

# Huffman decoder

## Obsah

1. Popis.....	1
2. Závislosti .....	1
3. Implementačné informácie .....	1
4. Piny .....	2
5. Konfigurácia .....	2
5.1 Konfiguračná štruktúra.....	2
5.2 Konfiguračný dialóg.....	2
6. Definícia triedy .....	2
7. Príklad použitia .....	3

## 1. Popis



Trieda *Huffman decoder* dekoduje dáta zakódované triedou *Huffman encoder*. Vstupom sú binárne dáta zakódované Huffmanovým kódom. Výstupom je pole celých čísel.

## 2. Závislosti

Trieda *Huffman decoder* používa tieto externé definičné súbory a knižnice:

**Headers:** *filtergraph.h*, *datatypes.h*, *bitstream.h*

**Libs:** *filtergraph.lib*, *bitstream.lib*

## 3. Implementačné informácie

Informácie o triede:

**Názov:** *Huffman decoder*

**Verzia:** *1.0*

**Magic:** *200*

Informácie o definíciách:

**Header:** *huffmandecoder.h*

**Konfiguračný header:** *huffmandecodercfg.h*

**Lib:** *huffmandecoder.lib*

Informácie o knižnici obsahujúcej triedu:

**Názov knižnice:** *Huffman decoder Library*

**Verzia knižnice:** *1.0*

**Dll súbor knižnice:** *huffmandecoder.dll*

## 4. Piny

Piny sú popísané spôsobom: „*[index pinu na filtri] názov pinu: popis pinu*“.

**[0] Cfg in:** Konfiguračný vstup. Vstupom je serializovaná konfigurácia.

**[1] Cfg out:** Konfiguračný výstup. Výstupom je serializovaná konfigurácia.

**[2] In:** Dátový vstup. Vstupom je buffer binárnych dát zakódovaný triedou *Huffman encoder*.

**[3] Out:** Dátový výstup. Výstupom je pole celých čísel dekódované z binárnych dát, zapísané v štruktúre *TIntArray*.

## 5. Konfigurácia

### 5.1 Konfiguračná štruktúra

Konfiguračná štruktúra je definovaná nasledovne:

```
typedef struct {  
} THuffmanDecoderConfig;
```

Táto trieda nemá žiadne nastavenia. Výstup deterministicky závisí od zakódovaných dát vo vstupe.

### 5.2 Konfiguračný dialóg



Konfiguračný dialóg neposkytuje žiadne voľby.

## 6. Definícia triedy

Trieda *Huffman decoder* je definovaná nasledovne:

```
class THuffmanDecoderFilter : public TFilter {  
private:  
    HINSTANCE hInstance;           // identifikácia inštancie knižnice  
  
    TBuffer Input;                 // vstupné dáta  
    TIntArray Output;             // výstupné pole celých čísel  
  
    TPin* CfgInPin;               // vstupný konfiguračný pin  
    TPin* CfgOutPin;              // výstupný konfiguračný pin  
    TPin* InPin;                  // vstupný pin  
    TPin* OutPin;                 // výstupný pin
```

```

bool StopFlag;                                // indikátor zastavenia

CHuffmanDecoder *HuffmanDecoder;  // inštancia dekodéra (definovaný interne)

int setConfigFromPin(TPin *pin);           // nastaví konfiguráciu z pinu
int putConfigOnPin(TPin *pin);            // sprístupní konfiguráciu na pine
int freeConfigOnPin(TPin *pin);           // uvoľní konfiguráciu na pine
int clearOutputPins();                    // vyčistí výstupné piny
int freeOutputData();                     // uvoľní výstupné dáta
public:
    THuffmanDecoderFilter();              // konštruktor
    THuffmanDecoderFilter(HINSTANCE hInst); // konštruktor s parametrami
    ~THuffmanDecoderFilter();              // deštruktor

int setConfigData(TBuffer config,int type); // nastaví konfiguráciu
TBuffer getConfigData(int type);           // vráti konfiguráciu
int initialize();                          // inicializácia
int run();                                 // beh
int reset();                              // reset stavu
int stop();                              // ukončenie behu
int finalize();                           // finalizácia
int showConfigDialog();                   // vyvolá konfiguračný dialóg
};

```

## 7. Príklad použitia

