

Kristýna-GIS

Veřejné rozhraní zásuvných modulů pro rastrové formáty

Specifikace rozhraní verze 1.2



Obsah

1. Úvod	... 2
2. Použité datové typy	... 3
3. Jak to pracuje	... 5
3.1. Přidání rastrových dat do zobrazení	... 5
3.2. Vykreslování a tisk rastrových dat	... 5
3.3. Zobrazení vlastností rastrového tématu	... 5
3.4. Export obsahu zobrazení do rastrového souboru	... 6
4. Veřejné rozhraní knihovny	... 7
4.1. Minimální funkční požadavky	... 9
5. Instalace Vaší vlastní knihovny	... 11
5.1. Identifikační číslo formátu	... 11
6. Knihovny poskytované s Kristýnou	... 12
7. Příklad	... 14
7.1. Jak příklad zkompileovat	... 14
7.2. Zdrojový kód	... 14

© 2005 – 2021 Josef Genserek

Všechna práva vyhrazena.

Informace obsažené v tomto dokumentu jsou předmětem dalšího vývoje.

Použité názvy programových produktů, firem apod. mohou být ochrannými známkami nebo registrovanými ochrannými známkami příslušných vlastníků.

1. Úvod

Dokument popisuje rozhraní pro dynamicky připojované knihovny, které může být použito pro čtení a vykreslování prostorových rastrových dat. Rozhraní Vám dovoluje vytvořit Vaší vlastní knihovnu a tímto způsobem můžete rozšířit seznam rastrových formátů, které může Kristýna používat. Pokud vytvoříte svojí vlastní knihovnu, doporučujeme vytvořit také instalační program, který bude knihovnu instalovat. V kapitole *Instalace Vaší vlastní knihovny* naleznete několik doporučení.

Toto je popis rozhraní verze 1.2. Rozhraní je podporováno Kristýnou-GIS verze 3.0 a vyšší. Kristýna-GIS prohlížeč toto rozhraní nepodporuje. Tento materiál popisuje informace dostupné v čase jeho publikování a je v podstatě totožný s informacemi, které jsou obsaženy v nápovědě Kristýny. V některých případech můžete najít aktuálnější informace na internetových stránkách Kristýny (www.christine-gis.com).

2. Použité datové typy

Většina použitých datových typů jsou běžné datové typy používané v Microsoft Windows API.

BITMAP* - Ukazatel na strukturu, která definuje typ, šířku, výšku, barevný formát a data bitmapy.

```
typedef struct tagBITMAP {
    LONG    bmType;           // Specifikuje typ bitmapy (vždy nula)
    LONG    bmWidth;         // Specifikuje šířku bitmapy v pixelech
    LONG    bmHeight;        // Specifikuje výšku bitmapy v pixelech
    LONG    bmWidthBytes;    // Specifikuje počet bytů v každém řádku
    WORD    bmPlanes;        // Specifikuje počet barevných rovin (vždy 1)
    WORD    bmBitsPixel;     // Specifikuje počet bitů na pixel (vždy 24)
    LPVOID  bmBits;         // Ukazatel na data bitmapy
} BITMAP;
```

Každá trojice bytů v poli dat bitmapy představuje relativní intenzity modré, zelené a červené pro příslušný pixel. Data bitmapy jsou seřazena ze spodu na horu. To znamená, že první řádek v datech bitmapy je posledním řádkem, který bude zobrazen.

BOOL - Logický typ (nabývá hodnot TRUE nebo FALSE), je uchováván jako celočíselný typ se znaménkem

COLORREF - 32-bitová hodnota používaná pro specifikaci RGB barvy. Nejméně významný byte obsahuje hodnotu relativní intenzity červené složky, druhý byte obsahuje hodnotu pro zelenou, a třetí byte obsahuje hodnotu pro modrou. Nejvýznamnější byte musí být nulový. Maximální hodnota jednoho byte je 0xFF.

CRS2CRS – Ukazatel na funkci pro konverzi mezi souřadnicovými systémy. Ukazatel na funkci je deklarován takto:

```
typedef BOOL (__cdecl *CRS2CRS)(int, int, double*, double*, double*)
```

Funkce vrací TRUE pokud je konverze úspěšná, jinak vrací hodnotu FALSE. První parametr funkce je počet bodů ke konverzi. Druhý parametr je posun mezi hodnotami souřadnic v poli souřadnic. Třetí, čtvrtý a pátý parametr jsou ukazatelé na pole souřadnic x, y a z.

double - 64-bitové desetinné číslo se znaménkem

double* - Ukazatel na 64-bitové desetinné číslo se znaménkem

DWORD - 32-bitové celé číslo bez znaménka

EXTENT - Struktura, která definuje rozsah souřadnic X, Y, Z a M hodnot

```
typedef struct _EXTENT {
    double xMin;           // minimální hodnota souřadnice X
    double yMin;           // minimální hodnota souřadnice Y
    double xMax;           // maximální hodnota souřadnice X
    double yMax;           // maximální hodnota souřadnice Y
    double zMin;           // minimální hodnota souřadnice Z (ignorováno)
    double zMax;           // maximální hodnota souřadnice Z (ignorováno)
    double mMin;           // minimální hodnota hodnoty M (ignorováno)
    double mMax;           // maximální hodnota hodnoty M (ignorováno)
} EXTENT;
```

EXTENT* - Ukazatel na strukturu rozsahu

FALSE - Logická hodnota, uložena jako celé číslo se znaménkem s hodnotou nula

HDC - Popisovač kontextu zařízení

HWND - Popisovač okna

int - 32-bitové celé číslo se znaménkem

LONG - 32-bitové celé číslo se znaménkem

NULL - Nulová hodnota

RECT - Struktura definující souřadnice levého horního a pravého spodního rohu obdélníka

```
typedef struct _RECT {  
    LONG left;        // x souřadnice levého horního rohu obdélníka  
    LONG top;         // y souřadnice levého horního rohu obdélníka  
    LONG right;       // x souřadnice pravého dolního rohu obdélníka  
    LONG bottom;     // y souřadnice pravého dolního rohu obdélníka  
} RECT;
```

RECT* - Ukazatel na strukturu popisující obdélník

TCHAR* - Ukazatel na pole znaků (řetězec)

TRUE - Logická hodnota, uložena jako celé číslo se znaménkem s nenulovou hodnotou

UINT - 32-bitové celé číslo bez znaménka

void - Použito jako návratový typ funkce specifikuje, že funkce nevrací žádnou hodnotu

void* - Ukazatel na nespecifikovaný datový typ nebo strukturu ("univerzální ukazatel")

WORD - 16-bitové celé číslo bez znaménka

3. Jak to pracuje

Kristýna volá funkce veřejného rozhraní popsané v kapitole *Veřejné rozhraní knihovny* dle potřeby. Knihovna je uvolňována po každém volání funkce, kromě volání funkcí `void SetColorChanging` (DWORD nCount, COLORREF* colors), `void SetTransparency` (DWORD transparencyPercentage), `void SetTransparentColor` (COLORREF color) a `void UseTransparentColor` (BOOL bUse) před voláním funkce `void Draw` (TCHAR* fileName, HWND hWndMainFrame, HDC hDC, EXTENT* imgExtent, EXTENT* drawExtent, RECT* clipRect, CRS2CRS Crs2Crs, DWORD* pId1, volatile DWORD* pId2).

3.1 Přidání rastrových dat do zobrazení

Kristýna najde v registrech Windows dostupné rastrové formáty. Kristýna použije hodnoty z klíčů *FormatName*, *FormatExtensions* a *FormatID* aby vytvořila seznam dostupných formátů a naplní jimi rozbalovací nabídku v dialogovém okně pro přidání tématu. Pokud uživatel zvolí rastrový formát, který používá toto rozhraní, Kristýna přečte v registrech Windows hodnotu klíče *LibFileName*.

Seznam volaných funkcí během akce:

```
EXTENT* GetExtent(TCHAR* fileName)
```

3.2 Vykreslování a tisk rastrových dat

Kristýna nahraje knihovnu do paměti a nastaví průsvitnost rastru a barvu která bude považována za průhlednou. Poté zavolá funkci Draw. Jakmile je vykreslení hotovo, uvolní knihovnu.

Seznam volaných funkcí během akce:

```
void Draw(TCHAR* fileName, HWND hWndMainFrame, HDC hDC, EXTENT* imgExtent,
EXTENT* drawExtent, RECT* clipRect, CRS2CRS Crs2Crs, DWORD* pId1, volatile
DWORD* pId2)
void SetColorChanging(DWORD nCount, COLORREF* colors)
void SetTransparency(DWORD transparencyPercentage)
void SetTransparentColor(COLORREF color)
void UseTransparentColor(BOOL bUse)
```

3.3 Zobrazení vlastností rastrového tématu

Před tím, než je zobrazen dialog Vlastnosti tématu, Kristýna zavolá sérii funkcí aby získala všechny potřebné informace.

Seznam volaných funkcí během akce:

```
DWORD GetColorDeep(TCHAR* fileName)
TCHAR* GetCompressionTypeString(TCHAR* fileName)
EXTENT* GetExtent(TCHAR* fileName)
TCHAR* GetFormatName(TCHAR* fileName)
TCHAR* GetFormatVersion(TCHAR* fileName)
DWORD GetHeight(TCHAR* fileName)
DWORD GetInterfaceVersion()
```

```
DWORD GetWidth(TCHAR* fileName)
TCHAR* GetWorldFileName(TCHAR* fileName)
BOOL HasPyramid(TCHAR* fileName)
BOOL SupportsColorChanging()
BOOL SupportsTransparency()
BOOL SupportsTransparentColor()
BOOL SupportsPyramids()
```

3.4 Export obsahu zobrazení do rastrového souboru

Kristýna najde v registrech Windows dostupné rastrové formáty. Pro každý formát si v registrech Windows přečte hodnotu klíče *LibFileName* a načte dynamickou knihovnu. Poté zavolá funkci *SupportsExport* a uvolní knihovnu. Pokud uvedená funkce vrátí hodnotu TRUE, Kristýna použije hodnoty klíčů *FormatName*, *FormatExtensions* a *FormatID* aby přidala formát do dialogového okna pro export. Když uživatel zvolí rastrový formát, který používá toto rozhraní, Kristýna znovu přečte hodnotu klíče *LibFileName* v registrech Windows, nahraje knihovnu a zavolá funkci *Export*. Jakmile je export ukončen, Kristýna uvolní knihovnu.

Seznam volaných funkcí během akce:

```
EXTENT* GetExtent(TCHAR* fileName)
BOOL Export(TCHAR* outFileName, HWND hWndMainFrame, BITMAP* pBitmap, EXTENT*
exportedExtent, UINT mapUnits, BOOL bCreateWorldFile)
BOOL SupportsExport()
```

4. Veřejné rozhraní knihovny

Tato kapitola popisuje veřejné rozhraní pro dynamické knihovny. Každá funkce v tomto rozhraní má přesně definované chování a je volána v situacích popsaných v sekci *Jak to pracuje*.

BOOL BuildPyramid(TCHAR* fileName)

Vytvoří pyramidu nad rastrovými daty. Pyramida je sada malých georeferencovaných kopií dat, které umožní rychlejší vykreslení rastrových dat v malých měřících. Funkce vrátí TRUE pokud je pyramida úspěšně vytvořena.

void Draw(TCHAR* fileName, HWND hWndMainFrame, HDC hDC, EXTENT* imgExtent, EXTENT* drawExtent, RECT* clipRect, CRS2CRS Crs2Crs, DWORD* pId1, volatile DWORD* pId2)

Vykreslí rastrová data na kontext zařízení s popisovačem hDC. Parametr fileName je celé jméno souboru s rastrovými daty. Parametr hWndMainFrame je popisovač hlavního okna Kristýny. Parametr imgExtent je rozsah rastrových dat v mapových jednotkách (v podstatě je to rozsah rastrového tématu). Parametr drawExtent je rozsah pro vykreslení v mapových jednotkách. Parametr clipRect je ořezávací obdélník v jednotkách obrazovky. Parametr Crs2Crs je ukazatel na funkci pro konverzi mezi souřadnicovými systémy. Pokud není konverze nutná má parametr Crs2Crs hodnotu NULL. Parametry pId1 a pId2 jsou ukazatelé na číselné identifikátory. Pokud jsou hodnoty identifikátorů totožné můžete pokračovat ve vykreslování, ale jakmile je identifikátor pId2 změněn měli by jste okamžitě zastavit vykreslování a uvolnit paměť.

BOOL Export(TCHAR* outFileFileName, HWND hWndMainFrame, BITMAP* pBitmap, EXTENT* exportedExtent, UINT mapUnits, BOOL bCreateWorldFile)

Převede bitmapu do požadovaného formátu a uloží ji do souboru. Parametr outFileFileName je celé jméno souboru. Parametr hWndMainFrame je popisovač hlavního okna Kristýny. Parametr pBitmap je ukazatel na strukturu BITMAP. Parametr exportedExtent je rozsah exportovaných dat v mapových jednotkách. Tento rozsah potřebujete znát aby jste byli schopni vytvořit souřadnicový soubor (world file). Parametr mapUnits specifikuje mapové jednotky a může nabýt následujících hodnot:

<i>Neznámé mapové jednotky</i>	40244
<i>Kilometry</i>	40245
<i>Metry</i>	40246
<i>Decimetry</i>	40253
<i>Centimetry</i>	40247
<i>Milimetry</i>	40248
<i>Míle</i>	40249
<i>Stopy</i>	40250
<i>Palce</i>	40251
<i>Yardy</i>	40254
<i>Námořní míle</i>	40255
<i>sáhy</i>	40256
<i>řetězy</i>	40257
<i>články</i>	40258
<i>US zeměměřické palce</i>	40259
<i>US zeměměřické stopy</i>	40260
<i>US zeměměřické yardy</i>	40261
<i>US zeměměřické řetězy</i>	40262
<i>US zeměměřické míle</i>	40263
<i>Indiánské stopy</i>	40264
<i>Indiánské yardy</i>	40265
<i>Indiánské řetězy</i>	40266
<i>Desetinné stupně</i>	40252

Pokud má parametr bCreateWorldFile hodnotu TRUE funkce by měla vytvořit souřadnicový soubor, nebo zajistit souřadnicové připojení způsobem běžným pro daný formát. Funkce vrátí TRUE pokud byl soubor úspěšně vytvořen, jinak vrátí FALSE.

DWORD GetColorDeep(TCHAR* fileName)

Vrací barevnou hloubku rastrových dat. Parametr fileName je celé jméno souboru s rastrovými daty.

TCHAR* **GetCompressionTypeString**(TCHAR* fileName)

Vrací ukazatel na řetězec se jménem kompresní metody. Pokud data nejsou komprimována vrací řetězec „žádná“. Parametr fileName je celé jméno souboru s rastrovými daty.

EXTENT* **GetExtent**(TCHAR* fileName)

Vrací ukazatel na původní (neprojektovaný) rozsah rastrových dat. Parametr fileName je celé jméno souboru s rastrovými daty.

TCHAR* **GetFormatName**(TCHAR* fileName)

Vrací ukazatel na řetězec se jménem datového formátu. Parametr fileName je celé jméno souboru s rastrovými daty.

TCHAR* **GetFormatVersion**(TCHAR* fileName)

Vrací ukazatel na řetězec s verzí formátu rastrových dat. Parametr fileName je celé jméno souboru s rastrovými daty.

DWORD **GetHeight**(TCHAR* fileName)

Vrací výšku rastrových dat v pixelech. Parametr fileName je celé jméno souboru s rastrovými daty.

DWORD **GetInterfaceVersion**()

Vrací verzi rozhraní. Horní byte obsahuje číslo verze (1) a dolní byte číslo podverze (2).

DWORD **GetWidth**(TCHAR* fileName)

Vrací šířku rastrových dat v pixelech. Parametr fileName je celé jméno souboru s rastrovými daty.

TCHAR* **GetWorldFileName**(TCHAR* fileName)

Vrací ukazatel na řetězec s celým jménem souřadnicového souboru (world file). Pokud rastrová data nemají souřadnicový soubor vrací NULL. Parametr fileName je celé jméno souboru s rastrovými daty.

BOOL **HasPyramid**(TCHAR* fileName)

Vrací TRUE pokud mají rastrová data vytvořeny pyramidu. Parametr fileName je celé jméno souboru s rastrovými daty.

void **RemovePyramid**(TCHAR* fileName)

Smaže pyramidu. Parametr fileName je celé jméno souboru s rastrovými daty.

void **SetColorChanging**(DWORD nCount, COLORREF* colors)

Nastaví hodnoty pro záměnu barev. Parametr nCount udává celkový počet barev v poli colors. Parametr colors je pole dvojic barev pro záměnu. Liché prvek ve dvojici je původní barvu, sudý prvek je nová barva. Používá se před voláním funkce Draw.

void **SetTransparency**(DWORD transparencyPercentage)

Nastaví průsvitnost na hodnotu transparencyPercentage. Používá se před voláním funkce Draw.

void **SetTransparentColor**(COLORREF color)

Nastaví průhlednou barvu na color. Používá se před voláním funkce Draw.

BOOL **SupportsColorChanging**()

Vrací TRUE pokud knihovna podporuje mechanismus záměny barev, jinak vrací FALSE.

BOOL **SupportsExport**()

Vrací TRUE pokud knihovna podporuje mechanismus exportu, jinak vrací FALSE.

BOOL **SupportsTransparency**()

Vrací TRUE pokud knihovna podporuje mechanismus průsvitnosti, jinak vrací FALSE.

BOOL **SupportsTransparentColor**()

Vrací TRUE pokud knihovna podporuje mechanismus průhledné barvy, jinak vrací FALSE.

BOOL **SupportsPyramids**()

Vrací TRUE pokud knihovna podporuje mechanismus pyramid, jinak vrací FALSE.

```
void UseTransparentColor(BOOL bUse)
```

Pokud má parametr bUse hodnotu TRUE, má být použita transparentní barva.

4.1 Minimální funkční požadavky

Minimální požadavek je vykreslení rastrových dat v zobrazení a tisk těchto dat na tiskárně. K vytvoření fungující knihovny, která toto splňuje musíte implementovat následující funkce s popsáním chováním:

```
BOOL BuildPyramid(TCHAR* fileName)
```

Vrátí FALSE.

```
void Draw(TCHAR* fileName, HWND hWndMainFrame, HDC hDC, EXTENT* imgExtent,  
EXTENT* drawExtent, RECT* clipRect, CRS2CRS Crs2Crs, DWORD* pId1, volatile  
DWORD* pId2)
```

Vykreslí rastrová data na kontext zařízení s popisovačem hDC.

```
BOOL Export(TCHAR* outFileName, HWND hWndMainFrame, BITMAP* pBitmap, EXTENT*  
exportedExtent, UINT mapUnits, BOOL bCreateWorldFile)
```

Vrátí FALSE.

```
DWORD GetColorDeep(TCHAR* fileName)
```

Vrátí barevnou hloubku rastrových dat.

```
TCHAR* GetCompressionTypeString(TCHAR* fileName)
```

Vrátí ukazatel na řetězec se jménem kompresní metody. Pokud data nejsou komprimována vrátí řetězec „žádná“.

```
EXTENT* GetExtent(TCHAR* fileName)
```

Vrátí ukazatel na rozsah rastrových dat.

```
TCHAR* GetFormatName(TCHAR* fileName)
```

Vrátí ukazatel na řetězec se jménem datového formátu.

```
TCHAR* GetFormatVersion(TCHAR* fileName)
```

Vrátí verzi formátu rastrových dat.

```
DWORD GetHeight(TCHAR* fileName)
```

Vrátí výšku rastrových dat v pixelech.

```
DWORD GetInterfaceVersion()
```

Vrátí verzi rozhraní. Horní byte obsahuje číslo verze (1) a dolní byte číslo podverze (2).

```
DWORD GetWidth(TCHAR* fileName)
```

Vrátí šířku rastrových dat v pixelech.

```
TCHAR* GetWorldFileName(TCHAR* fileName)
```

Vrátí ukazatel na řetězec s celým jménem souřadnicového souboru (world file). Pokud rastrová data nemají souřadnicový soubor vrátí NULL.

```
BOOL HasPyramid(TCHAR* fileName)
```

Vrátí FALSE.

```
void RemovePyramid(TCHAR* fileName)
```

Nedělá nic.

```
void SetColorChanging(DWORD nCount, COLORREF* colors)
```

Nedělá nic.

void **SetTransparency**(DWORD transparencyPercentage)

Nedělá nic.

void **SetTransparentColor**(COLORREF color)

Nedělá nic.

BOOL **SupportsColorChanging**()

Vrátí FALSE.

BOOL **SupportsExport**()

Vrátí FALSE.

BOOL **SupportsTransparency**()

Vrátí FALSE.

BOOL **SupportsTransparentColor**()

Vrátí FALSE.

BOOL **SupportsPyramids**()

Vrátí FALSE.

void **UseTransparentColor**(BOOL bUse)

Nedělá nic.

5. Instalace Vaší vlastní knihovny

Doporučujeme vytvořit instalační program pro instalaci knihovny. Instalační program musí umístit knihovnu do podadresáře dlls v instalačním adresáři Kristýny. Instalační program musí také zapsat následující informace do registrů Windows:

Klíč:

HKEY_LOCAL_MACHINE\Software\Christine-GIS\3.x\RasterFormats\[ID formátu]

Hodnoty:

Název	Hodnota	Typ	Max. délka (byte)	
LibFileName	[library.dll]	REG_SZ	255	povinné
FormatExtensions	[*.xxx;*.xxxx]	REG_SZ	63	povinné
FormatName	Název rastrového formátu	REG_SZ	255	povinné
FormatID	[ID formátu]	REG_DWORD	4	povinné
MimeType	[image/*]	REG_SZ	63	nepovinné

Příklad pro Tag Image File Format:

Klíč:

HKEY_LOCAL_MACHINE\Software\Christine-GIS\3.x\RasterFormats\3

Hodnoty:

Název	Hodnota
LibFileName	tiff.dll
FormatExtensions	*.tif;*.tiff
FormatName	Tag Image File Format
FormatID	3
MimeType	image/tiff

Poznámka: Hodnota *FormatID* musí být stejná jako název klíče. Nepovinná hodnota *MimeType* je nutná pouze když rastrový formát může být používán jako výstupní formát nějakého internetového mapového serveru.

5.1 Identifikační číslo formátu

Pokud chcete vytvářet vlastní knihovnu pro čtení rastrových dat, ověřte si na webové stránce Kristýny volná identifikační čísla formátu. Poté vyberte volné číslo, kontaktujte nás a my zveřejníme Vámi zvolené číslo na webu Kristýny abychom zabránili problémům s duplicitou. Identifikační číslo formátu je používáno interně Kristýnou a těmito metodami ve skriptovacím jazyce Kristýny: GetType(Number nType) ze třídy Theme, ExportToBmp(String sFullFileName, Number nType, Rect extent, Number nWidth, Number nHeight, Bool bCreateWorldFile) a AddTheme(String sFullFileName, Number nType) ze třídy View, ExportToVectorFormat(String sFullFileName, Number nFormatID, Bool bUseDefaultSettings) ze třídy FTheme, Export(String sFullFileName, Number nOutputType) ze třídy DSCTheme, ShowBmp(String sFullFileName, Number nDelay, Bool bFrame, Number nType) ze třídy MsgBox.

Doporučené rozsahy identifikačních čísel formátů jsou následující:

0 - 1000 pro rastrová a vektorová souborová data (typicky uložena na souborovém systému), rozsah 0 - 100 je rezervován pro Kristýnu-GIS, 101 - 1000 je k dispozici pro vývojáře třetích stran

1001 - 2000 pro rastrová a vektorová data z webovských služeb (typicky generována mapovými servery a zaslána Kristýně), rozsah 1001 - 1100 je rezervován pro Kristýnu-GIS, 1101 - 2000 je k dispozici pro vývojáře třetích stran

2001 - 3000 pro rastrová a vektorová data uložena v geodatabázi (pro budoucí využití, nyní nepodporováno), rozsah 2001 - 2100 je rezervován pro Kristýnu-GIS, 2101 - 3000 je k dispozici pro vývojáře třetích stran

6. Knihovny poskytované s Kristýnou

S Kristýnou-GIS verze 4.0 je poskytováno osm knihoven, které mají implementováno toto rozhraní. Všechny tyto knihovny jsou umístěny v podadresáři dlls v instalačním adresáři Kristýny.

Název formátu:	Tag Image File Format
Jméno knihovny:	tiff.dll
ID formátu:	3
Rozšíření souboru:	*.tif;*.tiff
MINE typ:	image/tiff
Verze rozhraní:	1.2
Podpora záměny barev:	Ano
Podpora exportu:	Ano
Podpora průsvitnosti:	Ano
Podpora průhledné barvy:	Ano
Podpora pyramid:	Ano
Název formátu:	Join Photographic Experts Group
Jméno knihovny:	jpeg.dll
ID formátu:	5
Rozšíření souboru:	*.jpg;*.jpeg
MINE typ:	image/jpeg
Verze rozhraní:	1.2
Podpora záměny barev:	Ano
Podpora exportu:	Ano
Podpora průsvitnosti:	Ano
Podpora průhledné barvy:	Ano
Podpora pyramid:	Ano
Název formátu:	Portable Network Graphics
Jméno knihovny:	png.dll
ID formátu:	7
Rozšíření souboru:	*.png
MINE typ:	image/png
Verze rozhraní:	1.2
Podpora záměny barev:	Ano
Podpora exportu:	Ano
Podpora průsvitnosti:	Ano
Podpora průhledné barvy:	Ano
Podpora pyramid:	Ano
Název formátu:	ER Mapper ECW
Jméno knihovny:	ecw.dll
ID formátu:	8
Rozšíření souboru:	*.ecw
Verze rozhraní:	1.2
Podpora záměny barev:	Ano
Podpora exportu:	Ano
Podpora průsvitnosti:	Ano
Podpora průhledné barvy:	Ano
Podpora pyramid:	Ano
Název formátu:	MrSID
Jméno knihovny:	mrsid.dll
ID formátu:	9
Rozšíření souboru:	*.sid
Verze rozhraní:	1.2
Podpora záměny barev:	Ano
Podpora exportu:	Ne
Podpora průsvitnosti:	Ano

Podpora průhledné barvy: Ano
Podpora pyramid: Ano

Název formátu: JPEG 2000
Jméno knihovny: jpeg2000.dll
ID formátu: 10
Rozšíření souboru: *.jp2
MINE typ: image/jp2
Verze rozhraní: 1.2
Podpora záměny barev: Ano
Podpora exportu: Ne
Podpora průsvitnosti: Ano
Podpora průhledné barvy: Ano
Podpora pyramid: Ano

Název formátu: Graphics Interchange Format
Jméno knihovny: gif.dll
ID formátu: 11
Rozšíření souboru: *.gif
MINE typ: image/gif
Verze rozhraní: 1.2
Podpora záměny barev: Ano
Podpora exportu: Ano
Podpora průsvitnosti: Ano
Podpora průhledné barvy: Ano
Podpora pyramid: Ano

Název formátu: Intergraph Raster File
Jméno knihovny: ingr.dll
ID formátu: 12
Rozšíření souboru: *.cit;*.g3;*.g4
Verze rozhraní: 1.2
Podpora záměny barev: Ano
Podpora exportu: Ne
Podpora průsvitnosti: Ano
Podpora průhledné barvy: Ano
Podpora pyramid: Ano

7. Příklad

Nabízíme Vám zdrojový kód prázdné kostry pro dynamickou knihovnu používající toto rozhraní. Kód je v jazyce C++ a je testován ve vývojovém prostředí Microsoft Visual Studio 2012.

7.1 Jak příklad zkompilovat

Tato krátká kapitola popisuje kroky vedoucí nutné ke kompilaci zdrojového kódu z kapitoly 7.2 v prostředí Microsoft Visual Studio 2012. Jednoduše následujte instrukce uvedené níže.

- ❑ Spusťte Microsoft Visual Studio 2012.
- ❑ Z nabídky File zvolte New a poté vyberte příkaz Project. Z kategorií projektů vyberte Win32 v sekci Visual C++. Z instalovaných šablon zvolte Win32 Project. Zvolte umístění pro nový projekt a poté vepište jméno projektu a stiskněte tlačítko OK. V následujícím průvodci zvolte volbu „DLL“ a stiskněte tlačítko Finish.
- ❑ Je vytvořen projekt, který obsahuje soubor dllmain.cpp.
- ❑ Zkopírujte zdrojový kód z kapitoly 7.2. do souboru dllmain.cpp.
- ❑ Z nabídky Project zvolte příkaz projectName Properties (Alt + F7). Vyberte sekci Configuration Properties/General a nastavte parametr „Character Set“ na hodnotu „Not Set“. Nyní vyberte sekci Configuration Properties/C/C++/Preprocessor a přidejte do Preprocessor Definitions hodnotu „_CRT_SECURE_NO_WARNINGS“.
- ❑ Z nabídky Build zvolte příkaz Build Solution (F7).

7.2 Zdrojový kód

```
#include "stdafx.h"
#include <stdio.h>
#include <stdlib.h>
#include <windows.h>

typedef BOOL (__cdecl *CRS2CRS)(int, int, double*, double*, double*);

typedef struct _EXTENT { // rozsah dat
    double xMin;
    double yMin;
    double xMax;
    double yMax;
    double zMin;
    double zMax;
    double mMin;
    double mMax;
} EXTENT;

// mapové jednotky
enum {
    map_units_unknown = 40244,
    map_units_kilometers = 40245,
    map_units_meters = 40246,
    map_units_decimeters = 40253,
    map_units_centimeters = 40247,
    map_units_millimeters = 40248,
    map_units_inches = 40251,
    map_units_feet = 40250,
    map_units_yards = 40254,
    map_units_statute_miles = 40249,
    map_units_nautical_miles = 40255,
    map_units_fathoms = 40256,
    map_units_chains = 40257,
    map_units_links = 40258,
    map_units_us_surveyors_inches = 40259,
    map_units_us_surveyors_feet = 40260,
    map_units_us_surveyors_yards = 40261,
    map_units_us_surveyors_chains = 40262,
    map_units_us_surveyors_statute_miles = 40263,
```

```

map_units_indian_feet = 40264,
map_units_indian_yards = 40265,
map_units_indian_chains = 40266,
map_units_decimal_degrees = 40252
};

// proměnné pro každé vlákno
typedef struct _THREADPARAMS {
    TCHAR          m_buffer[_MAX_PATH];
    BOOL           bUseTransparentColor;
    COLORREF       transparentColor;
    int            transparency;
    COLORREF*      colorChanges;
    DWORD          colorChangesCount;
    EXTENT         extent;
} THREADPARAMS;

HINSTANCE hInstanceDll;
static DWORD dwTlsIndex; // adresa sdílené paměti

BOOL WINAPI DllMain(HINSTANCE hInstDll, DWORD fdwReason, LPVOID lpReserved)
{
    LPVOID lpvData;
    // Podle důvodu volání proveďte patřičnou akci.
    switch(fdwReason)
    {
        case DLL_PROCESS_ATTACH:
            // Inicializace pro každý nový proces.
            // Vrací FALSE pokud natažení DLL selže.

            // Uložte si popisovač instance knihovny
            hInstanceDll = hInstDll;

            // alokují TLS index
            if ((dwTlsIndex = TlsAlloc()) == TLS_OUT_OF_INDEXES)
                return FALSE;
            // žádný break: Inicializují index pro první vlákno

        case DLL_THREAD_ATTACH:
            // Inicializace pro vlákno.
            // Inicializují TLS index pro toto vlákno
            lpvData = (LPVOID)new THREADPARAMS;
            if (lpvData == NULL) return FALSE;
            if (!TlsSetValue(dwTlsIndex, lpvData)) {
                delete (THREADPARAMS*)lpvData;
                return FALSE;
            }
            break;

        case DLL_THREAD_DETACH:
            // Úklid pro vlákno.
            // Uvolním alokovanou paměť pro toto vlákno
            lpvData = TlsGetValue(dwTlsIndex);
            if (lpvData != NULL) {
                THREADPARAMS* threadParams = (THREADPARAMS*)lpvData;
                if (threadParams) delete threadParams;
            }
            break;

        case DLL_PROCESS_DETACH:
            // Úklid pro proces.
            lpvData = TlsGetValue(dwTlsIndex);
            if (lpvData != NULL) {
                THREADPARAMS* threadParams = (THREADPARAMS*)lpvData;
                if (threadParams) delete threadParams;
            }
            // Uvolním TLS index.
            TlsFree(dwTlsIndex);
            break;
    }
    return TRUE;
}

// Funkce vrací ukazatel na parametry pro aktuální vlákno
THREADPARAMS* GetThreadParams()
{
    THREADPARAMS* threadParams = (THREADPARAMS*)TlsGetValue(dwTlsIndex);
    if (!threadParams) {
        threadParams = new THREADPARAMS;
    }
}

```



```

        if (!threadParams) return NULL;
        ZeroMemory(threadParams, sizeof(THREADPARAMS));
        if (!TlsSetValue(dwTlsIndex, (LPVOID)threadParams)) {
            delete threadParams;
            return NULL;
        }
    }
    return threadParams;
}

extern "C" __declspec( dllexport ) BOOL __cdecl BuildPyramid(TCHAR* fileName)
{
    // Není podporováno, nebo funkce selže
    return FALSE;

    // Vytvořte pyramidu
    // ...
    // zde vložte Váš kód
    // ...
    // pyramida byla úspěšně vytvořena
    // return TRUE;
}

extern "C" __declspec( dllexport ) void __cdecl Draw(TCHAR* fileName, HWND hWndMainFrame, HDC hDC,
EXTENT* imgExtent, EXTENT* drawExtent, RECT* clipRect, CRS2CRS Crs2Crs, DWORD* pId1, volatile DWORD*
pId2)
{
    // Funkce běží dokud *pId1 je rovno *pId2
    if (*pId1 != *pId2) return;

    SetStretchBltMode(hDC, COLORONCOLOR);

    THREADPARAMS* threadParams = GetThreadParams();
    if (!threadParams) return;

    // ...
    // zde vložte Váš kód
    // ...
    // Prosím nezapomeňte na hodnotu pId2. Kontrolujte tuto hodnotu
    // tak často jak je to možné a rozumné. Pokud je hodnota změněna
    // okamžitě uvolněte paměť a opusťte funkci.
}

extern "C" __declspec( dllexport ) BOOL __cdecl Export(TCHAR* outFileFileName, HWND hWndMainFrame,
BITMAP* pBitmap, EXTENT* exportedExtent, UINT mapUnits, BOOL bCreateWorldFile)
{
    // Pokud není implementováno nebo funkce selže vraťte FALSE,
    // jinak vraťte TRUE

    // Převeďte bitmapu do Vašeho formátu a uložte ji do souboru
    // ...
    // zde vložte Váš kód
    // ...

    // vytvořte souřadnicový soubor
    if (bCreateWorldFile) {
        // jméno souboru
        TCHAR worldFN[_MAX_PATH];
        lstrcpy(worldFN, outFileFileName);
        if (TCHAR* pDot = strrchr(worldFN, '.')) pDot[0] = '\\0';
        lstrcat(worldFN, ".xxw");
        // vytvořím soubor
        if (FILE* worldFile = fopen(worldFN, "wt")) {
            TCHAR buffer[256];
            sprintf(buffer, "%.14f\\n",
(exportedExtent->xMax - exportedExtent->xMin)/(double)pBitmap->bmWidth);
            fwrite(buffer, lstrlen(buffer), 1, worldFile);
            sprintf(buffer, "%.14f\\n", 0.0);
            fwrite(buffer, lstrlen(buffer), 1, worldFile);
            fwrite(buffer, lstrlen(buffer), 1, worldFile);
            sprintf(buffer, "%.14f\\n",
(-1)*((exportedExtent->yMax - exportedExtent->yMin)/(double)pBitmap->bmHeight));
            fwrite(buffer, lstrlen(buffer), 1, worldFile);
            sprintf(buffer, "%.14f\\n", exportedExtent->xMin);
            fwrite(buffer, lstrlen(buffer), 1, worldFile);
            sprintf(buffer, "%.14f\\n", exportedExtent->yMax);

```

```

        fwrite(buffer, lstrlen(buffer), 1, worldFile);
        fclose(worldFile);
    }
}
return TRUE;
}

extern "C" __declspec( dllexport ) DWORD __cdecl GetColorDeep(TCHAR* fileName)
{
    DWORD colorDeep = 0;

    // Zjistěte barevnou hloubku
    // ...
    // zde vložte Váš kód
    // ...

    return colorDeep;
}

extern "C" __declspec( dllexport ) TCHAR* __cdecl GetCompressionTypeString(TCHAR* fileName)
{
    THREADPARAMS* threadParams = GetThreadParams();
    if (!threadParams) return NULL;
    lstrcpy(threadParams->m_buffer, "žádná");

    // Pokud jsou data komprimována vyplňte threadParams->m_buffer
    // jménem použitého typu komprese
    // ...
    // zde vložte Váš kód
    // ...

    return threadParams->m_buffer;
}

extern "C" __declspec( dllexport ) EXTENT* __cdecl GetExtent(TCHAR* fileName)
{
    THREADPARAMS* threadParams = GetThreadParams();
    if (!threadParams) return NULL;
    memset(&(threadParams->extent), 0, sizeof(EXTENT));

    // Naplňte tyto členy struktury extent:
    // xMin, yMin, xMax a yMax
    // ...
    // zde vložte Váš kód
    // ...

    return &(threadParams->extent);
}

extern "C" __declspec( dllexport ) TCHAR* __cdecl GetFormatName(TCHAR* fileName)
{
    THREADPARAMS* threadParams = GetThreadParams();
    if (!threadParams) return NULL;
    threadParams->m_buffer[0] = '\\0';

    // Naplňte threadParams->m_buffer názvem rastrového formátu
    // ...
    // zde vložte Váš kód
    // ...

    return threadParams->m_buffer;
}

extern "C" __declspec( dllexport ) TCHAR* __cdecl GetFormatVersion(TCHAR* fileName)
{
    THREADPARAMS* threadParams = GetThreadParams();
    if (!threadParams) return NULL;
    threadParams->m_buffer[0] = '\\0';

    // Naplňte threadParams->m_buffer verzí rastrového formátu
    // ...
    // zde vložte Váš kód
    // ...

    return threadParams->m_buffer;
}

```

```

extern "C" __declspec( dllexport ) DWORD __cdecl GetHeight(TCHAR* fileName)
{
    DWORD height = 0;

    // Do proměnné height uložte výšku rastrových dat v pixelech
    // ...
    // zde vložte Váš kód
    // ...

    return height;
}

extern "C" __declspec( dllexport ) DWORD __cdecl GetInterfaceVersion()
{
    // vraťte verzi rozhraní (1.2)
    return MAKELONG(2, 1);
}

extern "C" __declspec( dllexport ) DWORD __cdecl GetWidth(TCHAR* fileName)
{
    DWORD width = 0;

    // Do proměnné width uložte šířku rastrových dat v pixelech
    // ...
    // zde vložte Váš kód
    // ...

    return width;
}

extern "C" __declspec( dllexport ) TCHAR* __cdecl GetWorldFileName(TCHAR* fileName)
{
    // Pokud rastrová data nemají žádný souřadnicový soubor vraťte NULL
    // return NULL;

    THREADPARAMS* threadParams = GetThreadParams();
    if (!threadParams) return NULL;
    threadParams->m_buffer[0] = '\\0';

    // Pokud rastrová data mají souřadnicový soubor naplňte
    // threadParams->m_buffer celým jménem tohoto souboru
    // ...
    // zde vložte Váš kód
    // ...

    return threadParams->m_buffer;
}

extern "C" __declspec( dllexport ) BOOL __cdecl HasPyramid(TCHAR* fileName)
{
    // Pokud není podporováno, nebo pyramida není vytvořena vraťte FALSE.
    // Pokud je pyramida vytvořena vraťte TRUE.
    // ...
    // zde vložte Váš kód
    // ...

    return FALSE;
}

extern "C" __declspec( dllexport ) void __cdecl RemovePyramid(TCHAR* fileName)
{
    // ...
    // zde vložte Váš kód
    // ...
}

extern "C" __declspec( dllexport ) void __cdecl SetColorChanging(DWORD nCount, COLORREF* colors)
{
    THREADPARAMS* threadParams = GetThreadParams();
    if (!threadParams) return;
    // Nastavení záměny barev
    threadParams->colorChanges = colors;
    threadParams->colorChangesCount = nCount;
}

extern "C" __declspec( dllexport ) void __cdecl SetTransparency(DWORD transparencyPercentage)
{

```

```
THREADPARAMS* threadParams = GetThreadParams();
if (!threadParams) return;
// Nastavení průsvitnosti v procentech.
if (transparencyPercentage < 0) transparencyPercentage = 0;
if (transparencyPercentage > 100) transparencyPercentage = 100;
threadParams->transparency = transparencyPercentage;
}

extern "C" __declspec( dllexport ) void __cdecl SetTransparentColor(COLORREF color)
{
    THREADPARAMS* threadParams = GetThreadParams();
    if (!threadParams) return;
    // Nastavení průhledné barvy.
    threadParams->transparentColor = color;
}

extern "C" __declspec( dllexport ) BOOL __cdecl SupportsColorChanging()
{
    // Pokud není podporováno vraťte FALSE, jinak vraťte TRUE
    return FALSE;
}

extern "C" __declspec( dllexport ) BOOL __cdecl SupportsExport()
{
    // Pokud není podporováno vraťte FALSE, jinak vraťte TRUE
    return FALSE;
}

extern "C" __declspec( dllexport ) BOOL __cdecl SupportsTransparency()
{
    // Pokud není podporováno vraťte FALSE, jinak vraťte TRUE
    return FALSE;
}

extern "C" __declspec( dllexport ) BOOL __cdecl SupportsTransparentColor()
{
    // Pokud není podporováno vraťte FALSE, jinak vraťte TRUE
    return FALSE;
}

extern "C" __declspec( dllexport ) BOOL __cdecl SupportsPyramids()
{
    // Pokud není podporováno vraťte FALSE, jinak vraťte TRUE
    return FALSE;
}

extern "C" __declspec( dllexport ) void __cdecl UseTransparentColor(BOOL bUse)
{
    THREADPARAMS* threadParams = GetThreadParams();
    if (!threadParams) return;
    threadParams->bUseTransparentColor = bUse;
}
```