 IS B.BIO ( bio-data )
      27 IS B.MSG ( messages )
      68 IS B.LIFE ( lifeforms )
      OTHERS UNRAVEL
      \ translate an instance class to its box species.

```

17

```

10-1-85) ( IT-OV - MAKE-CLASS-BOX BOX>TOCS
      AWK )
      HEAD: MAKE-CLASS-BOX ( class ( n n' -- n n' ) )
      T: CLASS>BOX-SPEC 11 SWAP 1 *CREATE T;

      \ creates proper species box for instance n' class.

      : BOX>TOCS ( class ( n -- n' ) )
      CLASS>BOX-SPEC 11 SWAP IFIND DROP ;

      \ put proper species box for instance n' class on top of
      \ context stack.

```


18

```

0 ( IT-OV - >BOX
1 : >BOX ( iaddr ( n -- n ) )
2 2DUP >C+S @INST-CLASS >R ICLOSE
3 IOPEN 11 I CLASS>BOX-SPEC IFIND 0=
4 IF I MAKE-CLASS-BOX THEN R> BOX>TOCS
5 2DUP >C+S @INST-SPECIES @INST-CLASS ICLOSE
6 DUP 26 =
7 IF OVER IOPEN IFIND ICLOSE
8 IF IOPEN
9 BEGIN DUP INEXT @INST-SPECIES = UNTIL DROP
10 >C+S INST-QTY @ ICLOSE INST-QTY +! ICLOSE
11 ELSE DROP CI IINSERT THEN
12 ELSE 2DROP CI IINSERT THEN ICLOSE ;
13
14 \ put iaddr (and a matching box if necessary) into super-box n.
15

```

19

```

0 ( IT-OV - ?ELEMENT {BOX>}
1
2
3 : ?ELEMENT ( { n -- n } f ) @INST-CLASS 26 = ;
4
5 : {BOX>} ( { n" n' n -- n" n' n' } iaddr )
6 ?ELEMENT INST-QTY @ ELEM-AMT @ > AND
7 IF ELEM-AMT @ NEGATE INST-QTY +! 26 @INST-SPECIES ICREATE
8 >C+S ELEM-AMT @ INST-QTY ! C>
9 ELSE IEXTRACT THEN ;
10
11 \ take instance on TOCS from its box and place it on the
12 \ parameter stack.
13
14
15

```

20

```

0 ( IT-OV - BOX>
1
2 : BOX> ( { n" n' n -- n"/n'/n } f iaddr )
3 {BOX>} CI OR 0=
4 IF ICLOSE IDELETE CI OR 0=
5 IF ICLOSE THEN 1
6 ELSE 0 THEN ROT ROT ;
7
8 \ perform {BOX>} and then if box is empty, delete it and set
9 \ flag to TRUE; if no boxes in super-box, CLOSE super-box.
10
11
12
13
14
15

```

21

```

10-1-85) ( IT-OV - OVERLAY SUFFIX ----- 10-1-85)
DISPOSE
CLOSE-OVERLAY
131 OVERLAY IT-OV
IT-OV
NEWT-DP
FORTH DEFINITIONS
HEAD: IT1 T: IT-OV IT-VOC DATE#>ADR OV-CANCEL T;
HEAD: IT2 T: IT-OV IT-VOC BOX>LIST OV-CANCEL T;
HEAD: IT3 T: IT-OV IT-VOC GET-BOXES OV-CANCEL T;
HEAD: IT4 T: IT-OV IT-VOC MAKE-SCROLL-BOX OV-CANCEL T;
HEAD: IT5 T: IT-OV IT-VOC DELETE-SCROLL-BOX OV-CANCEL T;
HEAD: IT6 T: IT-OV IT-VOC BOX>TOCS OV-CANCEL T;
HEAD: IT7 T: IT-OV IT-VOC >BOX OV-CANCEL T;
HEAD: IT8 T: IT-OV IT-VOC BOX> OV-CANCEL T;
HEAD: IT9 T: IT-OV IT-VOC {BOX>} OV-CANCEL T;

```

22

```

10-1-85) ( IT-OV - OVERLAY SUFFIX ----- 10-1-85)
HEAD: ITA T: IT-OV IT-VOC CLASS>BOX-SPEC OV-CANCEL T;
OV-CANCEL
7 WIDTH !
: DATE#>ADR ' IT1 MODULE ;
: BOX>LIST ' IT2 MODULE ;
: GET-BOXES ' IT3 MODULE ;
: MAKE-SCROLL-BOX ' IT4 MODULE ;
: DELETE-SCROLL-BOX ' IT5 MODULE ;
: BOX>TOCS ' IT6 MODULE ;
: >BOX ' IT7 MODULE ;
: BOX> ' IT8 MODULE ;
: {BOX>} ' IT9 MODULE ;
: CLASS>BOX-SPEC ' ITA MODULE ;
DISPOSE
31 WIDTH !

```

10-1-85)