' //roac(Direction s/ecifies t(e com/onents of t(e irection vector from %(ic(t(e %el g!n a//roac(es t(e /art.

DirF)#2 Clam ing Direction

-lam/ing Direction s/ecifies t(e com/onents of t(e clam/ing force irection vector.

DirF)#2 'ormal Direction

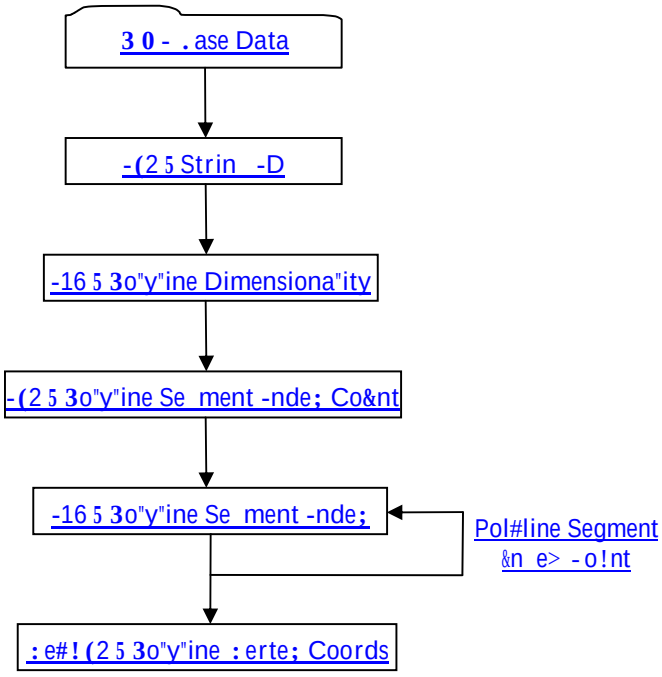
+ormal Direction s/ecifies t(e com/onents of t(e irection vector normal to t(e act!al s/ot %el .

/.#..#.1./.1 P9l)D Data

T(e P$& BD Data collection efines a ata format common to all BD 0ase P$& entities.

'long %it(t(e P$& :ase Data an String i entifier) t(is ata collection also incl! es non-te>t /ol#line ata efine 0# an arra# of in ices into an arra# of /ol#line segments /ac3e as 2D5BD verte> coor inates) s/ecif#ing %(ere eac(/ol#line segment 0egins an en s. ;o% /ol#lines are constr!cte from t(is in e> arra# an /ac3e verte> coor inates arra# is t(e same as t(at ill!strate in Fig!re 1DB of C.2.F.2.1.1.1.2.2 Te>t Pol#line Data.

!i &re 15(5 3 0 - (D Data data #o""e#tion



-om/lete escri/tion for P$& :ase Data can 0e fo!n in C.2.F.2.1.1.1.1 P$& :ase Data.

l)#2 String ID

String &D s/ecifies t(e i entifier for t(e c(aracter string. T(is i entifier is an in e> to a /artic!lar c(aracter string in t(e P$& String Ta0₱

/.#..#.1.;P9l 9easurement Point 6ntities

T(e P$& $eas!rement Point .ntities ata collection efines ata for a list of $eas!rement Point s#m0ols. $eas!rement Points are /re efine locations 1i.e. geometric entities or t(eoretical) 0!t meas!ra0le /oints) s!c(as s!rface locations2 %(ic(are meas!re on man!fact!re /arts to verif# t(e acc!rac# of t(e man!fact!ring /rocess.

Several ata fiel s of t(e P$& $eas!rement Point .ntities ata collection are onl# /resent if Version +!m0er) as efine in C.2.F.2P $& $anager $eta Data .lement) is greater t(an or eA!al to IDK.

!i &re 1555 3 0 - 0 eas&rement 3oint)ntities data #o""e#tion

-om/lete escri/tion for P$&BD Data can 0e fo!n in C.2.F.2.1.C.1 P$&BD Data.

l)#2 9P C

oint.

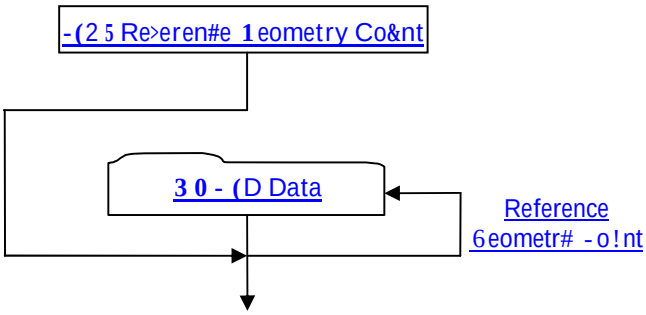
irection vector from %(ic(a - - $ 1-oor inate
ment.

irection vector anot(er $eas!rement Point on a
t on t(e first /art.

on vector normal to t(e act!al $eas!rement Point.

3o"y"ine Se ment -nde;P1Q	-m$"ied Re>eren#e 1eometry Ty$e
[[2	Pol#line
Z 2	Pol#gon

!i &re 15%5 3 0 - Re>eren#e 1eometry)ntities data #o""e#tion



-om/lete escri/tion for P$& BD Data can 0e fo!n in C.2.F.2.1.C.1 P$& BD Data.

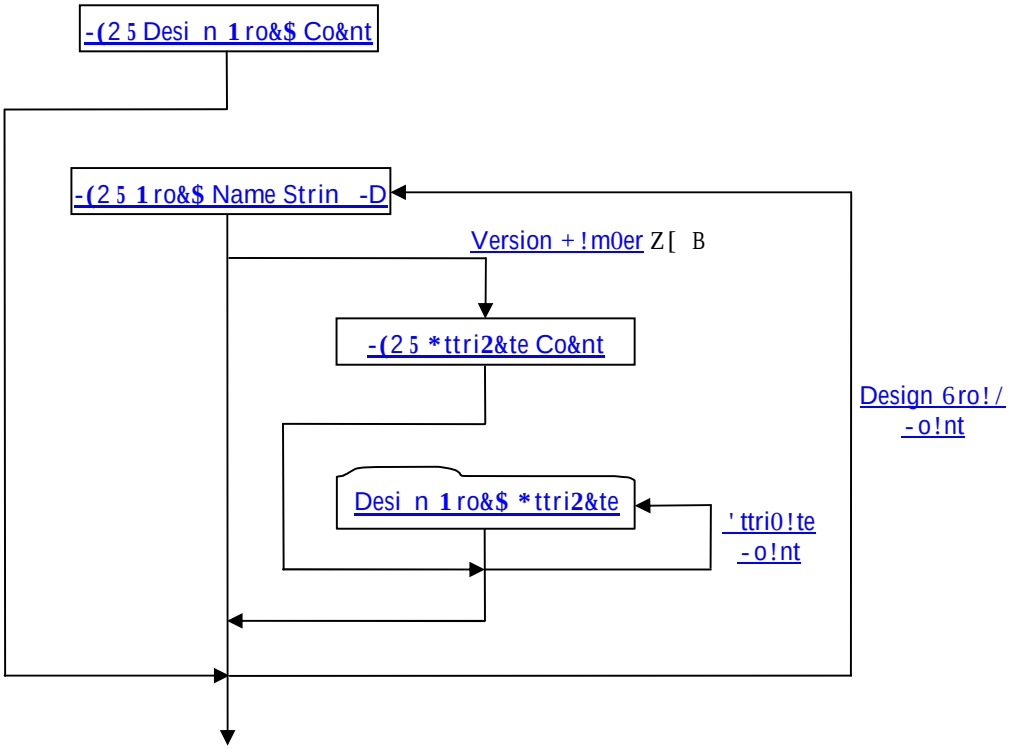
I)#2 Reference 5eometr! Count

Reference 6eometr# -o!nt s/ecifies t(e n!m0er of Reference 6eometr# entities.

/.#..#.1.1# P9I Design 5rou 6ntities

T(e P$& Design 6ro!/ .ntities ata collection efines ata for a list of Design 6ro!/s. Design 6ro!/s are collections of P$& create to organiHe a mo el into smaller s!0sets of information. T(is organiHation is ac(ieve via P$& 'ssociations 1see C.2.F.2.2 P$& 'ssociations2) %(ere s/ecific P$& entities are associate as I estinationsK to a Iso!rceK P$& Design 6ro!/.

!i &re 1585 3 0 - Desi n 1 ro&$)ntities data #o""e#tion



l)#2 Design 5rou Count

Design 6ro!/ -o!nt s/ecifies t(e n!m0er of Design 6ro!/ entities.

l)#2 5rou 'ame String ID

6ro!/ +ame String &D s/ecifies t(e i entifier for t(e gro!/ name c(aracter string. T(is i entifier is an in e> to a /artic!lar c(aracter string in t(e P$& String Ta0le as efine in C.2.F.2.D P$& String Ta0le. 'n i entifier val!e of I-1K in icates no string.

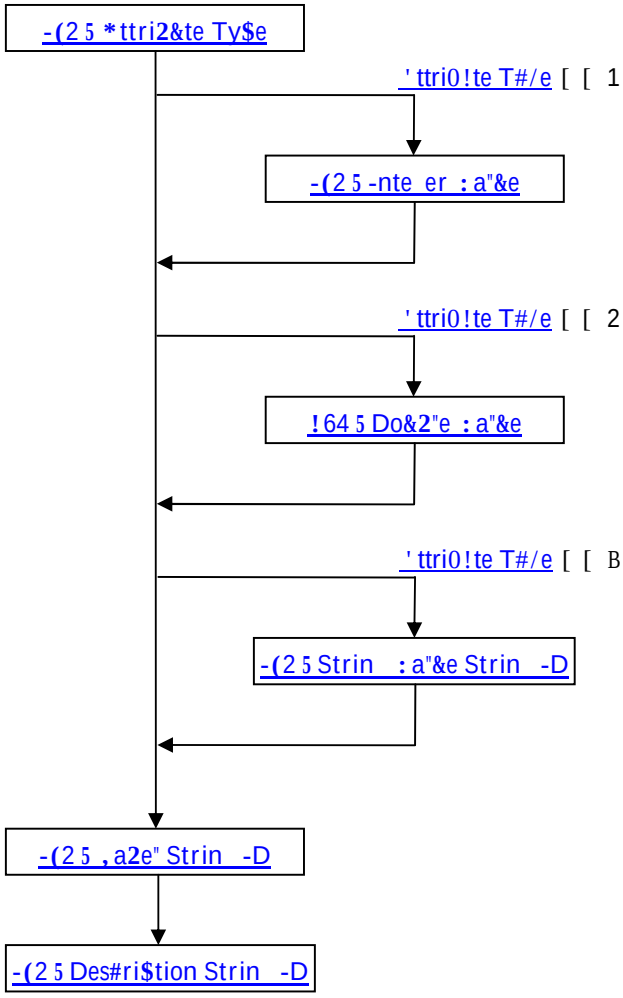
l)#2 +ttribute Count

'ttri0!te -o!nt s/ecifies t(e n!m0er of Design 6ro!/ 'ttri0!te ata collections

/.#..#.1.1#.1 Design 5rou +ttribute

T(e Design 6ro!/ 'ttri0!te ata collection efines a gro!/ /ro/ert#5attri0!te.

!i &re 15'5 Desi n 1 ro&$ *ttri2&te data #o""e#tion



I)# 2 +ttribute T! e

'ttri0!te T#/e s/ecifies t(e attri0!te t#/e. Vali t#/es incl! e t(e follo%ing*

[1	– &nteger
[2	– Do!0le
[B	– String

I)# 2 Integer Value

&nteger Val!e s/ecifies t(e val!e for IintegerK 'ttri0!te T#/es.

F., 2 Double Value

Do!0le Val!e s/ecifies t(e val!e for I o!0leK 'ttri0!te T#/es.

I)#2 String Value String ID

String Val!e String&D s/ecifies t(e string i entifier val!e for IstringK 'ttri0!te T#/es. T(is i entifier is an in e> to a /artic!lar c(aracter string in t(e P$& String Ta0le as efine in C.2.F.2.D P$& String Ta0le. 'n i entifier val!e of I-1K in icates no string.

I)#2 "abel String ID

"a0el String&D s/ecifies t(e string i entifier for t(e attri0!te la0el. T(is i entifier is an in e> to a /artic!lar c(aracter string in t(e P$& String Ta0le as efine in C.2.F.2.D P$& String Ta0le. 'n i entifier val!e of I-1K in icates no string.

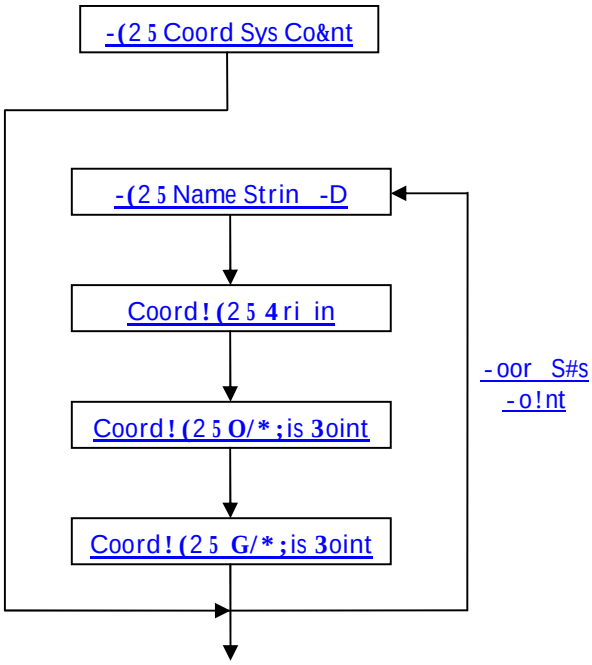
I)#2 Descri tion String ID

Descri/tion String&D s/ecifies t(e string i entifier for t(e attri0!te escri/tion. T(is i entifier is an in e> to a /artic!lar c(aracter string in t(e P$& String Ta0le as efine in C.2.F.2.D P$& String Ta0le. 'n i entifier val!e of I-1K in icates no string.

/.#..#.1.1) P9l Coor*inate S!stem 6ntities

T(e P$& -oor inate S#stem .ntities ata collection efines ata for a list of -oor inate S#stems.

!i &re 1605 3 0 - Coordinate System)ntities data #o""e#tion



I)#2 Coor* S!s Count

-oor S#s -o!nt s/ecifies t(e n!m0er of -oor inate S#stem entities.

I)#2 'ame String ID

+ame String &D s/ecifies t(e string i entifier for t(e -oor inate S#stem name. T(is i entifier is an in e> to a /artic!lar c(aracter string in t(e P$& String Ta0le as efine in C.2.F.2.D P$& String Ta0le. 'n i entifier val!e of I-1K in icates no string.

Coor*F)#2 4rigin

,rigin efines t(e origin of t(e coor inate s#stem.

Coor*F)#2 F-+:is Point

8-'>is Point efines a /oint along t(e 8-'>is of t(e coor inate s#stem.

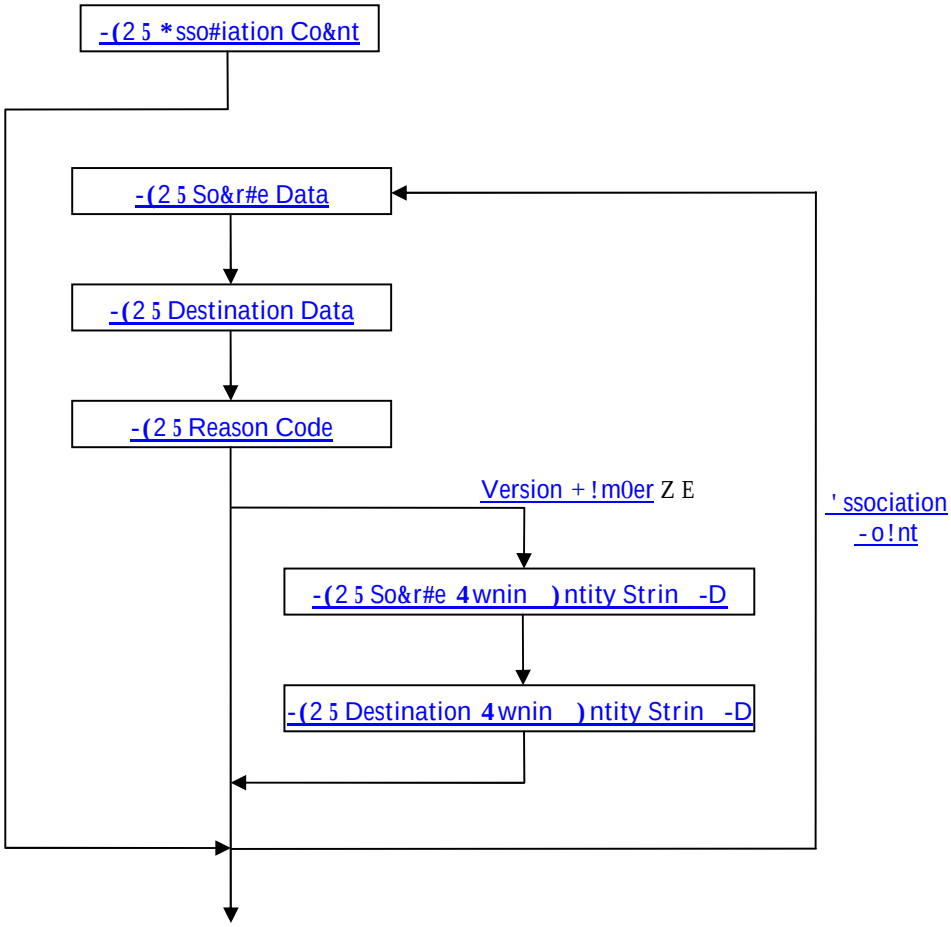
Coor*F)#2 @-+:is Point

9-'>is Point efines a /oint along t(e 9-'>is of t(e coor inate s#stem.

/.#..#.# P9I +ssociations

T(e P$& 'ssociations ata collection efines ata for a list of associations. 'n association efines a lin3 1Irelations(i/K2 0et%een t%o P$&) :-Re/) or =ireframe Re/ entities %(ere one entit# is efine as t(e Iso!rceK an t(e ot(er entit# is efine as t(e I estinationK.

!i &re 1615 **3 0** - *sso#iations data #o""e#tion



I)# 2 +ssociation Count

' ssociation - o!nt s/ecifies t(e n!m0er of associations.

I)# 2 Source Data

So!rce Data is a collection of so!rce entit# information enco e 5/ac3e %it(in a single &B2 !sing t(e follo%ing 0it allocation. ' ll !n oc!mente 0its are reserve .

: its 0 - 2B	So!rce . ntit# & entifier. T(e inter/retation of t(is i entifier ata is e/en ent !/on t(e val!e of : it B1 oc!mente 0elo%.
: its 2D -B0	So!rce . ntit# P $ & or : -Re/ t#/e. Vali t#/es incl! e t(e follo%ing* [0 – P $ & - Dimension [1 – P $ & - +ote [2 – P $ & - Dat!m Feat!re S#m0ol

	[B – P $ & - Dat!m Target [D – P $ & - Feat!re - ontrol Frame [E – P $ & - "ine = el [F – P $ & - S/ot = el [C – P $ & - $ eas!rement Point [8 – P $ & - S!rface Finis([9 – P $ & - "ocator Designator [10 – P $ & - Reference 6eometr# [11 – P $ & - - oor inate S#stem [12 – P $ & - Design 6ro!/ [1B – P $ & - 4ser 'ttri0!te [1D – :-Re/ - Verte> [1E – :-Re/ - . ge [1F – :-Re/ - Face [1C – P $ & - $ o el Vie% [18 – P $ & - 6eneric [19 – = ireframe Re/ - . ge [20 – P $ & - 4ns/ecifie t#/e [21 – Part &nstance
: it B1	&n irect & entifier Flag [0 Q Val!e in : its 0-2B is not t(e act!al - 'D i entifier) instea : its 0-2B is an in e> into t(e so!rce t#/e%s P $ & arra# or in e> of t(e e ge5face in :-Re/ or = ireframe Re/ for t(e so!rce entit#. [1 Q Val!e in : its 0-2B is not t(e act!al - 'D i entifier? instea : its 0-2B is an in e> into t(e list of - 'D Tags 1as oc!mente in C.2.F.2.C P $ & - 'D Tag Data2 i entif#ing t(e - 'D Tag 0elonging to t(e /artic!lar so!rce entit#.

I)#2 Destination Data

Destination Data is a collection of estination entit# information enco e 5/ac3e %it(in a single &B2. T(e enco ing sc(ema an inter/retation of t(is ata is t(e same as t(at oc!mente in So!rce Data.

I)#2 Reason Co*e

Reason - o e s/ecifies t(e IreasonK for t(e association. Vali Reason - o es incl! e t(e follo%ing*

[0	– 'ssociation is to t(e /rimar# entit# 0eing imensione
[1	– 'ssociation is to t(e secon ar# entit# 0eing imensione
[2	– 'ssociation is to t(e imension /lane
[E	– 'ssociation is to t(e entit# !se to s/ecif# t(e L-'>is of a coor inate s#stem
[10	– 'ssociation is to an entit# bassociate b to or bincl! e inb a P $ & s#m0ol
[11	– 'ssociation is to an entit# !se to battac(b a P $ & s#m0ol.
[12	– 'ssociation is to first entit# !se to Iattac(K a P $ & s#m0ol
[1B	– 'ssociation is to secon entit# !se to Iattac(K a P $ & s#m0ol
[1D	– S/ecif#ing P $ & gro!/ing) so!rce is P $ &5:-Re/ entit# an estination is esign gro!/.
[1E	– 'ssociation is to a %el line entit#

[1F	– ' ssociation is to a I(ot s/otK
[1C	– ' ssociation is to a c(il in a P $ & stac3
[C2	– ' ssociation is for P $ & miscellaneo!s relation.
[CB	– ' ssociation is for P $ & relate entit#.
[98	– ' ssociation is to s(o% t(e P $ & %(en associate $ o el Vie% is selecte . So!rce is t(e P $ &) an estination is $ o el Vie%.
[99	– ' ssociation is to s(o%5select P $ & :) if s(o%ing5selecting P $ & '. So!rce is P $ & ') an estination is P $ & :. T(is is ifferent from an Iattac(e K P $ &) %(ere t(e convention is to s(o% t(e P $ & visi0l# lin3e to one anot(er.
[100	– ' ssociation is to s(o% all /arts e>ce/t t(e associate /art instance. So!rce is t(e /art instance) an estination is $ o el Vie%

I)#2 Source 4(ning 6ntit! String ID

So!rce , %ning . ntit# String &D s/ecifies t(e string i entifier for t(e string %(ic(contains t(e !niA!e - ' D i entifier of t(e com/onent 1/art or assem0l#2 t(at o%ns t(e so!rce P $ & or :-Re/ entit#. T(is i entifier is an in e> to a /artic!lar c(aracter string in t(e P $ & String Ta0le as efine in C.2.F.2.D P $ & String Ta0le. ' n i entifier val!e of I-1K in icates no string an im/lies t(at t(e entit# is to 0e fo!n on t(e c!rrent no eUs P $ &5:-Re/5 = ireframe-Re/ segment. &t is vali for t(e so!rce o%ning entit# to 0e t(e same as t(e estination o%ning entit# 1i.e. an association 0et%een t%o P $ & or :-Re/ entities in t(e same /art5assem0l#2. T(is ata fiel is onl# /resent if Version + !m0er) as efine in C.2.F.2 P $ & $anager $eta Data . lement) is greater t(an IEK.

I)#2 Destination 4(ning 6ntit! String ID

Destination , %ning . ntit# String &D s/ecifies t(e string i entifier for t(e string %(ic(contains t(e !niA!e - ' D i entifier of t(e com/onent 1/art or assem0l#2 t(at o%ns t(e estination P $ & or :-Re/ entit#. T(is i entifier is an in e> to a /artic!lar c(aracter string in t(e P $ & String Ta0le as efine in C.2.F.2.D P $ & String Ta0le. ' n i entifier val!e of I-1K in icates no string an im/lies t(at t(e entit# is to 0e fo!n on t(e c!rrent no eUs P $ &5:-Re/5 = ireframe-Re/ segment. &t is vali for t(e so!rce o%ning entit# to 0e t(e same as t(e estination o%ning entit# 1i.e. an association 0et%een t%o P $ & or :-Re/ entities in t(e same /art5assem0l#2. T(is ata fiel is onl# /resent if Version + !m0er) as efine in C.2.F.2 P $ & $anager $eta Data . lement) is greater t(an IEK.

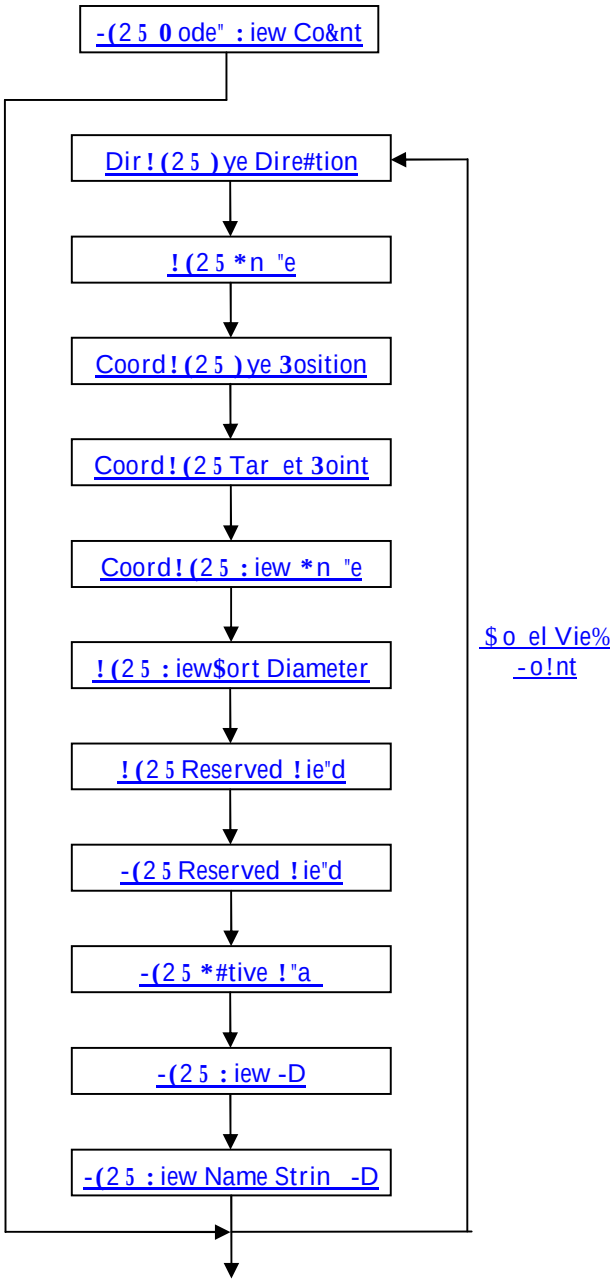
/.#..#.) P9l 1ser +ttributes

T(e P $ & 4ser ' ttri0!tes collection efines ata for a list of !ser attri0!tes. P $ & 4ser ' ttri0!tes are !se to a attri0!te ata to a /art5assem0l#. . ac(!ser attri0!te is com/ose of 3e#5val!e /air of strings.

JT F

J

!i &re 1645 3 0 - 0 ode" : iews data #o""e#tion



I)# 2 9 o*el Vie (Count

$ o el Vie% - o!nt s/ecifies t(e n!m0er of $ o el Vie%s.

DirF)#2 6!e Direction
.#e Direction s/ecifies t(e camera irection vector.

F)#2 +ngle
'ngle s/ecifies t(e camera rotation angle 1in egrees %(ere /ositive is co!nter-cloc3%ise2 a0o!t t(e .#e Direction. So t(is 'ngle in com0ination %it(t(e .#e Direction is eA!ivalent to s/ecif#ing a rotation !sing a>is-angle re/resentation.

Coor*F)#2 6!e Position
.#e Position s/ecifies t(e =-S coor inates of t(e e#e5camera Iloo3 fromK /osition.

Coor*F)#2 Target Point
Target Point s/ecifies t(e =-S coor inates of t(e e#e5camera Iloo3 atK /osition.

Coor*F)#2 Vie(+ngle
Vie% angle s/ecifies t(e 8) 9) L rotation angles 1in egrees2 of t(e mo elNs a>is. T(e rotations are efine %it(res/ect to an initial orientation %(ere t(e mo elNs a>is are aligne %it(t(e screenNs a>is 1i.e. \8 a>is /oints to rig(t) \9 a>is /oints !/) \L a>is /oints o!t at #o!2.

F)#2 Vie(ort Diameter
Vie%/ort Diameter s/ecifies t(e iameter in =-S coor inates of t(e largest /ossi0le circle t(at co!l 0e inscri0e %it(in vie%/ort. &f a large iameter val!e is s/ecifie)t(e mo el a//ears ver# small in relation to t(e vie%/ort? %(ereas if a small iameter val!e is s/ecifie a close-!/ 1IHoome -in2K vie% of t(e mo el res!lts.

F)#2 Reserve* Fiel*
Reserve Fiel is a ata fiel reserve for f!t!re JT format e>/ansion.

I)#2 Reserve* Fiel*
Reserve Fiel is a ata fiel reserve for f!t!re JT format e>/ansion

I)#2 +ctive Flag
'ctive Flag is a flag s/ecif#ing %(et(er t(is $o el Vie% is t(e IactiveK vie%. Vali val!es incl! e t(e follo%ing*

[0	– &s not t(e active $o el Vie%.
[1	– &s t(e active $o el Vie%

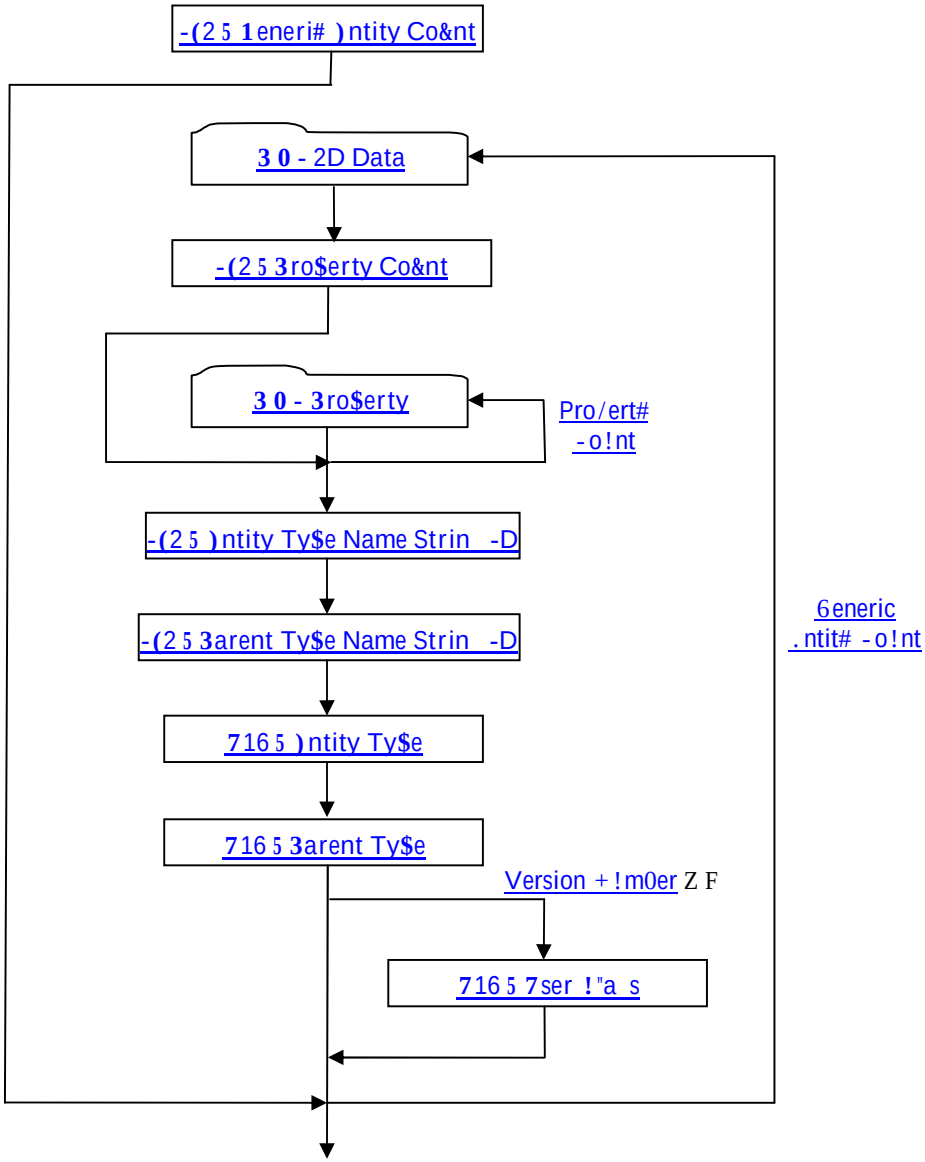
I)#2 Vie(ID
Vie%&D s/ecifies t(e $o el Vie% !niA!e i entifier.

I)#2 Vie('ame String ID
Vie% +ame String &D s/ecifies t(e string i entifier for t(e $o el Vie%Ns name. T(is i entifier is an in e> to a /artic!lar c(aracter string in t(e P$& String Ta0le as efine in C.2.F.2.D P$& String Ta0le. 'n i entifier val!e of I-1K in icates no string.

/.#..#.. 5eneric P9I 6ntities

T(e 6eneric P$& .ntities ata collection /rovi es a IgenericK format for efining vario!s P$& entit# t#/es) incl! ing !ser efine t#/es. T(e generic format efines t(e ata ma3ing !/ t(e P$& .ntit# t(ro!g(a com0ination of t(e P$& 2D Data collection an a list of P$& Pro/ert# ata collections.

!i &re 1655 1eneri# 3 0 -)ntities data #o""e#tion



-om/lete escri/tion for P$& 2D Data can 0e fo!n in C.2.F.2.1.1.1 P$& 2D Data.

I)#2 5eneric 6ntit! Count
6eneric .ntit# -o!nt s/ecifies t(e n!m0er of 6eneric P$& .ntities.

I)#2 Pro ert! Count
Pro/ert# -o!nt s/ecifies t(e n!m0er of P$& Pro/erties.

I)#2 6ntit! T! e 'ame String ID
.ntit# T#/e +ame String &D s/ecifies t(e string i entifier for t(e name of t(e 6eneric P$& .ntit# T#/e. T(is i entifier is an in e> to a /artic!lar c(aracter string in t(e P$& String Ta0le as efine in C.2.F.2.D P$& String Ta0le. 'n i entifier val!e of I-1K in icates no string.

I)#2 Parent T! e 'ame String ID
Parent T#/e +ame String &D s/ecifies t(e string i entifier for t(e name of t(e /arent 6eneric P$& .ntit# T#/e. T(is i entifier is an in e> to a /artic!lar c(aracter string in t(e P$& String Ta0le as efine in C.2.F.2.D P$& String Ta0le. 'n i entifier val!e of I-1K in icates no string.

11.2 6ntit! T! e
.ntit# T#/e s/ecifies t(e 6eneric P$& .ntit# T#/e. T(e vali .ntit# T#/e val!es 1in (e>a ecimal format2 are oc!mente in t(e follo%ing ta0le. +ote t(at for I!ser efine K 6eneric P$& .ntities a (e>a ecimal val!e of IO>011DK 1as oc!mente in ta0le 0elo%2 s(o!l 0e !se .

0>0001	– P$&1generall# onl# !se as a Parent T#/e2
0>0002	– =el
0>000D	– S/ot =el
0>0008	– "ine =el
0>0010	– 6roove =el
0>0011	– Fillet =el
0>0012	– Slot =el
0>001D	– . ge =el
0>0018	– 'rc S/ot =el
0>0020	– Resistance S/ot =el
0>0021	– Resistance Seam =el
0>0022	– Str!ct!ral ' (esive :ea S(a/e
0>002D	– Str!ct!ral ' (esive Ta/e S(a/e
0>0028	– Str!ct!ral ' (esive Dollo/ S(a/e
0>00D0	– $ec(anical -linc(-onnector
0>00D1	– S!rface Finis(
0>00D2	– $eas!rement Point
0>00DD	– Dat!m "ocator
0>00D8	– -ertification Point
0>0080	– 6eometric Dimensioning an Tolerancing
0>0081	– Feat!re -ontrol Frame
0>0082	– Dimension
0>008D	– Dat!m Feat!re S#m0ol
0>0088	– Dat!m Target
0>0100	– +ote

0>0101	– Face ' ttri0!te +ote
0>0102	– $o el Vie% "a0el +ote
0>010D	– -oor inate S#stem
0>0108	– Reference 6eometr#
0>0110	– Reference Point
0>0111	– Reference ' >is
0>0112	– Reference Plane
0>011D	– 4ser Define
0>0118	– $eas!rement "ocator
0>0120	– Dat!m Point
0>0121	– S!rface Vector $eas!rement Point
0>0122	– ;ole Vector $eas!rement Point
0>012D	– Trimme S(eet Vector $eas!rement Point
0>0128	– ;em Vector $eas!rement Point

11.2 Parent T!e

Parent T#/e s/ecifies t(e /arent 6eneric P$& .ntit# T#/e. T(e vali Parent T#/e val!es are t(e same as t(at oc!mente a0ove for .ntit# T#/e. T(e Parent T#/e is !se to create a class (ierarc(# of P$&%(en /resenting t(e P$& contents from a JT file.

11.21ser Flags

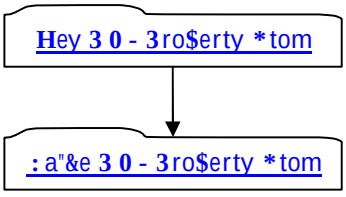
4ser Flags is a collection of flags. T(e flags are com0ine !sing t(e 0inar# , R o/erator an store vario!s state information for t(e 6eneric P$& .ntit#. ' ll !n oc!mente 0its are reserve .

0>0001	– S(o% P$& .ntit# Iflat to screen onl#K flag [0Q ' llo% P$& is/la# /lane to rotate %it(mo el. [1Q Dis/la# P$& entit# in t(e /lane of t(e screen) so t(at it oes not rotate %it(mo el.

/.#..#..1P9l Pro ert!

' P$& Pro/ert# ata collection consists of a 3e#5val!e /air an is !se to escri0e attri0!tes of 6eneric P$& .ntit# or ot(er s/ecific ata.

!i &re 1665 3 0 - 3ro$erty data #o""e#tion



:ot(<e# P$& Pro/ert# ' tom an Val!e P$& Pro/ert# ' tom (ave t(e same format as t(at oc!mente in C.2.F.2.F.1.1 P$& Pro/ert# ' tom.

'lt(o!g(t(ere is no reference com/liant reA!irements for %(at t(e P$& Pro/ert# 3e#5val!e /airs m!st 0e for eac(6eneric P$& .ntit# t#/e) t(ere are some common P$& Pro/ert# 3e#s an corres/on ing val!e formats t(at a//ear in JT File. T(e 0elo% ta0le oc!ments t(ese common P$& Pro/ert# 3e#s 1i.e. t(e 3e#s enco e string val!e2 an %(at t(e format of t(e val!e ata is in t(e val!es enco e string 1see C.2.F.2.F.1.1 P$& Pro/ert# 'tom for an e>/lanation of %(at is meant 0# Ienco e string val!eK for t(e I3e#K an Ival!eK ata2.

Ta2"e %5 Common 3ro$erty Heys and Their :a"&e)n#odin >ormats

KHeyM 3ro$erty *tom :a"&e Strin	K:a"&eM 3ro$erty *tom :a"&e Strin)n#odin !ormat	De#odin Notes
IP$&VPR,PV'+-;,RVP,&+Tb	IP> P# PHK	.ac(P>) P#) PH is a FB2 val!e !sing I c fK format
IP$&VPR,PV+,T.V;'SV4R"K	I0K or I1K	0 [[False? 1 [[Tr!e
IP$&VPR,PV+,R$'"VD&RK	ID> D# DHK	.ac(D>) D#) DH is a FB2 val!e !sing I c fK format
IP$&VPR,PV'PPR,'-;VD&RK	ID> D# DHK	.ac(D>) D#) DH is a FB2 val!e !sing I c fK format
IP$&VPR,PV-"'$P&+6VD&RK	ID> D# DHK	.ac(D>) D#) DH is a FB2 val!e !sing I c fK format
IP$&VPR,PV$.'SVD&RK	ID> D# DHK	.ac(D>) D#) DH is a FB2 val!e !sing I c fK format
IP$&VPR,PV-,,RDVD&RK	ID> D# DHK	.ac(D>) D#) DH is a FB2 val!e !sing I c fK format
IP$&VPR,PV$.'SV".V."K	IJK	&nteger re/resenting level n!m0er
IP$&Te>tForegro!n -olorK	IJK	;e>a ecimal integer re/resenting R6: color %(ere val!e (as I0>0000ggrrK form. T(e lo%-or er 0#te contains a val!e for t(e relative intensit# of re ? t(e secon 0#te contains a val!e for t(e relative intensit# of green? an t(e t(ir 0#te contains a val!e for t(e relative intensit# of 0l!e. T(e (ig(- or er 0#te m!st 0e Hero. T(e ma>im!m val!e for a single 0#te is 0>FF 1i.e. intensit# val!e is in t(e range R0*2EES2.
IP$&Te>t:ac3gro!n -olorK	IJK	Same as IP$&Te>tForegro!n -olorK
IP$&Te>t:ac3gro!n ,/acit#K	IJK	4nsigne ecimal integer re/resenting o/acit# /ercentage. 'ct!al o/acit# is* eco e J 5 100.0
IP$&Te>tS(o%:or erK	IJK	4nsigne ecimal integer* 0 [[False? 1 [[Tr!e
IP$&Te>tSiHeK	IJK	4nsigne ecimal integer re/resenting te>t siHe in !nits of /i>els.
IP$&Te>t&nPlaneK	IJK	4nsigne ecimal integer* 0 [[False? 1 [[Tr!e %(ere I1K in icates t(at te>t s(o!l 0e is/la#e in t(e /lane of t(e entit# so t(at it rotates %it(vie%.
IP$&6eometr#-olorK	IJK	Same as IP$&Te>tForegro!n -olorK
IP$&6eometr#=i t(K	IJK	4nsigne ecimal integer re/resenting line %i t(in !nits of /i>els.
-"&PV+,R$'"	IJ)J)JK	4se for .ntit# T#/e [I0>011DK an .ntit# T#/e +ame String [ISectionK to s/ecif# t(e normal to t(e cli//ing /lane. T(e cli//ing normal /oints to%ar t(e /iece of t(e mo el t(at %ill 0e cli//e a%a#. .ac(J is a FFD val!e !sing I c lfK format.
-"&PVP,S&T&,+	IJ)J)JK	4se for .ntit# T#/e [I0>011DK an .ntit# T#/e +ame String [ISectionK to s/ecif# one /oint on t(e cli//ing /lane. .ac(J is a FFD val!e !sing I c lfK

KHeyM 3ro$erty *tom :a"&e Strin	K:a"&eM 3ro$erty *tom :a"&e Strin)n#odin !ormat	De#odin Notes
		format.
TR'+SF,R$'T&,+V$'TR&8	IJ)J)J)J)J)J)J)J)J)J)J)J)JK	4se for .ntit# T#/e [IO>011DK an .ntit# T#/e +ame String [IPart TransformK to s/ecif# a transformation matri>. .ac(J is a FB2 val!e !sing I c fK format.

/.#..#...1.1 P9l Pro ert! +tom

P$& Pro/ert# 'tom ata collection re/resents t(e ata format for 0ot(t(e 3e# an val!e ata of a P$& Pro/ert# 3e#5val!e /air.

!i &re 16%5 3 0 - 3ro$erty *tom data #o""e#tion

9bString 2 Value

Val!e s/ecifies t(e /ro/ert# atom val!e enco e into a String. See Ta0le C* -ommon Pro/ert# <e#s an T(eir Val!e .nco ing formats a0ove for enco ing formats of t(e Val!e string.

1)#2 0i**en Flag

;i en Flag s/ecifies if t(e /ro/ert# is I(i enK or not. 'JT file rea er co!l !se t(is flag to control %(et(er rea /ro/erties s(o!l 0e e>/ose to t(e en !ser of t(e a//lication rea ing t(e JT file. Vali val!es incl! e t(e follo%ing*

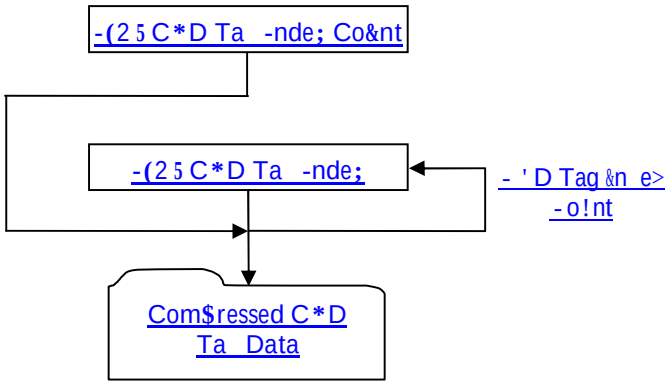
[0	– Pro/ert# is not (i en.
[1	– Pro/ert# is (i en.

/.#..#./ P9l C+D Tag Data

T(e P$& -'D Tag Data collection contains t(e list of /ersistent &Ds) as efine in t(e -'D Snd W nm0 0 0 r g mq m10 0

&f P$& -'D Tag Data collection is /resent) t(ere %ill 0e a -'D Tag for eac(P$& entit# as s/ecifie 0# t(e 0elo% oc!mente -'D Tag&n e> -o!nt form!la.

!i &re 1685 3 0 - C*D Ta Data data #o""e#tion



-om/lete escri/tion for -om/resse -'D Tag Data can 0e fo!n in 8.1.11 -om/resse -'D Tag Data.

I)#2 C+D Tag In*e: Count

-'D Tag&n e> -o!nt s/ecifies t(e total n!m0er of -'D Tag in ices. T(is val!e m!st 0e eA!al to t(e s!mmation of t(e /revio!sl# rea co!nt val!es for all t(e P$& entities s!//orting -'D Tags. T(e form!la is as follo%s*

-'D Tag&n e> -o!nt ["ine =el -o!nt \ S/ot =el -o!nt \ SF -o!nt \ $P -o!nt \ Reference 6eometr# -o!nt \ Dat!m Target -o!nt \ F-F -o!nt \ "ocator -o!nt \ Dimension -o!nt \ DFS -o!nt \ +ote -o!nt \ $o el Vie% -o!nt \ Design 6ro!/ -o!nt \ -oor S#s -o!nt \ 6eneric .ntit# -o!nt

I)#2 C+D Tag In*e:

-'D Tag&n e> s/ecifies an in e> into a list of -'D Tags) i entif#ing t(e -'D Tag 0elonging to a /artic!lar P$& entit#. T(ere %ill 0e a total of -'D Tag&n e> -o!nt n!m0er of -'D Tag&n ices an t(e or er of t(e in ices %ill 0e as efine 0# t(e a0ove oc!mente -'D Tag&n e> -o!nt form!la 1i.e. "ine =el -'D Tag&n ices are first) follo%e 0# t(e S/ot =el -'D Tag&n ices) follo%e 0# t(e S!rface Finis(-'D Tag&n ices) etc.2

/.#./ P9I Data Segment

T(e P$& $anager $eta Data .lement 1as oc!mente in C.2.F.2 P$& $anager $eta Data .lement2 can sometimes also 0e re/resente in a P$& Data Segment. T(is can occ!r %(en a /re JT 8 version file is migrate to JT 8.1 version file. So from a /arsing /oint of vie% a P$& Data Segment s(o!l 0e treate e>actl# t(e same as a C.2.F $eta Data Segment.

8 Data Com ression an* 6nco*ing

T(e JT File format !tiliHes 0est-in-class com/ression an enco ing algorit(ms to /ro !ce com/act an efficient re/resentations of ata. T(e t#/es of com/ression algorit(ms s!//orte 0# t(e JT format var# from stan ar ata t#/e agnostic L"&: eflation to a vance arit(metic algorit(ms t(at e>/loit 3no%le ge of t(e c(aracteristics of t(e ata t#/es t(e# are com/ressing. Some of t(e JT format ata collections are al%a#s store in a com/resse format) %(ereas ot(er ata collections s!//ort m!lti/le com/ression storage formats t(at A!alitativel# var# from I"osslessK com/ression to more aggressive strategies t(at em/lo# Iloss#K com/ression. T(is s!//ort 0# t(e JT format of var#ing A!alitative levels of com/ression allo%s /ro !cers of JT ata to fine t!ne t(e tra e off 0et%een com/ression ratio an fi elit# of t(e ata.

&n some instances) ata ma# 0e enco e 5com/resse !sing m!lti/le tec(niA!es a//lie on to/ of one anot(er in a serial fas(ion 1i.e. enco ing a//lie to t(e o!t/!t of anot(er enco er2. , ne common e>am/le of t(is m!lti/le enco ing is %(en an arra#5vector of floating /oint ata is first A!antiHe into some integer co es an t(en t(ese res!lting integer co es are f!rt(er com/resse 5enco e !sing a ;!ffman or 'rit(metic -, D. - 1see 8.2 .nco ing 'lgorit(ms2.

:e#on t(e ata collection s/ecific com/ression5enco ing) some JT format Data Segment t#/es 1see C.1.B Data Segment2 also s!//ort (aving a L"&: com/ression con itionall# a//lie to all t(e 0#tes of information /ersiste %it(in t(e segment. So in ivi !al fiel s or collections of ata ma# first (ave ata t#/e s/ecific enco ing5com/ression algorit(ms a//lie to t(em) an t(en if t(eir Data Segment t#/e s!//orts it) t(e res!lting ata ma# 0e a itional com/resse !sing a L"&: eflation algorit(m.

=(et(er) an at %(at A!alitative level) a /artic!lar Data SegmentNs ata is com/resse 5enco e is in icate t(ro!g(com/ression relate ata val!es store as /art of t(e /artic!lar Data Segment storage format. &n general) aggressive a//lication of a vance com/ression5enco ing tec(niA!es is reserve for t(e (eav#-%eig(t ren era0le geometric ata 1e.g. triangles an %ireframe lines2 %(ic(can e>ist in a JT File.

T(e follo%ing sections oc!ment t(e format of t(e ata com/ression5enco ing %it(in t(e JT file. 'long %it(oc!menting t(e format) a tec(nical escri/tion of t(e vario!s com/ression5enco ing algorit(ms is incl! e an an e>am/le im/lementation of t(e eco ing /ortion of t(e algorit(ms can 0e fo!n %it(in '//en i> -*Deco ing 'lgorit(ms Q 'n &m/lementation.

8.1 Common Com ression Data Collection Formats

For convenience an 0revit# in oc!menting t(e JT format) t(is section of t(e reference oc!ments t(e format for several common I ata com/ression5enco ingK relate ata collections t(at can e>ist in t(e JT format. 9o! %ill fin references to t(ese common com/ression ata collections in t(e C.2 Data Segments section of t(e oc!ment.

8.1.1 Int)# Com resse* Data Pac?et

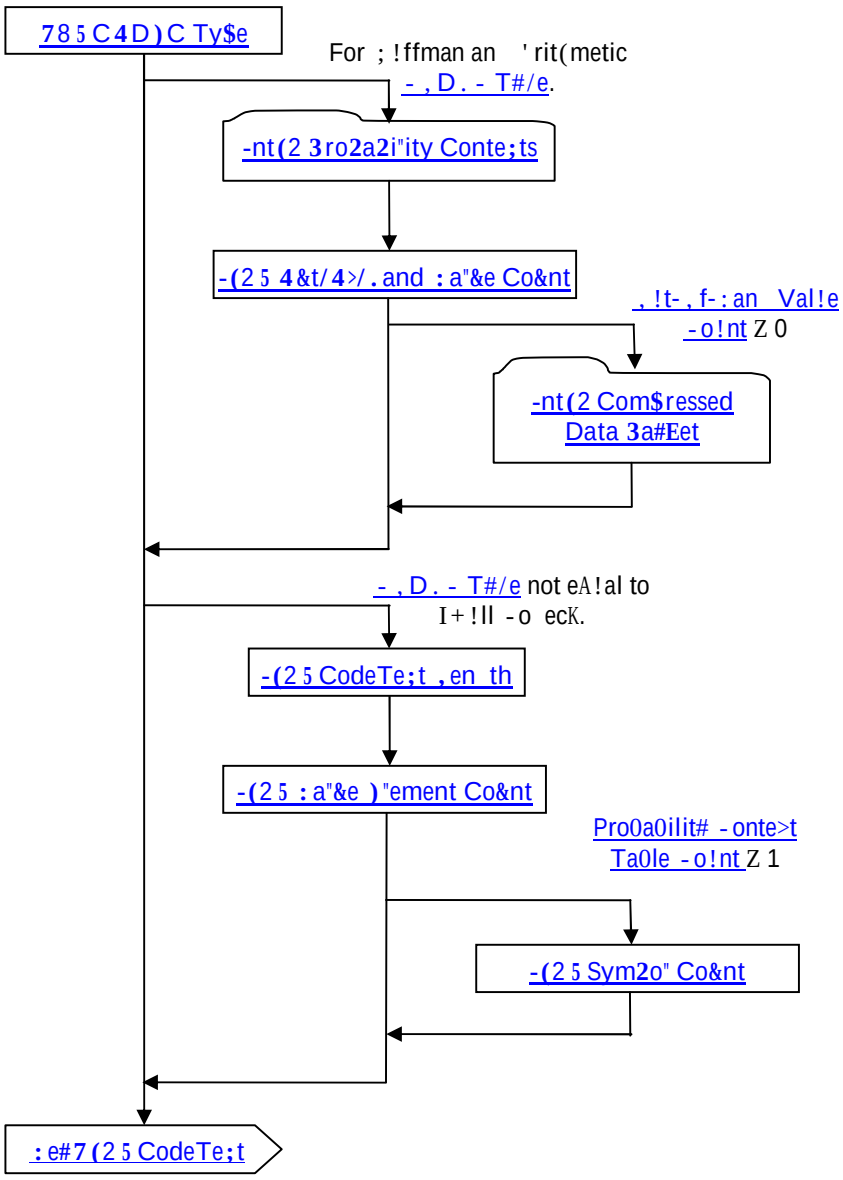
T(e &ntB2 -om/resse Data Pac3et collection re/resents t(e format !se to enco e5com/ress a collection of ata into a series of &ntB2 0ase s#m0ols. +ote t(at t(e &ntB2 -om/resse Data Pac3et collection can in itself contain anot(er &ntB2 -om/resse Data Pac3et collection if t(ere are an# I, !t-, f-: an ata.K &n t(e conte>t of t(e JT format ata com/ression algorit(ms an &ntB2 -om/resse Data Pac3et) Io!t-of-0an ataK (as t(e follo%ing meaning.

-, D.-s li3e arit(metic an ;!ffman 1see 8.2 .nco ing 'lgorit(ms for tec(nical escri/tion2 e>/loit t(e statistics /resent in t(e relative freA!encies of t(e val!es 0eing enco e . Val!es t(at occ!r freA!entl# eno!g(allo% 0ot(of t(ese met(o s to enco e eac(of t(e val!es as a Is#m0olK in fe%er 0its t(at it %o!l ta3e to enco e t(e val!e itself. Val!es t(at occ!r too infreA!entl# to ta3e a vantage of t(is /ro/ert# are %ritten *aside* into t(e Io!t-of-0an ataK arra# to 0e enco e se/aratel#. 'n Iesca/eK s#m0ol is enco e in t(eir /lace as a /lace(ol er in t(e /rimal -, D.- 1note) see IS#m0olK ata fiel efinition in 8.1.1.2 &ntB2 Pro0a0ilit# -onte>t Ta0le .ntr# for f!t(er etails on t(e re/resentation of Iesca/eK s#m0ol2.

.ssentiall# t(e Io!t-of-0an ataK is t(e (ig(-entro/# resi !e left over after t(e -, D.-s (ave sA!eeHe all t(e a vantage o!t of t(e original ata stream t(at t(e# can. ;o%ever) t(is Io!t-of-0an ataK is sent 0ac3 aro!n for anot(er /ass 0eca!se sometimes t(ere are *different* statistics to 0e ta3en a vantage of. =(en all ot(er co ing o/tions (ave 0een e>(a!ste) t(e 0itlengt(-, D.- is invo3e . T(e 0itlengt(-, D.- irectl# enco es all val!es given to it) oes not reA!ire a /ro0a0ilit# conte>t) an (ence never /ro !ces a itional Io!t-of-0an ataK. T(e 0#te sto/s t(ere) in ot(er %or s.

&n some cases) all val!es ma# %ritten bo!t of 0an b %(en t(e -o ec cannot /erform *any* !sef!l com/ression. &n t(is case) t(e enco e &B2*-o eTe>t "engt(fiel %ill 0e 0) an t(e &B2*, !t-, f-:an Val!e -o!nt %ill 0e eA!al to &B2*Val!e .lement -o!nt. T(e im/lie action in t(is case is to merel# co/# t(e , !t-, f-:an val!e ata into t(e o!t/!t Val!e .lement arra# instea of invo3ing t(e -o ec.

!i &re 16'5 -nt(2 Com$ressed Data 3a#Eet data #o""e#tion



182C4D6C T! e

-,D.- T#/e s/ecifies t(e algorit(m !se to enco e5 eco e t(e ata. See 8.2 .nco ing 'lgorit(ms for com/lete e>/lanation of eac(of t(e enco ing algorit(ms.

[0	– +!ll -,D.-
[1	– :itlengt(-,D.-

[2	– ; !ffman - , D . -
[B	– 'rit(metic - , D . -

I)# 2 4ut-4f-3an* Value Count

, !t- , f- : an Val!e -o!nt s/ecifies t(e n!m0er of val!es t(at are I , !t- , f- : an .K T(is ata fiel is onl# /resent for ; !ffman an 'rit(metic - , D . - T#/es.

I)# 2 Co*eTe:t "engt%

-o eTe>t "engt(s/ecifies t(e total n!m0er of 0its of -o eTe>t ata 1-o eTe>t ata fiel is escri0e 0elo%2. T(is ata fiel is onl# /resent if - , D . - T#/e is not eA!al to I+!ll - , D . -.K

I)# 2 Value 6lement Count

Val!e . lement -o!nt s/ecifies t(e n!m0er of val!es t(at t(e - , D . - is e>/ecte to eco e 1i.e. itNs li3e t(e Ilengt(K fiel %ritten if #o!Nre 7!st %riting o!t a vector of integers2. T(is ata fiel is onl# /resent if - , D . - T#/e is not eA!al to I+!ll - , D . -.K 4/on com/letion of eco ing t(e -o eTe>t ata fiel 0elo%) t(e n!m0er of eco e Val!es s(o!l 0e eA!al to Val!e . lement -o!nt. =(en onl# a single Pro0a0ilit# -onte>t Ta0le is !se) Val!e . lement -o!nt %ill also 0e eA!al to t(e n!m0er of S#m0ols eco e !/on com/letion of eco ing.

I)# 2 S!mbol Count

=(en t%o Pro0a0ilit# -onte>t Ta0les are 0eing !se) S#m0ol -o!nt s/ecifies t(e n!m0er of S#m0ols to 0e eco e 0# t(e 'rit(metic - , D . -. T(ere is a s!0tlet# /resent in t(e met(o -o ecDriver**a , !t/!tS#m0ol12 %(en it is /asse an . sca/e s#m0ol. , nl# if t(e -o ec is !sing Pro0a0ilit# -onte>t Ta0le 0 %(en it receives an . sca/e s#m0ol oes it emit a Val!e from t(e b , !t- , f- : an b ata arra#. : eca!se of t(is s!0tlet#) t(e n!m0er of S#m0ols eco e can 0e larger t(an t(e n!m0er of Val!es /ro !ce) t(!s t(e reason for %riting t(is fiel istinct from Val!e . lement -o!nt.

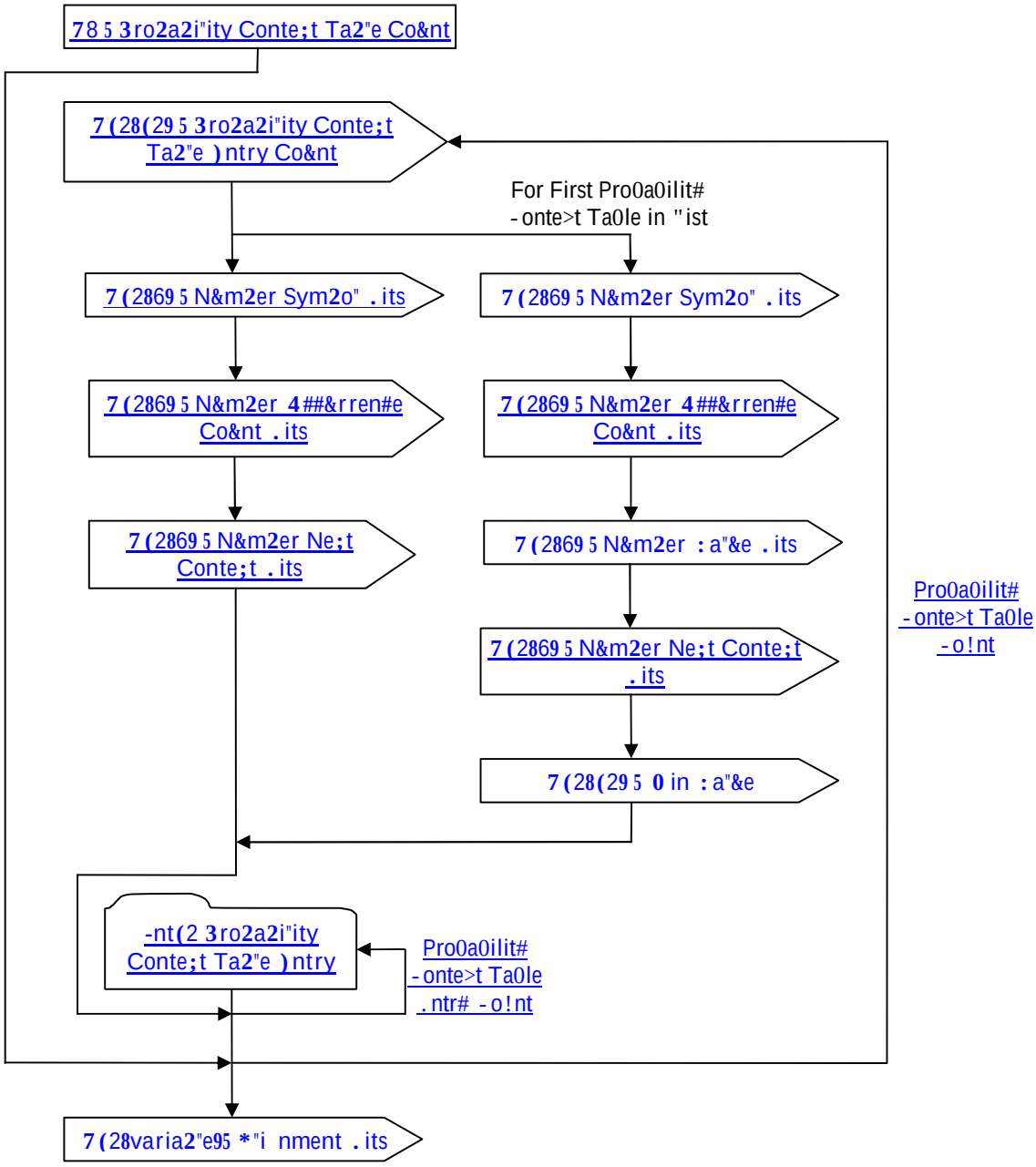
Vec1)# 2 Co*eTe:t

-o eTe>t is t(e arra#5vector of enco e s#m0ols. For - , D . - T#/e not eA!al to I+!ll - , D . -K) t(e total n!m0er of 0its of enco e ata in t(is arra# is in icate 0# t(e /revio!sl# escri0e -o eTe>t "engt(ata fiel .

8.1.1.1 Int)# Probabilit! Conte:ts

&ntB2 Pro0a0ilit# -onte>ts ata collection is a list of Pro0a0ilit# -onte>t Ta0les. T(e &ntB2 Pro0a0ilit# -onte>ts ata collection is onl# /resent for ; !ffman an 'rit(metic - , D . - T#/es. ' Pro0a0ilit# -onte>t Ta0le is a trimme an scale (istogram of t(e in/!t val!es. &t tallies t(e freA!encies of t(e several most freA!entl# occ!rring val!es. &t is central to t(e o/eration of t(e arit(metic - , D . -) an gives all t(e information necessar# to reconstr!ct t(e ; !ffman co es for t(e ; !ffman - , D . -.

!i &re 1%05 -nt(2 3ro2a2i"ity Conte;ts data #o""e#tion



18 2 Probabilit! Conte:t Table Count

Pro0a0ilit# -onte>t Ta0le -o!nt s/ecifies t(e n!m0er of Pro0a0ilit# -onte>t Ta0les to follo% an %ill al%a#s (ave a val!e of eit(er I1K or I2K.

1)#B)#D2 Probabilit! Conte:t Table 6ntr! Count
Pro0a0ilit# -onte>t Ta0le .ntr# -o!nt s/ecifies t(e n!m0er of entries in t(is Pro0a0ilit# -onte>t Ta0le.

1)#B.D2 'umber S!mbol 3its
+!m0er S#m0ol : its s/ecifies t(e n!m0er of 0its !se to enco e t(e S#m0ol range.

1)#B.D2 'umber 4ccurrence Count 3its
+!m0er ,cc!rrence -o!nt : its s/ecifies t(e n!m0er of 0its !se to enco e t(e ,cc!rrence -o!nt range.

1)#B.D2 'umber Value 3its
+!m0er Val!e : its s/ecifies t(e n!m0er of 0its !se to enco e t(e 'ssociate Val!e range. +ote t(at +!m0er Val!e : its is onl# s/ecifie in t(e JT file for t(e *first* Pro0a0ilit# -onte>t Ta0le. &f a secon Pro0a0ilit# -onte>t Ta0le is /resent) t(e +!m0er Val!e : its from t(e first s(o!l 0e !se for t(e secon as %ell.

1)#B.D2 'umber 'e:t Conte:t 3its
+!m0er +e>t -onte>t Fiel : its s/ecifies t(e n!m0er of 0its !se for t(e +e>t -onte>t Fiel in 8.1.1.2 &ntB2 Pro0a0ilit# -onte>t Ta0le .ntr#.

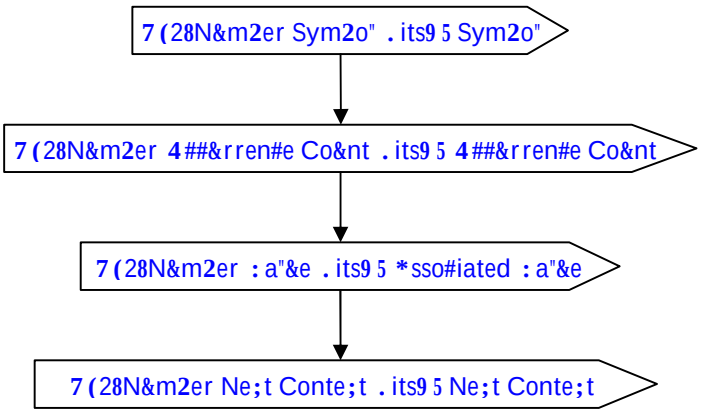
1)#B)#D2 9in Value
$in Val!e s/ecifies t(e minim!m of all 'ssociate Val!es 1i.e. one /er ta0le entr#2 store in t(is Pro0a0ilit# -onte>t Ta0le. T(is val!e is !se to com/!te t(e real 'ssociate Val!e for a Pro0a0ilit# -onte>t Ta0le .ntr#. See 'ssociate Val!e escri/tion in 8.1.1.2 &ntB2 Pro0a0ilit# -onte>t Ta0le .ntr#.

1)#BvariableD2 +lignment 3its
'lignment : its re/resents t(e n!m0er of a itional /a ing 0its store to arrive at t(e ne>t even m!lti/le of 8 0its. Val!es of IOK are store in t(e alignment 0its.

+ote* Data %ritten into t(e JtFile is al%a#s aligne on 0#tes. T(erefore after rea ing in a 0loc3 of 0it ata s!c(as t(e /ro0a0ilit# conte>t ta0les it is necessar# to iscar an# remaining 0its on t(e last 0#te t(at is rea in. T(is is re/resente 0# t(e I 'lignment : itsK entr#.

8.1.1.# Int)# Probabilit! Conte:t Table 6ntr!

!i &re 1%15 -nt(2 3ro2a2i"ity Conte;t Ta2"e)ntry data #o""e#tion



1)#B'umber S!mbol 3itsD 2 S!mbol

S#m0ol is a small integer n!m0er associate %it(a s/ecific val!e in t(e conte>t ta0le. &t serves onl# to im/ose an or er on t(e entries in t(e Pro0a0ilit# -onte>t Ta0le. T(e s#m0ol is store %it(a I\2K a e to t(e val!e an t(!s a rea er m!st s!0tract I2K from t(e rea val!e to get t(e tr!e s#m0ol val!e. -om/lete escri/tion for +!m0er S#m0ol :its can 0e fo!n in 8.1.1.1 &ntB2 Pro0a0ilit# -onte>ts.

+ote* .ven t(o!g(t(e s#m0ol is %ritten as a 4B2W+!m0er S#m0ol :itsX it is /ossi0le to en !/ %it(a negative n!m0er after s!0tracting I2K from t(e rea in val!e. ,ne e>am/le t(at %ill occ!r freA!entl# is t(e esca/e s#m0ol !se for o!t-of-0an ata %(ic(%ill (ave t(e val!e IOK in t(e file) (o%ever it %ill 0ecome I-2K) its tr!e s#m0ol val!e) after s!0tracting I2K from t(e rea in IOK val!e.

1)#B'umber 4ccurrence Count 3itsD 2 4ccurrence Count

,cc!rrence -o!nt s/ecifies t(e relative freA!enc# of t(e val!e. -om/lete escri/tion for +!m0er ,cc!rrence -o!nt :its can 0e fo!n in 8.1.1.1 &ntB2 Pro0a0ilit# -onte>ts.

+ote* ,cc!rrence -o!nts for all s#m0ols are normaliHe 1converte to a relative freA!enc#2 !ring t(e %rite /rocess in or er to ens!re t(e minim!m amo!nt of 0its /ossi0le is !se to %rite t(em. T(is (as several im/lications t(e rea er s(o!l 0e a%are of*

- T(e s!m of all ,cc!rrence -o!nts is not g!arantee to eA!al t(e n!m0er of s#m0ols to 0e eco e 1see Val!e .lement -o!nt in section 8.1.1 for n!m0er of s#m0ols to 0e eco e 2.
- D!ring 'rit(metic eco ing as escri0e in -.D.2.
    - *pDriver->numSymbolsToRead()* Q Refers to t(e total n!m0er of s#m0ols to 0e eco e 1i.e. Val!e .lement -o!nt in section 8.1.1 %(en t(e n!m0er of Pro0a0ilit# -onte>t Ta0les is eA!al to 1) or S#m0ol -o!nt %(en t(e n!m0er of Pro0a0ilit# -onte>t Ta0les is 22.

- *pCurrContext->totalCount()* Q Refers to t(e s!m of t(e I , cc!rrence -o!ntK val!es for all t(e s#m0ols associate %it(a Pro0a0ilit# -onte>t.

1)#B'umber Value 3itsD2 +ssociate* Value

'ssociate Val!e is t(e val!e 1from t(e in/!t ata2 t(at t(e s#m0ol re/resents. T(e - , D . -s onNt irectl# enco e val!es) t(e# enco e s#m0ols. S#m0ols) t(en) are associate %it(s/ecific val!es) so %(en t(e - , D . - eco es an arra# of s#m0ols) #o! can reconstr!ct t(e arra# of val!es t(at %as inten e 0# loo3ing !/ t(e s#m0ols in t(e Pro0a0ilit# -onte>t Ta0le. T(is val!e is store %it(I $ in Val!eK s!0tracte from t(e val!e. -om/lete escri/tions for I $ in Val!eK an +!m0er Val!e : its can 0e fo!n in 8.1.1.1 &ntB2 Pro0a0ilit# -onte>ts.

+ote* T(e associate val!e for an esca/e s#m0ol is !n efine an t(erefore can 0e an# vali 4B2 n!m0er.

1)#B'umber 'e:t Conte:t 3itsD2 'e:t Conte:t

+e>t -onte>t fiel s/ecifies %(ic(Pro0a0ilit# -onte>t Ta0le to !se %(en eco ing t(e ne>t s#m0ol. T(e val!e of t(is fiel %ill 0e greater t(an or eA!al to 0) an less t(an Pro0a0ilit# -onte>t Ta0le -o!nt.

8.1.# Float., Com resse* Data Pac?et

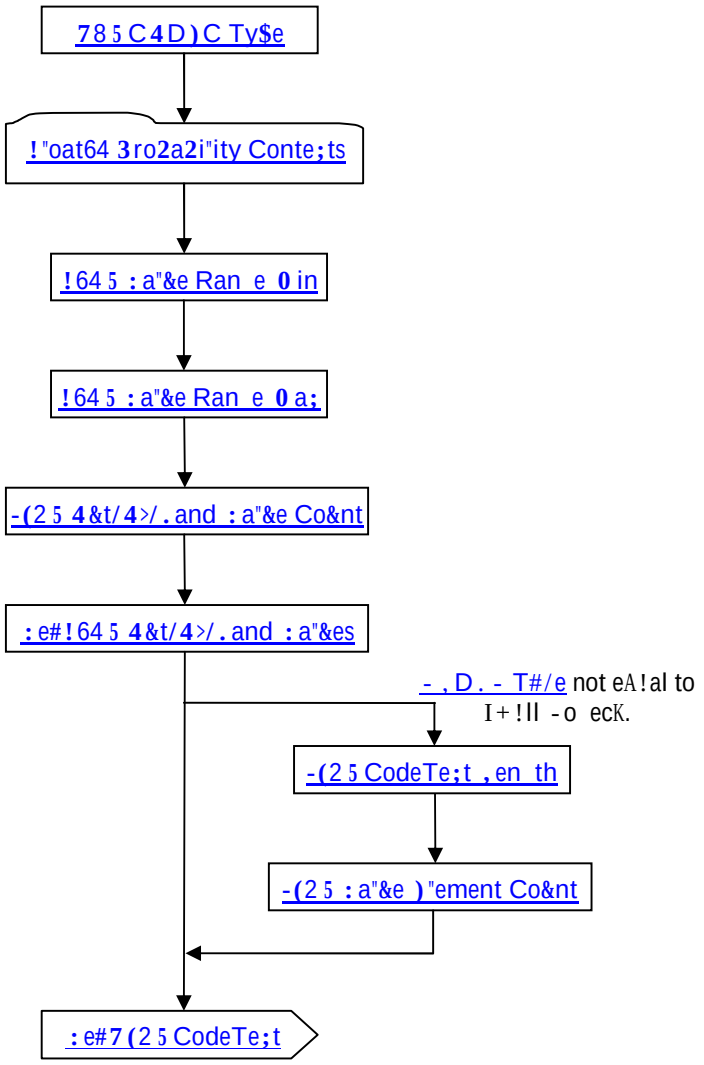
T(e FloatFD -om/resse Data Pac3et collection re/resents t(e format !se to enco e5com/ress a collection of ata into a series of FloatFD 0ase s#m0ols. T(is com/ression format also !ses t(e conce/t of Io!t-of-0an ataK in its ata contents efinition. &n t(e conte>t of t(e JT format ata com/ression algorit(ms an FloatFD -om/resse Data Pac3et) Io!t-of-0an ataK (as t(e follo%ing meaning.

- , D . -s li3e arit(metic an ; !ffman 1see 8.2 .nco ing 'lgorit(ms for tec(nical escri/tion2 e>/loit t(e statistics /resent in t(e relative freA!encies of t(e val!es 0eing enco e . Val!es t(at occ!r freA!entl# eno!g(allo% 0ot(of t(ese met(o s to enco e eac(of t(e val!es as a Is#m0olK in fe%er 0its t(at it %o!l ta3e to enco e t(e val!e itself. Val!es t(at occ!r too infreA!entl# to ta3e a vantage of t(is /ro/ert# are %ritten *aside* into t(e Io!t-of-0an ataK arra#. 'n Iesca/eK s#m0ol 1i.e. val!e of I-2K2 is enco e in t(eir /lace as a /lace(ol er in t(e /rimal - , D . -. .ssentiall# t(e Io!t-of-0an ataK is t(e (ig(-entro/# 7!n35resi !e5slag left over after t(e - , D . -s (ave sA!eeHe all t(e a vantage o!t t(at t(e# can.

= (ereas t(e &ntB2 -om/resse Data Pac3et 1see 8.1.1 &ntB2 -om/resse Data Pac3et2 t(en sen s t(is Io!t-of-0an ataK 0ac3 aro!n t(ro!g(a ne% - , D . - loo3ing for *different* statistics to 0e ta3en a vantage of) t(e FloatFD -om/resse Data Pac3et sim/l# %rites o!t t(e Io!t-of-0an ataK arra# %it(no a itional enco ing attem/te .

&n some cases) all val!es ma# %ritten bo!t of 0an b %(en t(e -o ec cannot /erform *any* !sef!l com/ression. &n t(is case) t(e enco e &B2 * -o eTe>t "engt(fiel %ill 0e 0) an t(e &B2 * i n e0[())-3.48478( )]TJ

!i &re 1%25 !"oat64 Com$ressed Data 3a#Eet data #o""e#tion



182C4D6C T! e

-,D.- T#/e s/ecifies t(e algorit(m !se to enco e5 eco e t(e ata. See 8.2 .nco ing 'lgorit(ms for com/lete e>/lanation of eac(of t(e enco ing algorit(ms.

[0	– +!ll -,D.-
[1	– :itlengt(-,D.-
[2	– ;!ffman -,D.-
[B	– 'rit(metic -,D.-

F., 2 Value Range 9in

Val!e Range $ in s/ecifies t(e minim!m of t(e val!e range !se to enco e t(e val!es.

F., 2 Value Range 9a:

Val!e Range $a> s/ecifies t(e ma>im!m of t(e val!e range !se to enco e t(e val!es.

I)#2 4ut-4f-3an* Value Count

, !t-, f-:an Val!e -o!nt s/ecifies t(e n!m0er of val!es t(at are I, !t-, f-:an .K

VecF., 2 4ut-4f-3an* Values

, !t-, f-:an Val!es s/ecifies t(e vector5list of I, !t-, f-:an K val!es.

I)#2 Co*eTe:t "engt%

-o eTe>t "engt(s/ecifies t(e total n!m0er of 0its of -o eTe>t ata 1 escri0e 0elo%2. T(is ata fiel is onl# /resent if -, D.- T#/e is not eA!al to I+!ll -, D.-.K

I)#2 Value 6lement Count

Val!e .lement -o!nt s/ecifies t(e n!m0er of val!es t(at t(e -, D.- is e>/ecte to eco e 1i.e. itNs li3e t(e Ilengt(K fiel %ritten if #o!Nre 7!st %riting o!t a vector of integers2. T(is ata fiel is onl# /resent if -, D.- T#/e is not eA!al to I+!ll -, D.-.K 4/on com/letion of eco ing t(e -o eTe>t ata fiel 0elo%) t(e n!m0er of eco e s#m0ol val!es s(o!l 0e eA!al to Val!e .lement -o!nt.

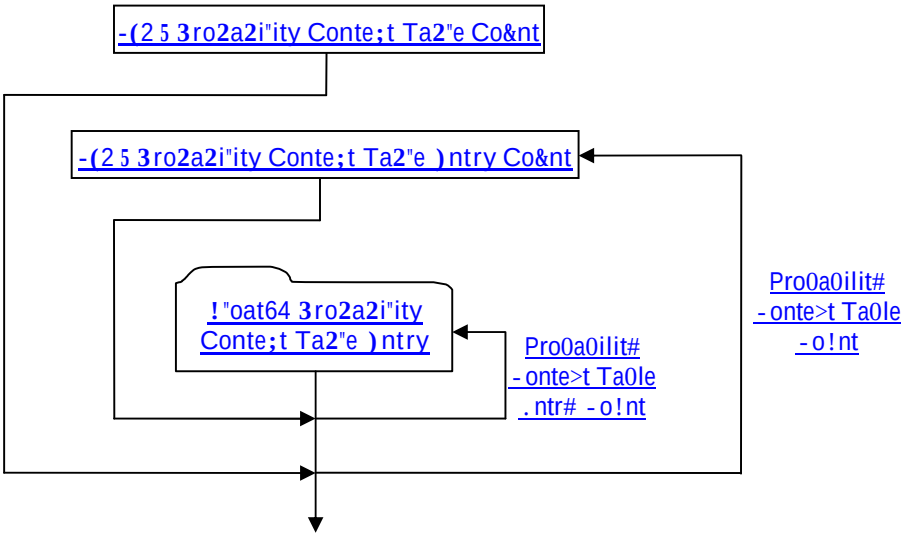
Vec1)#2 Co*eTe:t

-o eTe>t is t(e arra#5vector of enco e s#m0ols. For -, D.- T#/e not eA!al to I+!ll -, D.-K) t(e total n!m0er of 0its of enco e ata in t(is arra# is in icate 0# t(e /revio!sl# escri0e -o eTe>t "engt(ata fiel .

8.1.#.1 Float., Probabilit! Conte:ts

FloatFD Pro0a0ilit# -onte>ts ata collection is a list of Pro0a0ilit# -onte>t Ta0les. ' Pro0a0ilit# -onte>t Ta0le is a trimme an scale (istogram of t(e in/!t val!es. &t tallies t(e freA!encies of t(e several most freA!entl# occ!rring val!es. &t is central to t(e o/eration of t(e arit(metic -, D.-) an gives all t(e information necessar# to reconstr!ct t(e ;!ffman co es for t(e ;!ffman -, D.-.

!i &re 1%(5 !"oat64 3ro2a2i"ity Conte;ts data #o""e#tion



I)#2 Probabilit! Conte:t Table Count

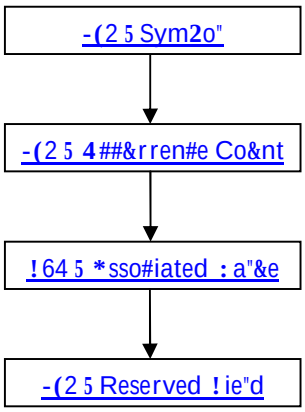
Pro0a0ilit# -onte>t Ta0le -o!nt s/ecifies t(e n!m0er of Pro0a0ilit# -onte>t Ta0les to follo% an %ill al%a#s (ave a val!e of eit(er I1K or I2K.

I)#2 Probabilit! Conte:t Table 6ntr! Count

Pro0a0ilit# -onte>t Ta0le .ntr# -o!nt s/ecifies t(e n!m0er of entries in t(is Pro0a0ilit# -onte>t Ta0le.

8.1.#.1.1 Float., Probabilit! Conte:t Table 6ntr!

!i &re 1%45 !"oat64 3ro2a2i"ity Conte;t Ta2"e)ntry data #o""e#tion



I)#2 S!mbol

S#m0ol is a small integer n!m0er associate %it(a s/ecific val!e in t(e conte>t ta0le. &t serves onl# to im/ose an or er on t(e entries in t(e Pro0a0ilit# -onte>t Ta0le. +ote t(at a val!e of I-2K re/resents t(e Iesca/eK s#m0ol /lace(ol er enco e for Io!t-of-0an ataK 1see 8.1.2 FloatFD -om/resse Data Pac3et for a itional etails2.

I)#2 4ccurrence Count

,cc!rrence -o!nt s/ecifies t(e relative freA!enc# of t(e val!e.

F.,2 +ssociate* Value

'ssociate Val!e is t(e val!e 1from t(e in/!t ata2 t(at t(e s#m0ol re/resents. T(e -,D.-s onNt irectl# enco e *values,* t(e# enco e *symbols*. S#m0ols) t(en) are associate %it(s/ecific val!es) so %(en t(e -,D.- eco es an arra# of s#m0ols) #o! can reconstr!ct t(e arra# of val!es t(at %as inten e 0# loo3ing !/ t(e s#m0ols in t(e Pro0a0ilit# -onte>t Ta0le.

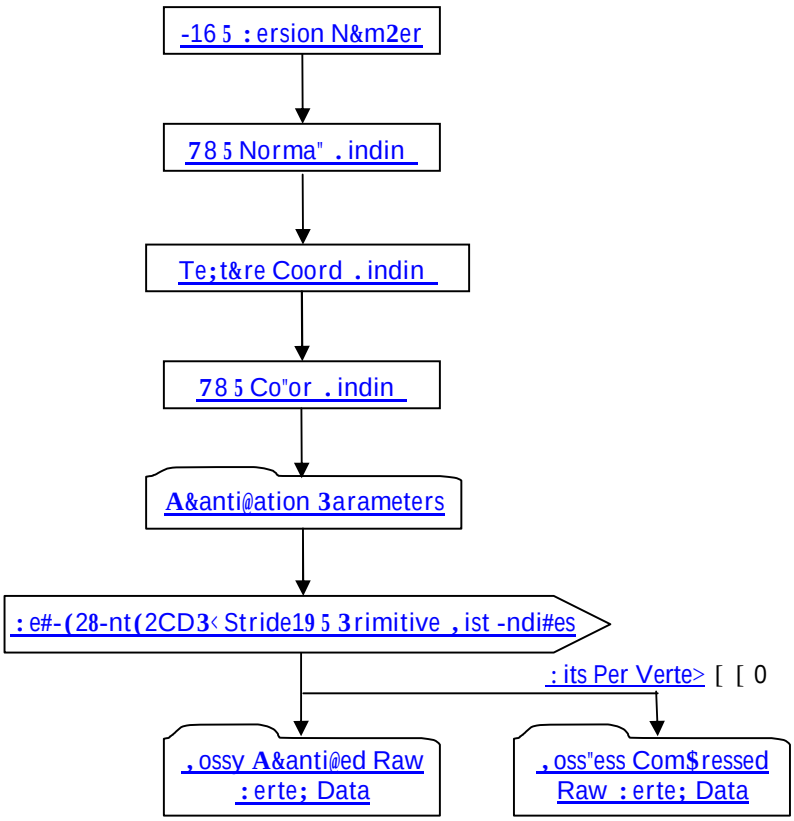
I)#2 Reserve* Fiel*

Reserve Fiel is a ata fiel reserve for f!t!re JT format e>/ansion.

8.1.) Verte: 3ase* S%a e Com resse* Re Data

T(e Verte> :ase S(a/e -om/resse Re/ Data collection is t(e com/resse an 5or enco e re/resentation of t(e verte> coor inates) normal) te>t!re coor inate) an color ata for a verte> 0ase s(a/e. 'll verte> 0ase s(a/e elements 1e.g. Tri-Stri/ Set S(a/e",D.lement) Pol#line Set S(a/e",D.lement2 !se t(is ata collection format to com/ress5enco e t(eir geometric ata.

!i &re 1%55 :erte; .ased Sha$e Com$ressed Re$ Data data #o""e#tion



-om/lete escri/tion for G!antiHation Parameters can 0e fo!n in C.2.1.1.1.10.2.1.1G!antiHation Parameters.

I1.2 Version 'umber

Version +!m0er is t(e version i entifier for t(is Verte> :ase S(a/e Re/ Data. Version n!m0er IO>0001K is c!rrentl# t(e onl# vali val!e.

182 'ormal 3in*ing

+ormal :in ing s/ecifies (o% 1at %(at gran!larit#2 normal vector ata is s!//lie 1IOo!n K2 for t(e S(a/e Re/ in eit(er t(e "ossless -om/resse Ra% Verte> Data or "oss# G!antiHe Ra% Verte> Data collections.

[0	– +one. +o normal ata.
[1	– Per Verte>. +ormal vector for ever# verte>.
[2	– Per Facet. +ormal vector for ever# face5/ol#gon.
[B	– Per Primitive. S(a/e (as a normal vector for eac(s(a/e /rimitive 1e.g. a C.2.1.1.1.10.B Tri-Stri/ Set S(a/e +o e .lement is ma e !/ of a collection of in e/en ent an !nconnecte triangle stri/s? %(ere eac(stri/ constit!tes one /rimitive of t(e s(a/e an t(!s t(ere %o!l 0e a normal /er triangle stri/2.

8.1.).1 Te:ture Coor* 3in*ing

Te>t!re -oor :in ing s/ecifies (o% 1at %(at gran!larit#2 te>t!re coor inate ata is s!//lie 1IOo!n K2 for t(e S(a/e Re/ in eit(er t(e "ossless -om/resse Ra% Verte> Data
or "oss# G!antiHe Ra% Verte> Data collections. Vali val!es are t(e same as oc!mente for +ormal :in ing ata fiel .

18 2 Color 3in*ing

-olor :in ing s/ecifies (o% 1at %(at gran!larit#2 color ata is s!//lie 1IOo!n K2 for t(e S(a/e Re/ in eit(er t(e "ossless -om/resse Ra% Verte> Data
or "oss# G!antiHe Ra% Verte> Data collections. Vali val!es are t(e same as oc!mente for +ormal :in ing ata fiel .

VecI)#BInt)#CDP¢ Stri*e1D 2 Primitive "ist In*ices

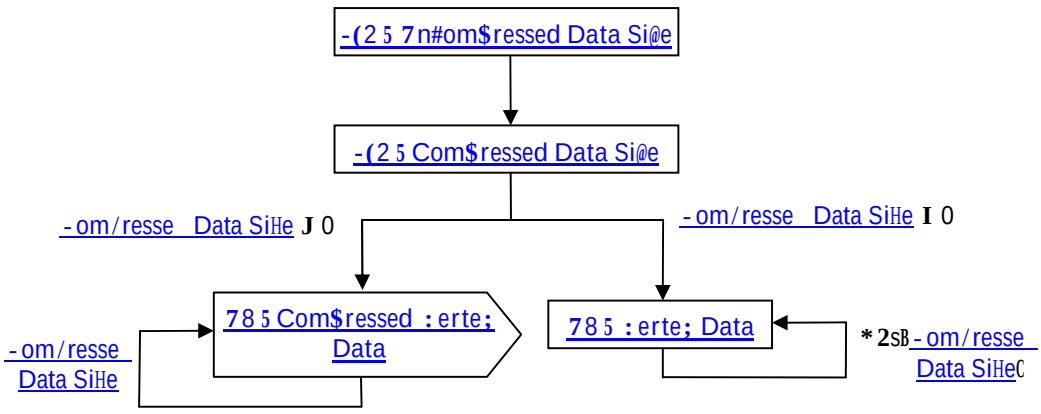
Primitive "ist &n ices is a vector of in ices into t(e !ncom/resse Ra% Verte> Data mar3ing t(e start5Oeginning of /rimitives. Primitive "ist &n ices !ses t(e &ntB2 version of t(e -,D.- to com/ress an enco e ata.

8.1.).# "ossless Com resse* Ra(Verte: Data

T(e "ossless -om/resse Ra% Verte> Data
collection contains all t(e /er-verte> information 1i.e. 4V te>t!re coor inates) color) normal vector) 8 9L coor inate2 store in a IlosslessK com/ression format for all /rimitives of t(e s(a/e. T(e "ossless -om/resse Ra% Verte> Data
collection is onl# /resent %(en t(e G!antiHation Parameters :its Per Verte> ata fiel eA!als IOK 1See C.2.1.1.1.10.2.1.1 G!antiHation Parameters for com/lete escri/tion2.

!i &re 1%65 ,oss"ess Com$ressed Raw :erte; Data

data #o""e#tion



I)# 2 1ncom resse* Data Si=e

4ncom/resse Data siHe s/ecifies t(e !ncom/resse siHe of Verte> Data or -om/resse Verte> Data in 0#tes.

I)#2 Com resse* Data Si=e

-om/resse Data SiHe s/ecifies t(e com/resse siHe of Verte> Data or -om/resse Verte> Data in 0#tes. &f t(e -om/resse Data SiHe is negative) t(en t(e -om/resse Verte> Data fiel is not /resent 1i.e. ata is not com/resse 2 an t(e a0sol!te val!e of -om/resse Data SiHe s(o!l 0e eA!al to 4ncom/resse Data SiHe val!e.

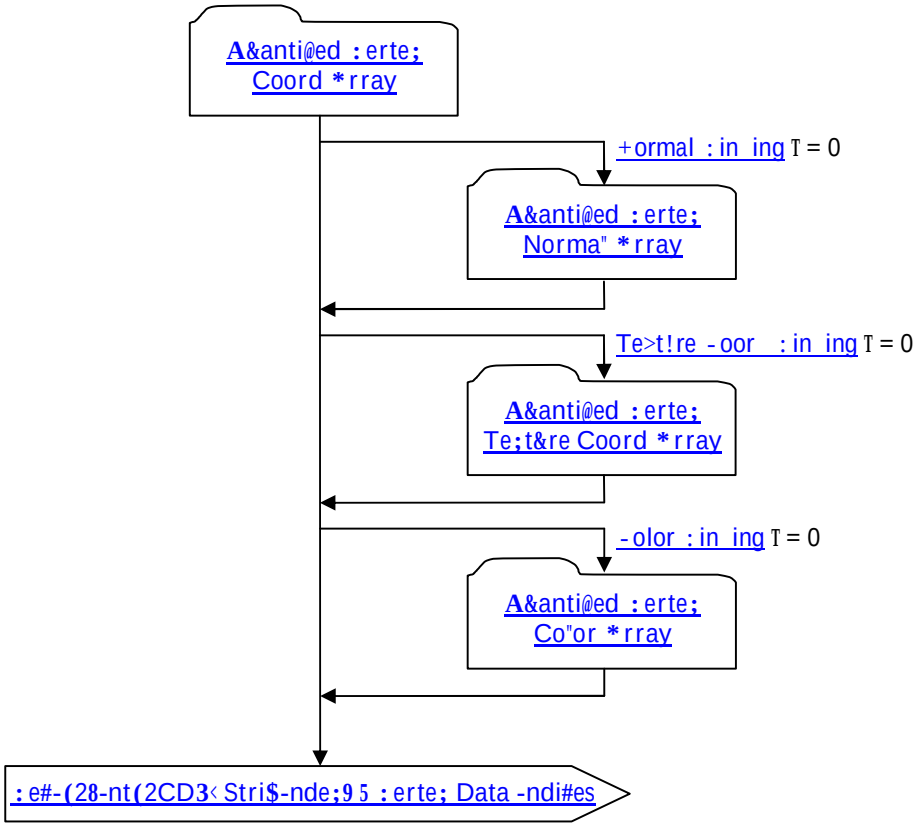
182 Verte: Data

T(e Verte> Data fiel is a /ac3e arra# of t(e ra% /er verte> ata 1i.e. 4V te>t!re coor inates) color) normal vector) 8 9L coor inate2. T(e Verte> Data fiel is onl# /resent if -om/resse Data SiHe val!e is less t(an Hero.

T(e e>istence of te>t!re coor inate) color) an normal vector ata %it(in Verte> Data arra# is

Data &n ices !se to e>/an t(is !niA!e list into t(e Ra% Verte> Data list conformal %it(t(e Primitive "ist &n ices of Fig!re 1CE.

!i &re 1%%5 ,ossy A&anti@ed Raw :erte; Data data #o""e#tion



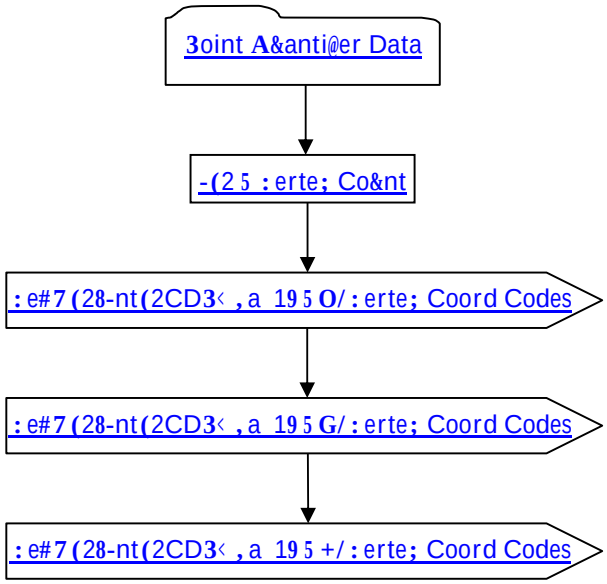
Vecl)#BInt)#CDP(Stri In*e:D 2 Verte: Data In*ices

Verte> Data &n ices is a vector of in ices 1one /er verte>2 into t(e !ncom/resse 5 eA!antiHe !niA!e verte> ata arra#s 1Verte> -oor s) Verte> +ormals) Verte> Te>t!re -oor s) Verte> -olors2 i entif#ing eac(Verte>Ns ata 1i.e. for eac(Verte> t(ere is an in e> i entif#ing t(e location %it(in t(e !niA!e arra#s of t(e /artic!lar Verte>Ns ata2. T(e -om/resse Verte> &n e> "ist !ses t(e &ntB2 version of t(e - , D. - to com/ress an enco e ata.

8.1.).).1 >uanti=e* Verte: Coor* +rra!

T(e G!antiHe Verte> -oor 'rra# ata collection contains t(e A!antiHation ata5re/resentation for a set of verte> coor inates.

!i &re 1%85 A&anti@ed :erte; Coord *rray data #o""e#tion



-om/lete escri/tion for Point G!antiHer Data can 0e fo!n in 8.1.D Point G!antiHer Data.

I)# 2 Verte: Count

Verte> -o!nt s/ecifies t(e co!nt 1n!m0er of !niA!e2 vertices in t(e Verte> -o es arra#s.

Vec1)#BInt)#CDP("ag1D 2 F-Verte: Coor* Co*es

8 Verte> -oor -o es is a vector of A!antiHer Ico esK for all t(e 8-com/onents of a set of verte> coor inates. 8-Verte> -oor -o es !ses t(e &ntB2 version of t(e -,D.- to com/ress an enco e ata.

Vec1)#BInt)#CDP("ag1D 2 @-Verte: Coor* Co*es

9 Verte> -oor -o es is a vector of A!antiHer Ico esK for all t(e 9-com/onents of a set of verte> coor inates. 9-Verte> -oor -o es !ses t(e &ntB2 version of t(e -,D.- to com/ress an enco e ata.

Vec1)#BInt)#CDP("ag1D 2 8-Verte: Coor* Co*es

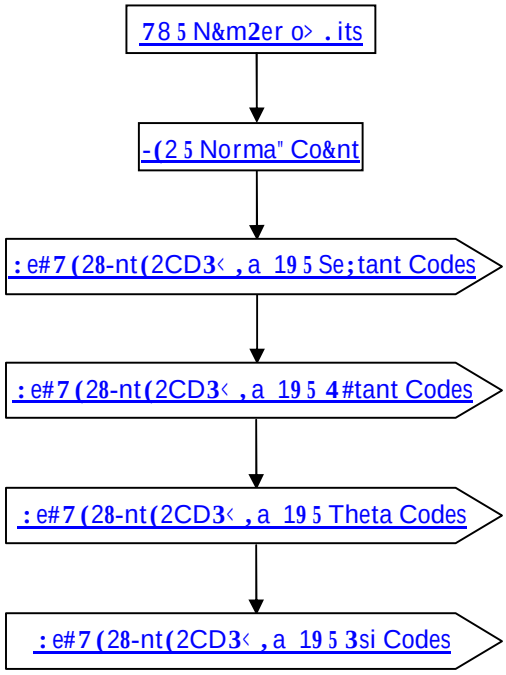
L Verte> -oor -o es is a vector of A!antiHer Ico esK for all t(e L-com/onents of a set of verte> coor inates. L-Verte> -oor -o es !ses t(e &ntB2 version of t(e -,D.- to com/ress an enco e ata.

8.1.).).# >uanti=e* Verte: 'ormal +rra!

T(e G!antiHe Verte> +ormal 'rra# ata collection contains t(e A!antiHation ata5re/resentation for a set of verte> normals. G!antiHe Verte> +ormal 'rra# ata collection is onl# /resent if /revio!sl# rea +ormal :in ing val!e is not eA!al to Hero 1See 8.1.BVerte> :ase S(a/e -om/resse Re/ Data for com/lete e>/lanation of +ormal :in ing ata fiel 2.

' variation of t(e -,D.- evelo/e 0# $ic(ael Deering at S!n $icros#stems is !se to enco e t(e normals. T(e variation 0eing t(at t(e ISe>tantsK are arrange ifferentl# t(an in DeeringNs sc(eme RFS) for 0etter elta enco ing. See 8.2.E Deering +ormal -,D.- for a com/lete e>/lanation on t(e Deering -,D.- !se .

!i &re 1%'5 A&anti@ed :erte; Norma" *rray data #o""e#tion



182 'umber of 3its

+!m0er of :its s/ecifies t(e A!antiHe siHe 1i.e. t(e n!m0er of 0its of /recision2 for t(e T(eta an PS& angles. T(is val!e m!st satisf# t(e follo%ing con ition* I0 Y[+!m0er of :its Y[1BK.

I)#2 'ormal Count

+ormal -o!nt s/ecifies t(e co!nt 1n!m0er of !niA!e2 +ormal -o es.

Vec1)#BInt)#CDP("ag1D2 Se:tant Co*es

Se>tant -o es is a vector of Ico esK 1one /er normal2 for a set of normals i entif#ing %(ic(Se>tant of t(e corres/on ing s/(ere ,ctant eac(normal is locate in. Se>tant -o es !ses t(e &ntB2 version of t(e -,D.- to com/ress an enco e ata.

Vec1)#BInt)#CDP("ag1D2 4ctant Co*es

,ctant -o es is a vector of Ico esK 1one /er normal2 for a set of normals i entif#ing %(ic(s/(ere ,ctant eac(normal is locate in. ,ctant -o es !ses t(e &ntB2 version of t(e -,D.- to com/ress an enco e ata.

Vec1)#BInt)#CDP("ag1D 2 T%eta Co*es

T(eta -o es is a vector of Ico esK 1one /er normal2 for a set of normals re/resenting in Se>tant coor inates t(e A!antiHe t(eta angle for eac(normalNs location on t(e !nit ra i!s s/(ere? %(ere t(eta angle is efine as t(e angle in s/(erical coor inates a0o!t t(e 9-a>is on a !nit ra i!s s/(ere. T(eta -o es !ses t(e &ntB2 version of t(e - , D . - to com/ress an enco e ata.

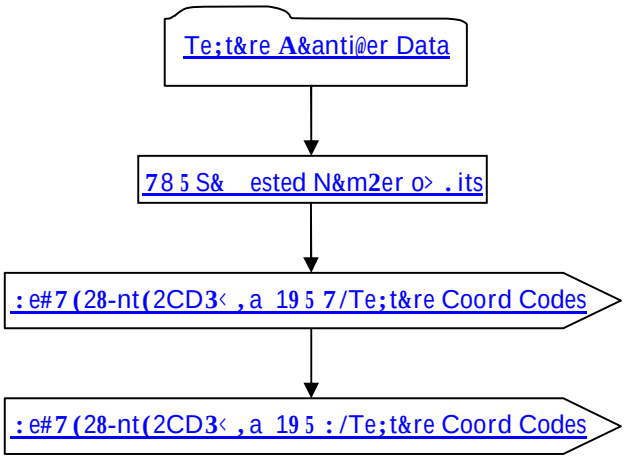
Vec1)#BInt)#CDP("ag1D 2 Psi Co*es

Psi -o es is a vector of Ico esK 1one /er normal2 for a set of normals re/resenting in Se>tant coor inates t(e A!antiHe Psi angle for eac(normalNs location on t(e !nit ra i!s s/(ere? %(ere Psi angle is efine as t(e longit! inal angle in s/(erical coor inates from t(e # [0 /lane on t(e !nit ra i!s s/(ere. Psi -o es !ses t(e &ntB2 version of t(e - , D . - to com/ress an enco e ata

8.1.).).) >uanti=e* Verte: Te:ture Coor* +rra!

T(e G!antiHe Verte> Te>t!re -oor 'rra# ata collection contains t(e A!antiHation ata5re/resentation for a set of verte> te>t!re coor inates. G!antiHe Verte> Te>t!re -oor 'rra# ata collection is onl# /resent if /revio!sl# rea Te>t!re -oor : in ing val!e is not eA!al to Hero 1See 8.1.BVerte> : ase S(a/e -om/resse Re/ Data for com/lete e>/lanation of Te>t!re -oor : in ing ata fiel 2.

!i &re 1805 A&anti@ed :erte; Te;t&re Coord *rray data #o""e#tion



-om/lete escri/tion for Te>t!re G!antiHer Data can 0e fo!n in 8.1.E Te>t!re G!antiHer Data.

182 Suggeste* 'umber of 3its

S!ggeste +!m0er of :its s/ecifies t(e s!ggeste n!m0er of A!antiHation 0its /er te>t!re coor inate 4 an V com/onents. &t is onl# a s!ggeste val!e 1an (as no real val!e for a JT file loa er5rea er2 0eca!se t(e act!al n!m0er of 0its !se ma# iffer 1increase or ecrease 2 e/en ing on t(e range of val!es for te>t!re coor inates. T(e act!al n!m0er of A!antiHation 0its !se is s/ecifie %it(in Te>t!re G!antiHer Data. Val!e m!st 0e %it(in range R0*2D5 incl!sive.

Vec1)#BInt)#CDPC "ag1D 2 1-Te:ture Coor* Co*es

4-Te>t!re -oor -o es is a vector of A!antiHer Ico esK for all t(e 4-com/onents of a set of verte> te>t!re coor inates. 4-Te>t!re -oor -o es !ses t(e &ntB2 version of t(e - , D. - to com/ress an enco e ata.

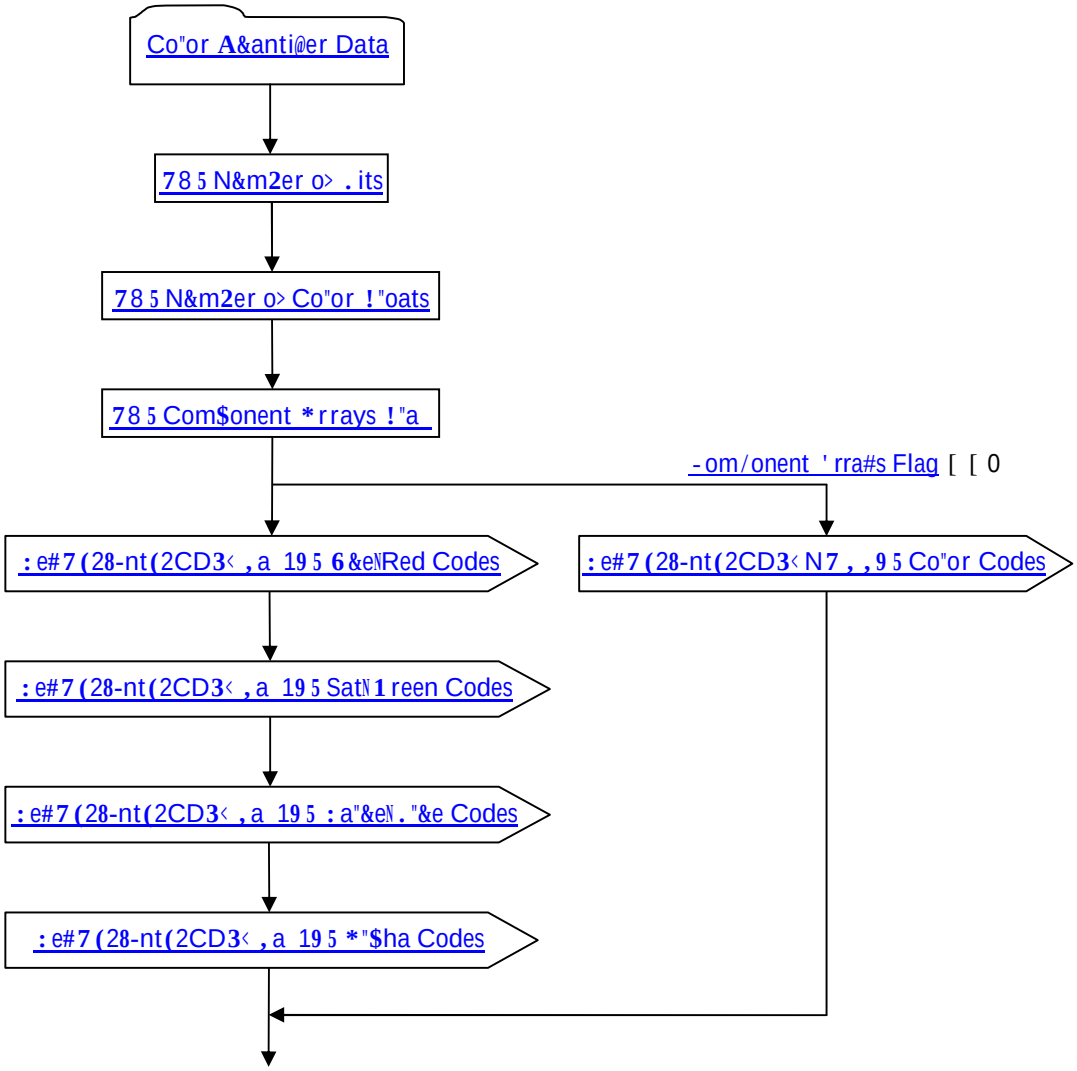
Vec1)#BInt)#CDPC "ag1D 2 V-Te:ture Coor* Co*es

V-Te>t!re -oor -o es is a vector of A!antiHer Ico esK for all t(e V-com/onents of a set of verte> te>t!re coor inates. V-Te>t!re -oor -o es !ses t(e &ntB2 version of t(e - , D. - to com/ress an enco e ata.

8.1.).)., >uanti=e* Verte: Color +rra!

T(e G!antiHe Verte> -olor 'rra# ata collection contains t(e A!antiHation ata5re/resentation for a set of verte> colors. G!antiHe Verte> -olor 'rra# ata collection is onl# /resent if /revio!sl# rea -olor :in ing val!e is not eA!al to Hero 1See 8.1.BVerte> :ase S(a/e -om/resse Re/ Data for com/lete e>/lanation of -olor :in ing ata fiel 2.

!i &re 1815 A&anti@ed :erte; Co"or *rray data #o""e#tion



-om/lete escri/tion for -olor G!antiHer Data can 0e fo!n in 8.1.F -olor G!antiHer Data.

182 'umber of 3its

+!m0er of :its s/ecifies t(e A!antiHe siHe 1i.e. t(e n!m0er of 0its of /recision2 for eac(of t(e B or D color com/onents. T(is val!e m!st satisf# t(e follo%ing con ition* I0 Y[+!m0er , f :its Y[8K.

182 'umber of Color Floats

+!m0er of -olor Floats s/ecifies t(e n!m0er of floating /oint val!es !se to re/resent t(e color com/onents. Vali val!es incl! e t(e follo%ing*

[1	– ' ll com/onents /ac3e into a single B2 0it val!e P 8 0its /er com/onent.
[B	– .ac(R6 :5 ; SV color com/onent re/resenting in its o%n floating /oint val!e. ' l/(a al%a#s ass!me to 0e 1.
[D	– .ac(R6 : '5 ; SV ' color com/onent re/resenting in its o%n floating /oint val!e.

18 2 Com onent +rra!s Flag

-om/onent ' rra#s Flag is a flag in icating %(et(er color com/onents are enco e as a single integer or if t(e enco ing is 0ro3en ! / 1se/arate 2 into an arra# for eac(color com/onent.

[0	– .nco e as single integer P t(!s one com/resse collection of co es
[1	– .nco ing is 0ro3en ! / 1se/arate 2 into an arra# of co es for eac(color com/onent P t(!s a com/resse collection of co es for eac(color com/onent.

Vec1)#BInt)#CDP("ag1D 2 0ueERe* Co*es

; !e5Re -o es is a vector of A!antiHer Ico esK for all t(e ; !e5Re color com/onents of a set of verte> colors. = (et(er ; SV or R6 : color mo el is 0eing !se 1i.e. ; !e or Re 2 is in icate 0# a flag store in t(e -olor G!antiHer Data. ; !e5Re -o es is onl# /resent %(en ata fiel -om/onent ' rra#s Flag [[1. ; !e5Re -o es !ses t(e &ntB2 version of t(e - , D . - to com/ress an enco e ata.

Vec1)#BInt)#CDP("ag1D 2 SatE5reen Co*es

Sat56reen -o es is a vector of A!antiHer Ico esK for all t(e Sat!ration56reen color com/onents of a set of verte> colors. = (et(er ; SV or R6 : color mo el is 0eing !se 1i.e. Sat!ration or 6reen2 is in icate 0# a flag store in t(e -olor G!antiHer Data. Sat56reen -o es is onl# /resent %(en ata fiel -om/onent ' rra#s Flag [[1. Sat56reen -o es !ses t(e &ntB2 version of t(e - , D . - to com/ress an enco e ata.

Vec1)#BInt)#CDP("ag1D 2 ValueE3lue Co*es

Val!e5 : l!e -o es is a vector of A!antiHer Ico esK for all t(e Val!e5 : l!e color com/onents of a set of verte> colors. = (et(er ; SV or R6 : color mo el is 0eing !se 1i.e. Val!e or : l!e2 is in icate 0# a flag store in t(e -olor G!antiHer Data. Val!e5 : l!e -o es is onl# /resent %(en ata fiel -om/onent ' rra#s Flag [[1. Val!e5 : l!e -o es !ses t(e &ntB2 version of t(e - , D . - to com/ress an enco e ata.

Vec1)#BInt)#CDP("ag1D 2 +l %a Co*es

' l/(a -o es is a vector of A!antiHer Ico esK for all t(e ' l/(a color com/onents of a set of verte> colors. ' l/(a -o es is onl# /resent %(en ata fiel -om/onent ' rra#s Flag [[1. ' l/(a -o es !ses t(e &ntB2 version of t(e - , D . - to com/ress an enco e ata.

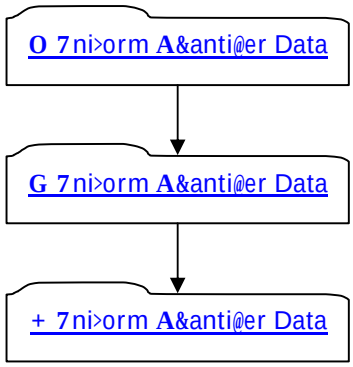
Vec1)#BInt)#CDP(' 1" "D 2 Color Co*es

-olor -o es is a vector of A!antiHer Ico esK for a set of verte> colors. -olor -o es is onl# /resent %(en ata fiel -om/onent ' rra#s Flag [[0. -olor -o es !ses t(e &ntB2 version of t(e - , D . - to com/ress an enco e ata.

8.1., Point >uanti=er Data

' Point G!antiHer Data collection is ma e !/ of t(ree 4niform G!antiHer Data collections? t(ere is a se/arate 4niform G!antiHer Data collection for t(e 8) 9) an L val!es of /oint coor inates.

!i &re 1825 3oint A&anti@er Data data #o""e#tion

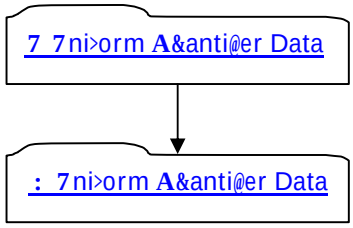


-om/lete escri/tion for 8 4niform G!antiHer Data) 9 4niform G!antiHer Data an L 4niform G!antiHer Data can 0e fo!n in 8.1.C 4niform G!antiHer Data.

8.1.- Te:ture >uanti=er Data

' Te>t!re G!antiHer Data collection is ma e !/ of t%o 4niform G!antiHer Data collections? t(ere is a se/arate 4niform G!antiHer Data collection for t(e 4) an V val!es of te>t!re coor inates.

!i &re 18(5 Te;t&re A&anti@er Data data #o""e#tion



-om/lete escri/tion for 4 4niform G!antiHer Data) an V 4niform G!antiHer Data can 0e fo!n in 8.1.C 4niform G!antiHer Data.

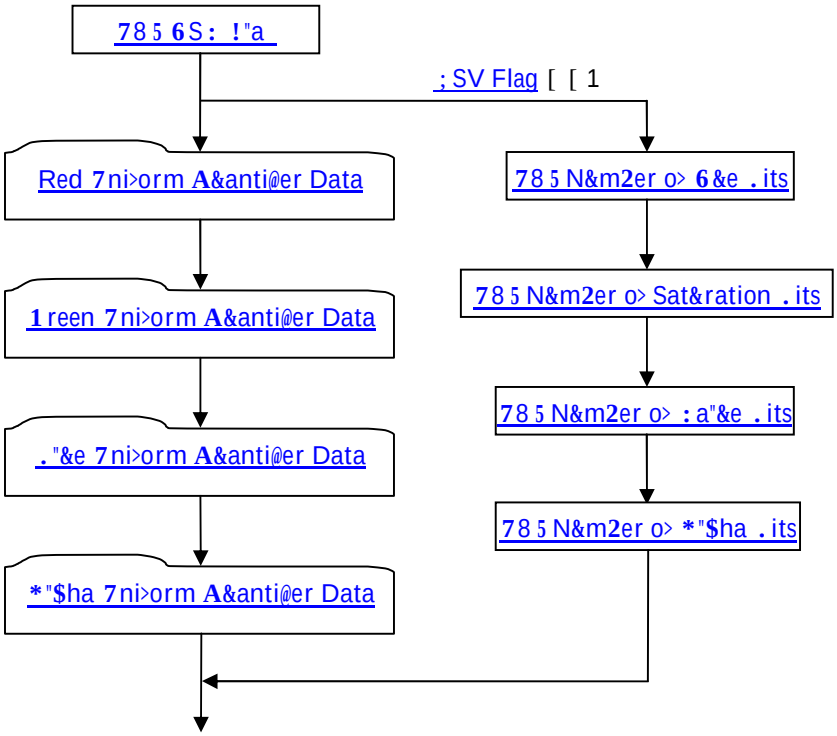
8.1.. Color >uanti=er Data

' -olor G!antiHer Data collection contains t(e A!antiHer information for eac(of t(e color com/onents. T(e -olor G!antiHer !tiliHes a se/arate 4niform G!antiHer Data collection for eac(of t(e D color com/onents) 0!t if t(e ; SV color mo el is 0eing !se) t(en it is not necessar# to store a com/lete 4niform G!antiHer Data -ollection.

For t(e ; SV mo el) since t(e range val!es for eac(color com/onent are constant) onl# t(e +!m0er of : its of /recision for eac(color com/onentNs 4niform G!antiHer is store . T(e 4niform G!antiHer range val!es for t(e ; SV color com/onents s(o!l al%a#s 0e ass!me to 0e t(e follo%ing*

Com$onent	A&anti@er Ran e	
	0 in	**0** a;
; !e	0.0	F.0
Sat!ration	0.0	1.0
Val!e	0.0	1.0
' l/(a	0.0	1.0

!i &re 1845 Co"or A&anti@er Data data #o""e#tion



-om/lete escri/tions for Re 4niform G!antiHer Data) 6reen 4niform G!antiHer Data) :l!e 4niform G!antiHer Data) an ' l/(a 4niform G!antiHer Data can 0e fo!n in 8.1.C 4niform G!antiHer Data. T(ese fo!r 4niform G!antiHer Data collections are onl# /resent %(en ata fiel ; SV Flag [[0.

18 2 0SV Flag

; SV Flag is a flag in icating %(et(er color com/onent ata is store in ; SV color mo el form.

[0	– -olor com/onent ata store in R6 : color mo el form.
[1	– -olor com/onent ata store in ; SV color mo el form.

18 2 'umber of **0**ue **3**its

+!m0er of ;!e :its s/ecifies t(e A!antiHe siHe 1i.e. t(e n!m0er of 0its of /recision2 for t(e ;!e com/onent of t(e color. +!m0er of ;!e :its ata is onl# /resent %(en ata fiel ; SV Flag [[1.

18 2 'umber of Saturation **3**its

+!m0er of Sat!ration :its s/ecifies t(e A!antiHe siHe 1i.e. t(e n!m0er of 0its of /recision2 for t(e Sat!ration com/onent of t(e color. +!m0er of Sat!ration :its ata is onl# /resent %(en ata fiel ; SV Flag [[1.

18 2 'umber of Value **3**its

+!m0er of Val!e :its s/ecifies t(e A!antiHe siHe 1i.e. t(e n!m0er of 0its of /recision2 for t(e Val!e com/onent of t(e color. +!m0er of Val!e :its ata is onl# /resent %(en ata fiel ; SV Flag [[1.

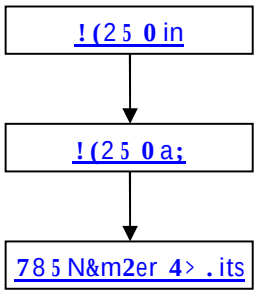
18 2 'umber of +l %a **3**its

+!m0er of 'l/(a :its s/ecifies t(e A!antiHe siHe 1i.e. t(e n!m0er of 0its of /recision2 for t(e 'l/(a com/onent of t(e color. +!m0er of 'l/(a :its ata is onl# /resent %(en ata fiel ; SV Flag [[1.

8.1.*1* **1**niform **>**uanti=er Data

T(e 4niform G!antiHer Data collection contains information t(at efines a scalar A!antiHer5 eA!antiHer 1enco er5 eco er2 %(ose range is ivi e into levels of eA!al s/acing.

!i &re 1855 7ni>orm **A**&anti@er Data data #o""e#tion



F)#2 **9**in

$in s/ecifies t(e minim!m of t(e A!antiHe range.

F)#2 **9**a:

$a> s/ecifies t(e ma>im!m of t(e A!antiHe range.

18 2 'umber **4**f **3**its

+!m0er of :its s/ecifies t(e A!antiHe siHe 1i.e. t(e n!m0er of 0its of /recision2. &n general) t(is val!e m!st satisf# t(e follo%ing con ition* I0 Y[+!m0er , f :its Y[B2K.

8.1.8 Com resse* 6ntit! "ist for 'on-Trivial Anot Vector

-om/resse .ntit# "ist for +on-Trivial <not Vector ata collection s/ecifies in e> i entifiers 1i.e. in ices to /artic!lar entities %it(in a list of entities2 for a set of entities t(at contain +on-Trivial <not Vectors. T(e entit# t#/es %(ic(can contain non-trivial 3not vectors incl! e*

- JT :-Re/ +4R:S S!rfaces
- JT :-Re/ P-S+4R:S -!rves
- JT :-Re/ $-S+4R:S -!rves
- =ireframe $-S+4R:S -!rves

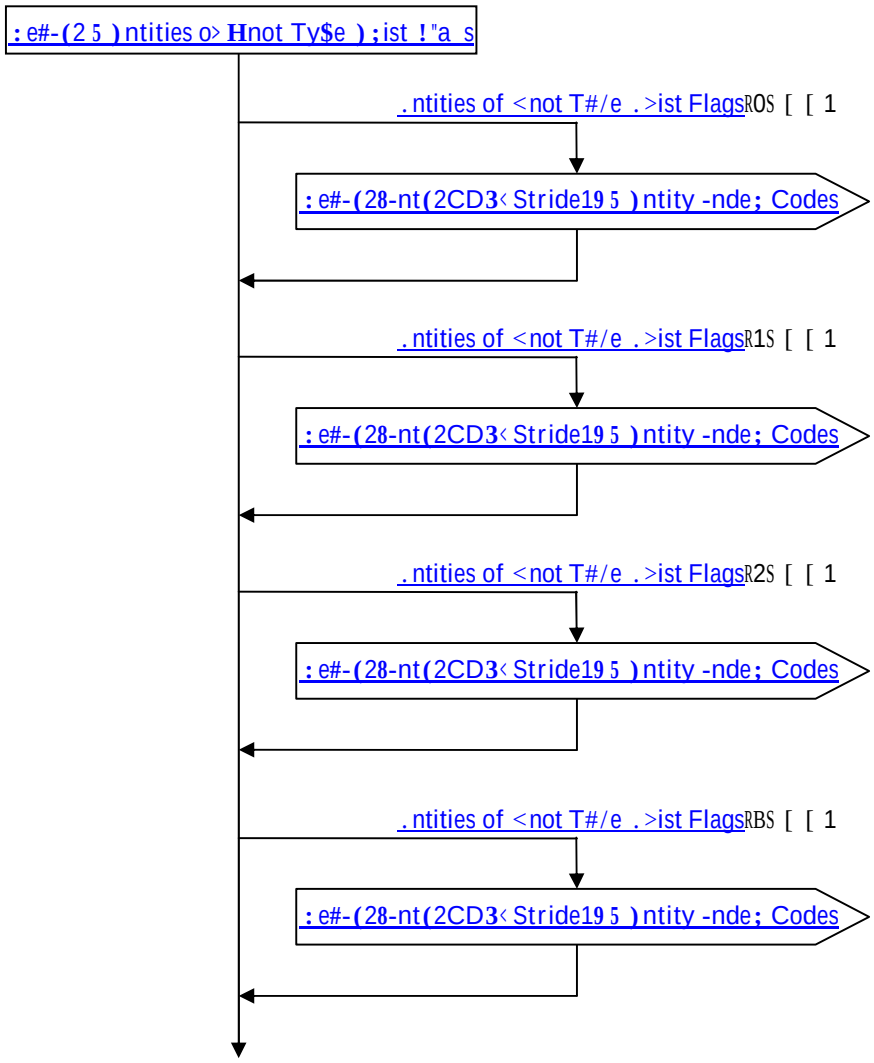
+ote t(at an# one occ!rrence of -om/resse .ntit# "ist for +on-Trivial <not Vector ata collection %ill onl# contain in e> i entifiers for one /artic!lar t#/e of t(e a0ove liste entities. T(e entit# t#/e is inferre 0ase on t(e ata collection %(ic(incl! es5references t(e -om/resse .ntit# "ist for +on-Trivial <not Vector.

' trivial 3not vector is one %(ic(com/letel# satisfies all con itions of at least one of t(e follo%ing cases*

1. -ase-1 for trivial 3not vector
   a. +!m0er of 3nots is an even n!m0er
   0. <not vector (as a R0*1S 3not range
   c. T(ere are no interior 3nots 1i.e. +!m0er<nots [[2] 1+!r0s.ntit#Degree \ 12
2. -ase-2 for trivial 3not vector
   a. +!m0er of 3nots is an even n!m0er.
   0. <not vector (as a R0*1S 3not range
   c. +!r0s.ntit#Degree Y B
   . Difference 0et%een s!ccessive non-re/eating 3nots 1i.e. <notDelta2 is*
      i. <notDelta [2.0 5 1+!m0er<nots Q 12.0] +!r0s.ntit#Degree22

'n# 3not vector %(ic(oes not satisf# one of t(e a0ove cases for Itrivial 3not vectorK is classifie as a Inon-trivial 3not vector.K

!i &re 1865 Com$ressed)ntity ,ist ›or Non/Trivia" Hnot :e#tor data #o""e#tion



Vecl)#2 6ntities of Anot T! e 6:ist Flags

.ntities of <not T#/e .›ist Flags) is a vector of flags in icating for eac(3not vector t#/e %(et(er .ntit# &n e› &D ata collections e›ist5follo% for t(at 3not vector t#/e. <not Vectors are categoriHe into t#/es 0ase on t(e follo%ing c(aracteristics* %(et(er internal 3nots occ!r in *adjacent pairs* an %(et(er t(e 3not range is R0*1S or some ot(er R›$1*›$2S range.

-!rrentl# t(ere are fo!r 3not vector t#/es) so t(is .ntities of <not T#/e .›ist Flags vector s(o!l 0e of lengt(fo!r. T(e fo!r flags (ave t(e follo%ing meaning*

ROS	Flag in icating %(et(er . ntit# &Ds ata collection e>ists for I . ven - o!nt RO*1S RangeK 3not t#/e. <nots in t(is categor# (ave t(eir 3not range on RO*1S) internal 3nots occ!r in *adjacent pairs*) *except* %(en t(ere are no internal 3nots) in %(ic(case T#/e [1 instea . [0 Q +o . ntit# &Ds ata collection e>ists. [1 Q . ntit# &Ds ata collection e>ists.
R1S	– Flag in icating %(et(er . ntit# &Ds ata collection e>ists for I . ven - o!nt R>$_1$*>$_2$S RangeK 3not t#/e. <nots in t(is categor# (ave t(eir 3not range on R>$_1$*>$_2$S) an internal 3nots occ!r in *adjacent pairs*. [0 Q +o . ntit# &Ds ata collection e>ists. [1 Q . ntit# &Ds ata collection e>ists.
R2S	– Flag in icating %(et(er . ntit# &Ds ata collection e>ists for I , - o!nt RO*1S RangeK 3not t#/e. <nots of t(is t#/e (ave t(eir 3not range on RO*1S) an are not T#/e 0. [0 Q +o . ntit# &Ds ata collection e>ists. [1 Q . ntit# &Ds ata collection e>ists.
RBS	– Flag in icating %(et(er . ntit# &Ds ata collection e>ists for I , - o!nt R>$_1$*>$_2$S RangeK 3not t#/e. <nots of t(is t#/e (ave t(eir 3not range on R>$_1$*>$_2$S) an are not T#/e 1. [0 Q +o . ntit# &Ds ata collection e>ists. [1 Q . ntit# &Ds ata collection e>ists.

.>am/les of 3not vectors of T#/e 0*

```
0 0 X X 1 1
0 0 X X Y Y 1 1
0 0 X X Y Y Z Z 1 1
```

.>am/les of 3not vectors of T#/e 1*

```
0 0 1 1          1+ote* T(is is t(e e>ce/tion to T#/e 02
X X Y Y
X X Y Y Z Z
X X Y Y Z Z W W
```

.>am/les of 3not vectors of T#/e 2*

```
0 0 X 1 1
0 0 X Y 1 1
0 0 X Y Z 1 1
0 0 X X X 1 1
0 0 X X Y Z Z 1 1
```

.>am/les of 3not vectors of T#/e B*

```
X X Y Z Z
X X Y Z W W
```

=it(t(is information in (an) t(e rea er is a0le to reconstr!ct com/lete 3not vectors in t(e follo%ing manner. =(en reconstr!cting t(e 3not vector) #o! onl# ta3e 7!st eno!g(val!es from t(e eco e 3not val!e arra#. T(is ma# 0e as fe% as one. 'll t(e ot(er val!es are inferre . ;ere¡s a s3etc(of t(e reconstr!ction algorit(m*

```
// Number of knots in the knot vector
cNumKnots = numCtlPts + degree + 1;
// Necessary knot multiplicity at both ends of the knot vector
cClamping = degree + 1;
switch (knotType) {
    // Clamping is 0..1, internal knots occur in ADJACENT PAIRS
    // *EXCEPT* when there are no internal knots, in which case
    // Type = 1 instead.
    case 0: numVals = (cNumKnots - 2 * cClamping)/2;
    // Clamping is X1..X2, internal knots occur in ADJACENT PAIRS
    case 1: numVals = (cNumKnots - 2 * cClamping)/2 + 2;
    // Clamping is 0..1, and not Type 0
    case 2: numVals = (cNumKnots - 2 * cClamping);
    // Clamping is X1..X2, and not Type 1
    case 3: numVals = (cNumKnots - 2 * cClamping) + 2;
}
// numVals is the number of non-inferrable knot values needed
// Let vVals be the knot vector value array
// vKnot will be the final output knot vector
if (knotType is either 0 or 2)
    Set vKnot[0 .. cClamping-1] to 0
    Set vKnot[cNumKnots-cClamping .. cNumKnots-1] to 1
else
    Set vKnot[0 .. cClamping-1] to vVals[0]
    Set vKnot[cNumKnots-cClamping .. cNumKnots-1] to vVals[numVals-1]
Set vKnot[cClamping .. cNumKnots-cClamping-1] from vVals[1 .. numVals-
2]
```

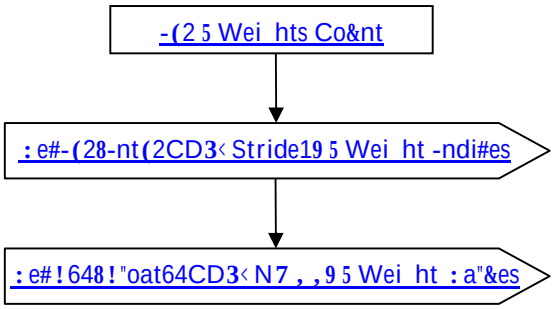
Vecl)#Blnt)#CDP(Stri*e1D 2 6ntit! In*e: Co*es

.ntit# &n e> -o es is a vector of A!antiHer Ico esK re/resenting entit# in e> i entifiers for a set of entities 1i.e. in ices to /artic!lar entities %it(in a list of entities2. .ntit# &n e> -o es !ses t(e &ntB2 version of t(e -,D.- to com/ress an enco e ata.

8.1.; Com resse* Control Point $eig%ts Data

-om/resse -ontrol Point =eig(ts Data collection is t(e com/resse an 5or enco e re/resentation of %eig(t ata for some set of -ontrol Points. 'll +4R:S 0ase geometr# !se t(is ata collection to com/ress5enco e -ontrol Point =eig(t ata.

!i &re 18%5 Com$ressed Contro" 3oint Wei hts Data data #o""e#tion



I)# 2 $eig%ts Count

=eig(ts -o!nt s/ecifies t(e total n!m0er of =eig(ts. T(is co!nt can iffer from t(e -ontrol Point co!nt 1see C.2.B.1.D.1.B +4R:S S!rface -ontrol Point -o!nts2 0eca!se if t(e -ontrol Point Dimensionalit# is non-rational 1see ata fiel +4R:S S!rface -ontrol Point Dimensionalit# in C.2.B.1.D.1 S!rfaces 6eometric Data2) t(en no =eig(t val!es are store for t(e /artic!lar -ontrol Point. =eig(ts -o!nt val!e also oes not necessaril# eA!ate to t(e act!al n!m0er of =eig(ts store) since if a /artic!lar -ontrol PointN's =eig(t val!es is I1K) t(en no act!al =eig(t val!e is store 1i.e. JT file loa ers5rea ers can infer t(at t(e =eig(t Val!e is I1K for -ontrol Points t(at onNt (ave a =eig(t val!e store 2.

VecI)#BInt)#CDP0 Stri*e1D 2 $eig%t In*ices

=eig(t &n ices is a vector of in ices re/resenting t(e in e> i entifiers for t(e con itional set of %eig(ts for %(ic(an act!al =eig(t Val!es is store in =eig(t Val!es. =eig(t &n ices !ses t(e &ntB2 version of t(e -,D.- to com/ress an enco e ata.

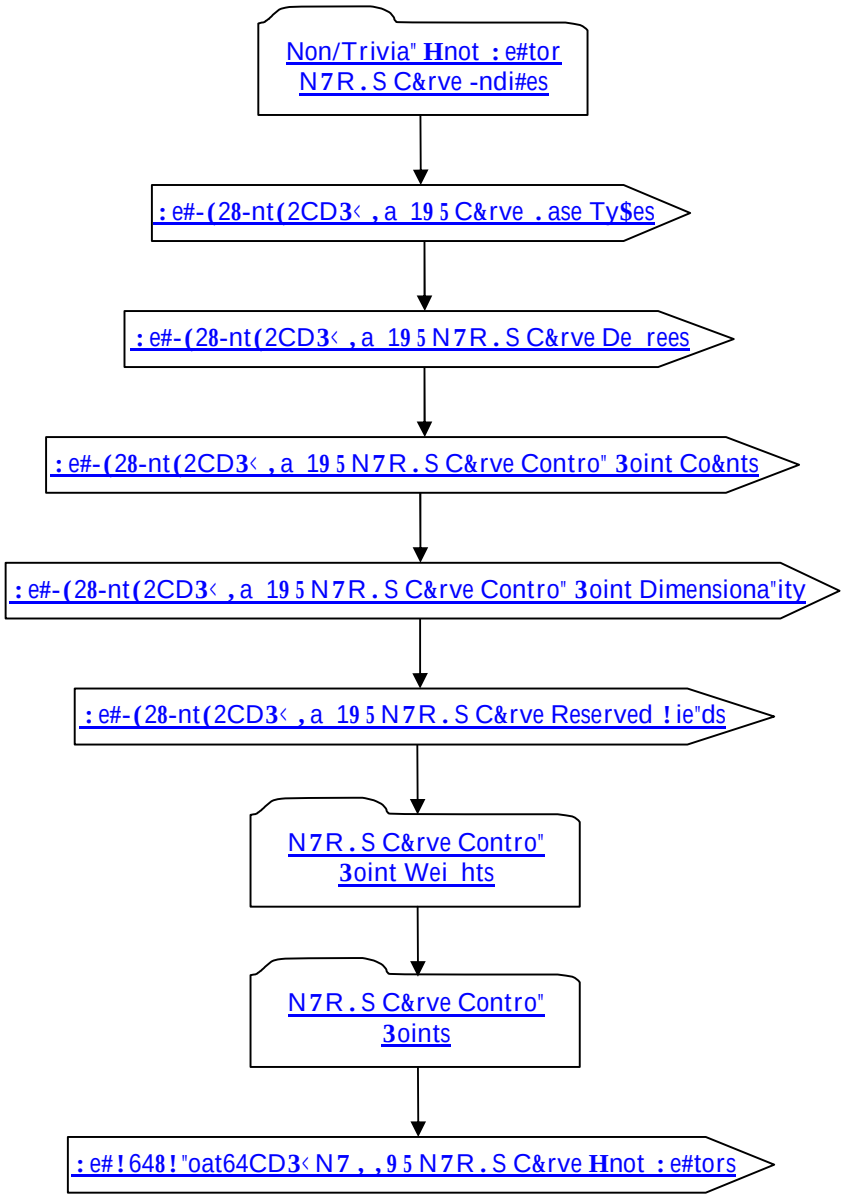
VecF.,BFloat.,CDP0 '1""D 2 $eig%t Values

=eig(t Val!es is a vector of %eig(t val!es for t(e con itional set of %eig(ts. =eig(t Val!es !ses t(e FloatFD version of t(e -,D.- to com/ress an enco e ata.

8.1.1< Com resse* Curve Data

-om/resse -!rve Data collection contains JT :-Re/ or =ireframe Re/ com/resse 5enco e geometric -!rve ata. -!rrentl# onl# +4R:S -!rve t#/es are s!//orte as /art of t(is ata collection. -om/lete oc!mentation for JT :-Re/ an =ireframe Re/ can 0e fo!n in sections C.2.B.1 JT :-Re/ .lement an C.2.E.1 =ireframe Re/ .lement res/ectivel#.

!i &re 1885 Com$ressed C&rve Data data #o""e#tion



Vecl)#BInt)#CDP("ag1D2 Curve 3ase T! es

.ac(-!rve is assigne a 0ase t#/e i entifier. -!rve :ase T#/es is a vector of 0ase t#/e i entifiers for eac(-!rve in a list of -!rves. -!rrentl# onl# +4R:S -!rve :ase T#/e is s!//orte)0!t a t#/e i entifier is still incl! e in t(e s/ecification to allo% for f!t!re e>/ansion of t(e JT Format to s!//ort ot(er c!rve t#/es.

&n an !ncom/resse 5 eco e form t(e - !rves 0ase t#/e i entifier val!es (ave t(e follo%ing meaning*

[1	– - !rve is a +4R:S c!rve

- !rve :ase T#/es !ses t(e &ntB2 version of t(e - , D. - to com/ress an enco e ata.

VecI)#BInt)#CDP("ag1D2 '1R3S Curve Degrees

+4R:S - !rve Degrees is a vector of - !rve egree val!es for eac(+4R:S - !rve in a list of - !rves 1t(ere is a store val!e for eac(+4R:S - !rve in t(e list2. +4R:S - !rve Degrees !ses t(e &ntB2 version of t(e - , D. - to com/ress an enco e ata.

VecI)#BInt)#CDP("ag1D2 '1R3S Curve Control Point Counts

+4R:S - !rve -ontrol Point -o!nts is a vector of control /oint co!nts for eac(+4R:S - !rve in a list of c!rves 1t(ere is a store val!e for eac(+4R:S - !rve in t(e list2. +4R:S - !rve -ontrol Point -o!nts !ses t(e &ntB2 version of t(e - , D. - to com/ress an enco e ata.

VecI)#BInt)#CDP("ag1D2 '1R3S Curve Control Point Dimensionalit!

+4R:S - !rve -ontrol Point Dimensionalit# is a vector of control /oint imensionalit# val!es for eac(+4R:S - !rve in a list of - !rve s1i.e. t(ere is a store val!es for eac(+4R:S - !rve in t(e list2.

&n an !ncom/resse 5 eco e form t(e control /oint imensionalit# val!es meaning is e/en ent !/on t(e +4R:S .ntit# t#/e.

For +4R:S 4V - !rve entities t(e imensionalit# val!e (as t(e follo%ing efinition*

[2	– +on-Rational 1eac(control /oint (as 2 coor inates2
[B	– Rational 1eac(control /oint (as B coor inates2

For +4R:S 89L - !rve entities t(e imensionalit# val!e (as t(e follo%ing efinition*

[B	– +on-Rational 1eac(control /oint (as B coor inates2
[D	– Rational 1eac(control /oint (as D coor inates2

+4R:S - !rve -ontrol Point Dimensionalit# !ses t(e &ntB2 version of t(e - , D. - to com/ress an enco e ata.

VecI)#BInt)#CDP("ag1D2 '1R3S Curve Reserve* Fiel*s

+4R:S - !rve Reserve Fiel s is a vector of ata reserve for f!t!re e>/ansion of t(e JT format. .ac(+4R:S - !rve in a list of - !rves (as one reserve ata fiel entr# in t(is +4R:S - !rve Reserve Fiel s vector. +4R:S - !rve Reserve Fiel s !ses t(e &ntB2 version of t(e - , D. - to com/ress an enco e ata

VecF.,BFloat.,CDP('1""D2 '1R3S Curve Anot Vectors

+4R:S - !rve <not Vectors is a list of 3not vector val!es for eac(+4R:S - !rve (aving non-trivial 3not vectors in a list of - !rves 1i.e. t(ere are store val!es for eac(non-trivial 3not vector +4R:S - !rve in t(e

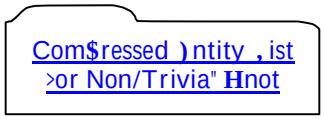
list2. ' ll t(ese +4R:S -!rve non-trivial 3not vectors are c!m!late into t(is single list in t(e same or er as t(e -!rve a//ears in t(e -!rve list 1i.e. -!rve-+ +on-Trivial <not Vector) -!rve-$ +on-Trivial <not Vector) etc.2. T(e +4R:S -!rves for %(ic(3not vectors are store 1i.e. t(ose containing non-trivial 3not vectors2 are i entifie in ata collection +on-Trivial <not Vector +4R:S -!rve &n ices oc!mente in 8.1.10.1 +on-Trivial <not Vector +4R:S -!rve &n ices. +4R:S -!rve <not Vectors !ses t(e FloatFD version of t(e -,D.- to com/ress an enco e ata.

8.1.1<.1 'on-Trivial Anot Vector '1R3S Curve In*ices

+on-Trivial <not Vector +4R:S -!rve &n ices ata collection s/ecifies t(e -!rve in e> i entifiers 1i.e. in ices to /artic!lar +4R:S -!rves %it(in a list of -!rves2 for all +4R:S -!rves containing non-trivial 3not vectors. ' escri/tion5 efinition for Inon-trivial 3not vectorK can 0e fo!n in 8.1.8 -om/resse .ntit# "ist for +on-Trivial <not Vector.

T(is -!rve in e> ata is store in a com/resse format.

!i &re 18'5 Non/Trivia" Hnot :e#tor N7R.S C&rve -ndi#es data #o""e#tion



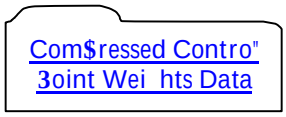
-om/lete escri/tion for -om/resse .ntit# "ist for +on-Trivial <not Vector can 0e fo!n in 8.1.8 -om/resse .ntit# "ist for +on-Trivial <not Vector.

8.1.1<.# '1R3S Curve Control Point $eig%ts

+4R:S -!rve -ontrol Point =eig(ts ata collection efines t(e =eig(t val!es for a con itional set of -ontrol Points for a list of +4R:S -!rves. T(e storing of t(e =eig(t val!e for a /artic!lar -ontrol Point is con itional) 0eca!se if +4R:S -!rve -ontrol Point Dimension is Inon-rationalK or t(e act!al -ontrol PointNs =eig(t val!e is I1K) t(en no =eig(t val!e is store for t(e -ontrol Point 1i.e. =eig(t val!e can 0e inferre to 0e I1K2.

T(e +4R:S -!rve -ontrol Point =eig(ts ata is store in a com/resse format.

!i &re 1'05 N7R.S C&rve Contro" 3oint Wei hts data #o""e#tion

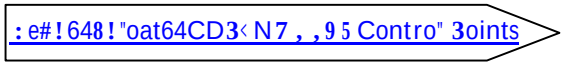


-om/lete escri/tion for -om/resse -ontrol Point =eig(ts Data can 0e fo!n in 8.1.9 -om/resse -ontrol Point =eig(ts Data.

8.1.1<.) '1R3S Curve Control Points

+4R:S -!rve -ontrol Points is t(e com/resse an 5or enco e re/resentation of t(e -ontrol Point coor inates for eac(+4R:S -!rve in a list of -!rves 1i.e. t(ere are store val!es for eac(+4R:S -!rve in t(e list2. +ote t(at t(ese are non-(omogeneo!s coor inates 1i.e. -ontrol Point coor inates (ave 0een ivi e 0# t(e corres/on ing -ontrol Point =eig(t val!es2.

!i &re 1'15 N7R.S C&rve Contro" 3oints data #o""e#tion



VecF.,BFloat.,CDP('1""D2 Control Points

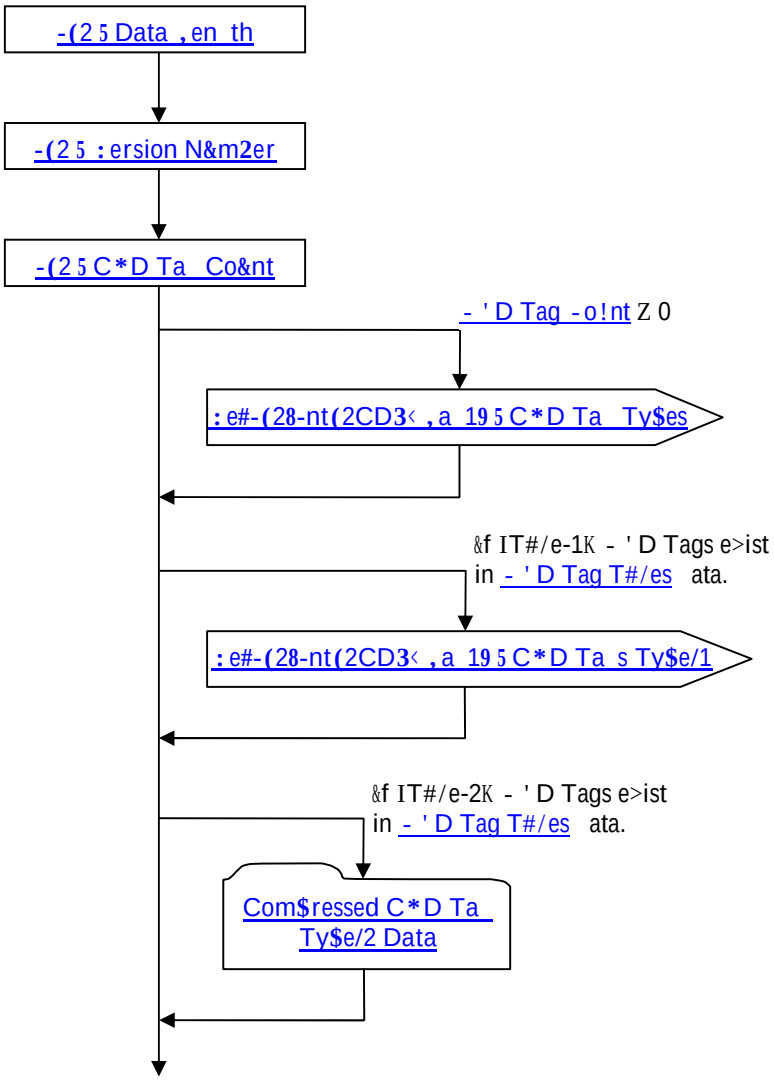
-ontrol Points is a vector of -ontrol Point coor inates for all t(e +4R:S -!rves in a list of -!rves. 'll t(e +4R:S -!rve -ontrol Point coor inates are c!m!late into t(is single vector in t(e same or er as t(e -!rve a//ears in t(e -!rve list 1i.e. -!rve-1 -ontrol Points) -!rve-2 -ontrol Points) etc.2. -ontrol Points !ses t(e FloatFD version of t(e -,D.- to com/ress an enco e ata in a IlosslessK manner.

8.1.11 Com resse* C+D Tag Data

T(e -om/resse -'D Tag Data collection contains t(e /ersistent &Ds) as efine in t(e -'D S#stem) to !niA!el# i entif# in ivi !al -'D entities 1e.g. Faces an . ges of a JT :-Re/) P$&) etc.2. .>actl# %(at -'D entit# t#/es (ave -'D Tags an %(at or er t(e# are store in -om/resse -'D Tag Data is efine 0# !sers of t(is ata collection 1e.g. C.2.B.1.F :-Re/ -'D Tag Data) C.2.F.2.C P$& -'D Tag Data2

=(at constit!tes a -'D Tag is o!tsi e t(e sco/e of t(e JT File format an is in ee /art of t(e -'D s#stem. T(e JT File format sim/l# /rovi es a %a# to store an# 3in of -'D Tag as /rovi e 0# t(e -'D s#stem %(ic(/ro !ce t(e -'D entit#.

!i &re 1'25 Com$ressed C*D Ta Data data #o""e#tion



I)#2 Data "engt%

Data "engt(s/ecifies t(e lengt(in 0#tes of t(e -om/resse - 'D Tag Data collection. ' JT file loa er5rea er ma# !se t(is information to com/!te t(e en /osition of t(e -om/resse - 'D Tag Data %it(in t(e JT file an t(!s s3i/ rea ing t(e remaining -om/resse - 'D Tag Data.

I)#2 Version 'umber

Version +!m0er is t(e version i entifier for t(e -om/resse - 'D Tag Data. Version n!m0er I1K is c!rrentl# t(e onl# vali val!e.

I)# 2 C+D Tag Count
- ' D Tag -o!nt s/ecifies t(e n!m0er of - ' D Tags

Vecl)#BInt)#CDP("ag1D 2 C+D Tag T! es
- ' D Tag T#/es is a vector of t#/e i entifiers for a list of - ' D Tags 1%(ere eac(- ' D Tag in t(e list (as a t#/e i entifier val!e2.

&n an !ncom/resse 5 eco e form t(e - ' D Tag t#/e i entifier val!es (ave t(e follo%ing meaning*

[1	– B2 : it &nteger - ' D Tag T#/e
[2	– FD : it &nteger - ' D Tag T#/e

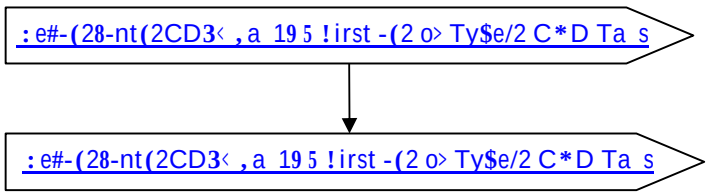
- ' D Tag T#/es !ses t(e &ntB2 version of t(e - , D . - to com/ress an enco e ata.

Vecl)#BInt)#CDP("ag1D 2 C+D Tags T! e-1
- ' D Tags T#/e-1 is a vector of t(e T#/e-1 1i.e. B2 : it &nteger T#/e2 - ' D Tags for a list of - ' D Tags. - ' D Tags T#/e-1 !ses t(e &ntB2 version of t(e - , D . - to com/ress an enco e ata. - ' D Tags T#/e-1 is onl# /resent if t(ere are T#/e-1 - ' D Tags in t(e - ' D Tag T#/es vector. T(!s a loa er5rea er of JT file m!st first !ncom/ress5 eco e an eval!ate t(e /revio!sl# rea - ' D Tag T#/es to etermine if t(ere are an# T#/e-1 - ' D Tags an if so) t(en t(e - ' D Tags T#/e-1 ata vector is /resent.

8.1.11.1 Com resse* C+D Tag T! e-# Data
-om/resse - ' D Tag T#/e-2 Data collection contains t(e T#/e-2 1i.e. FD : it integer T#/e2 - ' D Tag ata for a list of - ' D Tags.

T(e -om/resse - ' D Tag T#/e-2 Data collection is onl# /resent if t(ere are T#/e-2 - ' D Tags in t(e - ' D Tag T#/es vector. T(!s a loa er5rea er of JT file m!st first !ncom/ress5 eco e an eval!ate t(e /revio!sl# rea - ' D Tag T#/es vector to etermine if t(ere are an# T#/e-2 - ' D Tags an if so) t(en t(e -om/resse - ' D Tag T#/e-2 Data collection is /resent.

!i &re 1'(5 Com$ressed C*D Ta Ty$e/2 Data data #o""e#tion



Vecl)#BInt)#CDP("ag1D 2 First I)# of T! e-# C+D Tags
First &B2 of T#/e-2 - ' D Tags is a vector of t(e first B2 0its of eac(T#/e-2 - ' D Tag in t(e list of - ' D Tags. First &B2 , f T#/e-2 - ' D Tags !ses t(e &ntB2 version of t(e - , D . - to com/ress an enco e ata.

Vecl)#BInt)#CDP("ag1D 2 Secon* l)# of T! e-# C+D Tags

Secon &B2 of T#/e-2 - ' D Tags is a vector of t(e secon B2 0its of eac(T#/e-2 - ' D Tag in t(e list of - ' D Tags. Secon &B2 , f T#/e-2 - ' D Tags !ses t(e &ntB2 version of t(e - , D . - to com/ress an enco e ata.

8.# 6nco*ing +lgorit%ms

T(e follo%ing sections give a 0rief tec(nical overvie%5 escri/tions of t(e vario!s enco ing algorit(ms !se in t(e JT format. ' itional information on eac(of t(e algorit(ms can 0e fo!n %it(in references liste in B References an ' itional &nformation section of t(is oc!ment. ' lso) a sam/le im/lementation of t(e eco ing /ortion of eac(algorit(m can 0e fo!n in '//en i> -*Deco ing ' lgorit(ms Q ' n &m/lementation.

8.#.1 1niform Data >uanti=ation

4niform Data G!antiHation is a loss# enco ing algorit(m in %(ic(a contin!o!s set of in/!t val!es 1floating /oint ata2 is a//ro>imate %it(integral m!lti/liers 1i.e. integers2 of a common factor. ;o% close t(e A!antiHation o!t/!t a//ro>imates t(e original in/!t ata is e/en ent !/on t(e A!antiHation ata range an t(e n!m0er of 0its s/ecifie to (ol t(e res!lting integer val!e.

T(e A!antiHation is consi ere I!niformK 0eca!se t(e algorit(m ivi es t(e ata in/!t range into levels of eA!al s/acing 1i.e. a !niform scale2. T(e form of 4niform Data G!antiHation !se 0# t(e JT format is also consi ere scalar in nat!re) in t(at eac(in/!t val!e is treate se/aratel# in /ro !cing t(e o!t/!t integer val!e.

6iven t(e follo%ing efinitions*

in/!tVal*	in/!t floating /oint ata to A!antiHe
o!t/!tval*	res!lting A!antiHe o!t/!t integer val!e
min&n/!tRange*	s/ecifie minim!m val!e of in/!t ata range
ma>&n/!tRange*	s/ecifie ma>im!m val!e of in/!t ata range
n : its*	s/ecifie n!m0er of 0its of /recision 1A!antiHe siHe2

T(e 0asic algorit(m 1!sing - \ \ st#le s#nta>2 for 4niform Data G!antiHation is as follo%s*

4&ntB2 i $ a>-o e [1n : its Y B22 d 10>1 YY n : its2 - 1 * 0>ffffffff?

FloatFD enco e $!lti/lier [FloatFD1i $ a>-o e2 5 1ma>&n/!tRange Q min&n/!tRange2?

4&ntB2 o!t/!tVal [4&ntB21 1in/!tVal - min&n/!tRange2] enco e $!lti/lier \ 0.E 2?

> +ote* For reasons of ro0!stness) Io!t/!tValK m!st also 0e e>/licitl# clam/e to t(e range R0)i $ a>-o eS. T(is is 0eca!se floating-/oint ro!n off error in t(e calc!lation of Ienco e $!lti/lierK can ot(er%ise ca!se Io!t/!tValK to sometimes come o!t eA!al to Ii $ a>-o e \ 1K.

+ote t(at all com/ression algorit(ms in t(e follo%ing sections o/erate on A!antiHe integer ata.

J

) ;am$"e (5 3re>i; #ode to de#rement >ie"d width two times.

' /se! o-co e sam/le im/lementation of 0it lengt(eco ing is availa0le in '//en i> -*Deco ing 'lgorit(ms Q 'n &m/lementation.

8.#.) 0uffman C4D6C

T(e ;!ffman com/ression algorit(m is name after its inventor) Davi ;!ffman) an %as evelo/e in 19D8 %(ile $r. ;!ffman %as a P(.D. st! ent at t(e $assac(!setts &nstit!te of Tec(nolog# 1$&T2? t(e same #ear as -la! e S(annon of :ell "a0oratories /!0lis(e (is seminal /a/er I' mat(ematical t(eor# of comm!nicationK t(at la!nc(e t(e ne% fiel of &nformation T(eor#. &n t(at same class %it(Davi ;!ffman %as Peter .lias %(o re/orte l# evelo/e t(e first artic!lation of arit(metic co ing) 0!t it la# !n/!0lis(e !ntil 19CF) %(en Jorma Rissanen an Re(oo t o li 00ina c r(e

JT Fil

01101110100

So t(e n!m0er of 0its reA!ire to re/resent t(e arra#5seA!ence of integers in ; !ffman co e%or form is 11 0its 1i.e. total of I = eig(te -o e lengt(2) vers!s 192 0its in stan ar fi>e -siHe integer enco ing.

' /se! o-co e sam/le im/lementation of ; !ffman eco ing is availa0le in ' //en i> -*Deco ing ' lgorit(ms Q ' n &m/lementation.

8.#., +rit%metic C4D6C

' rit(metic enco ing is a lossless com/ression algorit(m t(at re/laces an in/ !t stream of s#m0ols or 0#tes %it(a single fi>e /oint o!t/ !t n!m0er 1i.e. onl# t(e mantissa 0its to t(e rig(t of t(e 0inar# /oint are o!t/ !t from $S: to "S:2. T(e total n!m0er of 0its nee e in t(e o!t/ !t n!m0er is e/en ent !/on t(e lengt(5com/le>it# of t(e in/ !t message 1i.e. t(e longer t(e in/ !t message t(e more 0its nee e in t(e o!t/ !t n!m0er2. T(is single fi>e /oint n!m0er o!t/ !t from an arit(metic enco ing /rocess m!st 0e !niA!el# eco a0le to create t(e e>act stream of in/ !t s#m0ols t(at %ere !se to create it.

&nitiall# all s#m0ols 0eing enco e (ave a /ro0a0ilit# val!e assigne to t(em 0ase on t(e li3eli(oo t(at t(e s#m0ol %ill occ!r ne>t in t(e in/ !t stream 1i.e. t(e freA!enc# of t(e s#m0ol in t(e in/ !t stream2. 6iven /ro0a0ilit# val!e assignments) eac(in ivi !al s#m0ol is t(en assigne an interval range along a nominal 0 to 1 I/ro0a0ilit# lineK) %(ere t(e siHe of eac(range corres/on s to t(e s#m0olNs /ro0a0ilit# val!e. +ote t(at a /artic!lar s#m0ol o%ns all val!es %it(in its assigne range !/ to) 0!t not incl! ing) t(e range (ig(val!e) an t(at it oes not matter %(ic(s#m0ols are assigne %(ic(segment of t(e range as long it is one in t(e same manner 0# 0ot(t(e enco er an t(e eco er.

6iven t(e a0ove escri0e in/ !t stream /ro0a0ilit# an interval range assignments) a (ig(level escri/tion of t(e arit(metic enco ing /rocess is as follo%s*

1. :egin %it(a Ic!rrent intervalK initialiHe to R0)12. +ote) t(at in interval range notation 1i.e. IR0)12K2) t(e IRI s#m0ol in icates incl!sive of t(e interval lo% limit an I2K s#m0ol in icates e>cl!sive of t(e interval (ig(limit.
2. SeA!entiall# for eac(s#m0ol of t(e in/ !t stream) /erform t%o ste/s
    a. S!0 ivi e t(e c!rrent interval into s!0intervals 0ase on t(e in/ !t stream s#m0ol /ro0a0ilit# val!es as escri0e a0ove.
    0. Select t(e s!0interval corres/on ing to t(e c!rrent in/ !t stream s#m0ol 0eing seA!entiall# /rocesse an ma3e it t(e ne% Ic!rrent intervalK.
B. ' fter all in/ !t stream s#m0ols (ave 0een seA!entiall# /rocesse ? o!t/ !t eno!g(0its to isting!is(t(e final Ic!rrent intervalK from all ot(er /ossi0le final intervals.

&n /se! o co e form) t(e algorit(m to accom/lis(t(e a0ove escri0e arit(metic enco ing for an in/ !t stream message of an# lengt(co!l loo3 as follo%s*

```
Set lo% to 0.0
Set (ig( to 1.0
= (ile t(ere are still in/ !t s#m0ols o
        c!rVs#m0ol [ get ne>t in/ !t s#m0ol
        range [ (ig( Q lo%
        (ig( [ lo% \ range ] (ig(Vrange1c!rVs#m0ol2
```

```
        lo% [ lo% \ range ] lo%Vrange1c!rVs#m0ol2
.n  of = (ile
, !t/!t lo%
```

So t(e arit(metic enco ing /rocess is sim/l# one in %(ic(%e narro% t(e range of /ossi0le n!m0ers %it(ever# ne% seA!entiall# /rocesse in/!t s#m0ol? %(ere t(e ne% narro%e range is /ro/ortional to t(e /re efine /ro0a0ilit# val!es assigne to eac(s#m0ol in t(e in/!t stream.

T(e arit(metic eco ing /rocess is t(e inverse /roce !re? %(ere t(e range is e>/an e in /ro/ortion to t(e /ro0a0ilit# of eac(s#m0ol as it is e>tracte . For t(e arit(metic eco ing /rocess %e fin t(e first s#m0ol in t(e message 0# seeing %(ic(s#m0ol o%ns t(e interval range t(at o!r enco e message falls in. T(en since %e 3no% t(e lo% an (ig(range limit val!es of t(e first s#m0ol %e can remove t(eir effects 0# reversing t(e /rocess t(at /!t t(em in.

&n /se! o co e form) t(e algorit(m for eco ing t(e incoming n!m0er co!l loo3 as follo%s*

```
6et enco e Vn!m0er
Do
        fin s#m0ol %(ose range stra les t(e enco e Vn!m0er
        o!t/!t t(e s#m0ol
        range [ s#m0olV(ig(Vval!e Q s#m0olVlo%Vval!e
        enco e Vn!m0er [ enco e Vn!m0er Q s#m0olVlo%Vval!e
        enco e Vn!m0er [ enco e Vn!m0er 5 range
!ntil no more s#m0ols
```

8.#.,.1 6:am le

Follo%ing is an e>am/le to emonstrate in /ractice t(e 0asic /rinci/les of arit(metic co ing.

S!//ose #o! %ant to com/ress) !sing arit(metic co ing) t(e follo%ing seA!ence5arra# of integer ata*

W2) 9) 12) 12) 0) C) 1) 20) E) 19X

For t(is in/!t stream of ata) t(e assigne /ro0a0ilit# val!es %ill 0e as follo%s*

N&m2er	3ro2a2i"ity
0	1510
1	1510
2	1510
E	1510
C	1510
9	1510
12	2510
19	1510
20	1510

T(en 0ase on eac(in/!t n!m0ers /ro0a0ilit# val!e) an interval range along a 0 to 1 I/ro0a0ilit# lineK can 0e assigne to eac(in/!t n!m0er as follo%s*

N&m2er	3ro2a2i"ity	Ran e
0	1510	R0.00) 0.102
1	1510	R0.10) 0.202
2	1510	R0.20) 0.B02
E	1510	R0.B0) 0.D02
C	1510	R0.D0) 0.E02
9	1510	R0.E0) 0.F02
12	2510	R0.F0) 0.802
19	1510	R0.80) 0.902
20	1510	R0.90) 1.002

+o% /rocee ing %it(enco ing t(e e>am/le in/!t integer seA!ence W2) 9) 12) 12) 0) C) 1) 20) E) 19X) t(e first n!m0er to 0e enco e is I2K? so t(e final enco e val!e %ill 0e a n!m0er t(at is greater t(an or eA!al to 0.20 an less t(an 0.B0. +o% as eac(s!0seA!ent n!m0er in t(e in/!t stream is seA!entiall# /rocesse for enco ing) t(e /ossi0le range of t(e o!t/!t n!m0er is f!rt(er restricte . &n o!r e>am/le t(e ne>t n!m0er to 0e enco e is I9K %(ic(o%ns t(e range R0.E0) 0.F02 %it(in t(e ne% s!0-range of R0.20) 0.B02? %(ic(no% f!rt(er restricts o!r o!t/!t n!m0er to t(e range R0.2E) 0.2F2. &f %e contin!e t(is logic for t(e com/lete in/!t integer seA!ence %e en !/ %it(t(e follo%ing*

New inte er n&m2er	, ow va"&e	6i h va"&e
	0.0	1.0
2	0.2	0.B
9	0.2E	0.2F
12	0.2EF	0.2E8
12	0.2EC2	0.2ECF
0	0.2EC20	0.2EC2D
C	0.2EC21F	0.2EC220
1	0.2EC21FD	0.2EC21F8
20	0.2EC21FCF	0.2EC21F8
E	0.2EC21FCC2	0.2EC21FCCF
19	0.25%216%%52	0.2EC21FCCEF

From t(e a0ove ta0le) are final lo% val!es is I0.2EC21FCCE2K %(ic(is t(e o!t/!t n!m0er t(at !niA!el# enco es t(e integer n!m0er seA!ence W2) 9) 12) 12) 0) C) 1) 20) E) 19X.

6iven t(is enco ing sc(eme) t(e eco ing %o!l sim/l# follo% t(e /rocess /revio!sl# escri0e . =e fin t(e first n!m0er in t(e seA!ence 0# loo3ing !/ in t(e /ro0a0ilit# range for t(e val!e) %(ose range) o!r enco e n!m0er I0.2EC21FCCE2K falls %it(in. &n o!r e>am/le t(is eA!ates to t(e val!e I2K an so o!r first eco e val!e m!st 0e I2K. T(en %e a//l# t(e /revio!sl# escri0e eco ing s!0traction an ivision ste/s to arrive at a ne% enco e val!e of I0.EC21FCCE2K. 4sing t(is ne% I0.EC21FCCE2K enco e val!e an t(e same logic of t(e first ste/) t(e secon eco e val!e %ill 0e I9K. =e contin!e t(is /rocess !ntil t(ere are no more n!m0ers to eco e.

&n /ractice) !e to floating /oint siHe 1i.e. n!m0er of 0its2 restrictions an /ossi0le ifferences in floating /oint formats on mac(ines) arit(metic enco ing is 0est im/lemente !sing 1F 0it or B2 0it integer mat(. 4sing 1F 0it or B2 0it integer mat() an incremental transmission sc(eme can 0e im/lemente) %(ere fi>e siHe integer state varia0les receive ne% 0its in at t(e lo% en an s(ift t(em o!t t(e (ig(en) forming a single n!m0er t(at can 0e as man# 0its long as are availa0le on t(e com/!terNs storage me i!m. 4sing o!r e>am/le as a g!i e) efine t(e starting range R0.0) 1.02 to instea 0e 0 to 0.999 1%(ic(is .111 in 0inar#2. T(en in or er to !se integer registers to store t(ese n!m0ers) 7!stif# t(e val!es so t(at t(e im/lie ecimal /oint is at t(e left (an si e of t(e %or . +o% loa t(e initial range val!es 0ase on t(e %or siHe %e are !sing. &n t(e case of a 1F 0it im/lementation t(e initial range val!es %ill 0e lo% eA!als 0>0000 an (ig(eA!als 0>FFFF. Since %e 3no% t(ese val!es %ill go on forever 1e.g. 0.999 P %ill contin!e %it(FFs2 %e can s(ift t(ose e>tra 0its in as nee e %it(no etrimental effects.

6oing 0ac3 to o!r e>am/le an !sing a E igit register) %e start %it(t(e range*

; ig(* 99999
"o%* 00000

' //l#ing t(e /revio!sl# escri0e enco ing algorit(m %e first calc!late t(e range 0et%een t(e lo% an (ig(val!es? %(ic(in t(is case is 100000 1not 9999 since %e ass!me t(e (ig(val!e (as an infinite n!m0er of 9Ns2. +e>t) %e calc!late t(e ne% (ig(val!e %(ic(in t(is e>am/le %ill 0e B0000. : !t 0efore %e store t(e ne% (ig(val!e %e m!st ecrement it to acco!nt for t(e im/lie igits a//en e to it? so ne% (ig(val!e %ill 0e 29999. ' //l#ing similar logic to com/!ting t(e ne% lo% val!e res!lts in a ne% range of*

; ig(* 29999 1999 P 2
"o%* 20000 1000 P 2

&n loo3ing at t(e ne%l# com/!te (ig(an lo% range val!es) it can 0e seen t(at t(e most significant igits of (ig(an lo% matc(. ' /ro/ert# of arit(metic co ing is t(at as t(is enco ing /rocess contin!es) t(e (ig(an lo% val!es %ill contin!e to get closer) 0!t %ill never matc(e>actl#. 6iven t(is /ro/ert#) once t(e most significant igit of (ig(an lo% matc() it %ill never c(ange) an t(!s %e can o!t/!t t(is most significant igit as t(e first n!m0er in t(e co e %or an contin!e %or3ing %it(7!st 1F 0it (ig(an lo% val!es. T(is o!t/!t /rocess is accom/lis(e 0# s(ifting 0ot(t(e (ig(an lo% val!es left 0# one igit an s(ifting in a I9K in t(e least significant igit of t(e (ig(val!e.

' //l#ing t(e /revio!sl# escri0e enco ing algorit(m an contin!ing t(e a0ove escri0e /rocess of s(ifting o!t most significant igit into t(e co e %or as (ig(an lo% contin!all# gro% closer toget(er loo3s as follo%s for enco ing o!r e>am/le integer n!m0er seA!ence W2) 9) 12) 12) 0) C) 1) 20) E) 19X*

	6i h	,ow	Ran e	C&m&"ative o&t$&t
&nitial State	99999	00000	100000	
.nco e I2K R0.2) 0.B2	29999	20000		
Shi>t o&t 2	99999	00000	100000	.2
.nco e I9K R0.E) 0.F2	E9999	E0000		.2
Shi>t o&t 5	99999	00000	100000	.2E
.nco e I12K R0.F) 0.82	C9999	F0000	20000	.2E
.nco e I12K R0.F) 0.82	CE999	C2000		.2E
Shi>t o&t %	E9999	20000	D0000	.2EC
.nco e IOK R0.0) 0.12	2B999	20000		.2EC
Shi>t o&t 2	B9999	00000	D0000	.2EC2

	6i h	,ow	Ran e	C&m&"ative o&t$&t
.nco e ICK R0.D) 0.E2	19999	1F000		.2EC2
Shi>t o&t 1	99999	F0000	D0000	.2EC21
.nco e I1K R0.1) 0.22	FC999	FD000		.2EC21
Shi>t o&t 6	C9999	D0000	D0000	.2EC21F
.nco e I2OK R0.9) 1.O2	C9999	CF000		.2EC21F
Shi>t o&t %	99999	F0000	D0000	.2EC21FC
.nco e IEK R0.B) 0.D2	CE999	C2000		.2EC21FC
Shi>t o&t %	E9999	20000	D0000	.2EC21FCC
.nco e I19K R0.8) 0.92	EE999	E2000		.2EC21FCC
Shi>t o&t 5	E9999	20000	D0000	.2EC21FCCE
Shi>t o&t 2				.2EC21FCCE2
Shi>t o&t 0				.2EC21FCCE20

's can 0e seen in t(e a0ove ta0le) after all val!es in t(e in/!t stream (ave 0een enco e an an# final matc(ing most significant igit (as 0een o!t/!t) t(e arit(metic co ing algorit(m reA!ires t(at t%o e>tra igits 0e s(ifte o!t of eit(er t(e (ig(or lo% val!e to finis(!/ t(e c!m!lative o!t/!t %or .

'lt(o!g(t(e a0ove e>am/le incrementall# enco es ver# nicel# %it(t(e arit(metic co ing algorit(m) t(ere are certain cases %(ere t(e com/!te (ig(an lo% val!es get closer) 0!t never act!all# converge to one val!e in t(e most significant igit 1e.g. ; ig([0.B00001) "o% [0.299922. T(!s after a fe% iterations t(e ifference 0et%een (ig(an lo% 0ecomes so small t(at 1F 0its is not s!fficient to re/resent an# ifference 0et%een t(e val!es 1i.e. all calc!lations ret!rn t(e same val!es2. T(is con itions is 3no%n as I!n erflo%K an s/ecial logic m!st a e to t(e arit(metic co ing algorit(m to recogniHe t(at I!n erflo%K is occ!rring an t(!s (ea it off 0efore t(e com/!tations reac(an im/asse.

T(e a itional logic for recogniHing t(at I!n erflo%K is occ!rring %o!l 0e e>ec!te after eac(recalc!lation of ; ig(an "o% val!e set) an in /se! o co e form t(is logic %o!l loo3 as follo%s*

```
!n erflo% [ F'"S.
if1 1 ; ig( an "o% val!eNs significant igits onNt matc( 0!t are on a 7acent n!m0ers2 TT
    12^n most significant igit of ; ig( is IOK an t(e 2^n most significant igit of lo% is I9K2 2
W
        !n erflo% [ TR4.
X
```

=(en5&f it is i entifie t(at I!n erflo%K is occ!rring) t(e enco ing algorit(m m!st /erform t(e follo%ing ste/s to sto/ t(e c!rrent I!n erflo%K*

- Delete t(e 2^n most significant igit from 0ot(t(e ; ig(an "o% val!e.
- S(ift t(e ot(er igits 1t(ose to t(e rig(t of t(e elete 2^n igit2 to t(e left to fill !/ t(e s/ace 1note t(at t(e most significant igit sta#s in /lace2.
- &ncrement a co!nter to remem0er t(at %e t(re% a%a# a igit an onNt 3no% %(et(er it %as going to converge to IOK or I9K.

' 0efore an after e>am/le of /erforming t(e a0ove ste/s to t(e ; ig(an "o% val!es %(en U!n erflo%K occ!rs is as follo%s*

	.e>ore	*>ter
;ig(	D0BDD	DBDD9
"o%	B9810	B8100
4n erflo%Vco!nter	0	1

+o% as t(e enco ing algorit(m contin!es an t(e most significant igit of ;ig(an "o% val!es once again converge to a common val!e) t(en t(at val!e m!st 0e o!t/!t to t(e co e %or along %it(I4n erflo%Vco!nterK n!m0er of I!n erflo%K igits t(at %ere /revio!sl# elete . T(e !n erflo% igits o!t/!t to t(e co e %or %ill eit(er 0e all 9s or 0s) e/en ing on %(et(er t(e ;ig(an "o% val!e converge to t(e (ig(er or lo%er val!e.

' /se! o-co e sam/le im/lementation of arit(metic eco ing is availa0le in ' //en i> -*Deco ing ' lgorit(ms Q ' n &m/lementation.

8.#.- Deering 'ormal C4D6C

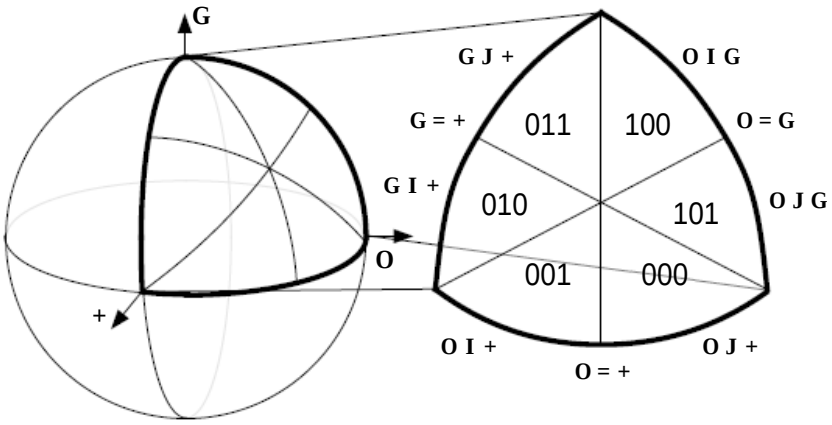
$. Deering first /!0lis(e (is %or3 on geometr# com/ression in 199E RES an later (el/e /resent a co!rse on t(e s!07ect at S&6 6R ' P ;N99 RFS. ' lt(o!g(DeeringNs a//roac(to geometric com/ression involves com/ression of vertices) colors an normals) t(e escri/tion etaile (ere %ill foc!s solel# on com/ression of normals since t(is is t(e onl# com/onent of DeeringNs a//roac(!se in t(e JT format.

T(ro!g(0ot(t(eoretical e>amination an em/irical testing) Deering fo!n t(at an ang!lar ensit# of 0.01 ra ians 0et%een normals 1a0o!t 100)000 normaliHe normals istri0!te over !nit s/(ere2 gave res!lts t(at %ere not vis!all# isting!is(a0le from res!lts o0taine from finer normal re/resentations. T(is o0servation re !ce t(e /ro0lem of (aving to Ie>actl#K re/resent an# general s!rface normal) to onl# (aving to re/resent a0o!t 100)000 s/ecific normals 1i.e. general s!rface normal re/lace 0# t(e a//ro/riate one of t(e 100)000 s/ecific normals2.

&f t(ere %ere no r!n-time memor# concerns an no concerns for on is3 foot/rint siHe) t(ese s/ecific 100)000 normals co!l 0e sim/l# re/resente in a ta0le t(at is in e>e into) to reference a /artic!lar normal. &nstea) DeeringNs a//roac(leverages s#mmetrical /ro/erties of t(e !nit s/(ere to re !ce t(e siHe of t(e ta0le an allo% an# normal to 0e re/resente 0#) at ma>) an 18 0it in e> as s!mmariHe 0elo%*

- ' ll normals are normaliHe 1i.e. can 0e re/resente as /oints on t(e s!rface of t(e !nit s/(ere2.
- 4nit s/(ere is ivi e into eig(t s#mmetrical octants 0ase on sign 0its of normalNs 8)9)L rectilinear re/resentation 1see Fig!re 19E2. 4sing t(ree 0its to re/resent t(e t(ree sign 0its of t(e normals 8 9L com/onents re !ces t(e /ro0lem s/ace to one eig(t(of t(e !nit s/(ere
- .ac(octant of t(e !nit s/(ere is ivi e into si> i entical se>tants 0# fol ing a0o!t t(e /lanes of s#mmetr#? >[#) >[H) an #[H 1see Fig!re 19E2. T(e /artic!lar se>tant can 0e enco e !sing anot(er t(ree 0its. So no% !nit s/(ere is ivi e into D8 i enticall# s(a/e triangle /atc(es re !cing t(e normal loo3-!/ ta0le to a0o!t 2000 entries 1i.e. 100000 5D82.
- T(en) a local rectang!lar ort(ogonal t%o imensional gri is create on t(e se>tant an all normals %it(in t(e se>tant are re/resente as t%o n-0it ang!lar a resses 1i.e. a A!antiHation of t%o ang!lar val!es along t(e !nit s/(ere2 %(ere InK is in t(e range from 0 to F 0its.
- Res!lting in a ma> gran total of 18 0its 1B \ B \ F \ F2 to re/resent an# normal on t(e !nit s/(ere.

!i &re 1'55 S$here divided into ei ht o#tants and o#tant divided into si; se;tants with ea#h se;tant assi ned an identi>yin three 2it #ode.



+ote t(at t(e se>tant t(ree 0it co e assignments !se 0# t(e JT format 1as seen in Fig!re 19E2 are slig(tl# mo ifie from t(e original assignments as s/ecifie 0# Deering.

T(e re/resentation of all normals %it(in a se>tant 0# t%o n-0it ang!lar a resses) as s!mmariHe a0ove) is 0ase on t(e follo%ing*

- &n s/(erical coor inates) /oints on a !nit s/(ere can 0e /arameteriHe 0# t%o angles) e an f? %(ere e is t(e angle a0o!t t(e # a>is an f is t(e longit! inal angle from t(e #[0 /lane.
- $a//ing 0et%een rectang!lar an s/(erical coor inates is*

  > [cose] cosf # [sinf H [sine] cosf

- 'll enco ing ta3es /lace in t(e /ositive octant.
- 'ngles e an f can 0e A!antiHe into t%o n-0it integers e$_n$ an f$_n$ 1%(ere InK is in t(e range of 0 to F2 an t(e relations(i/ 0et%een t(ese n-0it integers an angles e an f for a given InK is*

  e 1eN_n2 [asin tan 1f$_{max}$] 1n Q eN_n2 5 2^n 2

  f 1fN_n2 [f$_{max}$] fN_n 5 2^n

T(!s to enco e 1i.e. A!antiHe2 a given normal N into eN_n an fN_n*

- N m!st 0e first re/resente 1see Fig!re 19E2 in t(e /ositive octant an a//ro/riate se>tant %it(in t(at octant) res!lting in N'.
- T(en N' m!st 0e otte %it(all A!antiHe normals in t(e se>tant.
- For a fi>e InK) t(e corres/on ing eN_n an fN_n val!es of t(e A!antiHe se>tant normal t(at res!lt in t(e largest 1nearest !nit#2 ot /ro !ct efines t(e /ro/er eN_n an fN_n enco ing of N.

=it(t(is enco ing of normal N into eN_n an fN_n n-0it integers t(e com/lete 0it re/resentation of normal N can no% 0e efine as follo%s*

- 4//ermost t(ree 0its s/ecif# t(e octant.
- +e>t t(ree 0its s/ecif# t(e se>tant co e as efine in Fig!re 19E.
- +e>t t%o n-0it fiel s s/ecif# eN_n an fN_n val!es res/ectivel#.

8.) 8"I3 Com ression

L"&: com/ression is a lossless ata com/ression algorit(m an is essentiall# t(e same as t(at in gHi/ an Li/. LIiONs com/ression met(o) calle eflation) creates com/resse ata as a seA!ence of 0loc3s. T(e JT format !ses *Version 1.1.2* of t(e L"&: com/ression li0rar#.

; 3est Practices

T(e /rocee ing sections of t(is oc!ment s/ecif# t(e man ator# cla!ses for creating a reference com/liant Version 8.1 JT file. T(is I:est PracticesK section foc!sing on oc!menting format conventions t(at alt(o!g(not reA!ire to (ave a reference com/liant JT file) (ave 0ecome common/lace %it(in JT format translators to t(e /oint %(ere t(ese conventions are consi ere *best practices* for constr!cting JT files.

;.1 "ate-"oa*ing Data

From its ince/tion) t(e JT format %as esigne to scale from re/resenting ata necessar# for lig(t%eig(t %e0-0ase vie%ing) to re/resenting ata necessar# for f!ll /ro !ct igital moc3!/ an BD /ro !ct efinition. T(is a0ilit# of t(e JT format to re/resent s!c(a ro0!st BD /ro !ct efinition allo%s a single JT format 0ase BD igital asset to 0e leverage across t(e e>ten e enter/rise 0# man# issimilar a//lications %it(var#ing ata nee s5reA!irements.

=it(t(is s(aring of t(e single JT format 0ase BD igital asset 0# man# issimilar a//lications) comes t(e nee to 0e I/erformance sensitiveK 10ot(in r!ntime memor# foot/rint an act!al ata loa time2 to e>actl# %(at) (o% m!c() an %(en certain JT format ata m!st 0e loa e . To t(at en t(e JT format %as esigne 5str!ct!re to s!//ort not reA!iring all segments of ata to 0e seA!entiall# loa e 5rea in one /ass. T(is conce/t of ela#ing t(e loa ing of segments of ata !ntil act!all# nee e is referre to %it(in t(is JT Format Reference oc!ment as Ilate-loa ing ataK. T(e JT format (as man# str!ct!res in s!//ort of t(is conce/t of late-loa ing ata an it is recommen e as a 0est /ractice t(at %riters5loa ers of JT ata leverage t(ese constr!cts accor ingl#. .>am/les of t(ese JT format constr!cts in s!//ort of 10!t not necessaril# late-loa ing ata incl! e t(e follo%ing 1note t(at Iin s!//ort ofK oes not necessaril# mean t(at t(e constr!ct 1e.g. T, - Segment2 is onl# !se for /!r/oses of late loa ing ata2*

- T, - Segment
- Partition +o e . lement
- $eta Data +o e . lement
- "ate "oa e Pro/ert# 'tom . lement

;.# 3it Fiel*s

&n t(e C File Format section of t(is reference man# 0it fiel ata escri/tions 1e.g. C.2.1.1.1.1.1 :ase +o e Data I+o e FlagsK fiel 2 contain t(e %or s I*All undocumented bits are reserved*.K T(ese %or s s(o!l 0e inter/rete to mean t(at t(ese !n oc!mente 0its s(o!l 0e set to IOK %(en %riting t(e 0it fiel ata to a JT file.

;.) Reserve* Fiel*

&n t(e C File Format section of t(is reference some ata fiel s ma# 0e name 5 oc!mente IReserve Fiel K 1e.g. C.2.1.1.1.C.1", D +o e Data KReserve Fiel K fiel 2. ' IReserve Fiel K e>ists for /otential f!t!re e>/ansion of t(e Format an 0est /ractices s!ggests t(at t(ese fiel s s(o!l 0e treate as follo%s*

- &f #o! are %riting a JT file %(ose ata i not originate from rea ing a /revio!s JT file) t(en Reserve Fiel s s(o!l 0e set to a val!e a IOK %(en %riting t(e fiel to a JT file.
- &f #o! are %riting a JT file %(ose ata originate from rea ing a /revio!s JT file 1i.e. re%riting a JT File2) t(en IReserve Fiel sK s(o!l 0e %ritten %it(t(e same val!e t(at %as rea from t(e originating JT file.

;., 9eta*ata Conventions

' lt(o!g(t(ere are reall# no restrictions5limits5reA!irements on %(at meta ata 1i.e. /ro/erties2 can5m!st 0e attac(e to no es in t(e "S6 in or er to (ave a reference com/liant JT file) t(ere are some conventions t(at (ave 0een generall# follo%e in t(e in !str# %(en translating - ' D ata to t(e JT file format. See C.2.1.2 Pro/ert# ' tom .lements section of t(is oc!ment for com/lete escri/tion of t(e file .lements !se to attac(t(is /ro/ert# information to no es.

;.,.1 C+D Pro erties

T(e follo%ing ta0le lists t(e conventions t(at - ' D ata translators t#/icall# 1alt(o!g(not al%a#s2 follo% in /lacing - ' D information in a JT file as /ro/erties on vario!s "S6 no es. Some of t(ese /ro/erties are consi ere reA!ire in or er for t(e ata in t(e file to 0e inter/rete correctl# %(ile ot(er /ro/erties are o/tional. See flo%ing s!0-sections for a itional information on reA!ire vers!s o/tional /ro/erties.

T(e convention is to /lace t(ese 4nits /ro/erties on ever# Part an ' ssem0l# gro!/ing no e in t(e "S6. :# follo%ing t(is convention) JT file format rea ers5%riters are /rovi e ma>im!m fle>i0ilit# in !n erstan ing5in icating t(e a//ro/riate JT ata !nit /rocessing for 0ot() monolit(ic an s(attere JT file ' ssem0l# str!ct!res.

JT Pro ert! Ae!	9eaning	JT File Data T! e	6nco*e* Data T! e	Vali* Values	ReGuire* E 4 tional
JTVPR,PV$.'S4R.$.+TV4+&TS	$o el 4nits	$0String	$0String	millimeters centimeters meters inc(es feet #ar s	ReA!ire

JT Pro ert! Ae!	9eaning	JT File Data T! e	6nco*e* Data T! e	Vali* Values	Re6uire* E 4 tional
				micrometers ecimeters 3ilometers mils miles	
- 'DV$ 'SSV4+&TS	4nits of mass	$0String	$0String	micrograms milligrams grams 3ilograms o!nces /o!n s	ReA!ire
- 'DVS4RF' -.V'R.'	S!rface area of soli s %it(in /art.	$0String	FFD	n!meric	, /tional
- 'DVV,"4$.	Vol!me of soli s %it(in /art	$0String	FFD	n!meric	, /tional
- 'DVD.+S&T9	Densit# of soli s %it(in /art 1F2	$0String	FFD	n!meric	, /tional
- 'DV$ 'SS	$ass or %eig(t of soli s %it(in /art	$0String	FFD	n!meric	, /tional
- 'DV-.+T.RV,FV6R'V&T9	-enter of gravit# of soli s %it(in /art	$0String	B s/ace se/arate FFD	B n!meric val!es	, /tional
- 'DVPR,PV$ 'T.R&'"VT;&-<+.SS	S(eet t(ic3ness %it(in /art	$0String	FFD	n!meric	, /tional
- 'DVP'RTV+'$.	-om/onent name from translator	$0String	$0String	Y*string*Z	, /tional
- 'DVS,4R-.	-'D /rorgam t(er Part originate from	$0String	$0String	YstringZ	, /tional

Ta2"e 85 C*D 3ro$erty Conventions

;.,.1.1 Re6uire* Pro erties

T(e reA!ire !nit /ro/erties are reall# necessar# for vie%ers of JT file ata to /ro/erl# inter/ret n!meric ata for anal#sis o/erations 1e.g. meas!re2 an s!//ort t(e 0!il ing of assem0lies t(ro!g(t(e rea ing of m!lti/le JT files in is/arate !nits. T(ere are t%o !nits of meas!re t(at are relevant) !nits of istance an !nits of %eig(t.

T(e JTVPR,PV$.'S4R$.+TV4+&TS /ro/ert# is !se to s/ecif# !nits of istance. T(e -'DV$ 'SSV4+&TS /ro/ert# is !se to s/ecif# !nits for %eig(t. JTVPR,PV$.'S4R$.+TV4+&TS /ro/ert# is strictl# reA!ire) %(ile -'DV$ 'SSV4+&TS /ro/ert# is bo/tionall# reA!ire b. :# Io/tionall# reA!ire K) %e mean) it is reA!ire if ot(er o/tional meta ata inten s to s/ecif# /ro/erties t(at %o!l e/en on t(ese !nits of meas!re 1e.g. -'DVD.+S&T9 an -'DV$ 'SS2. +otice t(at t(e $ass !nits are s/ecifie) instea of t(e Densit# !nits) since Densit# is a erive !nit of $ass5Vol!me.

;.,.1.# 4 tional Pro erties

, /tional /ro/erties can 0e /rovi e) 0!t if t(e /ro/ert# is a !nits 0ase val!e) t(en t(e val!e m!st 0e in !nits t(at are consistent %it(t(e JTVPR , PV $. ' S4R $. +TV4 +&TS an - ' DV $ ' SSV4 +&TS /ro/erties. T(!s t(e !nits for t(e o/tional !nits 0ase /ro/erties m!st 0e as follo%s*

4$tiona" 3ro$erty	7nits
- ' DVS4RF ' - . V ' R . '	1JTVPR , PV $. ' S4R . $. +TV4 +&TS2[2]
- ' DVV , " 4 $.	1JTVPR , PV $. ' S4R . $. +TV4 +&TS2[B]
- ' DVD . +S&T9	- ' DV $ ' SSV4 +&TS51JTVPR , PV $. ' S4R . $. +TV4 +&TS2
- ' DV $ ' SS	- ' DV $ ' SSV4 +&TS
- ' DV - . +T . RV , FV6R ' V&T9	JTVPR , PV $. ' S4R . $. +TV4 +&TS
- ' DVPR , PV $ ' T . R& ' "VT ; & - < + . S S	JTVPR , PV $. ' S4R . $. +TV4 +&TS

Ta2"e '5 C*D 4$tiona" 3ro$erty 7nits

+ote of ca!tion regar ing t(e no e /lacement for t(e - ' DVD . +S&T9 /ro/ert#. Follo%ing t(e recommen e convention for t(e /lacing of - ' D /ro/erties 1see escri/tion in 9.D.1 - ' D Pro/erties2 im/lies t(at all soli s %it(in a single JT /art are of a !niform ensit#) %(ic(ma# not 0e tr!e in all cases.

;.,.# Tessellation Pro erties

= (en ealing %it(facette gra/(ical re/resentations 1i.e. " , Ds2 of /recise mo els 1e.g. JT : -Re/2) e/en ing on t(e esire !se it is often !sef!l5necessar# to 3no% %(at tessellation tolerances %ere !se to generate t(e facette re/resentation. To t(at en) t%o /ro/erties are t#/icall# store on Part +o e . lements 1if /art also (as /recise mo el2 to in icate t(e tessellation tolerances !se to generate eac(" , D. T(ese t%o tessellation /ro/erties are as follo%s

;.,.) 9iscellaneous Pro erties

T(e 0elo% ta0le oc!ments some miscellaneo!s /ro/erties often /lace on vario!s no es in t(e "S6 to comm!nicate s/ecific information a0o!t t(e no e or its contents.

JT Pro ert! Ae!	9eaning	JT File Data T! e	6nco*e* Data T! e	Vali* Values
P$&VT9P.VT':".	$a# 0e attac(e to Part +o e .lement to in icate t(e list of P$& t#/e val!es an associate names for all P$& t#/es 10asicall# eA!ivalent to t(e .ntit# T#/e fiel oc!mente in 6eneric P$& .ntities2. T(e string is a I.K an I)K elimite string of t(e follo%ing form* I10.6roove =el)11.Fillet =el)12.Pl!g5Slot =el)1D.. ge =el K	$0String	Y*string*Z	
JTVPR,PVS;'P.VD'T'VT9P.	$a# 0e attac(e to S(a/e +o e .lements to in icate %(at geometr# t#/e t(e s(a/e ata re/resents.	$0String	Y*string*Z	IS!rfaceK I=ireK
JTVPR,PVTR&STR&PVD'T'V"'9,4T	$a# 0e attac(e to Tri-Stri/ Set S(a/e +o e .lement to in icate t(at t(e SetNs tri-stri/ /rimitives are sorte s!c(t(at stri/s of lengt(1 1i.e. triangles2 come first an t(en stri/s of lengt(2 1i.e. A!a s2 ne>t an t(en all ot(er stri/s of lengt(greater t(an 2 follo% in no /artic!lar or er.	$0String	Y*string*Z	ITriStri/sSorte K
JTVPR,PV,R&6&+'T&+6V:R.PT9P.	$a# 0e attac(e to Part +o e .lement to in icate t(e t#/e of :-Re/ associate %it(t(e Part.	$0String	Y*string*Z	I+oneK IJt:re/K I8T:re/K
JTVPR,PV+'$.	$a# 0e attac(e to an# form of no e or attri0!te %it(%(ic(one %ants to associate a te>t!al name 1e.g. Part5'ssem0l#5&nstance name) $aterial name) "ig(t Set name) etc.2. For Part5'ssem0l#5&nstance names t(is string (as t(e follo%ing enco e form %(ere I?K is a elimiter an I*N is a terminator*	$0String	Y*string*Z	

JT Pro ert! Ae!	9eaning	JT File Data T! e	6nco*e* Data T! e	Vali* Values
	"AlignmentPin.part;0;1:" +ame Version J &nstance J For attri0!te names t(is string (as t(e follo%ing enco e form* "Chrome material" +ame			

;.- "S5 +ttribute +ccumulation Semantics

For a//lications /ro !cing or cons!ming JT format ata) it is im/ortant t(at t(e JT format semantics of (o% attri0!tes are meant to 0e a//lie an acc!m!late o%n t(e "S6 are follo%e . &f not follo%e) t(en consistenc# 0et%een t(e a//lications in terms of BD /ositioning an ren ering of "S6 mo el ata %ill not 0e ac(ieve .

'lt(o!g(eac(attri0!te t#/e efines its o%n a//lication an acc!m!lation "S6 semantics 1t(e etails of %(ic(can 0e fo!n in eac(attri0!te t#/e s!0-section !n er C.2.1.1.2 'ttri0!te .lements2) t(ere are some general r!les %(ic(a//l#*

1. 'ttri0!tes at lo%er level in t(e "S6 ta3e /rece ence an re/lace or acc!m!late %it(attri0!tes set at (ig(er levels
2. +o es %it(o!t associate attri0!tes in(erit t(ose of t(eir /arents.
B. 'ttri0!tes in(erit onl# from t(eir /arents) t(!s a no e№s attri0!tes o not affect t(at no e№s si0lings.
D. T(e root of a /artition in(erits t(e attri0!tes in effect at t(e referring /artition no e.
E. 'ttri0!tes can 0e eclare IfinalK) %(ic(terminates acc!m!lation of t(at attri0!te t#/e at t(at attri0!te an /ro/agates t(e acc!m!late val!es t(ere to all escen ants of t(e associate no e. Descen ants can e>/licitl# o a one-s(ot overri e of IfinalK !sing t(e attri0!te IforceK flag) 0!t o not 0# efa!lt. +ote t(at IforceK oes not t!rn , FF IfinalK Q it is sim/l# a one-s(ot overri e of IfinalK for t(e s/ecific attri0!te mar3e as Iforcing.K 'n analog# for t(is IforceK an IfinalK interaction is t(at IfinalK is a 0ac3- oor in t(e attri0!te acc!m!lation semantics) an t(at IforceK is a ogg#- oor in t(e 0ac3- oor_

;.. "S5 Part Structure

T(e JT Format Reference oes not man ate t(at a /artic!lar no e (ierarc(# 0e !se for mo eling /(#sical Parts %it(in a "S6 str!ct!re. &n fact t(ere are man# no e (ierarc(ies for re/resenting Parts in "S6 t(at %ill f!nction correctl# in most JT ena0le a//lications. Still) t(ere is a convention t(at most JT translators

follo% 1an some JT ena0le a//lications ma# ass!me e>ists2 for mo eling Parts %it(in a "S6. T(e convention is to mo el eac(Part %it(in a "S6 str!ct!re %it(t(e follo%ing no e (ierarc(#*

!i &re 1'65 FT !ormat Convention >or 0 ode"in ea#h 3art in ,S1

mTdm( )T j m-14 912

+ en*i: +2 4b7ect T! e l*entifiers

' ll o07ects store in a JT file are classifie 0# t#/e an t(!s incl! e an o07ect t#/e i entifier as /art of t(eir /ersiste ata. T(e ata format for t(ese , 07ect T#/e i entifiers is a 6 4&D. T(ese , 07ect T#/e i entifiers are consistent for all o07ects) of a /artic!lar t#/e) in all Version 8.1 JT files.

Ta0le 10* , 07ect T#/e & entifiers lists t(e assigne i entifier for eac(, 07ect T#/e t(at can e>ist in a Version 8.1 JT file.

GUID	42?e#t Ty$e
0>ffffffff) 0>ffff) 0>ffff) 0>ff) 0>ff) 0>ff) 0>ff) 0>ff) 0>ff) 0>ff	& entifier to signal .n -, f-.lements.
Ty$es Stored Within ,S1 Se ment BSe ment Ty$e = 1C	
0>10 10BE) 0>2ac8) 0>11 1) 0>90) 0>F0) 0>00) 0>80) 0>cC) 0>00) 0>E9) 0>9C	:ase +o e .lement
0>10 1010) 0>2ac8) 0>11 1) 0>90) 0>F0) 0>00) 0>80) 0>cC) 0>00) 0>E9) 0>9C	6ro!/ +o e .lement
0>10 102a) 0>2ac8) 0>11 1) 0>90) 0>F0) 0>00) 0>80) 0>cC) 0>00) 0>E9) 0>9C	&nstance +o e .lement
0>10 102c) 0>2ac8) 0>11 1) 0>90) 0>F0) 0>00) 0>80) 0>cC) 0>00) 0>E9) 0>9C	", D +o e .lement
0>ceBEC2DE) 0>B8f0) 0>11 1) 0>aE) 0>F) 0>0) 0>F0) 0>9C) 0>0) 0>cF) 0>e1	$eta Data +o e .lement
0> 2B9eC0F) 0> CC) 0>D289) 0>a0) 0>C) 0>00) 0>ee) 0>C9) 0>fC) 0>9D) 0>9D	+4"" S(a/e +o e .lement
0>ceBEC2DD) 0>B8f0) 0>11 1) 0>aE) 0>F) 0>0) 0>F0) 0>9C) 0>0) 0>cF) 0>e1	Part +o e .lement
0>10 10Be) 0>2ac8) 0>11 1) 0>90) 0>F0) 0>00) 0>80) 0>cC) 0>00) 0>E9) 0>9C	Partition +o e .lement
0>10 10Dc) 0>2ac8) 0>11 1) 0>90) 0>F0) 0>00) 0>80) 0>cC) 0>00) 0>E9) 0>9C	Range ", D +o e .lement
0>10 10fB) 0>2ac8) 0>11 1) 0>90) 0>F0) 0>00) 0>80) 0>cC) 0>00) 0>E9) 0>9C	S%itc(+o e .lement
0>10 10E9) 0>2ac8) 0>11 1) 0>90) 0>F0) 0>00) 0>80) 0>cC) 0>00) 0>E9) 0>9C	:ase S(a/e +o e .lement
0>981BDC1F) 0>0010) 0>0818) 0>19) 0>98) 0>08) 0>00) 0>09) 0>8B) 0>E) 0>Ea	Point Set S(a/e +o e .lement
0>10 10D8) 0>2ac8) 0>11 1) 0>90) 0>F0) 0>00) 0>80) 0>cC) 0>00) 0>E9) 0>9C	Pol#gon Set S(a/e +o e .lement
0>10 10DF) 0>2ac8) 0>11 1) 0>90) 0>F0) 0>00) 0>80) 0>cC) 0>00) 0>E9) 0>9C	Pol#line Set S(a/e +o e .lement
0>eD0BCBc1) 0>1a 9) 0>11 B) 0>9) 0>af) 0>0) 0>a0) 0>c9) 0>cC) 0>) 0>c2	Primitive Set S(a/e +o e .lement
0>10 10CC) 0>2ac8) 0>11 1) 0>90) 0>F0) 0>00) 0>80) 0>cC) 0>00) 0>E9) 0>9C	Tri-Stri/ Set S(a/e +o e .lement
0>10 10Cf) 0>2ac8) 0>11 1) 0>90) 0>F0) 0>00) 0>80) 0>cC) 0>00) 0>E9) 0>9C	Verte> S(a/e +o e .lement
0>DccCaE21) 0>C28) 0>11 B) 0>9) 0>80) 0>0) 0>a0) 0>c9) 0>cC) 0>) 0>c2	=ire ;arness Set S(a/e +o e .lement
0>10 1001) 0>2ac8) 0>11 1) 0>90) 0>F0) 0>00) 0>80) 0>cC) 0>00) 0>E9) 0>9C	:ase 'ttri0!te .lement
0>10 101D) 0>2ac8) 0>11 1) 0>90) 0>F0) 0>00) 0>80) 0>cC) 0>00) 0>E9) 0>9C	Dra% St#le 'ttri0!te .lement
0>a 8 ccc2) 0>Ca80) 0>DEF) 0>00) 0> E) 0>) 0>Ba) 0>0) 0>8) 0>21) 0>eC	Fragment S(a er 'ttri0!te .lement
0>10 108B) 0>2ac8) 0>11 1) 0>90) 0>F0) 0>00) 0>80) 0>cC) 0>00) 0>E9) 0>9C	6eometric Transform 'ttri0!te .lement
0>10 1028) 0>2ac8) 0>11 1) 0>90) 0>F0) 0>00) 0>80) 0>cC) 0>00) 0>E9) 0>9C	&nfinite "ig(t 'ttri0!te .lement
0>10 109F) 0>2ac8) 0>11 1) 0>90) 0>F0) 0>00) 0>80) 0>cC) 0>00) 0>E9) 0>9C	"ig(t Set 'ttri0!te .lement
0>10 10cD) 0>2ac8) 0>11 1) 0>90) 0>F0) 0>00) 0>80) 0>cC) 0>00) 0>E9) 0>9C	"inest#le 'ttri0!te .lement
0>10 10B0) 0>2ac8) 0>11 1) 0>90) 0>F0) 0>00) 0>80) 0>cC) 0>00) 0>E9) 0>9C	$aterial 'ttri0!te .lement
0>10 10DE) 0>2ac8) 0>11 1) 0>90) 0>F0) 0>00) 0>80) 0>cC) 0>00) 0>E9) 0>9C	Point "ig(t 'ttri0!te .lement
0>8 ECc010) 0>eEc0) 0>11 D) 0>8D) 0>e) 0>00) 0>a0) 0> 2) 0>18) 0>2f) 0>9	Pointst#le 'ttri0!te .lement

GUID	42?e#t Ty$e
0>aa108B1) 0>FeDC) 0>Dfee) 0>a8) 0>FE) 0>c) 0>Ce) 0>1f) 0>2f) 0>B9) [illegible]	[illegible]

+ en*i: 32 Semantic Value Class S%a*er Parameter Values

C.2.1.1.2.12 Verte> S(a er 'ttri0!te .lement an C.2.1.1.2.1B Fragment S(a er 'ttri0!te .lement contain s(a er /arameters. T(ese s(a er /arameters can 0e of a ISemanticK Val!e -lass %(ic(in icates t(at t(e s(a er /arameter is im/licitl# tie 50o!n to a /iece of eit(er , /en6" or JT gra/(ics s#stem state. Ta0le 11 0elo% oc!ments all t(e /ossi0le ISemanticK Val!e -lass s(a er /arameter Val!es 1i.e. t(e gra/(ics s#stem state t(e /arameter is 0o!n to2.

Ta2"e 115 Semanti# :a"&e C"ass Shader 3arameter :a"&es

Val!e	Descri/tion of Semanticall# :o!n 6ra/(ics State	+otes
[0	– 4n3no%n	
Re"ated to C&rrent 4$en1, State		
[B0	– Vie% Transform $atri>	-g onl#
[B1	– -om0ine $o el-Vie% Transform $atri>	-g onl#
[B2	– Pro7ection Transform $atri>	-g onl#
[BB	– Te>t!re Transform $atri>	-g onl#
[BD	– -om0ine $o el-Vie%-Pro7ection Transform $atri>	-g onl#
[BE	– Vie% $atri> Trans/ose	-g onl#
[BF	– -om0ine $o el-Vie% Transform $atri> Trans/ose	-g onl#
[BC	– Pro7ection Transform $atri> Trans/ose	-g onl#
[B8	– Te>t!re Transform $atri> Trans/ose	-g onl#
[B9	– -om0ine $o el-Vie%-Pro7ection Transform $atri> Trans/ose	-g onl#
[D0	– Vie% Transform $atri> &nverse	-g onl#
[D1	– -om0ine $o el-Vie% Transform $atri> &nverse	-g onl#
[D2	– Pro7ection Transform $atri> &nverse	-g onl#
[DB	– Te>t!re Transform $atri> &nverse	-g onl#
[DD	– -om0ine $o el-Vie%-Pro7ection Transform $atri> &nverse	-g onl#
[DE	– Vie% Transform $atri> &nverse Trans/ose	-g onl#
[DF	– -om0ine $o el-Vie% Transform $atri> &nverse Trans/ose	-g onl#
[DC	– Pro7ection Transform $atri> &nverse Trans/ose	-g onl#
[D8	– Te>t!re Transform $atri> &nverse Trans/ose	-g onl#
[D9	– -om0ine $o el-Vie%-Pro7ection Transform $atri> &nverse Trans/ose	-g onl#
Re"ated to C&rrent FT State		
[C0	– -!rrent $o el Transform	
[C1	– -!rrent $o el Transform Trans/ose	
[C2	– -!rrent $o el Transform &nverse	
[CB	– -!rrent $o el Transform &nverse Trans/ose	
[CE	– -!rrent $aterial .missive -olor	
[CF	– -!rrent $aterial Diff!se -olor	
[CC	– -!rrent $aterial S/ec!lar -olor	
[C8	– -!rrent $aterial 'm0ient -olor	
[C9	– -!rrent $aterial S(ininess	

Val!e	Descri/tion of Semanticall# :o!n 6ra/(ics State	+otes
[80	– -!rrent Fog -olor	
[81	– Se/arate S/ec!lar -olor Flag	
[82	– 6lo0al 'm0ient "ig(t -olor	
[99	– +!m0er of VP-S "ig(ts	
[100	– VP-S "ig(t-0 Diff!se -olor	
[101	– VP-S "ig(t-0 S/ec!lar -olor	
[102	– VP-S "ig(t-0 'm0ient -olor	
[10B	– VP-S "ig(t-0 'tten!ation	
[10D	– VP-S "ig(t-0 Position	
[10E	– VP-S "ig(t-0 Direction	
[10F	– VP-S "ig(t-0 S/ot Direction	
[10C	– VP-S "ig(t-0 S/ot -one 'ngle	
[108	– VP-S "ig(t-0 -osine of S/ot -one 'ngle	
[109	– VP-S "ig(t-0 S/ot .>/onent	
[110	– VP-S "ig(t-0 S(a o% ,/acit#	
[120 g 1B0	– Same as val!es 100 g 110 e>ce/t for VP-S "ig(t-1	
[1D0 g 1E0	– Same as val!es 100 g 110 e>ce/t for VP-S "ig(t-2	
[1F0 g 1C0	– Same as val!es 100 g 110 e>ce/t for VP-S "ig(t-B	
[180 g 190	– Same as val!es 100 g 110 e>ce/t for VP-S "ig(t-D	
[200 g 210	– Same as val!es 100 g 110 e>ce/t for VP-S "ig(t-E	
[220 g 2B0	– Same as val!es 100 g 110 e>ce/t for VP-S "ig(t-F	
[2D0 g 2E0	– Same as val!es 100 g 110 e>ce/t for VP-S "ig(t-C	
[D99	– +!m0er of $-S "ig(ts	
[E00 g E10	– Same as val!es 100 g 110 e>ce/t for $-S "ig(t-0	
[E20 g EB0	– Same as val!es 100 g 110 e>ce/t for $-S "ig(t-1	
[ED0 g EE0	– Same as val!es 100 g 110 e>ce/t for $-S "ig(t-2	
[EF0 g EC0	– Same as val!es 100 g 110 e>ce/t for $-S "ig(t-B	
[E80 g E90	– Same as val!es 100 g 110 e>ce/t for $-S "ig(t-D	
[F00 g F10	– Same as val!es 100 g 110 e>ce/t for $-S "ig(t-E	
[F20 g FB0	– Same as val!es 100 g 110 e>ce/t for $-S "ig(t-F	
[FD0 g FE0	– Same as val!es 100 g 110 e>ce/t for $-S "ig(t-C	
[899	– +!m0er of =-S "ig(ts	
[900 g 910	– Same as val!es 100 g 110 e>ce/t for =-S "ig(t-0	
[920 g 9B0	– Same as val!es 100 g 110 e>ce/t for =-S "ig(t-1	
[9D0 g 9E0	– Same as val!es 100 g 110 e>ce/t for =-S "ig(t-2	
[9F0 g 9C0	– Same as val!es 100 g 110 e>ce/t for =-S "ig(t-B	
[980 g 990	– Same as val!es 100 g 110 e>ce/t for =-S "ig(t-D	
[1000 g 1010	– Same as val!es 100 g 110 e>ce/t for =-S "ig(t-E	
[1020 g 10B0	– Same as val!es 100 g 110 e>ce/t for =-S "ig(t-F	
[10D0 g 10E0	– Same as val!es 100 g 110 e>ce/t for =-S "ig(t-C	

Val!e	Descri/tion of Semanticall# :o!n 6ra/(ics State	+otes
[1E00	– -!rrent Te>t!re ,07ect-0	-g onl#
[1E01	– -!rrent Te>t!re ,07ect-1	-g onl#
[1E02	– -!rrent Te>t!re ,07ect-2	-g onl#
[1E0B	– -!rrent Te>t!re ,07ect-B	-g onl#
[1E0D	– -!rrent Te>t!re ,07ect-D	-g onl#
[1E0E	– -!rrent Te>t!re ,07ect-E	-g onl#
[1E0F	– -!rrent Te>t!re ,07ect-F	-g onl#
[1E0C	– -!rrent Te>t!re ,07ect-C	-g onl#
[1F00	– -!rrent Te>t!re 4nit-0	6"S" onl#
[1F01	– -!rrent Te>t!re 4nit-1	6"S" onl#
[1F02	– -!rrent Te>t!re 4nit-2	6"S" onl#
[1F0B	– -!rrent Te>t!re 4nit-B	6"S" onl#
[1F0D	– -!rrent Te>t!re 4nit-D	6"S" onl#
[1F0E	– -!rrent Te>t!re 4nit-E	6"S" onl#
[1F0F	– -!rrent Te>t!re 4nit-F	6"S" onl#
[1F0C	– -!rrent Te>t!re 4nit-C	6"S" onl#
[1C00	– Te>t!re -(annel-0 V-S Te>t!re -oor inate 6eneration S-Plane	
[1C01	– Te>t!re -(annel-0 V-S Te>t!re -oor inate 6eneration T-Plane	
[1C02	– Te>t!re -(annel-0 V-S Te>t!re -oor inate 6eneration R-Plane	
[1C0B	– Te>t!re -(annel-0 V-S Te>t!re -oor inate 6eneration G-Plane	
[1C10 g 1C1B	– Same as 1C00 g 1C0B e>ce/t for -(anel-1 V-S	
[1C20 g 1C2B	– Same as 1C00 g 1C0B e>ce/t for -(anel-2 V-S	
[1CB0 g 1CBB	– Same as 1C00 g 1C0B e>ce/t for -(anel-B V-S	
[1CD0 g 1CDB	– Same as 1C00 g 1C0B e>ce/t for -(anel-D V-S	
[1CE0 g 1CEB	– Same as 1C00 g 1C0B e>ce/t for -(anel-E V-S	
[1CF0 g 1CFB	– Same as 1C00 g 1C0B e>ce/t for -(anel-F V-S	
[1CC0 g 1CCB	– Same as 1C00 g 1C0B e>ce/t for -(anel-C V-S	
[2000 g 200B	– Same as 1C00 g 1C0B e>ce/t for -(anel-0 $-S	
[2010 g 201B	– Same as 1C00 g 1C0B e>ce/t for -(anel-1 $-S	
[2020 g 202B	– Same as 1C00 g 1C0B e>ce/t for -(anel-2 $-S	
[20B0 g 20BB	– Same as 1C00 g 1C0B e>ce/t for -(anel-B $-S	
[20D0 g 20DB	– Same as 1C00 g 1C0B e>ce/t for -(anel-D $-S	
[20E0 g 20EB	– Same as 1C00 g 1C0B e>ce/t for -(anel-E $-S	
[20F0 g 20FB	– Same as 1C00 g 1C0B e>ce/t for -(anel-F $-S	
[20C0 g 20CB	– Same as 1C00 g 1C0B e>ce/t for -(anel-C $-S	
[B000	– Te>t!re -(annel-0 $atri>	
[B001	– Te>t!re -(annel-1 $atri>	
[B002	– Te>t!re -(annel-2 $atri>	

Val!e	Descri/tion of Semanticall# :o!n 6ra/(ics State	+otes
[B00B	– Te>t!re -(annel-B $atri>	
[B00D	– Te>t!re -(annel-D $atri>	
[B00E	– Te>t!re -(annel-E $atri>	
[B00F	– Te>t!re -(annel-F $atri>	
[B00C	– Te>t!re -(annel-C $atri>	
[B100	– Te>t!re -(annel-0 $a/ Resol!tion	
[B101	– Te>t!re -(annel-1 $a/ Resol!tion	
[B102	– Te>t!re -(annel-2 $a/ Resol!tion	
[B10B	– Te>t!re -(annel-B $a/ Resol!tion	
[B10D	– Te>t!re -(annel-D $a/ Resol!tion	
[B10E	– Te>t!re -(annel-E $a/ Resol!tion	
[B10F	– Te>t!re -(annel-F $a/ Resol!tion	
[B10C	– Te>t!re -(annel-C $a/ Resol!tion	
[B200	– Te>t!re -(annel-0 $a/ Resol!tion &nverses 1i.e. 1.0 5K $a/ Resol!tionK2	
[B201	– Te>t!re -(annel-1 $a/ Resol!tion &nverses	
[B202	– Te>t!re -(annel-2 $a/ Resol!tion &nverses	
[B20B	– Te>t!re -(annel-B $a/ Resol!tion &nverses	
[B20D	– Te>t!re -(annel-D $a/ Resol!tion &nverses	
[B20E	– Te>t!re -(annel-E $a/ Resol!tion &nverses	
[B20F	– Te>t!re -(annel-F $a/ Resol!tion &nverses	
[B20C	– Te>t!re -(annel-C $a/ Resol!tion &nverses	
[BB00	– Te>t!re -(annel-0 :len -olor	
[BB01	– Te>t!re -(annel-1 :len -olor	
[BB02	– Te>t!re -(annel-2 :len -olor	
[BB0B	– Te>t!re -(annel-B :len -olor	
[BB0D	– Te>t!re -(annel-D :len -olor	
[BB0E	– Te>t!re -(annel-E :len -olor	
[BB0F	– Te>t!re -(annel-F :len -olor	
[BB0C	– Te>t!re -(annel-C :len -olor	

+ en*i: C2 Deco*ing +lgorit%ms H +n Im lementation

T(is ' //en i> /rovi es a sam/le - \\ im/lementation for t(e eco ing /ortion of t(e vario!s com/ression - , D. -s 1as etaile in 8.2 .nco ing ' lgorit(ms2 !se in t(e JT format. T(is sam/le co e is not inten e to 0e f!ll# f!nctional eco er class im/lementations) 0!t is instea inten e to emonstrate t(e f!n amentals of im/lementing t(e eco ing /ortion of t(e - , D. - algorit(ms !se in t(e JT format.

C.1 Common classes

T(e follo%ing s!0-sections efine some general classes !se 0# all t(e eco ing algorit(ms.

C.1.1 Cnt:6ntr! class

```
//
// Type used to build probability context tables.
// Used by ProbabilityContext class.
//
class CntxEntry
{
public:

   Int32 iSym;          // Symbol
   Int32 cCount;        // Number of occurrences
   Int32 cCumCount;     // Cumulative number of occurrences
   Int32 iNextCntx = 0; // Next context if this symbol seen
};
```

C.1.# Probabilit!Conte:t class

```
//
// Type used to build probability context tables.
// Used by CodecDriver class.
//
class ProbabilityContext
{
public:

   // Returns total cumulative count for all context entries
   Int32 totalCount();

   // Returns number of context entries
   Int32 numEntries();

   // Returns context entry of index iEntry
   Bool getEntry(Int32 iEntry, CntxEntry& rpEntry);

   // Looks up the next context field given by the context entry
   // with input symbol 'iSymbol'
```

```
   Bool lookupNextContext(Int32 iSymbol, Int32& iNextContext);

   // Looks up the index of the context entry for the given
   // input symbol 'iSymbol'
   Bool lookupSymbol(Int32 iSymbol, Int32& iOutEntry);

   // Looks up the index of the context entry that falls just above
   // the accumulated count.
   Bool lookupEntryByCumCount(Int32 iCount, Int32& iOutEntry);
};
```

C.1.) Co*ecDriver class

```
//
// A class that deals with the conversions from SYMBOL to VALUE and
// provides end-consumer APIs for using the codecs.
//
class CodecDriver
{
public:
   // ---------- Codec Decoding Interface ----------
   // Returns the number of symbols to be read
   Int32 numSymbolsToRead();

   // Returns index of the first context entry and total number of bits
   Bool getDecodeData(Int32& iFirstContext, Int32& nTotalBits);

   // Returns the next 32 bits of CodeText
   Bool getNextCodeText(UInt32& uCodeText, Int32& nBits);

   // Adds the decoded symbol back to the driver
   Bool addOutputSymbol(Int32 iSymbol, Int32& iNextContext) ;

   // ---------- Symbol Probability Context Interface ----------
   Bool clearAllContexts();

   // Retrieves a new probability context
   Bool getNewContext(ProbabilityContext& rpCntx);

   // Returns the total number of contexts
   Int32 numContexts();

   // Returns the probability context for a given index
   Bool getContext(Int32 iSymContext, ProbabilityContext& rpCntx);

   // ---------- Predictor Type Residual Unpacking ----------

   typedef enum
   {
      PredLag1       = 0,
```

```
      PredLag2       = 1,
      PredStride1    = 2,
      PredStride2    = 3,
      PredStripIndex = 4,
      PredRamp       = 5,
      PredXor1       = 6,
      PredXor2       = 7,
      PredNULL       = 8
   } PredictorType;

   // Returns the original values from the predicted residual values.
   static Bool unpackResiduals(Vector<Int32>& rvResidual,
                               Vector<Int32>& rvVals,
                               PredictorType  ePredType);

   static Bool unpackResiduals(Vector<Float64>& rvResidual,
                               Vector<Float64>& rvVals,
                               PredictorType  ePredType);

   // Predict values
   static Int32 predictValue(Vector<Int32>& vVal,
                             Int32 iIndex,
                             PredictorType ePredType);

   static Float64 predictValue(Vector<Float64>& vVal,
                               Int32 iIndex,
                               PredictorType ePredType);
}

Bool CodecDriver::unpackResiduals(Vector<Int32>& rvResidual,
                                  Vector<Int32>& rvVals,
                                  PredictorType  ePredType)
{
   Int32 iPredicted;

   Int32 len = rvResidual.length();
   rvVals.setLength(len);
   Int32* aVals = (Int32 *) rvVals;
   Int32* aResidual = (Int32 *) rvResidual;

   for( Int32 i = 0; i < len; i++ )
   {
      if( i < 4 )
      {
         // The first four values are just primers
         aVals[i] = aResidual[i];
      }
      else
      {
         // Get a predicted value
         iPredicted = predictValue(rvVals, i, ePredType);
```

```
         if( ePredType == PredXor1 || ePredType == PredXor2 )
         {
            // Decode the residual as the current value XOR predicted
            aVals[i] = aResidual[i] ^ iPredicted;
         }
         else
         {
            // Decode the residual as the current value plus predicted
            aVals[i] = aResidual[i] + iPredicted;
         }
      }
   }

   return True;
}

Bool CodecDriver::unpackResiduals(Vector<Float64>& rvResidual,
                                  Vector<Float64>& rvVals,
                                  PredictorType  ePredType)
{
   if( ePredType == PredXor1 || ePredType == PredXor2 )
      return False;

   if( ePredType == PredNULL )
   {
      rvVals = rvResidual;
      return True;
   }

   Float64 iPredicted;
   Int32 len = rvResidual.length();
   rvVals.setLength(len);

   for( Int32 i = 0; i < len; i++ )
   {
      if( i < 4 )
      {
         // The first four values are just primers
         rvVals[i] = rvResidual[i];
      }
      else
      {
         // Get a predicted value
         iPredicted = predictValue(rvVals, i, ePredType);

         // Decode the value as the residual plus predicted
         rvVals[i]   = rvResidual[i] + iPredicted;
      }
   }
```

```
   return True;
}

Int32 CodecDriver::predictValue(Vector<Int32>& vVal,
                                Int32 iIndex,
                                PredictorType ePredType)
{
   Int32* aVals = (Int32 *) rvVals;
   JtInt32 iPredicted,
           v1 = aVals[iIndex-1],
           v2 = aVals[iIndex-2],
           v3 = aVals[iIndex-3],
           v4 = aVals[iIndex-4];

   switch( ePredType )
   {
      default:
      case PredLag1:
      case PredXor1:
         iPredicted = v1;
         break;

      case PredLag2:
      case PredXor2:
         iPredicted = v2;
         break;

      case PredStride1:
         iPredicted = v1 + (v1 - v2);
         break;

      case PredStride2:
         iPredicted = v2 + (v2 - v4);
         break;

      case PredStripIndex:
         if( v2 - v4 < 8 && v2 - v4 > -8 )
            iPredicted = v2 + (v2 - v4);
         else
            iPredicted = v2 + 2;
         break;

      case PredRamp:
         iPredicted = iIndex;
         break;
   }

   return iPredicted;
}
```

```
Float64 CodecDriverBase::predictValue(Vector<Float64>& vVal,
                                      Int32 iIndex,
                                      PredictorType ePredType)
{
   Float64* aVals = (Float64 *) rvVals;
   Float64 iPredicted,
           v1 = aVals[iIndex-1],
           v2 = aVals[iIndex-2],
           v3 = aVals[iIndex-3],
           v4 = aVals[iIndex-4];

   switch( ePredType )
   {
      default:
      case PredLag1:
         iPredicted = v1;
         break;

      case PredLag2:
         iPredicted = v2;
         break;

      case PredStride1:
         iPredicted = v1 + (v1 - v2);
         break;

      case PredStride2:
         iPredicted = v2 + (v2 - v4);
         break;

      case PredStripIndex:
         if( v2 - v4 < 8 && v2 - v4 > -8 )
            iPredicted = v2 + (v2 - v4);
         else
            iPredicted = v2 + 2;
         break;

      case PredRamp:
         iPredicted = iIndex;
         break;
   }

   return iPredicted;
}
```

C.# **3**itlengt% ***

```
{

   // Slice off as many bits as we can all at once.
   Int32 m = min(n - iBit, cNumCurBits - nAccBits);
   if( m < 32 )
   {
      uAccVal <<= m;
      uAccVal |= ((uVal >> (32 - m)) & ((1 << m) - 1));
      nAccBits += m;
      iBit += m - 1;

      // Advance the bit-marching counters
      uVal <<= m;
      nBits += m;
      nValBits -= m;
   }
   else
   {
      uAccVal = uVal;
      nAccBits += m;
      iBit += m - 1;

      // Advance the bit-marching counters
      uVal = 0;
      nBits += m;
      nValBits -= m;
   }

   if( nAccBits >= cNumCurBits )
   {
      // Convert and sign-extend the symbol
      iSymbol = Int32(uAccVal);
      iSymbol <<= (32 - cNumCurBits);
      iSymbol >>= (32 - cNumCurBits);

      // Output the symbol and restart
      pDriver->addOutputSymbol(iSymbol, iContext);
      iDecodeState = 0;
      uAccVal  = 0;
      nAccBits = 0;
   }
}
else
{
   // All other decode states are handled one bit at a time
   // inside this block.
   // Get the next bit
   uAccVal = (uVal >> 31);

   switch( iDecodeState )
   {
```

```
            // DecodeState = 0: Recognize prefix bit (0=Same size
            // code, 1=Different size code).
            case 0:
               // Recognize "same length" prefix code
               if( uAccVal == 0 )
                  iDecodeState = 2;
               else
               {
                  // Recognize "different length" prefix code
                  iDecodeState = 1;
                  uLastIncBit = 2;
               }

               uAccVal  = 0;
               break;

            case 1:  // Length adjustment mode
               // Recognize the terminator bit
               if( uLastIncBit != 2 && (uAccVal ^ uLastIncBit) )
               {
                  iDecodeState = 2;
                  uLastIncBit = 2;
               }
               else
               {
                  // Recognize the "increment" prefix code
                  if( uAccVal == 1 )
                  {
                     cNumCurBits += cStepBits;
                  }
                  else
                  {
                     // Recognize the "decrement" prefix code
                     cNumCurBits -= cStepBits;
                  }

                  uLastIncBit = uAccVal;
               }

               uAccVal  = 0;
               break;
         }

         // Advance the bit-marching counters that keep track of the
         // "current codetext bit", and how many bits are left.
         uVal <<= 1;
         nBits++;
         nValBits--;
      }
   }
```

```
   }

   // If the last symbol was zero and the current bit length
   // is also zero, then the above loop terminated before
   // actually decoding the last zero-valued symbol.  Test
   // for that condition here and decode it if necessary.
   if( iDecodeState == 2 && cNumCurBits == 0 )
   {
      // Output the symbol and restart
      iSymbol = Int32(0);
      pDriver->addOutputSymbol(iSymbol, iContext);
   }

   return True;
}
```

C.) 0uffman *eco*ing classes

T(e follo%ing s!0-sections contain a sam/le im/lementation of t(e eco ing /ortion of t(e ;!ffman -,D.- algorit(m. ' s!mmar# tec(nical e>/lnation of t(e ;!ffman -,D.- can 0e fo!n in 8.2.B ;!ffman -,D.-.

C.).1 0uffCo*eData class

;!ff-o eData is a (el/er class for 3ee/ing trac3 of a given s#m0ol an t(e 0its !se to escri0e it.

```
class HuffCodeData
{
public:
   HuffCodeData() :
      iSymbol(0), iBitCode(0), nCodeLen(0)
   {
   }

   HuffCodeData(Int32&  symbol,
                UInt32& bitCode,
                Int32&  codeLen) :
      iSymbol(symbol), iBitCode(bitCode), nCodeLen(codeLen)
   {
   }

   HuffCodeData(Int32& symbol) :
      iSymbol(symbol), iBitCode(0), nCodeLen(0)
   {
   }

   Bool operator < (HuffCodeData& rhs)
   {
      if( this->iSymbol < rhs.iSymbol )
         return True;
      else
         return False;
   }

   Bool operator == (HuffCodeData& rhs)
   {
      if( this->iSymbol == rhs.iSymbol )
         return True;
      else
         return False;
   }

   Int32  iSymbol;
   Int32  nCodeLen;
   UInt32 iBitCode;
};
```

C.).# 0uffTree'o*e class

;!ffTree+o e is a (el/er class !se in t(e constr!ction of t(e ;!ffman tree. &t contains t(e s#m0ol) its freA!enc#) t(e ;!ffman co e an its lengt() an /ointers to t(e UleftN an Urig(tN no es.

```
class HuffTreeNode
{
public:
   HuffTreeNode() :
      cSymcounts(0)
   {
   }

   Bool operator < (HuffTreeNode& rhs)
   {
      if( this->cSymCounts < rhs.cSymCounts )
         return True;
      else
         return False;
   }

   Int32        cSymCounts;
   HuffTreeNode* pLeft;
   HuffTreeNode* pRight;
   HuffCodeData sData;
};
```

C.).) 0uffCo*ecConte:t class

;!ff-o ec-onte>t is a class t(at efines t(e ;!ffman conte>t

```
class HuffCodecContext
{
public:
   HuffCodecContext() :
      cLength(0), nCodeLength(0), uCode(0)
   {
   }

   // Used when constructing the Huffman code
   Int32  cLength; // Length of Huffman code currently
                   // under construction.
   UInt32 uCode;   // Code under construction

   // Used to store the final Huffman code table
   OrderedVector<HuffCodeData> vCodes; // Ordered by symbol number

   // Used during encoding
   Int32 nCodeLength; // Used to tally up total encoded code length
};
```

C.).,0uffmanCo*ec class

;!ffman-o ec is t(e class t(at eco es ;!ffman enco e ata.

```
class HuffmanCodec
{
public:
   // Decodes the Huffman codetext present in the vInCode entries to
   // a list of symbols, placing the symbols onto the driver object.
   // This method must contruct a Huffman tree from the symbol
   // statistics present on driver object.
   Bool decode(CodecDriver* pDriver);

private:
   // Build Huffman tree for each probability context
   Bool buildHuffmanForest(CodecDriver* pDriver);

   // Build Huffman tree from symbol statistics
   Bool buildHuffmanTree(ProbabilityContext* pCntx,
                         HuffTreeNode* pRootNode);

   // Assign Huffman bit-codes to leaves of tree
   Bool assignCodeToTree(HuffTreeNode* pRoot,
                         HuffCodecContext& rCntxt);

   // Convert codetext vector to symbol vector
   Bool codetextToSymbols(CodecDriver* pDriver);

   Vector<HuffTreeNode*> vpHuffTrees; // Indexed by context number
   Vector<HuffCodecContext> vHuffCntx; // HuffmanCodecContexts
};

Bool HuffmanCodec::decode(CodecDriver* pDriver)
{
   // Build a Huffman tree for each probability context
   buildHuffmanForest(pDriver);

   // Convert codetext to symbols
   codetextToSymbols(pDriver);

   return True;
}

Bool HuffmanCodec::buildHuffmanForest(CodecDriver* pDriver)
{
   HuffTreeNode* pRoot = NULL;
   Int32 nCntx = pDriver->numContexts();
   Int32 i;
   for( i = 0; i < nCntx; i++ )
   {
      // Get the i'th context
```

```
      ProbabilityContext* pCntx = NULL;
      pDriver->getContext(i, pCntx);

      // Create Huffman tree from probability context
      buildHuffmanTree(pCntx, pRoot);

      // Assign Huffman codes
      assignCodeToTree(pRoot, vHuffCntx[i]);

      // Store the completed Huffman tree
      vpHuffTrees[i] = pRoot;
   }

   return True;
}

Bool HuffmanCodec::buildHuffmanTree(ProbabilityContext* pCntx,
                                    HuffTreeNode* pRootNode)
{
   HeapVector<HuffTreeNode*> heap;
   HuffTreeNode* pNode = NULL;

   // Initialize all the nodes and add them to the heap.
   Int32 nEntries = pCntx->numEntries();
   for( Int32 i = 0; i < nEntries; i++ )
   {
      CntxEntry* pEntry = NULL;
      pCntx->getEntry(i, pEntry);
      pNode->sData.iSymbol = pEntry->iSym;
      pNode->cSymCounts    = pEntry->cCount;
      pNode->pLeft = NULL;
      pNode->pRight = NULL;
      heap.add(pNode);
   }

   HuffTreeNode* pNewNode1 = NULL;
   HuffTreeNode* pNewNode2 = NULL;

   while( heap.length() > 1 )
   {
      // Get the two lowest-frequency nodes.
      heap.getMin(pNewNode1);
      heap.getMin(pNewNode2);

      //Combine the low-freq nodes into one node.
      pNode->sData.iSymbol = 0xdeadbeef;
      pNode->pLeft         = pNewNode1;
      pNode->pRight        = pNewNode2;
      pNode.cSymCounts = pNewNode1->cSymCounts + pNewNode2->cSymCounts;

      //Add the new node to the heap.
```

```
      heap.add(pNode);
   }

   // Set the root node.
   heap.getMin(pNode);
   pRootNode = pNode;

   return True;
}

Bool HuffmanCodec::assignCodeToTree(HuffTreeNode* pNode,
                                    HuffCodecContext& rCntxt)
{
   if( pRoot->pLeft != 0 )
   {
      rCntxt.uCode <<= 1;
      rCntxt.uCode |=  1;
      rCntxt.cLength++;
      assignCodeToTree(pRoot->pLeft, rCntxt);
      rCntxt.cLength--;
      rCntxt.uCode >>= 1;
   }

   if( pRoot->pRight != 0 )
   {
      rCntxt.uCode <<= 1;
      rCntxt.cLength++;
      assignCodeToTree(pRootpRight, rCntxt);
      rCntxt.cLength--;
      rCntxt.uCode >>= 1;
   }

   if( pRoot->pRight != 0 )
      return True;

   // Set the code and its length for the node.
   pRoot->sData.iBitCode = rCntxt.uCode;
   pRoot->sData.nCodeLen = rCntxt.cLength;

   // Setup the internal symbol look-up table.
   Int32 null = 0;
   rCntxt.vCodes.insert(HuffCodeData(pRoot->sData.iSymbol,
                                     pRoot->sData.iBitCode,
                                     pRoot->sData.nCodeLen), null);

   return True;
}

Bool HuffmanCodec::codetextToSymbols(CodecDriver* pDriver)
{
   HuffTreeNode* pHNode = NULL;
```

```
   UInt32 mask = 1 << 31;
   Int32 j,
         nBits = 0,
         nTotalBits = 0,
         nValBits = 0,
         iContext = 0;
   UInt32 uVal;

   pDriver->getDecodeData(iContext, nTotalBits);
   pHNode = vpHuffTrees[iContext];

   while( nBits < nTotalBits )
   {
      // Get the next 32 bits from the input stream
      pDriver->getNextCodeText(uVal, nValBits);

      // Scan through each bit either walking the Huffman code
      // tree or accumulating escaped bit values.
      for( j = 0; j < 32 && nBits < nTotalBits && nValBits > 0; j++ )
      {
         // March to the next node
         pHNode = (uVal & mask) ? pHNode->pLeft : pHNode->pRight;

         // If the node is a leaf, output a symbol and restart
         if( pHNode->pLeft == 0 && pHNode->pRight == 0 )
         {
            pDriver->addOutputSymbol(pHNode->sData.iSymbol, iContext);
            pHNode = vpHuffTrees[iContext];
         }

         uVal <<= 1;
         nBits++;
         nValBits--;
      }
   }

   return True;
}
```

C.2 Arithmetic encoding classes

The following sections contain a sample implementation of the encoding portion of the arithmetic algorithm. A summary technical explanation of the arithmetic can be found in 8.2.2 Arithmetic encoding.

C.2.1 ArithmeticProbabilityRange class

```
class ArithmeticProbabilityRange
{
public:
   UInt16 low_count;
   UInt16 high_count;
   UInt16 scale;
}
```

C.2.2 ArithmeticCodec class

Arithmetic codec class is the class that encodes arithmetic encoded data.

```
class ArithmeticCodec
{
public:
   ArithmeticCodec() :
      code = 0x0000,
      low = 0x0000,
      high = 0xffff,
      nUnderflowBits = 0,
      bitBuffer =0x00000000,
      nBits = 0
   {
   }

   // Decodes a list of symbols. The codecDriver provides the range
   // info for the symbols to decode.  It also stores the symbols as
   // they are decoded.
   Bool decode(CodecDriver* pDriver);

private:
   // Remove the most recently decoded symbol from the front of the
   // list of encoded symbols.
   Bool removeSymbolFromStream(ArithmeticProbabilityRange& sym,
                               CodecDriver* pDriver);

   //State variables used in decoding.
   UInt16 code;      // Present input code value, for decoding only
   UInt16 low;       // Start of the current code range
   UInt16 high;      // End of the current code range

   UInt32 bitBuffer; // Temporary i/o buffer
   Int32  nBits;     // Number of bits in _bitBuffer
};
```

```
Bool ArithmeticCodec::decode(CodecDriver* pDriver )
{
   ArithmeticProbabilityRange newSymbolRange;
   Int32 iCurrContext, nDummyTotalBits, cSymbolsCurrCtx, iCurrEntry;

   Int32 nSymbols = pDriver->numSymbolsToRead();

   ProbabilityContext* pCurrContext = NULL;
   CntxEntry* pCntxEntry = NULL;

   // Initialize decoding process
   Int32 nBitsRead = -1;
   pDriver->getNextCodeText(bitBuffer, nBitsRead);

   low  = 0;
   high = 0xffff;
   code = (bitBuffer >> 16);

   bitBuffer <<= 16;
   nBits = 16;

   // Begin decoding
   pDriver->getDecodeData(iCurrContext, nDummyTotalBits);
   for( Int32 ii = 0; ii < nSymbols; ii++ )
   {
      pDriver->getContext(iCurrContext, pCurrContext);

      cSymbolsCurrCtx = pCurrContext->totalCount();
      UInt16 rescaledCode =
         ((((UInt32)(code - low) + 1) * (UInt32) cSymbolsCurrCtx - 1) /
            ((UInt32)(high - low) + 1));

      pCurrContext->lookupEntryByCumCount((Int32)rescaledCode,
                                          iCurrEntry);

      pCurrContext->getEntry(iCurrEntry, pCntxEntry);

      newSymbolRange.high_count = pCntxEntry->cCumCount +
                                     pCntxEntry.cCount;
      newSymbolRange.low_count  = pCntxEntry->cCumCount;
      newSymbolRange.scale      = cSymbolsCurrCtx;

      removeSymbolFromStream(newSymbolRange, pDriver);

      pDriver->addOutputSymbol(pCntxEntry);

      iCurrContext = pCntxEntry->iNextCntx;
   }

   return True;
```