[JPEG压缩数据格式](http://blog.chinaunix.net/uid-12567959-id-165693.html)2011-03-11 10:46:23

**Annex B**

**Compressed data formats**

(This annex forms an integral part of this Recommendation | International Standard)

T本附录介绍三种压缩数据格式：

a) 交换格式，在B.2和B.3中介绍；

b) 压缩图像数据的缩略格式，在B.4中介绍；

c) 表规范数据的缩略格式，在B.5中介绍。

B.1 描述这些格式的构成部分。B.1.3和B.1.4给出了格式规范中使用到的插图和符号的约定。

**B.1 压缩数据规范的基本层面**

在结构上，压缩数据格式包含一个由参数、标记和熵编码数据段组成的有序集合。参数和标记往往又形成标记段。由于所有这些组成部分是由字节对齐的代码表示的，因此每一个压缩数据格式由一个8-bits字节的有序序列组成。对于每一个字节，都定义了一个最高有效位(MSB)和一个最低有效位(LSB)。

**B.1.1 组成部分**

这个小节给出了压缩数据格式每一个部分的概述。

**B.1.1.1 参数**

参数是整数，其值特定于具体的编码过程，源图像的特点，和应用程序可选的其他特性。参数由4-bit，1字节或2字节的代码赋值。除了某些可选的参数组，参数都编码重要的信息，如果没有这些信息，解码过程就无法正确的重建图像。

一个参数的代码应该是一个无符号整数，该整数具有以位为单位的规定长度，并具有特定的参数值。

对于长度2字节(16位)的参数，则在压缩数据字节的有序序列中，最高位字节应该在前。长度为4位的参数成对出现，并且这一对参数应被编码为一个单独的字节。参数对中的第一个4-bit参数应该占据字节的高4位。对于任何的16-，8-，或4-bit参数，MSB应该在前，LSB在后。

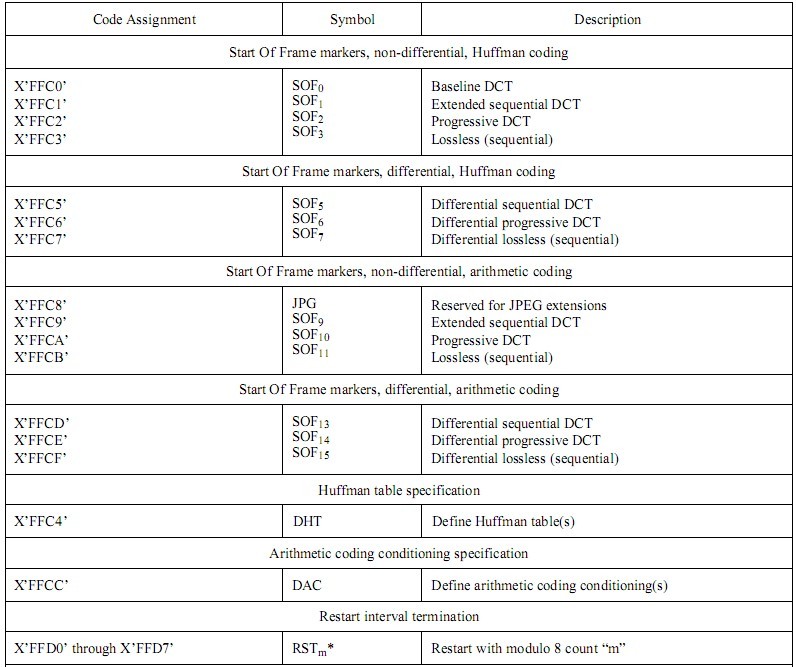
**B.1.1.2 标记**

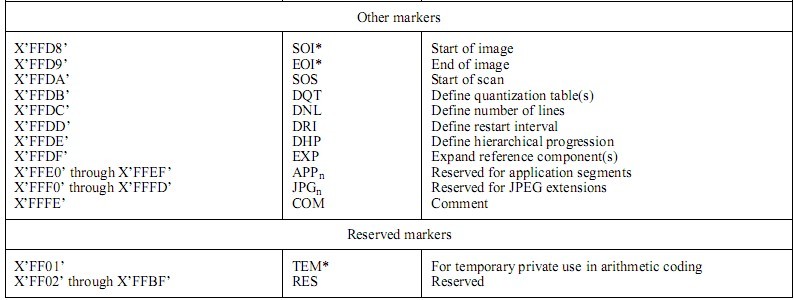
标记用于标识压缩数据格式的各个不同结构部分。大多数标记开启了一个标记段，而标记段则包含了一个相关的参数组。所有的标记均被分配一个2字节的代码：一个X’FF字节后跟一个不为0或X’FF’字节(参见表B.1)。所有的标记前面可以有一些可选的代码为X’FF的填充字节。

**注意** – 由于这种特别的代码分配结构，因此标记使解码器可以在不解码图像数据的其他段的情况下，而解析压缩数据并定位它的各个部分。

**B.1.1.3 标记分配**

所有的标记均被分配一个两字节的代码：一个X’FF’，及其后跟着的一个不为0或X’FF的第二个字节。每一个定义的标记的第二个字节在表B.1中指定。一个星号(\*)表示一个标记是独立的，即，它不是一个标记段的开始。

[](http://blog.chinaunix.net/attachment/201103/15/12567959_1300158639p2w5.jpg)

**[](http://blog.chinaunix.net/attachment/201103/15/12567959_13001586537DVv.jpg)**

**Table B.1 – Marker code assignments**

**B.1.1.4 标记段**

一个标记段由一个标记和该标记后面跟着的一个相关参数的序列组成。标记段中的第一个参数是2字节的长度参数。该长度参数编码了标记段以字节为单位的长度，包括长度参数，但不包括2字节的标记。以SOF和SOS标记码标识的标记段被称为头：分别为帧头和扫描头。

**B.1.1.5 熵编码数据段**

一个熵编码数据段包含了一个熵编码过程的输出。无论熵编码过程使用的是霍夫曼编码还是算术编码，它都由整数个自己组成。

**注意**

1按下面方法执行以使熵编码数据段具有整数个字节：对于霍夫曼编码，如果需要，则在压缩数据的最后填充值为1的位值来补齐段的最后一个字节。对于算术编码，字节对齐在终止熵编码段的过程中进行(参见D.1.8)。

2 为了确保一个标记不会出现在一个熵编码段中，则任何一个由霍夫曼编码或算术编码产生的X’FF’字节，或如上面注意1中所述，由值为1的位填充而得的一个X’FF’字节，其后要紧跟一个“填塞”0字节(参见D.1.6和F.1.2.3)。

**B.1.2 语法**

B.2和B.3中详述了交换格式的语法。就本规范的目的，语法规范的组成为：

– 要求的标记，参数和熵编码段的顺序；

– 可选的和有条件的组成部分的识别；

– 每一个标记和参数的名称、符号和识别；

– 每一个参数的允许值；

– 以上各条特定于各种编码过程的所有限制。

各组成部分的次序及可选的和有条件的部分的识别在语法插图B.2和B.3中示出。名称、符号、定义、允许值、条件和限制很快将在下面的各个语法插图中示出。

**B.1.3 语法插图约定**

B.2和B.3中的语法插图是交换格式规范的一部分。在图B.1中阐明的，下面的约定，应用于这些插图：

– 参数/标记指示符：一个细线的盒子包围一个标记或一个单独的参数；

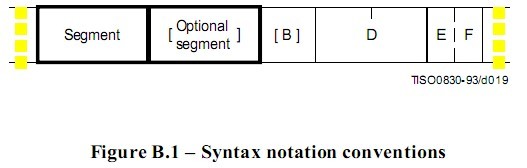
– 段指示符：一个细线的盒子包围一个标记段、一个熵编码段或他们的结合；

– 参数长度指示符：一个细线的盒子的宽度与标记的参数长度(4，8或16位，如图B.1中的E,B和D所示)或它包围的参数成比例；多个细线包围的盒子的宽度是无意义的；

– 可选的/有条件的指示符：方括号标明，在压缩数据中，一个标记段或标记只是可选的或有条件的。

– 次序：交换格式中，图中所示的参数或标记都先于其右边所示的参数或标记，并跟在它左边的参数后面。

– 熵编码数据指示符：尖括号标明，其包围的内容是已经经过了熵编码的。

[](http://blog.chinaunix.net/attachment/201103/11/12567959_129981200744Qy.jpg)

**B.1.4 符号，代码长度和值约定**

**在下面的B.2和B.3中的每一个语法插图，图中所示的每一个标记和参数符号、名称和定义都描述过了。对于每一个参数，长度和允许值也在表格中详述。**

**下面的约定应用于标记和参数的符号：**

**– 所有的标记符号有三个大写字母，而某些则还具有下标。如：SOI，SOFn；**

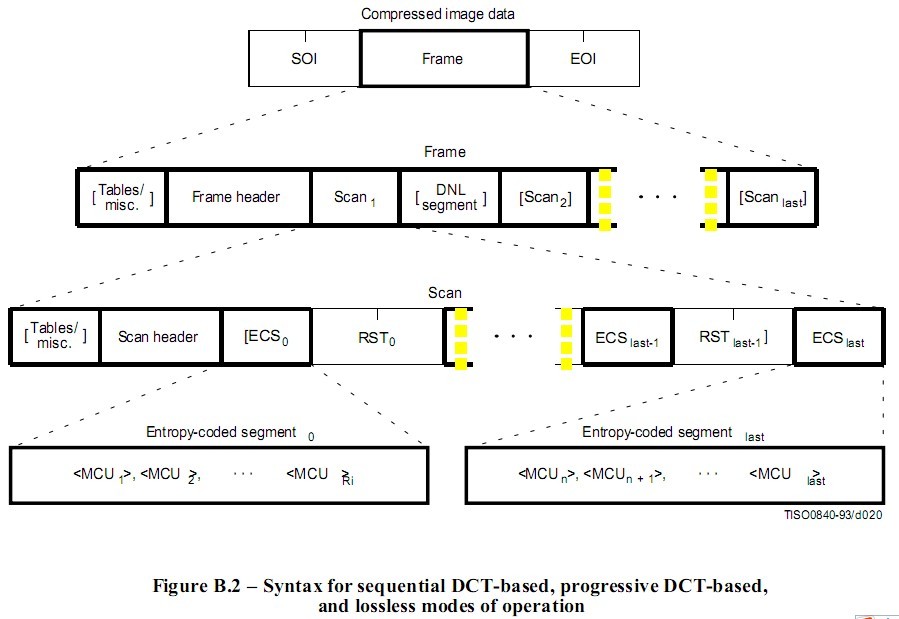
**– 所有的参数符号有一个大写字母；某些还具有一个小写字母和一些下标。如：Y，Nf，Hi，Tqi。**

**B.2 普通的顺序和增量语法**

这一章说明了应用于所有的编码过程的交换格式语法，包括基于DCT的顺序操作模式，基于DCT的增量操作模式和无损操作模式

**B.2.1 高层次语法**

图B.2说明了用于本规范中详述的所有的非层次式(non-hierarchical)编码过程的交换格式的高层次组成部分的顺序。

[](http://blog.chinaunix.net/attachment/201103/11/12567959_1299812127K8Ic.jpg)

图B.2所示的三个标记定义如下：

SOI: 图像开始标记 – 出现于交换格式或缩写格式中，标记一幅压缩图像的开始。

EOI: 图像结束标记 - 出现于交换格式或缩写格式中，标记一幅压缩图像的结束。

RSTm: 重新开始标记 – 只有重新开始使能时，放在熵编码段之间的一个有条件的标记。有8个唯一的重新开始标记(m = 0 - 7)，他们从0到7顺序的重复，每一个扫描的以0开始，以此来提供一个模8的重新开始间隔计数。

图B.2的最顶层说明了非层次式交换格式应该以SOI标记作为开始，包含一个帧，并以一个EOI标记结束。

图B.2的第二层说明一个帧应该以一个帧头开始并包含一个或多个扫描。一个帧头前面也许有一个或多个表规范或B.2.4中说明的多种标记段。如果有一个DNL段(参见B.2.5),那么它应该紧跟在第一扫描的后面。

对于基于DCT的顺序操作模式和无损过程，每一个扫描应该包含一到四个图像成分。如果一个扫描中2到4个成分，那么，它们应该在扫描中交替出现。对于基于DCT的增进式过程，每一个图像成分只是部分的包含于任何一个扫描中。只有成分(只包含DC系数数据)的第一个扫描可能是交替的。

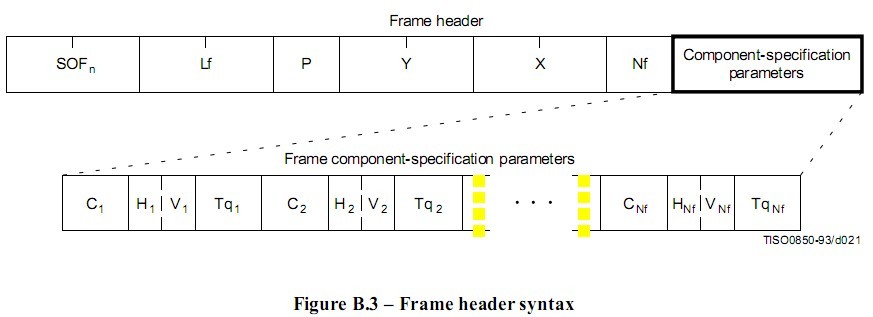
图B.2的第三层说明了一个扫描应该以一个扫描头开始，并包含一个或多个熵编码数据段。每一个扫描头前面可能有一个或多个表规范或多种标记段。如果重新开始没有被使能，则应该只有一个熵编码段(标记为“last”的那个)，并且没有重新开始标记。如果使能了重新开始，则熵编码段有图像的大小和DRI定义。这种情况下，一个重新开始标记应该跟在除了最后一个熵编码段之外的熵编码段之后。

图B.2的第四层说明每一个熵编码段由一系列熵编码MCUs组成。如果使能了重新开始，并且重新开始间隔被定义为Ri，则除了最后一个之外的每一个熵编码段应该包含Ri个MCUs，最后一个应该包含扫描的剩余的MCUs

图B.2说明了表规范可能出现的位置。然而，这里的规范特指交换格式应该包含的用于解码压缩图像的所有的表规范数据。因此，需要的表规范数据应该出现于一个或多个允许的位置。

**B.2.2 帧头语法**

图B.3说明了应该出现在一个帧的开始处的帧头。这个头说明了源图像特性(参见A.1)，包括包含于帧中的成分，每一个成分的采样因子，以及恢复每一个成分所使用的目标量化表。

[](http://blog.chinaunix.net/attachment/201103/11/12567959_12998122183Dyc.jpg)

图B.3中所示的标记和参数定义如下。每一个参数的大小和允许的值由表B.2给出。在表B.2(以及随后类似的表)中，可选的值由逗号隔开(e.g. 8, 12)，范围值的边界由破折号隔开(e.g. 0 - 3)。

SOFn: 帧标记的开始 – 标记帧参数的开始。下标n标识编码过程为基线顺序编码，扩展顺序编码，增量式编码，或无损编码，也标识使用了哪个熵编码过程。

SOF0: 基线DCT

SOF1: 扩展顺序DCT，霍夫曼编码

SOF2: 增量式DCT，霍夫曼编码

SOF3: 无损(顺序)的，霍夫曼编码

SOF9: 扩展顺序DCT, 算术编码

SOF10: 增量式DCT, 算术编码

SOF11: 无损(顺序)的，算术编码

Lf: 帧头的长度 – 说明了图B.3中所示的帧头的长度(参见B.1.1.4)。

P: 采样精度 – 以位为单位说明了帧中的成分的采样精度。

Y: 行数 – 说明了源图像的最大行数。它应该等于成分中具有最大垂直采样数的成分的行数(参见A.1.1)。值为0则表明行数应该由DNL标记定义，并且参数在第一个扫描的结尾(参见B.2.5)。

X: 每一行的采样数 - 说明了源图像中每一行的最大采样数。它应该等于成分中具有最大的水平采样数的成分的每一行的采样数。(参见A.1.1).

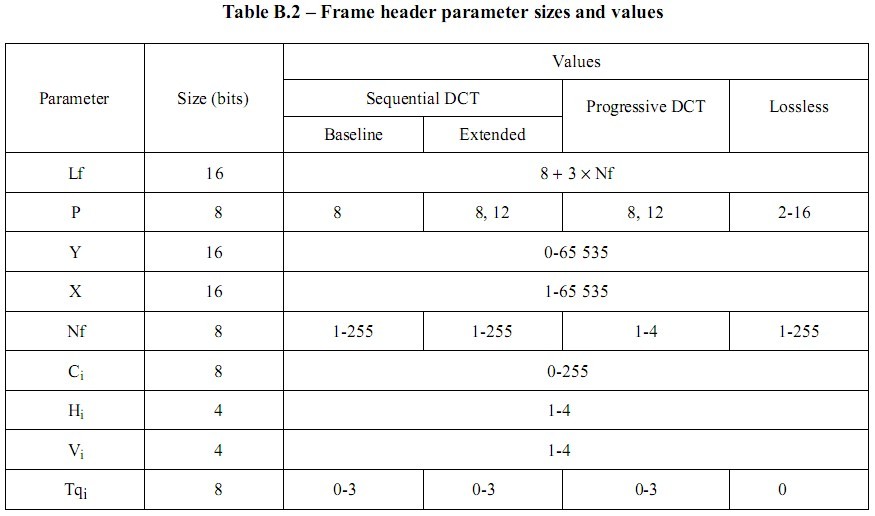
Nf: 帧中的图像成分数 – 指定了帧中源图像的成分数。Nf的值应该与出现在帧头中帧成分规范参数(Ci, Hi, Vi，and Tqi)的个数。

Ci: 成分标识符 – 顺序的为帧成分规范参数中的第i个参数分配一个唯一的标记。这些值应该被用于扫描头中来识别扫描中的成分Ci应该不同于C1到Ci − 1。

Hi: 水平采样因子 – 说明了成分的水平尺寸和最大的图像尺寸X之间的关系(参见A.1.1)；当一个扫描中编码了多于一个成分时，也指定了每一MCU中成分Ci的水平数据单元数。

Vi: 垂直采样因子 – 说明了成分的垂直尺寸和最大的图像尺寸Y之间的关系(参见A.1.1)；当一个扫描中编码了多于一个成分时，也指定了每一MCU中成分Ci的垂直数据单元数。

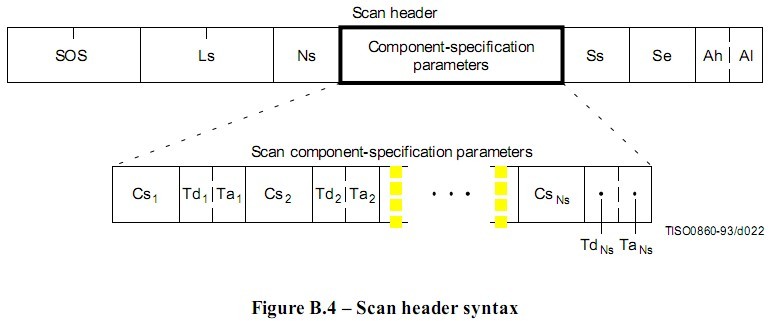
Tqi: 量化表目的选择子 – 指定了用于恢复成分Ci而对DCT系数反量化的四个可能的量化表中的一个。如果解码过程使用反量化例程，则在解码器已经准备好去解码扫描中包含的成分Ci时，这个表应该已经被安装在了目标位置。目标位置不应该被重新指定，或者改变它的内容，直到包含Ci的所有扫描均被解码完成。

[](http://blog.chinaunix.net/attachment/201103/11/12567959_1299812300RXFr.jpg)

**B.2.3 扫描头语法**

图B.4详细说明了应该出现在一个扫描的开始位置的扫描头。这个头指定扫描中包含哪些成分，恢复每一个成分时使用的熵编码表，和(用于增量式DCT)扫描中包含哪些量化后的DCT系数数据。对于无损过程，扫描参数指定预测值和点转换。

注意 – 如果扫描中只有一个图像成分，根据定义，那个成分时非交替的。如果扫描中有多个图像成分，根据定义，这些成分要交替出现。

[](http://blog.chinaunix.net/attachment/201103/11/12567959_12998123682O8O.jpg)

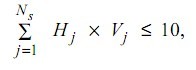
图B.4中所示的标记和参数定义如下。每一个参数大小和允许的值由表B.3给出。

SOS: 扫描标记的开始 – 标记扫描参数的开始。

Ls: 扫描头的长度 – 说明了图B.4中所示的扫描头的长度(参见B.1.1.4)。

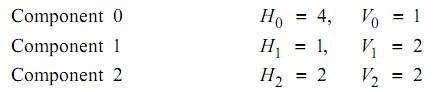
Ns: 扫描中图象成分的个数 – 说明了扫描中源图像成分的个数。Ns的值应该等于出现于扫描头中的扫描成分规范参数(Csj，Tdj，和Taj)的个数。

Csj: 扫描成分选择子 – 选择帧参数中详述的Nf个图像成分中的哪一个应该是扫描中的第j个成分。每一个Csj应该与帧头中详述的Ci值中的一个相对应，并且顺序应该与在帧头中的相同。如果Ns > 1，则MCU中交替的成分的顺序为Cs1第一，Cs2第二，等等。Ns> 1，则以下限制应该被施加于扫描中包含的图像成分上：

[](http://blog.chinaunix.net/attachment/201103/11/12567959_1299812418IOBS.jpg)

其中Hj和Vj是扫描成分j的水平和垂直采样因子。在帧头中这些采样因子被指定给成分i，其中i是帧成分规范索引，帧成分标识Ci对应于扫描成分选择子Csj。

例如，考虑一幅图像具有3个成分，最大尺寸为512行，每行512个采样值，并且采样因子如下：

[](http://blog.chinaunix.net/attachment/201103/11/12567959_1299812486to7J.jpg)

然后求和[http://blog.chinaunix.net/attachment/201103/11/12567959_1299812538Emy9.jpg](http://blog.chinaunix.net/attachment/201103/11/12567959_1299812538Emy9.jpg)

Csj的值应该不同于Cs1到Csj–1的值。

Tdj: DC熵编码表目标选择子 –指定四个可能的DC熵编码表中的一个，以满足恢复成分Csj的DC系数的需要。当解码器准备好解码当前的扫描时，DC熵编码表应该已经被安装在了这个目标位置(参见B.2.4.2和B.2.4.3)。对于无损过程，这个参数指定了熵编码表。

Taj: AC熵编码表目标选择子 - 指定四个可能的AC熵编码表中的一个，以满足恢复成分Csj的AC系数的需要。解码器准备好解码当前的扫描时，被选择的AC熵编码表应该已经被安装在了这个目标位置(参见B.2.4.2和B.2.4.3)。对于无损过程，这个参数为0.

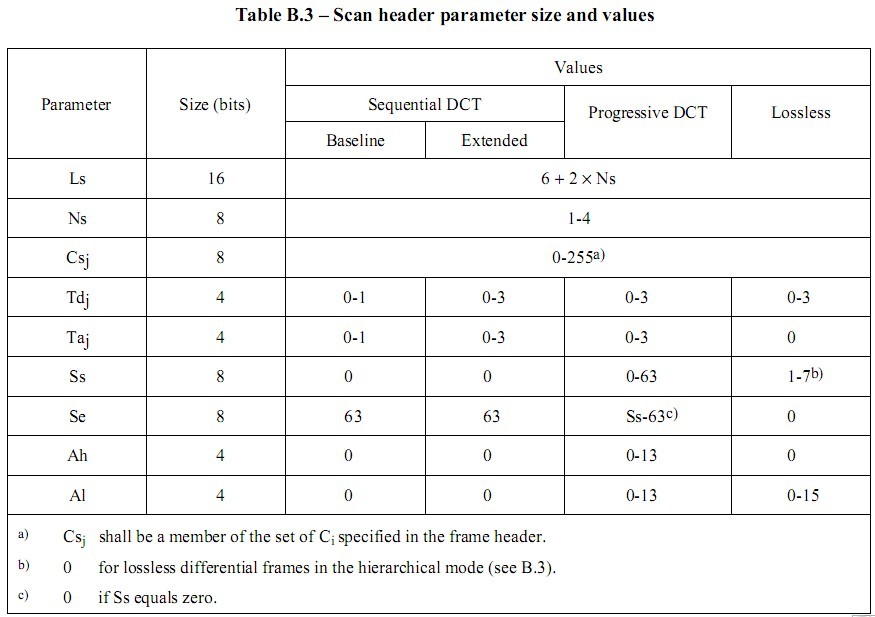
Ss: 频谱或预测值选择的开始 –在DCT操作模式中，这个参数指定了应该被编码进扫描中的每一个zig-zag序列的块的第一个DCT系数。对于顺序DCT过程，这个参数应该被设为0。在无损操作模式中，这个参数用于选择预测值。

Se: 频谱选择的结束 – 指定应该被编码进扫描中的每一个zig-zag序列的块的最后一个DCT系数。对于顺序DCT过程，这个参数应该被设为63。在无损操作模式中，这个参数无意义，它应该被设为0.

Ah: 逐次逼近位位置高 – 这个参数为由Ss和Se说明的系数的频带指定了用于前扫描的点转换(比如，在前扫描中逐次逼近位位置低)。对于系数的每一个频带的第一个扫描，这个参数应被设为0。在无损操作模式中，这个参数无意义，它应该被设为0.

Al: 逐次逼近位位置低或点转换 –在DCT操作模式中，这个参数指定点转换，比如，位位置低，在编码由Ss和Se说明的系数的频带前使用。对于顺序DCT过程，这个参数应被设为0。在无损操作模式中，这个参数指定点转换，Pt。

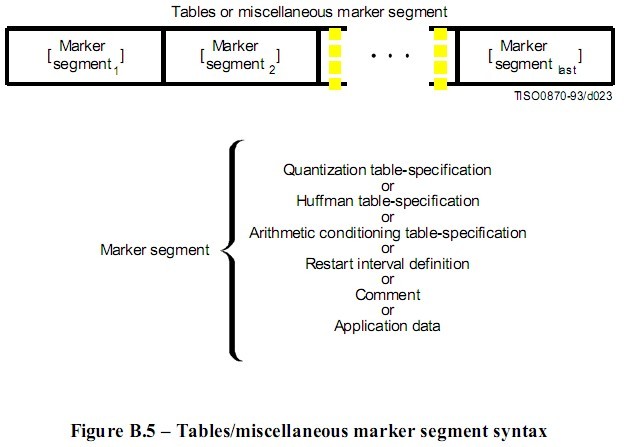
熵编码表目标选择子，Tdj和Taj，指定霍夫曼表（帧中使用霍夫曼编码）或者算术编码表(帧中使用算术编码)。在后一种情况中，熵编码表目标选择子指定一个算术编码调节表目标和一个相关的统计区域。

[](http://blog.chinaunix.net/attachment/201103/11/12567959_12998126147v3h.jpg)

**B.2.4 表规范和各种标记段语法**

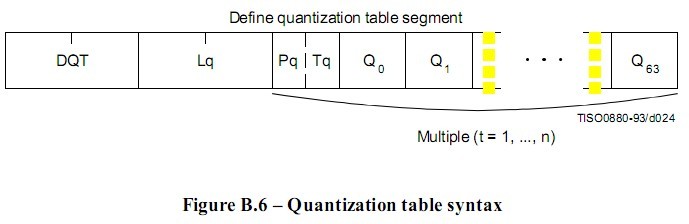
Figure B.5 specifies that, at the places indicated in Figure B.2, any of the table-specification segments or miscellaneous marker segments specified in B.2.4.1 through B.2.4.6 may be present in any order and with no limit on the number of segments.

如果任何一个为了特定目的的表规范出现在压缩数据中，那么它应该取代为了相同目的而指定的前面的表，并且，无论何时帧中剩余的扫描或者随后在压缩图像数据的缩略格式中出现的图像指定了这个目的，则都应该使用这个表规范。如果对于一个给定的目的有多余一个的表规范出现在压缩图像数据中，则每一个规范都应该取代前面的规范。量化表规范在一个给定成分的增量式DCT扫描之间不应该发生改变。

[](http://blog.chinaunix.net/attachment/201103/11/12567959_12998126756K56.jpg)

**B.2.4.1量化表规范语法**

图B.6详述了定义了一个或多个量化表的标记段。

[](http://blog.chinaunix.net/attachment/201103/11/12567959_1299812717n1bt.jpg)

图B.6所示的标记和参数定义如下。各个参数的大小和允许的值在表B.4中给出。

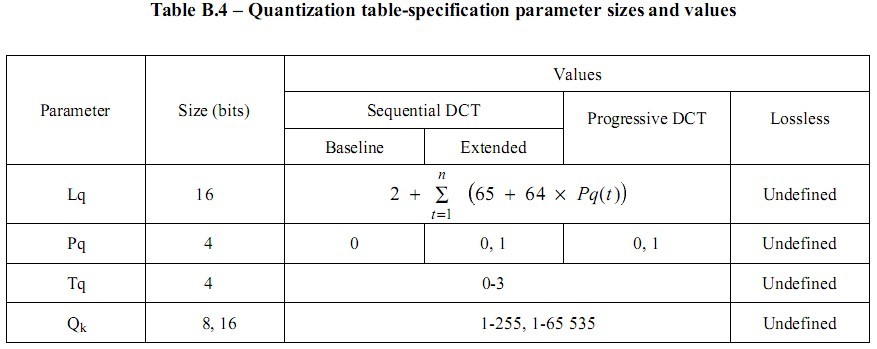
DQT: 定义量化表标记 – 标记量化表规范参数的开始。

Lq: 量化表定义的长度 – 说明了图B.6中所示的全部的量化表参数的长度(参见 B.1.1.4)。

Pq: 量化表元素精度 – 指定Qk值的精度。值0表示Qk 为8-bit；值1表示Qk为16-bit。对于8位采样精度P，Pq应该为0(参见B.2.2)。

Tq: 量化表目标标识 –为量化表将被安装到的解码器指定4个可能的目标中的一个。

Qk: 量化表元素 – 详述64个元素中的第k个元素，其中k是在DCT系数的zig-zag序列中的索引。量化元素应该以zig-zag扫描的顺序来制定。

[](http://blog.chinaunix.net/attachment/201103/11/12567959_12998127718GIg.jpg)

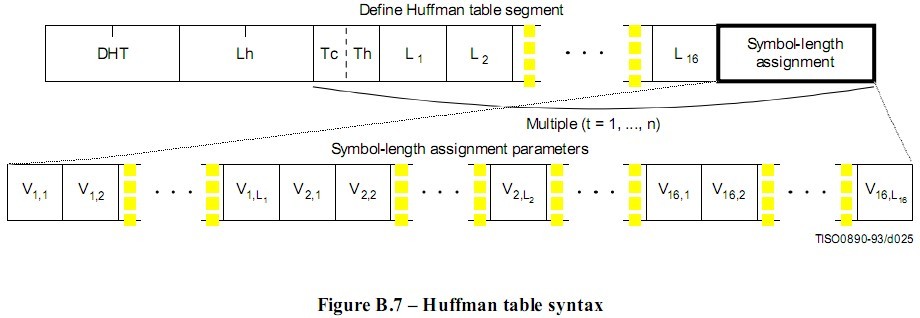
表B.4中的值n是DQT标记段描述的量化表的个数

如果已经为一个特定的目的定义了一个量化表，则它应该替代为那个目的而存储的先前的表，并且，在当前图像的剩余的扫描中，和随后在压缩图像数据的缩略格式中出现的图像中，需要引用该目的的表时，那个量化表应该得到使用。如果为了特定目的的表从来没有定义过，但在帧头中指定了这个目的时，则结果是不可预测的。

一个使用8-bit精度量化表的基于DCT的程序不应该使用一个16-bit精度的量化表。

**B.2.4.2霍夫曼表规范语法**

图B.7说明了用于定义一个或多个霍夫曼表规范的标记段。

[](http://blog.chinaunix.net/attachment/201103/11/12567959_1299812830hRfY.jpg)

图B.7中所示的标记和参数如下面定义。每一个参数的大小和允许的值有表B.5给出。

DHT: 定义霍夫曼表标记 – 标记霍夫曼表定义参数的开始。

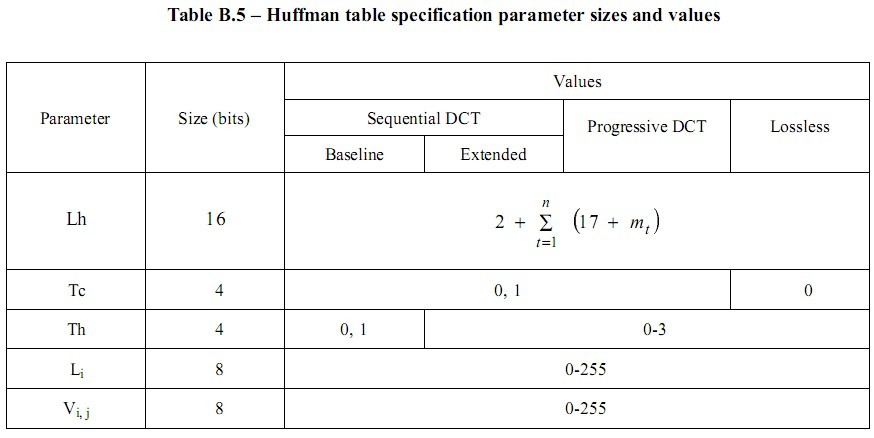
Lh: 霍夫曼表定义的长度 – 指定了图B.7中所示的所有霍夫曼表参数的长度(参见B.1.1.4)。

Tc: 表类 - 0 = DC表或无损的表，1 = AC表

Th: 霍夫曼表目标标识 – 在霍夫曼表应该被安装到的解码器中指定四个可能的目标之一。

Li: 霍夫曼编码长度为i的码字的个数– 说明本规范允许的霍夫曼编码的16个可能长度中的每一个的码字的个数。Li是表BITS的元素。

Vi,j: 与每一个霍夫曼编码相关的值 – 对于每一个i，指定与长度为i的每一个霍夫曼编码相关联的值。每一个值的含义由霍夫曼编码模型决定。Vi,j是表HUFFVAL的元素。

[](http://blog.chinaunix.net/attachment/201103/11/12567959_12998129113T3X.jpg)

表B.5中的值n是DHT标记段中描述的霍夫曼表的个数。mt是霍夫曼表t中紧跟在16个Li(t)参数后面的参数的个数，其值由下式给出：

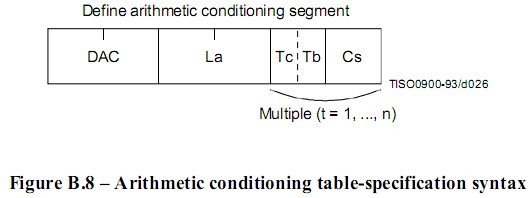
[http://blog.chinaunix.net/attachment/201103/11/12567959_1299812987r4wG.jpg](http://blog.chinaunix.net/attachment/201103/11/12567959_1299812987r4wG.jpg)

通常情况下，每一个表的mt值是不同的。

一旦已经为一个特定目标定义了一个霍夫曼表，则它应该取代前面的为了相同目的而定义的表，并且，在当前图像剩余的扫描中，和随后在压缩图像数据的缩略格式中出现的图像中，需要引用该目的的表时，那个量化表应该得到使用。如果为了特定目的的表从来没有定义过，但在帧头中指定了这个目的，则结果是不可预测的。

**B.2.4.3算术调节表规范语法**

图B.8说明了用于定义一个或多个算术调节表规范的标记段。他们替代由SOI标记为算术编码过程建立的算术编码调节表。参见(F.1.4.4.1.4和F.1.4.4.2.1)

[](http://blog.chinaunix.net/attachment/201103/11/12567959_1299813038vT69.jpg)

图B.8所示的标记和参数在下面定义。各个参数的大小和允许的值在表B.6中给出。

DAC: 定义算术编码调节标记 – 标记算术编码调节参数定义的开始。

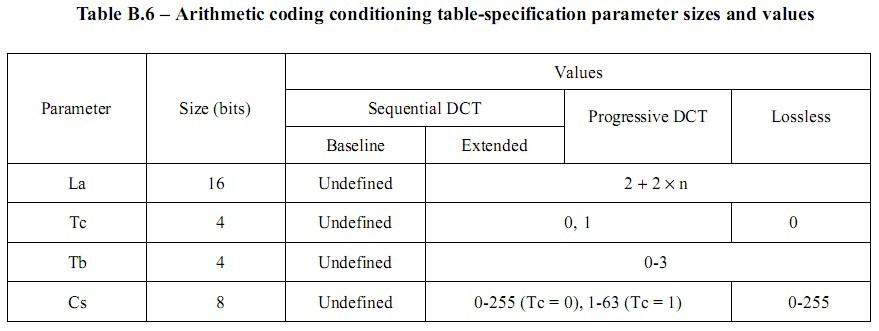
La: 算术编码调节定义长度 – 说明图B.8所示的所有算术编码调节参数的长度(参见B.1.1.4)

Tc: 表类 - 0 = DC表或无损的表，1 = AC表。

Tb: 算术编码调节表目的标识 -在算术编码调节表应该被安装到的解码器中指定四个可能的目标之一。

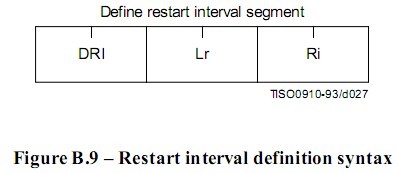
Cs: 调节表值 – 在AC或DC(和无损的)调节表中的值。Cs的一个单个的值应该跟在Tb的每一个值之后。对于AC调节表，Tc应该为1，并且Cs应该包含一个在范围1 ≤ Kx ≤ 63内的Kx值。对于DC(或无损的)调节表Tc应该为0，并且Cs应该包含两个4-bit参数，U和L。U和L应该在0 ≤ L ≤ U ≤ 15范围内，并且Cs的值应该为L + 16 × U

表B.6中的值n是DAC标记段中描述的算术编码调节表的个数。参数L和U是为DC系数编码和无损编码定义的在算术编码过程中使用的下部和上部调节边界。为DCT编码列出的范围在1-63的单独的值是用于AC系数编码的Kx调整。

[](http://blog.chinaunix.net/attachment/201103/11/12567959_1299813089xDdd.jpg)

**B.2.4.4重新开始间隔定义语法**

图B.9说明了用于定义重新开始间隔的标记段。

[](http://blog.chinaunix.net/attachment/201103/11/12567959_1299813147TrTv.jpg)

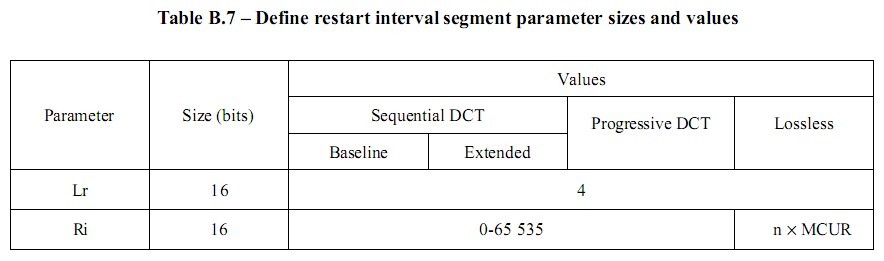
图B.9所示的标记和参数在下面定义。各个参数的大小和允许的值在表B.7中给出。

DRI: 重新开始间隔定义标记 – 标记了重新开始间隔定义参数的开始。

Lr: 重新开始间隔定义短长度 -说明了图B.9中所示的DRI段中的参数的长度(参见B.1.1.4)

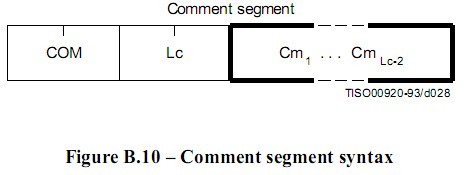
Ri: 重新开始间隔 – 指定了以MCU为单位的重新开始间隔。

表B.7中的值n是重新开始间隔中的MCU行数。MCUR值是扫描中构成每一个成分的一行采样值所需的MCU个数。SOI标记禁用重新开始标记。一个具有非零的Ri值的DRI标记段将出现以为接下来的扫描打开重新开始间隔过程。一个Ri值为0的DRI标记段将为接下来的扫描禁用重新开始间隔。

[](http://blog.chinaunix.net/attachment/201103/11/12567959_1299813198dwKh.jpg)

**B.2.4.5注释语法**

图B.10描述了用于一个注释段的标记段结构

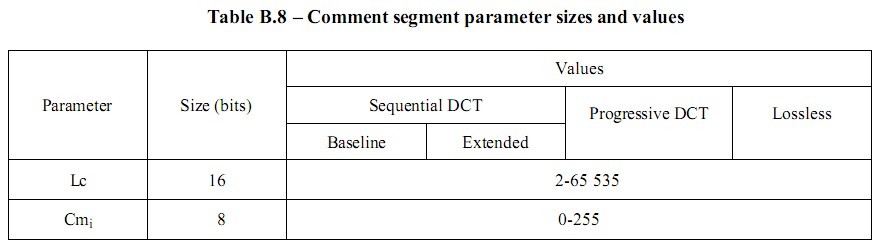
[](http://blog.chinaunix.net/attachment/201103/11/12567959_12998132484MkS.jpg)

图B.10所示的标记和参数在下面定义。各个参数的大小和允许的值在表B.8中给出。

COM: 注释标记 – 标记一个注释的开始。

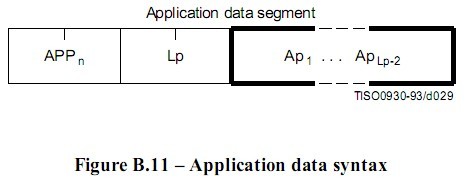
Lc: 注释段长度 – 说明了图B.10中所示的注释段的长度(参见B.1.1.4)。

Cmi: 注释字节 – 对注释的解释留给应用程序。

[](http://blog.chinaunix.net/attachment/201103/11/12567959_1299813280G55u.jpg)

**B.2.4.6应用程序数据语法**

图B.11详述了一个应用程序数据段的标记段结构。

[](http://blog.chinaunix.net/attachment/201103/11/12567959_1299813326ajZd.jpg)

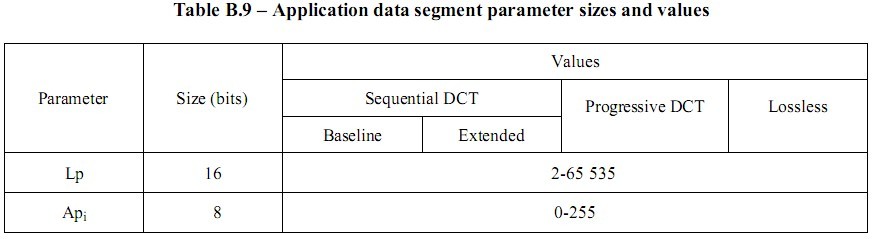
图B.11所示的标记和参数在下面定义。各个参数的大小和允许的值在表B.9中给出。

APPn: 应用程序数据标记 – 标记一个应用程序数据数据段的开始

Lp: 应用程序数据段长度 – 说明了图B.11中所示的应用程序数据段的长度(参见B.1.1.4)。

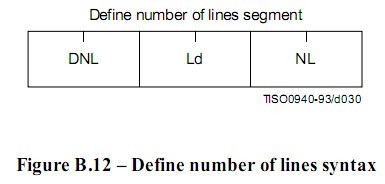
Api: 应用程序数据字节 – 解释权留给应用程序。

APPn(应用)段留给应用程序使用。由于这些段在不同的应用程序中可能有不同的定义，则当数据在应用环境之间交换式，他们应该被移除。

[](http://blog.chinaunix.net/attachment/201103/11/12567959_1299813386LLA2.jpg)

**B.2.5 行数定义语法**

图B.12详述了用于定义行数的标记段。DNL(Define Number of Lines)段为在第一个扫描结尾处定义或重新定义帧中行数(帧头中的Y参数)提供了一种机制。指定的值应该与第一个扫描中编码的MCU行数一致。如果使用了这个段，则它应该只出现在第一个扫描的结尾处，并且在编码了一个整数的MCU行之后使用。如果帧头中行数(Y)已经被指定为0，则这个标记段是必须的

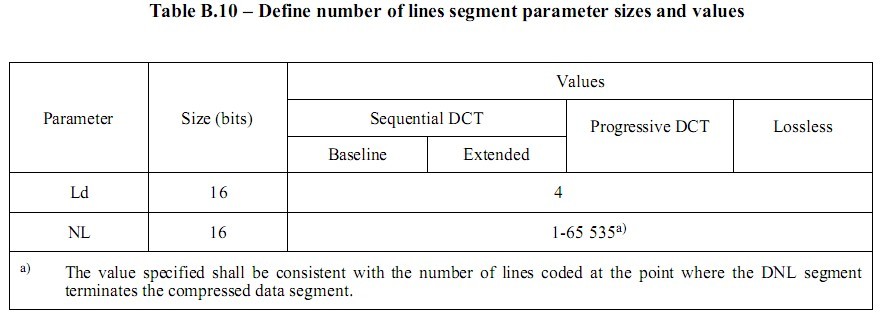
[](http://blog.chinaunix.net/attachment/201103/11/12567959_1299813465O5Po.jpg)

图B.12所示的标记和参数在下面定义。各个参数的大小和允许的值在表B.10中给出。

DNL: 行数定义标记 – 标记行数定义段的开始。

Ld: 行数定义段长度 – 说明了图B.12中所示的行数定义段的长度(参见B.1.1.4)

NL: 行数 – 指定了帧中的行数(参见B.2.2中的Y的定义)。

[](http://blog.chinaunix.net/attachment/201103/11/12567959_1299813501E5i5.jpg)