

**British Columbia Ministry of Sustainable
Resource Management
Terrestrial Information Branch**

Vegetation Resources Inventory 2000

Report: VGIS Data Validation Rule Set

Run By: VGIS

Report Date: OCT-28-03 08:20 AM

Database Instance: Production V1.0

Total Pages: 207

Parameter Values

Data Format: VEG INV GROUND SAMPLE 2000

Record Type: %

Record: VEG CH 01 1**Field:** RECORD ID**Datatype:** NUMBER**Position:** 1**Length:** 3**Optional:** N**Zero to Null:****Record:** VEG CH 01 1**Field:** PAGE NUMBER**Datatype:** NUMBER**Position:** 4**Length:** 2**Optional:** N**Zero to Null:****Rule Name:** Veg 2000 Validate Page Number**PL/SQL Source Code:**

```
--
-- This rule was migrated in February 2000 from the 1999 VRI Format
-- applied against the extended Sample Plot database.
--
```

```
p_no NUMBER;
p_total NUMBER;
rec_id VARCHAR2(10);
rec INTEGER;
transact NUMBER;
prev_transact NUMBER;
prev_p_no NUMBER;
prev_p_total NUMBER;
BEGIN
  BEGIN
    p_no := ffio.getn('PAGE NUMBER');
  EXCEPTION WHEN VALUE_ERROR THEN
    p_return_code := 1;
    p_return_message := ' Page number field ***' ||
      ffio.getc('PAGE NUMBER') || '*** is non numeric.';
    RETURN;
  END;

  BEGIN
    p_total := ffio.getn('TOTAL PAGES');
  EXCEPTION WHEN VALUE_ERROR THEN
    p_return_code := 2;
    p_return_message := 'Value of Total pages field ***' ||
      ffio.getc('TOTAL PAGES') || '*** is non numeric.';
    RETURN;
  END;

  rec_id := ffio.getc('RECORD ID');

  IF ( rec_id IN ('041', '061')) THEN
    BEGIN
      transact := ffio.getn('TRANSECT#');
    EXCEPTION WHEN VALUE_ERROR THEN
      p_return_code := 3;
      p_return_message := ' For VEG Card ' || ffio.getc('RECORD ID') ||
        ', the value of ***' || ffio.getc('TRANSECT#') ||
```

Record: VEG CH 01 1**Field:** SEQUENCE NUMBER**Datatype:** NUMBER**Position:** 6**Optional:** N**Length:** 4**Zero to Null:****Record:** VEG CH 01 1**Field:** PROJECT ID**Datatype:** CHAR**Position:** 10**Optional:** N**Length:** 4**Zero to Null:****Rule Name:** Veg 2000 Lookup Project Id**PL/SQL Source Code:**

```
--
-- Feb 20,2002 RGH changed projects to VGIS_projects on move to MSRM
--
input_project VARCHAR2(10);
BEGIN
input_project := ffio.getc('PROJECT ID');
BEGIN
    SELECT TO_CHAR(project_skey)
    INTO p_return_value
    FROM vgis_projects
    WHERE project_business_id = input_project;
    p_return_code:= 0;
EXCEPTION
WHEN NO_DATA_FOUND
THEN
    p_return_code := 1;
    p_return_message := 'The project ID *"'
        || input_project
        || '"** was not found in the projects codes table, or it is'
        || ' not associated with an investigation initiative. If it'
        || ' is a valid project ID, contact the Data Resources'
        || ' Manager';
WHEN TOO_MANY_ROWS
THEN
    p_return_code := 2;
    p_return_message := 'Multiple projects with ID *"'
        || input_project
        || '"** were found in the projects code table.'
        || ' Contact the Data Resources Manager';
END;
```

Rule Name: Veg 2000 CH 01 1 Look Up MoF Agency**PL/SQL Source Code:**

```
--
-- Feb 20,2002 RGH changed agencies to VGIS_agencies on move to MSRM
--

BEGIN
BEGIN
    SELECT TO_CHAR(agency_skey)
    INTO p_return_value
    FROM vgis_agencies
```

Record: VEG CH 01 1**Field:** PLOT SAMPLE NO**Datatype:** NUMBER**Position:** 14**Length:** 4**Optional:** N**Zero to Null:****Record:** VEG CH 01 1**Field:** PL**Datatype:** CHAR**Position:** 18**Length:** 1**Optional:** Y**Zero to Null:****Domain Name:** VEG SAMPLE PL - PLOT NO**Value****Meaning**

E

I

M

N

S

W

Record: VEG CH 01 1**Field:** SAMPLE TYPE**Datatype:** CHAR**Position:** 19**Length:** 1**Optional:** N**Zero to Null:****Domain Name:** VEG SAMPLE PLOT TYPE**Value****Meaning**

A

Audit (1997 and 1998)

C

Coarse woody debris

D

Timber Emphasis plus CWD

E

Ecological sample

F

National Forest Inventory

I

Intensive (1997 and 1998)

M

Management Level Monitoring

N

Net volume adjustment sample

O

VRI sample

Q

Timber Emphasis sample

R

Range sample

T

Timber Emphasis sample for 97 and Old growth for 20

W

Within polygon variation

Z

Timber Emphasis plus Site Series

Rule Name: Veg 2000 CH 01 1 Check for an Existing Plot Cluster**PL/SQL Source Code:**

--

-- Feb 20,2002 RGH changed projects to VGIS_projects on move to MSRM

--

```

proj      VARCHAR2(10);
sample    VARCHAR2(10);
s_type    VARCHAR2(10);
assoc_type VARCHAR2(10);
num_in_db  INTEGER;

```

Rule Name: Veg 2000 CH 01 1 Check for Required Data (VRI Cards) for the Sam
PL/SQL Source Code:

```
--
-----
--
-- Veg 1999 CH 01 1 Check for Required Data (VRI Cards) for the Sample Type
--
-- Modifications
--
-- Feb 5, 1999 TMC - wo 665, for sample type A, card Veg RS 04 is not required
--
-- Feb 9, 1999 TMC - wo 691, for sample type Q, card Veg TA 11 is not required
--
-- Jun 10, 1999 TMC - wo 675, add presence checks for new sample types Z, C, D,
--                      E, and W
--
-- Mar 8, 2000 RGH - Added sample type T for Veg 2000
-- Jul 5, 2000 RGH - ADDED sample type M(Management Level Monitoring)
--                      and F(National Forest Inventory)
-- Jul 19, 2000 CAW - For sample Type F, check for the 5 and 7 and fail for
--                      presence of the 11
-----
--
```

```
card_3 NUMBER;
card_4 NUMBER;
card_5 NUMBER;
card_6 NUMBER;
card_7 NUMBER;
card_8 NUMBER;
card_9 NUMBER;
card_10 NUMBER;
card_11 NUMBER;
card_12 NUMBER;
card_14 NUMBER;
card_15 NUMBER;
card_16 NUMBER;
VarMissingCards VARCHAR2(256);
VarMissingCardsSep VARCHAR2(2);
VarSrceDataSampleType VARCHAR2(5);
```

BEGIN

```
p_return_code := 0;
```

```
-- the Veg CH 01 and CP 02 cards are required for all sample types
```

```
IF ( ( ffio.getc ( 'RECORD ID' ) != '011' )
    OR ( ffio.get_next_record_by_item ( '021' ) <= 0 ) )
```

```
THEN
```

```
    p_return_code := 1;
```

```
    p_return_value := 'all sample types must have cards CH (1) and CP (2) in the'
```

Rule Name: Veg 2000 CH 01 1 Check Sample Type

PL/SQL Source Code:

```

s_type VARCHAR2(10);
BEGIN
p_return_code := 0;
p_return_value := NULL;
    s_type := ffio.getc('SAMPLE TYPE');

    IF s_type in ('O','R','Q','D','Z','E','C','N','W','T','M','F') THEN
        RETURN;
    ELSE
        p_return_code := 1;
        p_return_message := 'The SAMPLE TYPE ***' || s_type
            || '*** is not valid for VEG 2000 Samples.' ||
            ' Accepted types are
(O,R,Q,D,Z,E,C,N,W,T,M,F).';
        RETURN;
    END IF;

```

Rule Name: Veg 2000 Check the Persons Credentials

PL/SQL Source Code:

```

-----
-----
--
-- NOV.22,2000 RGH wo.891 Rule added to check Sample Type with FIELD
RESPONSIBILITY
-- field against the SAMPLING_CERTIFICATION table
--
-- NOV 24,2000 RGH removed SPD. Schema pointer from Sampling_Certifications table
-- to comply with portability
--SEPT 10,2001 RGH wo,961 edit error message error point 12-- switch Type 1&2
--
-----
-----
VarSrceDataPageNo NUMBER;
VarSrcOriRecNo NUMBER;
VarNextRec1B BOOLEAN;
VarNextRec2B BOOLEAN;
VarNextRec1 NUMBER;
VarNextRec2 NUMBER;
VarNowCurrentRec NUMBER;
VarNextname1 VARCHAR2(256);
VarNextname2 VARCHAR2(256);
plot_type VARCHAR2(2);
VarNextRecType1 VARCHAR2(256);
VarNextRecType2 VARCHAR2(256);
cert_type VARCHAR2(256);
ret_code NUMBER;
result NUMBER;
ReturnKey1 NUMBER;
ReturnKey2 NUMBER;
Match_skey NUMBER;
step_rec NUMBER;

BEGIN

```

Record: VEG CH 01 1**Field:** SAMPLE INTENT**Datatype:** CHAR**Position:** 20**Length:** 1**Optional:** N**Zero to Null:****Rule Name:** Veg 1999 CH 01 1 Check That the Sample Intent is Valid**PL/SQL Source Code:**

```

--
-----
--
-- Veg 1999 CH 01 1 Check That the Sample Intent is Valid
--
-----
--

VarSrceDataPageNo NUMBER;
VarSrceDataSampleIntent VARCHAR2(1);

BEGIN

-- initialize this function's return value, code, and message
p_return_value := NULL;
p_return_code := 0;
p_return_message := NULL;

-- get the page number value
VarSrceDataPageNo := ffio.getn ( 'PAGE NUMBER' );

-- get the sample intent
VarSrceDataSampleIntent := ffio.getc;

-- check if the sample intent is among the valid values
IF ( VarSrceDataSampleIntent IN ( 'A', 'M', 'O', 'R' ) )
THEN
    -- the sample intent is among the valid values

    -- this is a good thing

    -- continue processing
    NULL;
ELSE
    -- the sample intent is not among the valid values

    -- this is an erroneous condition

    -- set this function's return code and message to indicate the error
    -- condition
    p_return_code := 1;
    p_return_message := 'on page '
        || TO_CHAR ( VarSrceDataPageNo )
        || ', the sample intent of '
        || VarSrceDataSampleIntent

```

Record: VEG CH 01 1**Field:** MEASUREMENT NUMBER**Datatype:** NUMBER**Position:** 21**Optional:** N**Length:** 1**Zero to Null:****Record:** VEG CH 01 1**Field:** MEASUREMENT DATE**Datatype:** DATE**Date Mask:** YYYYMONDD**Position:** 22**Optional:** N**Length:** 9**Zero to Null:****Domain Name:** DATE MIN JAN 01 1997 - FULL**Value**

1997JAN01 - 3999JAN01

Meaning

minum date for 1999 Ground Sampling Format - YYM

Record: VEG CH 01 1**Field:** CREW PERSON #1**Datatype:** CHAR**Position:** 31**Optional:** N**Length:** 4**Zero to Null:****Rule Name:** Veg 2000 Check the Person's Initials**PL/SQL Source Code:**

```

name VARCHAR2(256);
ret_code NUMBER;
result NUMBER;
field_val VARCHAR2(100);
name_sign VARCHAR2(10);
BEGIN
p_return_code := 0;
name := ffio.getc;
field_val := ffio.get_current_item_name;
IF ( name IS NULL ) THEN
    p_return_code := 0;
    p_return_value := NULL;
    RETURN;
END IF;

result := NULL;
coms.GetPersonSkey ( name, ret_code, result );

IF ( ret_code > 1 ) THEN
    p_return_code := ret_code;
    P_return_message := 'The field ' || field_val ||
        ' has ' || ret_code || ' values for "' || name ||
        '***. Contact VRI Data Coordinator to correct the problem. ';
ELSIF ( ret_code <= 0 OR result IS NULL ) THEN
    p_return_code := 1;
    IF field_val LIKE 'CREW%'
    THEN
        name_sign := 'initials';
    ELSE
        name_sign := 'name';
    END IF;
    p_return_message := 'The '
    || name_sign

```

Rule Name: Veg 2000 Check CH1 REC2 Field Responsibility

PL/SQL Source Code:

```
--
-- Apr 11, 2000 RGH wo.822, Rule created to check that CREW MEMBER #1
--                  initials match the full name for FIELD RESPONSIBILITY
--
-- Apr 18, 2000 RGH wo. 844, Modify to check if either Fild Responsibility
--                  for TREE DATA or ECOLOGICAL matches CREW INITIAL #1 of card
01 1
--
-- VarSrceDataCurRec  VARCHAR2(20);
VarSrceDataCrewRespond1  VARCHAR2(40);
VarSrceDataCrewRespond2  VARCHAR2(40);
VarSrceDataCrewInitial   VARCHAR2(10);
VarSrceDataRecType1     NUMBER;
VarSrceDataRecType2     NUMBER;
VarSrceDataCrewName      VARCHAR2(40);
VarSrceDataRespondType1 VARCHAR2(40);
VarSrceDataRespondType2 VARCHAR2(40);
VarSrceDataPageNo       NUMBER;
BEGIN
--VarSrceDataCurRec := ffio.getc('RECORD ID');
VarSrceDataCrewInitial := ffio.getc;
VarSrceDataPageNo := ffio.getn('PAGE NUMBER');
p_return_code := 0;
VarSrceDataRecType1 := NULL;
VarSrceDataRecType2 := NULL;

VarSrceDataRecType1 := ffio.get_next_record_by_item('012', 'CH CARD REC TYPE',
'TREE DATA');
VarSrceDataRecType2 := ffio.get_next_record_by_item('012', 'CH CARD REC TYPE',
'ECOLOGICAL');

BEGIN
select (FIRST_NAME || ' ' || LAST_NAME)
into VarSrceDataCrewName
from PEOPLE
where PER_INITIAL = VarSrceDataCrewInitial;
EXCEPTION WHEN NO_DATA_FOUND THEN
p_return_code:= 1;
p_return_message :=
' The Crew # 1 initials ** '
|| VarSrceDataCrewInitial
|| ' ** not found in the people table';
p_return_value := null;

WHEN TOO_MANY_ROWS THEN
p_return_code := 2;
p_return_message :=
' There is multiple people'
|| 'found with initials ***'
|| VarSrceDataCrewInitial
|| '*** -- Contact System Administrator';
p_return_value := null;

END;
```

Record: VEG CH 01 1**Field:** CREW PERSON #2**Datatype:** CHAR**Position:** 35**Length:** 4**Optional:** Y**Zero to Null:****Rule Name:** Veg 2000 Check the Person's Initials**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG CH 01 1**Field:** CREW PERSON #3**Datatype:** CHAR**Position:** 39**Length:** 4**Optional:** Y**Zero to Null:****Rule Name:** Veg 2000 Check the Person's Initials**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG CH 01 1**Field:** CREW PERSON #4**Datatype:** CHAR**Position:** 43**Length:** 4**Optional:** Y**Zero to Null:****Rule Name:** Veg 2000 Check the Person's Initials**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG CH 01 1**Field:** TOTAL PAGES**Datatype:** NUMBER**Position:** 47**Length:** 2**Optional:** Y**Zero to Null:****Record:** VEG CH 01 1**Field:** POLYGON IDENTIFIER**Datatype:** CHAR**Position:** 49**Length:** 14**Optional:** Y**Zero to Null:****Record:** VEG CH 01 1**Field:** REFERENCE TREE TAG NUMBER**Datatype:** NUMBER**Position:** 63**Length:** 6**Optional:** N**Zero to Null:** N**Record:** VEG CH 01 1**Field:** PLOT CLUSTER INTEGRATED**Datatype:** CHAR**Position:** 69**Length:** 1**Optional:** N**Zero to Null:****Domain Name:** INDICATOR**Value**

N

Meaning

No (False)

Y

Yes (True)

Rule Name: Veg 2000 Validate Plot Cluster Data Collected Indicators**PL/SQL Source Code:**

```

cur_field VARCHAR2(256);
cur_value VARCHAR2(10);
last_word VARCHAR2(80);
rec INTEGER;
rec_msg VARCHAR2(20);
BEGIN
p_return_message := null;
cur_field := RTRIM(LTRIM(ffio.get_current_item_name));
cur_value := ffio.getc;
last_word := SUBSTR(cur_field,INSTR(cur_field,' ', -1, 1)+1, 80 );

IF ( last_word IN ('NORTH','SOUTH','EAST','WEST' )) THEN
  rec := ffio.get_next_record_by_item ( '111','PL',SUBSTR(last_word,1,1) );
  rec_msg := '111';
ELSIF ( last_word = 'RANGE' ) THEN
  rec := ffio.get_next_record_by_item ( '041' );
  rec_msg := '041';
ELSIF ( last_word = 'CWD' ) THEN
  rec := ffio.get_next_record_by_item ( '061' );
  rec_msg := '061';
ELSIF ( last_word = 'ECOLOGICAL' ) THEN
  rec := ffio.get_next_record_by_item ( '121' );
  rec_msg := '121';
ELSIF ( last_word = 'INTEGRATED' ) THEN
  rec := ffio.get_next_record_by_item ( '022',
    'LOCATION TYPE', 'INTEGRATED PLOT CENTRE');
  IF ( cur_value = 'Y' AND NVL(rec,0) <= 0 ) THEN
    p_return_code := 4;
    p_return_message := 'The Integrated Plot Centre ' ||
      'information on the VEG Compass Card *"'022' ||
      '"*** must be present, if ' || cur_field ||
      ' is checked as "' || cur_value || '".';
    RETURN;
  END IF;
  rec := ffio.get_next_record_by_item ( '041' );
  rec_msg := '041';
  IF ( rec <= 0 ) THEN
    rec := ffio.get_next_record_by_item ( '061' );
    rec_msg := '061';
  END IF;
  IF ( rec <= 0 ) THEN
    rec := ffio.get_next_record_by_item ( '081' );
    rec_msg := '081';
  END IF;
  FOR i IN 12..16 LOOP
    IF ( rec <= 0 ) THEN
      rec := ffio.get_next_record_by_item ( TO_CHAR(i)||'1');
      rec_msg := TO_CHAR(i)||'1';
    END IF;
  END LOOP;
  IF ( cur_value = 'Y' AND rec <= 0 ) THEN
    rec_msg := 'any corresponding';
  
```

Record: VEG CH 01 1**Field:** PLOT CLUSTER NORTH**Datatype:** CHAR**Position:** 70**Optional:** N**Length:** 1**Zero to Null:****Domain Name:** INDICATOR**Value**

N

Y

Meaning

No (False)

Yes (True)

Rule Name: Veg 2000 Validate Plot Cluster Data Collected Indicators**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG CH 01 1**Field:** PLOT CLUSTER EAST**Datatype:** CHAR**Position:** 71**Optional:** N**Length:** 1**Zero to Null:****Domain Name:** INDICATOR**Value**

N

Y

Meaning

No (False)

Yes (True)

Rule Name: Veg 2000 Validate Plot Cluster Data Collected Indicators**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG CH 01 1**Field:** PLOT CLUSTER SOUTH**Datatype:** CHAR**Position:** 72**Optional:** N**Length:** 1**Zero to Null:****Domain Name:** INDICATOR**Value**

N

Y

Meaning

No (False)

Yes (True)

Rule Name: Veg 2000 Validate Plot Cluster Data Collected Indicators**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG CH 01 1**Field:** PLOT CLUSTER WEST**Datatype:** CHAR**Position:** 73**Optional:** N**Length:** 1**Zero to Null:****Domain Name:** INDICATOR**Value**

N

Y

Meaning

No (False)

Yes (True)

Rule Name: Veg 2000 Validate Plot Cluster Data Collected Indicators**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG CH 01 1**Field:** PLOT CLUSTER RANGE**Datatype:** CHAR**Position:** 74**Optional:** N

	INDICATOR	Length:	1	Zero to Null:
Domain Name:	<u>Value</u>	Meaning		
	N	No (False)		
	Y	Yes (True)		

Rule Name: Veg 2000 Validate Plot Cluster Data Collected Indicators

PL/SQL Source Code: Rule code has already been printed.

Record: <u>VEG CH 01 1</u>	Field: <u>PLOT CLUSTER ECOLOGICAL</u>	
	Datatype: CHAR	
	Position: 75	Optional: N
	Length: 1	Zero to Null:

Domain Name: INDICATOR

<u>Value</u>	<u>Meaning</u>
N	No (False)
Y	Yes (True)

Rule Name: Veg 2000 Validate Plot Cluster Data Collected Indicators

PL/SQL Source Code: Rule code has already been printed.

Record: <u>VEG CH 01 1</u>	Field: <u>PLOT CLUSTER CWD</u>	
	Datatype: CHAR	
	Position: 76	Optional: N
	Length: 1	Zero to Null:

Domain Name: INDICATOR

<u>Value</u>	<u>Meaning</u>
N	No (False)
Y	Yes (True)

Rule Name: Veg 2000 Validate Plot Cluster Data Collected Indicators

PL/SQL Source Code: Rule code has already been printed.

Record: <u>VEG CH 01 1</u>	Field: <u>PLOT CLUSTER TREES</u>	
	Datatype: CHAR	
	Position: 77	Optional: N
	Length: 1	Zero to Null:

Domain Name: VEG SAMPLE INDICATOR WITH '-'

<u>Value</u>	<u>Meaning</u>
-	Observed but Not Applicable or Undetermined
N	No
Y	Yes

Rule Name: Veg 2000 CH 01 1 Plot Cluster Typical Trees or Ecology Indicato

PL/SQL Source Code:

--

Record: VEG CH 01 1**Field:** PLOT CLUSTER ECOLOGY**Datatype:** CHAR**Position:** 78**Optional:** N**Length:** 1**Zero to Null:****Domain Name:** VEG SAMPLE INDICATOR WITH '-'**Value****Meaning**

-

Observed but Not Applicable or Undetermined

N

No

Y

Yes

Rule Name: Veg 2000 CH 01 1 Plot Cluster Typical Trees or Ecology Indicato**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG CH 01 1**Field:** PLOT CLUSTER # OF PHOTOS**Datatype:** NUMBER**Position:** 79**Optional:** Y**Length:** 2**Zero to Null:****Record:** VEG CH 01 1**Field:** S1**Datatype:** CHAR**Position:** 81**Optional:** Y**Length:** 1**Zero to Null:****Record:** VEG CH 01 1**Field:** S2**Datatype:** CHAR**Position:** 82**Optional:** Y**Length:** 1**Zero to Null:****Record:** VEG CH 01 1**Field:** S3**Datatype:** CHAR**Position:** 83**Optional:** Y**Length:** 1**Zero to Null:****Record:** VEG CH 01 1**Field:** S4**Datatype:** CHAR**Position:** 84**Optional:** Y**Length:** 1**Zero to Null:****Record:** VEG CH 01 1**Field:** S5**Datatype:** CHAR**Position:** 85**Optional:** Y**Length:** 1**Zero to Null:****Record:** VEG CH 01 1**Field:** S6**Datatype:** CHAR**Position:** 86**Optional:** Y**Length:** 1**Zero to Null:**

Record: <u>VEG CH 01 1</u>	Field: <u>S7</u> Datatype: CHAR Position: 87 Length: 1 Optional: Y Zero to Null:
Record: <u>VEG CH 01 1</u>	Field: <u>S8</u> Datatype: CHAR Position: 88 Length: 1 Optional: Y Zero to Null:
Record: <u>VEG CH 01 1</u>	Field: <u>S9</u> Datatype: CHAR Position: 89 Length: 1 Optional: Y Zero to Null:
Record: <u>VEG CH 01 1</u>	Field: <u>S10</u> Datatype: CHAR Position: 90 Length: 1 Optional: Y Zero to Null:
Record: <u>VEG CH 01 2</u>	Field: <u>RECORD ID</u> Datatype: NUMBER Position: 1 Length: 3 Optional: N Zero to Null:
Record: <u>VEG CH 01 2</u>	Field: <u>PAGE NUMBER</u> Datatype: NUMBER Position: 4 Length: 2 Optional: N Zero to Null:
Record: <u>VEG CH 01 2</u>	Field: <u>SEQUENCE NUMBER</u> Datatype: NUMBER Position: 6 Length: 4 Optional: N Zero to Null:
Record: <u>VEG CH 01 2</u>	Field: <u>CH CARD REC TYPE</u> Datatype: CHAR Position: 10 Length: 10 Optional: N Zero to Null:
Domain Name: VEG CH RECORD 2 TYPE	
Value	Meaning
ECOLOGICAL	
TREE DATA	
Record: <u>VEG CH 01 2</u>	Field: <u>FIELD RESPONSIBILITY</u> Datatype: CHAR Position: 20 Optional: N

Length:

30

Zero to Null:**Rule Name:** Veg 2000 Check the Person's Initials**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG CH 01 2**Field:** FIELD CHECK BY**Datatype:** CHAR**Position:** 50**Length:** 30**Optional:** Y**Zero to Null:****Rule Name:** Veg 2000 Validate Presence of Name and Date**PL/SQL Source Code:**

```

name VARCHAR2(256);
c_date VARCHAR2(20);
cur_item VARCHAR2(100);
BEGIN
cur_item := ffio.get_current_item_name;
name := ffio.getc;
c_date := ffio.getc(REPLACE(cur_item, 'BY', 'DATE'));

IF ( name || c_date IS NULL OR
    ( name IS NOT NULL AND c_date IS NOT NULL ) ) THEN
    p_return_code := 0;
ELSE
    IF ( name is NULL ) THEN
        name := 'missing';
    ELSIF ( c_date is NULL ) THEN
        c_date := 'missing';
    END IF;
    p_return_code := 1;
    p_return_message := ' For the field ' || cur_item || ', the Name is ' ||
        name || ' and the Date is ' || c_date || '. **Both** must be ' ||
        'recorded or not recorded, only one present is not exceptable.';
END IF;

```

Rule Name: Veg 2000 Check the Person's Initials**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG CH 01 2**Field:** FIELD CHECK DATE**Datatype:** DATE**Position:** 80**Length:** 9**Date Mask:** YYYYMONDD**Optional:** Y**Zero to Null:****Record:** VEG CH 01 2**Field:** OFFICE CHECK BY**Datatype:** CHAR**Position:**

Length: 89
30
Optional: Y
Zero to Null:

Rule Name: Veg 2000 Validate Presence of Name and Date

PL/SQL Source Code: Rule code has already been printed.

Rule Name: Veg 2000 Check the Person's Initials

PL/SQL Source Code: Rule code has already been printed.

Record: VEG CH 01 2

Field: OFFICE CHECK DATE

Datatype: DATE

Date Mask: YYYYMONDD

Position: 119

Optional: Y

Length: 9

Zero to Null:

Record: VEG CH 01 3

Field: RECORD ID

Datatype: NUMBER

Position: 1

Optional: N

Length: 3

Zero to Null:

Record: VEG CH 01 3

Field: PAGE NUMBER

Datatype: NUMBER

Position: 4

Optional: N

Length: 2

Zero to Null:

Record: VEG CH 01 3

Field: SEQUENCE NUMBER

Datatype: NUMBER

Position: 6

Optional: N

Length: 4

Zero to Null:

Record: VEG CH 01 3

Field: CH GROUND PHOTO TYPE

Datatype: CHAR

Position: 10

Optional: N

Length: 14

Zero to Null:

Domain Name: VEG CH RECORD 3 TYPE

Value

Meaning

ADDITIONAL

PLOT PIN

RANDOM AZIMI

Random Azimuth

REPRESENTAT:

Record: VEG CH 01 3

Field: PHOTO INDICATOR

Datatype: CHAR

Position: 24

Optional: Y

Length: 1

Zero to Null:

Domain Name: INDICATOR

Value

Meaning

N

No (False)

Y	Yes (True)
Record: <u>VEG CH 01 3</u>	Field: <u>ADDITIONAL NOTES</u>
	Datatype: CHAR
	Position: 25
	Length: 256
	Optional: Y
	Zero to Null:
Record: <u>VEG CH 01 4</u>	Field: <u>RECORD ID</u>
	Datatype: NUMBER
	Position: 1
	Length: 3
	Optional: N
	Zero to Null:
Record: <u>VEG CH 01 4</u>	Field: <u>PAGE NUMBER</u>
	Datatype: NUMBER
	Position: 4
	Length: 2
	Optional: N
	Zero to Null:
Record: <u>VEG CH 01 4</u>	Field: <u>SEQUENCE NUMBER</u>
	Datatype: NUMBER
	Position: 6
	Length: 4
	Optional: N
	Zero to Null:
Record: <u>VEG CH 01 4</u>	Field: <u>NOTES</u>
	Datatype: CHAR
	Position: 10
	Length: 256
	Optional:
	Zero to Null:
Record: <u>VEG CH 01 5</u>	Field: <u>RECORD ID</u>
	Datatype: NUMBER
	Position: 1
	Length: 3
	Optional: N
	Zero to Null:
Record: <u>VEG CH 01 5</u>	Field: <u>PAGE NUMBER</u>
	Datatype: NUMBER
	Position: 4
	Length: 2
	Optional: N
	Zero to Null:
Record: <u>VEG CH 01 5</u>	Field: <u>SEQUENCE NUMBER</u>
	Datatype: NUMBER
	Position: 6
	Length: 4
	Optional: N
	Zero to Null:
Record: <u>VEG CH 01 5</u>	Field: <u>NOTES</u>
	Datatype: CHAR
	Position: 10
	Optional: N

	Length:	256	Zero to Null:
Record: <u>VEG CP 02 1</u>	Field: <u>RECORD ID</u>		
	Datatype: NUMBER		
	Position: 1		Optional: N
	Length: 3		Zero to Null:

Rule Name: Veg 2000 CP 02 2 Tie Points for Reference Pin Detail Records mus
PL/SQL Source Code:

```

FifteenmRecNo NUMBER;
TieRecNo NUMBER;
FifteenmPresent BOOLEAN := FALSE;
TiePresent BOOLEAN := FALSE;
TempMsg VARCHAR2(1000);
BEGIN
  p_return_code := 0;

  FifteenmRecNo := ffio.get_next_record_by_item('023', 'POINT TREE TYPE', '15M
POINT DETAIL');
  IF FifteenmRecNo > 0 THEN
    IF ffio.getc(FifteenmRecNo, 'TREE SPECIES') IS NOT NULL
      AND ffio.getc(FifteenmRecNo, 'DIAMETER') IS NOT NULL
      AND ffio.getc(FifteenmRecNo, 'AZIMUTH TREE TO DETAIL TYPE') IS NOT NULL
      AND ffio.getc(FifteenmRecNo, 'DISTANCE') IS NOT NULL THEN
      FifteenmPresent := TRUE;
    END IF;
  END IF;

  TieRecNo := ffio.get_next_record_by_item('023', 'POINT TREE TYPE', 'TIE POINT
DETAIL');
  IF TieRecNo > 0 THEN
    IF ffio.getc(TieRecNo, 'TREE SPECIES') IS NOT NULL
      AND ffio.getc(TieRecNo, 'DIAMETER') IS NOT NULL
      AND ffio.getc(TieRecNo, 'AZIMUTH TREE TO DETAIL TYPE') IS NOT NULL
      AND ffio.getc(TieRecNo, 'DISTANCE') IS NOT NULL THEN
      TiePresent := TRUE;
    END IF;
  END IF;

  IF NOT TiePresent OR NOT FifteenmPresent THEN
    TempMsg := 'On card CP 02, ';

    IF NOT TiePresent THEN
      TempMsg := 'the Tie Point Detail record is missing or incomplete';
    END IF;

    IF NOT FifteenmPresent THEN
      IF NOT TiePresent THEN
        TempMsg := TempMsg || ' and ';
      END IF;
      TempMsg := TempMsg || 'the 15m Point Detail record is missing or

```

Record: VEG CP 02 1**Field:** PAGE NUMBER**Datatype:** NUMBER**Position:** 4**Optional:** N**Length:** 2**Zero to Null:****Rule Name:** Veg 2000 Validate Page Number**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG CP 02 1**Field:** SEQUENCE NUMBER**Datatype:** NUMBER**Position:** 6**Optional:** N**Length:** 4**Zero to Null:****Record:** VEG CP 02 1**Field:** G.P.S TIE POINT FILE I.D.**Datatype:** CHAR**Position:** 10**Optional:** Y**Length:** 8**Zero to Null:****Record:** VEG CP 02 1**Field:** G.P.S TIE POINT UTM ZONE**Datatype:** NUMBER**Position:** 18**Optional:** Y**Length:** 2**Zero to Null:****Domain Name:** UTM ZONE ALL BCValue

7 - 12

Meaning**Record:** VEG CP 02 1**Field:** G.P.S TIE POINT NORTHING**Datatype:** NUMBER**Position:** 20**Optional:** Y**Length:** 7**Zero to Null:****Domain Name:** UTM NORTHINGValue

0 - 9999999.999

Meaning**Record:** VEG CP 02 1**Field:** G.P.S TIE POINT EASTING**Datatype:** NUMBER**Position:** 27**Optional:** Y**Length:** 6**Zero to Null:****Domain Name:** UTM EASTINGValue

0 - 999999.999

Meaning**Record:** VEG CP 02 1**Field:** G.P.S TIE POINT ELEVATION**Datatype:** NUMBER**Position:** 33**Optional:** Y**Length:** 4**Zero to Null:****Domain Name:** ELEVATIONValue

0 - 4000

Meaning

Record: VEG CP 02 1**Field:** G.P.S ACCESS POINT DESCR**Datatype:** CHAR**Position:** 37**Optional:** Y**Length:** 20**Zero to Null:****Record:** VEG CP 02 1**Field:** G.P.S ACCESS POINT FILE I.D**Datatype:** CHAR**Position:** 57**Optional:** Y**Length:** 8**Zero to Null:****Record:** VEG CP 02 1**Field:** G.P.S ACCESS POINT NORTHING**Datatype:** NUMBER**Position:** 65**Optional:** Y**Length:** 7**Zero to Null:****Domain Name:** UTM NORTHING**Value**

0 - 9999999.999

Meaning**Record:** VEG CP 02 1**Field:** G.P.S ACCESS POINT EASTING**Datatype:** NUMBER**Position:** 72**Optional:** Y**Length:** 6**Zero to Null:****Domain Name:** UTM EASTING**Value**

0 - 999999.999

Meaning**Record:** VEG CP 02 1**Field:** G.P.S ACCESS POINT ELEVATION**Datatype:** NUMBER**Position:** 78**Optional:** Y**Length:** 4**Zero to Null:****Domain Name:** ELEVATION**Value**

0 - 4000

Meaning**Record:** VEG CP 02 1**Field:** STRAIGHT LINE AZIMUTH TO I.F**Datatype:** NUMBER**Position:** 82**Optional:** N**Length:** 3**Zero to Null:****Domain Name:** AZIMUTH BRG 0 TO 359**Value**

0 - 359

Meaning**Record:** VEG CP 02 1**Field:** STRAIGHT LINE DISTANCE TO I.**Datatype:** NUMBER**Position:** 85**Optional:** N**Length:** 4**Zero to Null:****Record:** VEG CP 02 1**Field:** STRAIGHT LINE DECLINATION TO I**Datatype:** NUMBER**Position:** 89**Optional:** N

Record: <u>VEG CP 02 1</u>	Length: 3 Zero to Null:
Domain Name: AZIMUTH BRG 0 TO 359 Value 0 - 359	Field: <u>OFFSET AZIMUTH</u> Datatype: NUMBER Position: 92 Length: 3 Optional: Y Zero to Null:
Record: <u>VEG CP 02 1</u>	Field: <u>OFFSET DISTANCE</u> Datatype: NUMBER Position: 95 Length: 4 Optional: Y Zero to Null:
Record: <u>VEG CP 02 1</u>	Field: <u>TIE POINT TREE TAG NUMBER</u> Datatype: NUMBER Position: 99 Length: 6 Optional: N Zero to Null:
Record: <u>VEG CP 02 1</u>	Field: <u>OFFSET G.P.S. LOCATION AZIMUTH</u> Datatype: NUMBER Position: 105 Length: 3 Optional: Y Zero to Null: N
Domain Name: AZIMUTH BRG 0 TO 359 Value 0 - 359	Meaning Field: <u>OFFSET G.P.S. LOCATION DISTANCE</u> Datatype: NUMBER Position: 108 Length: 4 Optional: Y Zero to Null: N
Record: <u>VEG CP 02 1</u>	Field: <u>TOTAL PAGES</u> Datatype: NUMBER Position: 112 Length: 2 Optional: Y Zero to Null:
Record: <u>VEG CP 02 2</u>	Field: <u>RECORD ID</u> Datatype: NUMBER Position: 1 Length: 3 Optional: N Zero to Null:
Record: <u>VEG CP 02 2</u>	Field: <u>PAGE NUMBER</u> Datatype: NUMBER Position: 4 Optional: N

Record: VEG CP 02 2

Length: 2 **Zero to Null:**

Field: SEQUENCE NUMBER

Datatype: NUMBER

Position: 6 **Optional:** N

Length: 4 **Zero to Null:**

Record: VEG CP 02 2

Field: LOCATION TYPE

Datatype: CHAR

Position: 10 **Optional:** N

Length: 22 **Zero to Null:**

Domain Name: VEG CP RECORD 2 TYPE

Value

INTEGRATED I
TIE POINT

Meaning

Rule Name: Veg 2000 CP 02 2 Convert Location Type to Location Type Code

PL/SQL Source Code:

```
cur_type VARCHAR2(30);
BEGIN

p_return_code := 0;
IF ( ffio.getc('RECORD ID') = '022' ) THEN
  cur_type := ffio.getc('LOCATION TYPE');
ELSIF ( ffio.getc('RECORD ID') = '021' ) THEN
  cur_type := 'ACCESS';
ELSE
  cur_type := 'REFERENCE';
END IF;

IF ( cur_type = 'TIE POINT' ) THEN
  p_return_value := 'TIE POINT';
ELSIF ( cur_type = 'INTEGRATED PLOT CENTRE' ) THEN
  p_return_value := 'MAIN PLOT CENTRE';
ELSIF ( cur_type = 'ACCESS' ) THEN
  p_return_value := 'ACCESS POINT';
ELSIF ( cur_type = 'REFERENCE' ) THEN
  p_return_value := 'REFERENCE POINT';
ELSE
  p_return_code := 1;
END IF;
```

Rule Name: Veg 2000 CP 02 2 Convert Location Type to Point Type

PL/SQL Source Code:

```
cur_type VARCHAR2(30);
BEGIN
p_return_code := 0;
```

Record: VEG CP 02 2**Field:** MAP NO.**Datatype:** CHAR**Position:** 32**Length:** 7**Optional:** N**Zero to Null:****Rule Name:** Veg 2000 CP 02 2 Look Up BCGS Map Number**PL/SQL Source Code:**

```

l_map VARCHAR2(20);
l_chr VARCHAR2(1);
loc VARCHAR2(44);
    num_blk VARCHAR2(3);
    ltr_blk VARCHAR2(1);
    bcgs_6x12 VARCHAR2(3);
map_skey NUMBER;
BEGIN
p_return_code := 0;
map_skey := NULL;
l_map := ffio.getc;
IF (l_map IS NULL) THEN
    RETURN;
END IF;
    -- accomidate for the possibility that the first number in the map id
    -- is missing the preceeding zeros eg '82E045' is supplied for '082E045'
l_chr := SUBSTR(l_map,2,1);
IF (TRANSLATE(l_chr,'-0123456789','-') IS NOT NULL) THEN
    l_map := '00' || l_map;
ELSE
    l_chr := SUBSTR(l_map,3,1);
    IF (TRANSLATE(l_chr,'-0123456789','-') IS NOT NULL) THEN
        l_map := '0' || l_map;
    END IF;
END IF;
    num_blk := SUBSTR(l_map,1,3);
    ltr_blk := SUBSTR(l_map,4,1);
bcgs_6x12 := SUBSTR(l_map,5,3);
map_skey := coms.get_recursive_mapsheet_id (num_blk, ltr_blk, bcgs_6x12,
FALSE);
    IF map_skey IS NOT NULL THEN
p_return_value := map_skey;
p_return_code := 0;
    ELSE
p_return_code := 1;
loc := ffio.getc( 'LOCATION TYPE' );
p_return_message := 'The map number ***' || ffio.getc || '*** in the input
field ' ||
    ffio.get_current_item_name || ' for location ' || loc ||
    ' is invalid when compared to the BCGS' || ' Map lookup table. ';
p_return_value := NULL;
    END IF;

```

Record: VEG CP 02 2**Field:** TEMP. POLYGON NO.**Datatype:** NUMBER**Position:** 39**Length:** 5**Optional:** N**Zero to Null:**

Record: VEG CP 02 2**Field:** FLIGHT LINE**Datatype:** CHAR**Position:** 44**Length:** 11**Optional:** N**Zero to Null:****Record:** VEG CP 02 2**Field:** PHOTO NO.**Datatype:** CHAR**Position:** 55**Length:** 3**Optional:** N**Zero to Null:****Record:** VEG CP 02 3**Field:** RECORD ID**Datatype:** NUMBER**Position:** 1**Length:** 3**Optional:** N**Zero to Null:****Record:** VEG CP 02 3**Field:** PAGE NUMBER**Datatype:** NUMBER**Position:** 4**Length:** 2**Optional:** N**Zero to Null:****Record:** VEG CP 02 3**Field:** SEQUENCE NUMBER**Datatype:** NUMBER**Position:** 6**Length:** 4**Optional:** N**Zero to Null:****Record:** VEG CP 02 3**Field:** POINT TREE TYPE**Datatype:** CHAR**Position:** 10**Length:** 16**Optional:** N**Zero to Null:****Domain Name:** VEG CP RECORD 3 TYPE**Value**

15M POINT DI

TIE POINT DI

Meaning

15m Reference Point Tree Details

Time Point Tree Details

Rule Name: Veg 2000 Generate Point Type for Reference Pin**PL/SQL Source Code:**

BEGIN

p_return_code := 0;

IF (ffio.getc = '15M POINT DETAIL') THEN

p_return_value := 'PLOT REF';

ELSIF (ffio.getc = 'TIE POINT DETAIL') THEN

p_return_value := NULL;

ELSE

p_return_code := 1;

p_return_message := 'The POINT TREE TYPE must be either "15M POINT DETAIL"

Record: VEG CP 02 3**Field:** TREE SPECIES**Datatype:** CHAR**Position:** 26**Length:** 3**Optional:** Y**Zero to Null:****Rule Name:** Veg 2000 Tree Species Short Code**PL/SQL Source Code:**

```
--
-----
--
-- Veg Tree Species Short Code
--
-- try to find a record in the group_taxon_memberships table that is associated
-- with a species group definition group code of 'tree' and with a taxonomic
-- name that has a short code equal to the source data tree species code, that
-- is, try to find a valid tree species codes record for the source data tree
-- species code
--
-- if such a record cannot be found, try to find a record in the group_taxon_
-- memberships table that are associated with a species group definition group
-- code of 'tree', a taxonomic name that has a short code that is null, and a
-- group_taxon_memberships short code equal to the source data tree species
-- code, that is, try to find a valid unknown tree species codes record for the
-- source data tree species code
--
-- if successful in either search, return the skkey of the record found in the
-- group_taxon_memberships table
--
--
-- Modifications
--
-- Dec 17, 1998 TMC - wo 632, remove two criteria from the select statements,
-- ( sgdef_spp_group_skey = gto_sgdef_spp_group_skey ) and
-- ( taxc_taxonomic_cd_skey = gto_taxc_taxonomic_cd_skey );
-- these criteria are already in the view itself, and the
-- second criteria needed an outer join on the
-- taxc_taxonomic_cd_skey column anyway; the outer join is
-- now in the view, so this criteria doesn't need to be
-- changed, it just needs to be removed; reworded the error
-- messages
--
-- Nov 5, 1999 CAW - wo 687. Changing the rule to return NULL when the record is
-- the second 082 or 114 with the same tree number (i.e. when
-- there are > 4 logs ).
--
-- Nov 9, 1999 TMC - wo 746, a willow tree is always allowed for some cards
-- regardless of measurement date or the species'
-- effectiveness dates
--
-- Jul 25, 2000 LME - added species group cd 'NON-HERB MIX' to the species
-- code lookups
--
-- July 9,2001 JPK wo811; put indicator for plot type in error messages
--
```

Record: VEG CP 02 3**Field:** DIAMETER**Datatype:** NUMBER**Position:** 29**Length:** 4**Optional:** Y**Zero to Null:****Domain Name:** TREE DIAMETER 4 1**Value**

000.1 - 999.9

Meaning**Record:** VEG CP 02 3**Field:** AZIMUTH TREE TO DETAIL TYPE**Datatype:** NUMBER**Position:** 33**Length:** 3**Optional:** Y**Zero to Null:****Domain Name:** AZIMUTH BRG 0 TO 359**Value**

0 - 359

Meaning**Record:** VEG CP 02 3**Field:** DISTANCE**Datatype:** NUMBER**Position:** 36**Length:** 4**Optional:** Y**Zero to Null:****Record:** VEG CP 02 4**Field:** RECORD ID**Datatype:** NUMBER**Position:** 1**Length:** 3**Optional:** N**Zero to Null:****Record:** VEG CP 02 4**Field:** PAGE NUMBER**Datatype:** NUMBER**Position:** 4**Length:** 2**Optional:** N**Zero to Null:****Record:** VEG CP 02 4**Field:** SEQUENCE NUMBER**Datatype:** NUMBER**Position:** 6**Length:** 4**Optional:** N**Zero to Null:****Record:** VEG CP 02 4**Field:** NOTES**Datatype:** CHAR**Position:** 10**Length:** 256**Optional:** Y**Zero to Null:****Record:** VEG CL 03 1**Field:** RECORD ID**Datatype:** NUMBER**Position:** 1**Length:** 3**Optional:** N**Zero to Null:****Record:** VEG CL 03 1**Field:** PAGE NUMBER**Datatype:** NUMBER

Position: 4
Length: 2
Optional: Y
Zero to Null:

Rule Name: Veg 2000 Validate Page Number

PL/SQL Source Code: Rule code has already been printed.

Record: VEG CL 03 1 **Field:** SEQUENCE NUMBER
Datatype: NUMBER
Position: 6 **Optional:** N
Length: 4 **Zero to Null:**

Record: VEG CL 03 1 **Field:** G.P.S. I.P.C. FILE I.D.
Datatype: CHAR
Position: 10 **Optional:** Y
Length: 8 **Zero to Null:**

Record: VEG CL 03 1 **Field:** G.P.S. I.P.C. UTM ZONE
Datatype: NUMBER
Position: 18 **Optional:** Y
Length: 2 **Zero to Null:**

Domain Name: UTM ZONE
Value
7 - 11

Meaning

Record: VEG CL 03 1 **Field:** G.P.S. I.P.C. NORTHING
Datatype: NUMBER
Position: 20 **Optional:** Y
Length: 7 **Zero to Null:**

Domain Name: UTM NORTHING
Value
0 - 9999999.999

Meaning

Record: VEG CL 03 1 **Field:** G.P.S. I.P.C. EASTING
Datatype: NUMBER
Position: 27 **Optional:** Y
Length: 6 **Zero to Null:**

Domain Name: UTM EASTING
Value
0 - 999999.999

Meaning

Record: VEG CL 03 1 **Field:** G.P.S. I.P.C. ELEVATION
Datatype: NUMBER
Position: 33 **Optional:** Y
Length: 4 **Zero to Null:**

Domain Name: ELEVATION
Value
0 - 4000

Meaning

Record: VEG CL 03 1**Field:** OFFSET AZIMUTH**Datatype:** NUMBER**Position:** 37**Optional:** Y**Length:** 3**Zero to Null:****Domain Name:** AZIMUTH BRG 0 TO 359**Value**

0 - 359

Meaning**Record:** VEG CL 03 1**Field:** OFFSET DISTANCE**Datatype:** NUMBER**Position:** 40**Optional:** Y**Length:** 4**Zero to Null:****Rule Name:** Veg 2000 Validate Field as Required**PL/SQL Source Code:**

BEGIN

p_return_code := 0;

IF ffio.getc IS NULL THEN

p_return_code := 1;

p_return_message := 'Field ' || ffio.get_current_item_name ||
' is missing';

END IF;

Record: VEG CL 03 1**Field:** OFFSET G.P.S. LOCATION AZIMUTH**Datatype:** NUMBER**Position:** 44**Optional:** Y**Length:** 3**Zero to Null:****Domain Name:** AZIMUTH BRG 0 TO 359**Value**

0 - 359

Meaning**Record:** VEG CL 03 1**Field:** OFFSET G.P.S. LOCATION DISTANCE**Datatype:** NUMBER**Position:** 47**Optional:** Y**Length:** 4**Zero to Null:****Record:** VEG CL 03 1**Field:** AUXILIARY PLOT SPACING**Datatype:** NUMBER**Position:** 51**Optional:** N**Length:** 4**Zero to Null:****Record:** VEG CL 03 1**Field:** TOTAL PAGES**Datatype:** NUMBER**Position:** 55**Optional:** Y**Length:** 2**Zero to Null:****Record:** VEG CL 03 2**Field:** RECORD ID

	Datatype: NUMBER	
	Position: 1	Optional: N
	Length: 3	Zero to Null:
Record: <u>VEG CL 03 2</u>	Field: <u>PAGE NUMBER</u>	
	Datatype: NUMBER	
	Position: 4	Optional: N
	Length: 2	Zero to Null:
Record: <u>VEG CL 03 2</u>	Field: <u>SEQUENCE NUMBER</u>	
	Datatype: NUMBER	
	Position: 6	Optional: N
	Length: 4	Zero to Null:
Record: <u>VEG CL 03 2</u>	Field: <u>NOTES</u>	
	Datatype: CHAR	
	Position: 10	Optional: Y
	Length: 256	Zero to Null:
Record: <u>VEG RS 04 1</u>	Field: <u>RECORD ID</u>	
	Datatype: NUMBER	
	Position: 1	Optional: N
	Length: 3	Zero to Null:
Record: <u>VEG RS 04 1</u>	Field: <u>PAGE NUMBER</u>	
	Datatype: NUMBER	
	Position: 4	Optional: Y
	Length: 2	Zero to Null:
Record: <u>VEG RS 04 1</u>	Field: <u>SEQUENCE NUMBER</u>	
	Datatype: NUMBER	
	Position: 6	Optional: N
	Length: 4	Zero to Null:
Record: <u>VEG RS 04 1</u>	Field: <u>PL</u>	
	Datatype: CHAR	
	Position: 10	Optional: N
	Length: 1	Zero to Null:
Domain Name: VEG SAMPLE PL - PLOT NO		
Value	Meaning	
E		
I		
M		
N		
S		

W

Record: VEG RS 04 1**Field:** MEASUREMENT DATE**Datatype:** DATE**Date Mask:** YYYYMONDD**Position:** 11**Optional:** Y**Length:** 9**Zero to Null:****Domain Name:** DATE MIN JAN 01 1997 - FULL**Value****Meaning**

1997JAN01 - 3999JAN01 minmum date for 1999 Ground Sampling Format - YYM

Rule Name: Veg 2000 RS 04 1 EW 06 1 Validate Required Field for Transect 1**PL/SQL Source Code:**

```

trans    NUMBER;
cur_field VARCHAR2(60);

BEGIN
cur_field := ffio.get_current_item_name;
BEGIN
trans := ffio.getn('TRANSECT#');
EXCEPTION WHEN VALUE_ERROR THEN
p_return_code := 2;
p_return_message := ' The value for Transect # ****' ||
ffio.getc('TRANSECT#') || '**** is non numeric';
RETURN;
END;

IF ( trans = 1 AND ffio.getc IS NULL ) THEN
p_return_code := 1;
p_return_message := ' The field ****' || cur_field ||
'**** is required for TRANSECT #1 and ' ||
'is found to be missing..';
ELSE
p_return_code := 0;
END IF;

```

Record: VEG RS 04 1**Field:** CREW PERSON #1**Datatype:** CHAR**Position:** 20**Optional:** Y**Length:** 4**Zero to Null:****Rule Name:** Veg 2000 Validate Person Against First Crew List**PL/SQL Source Code:**

```

rec    NUMBER;
crew   VARCHAR2(10);
BEGIN
p_return_code := 0;
crew := ffio.getc;

IF ( crew IS NOT NULL ) THEN

```

Rule Name: Veg 2000 Check the Person's Initials

PL/SQL Source Code: Rule code has already been printed.

Record: VEG RS 04 1

Field: CREW PERSON #2

Datatype: CHAR

Position: 24

Length: 4

Optional: Y

Zero to Null:

Rule Name: Veg 2000 Validate Person Against First Crew List

PL/SQL Source Code: Rule code has already been printed.

Rule Name: Veg 2000 Check the Person's Initials

PL/SQL Source Code: Rule code has already been printed.

Record: VEG RS 04 1

Field: TRANSECT#

Datatype: NUMBER

Position: 28

Length: 1

Optional: N

Zero to Null:

Domain Name: TRANSECT NUMBER

Value

1

- 2

Meaning

Rule Name: Veg 2000 RS 04 1 Look Up Shrub Transect Spatial Delimiter and Cl

PL/SQL Source Code:

```
--
-----
--
-- Veg Look Up Shrub Transect Spatial Delimiter and Cluster Element
-- Specification
--
-- Modifications
--
-- Jan 8, 1999 TMC - wo 686, sometime prior to Nov 12, 1998 the
-- spatial_delimiters table was changed without any
-- communication to the DRM group, the change modified the
-- record with a sptld_type of LIN and a total_line_len of
-- 24.0: the sptld_type is now TRA, the length is now 24.0,
-- and the total_line_length is now null; changed the select
-- statement dealing with the spatial_delimiter table to
-- look for a sptld_type of TRA and to match the length with
-- the source data line length total
--
-----
--
spop NUMBER;
VarSptldSkeyForLineLenTot NUMBER;
VarSrceDataLineLenTot NUMBER;
BEGIN
p_return_code := 0;
VarSptldSkeyForLineLenTot := NULL;
VarSrceDataLineLenTot := ffio.getn ( 'LINE LENGTH TOTAL', -1 );
```

Record: VEG RS 04 1**Field:** TOTAL PAGES**Datatype:** NUMBER**Position:** 29**Optional:** Y**Length:** 2**Zero to Null:****Record:** VEG RS 04 1**Field:** TRANSECT AZIMUTH**Datatype:** NUMBER**Position:** 31**Optional:** N**Length:** 3**Zero to Null:****Domain Name:** AZIMUTH BRG 0 TO 359Value

0 - 359

Meaning**Rule Name:** Veg 2000 RS 04 1 EW 06 1 Validate Required Field for Transect 1**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG RS 04 1**Field:** LINE LENGTH OBSERVED**Datatype:** NUMBER**Position:** 34**Optional:** N**Length:** 3**Zero to Null:****Domain Name:** TRANSECT LENGTHValue

0 - 24.0

Meaning**Rule Name:** Veg 2000 RS 04 1 EW 06 1 Validate Required Field for Transect 1**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 RS 04 1 EW 06 1 Observed Transect Line Length**PL/SQL Source Code:**

```
--
-----
--
-- Veg RS 04 1 EW 06 1 Observed Transect Line Length
--
-- Modifications
--
-- Jan 22, 1999 TMC - wo 690, an observed line length of 0 is valid for any
--                    value of the total line length (recognizing the 0 of 0
--                    restrictions which are handled by bus rule 11,228)
--
-----
--
```

```
VarSrceDataLineLenObs NUMBER;
VarSrceDataLineLenTot NUMBER;
VarSrceDataRecId VARCHAR2(3);
VarSrceDataTransectNo VARCHAR2(1);
```

BEGIN

Record: VEG RS 04 1**Field:** LINE LENGTH TOTAL**Datatype:** NUMBER**Position:** 37**Optional:** N**Length:** 3**Zero to Null:****Domain Name:** VEG RS 04 1 LINE LENGTH TOTAL**Value****Meaning**

0

24

Rule Name: Veg 2000 RS 04 1 EW 06 1 Validate Required Field for Transect 1**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 RS 04 1 EW 06 1 Total Transect Line Length**PL/SQL Source Code:**

```
--
-----
--
-- Veg RS 04 1 EW 06 1 Total Transect Line Length
--
-- Nov 27, 2002 Chet - When a transect has been supplied with 0 of 0, check the
--                    other transect to make sure that it is present and is not
--                    also 0 of 0.
--
-----
--
```

```
VarSrceDataLineLenObs NUMBER;
VarSrceDataLineLenTot NUMBER;
VarSrceDataRecId VARCHAR2(3);
VarSrceDataTransectNo VARCHAR2(1);
rec NUMBER;
```

BEGIN

```
-- initialize this function's return value, code, and message
p_return_value := NULL;
p_return_code := 0;
p_return_message := NULL;
```

```
-- get the transect number from the source data record
VarSrceDataTransectNo := ffio.getc ( 'TRANSECT#' );
```

```
-- get the source data record type ID for the source data record; the only
-- expected values are:
-- 041 = Veg RS 04 1
-- 061 = Veg EW 06 1
VarSrceDataRecId := ffio.getc ( 'RECORD ID' );
```

```
-- check if the source data record type ID is either 041 or 061, that is, check
-- if the source data record type ID is acceptable for this rule
IF ( VarSrceDataRecId IN ( '041', '061' ) )
THEN
```

```
-- the source data record type ID is either 041 or 061, that is, the source
-- data record type ID is acceptable for this rule, this is a good thing
```

Record: VEG RS 04 1**Field:** PLOT1 UTIL CODE**Datatype:** CHAR**Position:** 40**Length:** 1**Optional:** Y**Zero to Null:****Rule Name:** Veg 2000 Look Up Forage Utilization Code**PL/SQL Source Code:**

```

code VARCHAR2(10);
trans_num NUMBER;
BEGIN
code := ffio.getc;
p_return_code := 0;
trans_num := ffio.getn('TRANSECT#', -1);

IF trans_num = -1 THEN
p_return_code := 3;
p_return_message := ' The Transect Number ****' ||ffio.getc('TRANSECT#')||
' is non mumeric.';
RETURN;
END IF;

IF ( code IS NULL ) THEN
IF ( trans_num = 1 ) THEN
p_return_code := 2;
p_return_message := ' For Transect #1, the field ' ||
ffio.get_current_item_name|| ' is missing.';
END IF;
RETURN;
END IF;

BEGIN
SELECT forage_utilzn_class_cd
INTO p_return_value
FROM forage_utilization_cds
WHERE forage_utilzn_class_cd = code;
EXCEPTION
WHEN NO_DATA_FOUND THEN
p_return_code := 1;
p_return_message := ' For Transect #' ||to_char(trans_num)||
', the field ' ||ffio.get_current_item_name||
' has the value ****' ||code|| '****. It is invalid or ' ||
'not recognized from the list of values in the Forage ' ||
'Utilization Code table.';
END;

```

Rule Name: Veg 2000 RS 04 1 EW 06 1 Validate Required Field for Transect 1**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG RS 04 1**Field:** PLOT2 UTIL CODE**Datatype:** CHAR**Position:** 41**Length:** 1**Optional:** Y**Zero to Null:**

Rule Name: Veg 2000 Look Up Forage Utilization Code

PL/SQL Source Code: Rule code has already been printed.

Rule Name: Veg 2000 RS 04 1 EW 06 1 Validate Required Field for Transect 1

PL/SQL Source Code: Rule code has already been printed.

Record: VEG RS 04 1

Field: PLOT3 UTIL CODE

Datatype: CHAR

Position: 42

Length: 1

Optional: Y

Zero to Null:

Rule Name: Veg 2000 Look Up Forage Utilization Code

PL/SQL Source Code: Rule code has already been printed.

Rule Name: Veg 2000 RS 04 1 EW 06 1 Validate Required Field for Transect 1

PL/SQL Source Code: Rule code has already been printed.

Record: VEG RS 04 1

Field: PLOT4 UTIL CODE

Datatype: CHAR

Position: 43

Length: 1

Optional: Y

Zero to Null:

Rule Name: Veg 2000 Look Up Forage Utilization Code

PL/SQL Source Code: Rule code has already been printed.

Rule Name: Veg 2000 RS 04 1 EW 06 1 Validate Required Field for Transect 1

PL/SQL Source Code: Rule code has already been printed.

Record: VEG RS 04 1

Field: FORAGE PLOT FULL

Datatype: CHAR

Position: 44

Length: 1

Optional: Y

Zero to Null: N

Domain Name: VEG SAMPLE X INDICATOR

Value

Meaning

-

No

X

Yes

Rule Name: Veg 2000 04 1Determine Observed Plot Fraction

PL/SQL Source Code:

BEGIN

p_return_code := 0;

IF (ffio.getc('TRANSECT#') = '2') THEN

RETURN;

END IF;

IF (LENGTH(TRANSLATE(ffio.getc('FORAGE PLOT FULL') ||
ffio.getc('FORAGE PLOT HALF') ||
ffio.getc('FORAGE PLOT QTR'), 'X-', 'X')) > 1) THEN

Record: VEG RS 04 1**Field:** FORAGE PLOT HALF**Datatype:** CHAR**Position:** 45**Optional:** Y**Length:** 1**Zero to Null:** N**Domain Name:** VEG SAMPLE X INDICATOR**Value****Meaning**

-

No

X

Yes

Record: VEG RS 04 1**Field:** FORAGE PLOT QTR**Datatype:** CHAR**Position:** 46**Optional:** Y**Length:** 1**Zero to Null:** N**Domain Name:** VEG SAMPLE X INDICATOR**Value****Meaning**

-

No

X

Yes

Record: VEG RS 04 1**Field:** GRAMINOIDS COLLECTED**Datatype:** CHAR**Position:** 47**Optional:** Y**Length:** 1**Zero to Null:****Domain Name:** INDICATOR**Value****Meaning**

N

No (False)

Y

Yes (True)

Rule Name: Veg 2000 Validate Graminoids Collected**PL/SQL Source Code:**

```

flag VARCHAR2(10);
weight NUMBER;
BEGIN
  BEGIN
    weight := ffio.getn ( 'Graminoids Oven Dry Weight' );
    flag := ffio.getc('Graminoids Collected');
    EXCEPTION WHEN VALUE_ERROR THEN
      p_return_code := 3;
      p_return_message := ' The value for GRAMINIODS OVEN DRIED WEIGHT ***'||
        ffio.getc( 'Graminoids Oven Dry Weight' ) || '*** is non numeric';
      RETURN;
    END;
  END;

  IF ( flag = 'Y' ) THEN
    IF ( weight IS NULL ) THEN
      p_return_code := 1;
      p_return_message := ' The flag "' || flag || '" for "' ||
        ffio.get_current_item_name || ' is set, ' ||
        'but no **Graminoids Dried Weight** has been supplied.';
    ELSE
      p_return_code := 0;
    END IF;
  ELSIF ( flag IS NULL and weight IS NOT NULL ) THEN
    p_return_code := 4;

```

Rule Name: Veg 2000 RS 04 1 EW 06 1 Validate Required Field for Transect 1

PL/SQL Source Code: Rule code has already been printed.

Record: VEG RS 04 1

Field: FORBS COLLECTED

Datatype: CHAR

Position: 48

Length: 1

Optional: Y

Zero to Null:

Domain Name: INDICATOR

Value

N

Y

Meaning

No (False)

Yes (True)

Rule Name: Veg 2000 Validate Forbs Collected

PL/SQL Source Code:

```
flag VARCHAR2(10);
weight NUMBER;
BEGIN
  BEGIN
    weight := ffio.getn ( 'Forbs Oven Dry Weight' );
    flag := ffio.getc('Forbs Collected');
    EXCEPTION WHEN VALUE_ERROR THEN
      p_return_code := 3;
      p_return_message := ' The value for FORBS OVEN DRIED WEIGHT ***' ||
        ffio.getc( 'Forbs Oven Dry Weight' ) || '*** is non numeric';
      RETURN;
    END;
    IF ( flag = 'Y' ) THEN
      IF ( weight IS NULL ) THEN
        p_return_code := 1;
        p_return_message := ' The flag "' || flag || '" for "' ||
          ffio.get_current_item_name || '" is set, "' ||
          'but no **Oven Dried Weight*** has been supplied.';
      ELSE
        p_return_code := 0;
      END IF;
    ELSIF ( flag IS NULL and weight IS NOT NULL ) THEN
      p_return_code := 4;
      p_return_message := ' Since the flag ***"is missing' || '*** for ' ||
        ffio.get_current_item_name ||
        ' then a Oven Dried Weight of ' || to_char(weight) ||
        ' is not needed or the flag should be set as " Y ".';
    ELSIF ( flag = 'N' and weight IS NOT NULL ) THEN
      p_return_code := 2;
      p_return_message := ' If the flag ***"N' || '*** is set for ' ||
        ffio.get_current_item_name ||
        ' then a Oven Dried Weight of ' || to_char(weight) ||
        ' is not needed or the flag should be set as " Y ".';
    ELSE
      p_return_code := 0;
    END IF;
```

Record: VEG RS 04 1

Field: GRAMINOIDS OVEN DRY WEIGHT

		Datatype: NUMBER		
		Position: 49	Optional: Y	
		Length: 5	Zero to Null:	
Record: <u>VEG RS 04 1</u>	Field: <u>FORBS OVEN DRY WEIGHT</u>	Datatype: NUMBER		
		Position: 54	Optional: Y	
		Length: 5	Zero to Null:	
Record: <u>VEG RS 04 2</u>	Field: <u>RECORD ID</u>	Datatype: NUMBER		
		Position: 1	Optional: N	
		Length: 3	Zero to Null:	
Record: <u>VEG RS 04 2</u>	Field: <u>PAGE NUMBER</u>	Datatype: NUMBER		
		Position: 4	Optional: Y	
		Length: 2	Zero to Null:	
Record: <u>VEG RS 04 2</u>	Field: <u>SEQUENCE NUMBER</u>	Datatype: NUMBER		
		Position: 6	Optional: N	
		Length: 4	Zero to Null:	
Record: <u>VEG RS 04 2</u>	Field: <u>TRANSECT#</u>	Datatype: NUMBER		
		Position: 10	Optional: N	
		Length: 1	Zero to Null:	
Domain Name: TRANSECT NUMBER	Meaning			
Value				
1 - 2				
Record: <u>VEG RS 04 2</u>	Field: <u>ITEM NO.</u>	Datatype: NUMBER		
		Position: 11	Optional: N	
		Length: 2	Zero to Null:	
Record: <u>VEG RS 04 2</u>	Field: <u>SHRUB LAYER</u>	Datatype: CHAR		
		Position: 13	Optional: Y	
		Length: 2	Zero to Null:	
Domain Name: VEGETATION LAYER- SHRUB SUBSET	Meaning			
Value				
B1	shrubs and woody plants, 2 to 10 m tall			
B2	shrubs, woody plants <2 m tall; not tree seedlings			
Record: <u>VEG RS 04 2</u>	Field: <u>LONG FORM SPECIES CD</u>			

Datatype: CHAR**Position:** 15**Length:** 8**Optional:** Y**Zero to Null:****Rule Name:** Veg 2000 RS 04 2 Species Code**PL/SQL Source Code:**

```
--
-----
--
-- Veg RS 04 2 Species Code
--
-- try to find a record in the group_taxon_memberships table that is associated
-- with a species group definition group code of 'shtshrub' or 'talshrub' when
-- the shrub layer is B1; and 'shtshrub', 'talshrub', or 'tree' when the shrub
-- layer is B2, and with a taxonomic name that has a long code equal to the
-- source data species code, that is, try to find a valid short shrub or tall
-- shrub species code record for the layer B1 source data species code, or try
-- to find a valid short shrub, tall shrub, or tree species code record for the
-- layer B2 source data species code
--
-- if the shrub layer is B1, then the source data species code is treated as
-- though it could be either a short shrub or a tall shrub species
--
-- if the shrub layer is B2, then the source data species code is treated as
-- though it could be either a short shrub, a tall shrub, or a tree species
--
-- if such a record cannot be found for shrub layer B1, try to find a record in
-- the group_taxon_memberships table that is associated with a species group
-- definition group code of 'shtshrub' or 'talshrub', a taxonomic name that has
-- a long code that is null, and a group_taxon_memberships long code equal to
-- the source data species code, that is, try to find a valid unknown short or
-- tall shrub species code record for the source data species code
--
-- if such a record cannot be found for shrub layer B2, try to find a record in
-- the group_taxon_memberships table that is associated with a species group
-- definition group code of 'shtshrub', 'talshrub', or 'tree', a taxonomic name
-- that has a long code that is null, and a group_taxon_memberships long code
-- equal to the source data species code, that is, try to find a valid unknown
-- short shrub, tall shrub, or tree species code record for the source data
-- species code
--
-- if successful in either search, return the skey of the record found in the
-- group_taxon_memberships table
--
-- Modifications
--
-- Nov 30, 1998 TMC - wo 673, change layer B1 by adding short shrubs, change
-- layer B2 by adding tall shrubs, reword the error messages
-- by including the measurement date
--
-- Dec 7, 1998 TMC - wo 673, add the TO_CHAR function to the data collected
-- date in the invalid species error message to prevent the
-- OCX generation from raising a concatenation error
--
```

Record: VEG RS 04 2**Field:** PHENOLOGY**Datatype:** NUMBER**Position:** 23**Optional:** Y**Length:** 2**Zero to Null:****Domain Name:** VEGETATIVE PHENOLOGICAL CLASSEValueMeaning

0	Closed Bud
1	Deciduous:Buds with green tips; or Conifers: Swollen
10	Deciduous: Bare
2	Deciduous:Green leaf out but not unfolded;Conifers
3	Deciduous:Leaf Unfolding 25%; Conifers: Shoot cappe
4	Deciduous: Leaf Unfolding 50%; Conifers: Shoot elo
5	Deciduous: Leaf unfolding 75%; Conifers: shoot ful
6	Deciduous: Full leaf unfolded; Conifers: shoot mat
7	Deciduous: First leaves turned yellow
8	Deciduous: Leaf yellowing up to 50%
9	Deciduous: leaf yellowing over 50%

Rule Name: Veg 2000 RS 04 2 Look Up Species and Phenology Combination**PL/SQL Source Code:**

```
--
-- CAW 23/11/99 W0 762, returns a NULL phenology for a species
--   code of 'NO SHRUB'
--
--
-- RGH 02/04/2000 W0 834, returns  for a species
--   code of 'UNKNOWN'
--
--
--   RGH      MAR 07, 2002 - added condition to use new long codes if
--   after Apr 1, 2001 for new view
--
-- June 27, 2002 CAW - changed condition date from 2001 to 2002
```

```
phen VARCHAR2(10);
species VARCHAR2(10);
tl_skey NUMBER;
num_phens NUMBER;
match_phen VARCHAR2(10);
VarDataCollectedDate DATE;
NewSpeciesCodeDate Date;
```

```
BEGIN
p_return_value := NULL;
tl_skey := NULL;
phen := ffio.getc('Phenology');
IF ( length(phen) > 1 ) THEN
  phen := LTRIM(phen,'0');
END IF;
species := ffio.getc('LONG FORM SPECIES CD');
IF ( phen IS NULL AND species IS NULL ) THEN
  p_return_code := 0;
  RETURN;
ELSIF ( UPPER(species) LIKE 'NO%SHRUB' ) THEN
```

Record: VEG RS 04 2**Field:** SHRUB COV. HOR. DISTANCE1**Datatype:** NUMBER**Position:** 25**Optional:** Y**Length:** 3**Zero to Null:****Rule Name:** Veg 2000 Validate Shrub Horizontal Distance Coverage**PL/SQL Source Code:**

```
--
-- 2002-Jun-11 Chet      - added note to error message when total exceeds line
length
--                        observed that appears when the layer is blank to
suggest
--                        that as being the source of the problem.
--
-- 2002-Nov-27 Chet      - shrub coverages are now allowed to exceed the observed
length.
--                        Added code to indicate an error if the line length is
--                        exceeded and NO-SHRUB segments have been recorded.
--
```

```
total NUMBER;
species VARCHAR2(15);
layer VARCHAR2(10);
phen NUMBER;
rec_num INTEGER;
num_length NUMBER;
shrub_cov NUMBER;
item_no  NUMBER;
no_shrub_found  BOOLEAN;
BEGIN
total := 0;
rec_num := 1;

--
-- Validate the Item Number
--
BEGIN
item_no := ffile.getn('ITEM NO. ');
EXCEPTION WHEN VALUE_ERROR THEN
p_return_code := 5;
p_return_message := ' The value for ITEM NUMBER ****' ||
ffile.getc('ITEM NO. ') || '**** is non numeric';
RETURN;
END;

--
-- Validate the phenology number
--
BEGIN
phen := ffile.getn('Phenology ');
EXCEPTION WHEN VALUE_ERROR THEN
p_return_code := 6;
p_return_message := ' The value for Phenology ****' ||
ffile.getc('Phenology ') || '**** is non numeric';
RETURN;
END;
```

<u>Record:</u> <u>VEG RS 04 2</u>	<u>Field:</u> <u>SHRUB COV. HOR. DISTANCE2</u> Datatype: NUMBER Position: 28 Length: 3 Optional: Y Zero to Null:
<u>Record:</u> <u>VEG RS 04 2</u>	<u>Field:</u> <u>SHRUB COV. HOR. DISTANCE3</u> Datatype: NUMBER Position: 31 Length: 3 Optional: Y Zero to Null:
<u>Record:</u> <u>VEG RS 04 2</u>	<u>Field:</u> <u>SHRUB COV. HOR. DISTANCE4</u> Datatype: NUMBER Position: 34 Length: 3 Optional: Y Zero to Null:
<u>Record:</u> <u>VEG RS 04 2</u>	<u>Field:</u> <u>SHRUB COV. HOR. DISTANCE5</u> Datatype: NUMBER Position: 37 Length: 3 Optional: Y Zero to Null:
<u>Record:</u> <u>VEG RS 04 2</u>	<u>Field:</u> <u>SHRUB COV. HOR. DISTANCE6</u> Datatype: NUMBER Position: 40 Length: 3 Optional: Y Zero to Null:
<u>Record:</u> <u>VEG RS 04 2</u>	<u>Field:</u> <u>SHRUB COV. HOR. DISTANCE7</u> Datatype: NUMBER Position: 43 Length: 3 Optional: Y Zero to Null:
<u>Record:</u> <u>VEG RS 04 2</u>	<u>Field:</u> <u>SHRUB COV. HOR. DISTANCE8</u> Datatype: NUMBER Position: 46 Length: 3 Optional: Y Zero to Null:
<u>Record:</u> <u>VEG RS 04 2</u>	<u>Field:</u> <u>SHRUB COV. HOR. DISTANCE9</u> Datatype: NUMBER Position: 49 Length: 3 Optional: Y Zero to Null:
<u>Record:</u> <u>VEG RS 04 2</u>	<u>Field:</u> <u>SHRUB COV. HOR. DISTANCE10</u> Datatype: NUMBER Position: 52 Length: 3 Optional: Y Zero to Null:
<u>Record:</u> <u>VEG RS 04 2</u>	<u>Field:</u> <u>SHRUB COV. HOR. DISTANCE11</u>

	Datatype: NUMBER	
	Position: 55	Optional: Y
	Length: 3	Zero to Null:
Record: <u>VEG RS 04 2</u>	Field: <u>SHRUB COV. HOR. DISTANCE12</u>	
	Datatype: NUMBER	
	Position: 58	Optional: Y
	Length: 3	Zero to Null:
Record: <u>VEG RS 04 2</u>	Field: <u>S1</u>	
	Datatype: CHAR	
	Position: 61	Optional: Y
	Length: 1	Zero to Null:
Record: <u>VEG RS 04 2</u>	Field: <u>S2</u>	
	Datatype: CHAR	
	Position: 62	Optional: Y
	Length: 1	Zero to Null:
Record: <u>VEG RS 04 2</u>	Field: <u>S3</u>	
	Datatype: CHAR	
	Position: 63	Optional: Y
	Length: 1	Zero to Null:
Record: <u>VEG RS 04 3</u>	Field: <u>RECORD ID</u>	
	Datatype: NUMBER	
	Position: 1	Optional: N
	Length: 3	Zero to Null:
Record: <u>VEG RS 04 3</u>	Field: <u>PAGE NUMBER</u>	
	Datatype: NUMBER	
	Position: 4	Optional: N
	Length: 2	Zero to Null:
Record: <u>VEG RS 04 3</u>	Field: <u>SEQUENCE NUMBER</u>	
	Datatype: NUMBER	
	Position: 6	Optional: N
	Length: 4	Zero to Null:
Record: <u>VEG RS 04 3</u>	Field: <u>TRANSECT#</u>	
	Datatype: NUMBER	
	Position: 10	Optional: N
	Length: 1	Zero to Null:
Domain Name: TRANSECT NUMBER		
Value	Meaning	

1 - 2

Record: VEG RS 04 3**Field:** NOTES**Datatype:** CHAR**Position:** 11**Length:** 256**Optional:** Y**Zero to Null:****Record:** VEG EW 06 1**Field:** RECORD ID**Datatype:** NUMBER**Position:** 1**Length:** 3**Optional:** N**Zero to Null:****Record:** VEG EW 06 1**Field:** PAGE NUMBER**Datatype:** NUMBER**Position:** 4**Length:** 2**Optional:** Y**Zero to Null:****Record:** VEG EW 06 1**Field:** SEQUENCE NUMBER**Datatype:** NUMBER**Position:** 6**Length:** 4**Optional:** N**Zero to Null:****Record:** VEG EW 06 1**Field:** PL**Datatype:** CHAR**Position:** 10**Length:** 1**Optional:** N**Zero to Null:****Domain Name:** VEG SAMPLE PL - PLOT NO**Value**

E

I

M

N

S

W

Meaning**Record:** VEG EW 06 1**Field:** MEASUREMENT DATE**Datatype:** DATE**Position:** 11**Length:** 9**Date Mask:** YYYYMONDD**Optional:** Y**Zero to Null:****Domain Name:** DATE MIN JAN 01 1997 - FULL**Value**

1997JAN01

- 3999JAN01

Meaning

minum date for 1999 Ground Sampling Format - YYM

Rule Name: Veg 2000 RS 04 1 EW 06 1 Validate Required Field for Transect 1**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG EW 06 1**Field:** CREW PERSON #1

Datatype: CHAR**Position:** 20**Length:** 4**Optional:** Y**Zero to Null:****Rule Name:** Veg 2000 Validate Person Against First Crew List**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 Check the Person's Initials**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG EW 06 1**Field:** CREW PERSON #2**Datatype:** CHAR**Position:** 24**Length:** 4**Optional:** Y**Zero to Null:****Rule Name:** Veg 2000 Validate Person Against First Crew List**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 Check the Person's Initials**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG EW 06 1**Field:** TRANSECT#**Datatype:** NUMBER**Position:** 28**Length:** 1**Optional:** N**Zero to Null:****Domain Name:** TRANSECT NUMBERValue

1

- 2

Meaning**Rule Name:** Veg 2000 EW 06 1 Look Up Coarse Woody Debris Transect Cluster El**PL/SQL Source Code:**

--

--

-- Lookup Coarse Woody Debris Transect Cluster Element Specification

--

--

-- Modifications

--

-- Nov 12, 1998 TMC - wo 562, the spatial_delimiters table was changed without

-- any communication to the DRM group, the change modified

-- the record with a sptld_type of LIN and a

-- total_line_len of 24.0: the sptld_type is now TRA, the

-- length is now 24.0, and the total_line_length is now

-- null; changed the first select statement to look for a

-- sptld_type of TRA and to match the length with the source

-- data horiz length total, added an error message for error

-- point 1

--

--

sd_skey NUMBER;

Record: VEG EW 06 1**Field:** TOTAL PAGES**Datatype:** NUMBER**Position:** 29**Optional:** Y**Length:** 2**Zero to Null:****Record:** VEG EW 06 1**Field:** RANDOM AZIMUTH**Datatype:** NUMBER**Position:** 31**Optional:** N**Length:** 3**Zero to Null:****Domain Name:** AZIMUTH BRG 0 TO 359**Value**

0 - 359

Meaning**Rule Name:** Veg 2000 RS 04 1 EW 06 1 Validate Required Field for Transect 1**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG EW 06 1**Field:** HORIZ. LENGTH OBSERVED**Datatype:** NUMBER**Position:** 34**Optional:** N**Length:** 3**Zero to Null:****Rule Name:** Veg 2000 RS 04 1 EW 06 1 Validate Required Field for Transect 1**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 RS 04 1 EW 06 1 Observed Transect Line Length**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 EW 06 1 Observed and Total Horizontal Lengths - Allow 0**PL/SQL Source Code:**

--

-- Veg EW 06 1 Observed and Total Horizontal Lengths - Allow 0m of 0m Only Once

--

-- Modifications

--

```
-- Jan 22, 1999 TMC - wo 690, this business rule no longer applies to the Veg
--                      RS 04 1 Line Length Observed field, the total line length
--                      cannot be 0 any more, so a 0 of 0 field value combination
--                      can't occur either
```

--

--

VarNextSrceDataRecNo NUMBER;

VarObsLineLenTran1 NUMBER;

VarObsLineLenTran2 NUMBER;

VarOrigRecNo NUMBER;

VarTotLineLenTran1 NUMBER;

VarTotLineLenTran2 NUMBER;

Record: VEG EW 06 1**Field:** HORIZ. LENGTH TOTAL**Datatype:** NUMBER**Position:** 37**Optional:** N**Length:** 3**Zero to Null:****Domain Name:** VEG EW 06 1 HORIZ LENGTH TOTAL**Value****Meaning**

12

24

30

6

Rule Name: Veg 2000 RS 04 1 EW 06 1 Validate Required Field for Transect 1**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 RS 04 1 EW 06 1 Total Transect Line Length**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG EW 06 2**Field:** RECORD ID**Datatype:** NUMBER**Position:** 1**Optional:** N**Length:** 3**Zero to Null:****Record:** VEG EW 06 2**Field:** PAGE NUMBER**Datatype:** NUMBER**Position:** 4**Optional:** Y**Length:** 2**Zero to Null:****Record:** VEG EW 06 2**Field:** SEQUENCE NUMBER**Datatype:** NUMBER**Position:** 6**Optional:** N**Length:** 4**Zero to Null:****Record:** VEG EW 06 2**Field:** TRANSECT#**Datatype:** NUMBER**Position:** 10**Optional:** N**Length:** 1**Zero to Null:****Domain Name:** TRANSECT NUMBER**Value****Meaning**

1 - 2

Record: VEG EW 06 2**Field:** ITEM NO.**Datatype:** NUMBER**Position:** 11**Optional:** N**Length:** 2**Zero to Null:****Record:** VEG EW 06 2**Field:** SPECIES**Datatype:** CHAR**Position:** 13**Optional:** Y**Length:** 3**Zero to Null:**

Rule Name: Veg 2000 Tree Species Short Code

PL/SQL Source Code: Rule code has already been printed.

Record: VEG EW 06 2

Field: DIAMETER

Datatype: NUMBER

Position: 16

Length: 4

Optional: Y

Zero to Null:

Domain Name: VEG SAMPLE CWD DIAMETER

Value

1.0 - 999.9

Meaning

Rule Name: Veg 2000 Check That the Field is Present When the Species is Pre

PL/SQL Source Code:

```
species VARCHAR2(10);
BEGIN
species := ffio.getc('SPECIES');
IF ( species IS NOT NULL AND ffio.getc IS NULL ) THEN
p_return_code := 1;
IF ( ffio.getc('RECORD ID') like '06_' ) THEN
p_return_message := 'For species "' || species || '" in Transect#'||
ffio.getc('TRANSECT#') || ' Item No ' || ffio.getc('ITEM NO.') ||
' the field ' || ffio.get_current_item_name ||
' must be supplied';
ELSE
p_return_message := 'For species "' || species ||
'" the field ' || ffio.get_current_item_name ||
' must be supplied';
END IF;
END IF;
```

Record: VEG EW 06 2

Field: LENGTH

Datatype: NUMBER

Position: 20

Length: 3

Optional: Y

Zero to Null:

Record: VEG EW 06 2

Field: TRANSECT % DECAY CLASS 1

Datatype: CHAR

Position: 23

Length: 2

Optional: Y

Zero to Null:

Domain Name: VEG SAMPLE ENCRYPTED PCT

Value

--
00 - 99
1 - 99

Meaning

100% (as well)
zero, ('0' is not allowed!)
100 % (true story!)

Rule Name: Veg 2000 EW 06 Convert Percent Decay Class 1

PL/SQL Source Code:

```
curr_field VARCHAR2(10);
```

Rule Name: Veg 2000 Check That the Field is Present When the Species is Pre
PL/SQL Source Code: Rule code has already been printed.

Record: VEG EW 06 2

Field: TRANSECT CLASS OTHER

Datatype: CHAR

Position: 25

Optional: Y

Length: 1

Zero to Null:

Domain Name: VEG SAMPLE DECAY CLASS >1

Value

Meaning

-

Not Applicable

2

intact, hard to partly decaying

3

hard, large pieces, partly decaying

4

small, blocky pieces

5

many small pieces, soft portions

Rule Name: Veg 2000 EW 06 Validate Decay Class Other for Multi-Class CWD Pi
PL/SQL Source Code:

BEGIN

p_return_code := 0;

p_return_value := NULL;

IF (ffio.getc('TRANSECT % Decay Class 1') != '--') THEN

IF (NVL(ffio.getc,'-') = '-') THEN

p_return_code := 1;

p_return_message := 'For item #' ||ffio.getc ('ITEM NO.')||

' on Transect #'||ffio.getc ('TRANSECT#')||

' % Decay Class 1 was < 100% so the Other Decay Class for '||

'the remaining section must be specified.';

ELSIF (ffio.getc = '1') THEN

p_return_code := 2;

p_return_message := 'For item #' ||ffio.getc ('ITEM NO.')||

' on Transect #'||ffio.getc ('TRANSECT#')||

' % Decay Class 1 was < 100% so the Other Decay Class for '||

'the remaining section cannot be specified as "1"';

END IF;

ELSE

IF (NVL(ffio.getc,'-') != '-') THEN

p_return_code := 3;

p_return_message := 'For item #' ||ffio.getc ('ITEM NO.')||

' on Transect #'||ffio.getc ('TRANSECT#')||

' % Decay Class 1 was 100% so the Other Decay Class for '||

'the remaining section must not be specified.';

END IF;

END IF;

Rule Name: Check for incorrect presence of Dashes

PL/SQL Source Code:

--

--

Record: VEG EW 06 2**Field:** TILT ANGLE DEGREE**Datatype:** NUMBER**Position:** 26**Optional:** Y**Length:** 2**Zero to Null:****Domain Name:** ANGLE DEGREEValue

0 - 90

Meaning**Rule Name:** Veg 2000 Check That the Field is Present When the Species is Pre**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG EW 06 2**Field:** MERCH**Datatype:** CHAR**Position:** 28**Optional:** Y**Length:** 1**Zero to Null:****Domain Name:** VEG SAMPLE X INDICATORValue

-

X

No

Yes

Meaning**Record:** VEG EW 06 2**Field:** P/R**Datatype:** CHAR**Position:** 29**Optional:** Y**Length:** 1**Zero to Null:****Domain Name:** VEG SAMPLE X INDICATORValue

-

X

No

Yes

Meaning**Record:** VEG EW 06 2**Field:** DECAY CLASS FOR PIECE**Datatype:** CHAR**Position:** 30**Optional:** Y**Length:** 1**Zero to Null:****Domain Name:** VEG SAMPLE DECAY CLASSValue

-

1

2

3

4

5

Not Applicable

intact, hard

intact, hard to partly decaying

hard, large pieces, partly decaying

hard, large pieces, partly decaying

many small pieces, soft portions

Meaning**Rule Name:** Veg 2000 Check That the Field is Present When the Species is Pre**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Check for incorrect presence of Dashes**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG EW 06 2**Field:** S1**Datatype:** CHAR**Position:** 31**Optional:** Y**Length:** 1**Zero to Null:**

Record: VEG EW 06 2**Field:** S2**Datatype:** CHAR**Position:** 32**Length:** 1**Optional:** Y**Zero to Null:****Record:** VEG EW 06 2**Field:** S3**Datatype:** CHAR**Position:** 33**Length:** 1**Optional:** Y**Zero to Null:****Record:** VEG EW 06 3**Field:** RECORD ID**Datatype:** NUMBER**Position:** 1**Length:** 3**Optional:** N**Zero to Null:****Record:** VEG EW 06 3**Field:** PAGE NUMBER**Datatype:** NUMBER**Position:** 4**Length:** 2**Optional:** N**Zero to Null:****Record:** VEG EW 06 3**Field:** SEQUENCE NUMBER**Datatype:** NUMBER**Position:** 6**Length:** 4**Optional:** N**Zero to Null:****Record:** VEG EW 06 3**Field:** TRANSECT#**Datatype:** NUMBER**Position:** 10**Length:** 1**Optional:** N**Zero to Null:****Domain Name:** TRANSECT NUMBER**Value**

1 - 2

Meaning**Record:** VEG EW 06 3**Field:** ITEM NO.**Datatype:** NUMBER**Position:** 11**Length:** 2**Optional:** N**Zero to Null:****Record:** VEG EW 06 3**Field:** SPECIES**Datatype:** CHAR**Position:** 13**Length:** 3**Optional:** Y**Zero to Null:**

Rule Name: Veg 2000 Tree Species Short Code

PL/SQL Source Code: Rule code has already been printed.

Record: VEG EW 06 3

Field: ON TRANSECT HORIZ. LENGTH

Datatype: NUMBER

Position: 16

Optional: Y

Length: 3

Zero to Null:

Rule Name: Veg 2000 Check That the Field is Present When the Species is Pre

PL/SQL Source Code: Rule code has already been printed.

Rule Name: Veg 2000 EW 06 3 Check Square Area of odd pieces to Area of roun

PL/SQL Source Code:

```
-----
--
-- Veg EW 06 3 Check Square Area of odd pieces to Area of round pieces
--
-----
--
-- Feb 11, 2001 RGH wo.936 make square area of odd pieces greater than pie x r
squared where
--   r is the radius of 3.8
--
```

```
VarSrceDataPageNo NUMBER;
VarSrceDataTranNo NUMBER;
VarSrceDataVerticalDis NUMBER;
VarSrceDataHorizontalDis NUMBER;
Min_value NUMBER;
Min_area NUMBER;
```

BEGIN

```
-- initialize this function's return value, code, and message
```

```
p_return_value := NULL;
```

```
p_return_code := 0;
```

```
p_return_message := NULL;
```

```
-- get the page number
```

```
VarSrceDataPageNo := ffio.getn ( 'PAGE NUMBER' );
```

```
-- get the transect number from the source data record
```

```
VarSrceDataTranNo := ffio.getn ( 'TRANSECT#' );
```

```
-- get the horizon distance from the source data record
```

```
VarSrceDataHorizontalDis := ffio.getn;
```

```
-- get the vertical distance from the source data record
```

```
VarSrceDataVerticalDis := ffio.getn ( 'ON TRANSECT VERTI. LENGTH' );
```

```
IF ( (VarSrceDataHorizontalDis IS NULL) AND (VarSrceDataVerticalDis IS NOT NULL))
THEN
```

```
-- this is an erroneous condition since both measurement are required
```

Record: VEG EW 06 3**Field:** ON TRANSECT VERTI. LENGTH**Datatype:** NUMBER**Position:** 19**Optional:** Y**Length:** 3**Zero to Null:****Rule Name:** Veg 2000 Check That the Field is Present When the Species is Pre**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG EW 06 3**Field:** LENGTH**Datatype:** NUMBER**Position:** 22**Optional:** Y**Length:** 3**Zero to Null:****Record:** VEG EW 06 3**Field:** TRANSECT % DECAY CLASS 1**Datatype:** CHAR**Position:** 25**Optional:** Y**Length:** 2**Zero to Null:****Domain Name:** VEG SAMPLE ENCRYPTED PCT**Value****Meaning**

--

100% (as well)

00

- 99

zero, ('0' is not allowed!)

1

- 99

—

100 % (true story!)

Rule Name: Veg 2000 EW 06 Convert Percent Decay Class 1**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 Check That the Field is Present When the Species is Pre**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG EW 06 3**Field:** TRANSECT CLASS OTHER**Datatype:** CHAR**Position:** 27**Optional:** Y**Length:** 1**Zero to Null:****Domain Name:** VEG SAMPLE DECAY CLASS >1**Value****Meaning**

-

Not Applicable

2

intact, hard to partly decaying

3

hard, large pieces, partly decaying

4

small, blocky pieces

5

many small pieces, soft portions

Rule Name: Veg 2000 EW 06 Validate Decay Class Other for Multi-Class CWD Pi**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Check for incorrect presence of Dashes**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG EW 06 3**Field:** MERCH**Datatype:** CHAR**Position:**

Length: 28
Optional: Y
Zero to Null:
Domain Name: VEG SAMPLE X INDICATOR
Value **Meaning**
 - No
 X Yes
Record: VEG EW 06 3 **Field:** P/R

Datatype: CHAR
Position: 29
Length: 1
Optional: Y
Zero to Null:
Domain Name: VEG SAMPLE X INDICATOR
Value **Meaning**
 - No
 X Yes
Record: VEG EW 06 3 **Field:** DECAY CLASS FOR PIECE

Datatype: CHAR
Position: 30
Length: 1
Optional: Y
Zero to Null:
Domain Name: VEG SAMPLE DECAY CLASS
Value **Meaning**
 - Not Applicable
 1 intact, hard
 2 intact, hard to partly decaying
 3 hard, large pieces, partly decaying
 4 hard, large pieces, partly decaying
 5 many small pieces, soft portions
Rule Name: Veg 2000 Check That the Field is Present When the Species is Pre
PL/SQL Source Code: Rule code has already been printed.
Rule Name: Check for incorrect presence of Dashes
PL/SQL Source Code: Rule code has already been printed.
Record: VEG EW 06 3 **Field:** S4

Datatype: CHAR
Position: 31
Length: 1
Optional: Y
Zero to Null:

Record: VEG EW 06 4 **Field:** RECORD ID
Datatype: NUMBER
Position: 1
Length: 3
Optional: N
Zero to Null:

Record: VEG EW 06 4 **Field:** PAGE NUMBER
Datatype: NUMBER
Position: 4
Length: 2
Optional: N
Zero to Null:

Record: VEG EW 06 4**Field:** SEQUENCE NUMBER**Datatype:** NUMBER**Position:** 6**Optional:** N**Length:** 4**Zero to Null:****Record:** VEG EW 06 4**Field:** TRANSECT#**Datatype:** NUMBER**Position:** 10**Optional:** N**Length:** 1**Zero to Null:****Domain Name:** TRANSECT NUMBERValue

1 - 2

Meaning**Record:** VEG EW 06 4**Field:** NOTES**Datatype:** CHAR**Position:** 11**Optional:** Y**Length:** 256**Zero to Null:****Record:** VEG TD 08 1**Field:** RECORD ID**Datatype:** NUMBER**Position:** 1**Optional:** N**Length:** 3**Zero to Null:****Rule Name:** Veg 2000 Validation of Site Tree Data for CMI Samples**PL/SQL Source Code:**

```

-----
-- Started with Rule 12790
-- Current Status: bulk of coding complete. Need to finish error conditions and
--                  update the error messages from rule 12790.
-- July 12,2001 RGH Supplied error messages and modified workings of rule from
-- CWH original stub. Remove error messages at location *****
-- for this is checked in another rule
-- July 12, 2001 CAW Accomidate for the lead and second species being the same.
--                  but also raise an error message.
-- May 30, 2002 CAW Changed the routine to find the first 082 record rather than
--                  the last.
-----
--
-- debug_flag NUMBER;
-- i INTEGER;
-- n INTEGER;
-- VarOrigPageNo VARCHAR2(2);
-- VarOrigRecID VARCHAR2(3);
-- VarOrigRecNo NUMBER;
-- VarProject VARCHAR2(10);
-- VarSamType VARCHAR2(10);
-- VarSrcPlotType VARCHAR2(3);
-- VarSrcPlotBAF NUMBER;

```

Record: VEG TD 08 1**Field:** PAGE NUMBER**Datatype:** NUMBER**Position:** 4**Optional:** N**Length:** 2**Zero to Null:****Rule Name:** Veg 2000 Validate Page Number**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TD 08 1**Field:** SEQUENCE NUMBER**Datatype:** NUMBER**Position:** 6**Optional:** N**Length:** 4**Zero to Null:****Record:** VEG TD 08 1**Field:** PL**Datatype:** CHAR**Position:** 10**Optional:** N**Length:** 1**Zero to Null:****Domain Name:** VEG SAMPLE PL - PLOT NO**Value****Meaning**

E

I

M

N

S

W

Rule Name: Veg 2000 Card TD082 & TA1142 Verify Leading Species Live Count**PL/SQL Source Code:**

```

-----
--
-- Veg TS 10 4 Tree Type Leading Species Make sure it is actually a leading
species
-- by doing a count of all live(L)species on the Card TD082 (IPC) and
TA114
-- (Auxiliary plots) then compare the max count for that species to the Leading(L)
-- Species on the TS104 card..
-- Now make sure it is actually a leading species
-- by doing a count of all live(S) species on the Card TD082 (IPC)
and TA114
-- (Auxiliary plots) then compare the max count for that species to the Second
Leading(S)
-- Species on the TS104 card..
-- Now Count all the other (O) species occuring on card TD082 and
TA114 insureing that there counts
-- are not higher than the (S) or (L) trees found on card TS104
--
-- Initially, this was only to be done where on card TD081 the IPC (I) type plot
is variable radius plot
-- and there is one or more of VARIABLE PLOT B.A.F., PRF, or DIOPTER filled in
and
-- one of VARIABLE PLOT FULL, VARIABLE PLOT HALF, or VARIABLE PLOT QRTR must be
filled in.

```

Record: VEG TD 08 1**Field:** MEASUREMENT DATE**Datatype:** DATE**Date Mask:** YYYYMONDD**Position:** 11**Optional:** N**Length:** 9**Zero to Null:****Domain Name:** DATE MIN JAN 01 1997 - FULL**Value****Meaning**

1997JAN01 - 3999JAN01 minmum date for 1999 Ground Sampling Format - YYM

Record: VEG TD 08 1**Field:** VARIABLE PLOT B.A.F.**Datatype:** NUMBER**Position:** 20**Optional:** Y**Length:** 4**Zero to Null:****Domain Name:** VEG SAMPLE BASAL AREA FTR**Value****Meaning**

02.97	- 03.03	3.00 m**2/ha Prism
03.96	- 04.04	4.00 m**2/ha Prism Size or Relaskop Value
04.95	- 05.05	5.00 m**2/ha Prism
05.01	- 05.11	5.06 m**2/ha Relascop Value
06.19	- 06.31	6.25 m**2/ha Relascop Value
06.93	- 07.07	7.0 m**2/ha Prism
08.91	- 09.09	9.00 m**2/ha Prism Size or Relaskop Value
11.88	- 12.12	12.00 m**2/ha Prism
12.13	- 12.37	12.25 m**2/ha Relascop Value
15.84	- 16.16	16.00 m**2/ha Prism Size or Relascop Value
18.80	- 20.20	20.00 m**2/ha Prism
20.05	- 20.45	20.25 m**2/ha Relascop Value
24.75	- 25.25	25.00 m**2/ha Prism size or Relaskop Value

Rule Name: Veg 2000 TD 08 1 TA 11 1 Plot Type (Variable or Fixed)**PL/SQL Source Code:**

```
--
-----
--
-- Veg TD 08 1 TA 11 1 Plot Type (Variable or Fixed)
--
-- Modifications
--
-- Feb 25, 1999 TMC - wo 708, add a check to see that either the PRF or diopter
--                    is entered when the BAF is entered; add a check to see
--                    that if the BAF is blank, that the fixed plot radius is
--                    entered, or then if the PRF and diopter are entered then
--                    raise a BAF is missing error message; rewrite the IF
--                    statements to take into account these new conditions and
--                    some new error message texts
--
-- Mar 08, 1999 CAW - 1999 Sample Format Modification: a higher precision on the
--                    diopter allows a finer comparison for the BAF, PRF and
--                    diopter relationships. Comparison expanded from 1 or 2
--                    decimals of precision to 3.
```

<u>Record:</u> <u>VEG TD 08 1</u>	<u>Field:</u> <u>VARIABLE PLOT FULL</u>
	Datatype: CHAR
	Position: 24
	Length: 1
	Optional: Y
	Zero to Null:
Domain Name: VEG SAMPLE X INDICATOR	
<u>Value</u>	<u>Meaning</u>
-	No
X	Yes
<u>Record:</u> <u>VEG TD 08 1</u>	<u>Field:</u> <u>VARIABLE PLOT HALF</u>
	Datatype: CHAR
	Position: 25
	Length: 1
	Optional: Y
	Zero to Null:
Domain Name: VEG SAMPLE X INDICATOR	
<u>Value</u>	<u>Meaning</u>
-	No
X	Yes
<u>Record:</u> <u>VEG TD 08 1</u>	<u>Field:</u> <u>VARIABLE PLOT QTR</u>
	Datatype: CHAR
	Position: 26
	Length: 1
	Optional: Y
	Zero to Null:
Domain Name: VEG SAMPLE X INDICATOR	
<u>Value</u>	<u>Meaning</u>
-	No
X	Yes
<u>Record:</u> <u>VEG TD 08 1</u>	<u>Field:</u> <u>VARIABLE PLOT BDRY</u>
	Datatype: CHAR
	Position: 27
	Length: 1
	Optional: Y
	Zero to Null:
Domain Name: VEG SAMPLE X INDICATOR	
<u>Value</u>	<u>Meaning</u>
-	No
X	Yes
<u>Record:</u> <u>VEG TD 08 1</u>	<u>Field:</u> <u>VARIABLE PLOT SPLIT</u>
	Datatype: CHAR
	Position: 28
	Length: 1
	Optional: Y
	Zero to Null:
Domain Name: VEG SAMPLE X INDICATOR	
<u>Value</u>	<u>Meaning</u>
-	No
X	Yes
<u>Record:</u> <u>VEG TD 08 1</u>	<u>Field:</u> <u>PRF</u>
	Datatype: NUMBER
	Position: 29
	Length: 5
	Optional: Y
	Zero to Null:
<u>Record:</u> <u>VEG TD 08 1</u>	<u>Field:</u> <u>DIOPTER</u>

Datatype: NUMBER**Position:** 34**Optional:** Y**Length:** 5**Zero to Null:****Record:** VEG TD 08 1**Field:** AZIMUTH BDRY/SPLIT**Datatype:** NUMBER**Position:** 39**Optional:** Y**Length:** 3**Zero to Null:****Domain Name:** AZIMUTH BRG 0 TO 359**Value****Meaning**

0 - 359

Rule Name: Veg 2000 Validate Azimuth against Plot Type**PL/SQL Source Code:**

```
--
-- MAY 26,2000 RGH wo811; put indicator for Page no. and plot type in error
message
--
  azimuth NUMBER;
  fixed_plot_full VARCHAR2(1);
  var_plot_full VARCHAR2(1);
  page_no NUMBER;
  v_plot_type VARCHAR2(10);

BEGIN

  azimuth := ffio.getn('AZIMUTH BDRY/SPLIT');
  fixed_plot_full := ffio.getc('FIXED PLOT FULL');
  var_plot_full := ffio.getc('VARIABLE PLOT FULL');
  page_no := ffio.getn('PAGE NUMBER');
  v_plot_type := ffio.getc ('PL');

  p_return_code := 0;
  p_return_value := NULL;

  IF azimuth IS NULL AND fixed_plot_full IS NULL AND var_plot_full IS NULL THEN
    p_return_code := 1;
    p_return_message := 'On page number '||TO_CHAR(page_no)||
      ' on the *** ' || v_plot_type || ' *** Plot' ||
      ', the combination of azimuth bdry/split, variable plot full and fixed plot
full is invalid.' ||
      ' When the azimuth bdry/split is blank then one of fixed or variable plot
types must be FULL.';
  END IF;
```

Record: VEG TD 08 1**Field:** FIXED PLOT RADIUS**Datatype:** NUMBER**Position:** 42**Optional:** Y**Length:** 4**Zero to Null:****Record:** VEG TD 08 1**Field:** FIXED PLOT FULL**Datatype:** CHAR

	Position: 46	Optional: Y
	Length: 1	Zero to Null:
Domain Name: VEG SAMPLE X INDICATOR		
<u>Value</u>	<u>Meaning</u>	
-	No	
X	Yes	
Record: <u>VEG TD 08 1</u>	Field: <u>FIXED PLOT HALF</u>	
	Datatype: CHAR	
	Position: 47	Optional: Y
	Length: 1	Zero to Null:
Domain Name: VEG SAMPLE X INDICATOR		
<u>Value</u>	<u>Meaning</u>	
-	No	
X	Yes	
Record: <u>VEG TD 08 1</u>	Field: <u>FIXED PLOT QRTR</u>	
	Datatype: CHAR	
	Position: 48	Optional: Y
	Length: 1	Zero to Null:
Domain Name: VEG SAMPLE X INDICATOR		
<u>Value</u>	<u>Meaning</u>	
-	No	
X	Yes	
Record: <u>VEG TD 08 1</u>	Field: <u>FIXED PLOT BDRY</u>	
	Datatype: CHAR	
	Position: 49	Optional: Y
	Length: 1	Zero to Null:
Domain Name: VEG SAMPLE X INDICATOR		
<u>Value</u>	<u>Meaning</u>	
-	No	
X	Yes	
Record: <u>VEG TD 08 1</u>	Field: <u>FIXED PLOT SPLIT</u>	
	Datatype: CHAR	
	Position: 50	Optional: Y
	Length: 1	Zero to Null:
Domain Name: VEG SAMPLE X INDICATOR		
<u>Value</u>	<u>Meaning</u>	
-	No	
X	Yes	

Rule Name: Veg 2000 TD 08 1 TA 11 1 Tree Cluster Element Specification

PL/SQL Source Code:

--

--

-- Veg Tree Cluster Element Specification

Rule Name: Veg 2000 Determine Tree Details Partial Start Bearing

PL/SQL Source Code:

```
--
-- MAY 26,2000 RGH wo811; put indicator for Page no. and plot type in error
message
--
typ VARCHAR2(10);
qual VARCHAR2(10);
azm NUMBER;
baf VARCHAR2(10);
fpr VARCHAR2(10);
v_plot_type VARCHAR2(10);
BEGIN
p_return_code := 0;
p_return_value := null;
p_return_message := null;
v_plot_type := ffio.getc ('PL');

azm := ffio.getn('AZIMUTH BDRY/SPLIT', -1);
IF azm = -1 THEN
p_return_code := 3;
p_return_message := ' The value acquired for the azimuth bdry/split ***' ||
ffio.getc('AZIMUTH BDRY/SPLIT') || '*** is non numeric,'||
' for the ***' || v_plot_type || ' *** Plot.';
END IF;
IF ( ffio.getn('FIXED PLOT RADIUS') IS NOT NULL ) THEN
IF ( ffio.getc('FIXED PLOT FULL') IS NOT NULL ) THEN
typ := 'F';
ELSIF ( ffio.getc('FIXED PLOT HALF') IS NOT NULL ) THEN
typ := 'H';
ELSIF ( ffio.getc('FIXED PLOT QRTR') IS NOT NULL ) THEN
typ := 'Q';
ELSE
p_return_code := 1;
fpr := to_char (ffio.getn ( 'FIXED PLOT RADIUS' ));
p_return_message := ' For a Fixed Plot Radius of '
|| fpr || ', one of the fields ***FIXED Plot FULL,' ||
' FIXED Plot HALF or FIXED Plot QRTR*** must have a entry,'||
' for the ***' || v_plot_type || ' *** Plot.';

END IF;
IF ( ffio.getc('FIXED PLOT BDRY') IS NOT NULL ) THEN
qual := 'B';
ELSIF ( ffio.getc('FIXED PLOT SPLIT') IS NOT NULL ) THEN
qual := 'S';
END IF;
ELSE
IF ( ffio.getc('VARIABLE PLOT FULL') IS NOT NULL ) THEN
typ := 'F';
ELSIF ( ffio.getc('VARIABLE PLOT HALF') IS NOT NULL ) THEN
typ := 'H';
ELSIF ( ffio.getc('VARIABLE PLOT QRTR') IS NOT NULL ) THEN
typ := 'Q';
ELSE
p_return_code := 2;
baf := to_char (ffio.getn ( 'VARIABLE PLOT B.A.F.' ));
```

Rule Name: Veg 2000 Determine Partial End Bearing

PL/SQL Source Code:

```
-- May 10,2001 JPK wo811; put indicator for plot type in error messages
```

```
typ VARCHAR2(10);
qual VARCHAR2(10);
azm NUMBER;
baf VARCHAR2(10);
fpr VARCHAR2(10);

v_plot_type VARCHAR2(10);

BEGIN

    p_return_code := 0;

    v_plot_type := ffio.getc ('PL');
    azm := ffio.getn('AZIMUTH BDRY/SPLIT', -1);

    IF azm = -1 THEN

        p_return_code := 3;
        p_return_message := ' The value acquired for the azimuth bdry/split ***' ||
ffio.getc('AZIMUTH BDRY/SPLIT') || '*** is non numeric.';

    END IF;

    IF ( ffio.getn('FIXED PLOT RADIUS') IS NOT NULL ) THEN

        IF ( ffio.getc('FIXED PLOT FULL') IS NOT NULL ) THEN
            typ := 'F';
        ELSIF ( ffio.getc('FIXED PLOT HALF') IS NOT NULL ) THEN
            typ := 'H';
        ELSIF ( ffio.getc('FIXED PLOT QRTR') IS NOT NULL ) THEN
            typ := 'Q';
        ELSE
            p_return_code := 1;
            fpr := to_char (ffio.getn ( 'FIXED PLOT RADIUS' ));
            p_return_message := ' For a Fixed Plot Radius of ***' || fpr || '***, one of
the fields FIXED Plot FULL,' ||
' FIXED Plot HALF or FIXED Plot QRTR must have a entry,' ||
' for the *** ' || v_plot_type || ' *** Plot.';
        END IF;

    IF ( ffio.getc('FIXED PLOT BDRY') IS NOT NULL ) THEN
        qual := 'B';
    ELSIF ( ffio.getc('FIXED PLOT SPLIT') IS NOT NULL ) THEN
        qual := 'S';
    END IF;

    ELSE

        IF ( ffio.getc('VARIABLE PLOT FULL') IS NOT NULL ) THEN
            typ := 'F';
```

<u>Record:</u> <u>VEG TD 08 1</u>	<u>Field:</u> <u>TOTAL PAGES</u>	
	Datatype: NUMBER	
	Position: 51	Optional: Y
	Length: 2	Zero to Null:
<u>Record:</u> <u>VEG TD 08 2</u>	<u>Field:</u> <u>RECORD ID</u>	
	Datatype: NUMBER	
	Position: 1	Optional: N
	Length: 3	Zero to Null:
<u>Record:</u> <u>VEG TD 08 2</u>	<u>Field:</u> <u>PAGE NUMBER</u>	
	Datatype: NUMBER	
	Position: 4	Optional: N
	Length: 2	Zero to Null:
<u>Record:</u> <u>VEG TD 08 2</u>	<u>Field:</u> <u>SEQUENCE NUMBER</u>	
	Datatype: NUMBER	
	Position: 6	Optional: N
	Length: 4	Zero to Null:
<u>Record:</u> <u>VEG TD 08 2</u>	<u>Field:</u> <u>TREE NUMBER</u>	
	Datatype: NUMBER	
	Position: 10	Optional: Y
	Length: 3	Zero to Null:

Rule Name: Veg 2000 TD 08 2 Check the Tree Details and Site Tree Data

PL/SQL Source Code:

```
--
-----
--
-- Veg 1999 TD 08 2 Check the Tree Details and Site Tree Data
--
-- Modifications
--
-- Nov 4, 1999 TMC - wo 704, add a check to see that the TS 10 4 boring
--                   diameter is equal to the TD 08 2 DBH
--
-- May 2 , 2000 RGH - wo 817, add ability to check more than one TS 10 4 record
--                   with same
--                   tree #
--
-- Oct 19, 2000 RGH - wo 912, exclude CMI sample types M and F to be <= to 5.64
--
-- Oct 21, 2002 CAW - wo 1014, accomidate for the process that states that stem
--                   mapping of CMI samples is from the centre of the tree, not
--                   the fact of the tree as per VRI samples.
-----
--
```

Rule Name: Veg 2000 TD 08 2 TA 11 4 Check for the Presence of Tree Loss Ind

PL/SQL Source Code:

```
--
-- MAY 23,2000 RGH wo811; put indicator for Page no. and plot type in error
message
--
tree_no NUMBER;
rec NUMBER;
curr_rec VARCHAR2(20);
next_rec VARCHAR2(10);
v_page_no VARCHAR2(10);
v_plot_rec VARCHAR2(10);
v_plot_type VARCHAR2(10);
v_plot_id NUMBER;
BEGIN
p_return_code := 0;
curr_rec := ffio.getc('RECORD ID');
v_page_no := ffio.getc('PAGE NUMBER');
v_plot_type := NULL;
v_plot_rec := NULL;
v_plot_id := NULL;

IF ( curr_rec = '082' ) THEN
next_rec := '092';
v_plot_rec := '091';
ELSIF ( ffio.getc('M OR E 1') ||ffio.getc('REM BARK')||
ffio.getc('TREE LENGTH')||ffio.getc('HEIGHT TO LIVE CROWN') IS NOT NULL )
THEN
next_rec := '115';
v_plot_rec := '111';
ELSE
RETURN;
END IF;

-- get the first record of the VEG CARD

IF v_plot_rec = '091' THEN
v_plot_id := ffio.get_next_record_by_item(v_plot_rec, 'PAGE NUMBER',v_page_no);
ELSIF v_plot_rec = '111' THEN
v_plot_id := ffio.get_previous_record_by_item(v_plot_rec, 'PAGE
NUMBER',v_page_no);
END IF;

BEGIN
tree_no := ffio.getn('TREE NUMBER');
EXCEPTION WHEN VALUE_ERROR THEN
p_return_code := 2;
p_return_message := ' The Tree No. ***' || ffio.getc('TREE NUMBER') ||
'*** for Veg Card ' || curr_rec || ' is non numeric.';
RETURN;
END;

rec := ffio.get_next_record_by_item (next_rec, 'DAMAGE AGENT TREE
NUMBER',TO_CHAR(tree_no),
'PAGE NUMBER',v_page_no);
```

Rule Name: Veg 2000 TS 10 4 Check That There is a Random Tree

PL/SQL Source Code:

```
--
-----
--
-- Veg TS 08 2 Live/Dead
--
-- This rule will fire for each 08 2 record. The rule will check that
-- if there are live trees on the 08 card then one of them must be a random
-- tree on the 10 card and if there is a random tree on the 10 card then that
-- tree must be a live tree on the 08 card.
--
-- June 14, 2000 LME - there should only be a random tree if it appears
--                      as a live tree at the IPC and if there are live
--                      trees at the IPC there must be a random tree
--
-- Jul 12, 2000 LME - changed the rule so that it is fired off the 08 2 record
--                      instead of the 10 4 record so that it is always fired
--
-- Jul 25, 2000 CAW - for National Forest Inventory (NFI) sample types F and M,
--                      there should be no random trees.
--
-----
--
VarOrigPageNo  VARCHAR2(20);
VarOrigRecID   VARCHAR2(30);
VarOrigRecNo   NUMBER;
VarRandomRecID VARCHAR2(30) := '104';
VarSrceDataTreeNo VARCHAR2(10);
VarAnotherRec  NUMBER;
VarIPCLiveTree VARCHAR2(10);
VarFoundLiveTree NUMBER;
VarFindRandomTrees NUMBER;
VarCountRandomTrees NUMBER;
VarSampleType  VARCHAR2(10);

BEGIN
-- initialize this function's return value and code
  p_return_value := NULL;
  p_return_code := 0;

-- get the record ID for the current record
  VarOrigRecID := ffio.getc ( 'RECORD ID' );

-- get the record number for the current record
  VarOrigRecNo := ffio.get_current_record;

-- get the page number for the current record
  VarOrigPageNo := ffio.getc ( 'PAGE NUMBER' );

-- get the live/dead indicator of the first tree
  VarIPCLiveTree := ffio.getc('LIVE/DEAD');

-- check to see if there was a previous 08 2 record
  VarAnotherRec := ffio.get_previous_record_by_item ( VarOrigRecID,
    'PAGE NUMBER', VarOrigPageNo);
```

Rule Name: Veg 2000 Check for Unwanted TA 11 4/TD 08 2 Duplicate Tree Numbe
PL/SQL Source Code:

```
--
-- 2000
-----
--
-- Veg 2000 Check for Unwanted TA 11 4/TD 08 2 Duplicate Tree Numbers on the
Current Page
--
-----
--
-- May 03,2001   JPK   wo811; put indicator for plot type in error messages

VarSrceDataPageNo NUMBER;
VarSrceDataRecId VARCHAR2(20);
VarSrceDataTreeNo VARCHAR2(10) := NULL;
VarSrceDataFieldName VARCHAR2(100);
VarSameTreeNoSeenBefore VARCHAR2(10);
VarSrceLogGrade VARCHAR2(1);
VarSrceLogLength VARCHAR2(2);
VarSrceLogSoundPct VARCHAR2(2);

v_plot_type VARCHAR2(10);
v_plot_rec VARCHAR2(10);
v_plot_id NUMBER;

BEGIN

-- initialize this function's return code, and message
p_return_code := 0;
p_return_message := NULL;

-- initialize the variables to be used for deriving the plot type
v_plot_type := NULL;
v_plot_rec := NULL;
v_plot_id := NULL;

-- get the record ID
VarSrceDataRecId := ffio.getc ( 'RECORD ID' );

-- get the page number
VarSrceDataPageNo := ffio.getn ( 'PAGE NUMBER' );

-- get the tree number
VarSrceDataTreeNo := ffio.getc ( 'TREE NUMBER' );

IF VarSrceDataRecId = '082' THEN
    v_plot_rec := '081';
ELSE
    v_plot_rec := '111';
END IF;

v_plot_id := ffio.get_previous_record_by_item(v_plot_rec, 'PAGE
```

Record: VEG TD 08 2**Field:** SPECIES**Datatype:** CHAR**Position:** 13**Length:** 3**Optional:** Y**Zero to Null:****Rule Name:** Veg 2000 Check That the Field is Present When the Tree Number is**PL/SQL Source Code:**

```
--
-----
--
-- Veg 1999 Check That the Field is Present When the Tree Number is Present
--
-- Modifications
--
-- Jun 23, 1999 TMC - wo 675, include the page number when finding a previous
--                   Veg card with the same record ID and tree number
--
--
-- Apr 04, 2000 RGH - wo 827, modify error point 1 for clarity - add record name
--
-- Nov 29, 2000 RGH - wo 924, add exception for 104 record where BORING DIAMETER
-- can be
-- be NULL if measure code in WHO or OUT
-- July 13, 2001 JPK wo811; put indicator for plot type in error messages
-----
--
```

```
VarAnotherVegCardRecNo INTEGER := 0;
VarSrceDataRecId VARCHAR2(20);
VarSrceDataTreeNo VARCHAR2(10) := NULL;
VarSrceDataRecName VARCHAR2(30);
```

```
VarSrceDataOrigRecId VARCHAR2(10);
VarSrceDataPageNo NUMBER;
v_plot_type VARCHAR2(10);
v_plot_rec VARCHAR2(10);
v_plot_id NUMBER;
```

```
BEGIN
```

```
-- initialize this function's return value, code, and message
p_return_value := NULL;
p_return_code := 0;
p_return_message := NULL;
```

```
-- initialize the variables to be used for deriving the plot type
v_plot_type := NULL;
v_plot_rec := NULL;
v_plot_id := NULL;
```

```
-- get the record ID of the current record;
VarSrceDataOrigRecId := ffio.getc ( 'RECORD ID' );
```

Rule Name: Veg 2000 Tree Species Short Code

PL/SQL Source Code: Rule code has already been printed.

Record: VEG TD 08 2

Field: LIVE/DEAD

Datatype: CHAR

Position: 16

Length: 1

Optional: Y

Zero to Null:

Domain Name: VEG SAMPLE EXTANT CODE

Value

Meaning

D

Dead

L

Live

Rule Name: Veg 2000 Check That the Field is Present When the Tree Number is

PL/SQL Source Code: Rule code has already been printed.

Rule Name: Veg 2000 Card TA114 & TD082 Check for Valid Crown Class for LIVE

PL/SQL Source Code:

```
-- 2000
-- Apr 6, 2000  RGH for wo 827
-- Jun 5, 2000  LME for wo 858
-- May 03,2001  JPK wo811; put indicator for plot type in error messages
--
```

```
page_no NUMBER;
tree_no NUMBER;
crown_class VARCHAR2(1);
live_dead   VARCHAR2(10);
stand_fall  VARCHAR2(10);
err_msg     VARCHAR2(100);
rec_id      VARCHAR2(3);
```

```
v_plot_type VARCHAR2(10);
v_plot_rec  VARCHAR2(10);
v_plot_id   NUMBER;
```

```
debug_flag BOOLEAN;
```

BEGIN

```
--debug_flag := TRUE;
debug_flag := FALSE;
```

```
-- initialize the variables to be used for deriving the plot type
v_plot_type := NULL;
v_plot_rec  := NULL;
v_plot_id   := NULL;
```

```
p_return_code := 0;
p_return_value := NULL;
rec_id := ffio.getc('RECORD ID');
page_no := ffio.getn('PAGE NUMBER');
tree_no := ffio.getn('TREE NUMBER');
crown_class := ffio.getc('CR. CL. DCIS');
stand_fall := ffio.getc('STAND/FALL');
live_dead := ffio.getc;
```

Record: VEG TD 08 2**Field:** STAND/FALL**Datatype:** CHAR**Position:** 17**Optional:** Y**Length:** 1**Zero to Null:****Domain Name:** VEG SAMPLE TREE STANCEValueMeaning

F

Fallen - not self supporting

S

Rule Name: Veg 2000 Check That the Field is Present When the Tree Number is**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TD 08 2**Field:** DBH**Datatype:** NUMBER**Position:** 18**Optional:** Y**Length:** 4**Zero to Null:****Domain Name:** VEG SAMPLE TREE DBH >= 4CMValueMeaning

4.0

- 999.9

Rule Name: Veg 2000 Check That the Field is Present When the Tree Number is**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TD 08 2**Field:** M OR E 1**Datatype:** CHAR**Position:** 22**Optional:** Y**Length:** 1**Zero to Null:****Domain Name:** VEG SAMPLE MEASURED/ESTIMATEDValueMeaning

E

Estimated

M

Measured

Rule Name: Veg 2000 Check That the Field is Present When the Tree Number is**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TD 08 2**Field:** REM BARK %**Datatype:** CHAR**Position:** 23**Optional:** Y**Length:** 2**Zero to Null:****Domain Name:** VEG SAMPLE ENCRYPTED PCTValueMeaning

--

100% (as well)

00

- 99

zero, ('0' is not allowed!)

1

- 99

—

100 % (true story!)

Rule Name: Veg 2000 Check That the Field is Present When the Tree Number is**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Validate Remaining Bark Percent against Wildlife Bark Code for T**PL/SQL Source Code:**

Record: VEG TD 08 2**Field:** TREE LENGTH**Datatype:** NUMBER**Position:** 25**Length:** 3**Optional:** Y**Zero to Null:****Domain Name:** VEG SAMPLE TREE LENGTH > 1.3M**Value**

1.3 - 99.9

Meaning**Rule Name:** Veg 2000 Check That the Field is Present When the Tree Number is**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TD 08 2**Field:** M OR E 2**Datatype:** CHAR**Position:** 28**Length:** 1**Optional:** Y**Zero to Null:****Domain Name:** VEG SAMPLE MEASURED/ESTIMATED**Value**

E

M

Meaning

Estimated

Measured

Rule Name: Veg 2000 Check That the Field is Present When the Tree Number is**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TD 08 2**Field:** CR. CL. DCIS**Datatype:** CHAR**Position:** 29**Length:** 1**Optional:** Y**Zero to Null:****Domain Name:** VEG SAMPLE CROWN CLASS**Value**

-

C

D

I

S

Meaning

none for dead trees

Codominant

Dominant

Intermediate

Suppressed

Rule Name: Veg 2000 Check That the Field is Present When the Tree Number is**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TD 08 2**Field:** HEIGHT TO LIVE CROWN**Datatype:** CHAR**Position:** 30**Length:** 2**Optional:** Y**Zero to Null:****Rule Name:** Veg 2000 Check That the Field is Present When the Tree Number is**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 TD 08 2 TA 11 4 Check Height to Live Crown**PL/SQL Source Code:**

Record: VEG TD 08 2**Field:** LOG 1/5 GRADE**Datatype:** CHAR**Position:** 32**Length:** 1**Optional:** Y**Zero to Null:****Rule Name:** Veg 2000 TD 08 2 TA 11 4 Log Grade Code**PL/SQL Source Code:**

```

--
-----
--
-- Veg Log Grade Code
--
-- Modifications
--
-- Mar 25, 1998 CAW - stated as rule 7760, converted for VRI Ground Sampling 1999
--
-- Note - wo 673, change the order of the tables in the select
--          statement so that the rule runs faster in the OCX; before
--          the tables were ordered group_taxon_memberships,
--          species_group_definitions, and taxonomic_names; now they
--          are ordered taxonomic_names, group_taxon_memberships, and
--          species_group_definitions; now the OCX finds a matching
--          record in taxonomic_names table first, then it tries to
--          find a matching record in the group_taxon_memberships
--          table, and then one in the species_group_definitions
--          table; before, it would find a matching record in the
--          group_taxon_memberships table first, and then try to find
--          a matching record in the species_group_definitions table,
--          and finally in the taxonomic names table; if it couldn't
--          find a match on all three, it would start with the third
--          table and try to find another record, and eventually
--          failing that, would go back to the second table and find
--          another matching record and then go back to the third
--          table and start searching all over again
--
-- July 5,2001   JPK   wo811; put indicator for plot type in error messages
--
-- Mar 07,2002 RGH -- UPPER(taxc.short_cd) because short cds are now like Ag
-----
--
VarCurrCardId VARCHAR2(10);
VarCurrFieldName VARCHAR2(80);
VarCurrRecId VARCHAR2(10);
VarCurrRecNo NUMBER;
VarDataCollectedDate DATE;
VarLgcMinLogLength NUMBER := 0;
VarLgcSkey NUMBER;
VarPrevRecCnt NUMBER;
ret_code NUMBER;
VarSrceDataLogGrade VARCHAR2(10) := NULL;
VarSrceDataLogLength NUMBER := NULL;
VarSrceDataSppCd VARCHAR2(10) := NULL;
VarSrceDataTreeNo VARCHAR2(4);
VarTaxcSkeyForSppCd NUMBER;
VarSgdefSkeyForSppCd NUMBER;

```

Rule Name: Veg 2000 TD 08 2 TA 11 4 Log Grade Combination

PL/SQL Source Code:

```
--
-- Mar 28, 2001  RGH  - wo.948 changed error code 8
--   ( comment back in ?? Why was it commented out in the first place don't know)
--
-- May 23,2001    JPK  wo811; put indicator for plot type in error messages

grade      VARCHAR2(10);
curr_field  VARCHAR2(50);
len         INTEGER;
sound_str   VARCHAR2(10);
sound       NUMBER;
broken_top  NUMBER;
projected_height  NUMBER;
dbh         NUMBER;
tree_no     VARCHAR2(10);

VarSrceDataOrigRecId  VARCHAR2(10);
VarSrceDataPageNo    NUMBER;
v_plot_type  VARCHAR2(10);
v_plot_rec   VARCHAR2(10);
v_plot_id    NUMBER;

BEGIN

    p_return_code := 0;
    p_return_message := null;

    -- initialize the variables to be used for deriving the plot type
    v_plot_type := NULL;
    v_plot_rec := NULL;
    v_plot_id := NULL;

    -- get the record ID of the current record;
    VarSrceDataOrigRecID := ffio.getc ( 'RECORD ID' );

    -- get the page number value
    VarSrceDataPageNo := ffio.getn ( 'PAGE NUMBER' );

    IF VarSrceDataOrigRecID = '114' THEN
        -- get the plot type (I,N,S,E,W)
        v_plot_rec := '111';
        v_plot_id := ffio.get_previous_record_by_item(v_plot_rec, 'PAGE
NUMBER',VarSrceDataPageNo);
        IF v_plot_id > 0 THEN
            v_plot_type := ffio.getc(v_plot_id, 'PL');
        END IF;
    END IF;

    grade := ffio.getc;
    curr_field := ffio.get_current_item_name;
    tree_no := ffio.getc('TREE NUMBER');
    len := ffio.getn(REPLACE(curr_field,'GRADE','LENGTH'), -1);

    IF len = 0 THEN
```

Rule Name: Veg 2000 TD 08 2 TA 11 4 Last Log Length

PL/SQL Source Code:

```
--
-----
-- Veg 2000 TD 08 2 TA 11 4 Last Log Length
--
-- May 24, 2000 RGH - wo 811, add Plot Type indicator N,S,W or E in error message
--
-----
tree_no  NUMBER:=0;
rec_id   VARCHAR2(10);
page_no  NUMBER;
logs_for_tree NUMBER:=0;
last_log_len VARCHAR2(10);
card     VARCHAR2(10);
v_plot_rec VARCHAR2(10);
v_plot_type VARCHAR2(10);
v_plot_id NUMBER;

BEGIN
p_return_code := 0;
v_plot_type := NULL;
v_plot_rec := NULL;
v_plot_id := NULL;

BEGIN
tree_no := ffio.getn ( 'TREE NUMBER' );
EXCEPTION WHEN VALUE_ERROR THEN
p_return_code := 3;
p_return_message := ' The value for Tree Number *"' ||
ffio.getc('TREE NUMBER') || '"* is non numeric';
RETURN;
END;

rec_id := ffio.getc('RECORD ID');
page_no := ffio.getn('PAGE NUMBER');
card := SUBSTR(rec_id,1,2);

v_plot_rec := card || '1';
v_plot_id := ffio.get_previous_record_by_item(v_plot_rec, 'PAGE NUMBER',page_no);
v_plot_type := ffio.getc ( v_plot_id, 'PL' );

IF (( ffio.get_next_record_by_item(rec_id, 'PAGE NUMBER', page_no, 'TREE NUMBER',
tree_no) > 0 )
OR ( ffio.getc('M OR E 1') ||ffio.getc('REM BARK')||
ffio.getc('TREE LENGTH')||ffio.getc('HEIGHT TO LIVE CROWN') IS NULL
)) THEN
RETURN;
END IF;

logs_for_tree := 0;
last_log_len := NULL;

FOR i in 1..4 LOOP
IF ffio.getc('LOG ' || to_char(i) || '/' || to_char(i+4) || ' GRADE')
IS NOT NULL
```

Rule Name: Veg 2000 Check For Valid Tree Species Code for a Log Grade of 'C'
PL/SQL Source Code:

```
--
-- Mar28,2001  RGH - wo 944  Add case where log grade B can only have species
-- 'FD', 'FDC', 'FDI'
--
page_no NUMBER;
tree_no NUMBER;
log_grade_cd VARCHAR2(10);
tree_species VARCHAR2(10);
BEGIN
p_return_code := 0;
p_return_message := NULL;

page_no := ffio.getn ( 'PAGE NUMBER' );

tree_no := ffio.getn ( 'TREE NUMBER' );

log_grade_cd := ffio.getc;

tree_species := ffio.getc('SPECIES');

IF ( log_grade_cd = 'C' ) AND
  ( UPPER( tree_species ) NOT IN ( 'ACT', 'FD', 'FDC', 'FDI' ) ) THEN
  p_return_code := 1;
  p_return_message := 'on page '
    || TO_CHAR ( page_no )
    || ', for tree '
    || TO_CHAR ( tree_no )
    || ' || tree_species || ' is an invalid tree species for a log grade of
  'C';
ELSIF ( log_grade_cd = 'B' ) AND
  ( UPPER( tree_species ) NOT IN ( 'FD', 'FDC', 'FDI' ) ) THEN
  p_return_code := 2;
  p_return_message := 'on page '
    || TO_CHAR ( page_no )
    || ', for tree '
    || TO_CHAR ( tree_no )
    || ' || tree_species || ' is an invalid tree species for a log grade of
  'B';
END IF;
```

Record: VEG TD 08 2

Field: LOG 1/5 LENGTH

Datatype: NUMBER

Position: 33

Length: 2

Optional: Y

Zero to Null: Y

Rule Name: Veg 2000 TD 08 2 TA 11 4 Calculate Log and Tree Lengths
PL/SQL Source Code:

--

Record: VEG TD 08 2**Field:** LOG 1/5 SOUND %**Datatype:** CHAR**Position:** 35**Optional:** Y**Length:** 2**Zero to Null:****Domain Name:** VEG SAMPLE ENCRYPTED PCT**Value**

--

00

- 99

1

- 99

—

Meaning

100% (as well)

zero, ('0' is not allowed!)

100 % (true story!)

Record: VEG TD 08 2**Field:** LOG 2/6 GRADE**Datatype:** CHAR**Position:** 37**Optional:** Y**Length:** 1**Zero to Null:****Rule Name:** Veg 2000 TD 08 2 TA 11 4 Log Grade Code**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 TD 08 2 TA 11 4 Log Grade Combination**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 Check For Valid Tree Species Code for a Log Grade of 'C**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TD 08 2**Field:** LOG 2/6 LENGTH**Datatype:** NUMBER**Position:** 38**Optional:** Y**Length:** 2**Zero to Null:** Y**Rule Name:** Veg 2000 TD 08 2 TA 11 4 Calculate Log and Tree Lengths**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TD 08 2**Field:** LOG 2/6 SOUND %**Datatype:** CHAR**Position:** 40**Optional:** Y**Length:** 2**Zero to Null:****Domain Name:** VEG SAMPLE ENCRYPTED PCT**Value**

--

00

- 99

1

- 99

—

Meaning

100% (as well)

zero, ('0' is not allowed!)

100 % (true story!)

Record: VEG TD 08 2**Field:** LOG 3/7 GRADE**Datatype:** CHAR**Position:** 42**Optional:** Y**Length:** 1**Zero to Null:**

Rule Name: Veg 2000 TD 08 2 TA 11 4 Log Grade Code

PL/SQL Source Code: Rule code has already been printed.

Rule Name: Veg 2000 TD 08 2 TA 11 4 Log Grade Combination

PL/SQL Source Code: Rule code has already been printed.

Rule Name: Veg 2000 Check For Valid Tree Species Code for a Log Grade of 'C

PL/SQL Source Code: Rule code has already been printed.

Record: VEG TD 08 2

Field: LOG 3/7 LENGTH

Datatype: NUMBER

Position: 43

Length: 2

Optional: Y

Zero to Null: Y

Rule Name: Veg 2000 TD 08 2 TA 11 4 Calculate Log and Tree Lengths

PL/SQL Source Code: Rule code has already been printed.

Record: VEG TD 08 2

Field: LOG 3/7 SOUND %

Datatype: CHAR

Position: 45

Length: 2

Optional: Y

Zero to Null:

Domain Name: VEG SAMPLE ENCRYPTED PCT

Value

--

00

- 99

1

- 99

—

Meaning

100% (as well)

zero, ('0' is not allowed!)

100 % (true story!)

Record: VEG TD 08 2

Field: LOG 4/8 GRADE

Datatype: CHAR

Position: 47

Length: 1

Optional: Y

Zero to Null:

Rule Name: Veg 2000 TD 08 2 TA 11 4 Log Grade Code

PL/SQL Source Code: Rule code has already been printed.

Rule Name: Veg 2000 TD 08 2 TA 11 4 Log Grade Combination

PL/SQL Source Code: Rule code has already been printed.

Rule Name: Veg 2000 Check For Valid Tree Species Code for a Log Grade of 'C

PL/SQL Source Code: Rule code has already been printed.

Record: VEG TD 08 2

Field: LOG 4/8 LENGTH

Datatype: NUMBER

Position: 48

Length: 2

Optional: Y

Zero to Null: Y

Rule Name: Veg 2000 TD 08 2 TA 11 4 Calculate Log and Tree Lengths

PL/SQL Source Code: Rule code has already been printed.

Record: VEG TD 08 2**Field:** LOG 4/8 SOUND %**Datatype:** CHAR**Position:** 50**Optional:** Y**Length:** 2**Zero to Null:****Domain Name:** VEG SAMPLE ENCRYPTED PCT**Value****Meaning**

--

100% (as well)

00 - 99

zero, ('0' is not allowed!)

1 - 99

100 % (true story!)

Record: VEG TD 08 2**Field:** WILDLIFE APPEARANCE CODE**Datatype:** CHAR**Position:** 52**Optional:** Y**Length:** 1**Zero to Null:****Rule Name:** Veg 2000 Look Up Wildlife Feature Code and Return Feature Name**PL/SQL Source Code:**

```

input_code VARCHAR2(10);
input_field VARCHAR2(80);
spc1 INTEGER;
spc2 INTEGER;
tree_num NUMBER;
cur_field VARCHAR2(100);
rec_no INTEGER;

VarSrceDataOrigRecId VARCHAR2(10);
VarSrceDataPageNo NUMBER;
v_plot_type VARCHAR2(10);
v_plot_rec VARCHAR2(10);
v_plot_id NUMBER;

BEGIN
  p_return_code := 0;
  p_return_value := null;

  -- initialize the variables to be used for deriving the plot type
  v_plot_type := NULL;
  v_plot_rec := NULL;
  v_plot_id := NULL;

  -- get the record ID of the current record;
  VarSrceDataOrigRecId := ffio.getc ( 'RECORD ID' );

  -- get the page number value
  VarSrceDataPageNo := ffio.getn ( 'PAGE NUMBER' );

  IF VarSrceDataOrigRecId = '114' THEN

    -- get the plot type (I,N,S,E,W)
    v_plot_rec := '111';
    v_plot_id := ffio.get_previous_record_by_item(v_plot_rec, 'PAGE
NUMBER',VarSrceDataPageNo);

```

Record: VEG TD 08 2**Field:** WILDLIFE CROWN CODE**Datatype:** CHAR**Position:** 53**Optional:** Y**Length:** 1**Zero to Null:****Rule Name:** Veg 2000 Look Up Wildlife Feature Code and Return Feature Name**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TD 08 2**Field:** WILDLIFE BARK CODE**Datatype:** CHAR**Position:** 54**Optional:** Y**Length:** 1**Zero to Null:****Rule Name:** Veg 2000 Look Up Wildlife Feature Code and Return Feature Name**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TD 08 2**Field:** WILDLIFE WOOD CODE**Datatype:** CHAR**Position:** 55**Optional:** Y**Length:** 1**Zero to Null:****Rule Name:** Veg 2000 Look Up Wildlife Feature Code and Return Feature Name**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TD 08 2**Field:** WILDLIFE LICHEN CODE**Datatype:** CHAR**Position:** 56**Optional:** Y**Length:** 1**Zero to Null:****Rule Name:** Veg 2000 Look Up Wildlife Feature Code and Return Feature Name**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TD 08 2**Field:** WILDLIFE USE CODE**Datatype:** CHAR**Position:** 57**Optional:** Y**Length:** 2**Zero to Null:****Rule Name:** Veg 2000 Lookup Wildlife Use Code**PL/SQL Source Code:**

```
code  VARCHAR2(10);
mark  BOOLEAN;
position  NUMBER(6);
item  VARCHAR2(6);
```

```
v_plot_type  VARCHAR2(10);
v_plot_rec   VARCHAR2(10);
```

Rule Name: Veg 2000 TD 08 2 TA 11 4 Check Wildlife Use Code if Tree Number
PL/SQL Source Code:

```
-- May 08,2001 JPK wo811; put indicator for plot type in error messages
```

```
VarCurrItemName VARCHAR2(40);  
VarRecNo INTEGER := 0;  
VarTreeNo VARCHAR2(10) := NULL;  
VarTreeStance VARCHAR2(1);
```

```
VarSrceDataOrigRecId VARCHAR2(10);  
VarSrceDataPageNo NUMBER;  
v_plot_type VARCHAR2(10);  
v_plot_rec VARCHAR2(10);  
v_plot_id NUMBER;
```

```
BEGIN
```

```
-- initialize this function's return code and value  
p_return_code := 0;  
p_return_value := NULL;
```

```
-- initialize the variables to be used for deriving the plot type  
v_plot_type := NULL;  
v_plot_rec := NULL;  
v_plot_id := NULL;
```

```
-- get the record ID of the current record;  
VarSrceDataOrigRecId := ffio.getc ( 'RECORD ID' );
```

```
-- get the page number value  
VarSrceDataPageNo := ffio.getn ( 'PAGE NUMBER' );
```

```
IF VarSrceDataOrigRecId = '114' THEN
```

```
-- get the plot type (I,N,S,E,W)  
v_plot_rec := '111';  
v_plot_id := ffio.get_previous_record_by_item(v_plot_rec, 'PAGE  
NUMBER',VarSrceDataPageNo);
```

```
IF v_plot_id > 0 THEN  
v_plot_type := ffio.getc(v_plot_id, 'PL');  
END IF;
```

```
END IF;
```

```
-- get the name of the current item  
VarCurrItemName := ffio.get_current_item_name;
```

```
-- get the tree number  
VarTreeNo := ffio.getc ( 'TREE NUMBER' );
```

```
-- get the tree stance  
VarTreeStance := ffio.getc ( 'STAND/FALL' );
```

Record: VEG TD 08 2**Field:** BROKEN TOP DIAMETER**Datatype:** CHAR**Position:** 59**Optional:** Y**Length:** 3**Zero to Null:****Rule Name:** Veg 2000 Validate Broken Top Diameter**PL/SQL Source Code:**

```

btd  VARCHAR2(10);

VarSrceDataOrigRecId  VARCHAR2(10);
VarSrceDataPageNo  NUMBER;
v_plot_type  VARCHAR2(10);
v_plot_rec  VARCHAR2(10);
v_plot_id  NUMBER;

BEGIN

    p_return_code := 0;

    -- initialize the variables to be used for deriving the plot type
    v_plot_type := NULL;
    v_plot_rec := NULL;
    v_plot_id := NULL;

    -- get the record ID of the current record;
    VarSrceDataOrigRecID := ffio.getc ( 'RECORD ID' );

    -- get the page number value
    VarSrceDataPageNo := ffio.getn ( 'PAGE NUMBER' );

    IF VarSrceDataOrigRecID = '114' THEN

        -- get the plot type (I,N,S,E,W)
        v_plot_rec := '111';
        v_plot_id := ffio.get_previous_record_by_item(v_plot_rec, 'PAGE
NUMBER',VarSrceDataPageNo);

        IF v_plot_id > 0 THEN
            v_plot_type := ffio.getc(v_plot_id, 'PL');
        END IF;

    END IF;

    btd := REPLACE(ffio.getc('Broken Top Diameter'),'---', NULL);

    IF ( btd IS NOT NULL ) THEN

BEGIN

    IF ROUND(ffio.getn('DBH')) < to_number(btd) THEN

        IF VarSrceDataOrigRecID = '114' THEN
            p_return_code := 1;
            p_return_message := ' on page '
                || TO_CHAR ( VarSrceDataPageNo )

```

Record: VEG TD 08 2**Field:** BROKEN TOP PROJ. HEIGHT**Datatype:** CHAR**Position:** 62**Optional:** Y**Length:** 2**Zero to Null:****Rule Name:** Veg 2000 TD 08 2 TA 11 4 Validate Broken Top**PL/SQL Source Code:**

```

grade  VARCHAR2(10);
len    VARCHAR2(10);
sound  VARCHAR2(10);
iden   VARCHAR2(10);
tree_no NUMBER;
bt_diam NUMBER;
pj_height NUMBER;
rec_no  NUMBER;
lst     INTEGER:=NULL;
rtyp    VARCHAR2(4);
tree_len NUMBER;
prev_tree NUMBER;
prev_lenght NUMBER;
next_tree NUMBER;
initial_id VARCHAR2(10);
dam_tree NUMBER;
page_no  NUMBER;

```

```

VarSrceDataOrigRecId  VARCHAR2(10);
VarSrceDataPageNo     NUMBER;
v_plot_type  VARCHAR2(10);
v_plot_rec   VARCHAR2(10);
v_plot_id    NUMBER;

```

```

BEGIN

```

```

    p_return_code:= 0;
    p_return_message := null;

```

```

    -- initialize the variables to be used for deriving the plot type
    v_plot_type := NULL;
    v_plot_rec := NULL;
    v_plot_id := NULL;

```

```

    -- get the record ID of the current record;
    VarSrceDataOrigRecID := ffio.getc ( 'RECORD ID' );

```

```

    -- get the page number value
    VarSrceDataPageNo := ffio.getn ( 'PAGE NUMBER' );

```

```

    IF VarSrceDataOrigRecID = '114' THEN

```

```

        -- get the plot type (I,N,S,E,W)
        v_plot_rec := '111';
        v_plot_id := ffio.get_previous_record_by_item(v_plot_rec, 'PAGE
NUMBER',VarSrceDataPageNo);

```

```

        IF v_plot_id > 0 THEN
            v_plot_type := ffio.getc(v_plot_id, 'PL');

```

Record: VEG TD 08 2**Field:** S1**Datatype:** CHAR**Position:** 64**Length:** 1**Optional:** Y**Zero to Null:****Rule Name:** Validate Tree Sector for CMI and NFI Samples**PL/SQL Source Code:**

```

--
-----
--
-- Veg TD 08 2 Check the S1 field for CMI sample (F and M)
--
-- Modifications
--
-- Jan 11, 2001 RGH - wo 931, add a check to allow if LOGS are more than 4
--                      (ie. there is another record for the tree in the flatfile).
--                      There need not be a S1 field on this record
--
-----
--
VarSampleType VARCHAR2(10);
VarSector VARCHAR2(10);
VarTreeNum NUMBER;
rec_id VARCHAR2(10);
page_no NUMBER;
card VARCHAR2(10);
v_plot_rec VARCHAR2(10);
v_plot_type VARCHAR2(10);
v_plot_id NUMBER;
VarPrevTS082RecNo NUMBER;
BEGIN
  p_return_code := 0;
  v_plot_type := NULL;
  v_plot_rec := NULL;
  v_plot_id := NULL;

  BEGIN
    VarTreeNum := ffio.getn ( 'TREE NUMBER' );
    EXCEPTION WHEN VALUE_ERROR THEN
      p_return_code := 1;
      p_return_message := ' The value for Tree Number ***' ||
        ffio.getc('TREE NUMBER') || '*** is non numeric';
      RETURN;
  END;

  END;

  rec_id := ffio.getc('RECORD ID');
  page_no := ffio.getn('PAGE NUMBER');
  card := SUBSTR(rec_id,1,2);

  v_plot_rec := card || '1';
  v_plot_id := ffio.get_previous_record_by_item(v_plot_rec, 'PAGE NUMBER',page_no);
  v_plot_type := ffio.getc ( v_plot_id, 'PL' );

```

Record: VEG TD 08 2**Field:** S2**Datatype:** CHAR**Position:** 65**Length:** 1**Optional:** Y**Zero to Null:****Rule Name:** Determine Residual Classification for CMI and NFI Samples**PL/SQL Source Code:**

```
--
-----
-- Jul 26, 2000 CAW - Initial Version for CFI and NFI samples and then later
--                   have the rule return Y to indicate that Yes it is a
--                   residual tree, N for No and NULL for unknown.
-- Aug 2, 2000 CAW - type in error message 2,S1 -> S2
--
--
-- Jan 11, 2001 RGH - wo 931, add a check to allow if LOGS are more than 4
--                   (ie. there is another record for the tree in the flatfile).
--                   There need not be a S2 field on this record
--
-----

VarSampleType VARCHAR2(10);
VarResidual VARCHAR2(10);
VarTreeNum NUMBER;
rec_id VARCHAR2(10);
page_no NUMBER;
card VARCHAR2(10);
v_plot_rec VARCHAR2(10);
v_plot_type VARCHAR2(10);
v_plot_id NUMBER;
VarPrevTS082RecNo NUMBER;
BEGIN
  p_return_code := 0;
  v_plot_type := NULL;
  v_plot_rec := NULL;
  v_plot_id := NULL;

  BEGIN
    VarTreeNum := ffio.getn ( 'TREE NUMBER' );
    EXCEPTION WHEN VALUE_ERROR THEN
      p_return_code := 1;
      p_return_message := ' The value for Tree Number ***' ||
        ffio.getc('TREE NUMBER') || '*** is non numeric';
      RETURN;
  END;

  END;

  rec_id := ffio.getc('RECORD ID');
  page_no := ffio.getn('PAGE NUMBER');
  card := SUBSTR(rec_id,1,2);

  v_plot_rec := card || '1';
  v_plot_id := ffio.get_previous_record_by_item(v_plot_rec, 'PAGE NUMBER',page_no);
  v_plot_type := ffio.getc ( v_plot_id, 'PL' );

  VarSampleType := ffio.getc(1, 'SAMPLE TYPE');
```

Record: VEG TD 08 3**Field:** RECORD ID**Datatype:** NUMBER**Position:** 1**Length:** 3**Optional:** N**Zero to Null:****Record:** VEG TD 08 3**Field:** PAGE NUMBER**Datatype:** NUMBER**Position:** 4**Length:** 2**Optional:** N**Zero to Null:****Record:** VEG TD 08 3**Field:** SEQUENCE NUMBER**Datatype:** NUMBER**Position:** 6**Length:** 4**Optional:** N**Zero to Null:****Record:** VEG TD 08 3**Field:** NOTES**Datatype:** CHAR**Position:** 10**Length:** 256**Optional:** Y**Zero to Null:****Record:** VEG TL 09 1**Field:** RECORD ID**Datatype:** CHAR**Position:** 1**Length:** 3**Optional:** N**Zero to Null:****Record:** VEG TL 09 1**Field:** PAGE NUMBER**Datatype:** NUMBER**Position:** 4**Length:** 2**Optional:** N**Zero to Null:****Rule Name:** Veg 2000 Validate Page Number**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TL 09 1**Field:** SEQUENCE NUMBER**Datatype:** NUMBER**Position:** 6**Length:** 4**Optional:** N**Zero to Null:****Record:** VEG TL 09 1**Field:** PL**Datatype:** CHAR**Position:** 10**Length:** 1**Optional:** N**Zero to Null:****Domain Name:** VEG SAMPLE PL - PLOT NO**Value**

E

I

Meaning

M
N
S
W

Record: VEG TL 09 1**Field:** TOTAL PAGES**Datatype:** NUMBER**Position:** 11**Length:** 2**Optional:** Y**Zero to Null:****Record:** VEG TL 09 2**Field:** RECORD ID**Datatype:** NUMBER**Position:** 1**Length:** 3**Optional:** N**Zero to Null:****Record:** VEG TL 09 2**Field:** PAGE NUMBER**Datatype:** NUMBER**Position:** 4**Length:** 2**Optional:** N**Zero to Null:****Record:** VEG TL 09 2**Field:** SEQUENCE NUMBER**Datatype:** NUMBER**Position:** 6**Length:** 4**Optional:** N**Zero to Null:****Record:** VEG TL 09 2**Field:** DAMAGE AGENT TREE NUMBER**Datatype:** NUMBER**Position:** 10**Length:** 3**Optional:** Y**Zero to Null:****Rule Name:** Veg 2000 TL 09 2 Validate Tree Loss Tree No Against Tree Detail**PL/SQL Source Code:**

```

tree_num NUMBER;
rec_no NUMBER;
BEGIN
p_return_code := 0;
p_return_message := null;
tree_num := ffile.getn('DAMAGE AGENT TREE NUMBER', -1);
IF (tree_num = -1 ) THEN
p_return_code := 2;
p_return_message := ' Damage Agent Tree Number ***' || ffile.getc('DAMAGE AGENT
TREE NUMBER')||
'***' is non numeric.';
END IF;
IF ( tree_num IS NOT NULL ) THEN
rec_no := ffile.get_previous_record_by_item ( '082', 'TREE NUMBER',
TO_CHAR(tree_num) );

```

Rule Name: Veg 2000 TL 09 2 Check For Blank Damage Agent Records

PL/SQL Source Code:

```
tree_num NUMBER;
BEGIN
  p_return_code := 0;
  p_return_message := null;
  tree_num := ffio.getn('DAMAGE AGENT TREE NUMBER');
  IF tree_num IS NULL THEN
    p_return_code := 1;
    p_return_message := 'There is a blank VEG TL 09 2 record.';
  END IF;
```

Record: VEG TL 09 2

Field: DAMAGE AGENT CODE A

Datatype: CHAR

Position: 13

Length: 3

Optional: Y

Zero to Null:

Rule Name: Veg 2000 TL 09 2 TA 11 5 Damage Agent Code

PL/SQL Source Code:

```
--
-----
--
-- Veg Damage Agent Code
--
-- use the coms.get_damage_agent_skey function to get the skey of the record in
-- the damage_agent_cds table which has a damage_agent_cd equal to the source
-- data damage agent code
--
--
-- Modifications
--
-- Dec 1, 1998 TMC - wo 639, change the way a hyphenated damage agent code is
-- handled; instead of converting '---' to null, now convert
-- '-', '--', and '---' to the new code UKN which represents
-- an unknown damage agent
--
-- Mar 13, 2000 RGH -modified to point to THE.DAMAGE_AGENT_ST_TBL. new damage
-- agent table
-- for VRI 2000. Doesn't use the coms function.
--
-- Sept 28,2000 RGH - added return in the Error handling to give the right
-- messages
-- July 05,2001 JPK wo811; put indicator for plot type in error messages
-----
--
```

```
fld_name VARCHAR2(40);
result VARCHAR2(40);
VarSrceDataDamageAgentCode VARCHAR2(10);
```

Rule Name: Veg 2000 Check That the Field is Present When the Tree Number is
PL/SQL Source Code: Rule code has already been printed.

Record: VEG TL 09 2

Field: DAMAGE AGENT SEVERITY A

Datatype: CHAR

Position: 16

Optional: Y

Length: 2

Zero to Null:

Rule Name: Veg 2000 TL 09 2 TA 11 5 Damage Agent Severity

PL/SQL Source Code:

```
--
--
-----
--
-- Veg Damage Agent Severity
--
-- Modifications
--
-- Dec 1, 1998 TMC - wo 639, change the way a hyphened damage agent code is
--                  handled; instead of converting '---' to null, now convert
--                  '-', '--', and '---' to the new code UKN which represents
--                  an unknown damage agent
--
-- Sep 26, 2000 RGH - wo 905, changed UNK code to U for Veg2000 and updated how
--                  local_da_skey is found to match method in Rule 12536 for
--                  2000 format.
--
--                  NOTE: this is a micky mouse BEGIN-END routine to componsate
for
--                  adding in a column for corporate damage agents,used in 2000.
--                  Their are 155 codes for 1999 and 141 for 2000.
--                  This causes many to many relationships between the codes.
--                  Therefore the rule has to make the decisions on what Damage agent skey to
--                  to use based on the 1999 damage agents.
--
-- Nov 29, 2000 RGH - wo 923, Nov 29, 2000 RGH added the condition where there is
--                  to be no Severity rating input for the Damage agent DSA
--                  per 'Vern Sundstron' email from John A. Muir
--
-- May 23,2001   JPK   wo811; put indicator for plot type in error messages
-- Mar 23, 2003   CAW   - refuse the values '--' or '-'
--
-----
--
VarDamageAgentCodeFieldName VARCHAR2(80);
iter      NUMBER;
local_da_skey      NUMBER;
result      NUMBER;
VarSrceDataDamageAgentCode VARCHAR2(10);
VarSrceDataDamageAgentSeverity VARCHAR2(10);

VarSrceDataOrigRecId   VARCHAR2(10);
VarSrceDataPageNo      NUMBER;
v_plot_type VARCHAR2(10);
v_plot_rec  VARCHAR2(10);
```

Record: VEG TL 09 2**Field:** DAMAGE AGENT CODE B**Datatype:** CHAR**Position:** 18**Optional:** Y**Length:** 3**Zero to Null:****Rule Name:** Veg 2000 TL 09 2 TA 11 5 Damage Agent Code**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TL 09 2**Field:** DAMAGE AGENT SEVERITY B**Datatype:** CHAR**Position:** 21**Optional:** Y**Length:** 2**Zero to Null:****Rule Name:** Veg 2000 TL 09 2 TA 11 5 Damage Agent Severity**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TL 09 2**Field:** LOSS INDICATOR CODE 1**Datatype:** CHAR**Position:** 23**Optional:** Y**Length:** 3**Zero to Null:****Rule Name:** Veg 2000 TL 09 2 TA 11 5 Tree Loss Indicator Code**PL/SQL Source Code:**

```
--
-----
--
-- Veg TL 09 2 TA 11 5 Tree Loss Indicator Code
--
-- Modifications
--
-- Jan 12, 1999 TMC - wo 621, convert the tree loss indicator code to its
--                   upercase equivalent, change a tree loss indicator code
--                   of CRK to CRO
-- Oct 22, 1999 CAW - wo 773, disable the conversion of CRK to CRO for 1999 data.
-- July 3,2001   JPK wo811; put indicator for plot type in error messages
-- Sept 17,2001  RGH wo 964 made the code NGC valid after2001 and AGC not
-- May 13, 2002  JPK wo973 prevent usage of DDI (should be DDH)
-----
--
```

```
VarCurrItemName VARCHAR2(40);
VarCurrTreeNo VARCHAR2(10);
VarSrceDataTreeLossIndCode VARCHAR2(10);
```

```
VarSrceDataOrigRecId VARCHAR2(10);
VarSrceDataPageNo NUMBER;
v_plot_type VARCHAR2(10);
```

Rule Name: Veg 2000 TL 09 2 TA 11 5 Check a Broken Top Loss Indicator to th
PL/SQL Source Code:

```
--
-----
--
-- TL 09 2 TA 11 5 Check a Broken Top Loss Indicator to the Broken Top
-- Measurements
--
-----
--
-- May 02,2001   JPK   wo811; put indicator for plot type in error message

ConstAuxPlotLossIndsRecId CONSTANT VARCHAR2(3) := '115';
ConstAuxPlotTreeDetailsRecId CONSTANT VARCHAR2(3) := '114';
ConstTreeDetailsRecId CONSTANT VARCHAR2(3) := '082';
ConstTreeLossIndsRecId CONSTANT VARCHAR2(3) := '092';

VarSrceDataBrokenTopDiam VARCHAR2(3);
VarSrceDataBrokenTopProjHt VARCHAR2(2);
VarSrceDataLossIndCode VARCHAR2(3);
VarSrceDataOrigRecId VARCHAR2(10);
VarSrceDataOrigRecNo NUMBER;
VarSrceDataPageNo NUMBER;
VarSrceDataTreeNo NUMBER;
VarSrchTreeDetailsRecId VARCHAR2(3);
VarSrchTreeDetailsRecNo NUMBER;

v_plot_type VARCHAR2(10);
v_plot_rec VARCHAR2(10);
v_plot_id NUMBER;

debug_flag BOOLEAN;

BEGIN

--debug_flag := TRUE;
debug_flag := FALSE;

-- initialize this function's return value, code, and message
p_return_value := NULL;
p_return_code := 0;
p_return_message := NULL;

-- initialize the variables to be used for deriving the plot type
v_plot_type := NULL;
v_plot_rec := NULL;
v_plot_id := NULL;

-- get the record number of this, the current or original, source data record
VarSrceDataOrigRecNo := ffio.get_current_record;

-- get the page number value
VarSrceDataPageNo := ffio.getn ( 'PAGE NUMBER' );
```

Record: VEG TL 09 2**Field:** LS IND 1 LOCATION +/- 1**Datatype:** CHAR**Position:** 26**Optional:** Y**Length:** 1**Zero to Null:****Domain Name:** NUMERIC SIGN**Value**

+

-

Meaning

Positive

Negative

Record: VEG TL 09 2**Field:** LS IND 1 LOCATION FROM**Datatype:** NUMBER**Position:** 27**Optional:** Y**Length:** 3**Zero to Null:**

Rule Name: Veg 2000 TL 09 2 TA 11 5 Check That the Loss Indicator Location
PL/SQL Source Code:

```
--
-----
--
-- Veg 1999 TL 09 2 TA 11 5 Check That the Loss Indicator Location is Within
-- the Tree Length
--
-- Modifications
--
-- Oct 29, 1999 TMC - wo 709, a loss indicator can only be for a live tree, if
-- the tree has a broken top, the loss indicator location
-- must be equal to the tree length
--
-- Nov 4, 1999 TMC - wo 778, change the search for a matching tree details
-- record to start at the beginning of the sample rather
-- than going backwards from the current location; multiple
-- tree details records don't have all the fields entered
-- that are entered on the first tree details record
--
-- Nov 4, 1999 TMC - wo 778, the restriction that the tree must be live was
-- revoked
--
-- July 18,2001 JPK wo811; put indicator for plot type in error messages
-----
--
```

```
VarLossIndCdFldName VARCHAR2(32);
VarLossIndLocnFldName VARCHAR2(32);
VarSrceDataLossIndCd VARCHAR2(3);
VarSrceDataLossIndLocn NUMBER;
VarSrceDataOrigRecId VARCHAR2(10);
VarSrceDataOrigRecNo NUMBER;
VarSrceDataPageNo NUMBER;
VarSrceDataTreeLen NUMBER;
VarSrceDataTreeNo NUMBER;
VarTreeDetailsRecNo NUMBER;
```

Rule Name: Veg 2000 TL 09 2 TA 11 5 Tree Loss Indicator Fields Optionality
PL/SQL Source Code:

```
-----  
--  
-- Veg Tree Loss Indicator Fields Optionality for Tree Loss Indicator Code  
--  
-- Modifications  
--  
-- Jan 11, 1999 TMC - wo 636, add a condition to check if the loss indicator  
--                      code is SNG, and if it is, the tree loss indicator  
--                      fields are optional, so don't check for their presence  
-- May 09,2001 JPK wo811; put indicator for plot type in error messages  
--  
-----  
--  
  
VarCurrItemName VARCHAR2(100);  
VarCurrLossIndCodeNo VARCHAR2(4);  
VarCurrTreeNo VARCHAR2(10);  
VarSrceDataLossIndCode VARCHAR2(10);  
  
VarSrceDataOrigRecId VARCHAR2(10);  
VarSrceDataPageNo NUMBER;  
v_plot_type VARCHAR2(10);  
v_plot_rec VARCHAR2(10);  
v_plot_id NUMBER;  
  
BEGIN  
  
    p_return_code := 0;  
  
    -- initialize the variables to be used for deriving the plot type  
    v_plot_type := NULL;  
    v_plot_rec := NULL;  
    v_plot_id := NULL;  
  
    -- get the record ID of the current record;  
    VarSrceDataOrigRecID := ffio.getc ( 'RECORD ID' );  
  
    -- get the page number value  
    VarSrceDataPageNo := ffio.getn ( 'PAGE NUMBER' );  
  
    IF VarSrceDataOrigRecID = '115' THEN  
  
        -- get the plot type (I,N,S,E,W)  
        v_plot_rec := '111';  
        v_plot_id := ffio.get_previous_record_by_item(v_plot_rec, 'PAGE  
NUMBER',VarSrceDataPageNo);  
  
        IF v_plot_id > 0 THEN  
            v_plot_type := ffio.getc(v_plot_id, 'PL');
```

Record: VEG TL 09 2**Field:** LS IND 1 LOCATION +/- 2**Datatype:** CHAR**Position:** 30**Optional:** Y**Length:** 1**Zero to Null:****Domain Name:** NUMERIC SIGNValue

+

-

Meaning

Positive

Negative

Record: VEG TL 09 2**Field:** LS IND 1 LOCATION TO**Datatype:** NUMBER**Position:** 31**Optional:** Y**Length:** 3**Zero to Null:****Rule Name:** Veg 2000 TL 09 2 TA 11 5 Check That the Loss Indicator Location**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 TL 09 2 TA 11 5 Tree Loss Indicator Fields Optionality**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TL 09 2**Field:** LS IND 1 FREQUENCY**Datatype:** CHAR**Position:** 34**Optional:** Y**Length:** 1**Zero to Null:****Rule Name:** Veg 2000 TL 09 2 TA 11 5 Tree Loss Indicator Fields Optionality**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TL 09 2**Field:** LOSS INDICATOR CODE 2**Datatype:** CHAR**Position:** 35**Optional:** Y**Length:** 3**Zero to Null:****Rule Name:** Veg 2000 TL 09 2 TA 11 5 Tree Loss Indicator Code**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 TL 09 2 TA 11 5 Check a Broken Top Loss Indicator to th**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TL 09 2**Field:** LS IND 2 LOCATION +/- 1**Datatype:** CHAR**Position:** 38**Optional:** Y**Length:** 1**Zero to Null:****Domain Name:** NUMERIC SIGNValue

+

-

Meaning

Positive

Negative

Record: VEG TL 09 2**Field:** LS IND 2 LOCATION FROM**Datatype:** NUMBER

Position: 39**Length:** 3**Optional:** Y**Zero to Null:****Rule Name:** Veg 2000 TL 09 2 TA 11 5 Check That the Loss Indicator Location**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 TL 09 2 TA 11 5 Tree Loss Indicator Fields Optionality**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TL 09 2**Field:** LS IND 2 LOCATION +/- 2**Datatype:** CHAR**Position:** 42**Length:** 1**Optional:** Y**Zero to Null:****Domain Name:** NUMERIC SIGN**Value**

+

-

Meaning

Positive

Negative

Record: VEG TL 09 2**Field:** LS IND 2 LOCATION TO**Datatype:** NUMBER**Position:** 43**Length:** 3**Optional:** Y**Zero to Null:****Rule Name:** Veg 2000 TL 09 2 TA 11 5 Check That the Loss Indicator Location**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 TL 09 2 TA 11 5 Tree Loss Indicator Fields Optionality**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TL 09 2**Field:** LS IND 2 FREQUENCY**Datatype:** CHAR**Position:** 46**Length:** 1**Optional:** Y**Zero to Null:****Rule Name:** Veg 2000 TL 09 2 TA 11 5 Tree Loss Indicator Fields Optionality**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TL 09 2**Field:** LOSS INDICATOR CODE 3**Datatype:** CHAR**Position:** 47**Length:** 3**Optional:** Y**Zero to Null:****Rule Name:** Veg 2000 TL 09 2 TA 11 5 Tree Loss Indicator Code**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 TL 09 2 TA 11 5 Check a Broken Top Loss Indicator to th**PL/SQL Source Code:** Rule code has already been printed.

Record: VEG TL 09 2**Field:** LS IND 3 LOCATION +/- 1**Datatype:** CHAR**Position:** 50**Optional:** Y**Length:** 1**Zero to Null:****Domain Name:** NUMERIC SIGN**Value**

+

-

Meaning

Positive

Negative

Record: VEG TL 09 2**Field:** LS IND 3 LOCATION FROM**Datatype:** NUMBER**Position:** 51**Optional:** Y**Length:** 3**Zero to Null:****Rule Name:** Veg 2000 TL 09 2 TA 11 5 Check That the Loss Indicator Location**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 TL 09 2 TA 11 5 Tree Loss Indicator Fields Optionality**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TL 09 2**Field:** LS IND 3 LOCATION +/- 2**Datatype:** CHAR**Position:** 54**Optional:** Y**Length:** 1**Zero to Null:****Domain Name:** NUMERIC SIGN**Value**

+

-

Meaning

Positive

Negative

Record: VEG TL 09 2**Field:** LS IND 3 LOCATION TO**Datatype:** NUMBER**Position:** 55**Optional:** Y**Length:** 3**Zero to Null:****Rule Name:** Veg 2000 TL 09 2 TA 11 5 Check That the Loss Indicator Location**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 TL 09 2 TA 11 5 Tree Loss Indicator Fields Optionality**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TL 09 2**Field:** LS IND 3 FREQUENCY**Datatype:** CHAR**Position:** 58**Optional:** Y**Length:** 1**Zero to Null:****Rule Name:** Veg 2000 TL 09 2 TA 11 5 Tree Loss Indicator Fields Optionality**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TL 09 2**Field:** STEM MAP AZIMUTH TO TREE**Datatype:** NUMBER

	Position: 59	Optional: Y
	Length: 3	Zero to Null:
Domain Name: AZIMUTH BRG 0 TO 359		
<u>Value</u>	<u>Meaning</u>	
0 - 359		

Rule Name: Veg 2000 Validate Stem Map Location to Ensure Unique Tree Positi

PL/SQL Source Code:

```

tree_no VARCHAR2(10);
dist NUMBER;
azm NUMBER;
rec NUMBER;
rec_id VARCHAR2(10);
BEGIN
p_return_code := 0;
p_return_value := NULL;

tree_no := ffio.getc('TREE NUMBER');
dist := ffio.getn ( 'STEM MAP DISTANCE TO TREE FACE' );
azm := ffio.getn ( 'STEM MAP AZIMUTH TO TREE' );
rec_id := ffio.getc('RECORD ID');

IF ( tree_no IS NULL OR dist IS NULL OR azm IS NULL ) THEN
RETURN;
END IF;

LOOP
rec := ffio.get_previous_record_by_item ( rec_id );
IF ( rec <= 0 OR ffio.set_current_record ( rec ) <= 0 ) THEN
EXIT;
END IF;

IF ( dist = ffio.getn ( 'STEM MAP DISTANCE TO TREE FACE' ) AND
azm = ffio.getn ( 'STEM MAP AZIMUTH TO TREE' ) AND
tree_no != ffio.getc('TREE NUMBER') ) THEN

p_return_code := 1;
p_return_message := 'The trees identified by Tree Numbers '||tree_no||
' and '||ffio.getc('TREE NUMBER')||
' both share the same physical space with duplicate'||
' positions of azimuth '||to_char(azm)||' and distance '||
to_char(dist);
RETURN;
END IF;
END LOOP;

```

Rule Name: Veg 2000 Non-NFI Samples: Check That the Field is Present When t

PL/SQL Source Code:

```

--
-----
--

```

```

VarAnotherVegCardRecNo INTEGER := 0;

```

Record: VEG TL 09 2**Field:** STEM MAP DISTANCE TO TREE F/**Datatype:** NUMBER**Position:** 62**Optional:** Y**Length:** 4**Zero to Null:****Rule Name:** Veg 2000 Convert Stem Map to X Coordinate**PL/SQL Source Code:**

```

x NUMBER;
y NUMBER;
azm NUMBER;
dist NUMBER;
curr_tree VARCHAR2(4);

BEGIN

p_return_code := 0;
p_return_value := null;
p_return_message := null;

curr_tree := ffio.getc ( 'DAMAGE AGENT TREE NUMBER' );
x := NULL;
y := NULL;

azm := ffio.getn('STEM MAP AZIMUTH TO TREE', -1);

IF azm = -1 THEN
    p_return_code := 4;
    p_return_message := ' For Tree Number ' || curr_tree ||
        ', the stem map azimuth to tree value ****' ||
        ffio.getc('STEM MAP AZIMUTH TO TREE') ||
        '*** is non numeric.';
END IF;

dist := ffio.getn('STEM MAP DISTANCE TO TREE FACE', -1);

IF dist = -1 THEN
    p_return_code := 5;
    p_return_message := ' For Tree Number ' || curr_tree ||
        ', the stem map distance to tree face value ****'
        || ffio.getc('STEM MAP DISTANCE TO TREE FACE')
        || '*** is non numeric.';
END IF;

IF ( dist IS NULL AND azm IS NULL ) THEN
    p_return_code := 0;

ELSIF ( dist IS NULL ) THEN
    p_return_code := 1;
    p_return_message := ' For Tree Number ' || curr_tree ||
        ' an azimuth of ' || to_char (azm) ||
        ' degrees was supplied, but no ***"Stem Map Distance"*** '
        || 'was supplied.';

```

Rule Name: Veg 2000 Validate that the Tree is actually located within the p
PL/SQL Source Code:

```
--
-- Chet Watmough 29/11/99 W0 791 - Too much of the data does not
--      pass the new restriction. Changing the 5cm
--      tolerance to 5%.
--
v_radius NUMBER;
v_prf  NUMBER;
v_rec_no NUMBER;
v_stem_dist NUMBER;
v_stem_azm NUMBER;
v_tree_dbh NUMBER;
v_tree_no VARCHAR2(10);
v_max_dist NUMBER;
BEGIN

p_return_code := 0;

v_tree_no := ffio.getc('DAMAGE AGENT TREE NUMBER');

--
-- Get the tree location
--

BEGIN
  v_stem_dist := ffio.getn ( 'STEM MAP DISTANCE TO TREE FACE');
EXCEPTION
  WHEN VALUE_ERROR THEN
    p_return_code := 1;
    p_return_message := 'Invalid STEM MAP DISTANCE for tree "'||
      v_tree_no||'";
    RETURN;
END;

BEGIN
  v_stem_azm := ffio.getn ( 'STEM MAP AZIMUTH TO TREE');
EXCEPTION
  WHEN VALUE_ERROR THEN
    p_return_code := 2;
    p_return_message := 'Invalid STEM MAP AZIMUTH for tree "'||
      v_tree_no||'";
    RETURN;
END;

--
-- Find the header record that contains the plot definition
--

IF ( ffio.getc('RECORD ID') = '092' ) THEN
  v_rec_no := ffio.get_previous_record_by_item ( '081' );
ELSE
  v_rec_no := ffio.get_previous_record_by_item ( '111',
    'PAGE NUMBER', ffio.getc('PAGE NUMBER'));
END IF;
```

Rule Name: Veg 2000 Non-NFI Samples: Check That the Field is Present When t
PL/SQL Source Code: Rule code has already been printed.

Record: <u>VEG TL 09 2</u>	Field: <u>S1</u> Datatype: CHAR Position: 66 Length: 1	Optional: Y Zero to Null:
Record: <u>VEG TL 09 2</u>	Field: <u>S2</u> Datatype: CHAR Position: 67 Length: 1	Optional: Y Zero to Null:
Record: <u>VEG TL 09 3</u>	Field: <u>RECORD ID</u> Datatype: NUMBER Position: 1 Length: 3	Optional: N Zero to Null:
Record: <u>VEG TL 09 3</u>	Field: <u>PAGE NUMBER</u> Datatype: NUMBER Position: 4 Length: 2	Optional: N Zero to Null:
Record: <u>VEG TL 09 3</u>	Field: <u>SEQUENCE NUMBER</u> Datatype: NUMBER Position: 6 Length: 4	Optional: N Zero to Null:
Record: <u>VEG TL 09 3</u>	Field: <u>NOTES</u> Datatype: CHAR Position: 10 Length: 256	Optional: Y Zero to Null:
Record: <u>VEG TS 10 1</u>	Field: <u>RECORD ID</u> Datatype: NUMBER Position: 1 Length: 3	Optional: N Zero to Null:
Record: <u>VEG TS 10 1</u>	Field: <u>PAGE NUMBER</u> Datatype: NUMBER Position: 4 Length: 2	Optional: Y Zero to Null:

Rule Name: Veg 2000 Validate Page Number

PL/SQL Source Code: Rule code has already been printed.

Record: VEG TS 10 1

Field: SEQUENCE NUMBER

Datatype: NUMBER

Position: 6

Optional: N

Length: 4

Zero to Null:

Record: VEG TS 10 1

Field: PL

Datatype: CHAR

Position: 10

Optional: N

Length: 1

Zero to Null:

Domain Name: VEG SAMPLE PL - PLOT NO

Value

Meaning

E

I

M

N

S

W

Record: VEG TS 10 1

Field: PLOT RADIUS

Datatype: NUMBER

Position: 11

Optional: N

Length: 4

Zero to Null:

Record: VEG TS 10 1

Field: PLOT FULL

Datatype: CHAR

Position: 15

Optional: Y

Length: 1

Zero to Null:

Domain Name: VEG SAMPLE X INDICATOR

Value

Meaning

-

No

X

Yes

Record: VEG TS 10 1

Field: PLOT HALF

Datatype: CHAR

Position: 16

Optional: Y

Length: 1

Zero to Null:

Domain Name: VEG SAMPLE X INDICATOR

Value

Meaning

-

No

X

Yes

Record: VEG TS 10 1

Field: PLOT QRTR

Datatype: CHAR

Position: 17

Optional: Y

Length: 1

Zero to Null:

VEG SAMPLE X INDICATOR

<u>Value</u>	<u>Meaning</u>
Domain Name:	No
X	Yes
Record: <u>VEG TS 10 1</u>	Field: <u>PLOT BDRY</u>
	Datatype: CHAR
	Position: 18
	Length: 1
Domain Name: VEG SAMPLE X INDICATOR	Optional: Y
	Zero to Null:
<u>Value</u>	<u>Meaning</u>
-	No
X	Yes
Record: <u>VEG TS 10 1</u>	Field: <u>PLOT SPLIT</u>
	Datatype: CHAR
	Position: 19
	Length: 1
Domain Name: VEG SAMPLE X INDICATOR	Optional: Y
	Zero to Null:
<u>Value</u>	<u>Meaning</u>
-	No
X	Yes

Rule Name: Veg 2000 TS 10 1 Site Tree Plot End Bearing

PL/SQL Source Code:

```
--
-----
--
-- Veg TS 10 1 Determine Site Tree Plot End Bearing
--
-- Modifications
--
-- Mar 16, 1999 TMC - wo 716, wo 711, return a value of 360 for the plot end
--                    bearing for an odd numbered plot's half plot
--
-- Mar 29, 2000 LME - wo 830, return a value of 0 for the plot end
--                    bearing for an odd numbered plot's half plot
--
-----
--

plot_no NUMBER;
typ VARCHAR2(10);

BEGIN

p_return_code := 0;
```

Rule Name: Veg 2000 TS 10 1 Small Live Site Tree Cluster Element Specificat

PL/SQL Source Code:

```

sd_skey NUMBER;
spop NUMBER;
baf NUMBER;
rad NUMBER;
BEGIN
p_return_code := 0;
sd_skey := NULL;
rad := ffio.getn('PLOT RADIUS');
BEGIN
SELECT spatial_delim_skey
INTO sd_skey
FROM spatial_delimiters
WHERE ( radius = rad AND sptld_type = 'FXPLT' );
EXCEPTION WHEN NO_DATA_FOUND THEN
p_return_code:= 1;
WHEN TOO_MANY_ROWS THEN
p_return_code := 2;
END;
IF ( sd_skey IS NOT NULL ) THEN
SELECT MAX(sub_pop_skey)
INTO spop
FROM subpopulations,
species_group_definitions
WHERE popltn_elmnt_cd = 'TREE'
AND max_dbh = 3.9
AND min_tree_ht = .1
AND subpopulations.sgdef_spp_group_skey = spp_group_skey
AND group_cd = 'TREE'
AND tree_extant_cd = 'LIVE';
IF ( spop IS NULL ) THEN
p_return_code := 3;
ELSE
BEGIN
SELECT bd_spec_skey
INTO p_return_value
FROM biometric_design_specification
WHERE subp_sub_pop_skey = spop
AND sptld_spatial_delim_skey = sd_skey;
p_return_code := 0;
EXCEPTION WHEN NO_DATA_FOUND THEN
p_return_code := 4;
WHEN TOO_MANY_ROWS THEN
p_return_code := 5;
END;
END IF;
END IF;

```

Record: VEG TS 10 1

Field: STUMP PLOT RADIUS

Datatype: NUMBER

Position: 20

Length: 4

Optional: Y

Zero to Null:

<u>Record:</u> <u>VEG TS 10 1</u>	<u>Field:</u> <u>STUMP PLOT FULL</u>
	Datatype: CHAR
	Position: 24
	Length: 1
	Optional: Y
	Zero to Null:
Domain Name: VEG SAMPLE X INDICATOR	
<u>Value</u>	<u>Meaning</u>
-	No
X	Yes
<u>Record:</u> <u>VEG TS 10 1</u>	<u>Field:</u> <u>STUMP PLOT HALF</u>
	Datatype: CHAR
	Position: 25
	Length: 1
	Optional: Y
	Zero to Null:
Domain Name: VEG SAMPLE X INDICATOR	
<u>Value</u>	<u>Meaning</u>
-	No
X	Yes
<u>Record:</u> <u>VEG TS 10 1</u>	<u>Field:</u> <u>STUMP PLOT QRTR</u>
	Datatype: CHAR
	Position: 26
	Length: 1
	Optional: Y
	Zero to Null:
Domain Name: VEG SAMPLE X INDICATOR	
<u>Value</u>	<u>Meaning</u>
-	No
X	Yes
<u>Record:</u> <u>VEG TS 10 1</u>	<u>Field:</u> <u>STUMP PLOT BDRY</u>
	Datatype: CHAR
	Position: 27
	Length: 1
	Optional: Y
	Zero to Null:
Domain Name: VEG SAMPLE X INDICATOR	
<u>Value</u>	<u>Meaning</u>
-	No
X	Yes
<u>Record:</u> <u>VEG TS 10 1</u>	<u>Field:</u> <u>TOTAL PAGES</u>
	Datatype: NUMBER
	Position: 28
	Length: 2
	Optional: Y
	Zero to Null:
<u>Record:</u> <u>VEG TS 10 2</u>	<u>Field:</u> <u>RECORD ID</u>
	Datatype: NUMBER
	Position: 1
	Length: 3
	Optional: N
	Zero to Null:
<u>Record:</u> <u>VEG TS 10 2</u>	<u>Field:</u> <u>PAGE NUMBER</u>
	Datatype: NUMBER
	Position: 4
	Length: 2
	Optional: Y
	Zero to Null:

Record: VEG TS 10 2**Field:** SEQUENCE NUMBER**Datatype:** NUMBER**Position:** 6**Length:** 4**Optional:** N**Zero to Null:****Record:** VEG TS 10 2**Field:** SPECIES**Datatype:** CHAR**Position:** 10**Length:** 3**Optional:** N**Zero to Null:****Rule Name:** Veg 2000 Tree Species Short Code**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TS 10 2**Field:** TALLY TOTAL (10 - 30)**Datatype:** CHAR**Position:** 13**Length:** 2**Optional:** Y**Zero to Null:****Rule Name:** Field must be blank for NFI and CMI samples**PL/SQL Source Code:**

```

VarSampleType VARCHAR2(10);
BEGIN

```

```

p_return_code := 0;

```

```

VarSampletype := ffio.getc(1,'SAMPLE TYPE');

```

```

IF ( VarSampleType IN ('F','M' ) AND LTRIM(ffio.getc,'-') IS NOT NULL ) THEN

```

```

    p_return_code := 1;

```

```

    p_return_message := 'For sample types "F" and "M", the field '

```

```

        || ffio.get_current_item_name|| ' is not to be populated '

```

```

        || ' but a value of '||ffio.getc|| ' has been supplied';

```

```

END IF;

```

Rule Name: Veg 2000 Check That the Field is Present When the Species is Pre**PL/SQL Source Code:**

```

-- Jul 28, 2000 CAW - copied from rule 12454 to excluded rule for CMI and NFI
samples

```

```

VarSampletype VARCHAR2(10);

```

```

species VARCHAR2(10);

```

```

BEGIN

```

```

VarSampleType := ffio.getc(1, 'SAMPLE TYPE');

```

```

species := ffio.getc('SPECIES');

```

```

IF ( species IS NOT NULL AND ffio.getc IS NULL AND VarSampleType NOT IN ('F','M')

```

Record: VEG TS 10 2**Field:** TALLY TOTAL (31- 1.3)**Datatype:** NUMBER**Position:** 15**Optional:** Y**Length:** 2**Zero to Null:****Rule Name:** Veg 2000 Check That the Field is Present When the Species is Pre**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TS 10 2**Field:** TALLY TOTAL (> 1.3)**Datatype:** NUMBER**Position:** 17**Optional:** Y**Length:** 2**Zero to Null:****Rule Name:** Veg 2000 Check That the Field is Present When the Species is Pre**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TS 10 2**Field:** S1**Datatype:** CHAR**Position:** 19**Optional:** Y**Length:** 1**Zero to Null:****Record:** VEG TS 10 2**Field:** S2**Datatype:** CHAR**Position:** 20**Optional:** Y**Length:** 1**Zero to Null:****Record:** VEG TS 10 2**Field:** S3**Datatype:** CHAR**Position:** 21**Optional:** Y**Length:** 1**Zero to Null:****Record:** VEG TS 10 2**Field:** S4**Datatype:** CHAR**Position:** 22**Optional:** Y**Length:** 1**Zero to Null:****Record:** VEG TS 10 3**Field:** RECORD ID**Datatype:** NUMBER**Position:** 1**Optional:** N**Length:** 3**Zero to Null:****Record:** VEG TS 10 3**Field:** PAGE NUMBER**Datatype:** NUMBER**Position:** 4**Optional:** N**Length:** 2**Zero to Null:**

Record: VEG TS 10 3**Field:** SEQUENCE NUMBER**Datatype:** NUMBER**Position:** 6**Optional:** N**Length:** 4**Zero to Null:****Record:** VEG TS 10 3**Field:** SPECIES**Datatype:** CHAR**Position:** 10**Optional:** Y**Length:** 3**Zero to Null:****Rule Name:** Veg 2000 Lookup Stump Site Tree Cluster Element Specification**PL/SQL Source Code:**

```
--
--
-- Modifications:
--
-- Chet Watmough      Mar 8, 1998  Changed rule to reference plot radius to the
--                                recently added field STUMP PLOT RADIUS from
--                                the old PLOT RADIUS fields.
--
sd_skey NUMBER;
spop NUMBER;
baf NUMBER;
rad NUMBER;
rec NUMBER;
BEGIN
  p_return_code := 0;
  sd_skey := NULL;
  IF ( ffio.getc('SPECIES') IS NULL ) THEN
    RETURN;
  END IF;
  rec := ffio.get_previous_record_by_item ( '101');
  IF ( rec <= 0 ) THEN
    p_return_code := 6;
    p_return_message := ' There is no header record for Small tree, Stump' ||
      ' and Site tree data! ';
    RETURN;
  END IF;
  rad := ffio.getn(rec, 'STUMP PLOT RADIUS'); -- Changed for Veg 99
  IF ( rad IS NULL ) THEN
    p_return_code := 7;
    p_return_message := 'For Stump Data the radius is required but has not
been specified.';
    RETURN;
  END IF;
BEGIN
  SELECT spatial_delim_skey
  INTO sd_skey
  FROM spatial_delimiters
  WHERE ( radius = rad AND sptld_type = 'FXPLT' );
EXCEPTION WHEN NO_DATA_FOUND THEN
  p_return_code:= 1;
  p_return_message := ' The Fixed Plot with a Radius of ' ||rad||
    ' is not defined in the Spatial Delimiters table! ' ||
```

Rule Name: Veg 2000 Tree Species Short Code

PL/SQL Source Code: Rule code has already been printed.

Record: VEG TS 10 3

Field: FREQUENCY

Datatype: NUMBER

Position: 13

Length: 2

Optional: Y

Zero to Null:

Rule Name: Veg 2000 Check That the Field is Present When the Species is Pre

PL/SQL Source Code: Rule code has already been printed.

Record: VEG TS 10 3

Field: DIB

Datatype: NUMBER

Position: 15

Length: 4

Optional: Y

Zero to Null:

Rule Name: Veg 2000 Check That the Field is Present When the Species is Pre

PL/SQL Source Code: Rule code has already been printed.

Record: VEG TS 10 3

Field: LENGTH

Datatype: NUMBER

Position: 19

Length: 2

Optional: Y

Zero to Null:

Rule Name: Veg 2000 Check That the Field is Present When the Species is Pre

PL/SQL Source Code: Rule code has already been printed.

Rule Name: Veg 2000 TS 10 3 Stump Length

PL/SQL Source Code:

```
--  
-----  
--  
-- Veg TS 10 3 Stump Length  
--  
-----  
--
```

```
VarSrceDataStumpLength VARCHAR(2);  
VarSrceDataStumpLengthNum NUMBER;
```

```
BEGIN
```

```
-- initialize this function's return value, code, and message  
p_return_value := NULL;  
p_return_code := 0;  
p_return_message := NULL;
```

Record: VEG TS 10 3**Field:** SOUND %**Datatype:** CHAR**Position:** 21**Optional:** Y**Length:** 2**Zero to Null:****Domain Name:** VEG SAMPLE ENCRYPTED PCT**Value****Meaning**

--

100% (as well)

00

- 99

zero, ('0' is not allowed!)

1

- 99

—

100 % (true story!)

Rule Name: Veg 2000 Check That the Field is Present When the Species is Pre**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TS 10 3**Field:** WILD LIFE BARK CODE**Datatype:** NUMBER**Position:** 23**Optional:** Y**Length:** 1**Zero to Null:****Rule Name:** Veg 2000 Look Up Wildlife Feature Code and Return Feature Name**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 Check That the Field is Present When the Species is Pre**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TS 10 3**Field:** WILD LIFE WOOD CODE**Datatype:** NUMBER**Position:** 24**Optional:** Y**Length:** 1**Zero to Null:****Rule Name:** Veg 2000 Check That the Field is Present When the Species is Pre**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 Look Up Wildlife Feature Code and Return Feature Name**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TS 10 3**Field:** WILD LIFE USE CODE**Datatype:** CHAR**Position:** 25**Optional:** Y**Length:** 2**Zero to Null:****Rule Name:** Veg 2000 Lookup Wildlife Use Code**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 Check That the Field is Present When the Species is Pre**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TS 10 3**Field:** ROOT ROT SP.**Datatype:** CHAR**Position:** 27**Optional:** Y

Length:

3

Zero to Null:**Rule Name:** Veg 2000 TS 10 3 Root Rot Damage Agent Species**PL/SQL Source Code:**

```

spp VARCHAR2(10);
result VARCHAR2(10);
BEGIN
result := NULL;
p_return_code := 0;
p_return_message := null;

spp := REPLACE(ffio.getc, '-');

IF ( spp IS NULL ) THEN
result := NULL;
ELSIF ( NOT spp LIKE 'DR%' ) THEN
p_return_code := 1;
p_return_message := ' Invalid Root Rot Code of ***' || spp ||
'*** was recorded for Species ' || ffio.getc ( 'SPECIES' ) ||
'. It must be like DR..';
ELSE
--result := coms.get_damage_agent_skey ( spp );
BEGIN
SELECT code_argument
INTO result
FROM VIEW_THE_DAMAGE_AGENT_ST
WHERE LTRIM(RTRIM(UPPER(code_argument)))
= LTRIM(RTRIM(UPPER(spp)));
p_return_value := result;

EXCEPTION WHEN NO_DATA_FOUND THEN
p_return_code := 2;
p_return_message := ' Root Rot Code ***' || spp ||
'*** not found in Valid Damage Agent Codes for' ||
' Species ' || ffio.getc ( 'SPECIES' ) || '.';
p_return_value := null;
WHEN TOO_MANY_ROWS THEN
p_return_code := 3;
p_return_message :=
'There is multiple damage agent codes'
|| 'found -- Contact System Administrator';
p_return_value := null;

END;
END IF;

-- p_return_value := result;

```

Rule Name: Veg 2000 Check That the Field is Present When the Species is Pre**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TS 10 3**Field:** S5**Datatype:** CHAR**Position:**

Record: <u>VEG TS 10 3</u>	Length: 30	Optional: Y
	1	Zero to Null:
Record: <u>VEG TS 10 3</u>	Field: <u>S6</u>	
	Datatype: CHAR	
Record: <u>VEG TS 10 3</u>	Position: 31	Optional: Y
	Length: 1	Zero to Null:
Record: <u>VEG TS 10 3</u>	Field: <u>S7</u>	
	Datatype: CHAR	
Record: <u>VEG TS 10 3</u>	Position: 32	Optional: Y
	Length: 1	Zero to Null:
Record: <u>VEG TS 10 3</u>	Field: <u>S8</u>	
	Datatype: CHAR	
Record: <u>VEG TS 10 4</u>	Position: 33	Optional: Y
	Length: 1	Zero to Null:
Record: <u>VEG TS 10 4</u>	Field: <u>RECORD ID</u>	
	Datatype: NUMBER	
Record: <u>VEG TS 10 4</u>	Position: 1	Optional: N
	Length: 3	Zero to Null:
Record: <u>VEG TS 10 4</u>	Field: <u>PAGE NUMBER</u>	
	Datatype: NUMBER	
Record: <u>VEG TS 10 4</u>	Position: 4	Optional: N
	Length: 2	Zero to Null:
Record: <u>VEG TS 10 4</u>	Field: <u>SEQUENCE NUMBER</u>	
	Datatype: NUMBER	
Record: <u>VEG TS 10 4</u>	Position: 6	Optional: N
	Length: 4	Zero to Null:

Rule Name: Veg 2000 Determine Stump Tree Plot Start Bearing

PL/SQL Source Code:

```

typ VARCHAR2(10);
qual VARCHAR2(10);
plot_no NUMBER;
rec_no NUMBER;
BEGIN
p_return_code := 0;
rec_no := ffio.get_previous_record_by_item ( '101' );
IF ( rec_no < 0 ) THEN
p_return_code := 4;
p_return_message := 'Small Tree and Stump Header Card not found

```

Record: VEG TS 10 4**Field:** PLOT TYPE**Datatype:** CHAR**Position:** 10**Length:** 10**Optional:** N**Zero to Null:****Rule Name:** Veg 2000 Lookup Site Tree Top Height Cluster Element Specificati**PL/SQL Source Code:**

```

sd_skey NUMBER;
spop NUMBER;
rad NUMBER;
BEGIN
p_return_code := 0;
sd_skey := NULL;
rad := 5.64;
BEGIN
  SELECT spatial_delim_skey
  INTO sd_skey
  FROM spatial_delimiters
  WHERE ( radius = rad AND sptld_type = 'FXPLT' );
EXCEPTION WHEN NO_DATA_FOUND THEN
  p_return_code:= 1;
  p_return_message := ' The Plot Type Radius' ||
    ' is not defined in the Spatial Delimiters table! '
    || ' Contact the Data Resources Manager. ';
WHEN TOO_MANY_ROWS THEN
  p_return_code := 2;
  p_return_message := ' Too many values aquired from' ||
    ' Spatial Delimiters table! ' ||
    ' Contact the Data Resources Manager. ';
END;
IF ( sd_skey IS NOT NULL ) THEN
  SELECT MAX(sub_pop_skey)
  INTO spop
  FROM subpopulations,
  species_group_definitions
  WHERE popltn_elmnt_cd = 'TREE'
  AND min_dbh = 4.0
  AND min_tree_ht = 1.3
  AND subpopulations.sgdef_spp_group_skey = spp_group_skey
  AND group_cd = 'TREE';
  IF ( spop IS NULL ) THEN
    p_return_code := 3;
    p_return_message := ' Problems getting information from the ' ||
      'Subpopulations and Species Group Definitions ' ||
      ' tables! Contact the Data Resources Manager. ';
  ELSE
    BEGIN
      SELECT bd_spec_skey
      INTO p_return_value
      FROM biometric_design_specification
      WHERE subp_sub_pop_skey = spop
      AND sptld_spatial_delim_skey = sd_skey;
      p_return_code := 0;
    EXCEPTION WHEN NO_DATA_FOUND THEN
      p_return_code := 4;
      p_return_message := ' No information acquired from ' ||

```

Record: VEG TS 10 4**Field:** TREE NO.**Datatype:** NUMBER**Position:** 20**Length:** 3**Optional:** Y**Zero to Null:****Record:** VEG TS 10 4**Field:** TREE TYPE**Datatype:** CHAR**Position:** 23**Length:** 1**Optional:** N**Zero to Null:****Domain Name:** VEG TS RECORD 4 TYPE**Value****Meaning**

L	Leading Species
O	Other Top Height (Substitute) Tree - in plot
R	Random tree
S	Second species top height tree
T	Top height tree
V	Veteran Tree
X	Other Top Height (Substitute) Tree - out of plot
Z	Non-Standard Tree

Rule Name: Veg 2000 TS 10 4 Check Tree Type**PL/SQL Source Code:**

```
--
-----
--
-- Veg TS 10 4 Tree Type
--
-- This rule replaces the rule 11128 from VRI 97 format. In the VRI 97 format,
-- the tree types were fewer with combinations allowed - primarily an R tree
-- type, a Random tree, and a T tree type, a top height tree. With the changes
-- in the VRI 99 format, a further tree type has been introduced, L for leading
-- species and the type S change to Second Leading Species. To better support the
-- multiple tree types for a single tree, the database model was changed to
support
-- the 1-to-many relationship between the tree measurement and the tree type. With
-- this change, the primary reason for this rule, to produce a overloaded code
-- constaining multiple tree types, has been reduced to validating combinations of
-- tree types .
--
-- For VRI 99 the possible tree types are:
--
-- 'T' for Top Height, largest DBH in plot, regardless of species
-- 'L' for Leading Species, largest DBH of leading species in plot
-- 'S' for Second Species, largest DBH of second species in plot
-- 'O' for Other Site Tree, an alterntive tree within the plot
-- 'X' for eXtra Site Tree, an alternative tree outside of tyhe plot
-- 'R' randomly chosen tree.
-- 'V' veteran tree.
--
-- It is possible for single tree to be chosen for multiple reasons - a tree can
-- be the Top Height Tree, the Leading Species Tree and by chance also chosen as
```

Rule Name: Veg 2000 TS 10 4 TA 11 3 Determine Tree Selection Method

PL/SQL Source Code:

```

    cur_field    VARCHAR2(20);
    v_code       VARCHAR2(10);
--
--     MARCH,29 2000 Added tree type Z with a selection code of Nonstandard
--     June 27, 2002 CAW WO 979 - Added tree type V for Veteran trees.
--
BEGIN
    p_return_code := 0;
    cur_field := ffio.getc;
    IF ( cur_field = 'T' ) THEN
        v_code := 'TOP92';
    ELSIF ( cur_field = 'L' ) THEN
        v_code := 'LEADSP';
    ELSIF ( cur_field = 'S' ) THEN
        v_code := 'SECNDSP';
    ELSIF ( cur_field = 'R' ) THEN
        v_code := 'RANDOM';
    ELSIF ( cur_field = 'O' ) THEN
        v_code := 'INPTSUB';
    ELSIF ( cur_field = 'X' ) THEN
        v_code := 'OUTPTSUB';
    ELSIF ( cur_field = 'Z' ) THEN
        v_code := 'NONSTAND';
    ELSIF ( cur_field = 'V' ) THEN
        v_code := 'VETERAN';
    ELSE
        p_return_code:= 1;
        p_return_message := 'Unsupported Tree Type of '''||cur_field||''';
        RETURN;
    END IF;

    BEGIN
        SELECT selection_skey
        INTO    p_return_value
        FROM    selection_methods
        WHERE   method_cd = v_code;
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            p_return_code := 2;
            p_return_message := 'No Selection Method code found for
'''||cur_field||''';
        WHEN TOO_MANY_ROWS THEN
            p_return_code := 3;
            p_return_message := 'Too many Selection Method Codes found for
'''||cur_field||''';
    END;

```

Rule Name: Veg 2000 TA113&TS104 Check for more than one L type tree

PL/SQL Source Code:

```

-----
--
-- This rule will fire each 10 4 and 11 3 record.

```

Rule Name: Veg 2000 TS 10 4 Check plot type for random trees

PL/SQL Source Code:

```
tree_no NUMBER;
tree_type VARCHAR2(10);
plot_type VARCHAR2(10);
BEGIN
tree_no := ffio.getn('TREE NO.');
```

tree_type := ffio.getc;

plot_type := ffio.getc('PLOT TYPE');

IF (tree_type = 'R' AND lower(plot_type) = '5.64 m') THEN

p_return_code := 1;

p_return_message := 'On card 10, in the height and age information for tree number '

|| TO_CHAR(tree_no) ||

', you cannot have a random tree with a 5.64 m plot type.';

ELSE

p_return_code := 0;

END IF;

Record: VEG TS 10 4

Field: SPECIES

Datatype: CHAR

Position: 24

Length: 3

Optional: Y

Zero to Null:

Rule Name: Veg 2000 Check That the Field is Present When the Tree Number is

PL/SQL Source Code: Rule code has already been printed.

Rule Name: Veg 2000 Tree Species Short Code

PL/SQL Source Code: Rule code has already been printed.

Record: VEG TS 10 4

Field: CR. CL.

Datatype: CHAR

Position: 27

Length: 1

Optional: Y

Zero to Null:

Domain Name: VEG SAMPLE CROWN CLASS

Value

Meaning

-

none for dead trees

C

Codominant

D

Dominant

I

Intermediate

S

Suppressed

Rule Name: Veg 2000 Check That the Field is Present When the Tree Number is

PL/SQL Source Code: Rule code has already been printed.

Rule Name: Veg 2000 TS 10 4 Check That the Crown Class is Valid for the Tre

PL/SQL Source Code:

```
--
-----
-- Veg 1999 TS 10 4 Check That the Crown Class is Valid for the Tree Type
--
-----
--
-- Feb 14, 2001    RGH wo.938 also add condition that - is not excepted for tree
type L, T, S, O or X
--

VarSrceDataCrownClass VARCHAR2(1);
VarSrceDataPageNo NUMBER;
VarSrceDataTreeNo NUMBER;
VarSrceDataTreeType VARCHAR2(1);

BEGIN

-- initialize this function's return value, code, and message
p_return_value := NULL;
p_return_code := 0;
p_return_message := NULL;

-- get the page number
VarSrceDataPageNo := ffio.getn ( 'PAGE NUMBER' );

-- get the tree number from the source data record
VarSrceDataTreeNo := ffio.getn ( 'TREE NO.' );

-- get the tree type from the source data record
VarSrceDataTreeType := ffio.getc ( 'TREE TYPE' );

-- get the crown class from the source data record, which should be the
-- field that is currently being processed
VarSrceDataCrownClass := ffio.getc;

-- check if the tree type is L, T, S, O or X
IF ( LOWER ( VarSrceDataTreeType ) IN ( 'l', 't', 's', 'o', 'x' ) )
THEN
    -- the tree type is L, T, S, O or X

    -- these are the only tree types that are needed to check the crown class,
    -- and since the tree type is one of them, we can continue to do further
    -- checking

    -- continue processing
    NULL;
ELSE
    -- the tree type is not L, T, S, O or X
```

Record: VEG TS 10 4**Field:** BORING DIAMETER**Datatype:** NUMBER**Position:** 28**Optional:** Y**Length:** 4**Zero to Null:****Rule Name:** Veg 2000 Check That the Field is Present When the Tree Number is**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TS 10 4**Field:** TREE LENGTH**Datatype:** NUMBER**Position:** 32**Optional:** Y**Length:** 4**Zero to Null:****Rule Name:** Veg 2000 Check That the Field is Present When the Tree Number is**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TS 10 4**Field:** SUITABLE HEIGHT TREE**Datatype:** CHAR**Position:** 36**Optional:** N**Length:** 1**Zero to Null:** N**Domain Name:** INDICATOR**Value**

N

Y

Meaning

No (False)

Yes (True)

Record: VEG TS 10 4**Field:** BARK THICKNESS**Datatype:** NUMBER**Position:** 37**Optional:** Y**Length:** 3**Zero to Null:****Rule Name:** Veg 2000 Validate Bark Thickness Against Tree Diameter**PL/SQL Source Code:**

```
--
-- June 14, 2000 RGH - wo 811, add Plot Type indicator N,S,W or E in error message
--
```

```
-----
--
```

```
diam NUMBER;
bark NUMBER;
v_page_no VARCHAR2(10);
curr_rec   VARCHAR2(20);
v_plot_rec VARCHAR2(10);
v_plot_type VARCHAR2(10);
v_plot_id  NUMBER;
```

```
BEGIN
```

```
curr_rec := ffio.getc('RECORD ID');
v_page_no := ffio.getc('PAGE NUMBER');
v_plot_type := NULL;
```

Rule Name: Veg 2000 Non-NFI Samples: Check That the Field is Present When t
PL/SQL Source Code: Rule code has already been printed.

Record: VEG TS 10 4

Field: BORED HEIGHT

Datatype: NUMBER

Position: 40

Optional: Y

Length: 2

Zero to Null:

Domain Name: MEASMT HEIGHT

Value

0.00 - 3.00

Meaning

Rule Name: Veg 2000 Check That the Field is Present When the Tree Number is
PL/SQL Source Code: Rule code has already been printed.

Record: VEG TS 10 4

Field: MEASURE CODE

Datatype: CHAR

Position: 42

Optional: Y

Length: 3

Zero to Null:

Domain Name: VEG SAMPLE NOT BORED REASON

Value

CRC

OUT

PHY

ROT

WHO

Meaning

Physiological age is recorded or calculated on tha

Increment borer cannot reach centre (tree too larg

surrogate age data from an out of plot tree

Physiological age is recorded or calculated on tha

Core of tree rotten or missing

Whorl count

Rule Name: Veg 2000 Check That the Field is Present When the Tree Number is
PL/SQL Source Code: Rule code has already been printed.

Record: VEG TS 10 4

Field: OFFICE BORED HEIGHT AGE

Datatype: NUMBER

Position: 45

Optional: Y

Length: 4

Zero to Null:

Rule Name: Veg 2000 TS 10 4 TA 11 3 Validate Bored Measure Code and Bored H
PL/SQL Source Code:

--

--

-- Veg 1999 TS 10 4 TA 11 3 Validate Bored Measure Code and Bored Height Age

--

--

-- Modifications

--

-- Aug 11, 1999 TMC - wo 747, remove the check for ROT and CRC against a non-
 -- blank field bored height age

Record: VEG TS 10 4**Field:** GROWTH 5 YR.**Datatype:** CHAR**Position:** 49**Length:** 2**Optional:** Y**Zero to Null:****Rule Name:** Veg 2000 TS 10 4 TA 11 3 Validate Tree Growth Ring 5 10 20 Data**PL/SQL Source Code:**

```
--
-----
--
-- Veg 1999 TS 10 4 TA 11 3 Validate Tree Growth Ring 5 10 20 Data
--
-- Modifications
--
-- Oct 7, 1999 TMC - wo 737, check that the maximum radial increment year
--                  entered is less than or equal to the office bored height
--                  age
-- Oct 18, 1999 CAW - a NULL Office Bored Height Age was always causing an error.
--                  Used NVL(,999) on the comparison. Prorate and field age is
--                  now being checked against the max inc year entered.
-- Nov 29, 1999 CAW - removing check on Field Bored Height age because it's dark
--                  and cloudy and cold and it's hard (the fog set in as well).
-- June 12, 2000 RGH - updated error message to add what plot.
--
-- NOV 28, 2000 RGH - System error corrected error 12 and 13 text message too long
--
-- NOV 23, 2000 RGH - wo. 925 Added the case for a Measure code of PHY that the
prorate ring
-- count can be less than the oldest radial increment year if it is 20.
-----
--
```

```
bar_thic NUMBER;
bor_dia NUMBER;
VarMaxRadIncrYrEntered NUMBER;
TextVarMaxRadIncrYrEntered VARCHAR2(10);
radius NUMBER;
VarSrceDataBoredHtAge NUMBER;
VarSrceDataGrowth5Yr NUMBER;
VarSrceDataGrowth10Yr NUMBER;
VarSrceDataGrowth20Yr NUMBER;
VarSrceDataPageNo NUMBER;
VarSrceDataTreeNo NUMBER;
VarSrceDataMeasureCode VARCHAR2(10);
VarSrceProrateAge NUMBER;
VarSrceFieldAge NUMBER;
curr_rec VARCHAR2(10);
v_plot_rec VARCHAR2(10);
v_plot_type VARCHAR2(10);
v_plot_id NUMBER;
```

BEGIN

Record: VEG TS 10 4**Field:** GROWTH 10 YR.**Datatype:** CHAR**Position:** 51**Optional:** Y**Length:** 2**Zero to Null:****Record:** VEG TS 10 4**Field:** GROWTH 20 YR.**Datatype:** CHAR**Position:** 53**Optional:** Y**Length:** 3**Zero to Null:****Record:** VEG TS 10 4**Field:** PRORATE LENGTH**Datatype:** NUMBER**Position:** 56**Optional:** Y**Length:** 4**Zero to Null:****Rule Name:** Veg 2000 Validate Presence of Prorate Data**PL/SQL Source Code:**

```
--
-- June 15, 2000 RGH - wo 811, add Plot Type indicator N,S,W or E in error message
--
-----
--
curr_rec VARCHAR2(10);
v_pg_no  NUMBER;
v_plot_rec VARCHAR2(10);
v_plot_type VARCHAR2(10);
v_plot_id  NUMBER;

BEGIN

p_return_code := 0;
curr_rec := ffio.getc('RECORD ID');
v_plot_type := NULL;
v_plot_rec := NULL;
v_plot_id := NULL;

-- get the page number

v_pg_no := ffio.getn ( 'PAGE NUMBER' );

-- set up which plot we have come in on

IF ( curr_rec = '104' ) THEN
    v_plot_rec := '101';
ELSIF ( curr_rec = '113' ) THEN
    v_plot_rec := '111';
ELSE
    RETURN;
END IF;

v_plot_id := ffio.get_previous_record_by_item(v_plot_rec, 'PAGE NUMBER',v_pg_no);
v_plot_type := ffio.getc ( v_plot_id, 'PL' );
```

Record: VEG TS 10 4**Field:** PRORATE RING COUNT**Datatype:** NUMBER**Position:** 60**Optional:** Y**Length:** 3**Zero to Null:****Rule Name:** Veg 2000 Validate Presence of Prorate Data**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TS 10 4**Field:** DIRECT AGE CORR.**Datatype:** NUMBER**Position:** 63**Optional:** Y**Length:** 2**Zero to Null:****Record:** VEG TS 10 4**Field:** TOTAL AGE**Datatype:** NUMBER**Position:** 65**Optional:** Y**Length:** 3**Zero to Null:****Record:** VEG TS 10 4**Field:** PHY AGE**Datatype:** NUMBER**Position:** 68**Optional:** Y**Length:** 3**Zero to Null:****Rule Name:** Veg 2000 TS 10 4 TA 11 3 Validate at Least One Age Supplied**PL/SQL Source Code:**

```
--
-- MAY 25,2000 RGH wo811; put indicator for Page no. and plot type in error
message
--
```

```
obh_age NUMBER;
fbh_age NUMBER;
r_count    NUMBER;
total_age  NUMBER;
phy_age    NUMBER;
v_page_no  NUMBER;
curr_rec   VARCHAR2(20);
v_plot_rec VARCHAR2(10);
v_plot_type VARCHAR2(10);
v_plot_id  NUMBER;
```

BEGIN

BEGIN

```
obh_age := nvl(ffio.getn('OFFICE BORED HEIGHT AGE'),0);
fbh_age := nvl(ffio.getn('FIELD BORED HEIGHT AGE'),0);
r_count := nvl ( ffio.getn ( 'PRORATE RING COUNT' ), 0 );
total_age := nvl(ffio.getn('TOTAL AGE'),0);
phy_age := nvl(ffio.getn('PHY AGE'),0);
curr_rec := ffio.getc('RECORD ID');
```

Record: VEG TS 10 4**Field:** FIELD BORED HEIGHT AGE**Datatype:** NUMBER**Position:** 71**Optional:** Y**Length:** 4**Zero to Null:** N**Record:** VEG TS 10 4**Field:** SUITABLE AGE TREE**Datatype:** CHAR**Position:** 75**Optional:** N**Length:** 1**Zero to Null:****Domain Name:** INDICATOR**Value**

N

Y

Meaning

No (False)

Yes (True)

Record: VEG TS 10 5**Field:** RECORD ID**Datatype:** NUMBER**Position:** 1**Optional:** N**Length:** 3**Zero to Null:****Record:** VEG TS 10 5**Field:** PAGE NUMBER**Datatype:** NUMBER**Position:** 4**Optional:** N**Length:** 2**Zero to Null:****Record:** VEG TS 10 5**Field:** SEQUENCE NUMBER**Datatype:** NUMBER**Position:** 6**Optional:** N**Length:** 4**Zero to Null:****Record:** VEG TS 10 5**Field:** NOTES**Datatype:** CHAR**Position:** 10**Optional:** Y**Length:** 256**Zero to Null:****Record:** VEG TA 11 1**Field:** RECORD ID**Datatype:** NUMBER**Position:** 1**Optional:** N**Length:** 3**Zero to Null:****Record:** VEG TA 11 1**Field:** PAGE NUMBER**Datatype:** NUMBER**Position:** 4**Optional:** N**Length:** 2**Zero to Null:****Rule Name:** Veg 2000 Validate Page Number**PL/SQL Source Code:** Rule code has already been printed.

Record: VEG TA 11 1**Field:** SEQUENCE NUMBER**Datatype:** NUMBER**Position:** 6**Optional:** N**Length:** 4**Zero to Null:****Record:** VEG TA 11 1**Field:** PL**Datatype:** CHAR**Position:** 10**Optional:** N**Length:** 1**Zero to Null:****Domain Name:** VEG SAMPLE PL - PLOT NOValueMeaning

E

I

M

N

S

W

Record: VEG TA 11 1**Field:** VARIABLE PLOT B.A.F.**Datatype:** NUMBER**Position:** 11**Optional:** Y**Length:** 4**Zero to Null:****Domain Name:** VEG SAMPLE BASAL AREA FTRValueMeaning

02.97 - 03.03

3.00 m**2/ha Prism

03.96 - 04.04

4.00 m**2/ha Prism Size or Relaskop Value

04.95 - 05.05

5.00 m**2/ha Prism

05.01 - 05.11

5.06 m**2/ha Relascop Value

06.19 - 06.31

6.25 m**2/ha Relascop Value

06.93 - 07.07

7.0 m**2/ha Prism

08.91 - 09.09

9.00 m**2/ha Prism Size or Relaskop Value

11.88 - 12.12

12.00 m**2/ha Prism

12.13 - 12.37

12.25 m**2/ha Relascop Value

15.84 - 16.16

16.00 m**2/ha Prism Size or Relascop Value

18.80 - 20.20

20.00 m**2/ha Prism

20.05 - 20.45

20.25 m**2/ha Relascop Value

24.75 - 25.25

25.00 m**2/ha Prism size or Relaskop Value

Rule Name: Veg 2000 TD 08 1 TA 11 1 Plot Type (Variable or Fixed)**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 TA 11 1 Validate BAF and Radius for Consistency**PL/SQL Source Code:**

--

-- MAY 26,2000 RGH wo811; put indicator for Page no. and plot type in error message

--

baf1 NUMBER;

diopter1 NUMBER;

prf1 NUMBER;

Record: <u>VEG TA 11 1</u>	Field: <u>VARIABLE PLOT FULL</u>
	Datatype: CHAR
	Position: 15
	Length: 1
	Optional: Y
	Zero to Null:
Domain Name: VEG SAMPLE X INDICATOR	
<u>Value</u>	<u>Meaning</u>
-	No
X	Yes
Record: <u>VEG TA 11 1</u>	Field: <u>VARIABLE PLOT HALF</u>
	Datatype: CHAR
	Position: 16
	Length: 1
	Optional: Y
	Zero to Null:
Domain Name: VEG SAMPLE X INDICATOR	
<u>Value</u>	<u>Meaning</u>
-	No
X	Yes
Record: <u>VEG TA 11 1</u>	Field: <u>VARIABLE PLOT QTR</u>
	Datatype: CHAR
	Position: 17
	Length: 1
	Optional: Y
	Zero to Null:
Domain Name: VEG SAMPLE X INDICATOR	
<u>Value</u>	<u>Meaning</u>
-	No
X	Yes
Record: <u>VEG TA 11 1</u>	Field: <u>VARIABLE PLOT BDRY</u>
	Datatype: CHAR
	Position: 18
	Length: 1
	Optional: Y
	Zero to Null:
Domain Name: VEG SAMPLE X INDICATOR	
<u>Value</u>	<u>Meaning</u>
-	No
X	Yes
Record: <u>VEG TA 11 1</u>	Field: <u>VARIABLE PLOT SPLIT</u>
	Datatype: CHAR
	Position: 19
	Length: 1
	Optional: Y
	Zero to Null:
Domain Name: VEG SAMPLE X INDICATOR	
<u>Value</u>	<u>Meaning</u>
-	No
X	Yes
Record: <u>VEG TA 11 1</u>	Field: <u>PRF</u>
	Datatype: NUMBER
	Position: 20
	Length: 5
	Optional: Y
	Zero to Null:
Record: <u>VEG TA 11 1</u>	Field: <u>DIOPTER</u>

Datatype: NUMBER**Position:** 25**Optional:** Y**Length:** 5**Zero to Null:****Record:** VEG TA 11 1**Field:** AZIMUTH BDRY/SPLIT**Datatype:** NUMBER**Position:** 30**Optional:** Y**Length:** 3**Zero to Null:****Domain Name:** AZIMUTH BRG 0 TO 359ValueMeaning

0

- 359

Rule Name: Veg 2000 Validate Azimuth against Plot Type**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TA 11 1**Field:** FIXED PLOT RADIUS**Datatype:** NUMBER**Position:** 33**Optional:** Y**Length:** 4**Zero to Null:****Record:** VEG TA 11 1**Field:** FIXED PLOT FULL**Datatype:** CHAR**Position:** 37**Optional:** Y**Length:** 1**Zero to Null:****Domain Name:** VEG SAMPLE X INDICATORValueMeaning

-

No

X

Yes

Record: VEG TA 11 1**Field:** FIXED PLOT HALF**Datatype:** CHAR**Position:** 38**Optional:** Y**Length:** 1**Zero to Null:****Domain Name:** VEG SAMPLE X INDICATORValueMeaning

-

No

X

Yes

Record: VEG TA 11 1**Field:** FIXED PLOT QRTR**Datatype:** CHAR**Position:** 39**Optional:** Y**Length:** 1**Zero to Null:****Domain Name:** VEG SAMPLE X INDICATORValueMeaning

-

No

X

Yes

Record: VEG TA 11 1**Field:** FIXED PLOT BDRY**Datatype:** CHAR**Position:**

Domain Name:	VEG SAMPLE X INDICATOR	Length:	40	Optional:	Y
Value		Meaning	1	Zero to Null:	
-		No			
X		Yes			

Record: VEG TA 11 1 **Field:** FIXED PLOT SPLIT

Datatype: CHAR

Position: 41

Length: 1

Optional: Y

Zero to Null:

Domain Name: VEG SAMPLE X INDICATOR

Value

Meaning

-

No

X

Yes

Rule Name: Veg 2000 Determine Tree Details Partial Start Bearing

PL/SQL Source Code: Rule code has already been printed.

Rule Name: Veg 2000 Determine Partial End Bearing

PL/SQL Source Code: Rule code has already been printed.

Rule Name: Veg 2000 TD 08 1 TA 11 1 Tree Cluster Element Specification

PL/SQL Source Code: Rule code has already been printed.

Record: VEG TA 11 1 **Field:** TOTAL PAGES

Datatype: NUMBER

Position: 42

Length: 2

Optional: N

Zero to Null:

Record: VEG TA 11 1 **Field:** AUXILIARY PLOT PIN AZIMUTH

Datatype: NUMBER

Position: 44

Length: 3

Optional: Y

Zero to Null:

Rule Name: Veg 2000 Convert Auxiliary Plot Azimuth to Azimuth from Plot Cen

PL/SQL Source Code:

--

-- MAY 26,2000 RGH wo811; put indicator for Page no. and plot type in error message

--

```
ref_x NUMBER;
ref_y NUMBER;
pin_x NUMBER;
pin_y NUMBER;
hdist NUMBER;
brg NUMBER;
pl VARCHAR2(5);
```

Record: VEG TA 11 1**Field:** AUXILIARY PLOT PIN DISTANCE**Datatype:** NUMBER**Position:** 47**Optional:** N**Length:** 4**Zero to Null:****Rule Name:** Veg 2000 Convert Auxiliary Plot Azimuth to Azimuth from Plot Cen**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TA 11 3**Field:** RECORD ID**Datatype:** NUMBER**Position:** 1**Optional:** N**Length:** 3**Zero to Null:****Record:** VEG TA 11 3**Field:** PAGE NUMBER**Datatype:** NUMBER**Position:** 4**Optional:** N**Length:** 2**Zero to Null:****Record:** VEG TA 11 3**Field:** SEQUENCE NUMBER**Datatype:** NUMBER**Position:** 6**Optional:** N**Length:** 4**Zero to Null:****Record:** VEG TA 11 3**Field:** PLOT TYPE**Datatype:** CHAR**Position:** 10**Optional:** N**Length:** 10**Zero to Null:****Record:** VEG TA 11 3**Field:** TREE NO.**Datatype:** NUMBER**Position:** 20**Optional:** Y**Length:** 3**Zero to Null:****Rule Name:** Veg 2000 Validate Auxiliary Top Height Trees Against Live Standi**PL/SQL Source Code:**

--

-- June 13, 2000 RGH wo.811 add Plot location for error messages

--

```

-----
tree_num NUMBER;
species VARCHAR2(24);
dbh NUMBER;
bored NUMBER;
c_type VARCHAR2(10);
rec_no NUMBER;
p_no NUMBER;
v_plot_rec VARCHAR2(10);
v_plot_type VARCHAR2(10);

```

Rule Name: Veg 2000 TA 11 3 Validate Top Height Tree and Enhanced Tree Info
PL/SQL Source Code:

```
--
-----
-- Veg 1999 TA 11 3 Validate Top Height Tree and Enhanced Tree Information
--
-- Modifications
--
-- Aug 18, 1999 TMC - wo 748, changed the tree number variable's length from 2
--                   to 3, which is its length on the TA 11 3 card
--
-- Nov 15, 1999 CAW - rule is failing for trees that are not fully sampled in
--                   the 114 cards. It is not clear if Top Height trees must be
--                   fully sampled in the Aux. plots, so until we're sure the
--                   partially sampled (count) trees don't have to match crown
--                   class and tree length.
--
-- May 08, 2001   JPK   wo811; put indicator for plot type in error messages
--
-- Oct 25, 2002 CAW - This rule limits the comparison to type 'T' trees, but these
--                   values should match for all tree types. Removing Top Height
--                   trees restriction.
--
-- Jan 22, 2003 CAW - Only require the DBH and the Boring Diameter to match for
--                   trees that are bored at 1.3m
--
-----
--

top_ht_page_no  VARCHAR2(2);
top_ht_tree_no  VARCHAR2(3);
top_ht_tree_type VARCHAR2(1);
top_ht_cr_cl    VARCHAR2(1);
top_ht_species  VARCHAR2(3);
top_ht_tree_length NUMBER(4);
top_ht_boring_diameter NUMBER(4);
top_ht_bored_height NUMBER;
enhanced_rec_id  VARCHAR2(3) := '114';
enhanced_cr_cl   VARCHAR2(1);
enhanced_species VARCHAR2(3);
enhanced_tree_length NUMBER(4);
enhanced_boring_diameter NUMBER(4);
rec_id          NUMBER;

v_plot_type     VARCHAR2(10);
v_plot_rec      VARCHAR2(10);
v_plot_id       NUMBER;

v_err_point     NUMBER := 0;
BEGIN
BEGIN
```

Record: VEG TA 11 3**Field:** TREE TYPE**Datatype:** CHAR**Position:** 23**Length:** 1**Optional:** N**Zero to Null:****Domain Name:** VEG TS RECORD 4 TYPE**Value****Meaning**

L	Leading Species
O	Other Top Height (Substitute) Tree - in plot
R	Random tree
S	Second species top height tree
T	Top height tree
V	Veteran Tree
X	Other Top Height (Substitute) Tree - out of plot
Z	Non-Standard Tree

Rule Name: Veg 2000 TA113&TS104 Check for more than one L type tree**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TA 11 3**Field:** SPECIES**Datatype:** CHAR**Position:** 24**Length:** 3**Optional:** Y**Zero to Null:****Rule Name:** Veg 2000 Check That the Field is Present When the Tree Number is**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 Tree Species Short Code**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TA 11 3**Field:** CR. CL.**Datatype:** CHAR**Position:** 27**Length:** 1**Optional:** Y**Zero to Null:****Domain Name:** VEG SAMPLE CROWN CLASS**Value****Meaning**

-	none for dead trees
C	Codominant
D	Dominant
I	Intermediate
S	Suppressed

Rule Name: Veg 2000 Check That the Field is Present When the Tree Number is**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 TA 11 3 Validate Crown Class is Valid for Tree Type**PL/SQL Source Code:**

```

-----
--
-- Veg TA 11 4 Check That the Crown Class is Valid for the Tree Type
--
-----

```

Record: VEG TA 11 3**Field:** BORING DIAMETER**Datatype:** NUMBER**Position:** 28**Length:** 4**Optional:** Y**Zero to Null:****Record:** VEG TA 11 3**Field:** TREE LENGTH**Datatype:** NUMBER**Position:** 32**Length:** 4**Optional:** Y**Zero to Null:****Domain Name:** VEG SAMPLE TREE LENGTH > 1.3MValue

1.3 - 99.9

Meaning**Rule Name:** Veg 2000 Check That the Field is Present When the Tree Number is**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TA 11 3**Field:** SUITABLE HEIGHT TREE**Datatype:** CHAR**Position:** 36**Length:** 1**Optional:** N**Zero to Null:** N**Domain Name:** INDICATORValue

N

Y

Meaning

No (False)

Yes (True)

Record: VEG TA 11 3**Field:** BARK THICKNESS**Datatype:** NUMBER**Position:** 37**Length:** 3**Optional:** Y**Zero to Null:****Rule Name:** Veg 2000 Validate Bark Thickness Against Tree Diameter**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TA 11 3**Field:** BORED HEIGHT**Datatype:** NUMBER**Position:** 40**Length:** 2**Optional:** Y**Zero to Null:****Domain Name:** MEASMT HEIGHTValue

0.00 - 3.00

Meaning**Rule Name:** Veg 2000 Check That the Field is Present When the Tree Number is**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TA 11 3**Field:** MEASURE CODE**Datatype:** CHAR**Position:** 42**Length:** 3**Optional:** Y**Zero to Null:****Domain Name:** VEG SAMPLE NOT BORED REASONValueMeaning

---	Physiological age is recorded or calculated on tha
CRC	Increment borer cannot reach centre (tree too larg
OUT	surrogate age data from an out of plot tree
PHY	Physiological age is recorded or calculated on tha
ROT	Core of tree rotten or missing
WHO	Whorl count

Rule Name: Veg 2000 Check That the Field is Present When the Tree Number is

PL/SQL Source Code: Rule code has already been printed.

Rule Name: Veg 2000 TA 11 3 Validate Bored Measure Code and Bored Height

PL/SQL Source Code:

```
--
-- 392 data format
-----
--
-- Veg 2000 TA 11 3 Validate Bored Measure Code and Bored Height
--
-- Modifications
--
--Sept. 21, 1999 RGH - new rule to check Bored measure code for Bored Height
-- 0 or greater than zeros bark thickness and Boring diameter.
-- If Code is OUT of WHO and the Bored Height is 0, then Bark Thickness
-- and Boring Diameter anre Blank.
-- Rule 11,666 before this checks the fields Bark Thickness and Boring Diameter
-- that they are not Null. It must be removed from these fields
-- and this new rule triggered.
--
-- April 10, 2000 LME - changed > 0 to >= 0 for Bored Height on first three
IF/ELSIF conditions.
-- Now we don't need the last three ELSIF conditions so they are commented
-- out.
--
-- May 2, 2000 LME - uncommented and changed order of elsif statements
--
-- May 02,2001 JPK wo811; put indicator for plot type in error messages
-----
--

VarSrceDataMeasureCode VARCHAR2(10);
VarSrceDataPageNo NUMBER;
VarSrceDataTotalAge NUMBER;
VarSrceDataTreeNo NUMBER;
VarSrceDataBoredDiam NUMBER;
VarSrceDataBoredHeight NUMBER;
VarSrceDataBarkThick NUMBER;

v_plot_type VARCHAR2(10);
v_plot_rec VARCHAR2(10);
v_plot_id NUMBER;

debug_flag BOOLEAN;

BEGIN
```

Record: VEG TA 11 3**Field:** OFFICE BORED HEIGHT AGE**Datatype:** NUMBER**Position:** 45**Optional:** Y**Length:** 4**Zero to Null:****Rule Name:** Veg 2000 TS 10 4 TA 11 3 Validate Bored Measure Code and Bored H**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TA 11 3**Field:** GROWTH 5 YR.**Datatype:** NUMBER**Position:** 49**Optional:** Y**Length:** 2**Zero to Null:****Domain Name:** RAD INC**Value**

1 - 450

Meaning**Rule Name:** Veg 2000 TS 10 4 TA 11 3 Validate Tree Growth Ring 5 10 20 Data**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TA 11 3**Field:** GROWTH 10 YR.**Datatype:** NUMBER**Position:** 51**Optional:** Y**Length:** 2**Zero to Null:****Domain Name:** RAD INC**Value**

1 - 450

Meaning**Record:** VEG TA 11 3**Field:** GROWTH 20 YR.**Datatype:** NUMBER**Position:** 53**Optional:** Y**Length:** 3**Zero to Null:****Domain Name:** RAD INC**Value**

1 - 450

Meaning**Record:** VEG TA 11 3**Field:** PRORATE LENGTH**Datatype:** NUMBER**Position:** 56**Optional:** Y**Length:** 4**Zero to Null:****Rule Name:** Veg 2000 Validate Presence of Prorate Data**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TA 11 3**Field:** PRORATE RING COUNT**Datatype:** NUMBER**Position:** 60**Optional:** Y**Length:** 3**Zero to Null:**

Rule Name: Veg 2000 Validate Presence of Prorate Data

PL/SQL Source Code: Rule code has already been printed.

Record: VEG TA 11 3

Field: DIRECT AGE CORR.

Datatype: NUMBER

Position: 63

Optional: Y

Length: 2

Zero to Null:

Record: VEG TA 11 3

Field: TOTAL AGE

Datatype: NUMBER

Position: 65

Optional: Y

Length: 3

Zero to Null:

Record: VEG TA 11 3

Field: PHY AGE

Datatype: NUMBER

Position: 68

Optional: Y

Length: 3

Zero to Null:

Rule Name: Veg 2000 TS 10 4 TA 11 3 Validate at Least One Age Supplied

PL/SQL Source Code: Rule code has already been printed.

Record: VEG TA 11 3

Field: FIELD BORED HEIGHT AGE

Datatype: NUMBER

Position: 71

Optional: Y

Length: 4

Zero to Null: N

Record: VEG TA 11 3

Field: SUITABLE AGE TREE

Datatype: CHAR

Position: 75

Optional: N

Length: 1

Zero to Null: N

Domain Name: INDICATOR

Value

N

Y

Meaning

No (False)

Yes (True)

Record: VEG TA 11 4

Field: RECORD ID

Datatype: NUMBER

Position: 1

Optional: N

Length: 3

Zero to Null:

Record: VEG TA 11 4

Field: PAGE NUMBER

Datatype: NUMBER

Position: 4

Optional: N

Length: 2

Zero to Null:

Record: VEG TA 11 4

Field: SEQUENCE NUMBER

Datatype: NUMBER**Position:** 6**Length:** 4**Optional:** N**Zero to Null:****Record:** VEG TA 11 4**Field:** TREE NUMBER**Datatype:** NUMBER**Position:** 10**Length:** 3**Optional:** N**Zero to Null:****Rule Name:** Veg 2000 Validate Auxiliary Measured Trees Against Live Standing**PL/SQL Source Code:**

```
-- July 04,2001    JPK  wo811; put indicator for plot type in error messages
```

```
tree_num  NUMBER;
species   VARCHAR2(10);
dbh       NUMBER;
rec_no    NUMBER;
p_no      NUMBER;
rec_id    VARCHAR2(10);
```

```
VarSrceDataOrigRecId  VARCHAR2(10);
VarSrceDataPageNo     NUMBER;
v_plot_type           VARCHAR2(10);
v_plot_rec            VARCHAR2(10);
v_plot_id             NUMBER;
```

```
BEGIN
```

```
    p_return_code := 0;
```

```
    -- initialize the variables to be used for deriving the plot type
```

```
    v_plot_type := NULL;
```

```
    v_plot_rec := NULL;
```

```
    v_plot_id := NULL;
```

```
    -- get the record ID of the current record;
```

```
    VarSrceDataOrigRecId := ffio.getc ( 'RECORD ID' );
```

```
    -- get the page number value
```

```
    VarSrceDataPageNo := ffio.getn ( 'PAGE NUMBER' );
```

```
    -- get the plot type (I,N,S,E,W)
```

```
    v_plot_rec := '111';
```

```
    v_plot_id := ffio.get_previous_record_by_item(v_plot_rec, 'PAGE
NUMBER',VarSrceDataPageNo);
```

```
    IF v_plot_id > 0 THEN
```

```
        v_plot_type := ffio.getc(v_plot_id, 'PL');
```

```
    END IF;
```

```
    BEGIN
```

```
    tree_num := ffio.getn('TREE NUMBER');
```

```
    EXCEPTION
```

Rule Name: Veg 2000 TD 08 2 TA 11 4 Check for the Presence of Tree Loss Ind

PL/SQL Source Code: Rule code has already been printed.

Rule Name: Veg 2000 Check for Unwanted TA 11 4/TD 08 2 Duplicate Tree Numbe

PL/SQL Source Code: Rule code has already been printed.

Record: VEG TA 11 4

Field: SPECIES

Datatype: CHAR

Position: 13

Length: 3

Optional: Y

Zero to Null:

Rule Name: Veg 2000 Check That the Field is Present When the Tree Number is

PL/SQL Source Code: Rule code has already been printed.

Rule Name: Veg 2000 Tree Species Short Code

PL/SQL Source Code: Rule code has already been printed.

Record: VEG TA 11 4

Field: LIVE/DEAD

Datatype: CHAR

Position: 16

Length: 1

Optional: Y

Zero to Null:

Domain Name: VEG SAMPLE EXTANT CODE

Value

D

L

Meaning

Dead

Live

Rule Name: Veg 2000 Validate Auxiliary Plot Tree Extant

PL/SQL Source Code:

```
plot_no VARCHAR2(5);
plot_type VARCHAR2(5);
extant VARCHAR2(5);
rec_num NUMBER;
```

```
VarSrceDataOrigRecId VARCHAR2(10);
VarSrceDataPageNo NUMBER;
v_plot_type VARCHAR2(10);
v_plot_rec VARCHAR2(10);
v_plot_id NUMBER;
```

BEGIN

```
-- initialize the variables to be used for deriving the plot type
v_plot_type := NULL;
v_plot_rec := NULL;
v_plot_id := NULL;

-- get the record ID of the current record;
VarSrceDataOrigRecId := ffio.getc ( 'RECORD ID' );

-- get the page number value
VarSrceDataPageNo := ffio.getn ( 'PAGE NUMBER' );
```


Record: VEG TA 11 4**Field:** REM BARK**Datatype:** CHAR**Position:** 23**Length:** 2**Optional:** Y**Zero to Null:****Domain Name:** VEG SAMPLE ENCRYPTED PCT**Value****Meaning**

--

100% (as well)

00

- 99

zero, ('0' is not allowed!)

1

- 99

—

100 % (true story!)

Rule Name: Veg 2000 Check that all data is present for an enhanced tree**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Validate Remaining Bark Percent against Wildlife Bark Code for T**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TA 11 4**Field:** TREE LENGTH**Datatype:** NUMBER**Position:** 25**Length:** 3**Optional:** Y**Zero to Null:** Y**Domain Name:** VEG SAMPLE TREE LENGTH > 1.3M**Value****Meaning**

1.3

- 99.9

Rule Name: Veg 2000 TA 11 4 Check for the Presence of Tree Data for a Live**PL/SQL Source Code:**

--

--

-- Veg 1999 TA 11 4 Check for the Presence of Tree Data for a Live Tree Species

--

--

-- Modifications

--

-- Dec 8, 1999 TMC - wo 794, re-write the logic that determines if a species
 -- is entered once either as a live tree in the IPC on any
 -- TD 08 card or as an enhanced auxiliary plot live tree on
 -- any TA 11 card

--

-- Feb 28, 2000 TMC - wo 823, if the tree species on the TA 11 4 card is blank,
 -- it is okay, it means that the card is a subsequent one of
 -- the first TA 11 4 for that tree number; this rule is
 -- intended for the first occurrence of a TA 11 4 card for a
 -- specific tree, it does not have to fire for subsequent
 -- cards; change the rule to skip over any TA 11 4 card that
 -- has a blank tree species

--

--

-- May 23, 2000 RGH - wo 811, add Plot Type indicator N,S,W or E in error message

--

--

--

--

Rule Name: Veg 2000 Check that all data is present for an enhanced tree

PL/SQL Source Code: Rule code has already been printed.

Record: VEG TA 11 4

Field: M O R E 2

Datatype: CHAR

Position: 28

Optional: Y

Length: 1

Zero to Null:

Domain Name: VEG SAMPLE MEASURED/ESTIMATED

Value

Meaning

E

Estimated

M

Measured

Rule Name: Veg 2000 Check that all data is present for an enhanced tree

PL/SQL Source Code: Rule code has already been printed.

Record: VEG TA 11 4

Field: CR. CL. DCIS

Datatype: CHAR

Position: 29

Optional: Y

Length: 1

Zero to Null:

Domain Name: VEG SAMPLE CROWN CLASS

Value

Meaning

-

none for dead trees

C

Codominant

D

Dominant

I

Intermediate

S

Suppressed

Rule Name: Veg 2000 Check that all data is present for an enhanced tree

PL/SQL Source Code: Rule code has already been printed.

Record: VEG TA 11 4

Field: HEIGHT TO LIVE CROWN

Datatype: CHAR

Position: 30

Optional: Y

Length: 2

Zero to Null:

Rule Name: Veg 2000 TD 08 2 TA 11 4 Check Height to Live Crown

PL/SQL Source Code: Rule code has already been printed.

Rule Name: Veg 2000 Check that all data is present for an enhanced tree

PL/SQL Source Code: Rule code has already been printed.

Record: VEG TA 11 4

Field: LOG 1/5 GRADE

Datatype: CHAR

Position: 32

Optional: Y

Length: 1

Zero to Null:

Rule Name: Veg 2000 TD 08 2 TA 11 4 Log Grade Code

PL/SQL Source Code: Rule code has already been printed.

Rule Name: Veg 2000 TD 08 2 TA 11 4 Log Grade Combination

PL/SQL Source Code: Rule code has already been printed.

Rule Name: Veg 2000 TD 08 2 TA 11 4 Last Log Length

PL/SQL Source Code: Rule code has already been printed.

Rule Name: Veg 2000 Check that all data is present for an enhanced tree

PL/SQL Source Code: Rule code has already been printed.

Rule Name: Veg 2000 Check For Valid Tree Species Code for a Log Grade of 'C

PL/SQL Source Code: Rule code has already been printed.

Record: VEG TA 11 4

Field: LOG 1/5 LENGTH

Datatype: NUMBER

Position: 33

Optional: Y

Length: 2

Zero to Null: Y

Rule Name: Veg 2000 TD 08 2 TA 11 4 Calculate Log and Tree Lengths

PL/SQL Source Code: Rule code has already been printed.

Rule Name: Veg 2000 Check that all data is present for an enhanced tree

PL/SQL Source Code: Rule code has already been printed.

Record: VEG TA 11 4

Field: LOG 1/5 SOUND %

Datatype: CHAR

Position: 35

Optional: Y

Length: 2

Zero to Null:

Domain Name: VEG SAMPLE ENCRYPTED PCT

Value

Meaning

--

100% (as well)

00

- 99

zero, ('0' is not allowed!)

1

- 99

—

100 % (true story!)

Rule Name: Veg 2000 Check that all data is present for an enhanced tree

PL/SQL Source Code: Rule code has already been printed.

Record: VEG TA 11 4

Field: LOG 2/6 GRADE

Datatype: CHAR

Position: 37

Optional: Y

Length: 1

Zero to Null:

Rule Name: Veg 2000 TD 08 2 TA 11 4 Log Grade Code

PL/SQL Source Code: Rule code has already been printed.

Rule Name: Veg 2000 TD 08 2 TA 11 4 Log Grade Combination

PL/SQL Source Code: Rule code has already been printed.

Rule Name: Veg 2000 Check For Valid Tree Species Code for a Log Grade of 'C

PL/SQL Source Code: Rule code has already been printed.

Record: VEG TA 11 4

Field: LOG 2/6 LENGTH

Datatype: NUMBER

Position: 38

Optional: Y

Length: 2

Zero to Null: Y

Rule Name: Veg 2000 TD 08 2 TA 11 4 Calculate Log and Tree Lengths

PL/SQL Source Code: Rule code has already been printed.

Record: VEG TA 11 4

Field: LOG 2/6 SOUND %

Datatype: CHAR

Position: 40

Optional: Y

Length: 2

Zero to Null:

Domain Name: VEG SAMPLE ENCRYPTED PCT

Value

Meaning

--

100% (as well)

00 - 99

zero, ('0' is not allowed!)

1 - 99

100 % (true story!)

Record: VEG TA 11 4

Field: LOG 3/7 GRADE

Datatype: CHAR

Position: 42

Optional: Y

Length: 1

Zero to Null:

Rule Name: Veg 2000 TD 08 2 TA 11 4 Log Grade Code

PL/SQL Source Code: Rule code has already been printed.

Rule Name: Veg 2000 TD 08 2 TA 11 4 Log Grade Combination

PL/SQL Source Code: Rule code has already been printed.

Rule Name: Veg 2000 Check For Valid Tree Species Code for a Log Grade of 'C

PL/SQL Source Code: Rule code has already been printed.

Record: VEG TA 11 4

Field: LOG 3/7 LENGTH

Datatype: NUMBER

Position: 43

Optional: Y

Length: 2

Zero to Null: Y

Rule Name: Veg 2000 TD 08 2 TA 11 4 Calculate Log and Tree Lengths

PL/SQL Source Code: Rule code has already been printed.

Record: VEG TA 11 4

Field: LOG 3/7 SOUND %

Datatype: CHAR

Position: 45

Optional: Y

Length:

VEG SAMPLE ENCRYPTED PCT

<u>Domain</u>	<u>Value</u>	<u>Meaning</u>	<u>Zero to Null:</u>
	00	100% (as well)	
	1	zero, ('0' is not allowed!)	
	—	100 % (true story!)	

Record: VEG TA 11 4 **Field:** LOG 4/8 GRADE
Datatype: CHAR
Position: 47 **Optional:** Y
Length: 1 **Zero to Null:**

Rule Name: Veg 2000 TD 08 2 TA 11 4 Log Grade Code

PL/SQL Source Code: Rule code has already been printed.

Rule Name: Veg 2000 TD 08 2 TA 11 4 Log Grade Combination

PL/SQL Source Code: Rule code has already been printed.

Rule Name: Veg 2000 Check For Valid Tree Species Code for a Log Grade of 'C

PL/SQL Source Code: Rule code has already been printed.

Record: VEG TA 11 4 **Field:** LOG 4/8 LENGTH
Datatype: NUMBER
Position: 48 **Optional:** Y
Length: 2 **Zero to Null:** Y

Rule Name: Veg 2000 TD 08 2 TA 11 4 Calculate Log and Tree Lengths

PL/SQL Source Code: Rule code has already been printed.

Record: VEG TA 11 4 **Field:** LOG 4/8 SOUND %
Datatype: CHAR
Position: 50 **Optional:** Y
Length: 2 **Zero to Null:**

Domain Name: VEG SAMPLE ENCRYPTED PCT

<u>Value</u>	<u>Meaning</u>	<u>Zero to Null:</u>
--	100% (as well)	
00	zero, ('0' is not allowed!)	
1	100 % (true story!)	

Record: VEG TA 11 4 **Field:** WILDLIFE APPEARANCE CODE
Datatype: NUMBER
Position: 52 **Optional:** Y
Length: 1 **Zero to Null:**

Rule Name: Veg 2000 Look Up Wildlife Feature Code and Return Feature Name

PL/SQL Source Code: Rule code has already been printed.

Rule Name: Veg 2000 Check that all data is present for an enhanced tree

PL/SQL Source Code: Rule code has already been printed.

Rule Name: Veg 2000 TA 11 4 Check for duplicate wildlife codes

PL/SQL Source Code:

```

VarSrceDataPageNo NUMBER;
VarSrceDataRecId VARCHAR2(20);
VarSrceDataTreeNo VARCHAR2(10);
VarSrceDataWildlifeCd VARCHAR2(10);
VarSrceDataFieldName VARCHAR2(100);
VarTreeNoSeenBefore NUMBER;

BEGIN
  p_return_code := 0;
  p_return_value := NULL;
  p_return_message := NULL;

  -- get the record ID
  VarSrceDataRecId := ffio.getc ( 'RECORD ID' );

  -- get the page number
  VarSrceDataPageNo := ffio.getn ( 'PAGE NUMBER' );

  -- get the tree number
  VarSrceDataTreeNo := ffio.getc ( 'TREE NUMBER' );

  -- get the wildlife code
  VarSrceDataWildlifeCd := ffio.getc;

  -- get the current field name
  VarSrceDataFieldName := ffio.get_current_item_name;

  IF ( VarSrceDataWildlifeCd IS NOT NULL ) THEN
  -- get a previous source data record with the same record ID, page number,
  -- and tree number as the current source data record
    VarTreeNoSeenBefore := ffio.get_previous_record_by_item ( VarSrceDataRecId,
    'PAGE NUMBER',
      VarSrceDataPageNo, 'TREE NUMBER', VarSrceDataTreeNo );

  -- if a matching record is found, return an error
    IF ( VarTreeNoSeenBefore > 0 ) THEN
      p_return_code := 1;
      p_return_message := 'on page '
      || TO_CHAR ( VarSrceDataPageNo )
      || ', for tree number '
      || VarSrceDataTreeNo
      || ', the ' || VarSrceDataFieldName || ' field is duplicated';
    END IF;
  END IF;
END IF;
```

Record: VEG TA 11 4

Field: WILDLIFE CROWN CODE

Datatype: NUMBER

Position: 53

Length: 1

Optional: Y

Zero to Null:

Rule Name: Veg 2000 Look Up Wildlife Feature Code and Return Feature Name

PL/SQL Source Code: Rule code has already been printed.

Rule Name: Veg 2000 Check that all data is present for an enhanced tree

PL/SQL Source Code: Rule code has already been printed.

Rule Name: Veg 2000 TA 11 4 Check for duplicate wildlife codes

PL/SQL Source Code: Rule code has already been printed.

Record: VEG TA 11 4

Field: WILDLIFE BARK CODE

Datatype: NUMBER

Position: 54

Optional: Y

Length: 1

Zero to Null:

Rule Name: Veg 2000 Look Up Wildlife Feature Code and Return Feature Name

PL/SQL Source Code: Rule code has already been printed.

Rule Name: Veg 2000 Check that all data is present for an enhanced tree

PL/SQL Source Code: Rule code has already been printed.

Rule Name: Veg 2000 TA 11 4 Check for duplicate wildlife codes

PL/SQL Source Code: Rule code has already been printed.

Record: VEG TA 11 4

Field: WILDLIFE WOOD CODE

Datatype: NUMBER

Position: 55

Optional: Y

Length: 1

Zero to Null:

Rule Name: Veg 2000 Look Up Wildlife Feature Code and Return Feature Name

PL/SQL Source Code: Rule code has already been printed.

Rule Name: Veg 2000 Check that all data is present for an enhanced tree

PL/SQL Source Code: Rule code has already been printed.

Rule Name: Veg 2000 TA 11 4 Check for duplicate wildlife codes

PL/SQL Source Code: Rule code has already been printed.

Record: VEG TA 11 4

Field: WILDLIFE LICHEN CODE

Datatype: NUMBER

Position: 56

Optional: Y

Length: 1

Zero to Null:

Rule Name: Veg 2000 Look Up Wildlife Feature Code and Return Feature Name

PL/SQL Source Code: Rule code has already been printed.

Rule Name: Veg 2000 Check that all data is present for an enhanced tree

PL/SQL Source Code: Rule code has already been printed.

Rule Name: Veg 2000 TA 11 4 Check for duplicate wildlife codes

PL/SQL Source Code: Rule code has already been printed.

Record: VEG TA 11 4

Field: WILDLIFE USE CODE

Datatype: CHAR

Position: 57

Length: 2

Optional: Y

Zero to Null:

Rule Name: Veg 2000 Lookup Wildlife Use Code

PL/SQL Source Code: Rule code has already been printed.

Rule Name: Veg 2000 TD 08 2 TA 11 4 Check Wildlife Use Code if Tree Number

PL/SQL Source Code: Rule code has already been printed.

Rule Name: Veg 2000 Check that all data is present for an enhanced tree

PL/SQL Source Code: Rule code has already been printed.

Rule Name: Veg 2000 TA 11 4 Check for duplicate wildlife codes

PL/SQL Source Code: Rule code has already been printed.

Record: VEG TA 11 4

Field: BROKEN TOP DIAMETER

Datatype: NUMBER

Position: 59

Length: 3

Optional: Y

Zero to Null:

Rule Name: Veg 2000 Validate Broken Top Diameter

PL/SQL Source Code: Rule code has already been printed.

Record: VEG TA 11 4

Field: BROKEN TOP PROJ. HEIGHT

Datatype: NUMBER

Position: 62

Length: 2

Optional: Y

Zero to Null:

Rule Name: Veg 2000 TD 08 2 TA 11 4 Validate Broken Top

PL/SQL Source Code: Rule code has already been printed.

Record: VEG TA 11 4

Field: S1

Datatype: CHAR

Position: 64

Length: 1

Optional: Y

Zero to Null:

Record: VEG TA 11 4

Field: S2

Datatype: CHAR

Position: 65

Length: 1

Optional: Y

Zero to Null:

Record: VEG TA 11 4

Field: S3

Datatype: CHAR**Position:** 66**Length:** 1**Optional:** Y**Zero to Null:****Record:** VEG TA 11 5**Field:** RECORD ID**Datatype:** NUMBER**Position:** 1**Length:** 3**Optional:** N**Zero to Null:****Record:** VEG TA 11 5**Field:** PAGE NUMBER**Datatype:** NUMBER**Position:** 4**Length:** 2**Optional:** N**Zero to Null:****Record:** VEG TA 11 5**Field:** SEQUENCE NUMBER**Datatype:** NUMBER**Position:** 6**Length:** 4**Optional:** N**Zero to Null:****Record:** VEG TA 11 5**Field:** DAMAGE AGENT TREE NUMBER**Datatype:** NUMBER**Position:** 10**Length:** 3**Optional:** Y**Zero to Null:****Rule Name:** Veg 2000 Validate Auxiliary Damaged Trees Against Live Standing**PL/SQL Source Code:**

```

page_no  NUMBER;
tree_num NUMBER;
rec_no   NUMBER;

```

```

VarSrceDataOrigRecId  VARCHAR2(10);
VarSrceDataPageNo     NUMBER;
v_plot_type            VARCHAR2(10);
v_plot_rec             VARCHAR2(10);
v_plot_id              NUMBER;

```

```

BEGIN

```

```

    p_return_code := 0;
    page_no := ffio.getn('PAGE NUMBER');

```

```

    -- initialize the variables to be used for deriving the plot type
    v_plot_type := NULL;
    v_plot_rec := NULL;
    v_plot_id := NULL;

```

```

    -- get the record ID of the current record;
    VarSrceDataOrigRecID := ffio.getc ( 'RECORD ID' );

```

Rule Name: Veg 2000 Validate Presence of TA 11 4: Auxiliary Plot Card Tree
PL/SQL Source Code:

```

-----
--
-- Validate Presence of TA 11 4: Auxiliary Plot Card Tree Measurement
--
-- check to see when a TA 11 5 card is present, as it is when this rule is run,
-- that there is a matching TA 11 4 card, based on the tree number being the
-- same on both cards
--
-----
--
-- May 08,2001   JPK   wo811; put indicator for plot type in error messages

VarTa114RecNo      NUMBER;
VarTa115DamAgTreeNo  NUMBER;
VarSrceDataPageNo  NUMBER;

v_plot_type        VARCHAR2(10);
v_plot_rec         VARCHAR2(10);
v_plot_id          NUMBER;

BEGIN

    -- initialize this function's return value and code
    p_return_value := NULL;
    p_return_code  := 0;

    -- initialize the variables to be used for deriving the plot type
    v_plot_type := NULL;
    v_plot_rec  := NULL;
    v_plot_id   := NULL;

    -- get the page number value
    VarSrceDataPageNo := ffio.getn ( 'PAGE NUMBER' );

    -- get the plot type (I,N,S,E,W)
    v_plot_rec := '111';
    v_plot_id := ffio.get_previous_record_by_item(v_plot_rec, 'PAGE
NUMBER',VarSrceDataPageNo);
    IF v_plot_id > 0 THEN
        v_plot_type := ffio.getc(v_plot_id, 'PL');
    END IF;

    -- get the TA 11 5 damage agent tree number, return the number -88 if it could
    -- not be retrieved
    VarTa115DamAgTreeNo := ffio.getn ( 'DAMAGE AGENT TREE NUMBER', -88 );

    IF ( VarTa115DamAgTreeNo = -88 ) THEN
        -- TA 11 5 damage agent tree num = -88
        p_return_code := 1;
        p_return_message := 'on page '

```

Record: VEG TA 11 5**Field:** DAMAGE AGENT CODE A**Datatype:** CHAR**Position:** 13**Optional:** Y**Length:** 3**Zero to Null:****Rule Name:** Veg 2000 TL 09 2 TA 11 5 Damage Agent Code**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 Check That the Field is Present When the Tree Number is**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TA 11 5**Field:** DAMAGE AGENT SEVERITY A**Datatype:** CHAR**Position:** 16**Optional:** Y**Length:** 2**Zero to Null:****Rule Name:** Veg 2000 TL 09 2 TA 11 5 Damage Agent Severity**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TA 11 5**Field:** DAMAGE AGENT CODE B**Datatype:** CHAR**Position:** 18**Optional:** Y**Length:** 3**Zero to Null:****Rule Name:** Veg 2000 TL 09 2 TA 11 5 Damage Agent Code**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TA 11 5**Field:** DAMAGE AGENT SEVERITY B**Datatype:** CHAR**Position:** 21**Optional:** Y**Length:** 2**Zero to Null:****Rule Name:** Veg 2000 TL 09 2 TA 11 5 Damage Agent Severity**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TA 11 5**Field:** LOSS INDICATOR CODE 1**Datatype:** CHAR**Position:** 23**Optional:** Y**Length:** 3**Zero to Null:****Rule Name:** Veg 2000 TL 09 2 TA 11 5 Tree Loss Indicator Code**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 TL 09 2 TA 11 5 Check a Broken Top Loss Indicator to th**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TA 11 5**Field:** LS IND 1 LOCATION +/- 1**Datatype:** CHAR

Domain Name: NUMERIC SIGN	Position: 26	Optional: Y
<u>Value</u>	Length: 1	Zero to Null:
+	<u>Meaning</u>	
-	Positive	
	Negative	
<u>Record:</u> <u>VEG TA 11 5</u>	<u>Field:</u> <u>LS IND 1 LOCATION FROM</u>	
	Datatype: NUMBER	
	Position: 27	Optional: Y
	Length: 3	Zero to Null:
Rule Name: Veg 2000 TL 09 2 TA 11 5 Tree Loss Indicator Fields Optionality		
PL/SQL Source Code: Rule code has already been printed.		
Rule Name: Veg 2000 TL 09 2 TA 11 5 Check That the Loss Indicator Location		
PL/SQL Source Code: Rule code has already been printed.		
<u>Record:</u> <u>VEG TA 11 5</u>	<u>Field:</u> <u>LS IND 1 LOCATION +/- 2</u>	
	Datatype: CHAR	
	Position: 30	Optional: Y
	Length: 1	Zero to Null:
Domain Name: NUMERIC SIGN		
<u>Value</u>	<u>Meaning</u>	
+	Positive	
-	Negative	
<u>Record:</u> <u>VEG TA 11 5</u>	<u>Field:</u> <u>LS IND 1 LOCATION TO</u>	
	Datatype: NUMBER	
	Position: 31	Optional: Y
	Length: 3	Zero to Null:
Rule Name: Veg 2000 TL 09 2 TA 11 5 Tree Loss Indicator Fields Optionality		
PL/SQL Source Code: Rule code has already been printed.		
Rule Name: Veg 2000 TL 09 2 TA 11 5 Check That the Loss Indicator Location		
PL/SQL Source Code: Rule code has already been printed.		
<u>Record:</u> <u>VEG TA 11 5</u>	<u>Field:</u> <u>LS IND 1 FREQUENCY</u>	
	Datatype: CHAR	
	Position: 34	Optional: Y
	Length: 1	Zero to Null:
Rule Name: Veg 2000 TL 09 2 TA 11 5 Tree Loss Indicator Fields Optionality		
PL/SQL Source Code: Rule code has already been printed.		
<u>Record:</u> <u>VEG TA 11 5</u>	<u>Field:</u> <u>LOSS INDICATOR CODE 2</u>	
	Datatype: CHAR	

Position: 35**Length:** 3**Optional:** Y**Zero to Null:****Rule Name:** Veg 2000 TL 09 2 TA 11 5 Tree Loss Indicator Code**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 TL 09 2 TA 11 5 Check a Broken Top Loss Indicator to th**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TA 11 5**Field:** LS IND 2 LOCATION +/- 1**Datatype:** CHAR**Position:** 38**Length:** 1**Optional:** Y**Zero to Null:****Domain Name:** NUMERIC SIGN**Value**

+

-

Meaning

Positive

Negative

Record: VEG TA 11 5**Field:** LS IND 2 LOCATION FROM**Datatype:** NUMBER**Position:** 39**Length:** 3**Optional:** Y**Zero to Null:****Rule Name:** Veg 2000 TL 09 2 TA 11 5 Tree Loss Indicator Fields Optionality**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 TL 09 2 TA 11 5 Check That the Loss Indicator Location**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TA 11 5**Field:** LS IND 2 LOCATION +/- 2**Datatype:** CHAR**Position:** 42**Length:** 1**Optional:** Y**Zero to Null:****Domain Name:** NUMERIC SIGN**Value**

+

-

Meaning

Positive

Negative

Record: VEG TA 11 5**Field:** LS IND 2 LOCATION TO**Datatype:** NUMBER**Position:** 43**Length:** 3**Optional:** Y**Zero to Null:****Rule Name:** Veg 2000 TL 09 2 TA 11 5 Tree Loss Indicator Fields Optionality**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 TL 09 2 TA 11 5 Check That the Loss Indicator Location**PL/SQL Source Code:** Rule code has already been printed.

Record: VEG TA 11 5**Field:** LS IND 2 FREQUENCY**Datatype:** CHAR**Position:** 46**Optional:** Y**Length:** 1**Zero to Null:****Rule Name:** Veg 2000 TL 09 2 TA 11 5 Tree Loss Indicator Fields Optionality**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TA 11 5**Field:** LOSS INDICATOR CODE 3**Datatype:** CHAR**Position:** 47**Optional:** Y**Length:** 3**Zero to Null:****Rule Name:** Veg 2000 TL 09 2 TA 11 5 Tree Loss Indicator Code**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 TL 09 2 TA 11 5 Check a Broken Top Loss Indicator to th**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TA 11 5**Field:** LS IND 3 LOCATION +/- 1**Datatype:** CHAR**Position:** 50**Optional:** Y**Length:** 1**Zero to Null:****Domain Name:** NUMERIC SIGNValue

+

-

Meaning

Positive

Negative

Record: VEG TA 11 5**Field:** LS IND 3 LOCATION FROM**Datatype:** NUMBER**Position:** 51**Optional:** Y**Length:** 3**Zero to Null:****Rule Name:** Veg 2000 TL 09 2 TA 11 5 Tree Loss Indicator Fields Optionality**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 TL 09 2 TA 11 5 Check That the Loss Indicator Location**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TA 11 5**Field:** LS IND 3 LOCATION +/- 2**Datatype:** CHAR**Position:** 54**Optional:** Y**Length:** 1**Zero to Null:****Domain Name:** NUMERIC SIGNValue

+

-

Meaning

Positive

Negative

Record: VEG TA 11 5**Field:** LS IND 3 LOCATION TO**Datatype:** NUMBER

Position: 55**Length:** 3**Optional:** Y**Zero to Null:****Rule Name:** Veg 2000 TL 09 2 TA 11 5 Tree Loss Indicator Fields Optionality**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 TL 09 2 TA 11 5 Check That the Loss Indicator Location**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TA 11 5**Field:** LS IND 3 FREQUENCY**Datatype:** CHAR**Position:** 58**Length:** 1**Optional:** Y**Zero to Null:****Rule Name:** Veg 2000 TL 09 2 TA 11 5 Tree Loss Indicator Fields Optionality**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG TA 11 5**Field:** S4**Datatype:** CHAR**Position:** 59**Length:** 1**Optional:** Y**Zero to Null:****Record:** VEG TA 11 5**Field:** S5**Datatype:** CHAR**Position:** 60**Length:** 1**Optional:** Y**Zero to Null:****Record:** VEG TA 11 5**Field:** S6**Datatype:** CHAR**Position:** 61**Length:** 1**Optional:** Y**Zero to Null:****Record:** VEG TA 11 5**Field:** S7**Datatype:** CHAR**Position:** 62**Length:** 1**Optional:** Y**Zero to Null:****Record:** VEG TA 11 5**Field:** S8**Datatype:** CHAR**Position:** 63**Length:** 1**Optional:** Y**Zero to Null:****Record:** VEG TA 11 5**Field:** S9**Datatype:** CHAR**Position:** 64**Optional:** Y

Record: <u>VEG TA 11 5</u>	Length:	1	Zero to Null:
	Field: <u>S10</u>		
	Datatype: CHAR Position: 65 Length: 1		Optional: Y Zero to Null:
Record: <u>VEG TA 11 5</u>	Field: <u>S11</u>		
	Datatype: CHAR Position: 66 Length: 1		Optional: Y Zero to Null:
Record: <u>VEG TA 11 5</u>	Field: <u>S12</u>		
	Datatype: CHAR Position: 67 Length: 1		Optional: Y Zero to Null:
Record: <u>VEG TA 11 6</u>	Field: <u>RECORD ID</u>		
	Datatype: NUMBER Position: 1 Length: 3		Optional: N Zero to Null:
Record: <u>VEG TA 11 6</u>	Field: <u>PAGE NUMBER</u>		
	Datatype: NUMBER Position: 4 Length: 2		Optional: N Zero to Null:
Record: <u>VEG TA 11 6</u>	Field: <u>SEQUENCE NUMBER</u>		
	Datatype: NUMBER Position: 6 Length: 4		Optional: N Zero to Null:
Record: <u>VEG TA 11 6</u>	Field: <u>NOTES</u>		
	Datatype: CHAR Position: 10 Length: 256		Optional: Y Zero to Null:
Record: <u>VEG EP 12 1</u>	Field: <u>RECORD ID</u>		
	Datatype: NUMBER Position: 1 Length: 3		Optional: N Zero to Null:
Record: <u>VEG EP 12 1</u>	Field: <u>PAGE NUMBER</u>		
	Datatype: NUMBER		

Position: 4
Length: 2

Optional: Y
Zero to Null:

Rule Name: Veg 2000 Validate Page Number

PL/SQL Source Code: Rule code has already been printed.

Record: VEG EP 12 1

Field: SEQUENCE NUMBER

Datatype: NUMBER

Position: 6

Length: 4

Optional: N
Zero to Null:

Record: VEG EP 12 1

Field: PL

Datatype: CHAR

Position: 10

Length: 1

Optional: N
Zero to Null:

Domain Name: VEG SAMPLE PL - PLOT NO

Value

Meaning

E

I

M

N

S

W

Record: VEG EP 12 1

Field: POLYGON IDENTIFIER

Datatype: CHAR

Position: 11

Length: 14

Optional: Y
Zero to Null:

Rule Name: Veg 2000 EP 12 1 EP 12 2 Field Optionality for Lignum Projects F

PL/SQL Source Code:

```
curr_fld_name VARCHAR2(80) := null;
curr_fld_value VARCHAR2(80) := null;
curr_proj VARCHAR2(8) := null;
sample_date VARCHAR2(20);
```

BEGIN

```
p_return_code := 0;
p_return_value := null;
p_return_message := null;
```

```
curr_fld_name := ffio.get_current_item_name;
curr_fld_value := ffio.getc;
curr_proj := ffio.getc(ffio.first_record, 'PROJECT ID');
```

```
sample_date := ffio.getc(ffio.first_record, 'MEASUREMENT DATE');
```

```
IF curr_fld_value IS NULL THEN
```

Record: VEG EP 12 1**Field:** MEASUREMENT DATE**Datatype:** DATE**Date Mask:** YYYYMONDD**Position:** 25**Optional:** N**Length:** 9**Zero to Null:****Domain Name:** DATE MIN JAN 01 1997 - FULL**Value****Meaning**

1997JAN01 - 3999JAN01 minnum date for 1999 Ground Sampling Format - YYM

Rule Name: Veg 2000 EP 12 1 EP 12 2 Field Optionality for Lignum Projects F**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG EP 12 1**Field:** CREW PERSON #1**Datatype:** CHAR**Position:** 34**Optional:** N**Length:** 4**Zero to Null:****Rule Name:** Veg 2000 Validate Person Against First Crew List**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 Check the Person's Initials**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 EP 12 1 EP 12 2 Field Optionality for Lignum Projects F**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG EP 12 1**Field:** CREW PERSON #2**Datatype:** CHAR**Position:** 38**Optional:** Y**Length:** 4**Zero to Null:****Rule Name:** Veg 2000 Validate Person Against First Crew List**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 Check the Person's Initials**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG EP 12 1**Field:** TOTAL PAGES**Datatype:** NUMBER**Position:** 42**Optional:** N**Length:** 2**Zero to Null:****Record:** VEG EP 12 1**Field:** PLOT SIZE**Datatype:** NUMBER**Position:** 44**Optional:** Y**Length:** 4**Zero to Null:****Domain Name:** VEG SAMPLE FIXED VALUE 10**Value****Meaning**

10.00

ten

Rule Name: Veg 2000 EP 12 1 EP 12 2 Field Optionality for Lignum Projects F**PL/SQL Source Code:** Rule code has already been printed.

Record: VEG EP 12 1**Field:** PLOT UNIFORMITY CODE**Datatype:** NUMBER**Position:** 48**Optional:** Y**Length:** 1**Zero to Null:****Domain Name:** ECOSYSTEM UNIFORMITY CLASS**Value****Meaning**

1	>95% in one homogeneous ecosystem
2	81-95% of plot in one homogenous ecosystem
3	51-80% of plot in one homogenous ecosystem
4	20-50% of plot in one homogeneous ecosystem
5	mosaic with =<20% of plot in any homogenous ecosyst

Rule Name: Veg 2000 EP 12 1 EP 12 2 Field Optionality for Lignum Projects F**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG EP 12 1**Field:** BIOGEOCLIMATIC ZONE**Datatype:** CHAR**Position:** 49**Optional:** N**Length:** 4**Zero to Null:****Rule Name:** Veg 2000 Lookup Biogeoclimatic Unit**PL/SQL Source Code:**

```
--
-- NOV.23,2000 RGH wo.921 added TRUE to coms.pb call for phase
-- (BOOLEAN condition to denote it can be blank)
```

```
-----
l1 VARCHAR2(10);
l2 VARCHAR2(10);
l3 VARCHAR2(10);
l4 VARCHAR2(10);
no_value VARCHAR2(10);
result NUMBER;
BEGIN
p_return_code := 0;
p_return_value := NULL;
-- no_value := 0;
no_value := 'missing';
l1 := ffio.getc('Biogeoclimatic Zone');
l2 := ffio.getc('Biogeoclimatic Subzone');
l3 := LTRIM(ffio.getc('Biogeoclimatic Variant'),'0');
l4 := ffio.getc('Biogeoclimatic Phase');

IF ( l1||l2||l3||l4 IS NOT NULL ) THEN
result := coms.get_bgc_class_skey ( l1, l2, l3 , l4, TRUE);
IF ( result IS NULL ) THEN
p_return_code := 1;
IF ( l1 IS NULL ) THEN l1 := no_value;
ELSIF ( l2 IS NULL ) THEN l2 := no_value;
ELSIF ( l3 IS NULL ) THEN l3 := no_value;
ELSIF ( l4 IS NULL ) THEN l4 := no_value;
ELSE no_value := NULL;
END IF;
p_return_message :=
```

Record: VEG EP 12 1**Field:** BIOGEOCLIMATIC SUBZONE**Datatype:** CHAR**Position:** 53**Optional:** N**Length:** 3**Zero to Null:****Record:** VEG EP 12 1**Field:** BIOGEOCLIMATIC VARIANT**Datatype:** NUMBER**Position:** 56**Optional:** Y**Length:** 2**Zero to Null:****Record:** VEG EP 12 1**Field:** BIOGEOCLIMATIC PHASE**Datatype:** CHAR**Position:** 58**Optional:** Y**Length:** 2**Zero to Null:****Record:** VEG EP 12 1**Field:** SLOPE %**Datatype:** NUMBER**Position:** 60**Optional:** N**Length:** 3**Zero to Null:****Domain Name:** SLOPE PCT**Value**

0 - 150

Meaning**Record:** VEG EP 12 1**Field:** ASPECT**Datatype:** NUMBER**Position:** 63**Optional:** N**Length:** 3**Zero to Null:****Domain Name:** VEG SAMPLE ASPECT BRG**Value**

0 - 359

999

Meaning

Aspect Bearing

Level ground - less than 2% slope.

Rule Name: Veg 2000 Determine Aspect with Slope**PL/SQL Source Code:**

```

aspect NUMBER;
slope NUMBER;
BEGIN
aspect := ffio.getn('ASPECT', -1);
IF aspect = -1 THEN
p_return_code := 7;
p_return_message := ' The value for aspect ***' ||
ffio.getc('ASPECT') || '*** is non numeric.';
END IF;
slope := ffio.getn('SLOPE %', -1);
IF slope = -1 THEN
p_return_code := 8;
p_return_message := ' The value for slope ***' ||
ffio.getc('SLOPE %') || '*** is non numeric.';
END IF;

```

Record: VEG EP 12 1**Field:** ELEVATION**Datatype:** NUMBER**Position:** 66**Length:** 4**Optional:** N**Zero to Null:****Domain Name:** ELEVATION**Value**

0 - 4000

Meaning**Record:** VEG EP 12 1**Field:** SURFACE SHAPE**Datatype:** CHAR**Position:** 70**Length:** 2**Optional:** N**Zero to Null:****Domain Name:** SURFACE SHAPE**Value**

CV

CX

ST

Meaning

concave

convex

straight

Rule Name: Veg 2000 EP 12 1 EP 12 2 Field Optionality for Lignum Projects F**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG EP 12 1**Field:** MESO SLOPE**Datatype:** CHAR**Position:** 72**Length:** 1**Optional:** N**Zero to Null:****Domain Name:** MESO SLOPE**Value**

C

D

F

L

M

T

U

Meaning

Crest

Depression

Flat

Lower Slope

Mid Slope

Toe

Upper Slope

Rule Name: Veg 2000 EP 12 1 EP 12 2 Field Optionality for Lignum Projects F**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG EP 12 1**Field:** MICROTOPOGRAPHY**Datatype:** CHAR**Position:** 73**Length:** 2**Optional:** N**Zero to Null:****Domain Name:** MICROTOPOGRAPHY**Value**

EX

MO

SM

ST

Meaning

extremely mounded

moderately mounded

smooth

strongly mounded

Rule Name: Veg 2000 EP 12 1 EP 12 2 Field Optionality for Lignum Projects F

PL/SQL Source Code: Rule code has already been printed.

Record: VEG EP 12 1

Field: ROCKY COBBLES/STONES %

Datatype: CHAR

Position: 75

Optional: N

Length: 3

Zero to Null:

Domain Name: VEG SAMPLE PERCENTAGE 100

Value

Meaning

0 - 100

00 - 99

000 - 100

Record: VEG EP 12 1

Field: ROCKY BEDROCK

Datatype: CHAR

Position: 78

Optional: N

Length: 3

Zero to Null:

Domain Name: VEG SAMPLE PERCENTAGE 100

Value

Meaning

0 - 100

00 - 99

000 - 100

Record: VEG EP 12 1

Field: SLOPE FAILURE WITHIN PLOT

Datatype: CHAR

Position: 81

Optional: N

Length: 1

Zero to Null:

Domain Name: INDICATOR

Value

Meaning

N

No (False)

Y

Yes (True)

Rule Name: Veg 2000 EP 12 1 EP 12 2 Field Optionality for Lignum Projects F

PL/SQL Source Code: Rule code has already been printed.

Record: VEG EP 12 1

Field: SLOPE FAILURE BETWEEN PLOTS

Datatype: CHAR

Position: 82

Optional: Y

Length: 2

Zero to Null:

Domain Name: VEG SAMPLE INDICATOR WITH NV

Value

Meaning

N

No

NV

Not Visited

Y

Yes

Rule Name: Mandatory Field for Non-"Change Monitoring Inventory" Samples

PL/SQL Source Code:

BEGIN

p_return_code := 0;

IF (ffio.getc (1, 'SAMPLE TYPE') IN ('M','F')) THEN
NULL;

ELSIF (ffio.getc IS NULL) THEN
p_return_code := 1;
p_return_message := 'Required field is missing for '|
ffio.get_current_item_name ;
END IF;

Record: VEG EP 12 1

Field: GULLIES WITHIN MAIN PLOT

Datatype: CHAR

Position: 84

Length: 1

Optional: N

Zero to Null:

Domain Name: INDICATOR

Value

N

Y

Meaning

No (False)

Yes (True)

Rule Name: Veg 2000 EP 12 1 EP 12 2 Field Optionality for Lignum Projects F

PL/SQL Source Code: Rule code has already been printed.

Record: VEG EP 12 1

Field: GULLIES BETWEEN PLOTS

Datatype: CHAR

Position: 85

Length: 2

Optional: Y

Zero to Null:

Domain Name: VEG SAMPLE INDICATOR WITH NV

Value

N

NV

Y

Meaning

No

Not Visited

Yes

Rule Name: Manditory Field for Non-"Change Monitoring Inventory" Samples

PL/SQL Source Code: Rule code has already been printed.

Record: VEG EP 12 1

Field: FLOOD HAZARD

Datatype: CHAR

Position: 87

Length: 1

Optional: N

Zero to Null:

Domain Name: VEG SAMPLE FLOOD HAZARD

Value

F

N

Meaning

Frequent

NotApplicable

O Occasional
R Occasional

Rule Name: Veg 2000 EP 12 1 EP 12 2 Field Optionality for Lignum Projects F
PL/SQL Source Code: Rule code has already been printed.

Record: VEG EP 12 1 **Field:** OPEN WATER FLOWING
Datatype: CHAR
Position: 88 **Optional:** N
Length: 3 **Zero to Null:**
Domain Name: VEG SAMPLE PERCENTAGE 100

<u>Value</u>	<u>Meaning</u>

0	- 100
00	- 99
000	- 100

Record: VEG EP 12 1 **Field:** OPEN WATER STANDING
Datatype: CHAR
Position: 91 **Optional:** N
Length: 3 **Zero to Null:**
Domain Name: VEG SAMPLE PERCENTAGE 100

<u>Value</u>	<u>Meaning</u>

0	- 100
00	- 99
000	- 100

Record: VEG EP 12 1 **Field:** SURFICIAL MATERIAL 1
Datatype: CHAR
Position: 94 **Optional:** Y
Length: 2 **Zero to Null:**
Domain Name: VEG SAMPLE SURFICIAL MATERIAL

<u>Value</u>	<u>Meaning</u>
A	anthropogenic
C	Colluvial
E	eolian
F	
FG	glaciofluvial
L	lacustrine
M	moraine
O	organic
R	- bedrock
V	volcanic
W	marine

Rule Name: Veg 2000 EP 12 ED 13 Check That the Soil Features and the Soil D
PL/SQL Source Code:

```
--
-----
--
-- Veg EP 12 ED 13 Check That the Soil Features and the Soil Description Are
-- All Present or All Blank
--
--
-- Modifications
--
-- May 25, 1999 TMC - wo 657, include the Veg ED 13 card set to be checked in
--                   addition to the Veg EP 12 card set
--
--
-- June 29, 2000 RGH - wo868, modify as per order
--
-----
--
```

```
VarAboveGrndSoilDescIsNull BOOLEAN;
VarAboveGrndSoilDescIsPresent BOOLEAN;
VarBelowGrndSoilDescIsNull BOOLEAN;
VarBelowGrndSoilDescIsPresent BOOLEAN;
VarDepthToIsNull BOOLEAN;
VarDepthToIsPresent BOOLEAN;
VarEcoDescAboveGrndDescRecCnt NUMBER;
VarEcoDescAboveGrndDescRecId VARCHAR2(3);
VarEcoDescAboveGrndDescRecNo NUMBER;
VarEcoDescBelowGrndDescRecCnt NUMBER;
VarEcoDescBelowGrndDescRecId VARCHAR2(3);
VarEcoDescBelowGrndDescRecNo NUMBER;
VarEcoDescHeaderRecId VARCHAR2(3);
VarEcoDescHeaderRecNo NUMBER;
VarHumusFormSoilColorIsNull BOOLEAN;
VarHumusFormSoilColorIsPresent BOOLEAN;
VarOrigRecNo NUMBER;
VarSrceDataPageNo NUMBER;
VarSrceDataRecId NUMBER;
VarSurficialMaterialIsNull BOOLEAN;
VarSurficialMaterialIsPresent BOOLEAN;
```

```
BEGIN
```

```
p_return_value := NULL;
p_return_code := 0;
p_return_message := NULL;

-- get the rec ID
VarSrceDataRecId := ffio.getn ( 'RECORD ID' );

-- get the page number
VarSrceDataPageNo := ffio.getn ( 'PAGE NUMBER' );
```

Record: VEG EP 12 1**Field:** SURFICIAL MATERIAL 2**Datatype:** CHAR**Position:** 96**Optional:** Y**Length:** 2**Zero to Null:****Domain Name:** VEG SAMPLE SURFICIAL MATERIAL**Value****Meaning**

A

anthropogenic

C

Colluvial

E

eolian

F

FG

glaciofluvial

L

lacustrine

M

moraine

O

organic

R

- bedrock

V

volcanic

W

marine

Record: VEG EP 12 1**Field:** DEPTH TO WATER TABLE**Datatype:** NUMBER**Position:** 98**Optional:** Y**Length:** 3**Zero to Null:****Domain Name:** SOIL FEATURE DEPTH**Value****Meaning**

0

- 300.0

Record: VEG EP 12 1**Field:** DEPTH TO MOTTLES**Datatype:** NUMBER**Position:** 101**Optional:** Y**Length:** 3**Zero to Null:****Domain Name:** SOIL FEATURE DEPTH**Value****Meaning**

0

- 300.0

Record: VEG EP 12 1**Field:** DEPTH TO ROOT REST PAN.**Datatype:** NUMBER**Position:** 104**Optional:** Y**Length:** 3**Zero to Null:****Domain Name:** SOIL FEATURE DEPTH**Value****Meaning**

0

- 300.0

Record: VEG EP 12 1**Field:** DEPTH TO BEDROCK**Datatype:** NUMBER**Position:** 107**Optional:** Y**Length:** 3**Zero to Null:****Domain Name:** SOIL FEATURE DEPTH**Value****Meaning**

0

- 300.0

Record: VEG EP 12 1**Field:** DEPTH TO FROZEN LAYER

		Datatype: NUMBER	
		Position: 110	Optional: Y
		Length: 3	Zero to Null:
Domain Name:	SOIL FEATURE DEPTH		
<u>Value</u>	<u>Meaning</u>		
0	- 300.0		
Record:	<u>VEG EP 12 1</u>	Field:	<u>DEPTH TO CARBONATES</u>
		Datatype: NUMBER	
		Position: 113	Optional: Y
		Length: 3	Zero to Null:
Domain Name:	SOIL FEATURE DEPTH		
<u>Value</u>	<u>Meaning</u>		
0	- 300.0		
Record:	<u>VEG EP 12 1</u>	Field:	<u>DEPTH TO R1</u>
		Datatype: NUMBER	
		Position: 116	Optional: Y
		Length: 3	Zero to Null:
Domain Name:	SOIL FEATURE DEPTH		
<u>Value</u>	<u>Meaning</u>		
0	- 300.0		
Record:	<u>VEG EP 12 1</u>	Field:	<u>DEPTH TO R2</u>
		Datatype: NUMBER	
		Position: 119	Optional: Y
		Length: 3	Zero to Null:
Domain Name:	SOIL FEATURE DEPTH		
<u>Value</u>	<u>Meaning</u>		
0	- 300.0		
Record:	<u>VEG EP 12 1</u>	Field:	<u>DEPTH NOT APPLICABLE</u>
		Datatype: CHAR	
		Position: 122	Optional: Y
		Length: 1	Zero to Null:
Domain Name:	INDICATOR		
<u>Value</u>	<u>Meaning</u>		
N	No (False)		
Y	Yes (True)		

Rule Name: Veg 2000 EP 12 1 Check for a Soil Features Depth When the Soil F
PL/SQL Source Code:

```
--
-----
--
-- Veg EP 12 1 Check For a Soil Features Depth Value When the Soil Features
-- Depths are Applicable
--
-----
--
```

Record: VEG EP 12 1**Field:** HUMUS FORM**Datatype:** CHAR**Position:** 123**Length:** 2**Optional:** Y**Zero to Null:****Domain Name:** HUMUS FORM**Value**

MD

MM

MO

MR

MU

NA

Meaning

moder

mormoder

mor

mullmoder

mull

not applicable; observed, but no assessment possib

Record: VEG EP 12 1**Field:** SOIL COLOUR**Datatype:** CHAR**Position:** 125**Length:** 2**Optional:** Y**Zero to Null:****Domain Name:** SOIL COLOR**Value**

D

L

M

N

Meaning

dark; choc. brown, black, Munsell <4

light; very pale, Munsell >6

medium; Munsell 4 to 6

not applicable; observation precluded by site cond

Record: VEG EP 12 2**Field:** RECORD ID**Datatype:** NUMBER**Position:** 1**Length:** 3**Optional:** N**Zero to Null:****Record:** VEG EP 12 2**Field:** PAGE NUMBER**Datatype:** NUMBER**Position:** 4**Length:** 2**Optional:** N**Zero to Null:****Record:** VEG EP 12 2**Field:** SEQUENCE NUMBER**Datatype:** NUMBER**Position:** 6**Length:** 4**Optional:** N**Zero to Null:****Record:** VEG EP 12 2**Field:** LOCATION**Datatype:** CHAR**Position:** 10**Length:** 11**Optional:** N**Zero to Null:****Rule Name:** Veg 2000 EP 12 1 EP 12 2 Field Optionality for Lignum Projects F**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 Convert Ecosys Plot Location to Sequence Number

PL/SQL Source Code:

```
-- CAW 23/11/99 Rule expanded to accept P, 2 and 3 since the valeus
-- are right on the card and better error messaging and
-- rule added to
-- validation step as well.

cur_field VARCHAR2(256);
BEGIN
  p_return_code := 0;
  cur_field := ffio.getc;
  IF ( cur_field IN ('PLOT CENTRE','P') ) THEN
    p_return_value := '1';
  ELSIF ( cur_field IN ('SITE TWO','2') ) THEN
    p_return_value := '2';
  ELSIF ( cur_field IN ('SITE THREE','3') ) THEN
    p_return_value := '3';
  ELSIF ( cur_field IN ('SITE FOUR','4') ) THEN
    p_return_value := '4';
  ELSIF ( cur_field IN ('SITE FIVE','5') ) THEN
    p_return_value := '5';
  ELSE
    p_return_code := 1;
    p_return_message := 'The Ecosystem Location Code "'||
      cur_field||'" is not a valid code. The valid codes consist '||
      'of "PLOT CENTRE" or "P", "SITE TWO" or "2", "SITE THREE", or "3", '||
      '"SITE FOUR" or "4" or "SITE FIVE" or "5".';
  END IF;
```

Record: VEG EP 12 2**Field: SMR****Datatype:** CHAR**Position:** 21**Length:** 1**Optional:** Y**Zero to Null:****Domain Name:** HYGROTOPE**Value**

-

0

1

2

3

4

5

6

7

8

Meaning

Not Applicable (lakes, icefields, pavements)

very xeric, extremely dry

xeric, very dry

sub xeric, moderately dry

sub-mesic, slightly dry

mesic, fresh

sub hygric, moist

hygric, very moist

sub-hydric, wet

hydric, very wet

Rule Name: Veg 2000 EP 12 2 Validate Field Optionality for Ecosystem Descri**PL/SQL Source Code:**

BEGIN

Record: VEG EP 12 2**Field:** SNR**Datatype:** CHAR**Position:** 22**Length:** 1**Optional:** Y**Zero to Null:****Domain Name:** TROPHOTOPE**Value**

-

A

B

C

D

E

F

Meaning

Not Applicable (lakes, icefields, pavements)

oligotrophic, nutrient very poor

sub mesotrophic, nutrient poor

mesotrophic, nutrient medium

permesotrophic, nutrient rich

eutrophic, nutrient very rich

hypereutrophic, saline nutritional state

Rule Name: Veg 2000 EP 12 2 Validate Field Optionality for Ecosystem Descri**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG EP 12 2**Field:** SITE SERIES AND COVERAGE#**Datatype:** CHAR**Position:** 23**Length:** 3**Optional:** Y**Zero to Null:****Rule Name:** Veg 2000 Lookup Site Series**PL/SQL Source Code:**

```
--
-- NOV.23,200 RGH - wo.921 added TRUE to coms.pb call for phase
-- (BOOLEAN condition to denote it can be blank)
--
--March 26,2001 RGH - wo.915 Added in match on Region as per model change
-- Site series code now based on region and Bioclimatic
classification
--
-- Feb 20,2002 RGH changed projects to VGIS_projects on move to MSRM
--
-- Nov 27, 2002 CAW - removed limit of 2 characters for site series.
--
```

```
bgc_key NUMBER;
series VARCHAR2(10);
temp VARCHAR2(10);
rec INTEGER;
curr_loc VARCHAR2(22);
region_id NUMBER;
region_code VARCHAR2(10);
project_code VARCHAR2(10);
reg_description VARCHAR2(40);
project_pointer NUMBER;
BEGIN
p_return_code := 0;
p_return_value := null;
region_code := null;
region_id :=null;
```

Rule Name: Veg 2000 EP 12 2 Validate Field Optionality for Ecosystem Descri

PL/SQL Source Code: Rule code has already been printed.

Record: VEG EP 12 2

Field: SITE SERIES AND COVERAGE PE

Datatype: NUMBER

Position: 26

Optional: Y

Length: 3

Zero to Null:

Domain Name: PERCENTAGE 100 1

Value

0 - 100.0

Meaning

Rule Name: Veg 2000 Validate Ecosystem Description Coverage Percentages

PL/SQL Source Code:

```
total_pct NUMBER;
rec_pct NUMBER;
rec_no NUMBER;
temp NUMBER;
BEGIN
p_return_code := 0;
p_return_value := NULL;
--
-- if this rule has already been applied against a 122 record, exit
--
IF ( ffio.get_previous_record_by_item ( '122' ) > 0 ) THEN
RETURN;
END IF;

total_pct := 0;
rec_no := ffio.get_current_record;
WHILE ( rec_no > 0 AND total_pct < 200 ) LOOP
BEGIN
rec_no := ffio.set_current_record ( rec_no );
rec_pct := ffio.getn('SITE SERIES AND COVERAGE PER');
EXCEPTION WHEN VALUE_ERROR THEN
p_return_code := 4;
p_return_message := ' On VEG Card 12, at the Location ' ||
ffio.getc(rec_no, 'LOCATION') || ', the Coverage ' ||
' Percent ***' || ffio.getc('SITE SERIES AND COVERAGE PER') ||
'*** is non numeric. ';
RETURN;
END;
IF ( rec_pct IS NULL ) THEN
p_return_code := 1;
p_return_message := ' On VEG Card 12, at the Location ' ||
ffio.getc(rec_no, 'LOCATION') || ', the ***Coverage ' ||
'Percentage*** is missing.';
RETURN;
END IF;

total_pct := total_pct + rec_pct;

rec_no := ffio.get_next_record_by_item ( '122' );
END LOOP;
```

Rule Name: Veg 2000 EP 12 2 Validate Field Optionality for Ecosystem Descri

PL/SQL Source Code: Rule code has already been printed.

Record: VEG EP 12 2

Field: LAND COVER CLASSIFICATION1

Datatype: CHAR

Position: 29

Optional: N

Length: 1

Zero to Null:

Rule Name: Veg 2000 Lookup BC Land Cover Classification Code

PL/SQL Source Code:

```
-- Modifications:
--
-- Chet W. Dec 17, 2002 - Tranlating old level 5 RP code to new RZ code to
--                          support old data. Further translations expected.
--

L1 VARCHAR2(10);
L2 VARCHAR2(10);
L3 VARCHAR2(10);
L4 VARCHAR2(10);
L5 VARCHAR2(10);
result NUMBER;
BEGIN
L1 := ffio.getc('LAND COVER CLASSIFICATION1');
L2 := ffio.getc('LAND COVER CLASSIFICATION2');
L3 := ffio.getc('LAND COVER CLASSIFICATION3');
L4 := ffio.getc('LAND COVER CLASSIFICATION4');
L5 := ffio.getc('LAND COVER CLASSIFICATION5');

IF ( L1 || L2 || L3 || L4 || L5 IS NULL ) THEN
  p_return_code := 0;
  p_return_value := NULL;
ELSE
  result := coms.get_bc_land_class_skey ( L1, L2, L3, L4, L5 );

  -- If we didn't get a match, try to see if translating level 5
  -- codes to the new codes get a match

  IF ( result IS NULL AND L1 = 'N' ) THEN
    L5 := UPPER(L5);
    IF ( L5 = 'RP' ) THEN
      L5 := 'RZ';
    ELSIF ( L5 = 'SC' ) THEN
      L5 := 'PN';
    ELSIF ( L5 = 'MS' ) THEN
      L5 := 'MZ';
    ELSIF ( L5 = 'MO' ) THEN
      L5 := 'MN';
    ELSIF ( L5 = 'TS' ) THEN
      L5 := 'TZ';
    ELSIF ( L5 = 'RR' ) THEN
```

Rule Name: Veg 2000 EP 12 2 Validate Field Optionality for Ecosystem Descri

PL/SQL Source Code: Rule code has already been printed.

Record: VEG EP 12 2

Field: LAND COVER CLASSIFICATION2

Datatype: CHAR

Position: 30

Optional: N

Length: 1

Zero to Null:

Rule Name: Veg 2000 EP 12 2 Validate Field Optionality for Ecosystem Descri

PL/SQL Source Code: Rule code has already been printed.

Record: VEG EP 12 2

Field: LAND COVER CLASSIFICATION3

Datatype: CHAR

Position: 31

Optional: N

Length: 1

Zero to Null:

Rule Name: Veg 2000 EP 12 2 Validate Field Optionality for Ecosystem Descri

PL/SQL Source Code: Rule code has already been printed.

Record: VEG EP 12 2

Field: LAND COVER CLASSIFICATION4

Datatype: CHAR

Position: 32

Optional: Y

Length: 2

Zero to Null:

Record: VEG EP 12 2

Field: LAND COVER CLASSIFICATION5

Datatype: CHAR

Position: 34

Optional: N

Length: 2

Zero to Null:

Rule Name: Veg 2000 EP 12 2 Validate Field Optionality for Ecosystem Descri

PL/SQL Source Code: Rule code has already been printed.

Record: VEG EP 12 2

Field: S1

Datatype: CHAR

Position: 36

Optional: Y

Length: 1

Zero to Null:

Record: VEG EP 12 2

Field: S2

Datatype: CHAR

Position: 37

Optional: Y

Length: 1

Zero to Null:

Record: VEG EP 12 2

Field: S3

Datatype: CHAR

Position:

	Length: 38	Optional: Y
	1	Zero to Null:
Record: <u>VEG EP 12 2</u>	Field: <u>S4</u>	
	Datatype: CHAR	
	Position: 39	Optional: Y
	Length: 1	Zero to Null:
Record: <u>VEG EP 12 2</u>	Field: <u>S5</u>	
	Datatype: CHAR	
	Position: 40	Optional: Y
	Length: 1	Zero to Null:
Record: <u>VEG EP 12 3</u>	Field: <u>RECORD ID</u>	
	Datatype: NUMBER	
	Position: 1	Optional: N
	Length: 3	Zero to Null:
Record: <u>VEG EP 12 3</u>	Field: <u>PAGE NUMBER</u>	
	Datatype: NUMBER	
	Position: 4	Optional: N
	Length: 2	Zero to Null:
Record: <u>VEG EP 12 3</u>	Field: <u>SEQUENCE NUMBER</u>	
	Datatype: NUMBER	
	Position: 6	Optional: N
	Length: 4	Zero to Null:
Record: <u>VEG EP 12 3</u>	Field: <u>HORIZON</u>	
	Datatype: CHAR	
	Position: 10	Optional: Y
	Length: 3	Zero to Null:

Rule Name: Veg 2000 EP 12 ED 13 Soil Horizon Code
PL/SQL Source Code:

```
--
-----
--
-- Veg EP 12 ED 13 Soil Horizon Code
--
-- Modifications
--
-- Jan 18, 1999 TMC - wo 678, renumber the return codes to eliminate all the
```

Rule Name: Veg 2000 EP 12 ED 13 Check That the Soil Features and the Soil D
PL/SQL Source Code: Rule code has already been printed.

Rule Name: Veg 2000 EP 12 ED 13 Check That the Soil Description Data Record
PL/SQL Source Code:

```
--
-----
--
-- Veg EP 12 ED 13 Check That the Soil Description Data Record Has a Depth
-- Measurement
--
-----
--
--
-- Apr 18, 2000 RGH wo.838 made greter than or equal to 0 rather than greater than
0
--
--
-- May 16, 2000 RGH wo.849 revised above to make only for DOP, R and water soil
Horizon.
--

VarSrceDataDepthMeas NUMBER;
VarSrceDataPageNo NUMBER;
VarSrceDataSeqNo NUMBER;
VarSrceDataSoilHorizonCode VARCHAR2(3);

BEGIN

-- initialize this function's return value, code, and message
p_return_value := NULL;
p_return_code := 0;
p_return_message := NULL;

-- get the page number
VarSrceDataPageNo := ffio.getn ( 'PAGE NUMBER' );

-- get the sequence number from the source data record
VarSrceDataSeqNo := ffio.getn ( 'SEQUENCE NUMBER' );

-- get the soil horizon code from the source data record
VarSrceDataSoilHorizonCode := ffio.getc;

-- get the depth measurement from the source data record
VarSrceDataDepthMeas := ffio.getn ( 'DEPTH DISTANCE FROM ZERO' );

-- check if the depth measurement is equal to null, that is, check if there is
-- no depth measurement
IF ( VarSrceDataDepthMeas IS NULL )
THEN
-- the depth measurement is equal to null, that is, there is no depth
-- measurement
```

Record: VEG EP 12 3**Field:** DEPTH DISTANCE FROM ZERO**Datatype:** NUMBER**Position:** 13**Optional:** Y**Length:** 4**Zero to Null:****Record:** VEG EP 12 4**Field:** RECORD ID**Datatype:** NUMBER**Position:** 1**Optional:** N**Length:** 3**Zero to Null:****Record:** VEG EP 12 4**Field:** PAGE NUMBER**Datatype:** NUMBER**Position:** 4**Optional:** N**Length:** 2**Zero to Null:****Record:** VEG EP 12 4**Field:** SEQUENCE NUMBER**Datatype:** NUMBER**Position:** 6**Optional:** N**Length:** 4**Zero to Null:****Record:** VEG EP 12 4**Field:** HORIZON**Datatype:** CHAR**Position:** 10**Optional:** Y**Length:** 3**Zero to Null:****Rule Name:** Veg 2000 EP 12 ED 13 Soil Horizon Code**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 EP 12 ED 13 Check That the Soil Description Data Record**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 EP 12 ED 13 Check That the Soil Features and the Soil D**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG EP 12 4**Field:** DEPTH DISTANCE FROM ZERO**Datatype:** NUMBER**Position:** 13**Optional:** Y**Length:** 4**Zero to Null:****Rule Name:** Veg 2000 EP 12 4 ED 13 3 Check That the Below Zero Depth Meas is**PL/SQL Source Code:**

--

--

-- Veg EP 12 4 ED 13 3 Check That the Below Zero Depth Meas is Within the Depth
-- of Pit Meas

--

Record: VEG EP 12 4**Field:** TEXTURE**Datatype:** CHAR**Position:** 17**Length:** 4**Optional:** Y**Zero to Null:****Record:** VEG EP 12 4**Field:** COARSE FRAG. % TOTAL**Datatype:** NUMBER**Position:** 21**Length:** 3**Optional:** Y**Zero to Null:****Domain Name:** VEG SAMPLE PERCENTAGE 100ValueMeaning

0 - 100

00 - 99

000 - 100

Rule Name: Veg 2000 EP 12 4 ED 13 3 Coarse Fragments Percentage Total**PL/SQL Source Code:**

```

total NUMBER;
small NUMBER;
large NUMBER;
BEGIN
  BEGIN
    total := ffio.getn('COARSE FRAG. % TOTAL');
    small := ffio.getn('COARSE FRAG. % GRAVEL');
    large := ffio.getn('COARSE FRAG. %COBBLES AND STONES');

    IF ( total IS NULL AND (small IS NULL OR small = 0) AND
        (large IS NULL OR large = 0) ) THEN
      p_return_code := 0;
    ELSIF ( total IS NULL ) THEN
      p_return_code := 1;
      p_return_message := ' On VEG Card ' || substr(ffio.getc( 'RECORD ID' ), 1, 2) ||
        ' for Soil Horizon ' || ffio.getc( 'HORIZON' ) ||
        ', the ***Total % Fragments' || '*** is missing where ' ||
        '% Gravel is ' || to_char(ffio.getn( 'COARSE FRAG. % GRAVEL' )) ||
        ' and % Cobbles and Stone is ' ||
        to_char(ffio.getn( 'COARSE FRAG. %COBBLES AND STONES' )) || ' .';
    ELSIF ( small IS NULL ) THEN
      p_return_code := 2;
      p_return_message := ' On VEG Card ' || substr(ffio.getc( 'RECORD ID' ), 1, 2) ||
        ' for Soil Horizon ' || ffio.getc( 'HORIZON' ) ||
        ', the ***% Gravel' || '*** is missing where ' ||
        '% Cobbles and Stone is ' || to_char(ffio.getn( 'COARSE FRAG. %COBBLES AND
        STONES' )) ||
        ' and Total % Fragments is ' ||
        to_char(ffio.getn( 'COARSE FRAG. % TOTAL' )) || ' .';
    ELSIF ( large IS NULL ) THEN
      p_return_code := 3;
      p_return_message := ' On VEG Card ' || substr(ffio.getc( 'RECORD ID' ), 1, 2) ||
        ' for Soil Horizon ' || ffio.getc( 'HORIZON' ) ||
        ', the ***% Cobbles and Stone' || '*** is missing where ' ||
        '% Gravel is ' || to_char(ffio.getn( 'COARSE FRAG. % GRAVEL' )) ||
        ' and Total % Fragments is ' ||

```

Record: VEG EP 12 4**Field:** COARSE FRAG. % GRAVEL**Datatype:** NUMBER**Position:** 24**Length:** 3**Optional:****Zero to Null:****Domain Name:** VEG SAMPLE PERCENTAGE 100**Value****Meaning**

0 - 100

00 - 99

000 - 100

Record: VEG EP 12 4**Field:** COARSE FRAG. %COBBLES AND S1**Datatype:** NUMBER**Position:** 27**Length:** 3**Optional:****Zero to Null:****Domain Name:** VEG SAMPLE PERCENTAGE 100**Value****Meaning**

0 - 100

00 - 99

000 - 100

Record: VEG EP 12 4**Field:** S6**Datatype:** CHAR**Position:** 30**Length:** 1**Optional:** Y**Zero to Null:****Record:** VEG EP 12 4**Field:** S7**Datatype:** CHAR**Position:** 31**Length:** 1**Optional:** Y**Zero to Null:****Record:** VEG EP 12 4**Field:** S8**Datatype:** CHAR**Position:** 32**Length:** 1**Optional:** Y**Zero to Null:****Record:** VEG EP 12 4**Field:** S9**Datatype:** CHAR**Position:** 33**Length:** 1**Optional:** Y**Zero to Null:****Record:** VEG EP 12 5**Field:** RECORD ID**Datatype:** NUMBER**Position:** 1**Length:** 3**Optional:** N**Zero to Null:****Record:** VEG EP 12 5**Field:** PAGE NUMBER

Datatype: NUMBER**Position:** 4**Length:** 2**Optional:** N**Zero to Null:****Record:** VEG EP 12 5**Field:** SEQUENCE NUMBER**Datatype:** NUMBER**Position:** 6**Length:** 4**Optional:** N**Zero to Null:****Record:** VEG EP 12 5**Field:** NOTES**Datatype:** CHAR**Position:** 10**Length:** 256**Optional:** Y**Zero to Null:****Record:** VEG ED 13 1**Field:** RECORD ID**Datatype:** NUMBER**Position:** 1**Length:** 3**Optional:** N**Zero to Null:****Record:** VEG ED 13 1**Field:** PAGE NUMBER**Datatype:** NUMBER**Position:** 4**Length:** 2**Optional:** N**Zero to Null:****Rule Name:** Veg 2000 Validate Page Number**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG ED 13 1**Field:** SEQUENCE NUMBER**Datatype:** NUMBER**Position:** 6**Length:** 4**Optional:** N**Zero to Null:****Record:** VEG ED 13 1**Field:** PL**Datatype:** CHAR**Position:** 10**Length:** 1**Optional:** N**Zero to Null:****Domain Name:** VEG SAMPLE PL - PLOT NOValueMeaning

E

I

M

N

S

W

Record: VEG ED 13 1**Field:** TOTAL PAGES

Datatype: NUMBER**Position:** 11**Length:** 2**Optional:** Y**Zero to Null:****Record:** VEG ED 13 1**Field:** SLOPE %**Datatype:** NUMBER**Position:** 13**Length:** 3**Optional:** N**Zero to Null:****Domain Name:** SLOPE PCTValue

0 - 150

Meaning**Record:** VEG ED 13 1**Field:** ASPECT**Datatype:** NUMBER**Position:** 16**Length:** 3**Optional:** N**Zero to Null:****Domain Name:** VEG SAMPLE ASPECT BRGValue

0 - 359

999

Meaning

Aspect Bearing

Level ground - less than 2% slope.

Rule Name: Veg 2000 Determine Aspect with Slope**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG ED 13 1**Field:** ELEVATION**Datatype:** NUMBER**Position:** 19**Length:** 4**Optional:** Y**Zero to Null:****Domain Name:** ELEVATIONValue

0 - 4000

Meaning**Record:** VEG ED 13 1**Field:** SURFACE SHAPE**Datatype:** CHAR**Position:** 23**Length:** 2**Optional:** N**Zero to Null:****Domain Name:** SURFACE SHAPEValue

CV

CX

ST

Meaning

concave

convex

straight

Record: VEG ED 13 1**Field:** MESO SLOPE**Datatype:** CHAR**Position:** 25**Length:** 1**Optional:** N**Zero to Null:****Domain Name:** MESO SLOPEValue

C

D

Meaning

Crest

Depression

F	Flat
L	Lower Slope
M	Mid Slope
T	Toe
U	Upper Slope

Record: VEG ED 13 1**Field:** MICROTOPOGRAPHY**Datatype:** CHAR**Position:** 26**Optional:** N**Length:** 2**Zero to Null:****Domain Name:** MICROTOPOGRAPHY**Value****Meaning**

EX	extremely mounded
MO	moderately mounded
SM	smooth
ST	strongly mounded

Record: VEG ED 13 1**Field:** ROCKY COBBLES/STONES**Datatype:** CHAR**Position:** 28**Optional:** N**Length:** 3**Zero to Null:****Domain Name:** VEG SAMPLE PERCENTAGE 100**Value****Meaning**

0	- 100
00	- 99
000	- 100

Record: VEG ED 13 1**Field:** ROCKY BEDROCK**Datatype:** CHAR**Position:** 31**Optional:** N**Length:** 3**Zero to Null:****Domain Name:** VEG SAMPLE PERCENTAGE 100**Value****Meaning**

0	- 100
00	- 99
000	- 100

Record: VEG ED 13 1**Field:** FLOOD HAZARD**Datatype:** CHAR**Position:** 34**Optional:** N**Length:** 1**Zero to Null:****Domain Name:** VEG SAMPLE FLOOD HAZARD**Value****Meaning**

F	Frequent
N	NotApplicable
O	Occasional

R

Occasional

Record: VEG ED 13 1**Field:** OPEN WATER FLOWING**Datatype:** CHAR**Position:** 35**Optional:** N**Length:** 3**Zero to Null:****Domain Name:** VEG SAMPLE PERCENTAGE 100ValueMeaning

0 - 100

00 - 99

000 - 100

Record: VEG ED 13 1**Field:** OPEN WATER STANDING**Datatype:** CHAR**Position:** 38**Optional:** N**Length:** 3**Zero to Null:****Domain Name:** VEG SAMPLE PERCENTAGE 100ValueMeaning

0 - 100

00 - 99

000 - 100

Record: VEG ED 13 1**Field:** SURFICIAL MATERIAL 1**Datatype:** CHAR**Position:** 41**Optional:** Y**Length:** 2**Zero to Null:****Domain Name:** SURFICIAL MATERIALValueMeaning

A anthropogenic

C colluvial

E eolian

F fluvial

FG glacial fluvial

L lacustrine

M moraine

O organic

R bedrock

V volcanic

W marine

Rule Name: Veg 2000 EP 12 ED 13 Check That the Soil Features and the Soil D**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG ED 13 1**Field:** SURFICIAL MATERIAL 2**Datatype:** CHAR**Position:** 43**Optional:** Y

		Length:		
	SURFICIAL MATERIAL		2	Zero to Null:
Domain Name:	Value	Meaning		
	A	anthropogenic		
	C	colluvial		
	E	eolian		
	F	fluvial		
	FG	glacial fluvial		
	L	lacustrine		
	M	moraine		
	O	organic		
	R	bedrock		
	V	volcanic		
	W	marine		
Record:	<u>VEG ED 13 1</u>	Field:	<u>DEPTH TO WATER TABLE</u>	
		Datatype:	NUMBER	
		Position:	45	Optional: Y
		Length:	3	Zero to Null:
Domain Name:	SOIL FEATURE DEPTH	Meaning		
	Value			
	0	- 300.0		
Record:	<u>VEG ED 13 1</u>	Field:	<u>DEPTH TO MOTTLES</u>	
		Datatype:	NUMBER	
		Position:	48	Optional: Y
		Length:	3	Zero to Null:
Domain Name:	SOIL FEATURE DEPTH	Meaning		
	Value			
	0	- 300.0		
Record:	<u>VEG ED 13 1</u>	Field:	<u>DEPTH TO ROOT REST PAN.</u>	
		Datatype:	NUMBER	
		Position:	51	Optional: Y
		Length:	3	Zero to Null:
Domain Name:	SOIL FEATURE DEPTH	Meaning		
	Value			
	0	- 300.0		
Record:	<u>VEG ED 13 1</u>	Field:	<u>DEPTH TO BEDROCK</u>	
		Datatype:	NUMBER	
		Position:	54	Optional: Y
		Length:	3	Zero to Null:
Domain Name:	SOIL FEATURE DEPTH	Meaning		
	Value			
	0	- 300.0		
Record:	<u>VEG ED 13 1</u>	Field:	<u>DEPTH TO FROZEN LAYER</u>	
		Datatype:	NUMBER	

		Position: 57	Optional: Y
Domain Name: SOIL FEATURE DEPTH		Length: 3	Zero to Null:
<u>Value</u>		<u>Meaning</u>	
0 - 300.0			
Record: VEG ED 13 1	Field: DEPTH TO CARBONATES		
		Datatype: NUMBER	
		Position: 60	Optional: Y
		Length: 3	Zero to Null:
Domain Name: SOIL FEATURE DEPTH			
<u>Value</u>		<u>Meaning</u>	
0 - 300.0			
Record: VEG ED 13 1	Field: DEPTH TO R1		
		Datatype: NUMBER	
		Position: 63	Optional: Y
		Length: 3	Zero to Null:
Domain Name: SOIL FEATURE DEPTH			
<u>Value</u>		<u>Meaning</u>	
0 - 300.0			
Record: VEG ED 13 1	Field: DEPTH TO R2		
		Datatype: NUMBER	
		Position: 66	Optional: Y
		Length: 3	Zero to Null:
Domain Name: SOIL FEATURE DEPTH			
<u>Value</u>		<u>Meaning</u>	
0 - 300.0			
Record: VEG ED 13 1	Field: DEPTH NOT APPLICABLE		
		Datatype: CHAR	
		Position: 69	Optional: Y
		Length: 1	Zero to Null:
Domain Name: INDICATOR			
<u>Value</u>		<u>Meaning</u>	
N		No (False)	
Y		Yes (True)	
Rule Name: Veg 2000 EP 12 1 Check for a Soil Features Depth When the So			
PL/SQL Source Code: Rule code has already been printed.			
Record: VEG ED 13 1	Field: HUMUS FORM		
		Datatype: CHAR	
		Position: 70	Optional: Y
		Length: 2	Zero to Null:
Domain Name: HUMUS FORM			
<u>Value</u>		<u>Meaning</u>	
MD		moder	
MM		mormoder	

MO	mor
MR	mullmoder
MU	mull
NA	not applicable; observed, but no assessment possib

Record: VEG ED 13 1**Field:** SOIL COLOUR**Datatype:** CHAR**Position:** 72**Length:** 2**Optional:** Y**Zero to Null:****Domain Name:** VEG SAMPLE SOIL COLOR**Value****Meaning**

D

Dark

L

Light

M

Medium

N

Not Applicable

Record: VEG ED 13 2**Field:** RECORD ID**Datatype:** NUMBER**Position:** 1**Length:** 3**Optional:** N**Zero to Null:****Record:** VEG ED 13 2**Field:** PAGE NUMBER**Datatype:** NUMBER**Position:** 4**Length:** 2**Optional:** N**Zero to Null:****Record:** VEG ED 13 2**Field:** SEQUENCE NUMBER**Datatype:** NUMBER**Position:** 6**Length:** 4**Optional:** N**Zero to Null:****Record:** VEG ED 13 2**Field:** HORIZON**Datatype:** CHAR**Position:** 10**Length:** 3**Optional:** Y**Zero to Null:****Rule Name:** Veg 2000 EP 12 ED 13 Soil Horizon Code**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 EP 12 ED 13 Check That the Soil Description Data Record**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 EP 12 ED 13 Check That the Soil Features and the Soil D**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG ED 13 2**Field:** DEPTH DISTANCE FROM ZERO**Datatype:** NUMBER**Position:** 13**Optional:** Y

Record: <u>VEG ED 13 3</u>	Length: 4 Optional: Zero to Null:
	Field: <u>RECORD ID</u> Datatype: NUMBER Position: 1 Length: 3 Optional: N Zero to Null:
Record: <u>VEG ED 13 3</u>	Field: <u>PAGE NUMBER</u> Datatype: NUMBER Position: 4 Length: 2 Optional: Y Zero to Null:
Record: <u>VEG ED 13 3</u>	Field: <u>SEQUENCE NUMBER</u> Datatype: NUMBER Position: 6 Length: 4 Optional: N Zero to Null:
Record: <u>VEG ED 13 3</u>	Field: <u>HORIZON</u> Datatype: CHAR Position: 10 Length: 3 Optional: Y Zero to Null:
Rule Name: Veg 2000 EP 12 ED 13 Soil Horizon Code PL/SQL Source Code: Rule code has already been printed. Rule Name: Veg 2000 EP 12 ED 13 Check That the Soil Features and the Soil D PL/SQL Source Code: Rule code has already been printed. Rule Name: Veg 2000 EP 12 ED 13 Check That the Soil Description Data Record PL/SQL Source Code: Rule code has already been printed.	
Record: <u>VEG ED 13 3</u>	Field: <u>DEPTH DISTANCE FROM ZERO</u> Datatype: NUMBER Position: 13 Length: 4 Optional: Y Zero to Null:
Rule Name: Veg 2000 EP 12 4 ED 13 3 Check That the Below Zero Depth Meas is PL/SQL Source Code: Rule code has already been printed.	
Record: <u>VEG ED 13 3</u>	Field: <u>TEXTURE</u> Datatype: CHAR Position: 17 Length: 4 Optional: Y Zero to Null:
Record: <u>VEG ED 13 3</u>	Field: <u>COARSE FRAG. % TOTAL</u>

Datatype: NUMBER**Position:** 21**Length:** 3**Optional:** Y**Zero to Null:****Domain Name:** VEG SAMPLE PERCENTAGE 100**Value****Meaning**

0 - 100

00 - 99

000 - 100

Rule Name: Veg 2000 EP 12 4 ED 13 3 Coarse Fragments Percentage Total**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG ED 13 3**Field:** COARSE FRAG. % GRAVEL**Datatype:** NUMBER**Position:** 24**Length:** 3**Optional:****Zero to Null:****Domain Name:** VEG SAMPLE PERCENTAGE 100**Value****Meaning**

0 - 100

00 - 99

000 - 100

Record: VEG ED 13 3**Field:** COARSE FRAG. %COBBLES AND S1**Datatype:** NUMBER**Position:** 27**Length:** 3**Optional:****Zero to Null:****Domain Name:** VEG SAMPLE PERCENTAGE 100**Value****Meaning**

0 - 100

00 - 99

000 - 100

Record: VEG ED 13 3**Field:** S1**Datatype:** CHAR**Position:** 30**Length:** 1**Optional:** Y**Zero to Null:****Record:** VEG ED 13 3**Field:** S2**Datatype:** CHAR**Position:** 31**Length:** 1**Optional:** Y**Zero to Null:****Record:** VEG ED 13 3**Field:** S3**Datatype:** CHAR**Position:** 32**Length:****Optional:** Y

Record: <u>VEG ED 13 3</u>	Field: <u>S4</u> Datatype: CHAR Position: 33 Length: 1	Zero to Null: Optional: Y Zero to Null:
Record: <u>VEG ED 13 4</u>	Field: <u>RECORD ID</u> Datatype: NUMBER Position: 1 Length: 3	Optional: N Zero to Null:
Record: <u>VEG ED 13 4</u>	Field: <u>PAGE NUMBER</u> Datatype: NUMBER Position: 4 Length: 2	Optional: N Zero to Null:
Record: <u>VEG ED 13 4</u>	Field: <u>SEQUENCE NUMBER</u> Datatype: NUMBER Position: 6 Length: 4	Optional: N Zero to Null:
Record: <u>VEG ED 13 4</u>	Field: <u>NOTES</u> Datatype: CHAR Position: 10 Length: 256	Optional: Y Zero to Null:
Record: <u>VEG ET 14 1</u>	Field: <u>RECORD ID</u> Datatype: NUMBER Position: 1 Length: 3	Optional: N Zero to Null:
Record: <u>VEG ET 14 1</u>	Field: <u>PAGE NUMBER</u> Datatype: NUMBER Position: 4 Length: 2	Optional: Y Zero to Null:

Rule Name: Veg 2000 Validate Page Number

PL/SQL Source Code: Rule code has already been printed.

Record: <u>VEG ET 14 1</u>	Field: <u>SEQUENCE NUMBER</u> Datatype: NUMBER Position: 6 Length: 4	Optional: N Zero to Null:
-----------------------------------	---	--

Record: VEG ET 14 1**Field:** PL**Datatype:** CHAR**Position:** 10**Length:** 1**Optional:** N**Zero to Null:****Domain Name:** VEG SAMPLE PL - PLOT NO**Value****Meaning**

E

I

M

N

S

W

Record: VEG ET 14 1**Field:** MEASUREMENT DATE**Datatype:** DATE**Position:** 11**Length:** 9**Date Mask:** YYYYMONDD**Optional:** N**Zero to Null:****Domain Name:** DATE MIN JAN 01 1997 - FULL**Value****Meaning**

1997JAN01 - 3999JAN01 minmum date for 1999 Ground Sampling Format - YYM

Record: VEG ET 14 1**Field:** CREW PERSON #1**Datatype:** CHAR**Position:** 20**Length:** 4**Optional:** N**Zero to Null:****Rule Name:** Veg 2000 Validate Person Against First Crew List**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG ET 14 1**Field:** CREW PERSON #2**Datatype:** CHAR**Position:** 24**Length:** 4**Optional:** Y**Zero to Null:****Rule Name:** Veg 2000 Validate Person Against First Crew List**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG ET 14 1**Field:** TOTAL PAGES**Datatype:** NUMBER**Position:** 28**Length:** 2**Optional:** Y**Zero to Null:****Record:** VEG ET 14 1**Field:** OVERALL LAYER A ESTIMATE**Datatype:** CHAR**Position:** 30**Optional:** Y

Domain	Value	Length:	Zero to Null:
		Meaning	
	-	zero percent	
	--	zero percent	
	---	zero percent	
	0	- 9	percent
	00	- 99	percent
	000	- 100	percent

Rule Name: Veg 2000 Validate Overall Layer Cover Percent

PL/SQL Source Code:

```

VarSrceDataValue VARCHAR2(10);
VarSrceDataPageNo NUMBER;
VarSrceDataFldName VARCHAR2(100);

BEGIN

  -- initialize this function's return value, code, and message
  p_return_value := NULL;
  p_return_code := 0;
  p_return_message := NULL;

  -- get the page number
  VarSrceDataPageNo := ffio.getn ( 'PAGE NUMBER' );

  -- get the field name
  VarSrceDataFldName := ffio.get_current_item_name;

  -- get the value of the current field
  VarSrceDataValue := ffio.getc;

  IF LTRIM(VarSrceDataValue) IN ( '-', '--', '---' ) THEN
    p_return_value := 0;
  ELSE
    BEGIN
      p_return_value := TO_NUMBER(LTRIM(VarSrceDataValue));

      IF ( p_return_value > 100 ) Then
        p_return_code := 2;
        p_return_message := 'on page ' || TO_CHAR ( VarSrceDataPageNo )
          || ', the value in ' || INITCAP ( VarSrceDataFldName )
          || ' is ' || p_return_value || '% which is above 100%.';

      End if;

      EXCEPTION WHEN OTHERS THEN
        p_return_code := 1;
        p_return_message := 'on page ' || TO_CHAR ( VarSrceDataPageNo )
          || ', the ' || INITCAP ( VarSrceDataFldName )
          || ' field is not recognized as a valid numeric value';
      END;
    END;
  END IF;

```

Record: VEG ET 14 1**Field:** OVERALL LAYER B1 ESTIMATE**Datatype:** CHAR**Position:** 33**Optional:** Y**Length:** 3**Zero to Null:****Domain Name:** VEG OVERALL LAYER ESTIMATE PCT**Value****Meaning**

-		zero percent
--		zero percent
---		zero percent
0	- 9	percent
00	- 99	percent
000	- 100	percent

Rule Name: Veg 2000 Validate Overall Layer Cover Percent**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG ET 14 1**Field:** OVERALL LAYER B2 ESTIMATE**Datatype:** CHAR**Position:** 36**Optional:** Y**Length:** 3**Zero to Null:****Domain Name:** VEG OVERALL LAYER ESTIMATE PCT**Value****Meaning**

-		zero percent
--		zero percent
---		zero percent
0	- 9	percent
00	- 99	percent
000	- 100	percent

Rule Name: Veg 2000 Validate Overall Layer Cover Percent**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG ET 14 2**Field:** RECORD ID**Datatype:** NUMBER**Position:** 1**Optional:** N**Length:** 3**Zero to Null:****Record:** VEG ET 14 2**Field:** PAGE NUMBER**Datatype:** NUMBER**Position:** 4**Optional:** N**Length:** 2**Zero to Null:****Record:** VEG ET 14 2**Field:** SEQUENCE NUMBER**Datatype:** NUMBER**Position:** 6**Optional:** N**Length:** 4**Zero to Null:****Record:** VEG ET 14 2**Field:** ITEM NO.**Datatype:** NUMBER

Position: 10
Length: 2
Optional: Y
Zero to Null:

Record: VEG ET 14 2

Field: LONG FORM SPECIES CD

Datatype: CHAR
Position: 12
Length: 8
Optional: Y
Zero to Null:

Rule Name: Veg 2000 ET 14 2 Validate Presence of Tree and Shrub Data

PL/SQL Source Code:

```
layer_a VARCHAR2(10);
layer_b1 VARCHAR2(10);
layer_b2 VARCHAR2(10);
layer_d VARCHAR2(10);
BEGIN
layer_a := ffio.getc('% COVER LAYER A');
layer_b1 := ffio.getc('% COVER LAYER B1');
layer_b2 := ffio.getc('% COVER LAYER B2');
layer_d := ffio.getc('DH')||ffio.getc('DW')||ffio.getc('DR');

IF ( ffio.getc('LONG FORM SPECIES CD') IS NOT NULL ) THEN

  IF ( layer_b1 IS NOT NULL ) THEN
    IF ( ffio.getc('AVG HT B1') IS NULL ) THEN
      p_return_code := 3;
      p_return_message := 'For item '||ffio.getc('ITEM NO.')||
        ' species '|| ffio.getc('LONG FORM SPECIES CD') ||
        ' a % Cover B1 layer is supplied with no average height.';
    END IF;
  ELSIF ( layer_b2 IS NOT NULL ) THEN
    IF ( ffio.getc('AVG HT B2') IS NULL ) THEN
      p_return_code := 4;
      p_return_message := 'For item '||ffio.getc('ITEM NO.')||
        ' species '|| ffio.getc('LONG FORM SPECIES CD') ||
        ' a % Cover B2 layer is supplied with no average height.';
    END IF;
  ELSIF ( layer_a || layer_d IS NULL ) THEN
    p_return_code := 1;
    p_return_message := 'For item '||ffio.getc('ITEM NO.')||' species '||
      ffio.getc('LONG FORM SPECIES CD') || ' no layer measurements are present';
    END IF;

  ELSIF ( layer_a || layer_b1 || layer_b2 || layer_d IS NULL ) THEN
    p_return_code := 2;
    p_return_message := 'For item '||ffio.getc('ITEM NO.')||' species is blank but
    '||
    'layer measurements are present';
  END IF;
END IF;
```

Record: VEG ET 14 2

Field: % COVER LAYER A

Datatype: CHAR
Position:

Length: 20
4

Optional: Y
Zero to Null:

Rule Name: Veg 2000 Convert Encrypted Cover Percentage - One Decimal Place

PL/SQL Source Code:

```

input_value VARCHAR2(10);
curr_name VARCHAR2(40);
itm VARCHAR2(4);
rec_id VARCHAR2(6);
result NUMBER;
BEGIN
input_value := ffio.getc;
p_return_code := 0;

BEGIN
IF ( input_value LIKE '%H' ) THEN
result := TO_NUMBER(TRANSLATE(input_value, ' H', ' ')) * .1;
ELSIF ( input_value LIKE '%T' ) THEN
result := TO_NUMBER(TRANSLATE(input_value, ' T', ' ')) * .01;
ELSE
result := TO_NUMBER(input_value);
END IF;
p_return_value := TO_CHAR(ROUND(result*100));
EXCEPTION
WHEN VALUE_ERROR THEN
p_return_code := 1;
rec_id := ffio.getc ( 'RECORD ID' );
curr_name := ffio.get_current_item_name;

IF rec_id = ('142') THEN
itm := ffio.getc ( 'ITEM NO.' );
p_return_message := ' The item number ' || itm ||
' with the field value of ****'
|| substr (input_value, 2, 1) ||
'*** should be either H or T for ' || curr_name || '.';

ELSIF rec_id = ('152') THEN
itm := ffio.getc ( 'ITEM NUMBER' );
p_return_message := ' The item number ' || itm ||
' with the field value of ****'
|| substr (input_value, 2, 1) ||
'*** should be either H or T for ' || curr_name || '.';

ELSE itm := NULL;
p_return_message :=
' The field value ****'
|| substr (input_value, 2, 1) ||
'*** should be either H or T for ' || curr_name || '.';
END IF;
p_return_value := NULL;

END;
```

Rule Name: Veg 2000 ET 14 2 Percent Cover B2 Layer Short Shrub Species Cod

PL/SQL Source Code:

```
--
-----
--
-- Sourced from rule 11006 from VRI Ground Sample Format 1997
--
-- Veg ET 14 2 Percent Cover B2 Layer Short Shrub Species Code
--
-- try to find a record in the group_taxon_memberships table that is associated
-- with a species group definition group code of 'shtshrub', 'talshrub', or
-- 'tree' and with a taxonomic name that has a long code equal to the source
-- data species code, that is, try to find a valid short shrub, tall shrub, or
-- tree species code record for the source data species code
--
-- if there is a value in the Percentage Cover B2 Layer field, then the source
-- data species code is treated as a short shrub species
--
-- if such a record cannot be found, try to find a record in the
-- group_taxon_memberships table that is associated with a species group
-- definition group code of 'shtshrub', 'talshrub', or 'tree', a taxonomic name
-- that has a long code that is null, and a group_taxon_memberships long code
-- equal to the source data species code, that is, try to find a valid unknown
-- short shrub, tall shrub, or tree species code record for the source data
-- species code
--
-- if successful in either search, return the skey of the record found in the
-- group_taxon_memberships table
--
-- Modifications
--
-- Nov 13, 1998 TMC - wo 673, reword the error message at error point 3
--
-- Nov 23, 1998 TMC - wo 673, add the species group codes 'talshrub' and 'tree'
-- to the 'shtshrub' code in matching the source data
-- species code; add the measurement date to the error
-- message; reword the exception messages when getting the
-- percentage value
--
-- Dec 7, 1998 TMC - wo 673, add the TO_CHAR function to the data collected
-- date in the invalid species error message to prevent the
-- OCX generation from raising a concatenation error
--
-- Dec 8, 1998 TMC - wo 673, remove the when others exception when getting the
-- percentage value so that the Oracle variable SQLERRM is
-- not referenced and then the OCX will generate without an
-- undeclared identifier error
--
-- Mar 31, 1999 CAW - reformat the rule for VRI 99 Sampling to use exception
-- handling to control logic flow.
--
-- Jul 23, 1999 CAW - support multiple UNKNOWN matches
--
-- Jul 25, 2000 LME - added species group cd 'NON-HERB MIX' to the species
-- code lookups
```

Record: VEG ET 14 2**Field:** % COVER LAYER B1**Datatype:** CHAR**Position:** 24**Optional:** Y**Length:** 4**Zero to Null:****Rule Name:** Veg 2000 Convert Encrypted Cover Percentage - One Decimal Place**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 ET 14 2 Percent Cover B2 Layer Short Shrub Species Cod**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG ET 14 2**Field:** AVG HT B1**Datatype:** NUMBER**Position:** 28**Optional:** Y**Length:** 2**Zero to Null:****Rule Name:** Veg 2000 Validate Tree and Shrub Layer Average Heights**PL/SQL Source Code:**

```

cur_item VARCHAR2(100);
cur_value VARCHAR2(100);
BEGIN
p_return_code := 0;
cur_item := ffio.get_current_item_name;
BEGIN
cur_value := TO_NUMBER(TRANSLATE(ffio.getc, '0 -', '0'))/10;
EXCEPTION WHEN VALUE_ERROR THEN
p_return_code := 1;
p_return_message := ' The value for ' || cur_item || ' is ****' ||
ffio.getc || '***. It is non numeric';
RETURN;
END;

IF ( cur_item LIKE '%B1' ) THEN
IF ( cur_value < 2 ) THEN
p_return_code := 2;
p_return_message := ' For Item Number ' || ffio.getc('ITEM NO.') ||
', the Species ' || ffio.getc('LONG FORM SPECIES CD') ||
' has a Average Height B1 value of ****' || cur_value ||
'***. B1 layer contains trees and shrubs between 2 and 10 meters.';
ELSIF ( cur_value > 10 ) THEN
p_return_code := 3;
p_return_message := ' For Item Number ' || ffio.getc('ITEM NO.') ||
', the Species ' || ffio.getc('LONG FORM SPECIES CD') ||
' has a Average Height B1 value of ****' || cur_value ||
'***. B1 layer contains trees and shrubs between 2 and 10 meters.';
END IF;
ELSIF ( cur_item LIKE '%B2' ) THEN
IF ( cur_value >= 2 ) THEN
p_return_code := 4;
p_return_message := ' For Item Number ' || ffio.getc('ITEM NO.') ||
', the Species ' || ffio.getc('LONG FORM SPECIES CD') ||
' has a Average Height B2 value of ****' || cur_value ||
'***. B2 layer contains trees and shrubs less than ' ||

```

Record: VEG ET 14 2**Field:** % COVER LAYER B2**Datatype:** CHAR**Position:** 30**Length:** 4**Optional:** Y**Zero to Null:****Rule Name:** Veg 2000 Convert Encrypted Cover Percentage - One Decimal Place**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 ET 14 2 Percent Cover B2 Layer Short Shrub Species Cod**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG ET 14 2**Field:** AVG HT B2**Datatype:** NUMBER**Position:** 34**Length:** 2**Optional:** Y**Zero to Null:****Rule Name:** Veg 2000 Validate Tree and Shrub Layer Average Heights**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG ET 14 2**Field:** AB_LAYER_PLOT SIZE**Datatype:** NUMBER**Position:** 36**Length:** 4**Optional:** Y**Zero to Null:****Domain Name:** VEG SAMPLE FIXED VALUE 10Value

10.00

Meaning

ten

Record: VEG ET 14 2**Field:** AB_LAYER_PLOT SHAPE**Datatype:** CHAR**Position:** 40**Length:** 8**Optional:** Y**Zero to Null:****Domain Name:** VEG SAMPLE FIXED SHAPEValue

CIRCULAR

Meaning**Record:** VEG ET 14 2**Field:** DH**Datatype:** CHAR**Position:** 48**Length:** 3**Optional:** Y**Zero to Null:****Rule Name:** Veg 2000 Convert Encrypted Cover Percentage - One Decimal Place**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 ET 14 2 Tree Species Code**PL/SQL Source Code:**

--

--

-- Veg ET 14 2 Tree Species Code

--

Record: VEG ET 14 2**Field:** DW**Datatype:** CHAR**Position:** 51**Length:** 3**Optional:** Y**Zero to Null:****Rule Name:** Veg 2000 Convert Encrypted Cover Percentage - One Decimal Place**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 ET 14 2 Tree Species Code**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG ET 14 2**Field:** DR**Datatype:** CHAR**Position:** 54**Length:** 3**Optional:** Y**Zero to Null:****Rule Name:** Veg 2000 Convert Encrypted Cover Percentage - One Decimal Place**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 ET 14 2 Tree Species Code**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG ET 14 2**Field:** D_PLOT_SIZE**Datatype:** NUMBER**Position:** 57**Length:** 4**Optional:** Y**Zero to Null:****Domain Name:** VEG SAMPLE FIXED VALUE 5.64Value

5.64

Meaning**Record:** VEG ET 14 2**Field:** D_PLOT_SHAPE**Datatype:** CHAR**Position:** 61**Length:** 8**Optional:** Y**Zero to Null:****Domain Name:** VEG SAMPLE FIXED SHAPEValue

CIRCULAR

Meaning**Record:** VEG ET 14 2**Field:** S1**Datatype:** CHAR**Position:** 69**Length:** 1**Optional:** Y**Zero to Null:****Record:** VEG ET 14 2**Field:** S2**Datatype:** CHAR**Position:** 70**Length:** 1**Optional:** Y**Zero to Null:****Record:** VEG ET 14 2**Field:** S3**Datatype:** CHAR

Position: 71
Length: 1
Optional: Y
Zero to Null:

Record: VEG ET 14 3

Field: RECORD ID

Datatype: NUMBER
Position: 1
Length: 3
Optional: N
Zero to Null:

Record: VEG ET 14 3

Field: PAGE NUMBER

Datatype: NUMBER
Position: 4
Length: 2
Optional: N
Zero to Null:

Record: VEG ET 14 3

Field: SEQUENCE NUMBER

Datatype: NUMBER
Position: 6
Length: 4
Optional: N
Zero to Null:

Record: VEG ET 14 3

Field: NOTES

Datatype: CHAR
Position: 10
Length: 256
Optional: Y
Zero to Null:

Record: VEG EH 15 1

Field: RECORD ID

Datatype: NUMBER
Position: 1
Length: 3
Optional: N
Zero to Null:

Record: VEG EH 15 1

Field: PAGE NUMBER

Datatype: NUMBER
Position: 4
Length: 2
Optional: Y
Zero to Null:

Rule Name: Veg 2000 Validate Page Number

PL/SQL Source Code: Rule code has already been printed.

Record: VEG EH 15 1

Field: SEQUENCE NUMBER

Datatype: NUMBER
Position: 6
Length: 4
Optional: N
Zero to Null:

Record: VEG EH 15 1

Field: PL

Datatype: CHAR
Position: 10
Optional: N

Length: 1
Zero to Null:

Domain Name: VEG SAMPLE PL - PLOT NO
Value
 E
 I
 M
 N
 S
 W
Meaning

Record: VEG EH 15 1 **Field:** PERCENT COVERAGE1
Datatype: CHAR
Position: 11 **Optional:** Y
Length: 3 **Zero to Null:**

Domain Name: VEG OVERALL LAYER ESTIMATE PCT
Value
 - zero percent
 -- zero percent
 --- zero percent
 0 - 9 percent
 00 - 99 percent
 000 - 100 percent
Meaning

Rule Name: Veg 2000 Validate Overall Layer Cover Percent
PL/SQL Source Code: Rule code has already been printed.

Record: VEG EH 15 1 **Field:** PERCENT COVERAGE2
Datatype: CHAR
Position: 14 **Optional:** Y
Length: 3 **Zero to Null:**

Domain Name: VEG OVERALL LAYER ESTIMATE PCT
Value
 - zero percent
 -- zero percent
 --- zero percent
 0 - 9 percent
 00 - 99 percent
 000 - 100 percent
Meaning

Rule Name: Veg 2000 Validate Overall Layer Cover Percent
PL/SQL Source Code: Rule code has already been printed.

Record: VEG EH 15 1 **Field:** PERCENT COVERAGE3
Datatype: CHAR
Position: 17 **Optional:** Y
Length: 3 **Zero to Null:**

Domain Name: VEG OVERALL LAYER ESTIMATE PCT
Value
Meaning

-		zero percent
--		zero percent
---		zero percent
0	- 9	percent
00	- 99	percent
000	- 100	percent

Rule Name: Veg 2000 Validate Overall Layer Cover Percent

PL/SQL Source Code: Rule code has already been printed.

Record: VEG EH 15 1

Field: PERCENT COVERAGE4

Datatype: CHAR

Position: 20

Length: 3

Optional: Y

Zero to Null:

Domain Name: VEG OVERALL LAYER ESTIMATE PCT

Value

Meaning

-		zero percent
--		zero percent
---		zero percent
0	- 9	percent
00	- 99	percent
000	- 100	percent

Rule Name: Veg 2000 Validate Overall Layer Cover Percent

PL/SQL Source Code: Rule code has already been printed.

Record: VEG EH 15 1

Field: MEASUREMENT DATE

Datatype: DATE

Position: 23

Length: 9

Date Mask: YYYYMONDD

Optional: N

Zero to Null:

Domain Name: DATE MIN JAN 01 1997 - FULL

Value

Meaning

1997JAN01 - 3999JAN01 minmum date for 1999 Ground Sampling Format - YYM

Record: VEG EH 15 1

Field: PLOT SIZE

Datatype: NUMBER

Position: 32

Length: 4

Optional: Y

Zero to Null:

Domain Name: VEG SAMPLE FIXED VALUE 5.64

Value

Meaning

5.64

Record: VEG EH 15 1

Field: PLOT SHAPE

Datatype: CHAR

Position: 36

Length: 8

Optional: Y

Zero to Null:

Domain Name: VEG SAMPLE FIXED SHAPE

Value

Meaning

CIRCULAR

Record: VEG EH 15 1**Field:** TOTAL PAGES**Datatype:** NUMBER**Position:** 44**Length:** 2**Optional:** Y**Zero to Null:****Record:** VEG EH 15 2**Field:** RECORD ID**Datatype:** NUMBER**Position:** 1**Length:** 3**Optional:** N**Zero to Null:****Record:** VEG EH 15 2**Field:** PAGE NUMBER**Datatype:** NUMBER**Position:** 4**Length:** 2**Optional:** N**Zero to Null:****Record:** VEG EH 15 2**Field:** SEQUENCE NUMBER**Datatype:** NUMBER**Position:** 6**Length:** 4**Optional:** N**Zero to Null:****Record:** VEG EH 15 2**Field:** ITEM NUMBER**Datatype:** NUMBER**Position:** 10**Length:** 2**Optional:** Y**Zero to Null:****Record:** VEG EH 15 2**Field:** LONG FORM SPECIES CD**Datatype:** CHAR**Position:** 12**Length:** 8**Optional:** Y**Zero to Null:****Rule Name:** Veg 2000 Validate Herb/Bryoid Layer Presence**PL/SQL Source Code:**

```

BEGIN
  IF ( ffio.getc('LONG FORM SPECIES CD') IS NOT NULL AND
    ffio.getc('HERB LAYER C') IS NULL AND
    ffio.getc('DH') IS NULL AND
    ffio.getc('DW') IS NULL AND
    ffio.getc('DR') IS NULL ) THEN
    p_return_code := 1;
    p_return_message :=
      'For Item number ' || ffio.getc( 'ITEM NUMBER' ) ||
      ' with the Species Code ' || ffio.getc('LONG FORM SPECIES CD') ||
      ', a ***"Herb Layer or Bryoid Layer"*** percentage must be present.';
  ELSE
    p_return_code := 0;
  END IF;

```

Record: VEG EH 15 2**Field:** HERB LAYER C

Datatype: CHAR**Position:** 20**Optional:** Y**Length:** 3**Zero to Null:****Rule Name:** Veg 2000 Convert Encrypted Cover Percentage - One Decimal Place**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 EH 15 2 Herb Layer C Species Code**PL/SQL Source Code:**

```
--
-----
--
-- Veg EH 15 2 Herb Layer C Species Code
--
-- try to find a record in the group_taxon_memberships table that is associated
-- with a species group definition group code of 'herb' and with a taxonomic
-- name that has a long code equal to the source data species code, that is,
-- try to find a valid herb species code record for the source data species
-- code
--
-- if there is a value in the Herb Layer C field, then the source data species
-- code is treated as a herb species code
--
-- if such a record cannot be found, try to find a record in the group_taxon_
-- memberships table that is associated with a species group definition group
-- code of 'herb', a taxonomic name that has a long code that is null, and a
-- group_taxon_memberships long code equal to the source data species code,
-- that is, try to find a valid unknown herb species code record for the source
-- data species code
--
-- if successful in either search, return the skey of the record found in the
-- group_taxon_memberships table
--
-- Modifications
--
-- Aug 27, 1998 TMC - wo 642, use a view in the taxonomic select statements
--
-- April 6, 1999 CAW - VRI 99 Ground Sampling. Introduced exception handling
-- to avoid selecting the data twice.
--
-- MAR 07, 2002 RGH - added condition to use new long codes if after Apr 1, 2001
for new view
-- June 27, 2002 CAW - changed condition date from 2001 to 2002
-----
--
VarDataCollectedDate DATE;
VarGtoHerbCntForGtoSpp NUMBER;
VarGtoHerbCntForTaxcSpp NUMBER;
VarGtoHerbSkeyForGtoSpp NUMBER;
VarGtoHerbSkeyForTaxcSpp NUMBER;
VarSrceDataHerbLayerC NUMBER;
VarSrceDataSppCd VARCHAR2(20);
VarInputPercentage VARCHAR2(20);
NewSpeciesCodeDate Date;
```

Record: VEG EH 15 2**Field:** DH**Datatype:** CHAR**Position:** 23**Length:** 3**Optional:** Y**Zero to Null:****Rule Name:** Veg 2000 Convert Encrypted Cover Percentage - One Decimal Place**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 EH 15 2 Bryoid or Tree Species Code**PL/SQL Source Code:**

```
--
-----
--
-- Veg EH 15 2 Bryoid or Tree Species Code
--
-- try to find a record in the group_taxon_memberships table that is associated
-- with a species group definition group code of 'moss', 'livrwort', 'lichen',
-- or 'tree', and with a taxonomic name that has a long code equal to the
-- source data species code, that is, try to find a valid bryoid or tree
-- species code record for the source data species code
--
-- if there is a value in the bryoid field (whichever is referencing this rule:
-- be it the Soil Dh, Wood Dw, or Rock Dr field), then the source data species
-- code is treated as a bryoid or tree species code
--
-- if such a record cannot be found, try to find a record in the group_taxon_
-- memberships table that is associated with a species group definition group
-- code of 'moss', 'livrwort', 'lichen', or 'tree', a taxonomic name that has a
-- long code that is null, and a group_taxon_memberships long code equal to the
-- source data species code, that is, try to find a valid unknown bryoid or
-- tree species code record for the source data species code
--
-- if successful in either search, return the skey of the record found in the
-- group_taxon_memberships table
--
-- Modifications
--
-- Aug 27, 1998 TMC - use a view in the taxonomic select statements
--
-- Jul 25, 2000 LME - added species group cd 'NON-HERB MIX' to the species
-- code lookups
--
-- Jul 26, 2000 LME - when looking up the UNKNOWN species the rule should
-- only find the UNKNOWN BRYOID entry
--
--
-- MAR 07, 2001 RGH - added condition to use new long codes if after Apr 1, 2001
for new view
-- Jun 14, 2002 CAW - changed condition code to use new codes for Apr 1, 2002.
-----
--
VarDataCollectedDate DATE;
VarGtoBryTreeCntForGtoSpp NUMBER;
VarGtoBryTreeCntForTaxcSpp NUMBER;
VarGtoBryTreeSkeyForGtoSpp NUMBER;
```

Record: VEG EH 15 2**Field:** DW**Datatype:** CHAR**Position:** 26**Length:** 3**Optional:** Y**Zero to Null:****Rule Name:** Veg 2000 Convert Encrypted Cover Percentage - One Decimal Place**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 EH 15 2 Bryoid or Tree Species Code**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG EH 15 2**Field:** DR**Datatype:** CHAR**Position:** 29**Length:** 3**Optional:** Y**Zero to Null:****Rule Name:** Veg 2000 Convert Encrypted Cover Percentage - One Decimal Place**PL/SQL Source Code:** Rule code has already been printed.**Rule Name:** Veg 2000 EH 15 2 Bryoid or Tree Species Code**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG EH 15 2**Field:** S1**Datatype:** CHAR**Position:** 32**Length:** 1**Optional:** Y**Zero to Null:****Record:** VEG EH 15 2**Field:** S2**Datatype:** CHAR**Position:** 33**Length:** 1**Optional:** Y**Zero to Null:****Record:** VEG EH 15 2**Field:** S3**Datatype:** CHAR**Position:** 34**Length:** 1**Optional:** Y**Zero to Null:****Record:** VEG EH 15 2**Field:** S4**Datatype:** CHAR**Position:** 35**Length:** 1**Optional:** Y**Zero to Null:****Record:** VEG EH 15 2**Field:** S5**Datatype:** CHAR**Position:** 36**Length:** 1**Optional:** Y**Zero to Null:****Record:** VEG EH 15 2**Field:** S6

Datatype: CHAR**Position:** 37**Length:** 1**Optional:** Y**Zero to Null:****Record:** VEG EH 15 3**Field:** RECORD ID**Datatype:** NUMBER**Position:** 1**Length:** 3**Optional:** N**Zero to Null:****Record:** VEG EH 15 3**Field:** PAGE NUMBER**Datatype:** NUMBER**Position:** 4**Length:** 2**Optional:** N**Zero to Null:****Record:** VEG EH 15 3**Field:** SEQUENCE NUMBER**Datatype:** NUMBER**Position:** 6**Length:** 4**Optional:** N**Zero to Null:****Record:** VEG EH 15 3**Field:** NOTES**Datatype:** CHAR**Position:** 10**Length:** 256**Optional:** Y**Zero to Null:****Record:** VEG EO 16 1**Field:** RECORD ID**Datatype:** NUMBER**Position:** 1**Length:** 3**Optional:** N**Zero to Null:****Record:** VEG EO 16 1**Field:** PAGE NUMBER**Datatype:** NUMBER**Position:** 4**Length:** 2**Optional:** Y**Zero to Null:****Rule Name:** Veg 2000 Validate Page Number**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG EO 16 1**Field:** SEQUENCE NUMBER**Datatype:** NUMBER**Position:** 6**Length:** 4**Optional:** N**Zero to Null:****Record:** VEG EO 16 1**Field:** PL**Datatype:** CHAR**Position:** 10**Optional:** N

Length: 1
Zero to Null:

Domain Name: VEG SAMPLE PL - PLOT NO
Value: E
 I
 M
 N
 S
 W
Meaning:

Record: VEG EO 16 1 **Field:** MEASUREMENT DATE
Datatype: DATE **Date Mask:** YYYYMONDD
Position: 11 **Optional:** N
Length: 9 **Zero to Null:**

Domain Name: DATE MIN JAN 01 1997 - FULL
Value: 1997JAN01 - 3999JAN01
Meaning: minmum date for 1999 Ground Sampling Format - YYM

Record: VEG EO 16 1 **Field:** CREW PERSON #1
Datatype: CHAR
Position: 20 **Optional:** N
Length: 4 **Zero to Null:**

Rule Name: Veg 2000 Validate Person Against First Crew List
PL/SQL Source Code: Rule code has already been printed.

Record: VEG EO 16 1 **Field:** CREW PERSON #2
Datatype: CHAR
Position: 24 **Optional:** Y
Length: 4 **Zero to Null:**

Rule Name: Veg 2000 Validate Person Against First Crew List
PL/SQL Source Code: Rule code has already been printed.

Record: VEG EO 16 1 **Field:** TOTAL PAGES
Datatype: NUMBER
Position: 28 **Optional:** Y
Length: 2 **Zero to Null:**

Record: VEG EO 16 1 **Field:** F1
Datatype: CHAR
Position: 30 **Optional:** Y
Length: 2 **Zero to Null:**

Rule Name: Veg 2000 EO 16 1 Lookup Site Disturbance Code

PL/SQL Source Code:

```
fld_value VARCHAR2(10);
BEGIN
  p_return_code := 0;
  p_return_value := null;
  fld_value := ffio.getc;
  IF ( fld_value IS NOT NULL ) THEN
    BEGIN
      select to_char(site_distr_cd_skey)
        into p_return_value
        from site_disturbance_cds
        where UPPER(site_dist_cd) = UPPER(fld_value);
    EXCEPTION
      WHEN NO_DATA_FOUND THEN
        p_return_code := 1;
      WHEN OTHERS THEN
        p_return_code := -2;
    END;
  END IF;
```

Record: VEG EO 16 1**Field:** F2**Datatype:** CHAR**Position:** 32**Length:** 2**Optional:** Y**Zero to Null:****Rule Name:** Veg 2000 EO 16 1 Lookup Site Disturbance Code**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG EO 16 1**Field:** F3**Datatype:** CHAR**Position:** 34**Length:** 2**Optional:** Y**Zero to Null:****Rule Name:** Veg 2000 EO 16 1 Lookup Site Disturbance Code**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG EO 16 1**Field:** F4**Datatype:** CHAR**Position:** 36**Length:** 2**Optional:** Y**Zero to Null:****Rule Name:** Veg 2000 EO 16 1 Lookup Site Disturbance Code**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG EO 16 1**Field:** PREVIOUS SPECIES 1

Datatype: CHAR**Position:** 38**Length:** 3**Optional:** Y**Zero to Null:****Rule Name:** Veg 2000 Tree Species Short Code**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG EO 16 1**Field:** PREVIOUS SPECIES 2**Datatype:** CHAR**Position:** 41**Length:** 3**Optional:** Y**Zero to Null:****Rule Name:** Veg 2000 Tree Species Short Code**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG EO 16 1**Field:** CURRENT SPECIES 1**Datatype:** CHAR**Position:** 44**Length:** 3**Optional:** Y**Zero to Null:****Rule Name:** Veg 2000 Tree Species Short Code**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG EO 16 1**Field:** CURRENT SPECIES 2**Datatype:** CHAR**Position:** 47**Length:** 3**Optional:** Y**Zero to Null:****Rule Name:** Veg 2000 Tree Species Short Code**PL/SQL Source Code:** Rule code has already been printed.**Record:** VEG EO 16 1**Field:** TREE HARVESTING**Datatype:** CHAR**Position:** 50**Length:** 1**Optional:** N**Zero to Null:****Domain Name:** TREE HARVESTING ACTIVITY CLASS**Value**

C

N

P

Meaning

clearcut

no tree harvest activity

partial tree extraction

Record: VEG EO 16 1**Field:** SNAGS**Datatype:** CHAR**Position:** 51**Length:** 1**Optional:** N**Zero to Null:****Domain Name:** SNAGS CLASS

<u>Value</u>	<u>Meaning</u>
N	>5 snags on a 25 m radius plot)
S	no snags present
Record: <u>VEG EO 16 1</u>	some snags, (1-5 on 25m radius plot)
	Field: <u>SNAGS/CWD</u>
	Datatype: CHAR
	Position: 52
Domain Name: SNAGS CWD DECAY CLASS	Length: 1
	Optional: N
	Zero to Null:
<u>Value</u>	<u>Meaning</u>
N	no snags/CWD debris meet the size criteria
S	some, limited occurrence of snags/CWD to 3 consecutive
Y	yes, snags represented in most decay classes
Record: <u>VEG EO 16 1</u>	Field: <u>CANOPY GAP</u>
	Datatype: CHAR
	Position: 53
Domain Name: CANOPY GAPS FROM MORTALITY CLASS	Length: 1
	Optional: N
	Zero to Null:
<u>Value</u>	<u>Meaning</u>
C	common, >10% area coverage of gaps in 25m radius plot
N	no canopy gaps from natural tree mortality
S	some, <10% area coverage in 25m radius plot
Record: <u>VEG EO 16 1</u>	Field: <u>VERTICAL STRUCTURE</u>
	Datatype: CHAR
	Position: 54
Domain Name: VERTICAL STRUCTURE CLASS	Length: 1
	Optional: N
	Zero to Null:
<u>Value</u>	<u>Meaning</u>
C	complex, uneven height
M	moderate, uneven height
S	simple, even height, limited coverage in up to 20%
Record: <u>VEG EO 16 1</u>	Field: <u>SUCCESSIONAL STABILITY</u>
	Datatype: CHAR
	Position: 55
Domain Name: SUCCESSIONAL STABILITY CLASS	Length: 1
	Optional: N
	Zero to Null:
<u>Value</u>	<u>Meaning</u>
I	intermediate, canopy species change inevitable
S	stable, understory tree species same as canopy
U	unstable, major species change likely
Record: <u>VEG EO 16 1</u>	Field: <u>TREE AGE</u>
	Datatype: CHAR
	Position: 56
Domain Name: TREE AGE SPECIES/SERIES CLASS	Length: 1
	Optional: N
	Zero to Null:

<u>Value</u>	<u>Meaning</u>
0	Old
Y	Young

Record: VEG EO 16 1**Field:** TREE SIZE**Datatype:** CHAR**Position:** 57**Length:** 1**Optional:** N
Zero to Null:**Domain Name:** TREE SIZE SPECIES/SERIES CLASS

<u>Value</u>	<u>Meaning</u>
I	Intermediate
L	Large
S	Small

Record: VEG EO 16 1**Field:** STRUCTURAL STAGE**Datatype:** CHAR**Position:** 58**Length:** 2**Optional:** Y
Zero to Null:**Rule Name:** Veg 2000 Lookup Seral Stage Class Code**PL/SQL Source Code:**

```

input_code VARCHAR2(10);
BEGIN
p_return_code := 0;
input_code := ffio.getc;
IF ( input_code IS NOT NULL ) THEN
BEGIN
SELECT seral_stage_cd
INTO p_return_value
FROM seral_stage_class_cds
WHERE seral_stage_cd = input_code;

EXCEPTION
WHEN NO_DATA_FOUND THEN
p_return_code := 1;
p_return_message := ' The Structural Stage code ' || input_code ||
' does not match any of the codes in the ' ||
'Seral Stage Class Code table.';
END;
END IF;

```

Record: VEG EO 16 1**Field:** % OLD TREES STILL ALIVE**Datatype:** NUMBER**Position:** 60**Length:** 3**Optional:** Y
Zero to Null:**Domain Name:** VEG SAMPLE PERCENTAGE 100

<u>Value</u>	<u>Meaning</u>

0 - 100

00 - 99

000 - 100

Record: VEG EO 16 1**Field:** OLD GROWTH FOREST**Datatype:** CHAR**Position:** 63**Optional:** Y**Length:** 4**Zero to Null:****Domain Name:** OLD GROWTH EVALUATION CLASS**Value****Meaning**

N

no old growth characteristics

S

some old growth characteristics evident

Y

yes, considered to exhibit old growth characteri

Record: VEG EO 16 1**Field:** PLOT SIZE**Datatype:** NUMBER**Position:** 67**Optional:** Y**Length:** 4**Zero to Null:****Domain Name:** VEG SAMPLE FIXED VALUE 25**Value****Meaning**

25.00

Record: VEG EO 16 1**Field:** R1**Datatype:** CHAR**Position:** 71**Optional:** Y**Length:** 20**Zero to Null:****Record:** VEG EO 16 1**Field:** R2**Datatype:** CHAR**Position:** 91**Optional:** Y**Length:** 20**Zero to Null:****Record:** VEG EO 16 1**Field:** R3**Datatype:** CHAR**Position:** 111**Optional:** Y**Length:** 20**Zero to Null:****Record:** VEG EO 16 1**Field:** S1 CODE**Datatype:** CHAR**Position:** 131**Optional:** Y**Length:** 2**Zero to Null:****Record:** VEG EO 16 1**Field:** S1 VALUE**Datatype:** CHAR**Position:** 133**Optional:** Y**Length:** 1**Zero to Null:**

<u>Record:</u> <u>VEG EO 16 1</u>	<u>Field:</u> <u>S2 CODE</u> Datatype: CHAR Position: 134 Length: 2 Optional: Y Zero to Null:
<u>Record:</u> <u>VEG EO 16 1</u>	<u>Field:</u> <u>S2 VALUE</u> Datatype: CHAR Position: 136 Length: 1 Optional: Y Zero to Null:
<u>Record:</u> <u>VEG EO 16 1</u>	<u>Field:</u> <u>S3 CODE</u> Datatype: CHAR Position: 137 Length: 2 Optional: Y Zero to Null:
<u>Record:</u> <u>VEG EO 16 1</u>	<u>Field:</u> <u>S3 VALUE</u> Datatype: CHAR Position: 139 Length: 1 Optional: Y Zero to Null:
<u>Record:</u> <u>VEG EO 16 2</u>	<u>Field:</u> <u>RECORD ID</u> Datatype: NUMBER Position: 1 Length: 3 Optional: N Zero to Null:
<u>Record:</u> <u>VEG EO 16 2</u>	<u>Field:</u> <u>PAGE NUMBER</u> Datatype: NUMBER Position: 4 Length: 2 Optional: N Zero to Null:
<u>Record:</u> <u>VEG EO 16 2</u>	<u>Field:</u> <u>SEQUENCE NUMBER</u> Datatype: NUMBER Position: 6 Length: 4 Optional: N Zero to Null:
<u>Record:</u> <u>VEG EO 16 2</u>	<u>Field:</u> <u>NOTES</u> Datatype: CHAR Position: 10 Length: 256 Optional: Y Zero to Null: