RString

SPRÁVCE RESOURCE STRINGŮ A JAZYKOVÝCH VARIANT APLIKACE

Příručka uživatele a programátora

***SofCon*®spol. s r.o.**
Střešovická 49
162 00 Praha 6
tel/fax: +420 220 180 454
E-mail: sofcon@sofcon.cz
www: http://www.sofcon.cz

Verze dokumentu 1.00

Informace v tomto dokumentu byly pečlivě zkontrolovány a SofCon věří, že jsou spolehlivé, přesto SofCon nenese odpovědnost za případné nepřesnosti nebo nesprávnosti zde uvedených informací.
SofCon negarantuje bezchybnost tohoto dokumentu ani programového vybavení, které je v tomto dokumentu popsané. Uživatel přebírá informace z tohoto dokumentu a odpovídající programové vybavení ve stavu, jak byly vytvořeny a sám je povinen provést validaci bezchybnosti produktu, který s použitím zde popsaného programového vybavení vytvořil.
SofCon si vyhrazuje právo změny obsahu tohoto dokumentu bez předchozího oznámení a nenese žádnou odpovědnost za důsledky, které z toho mohou vyplynout pro uživatele.

Datum vydání: 21.01.2004
Datum posledního uložení dokumentu: 21.01.2004
(Datum vydání a posledního uložení dokumentu musí být stejné)

Obsah :

1. O dokumentu

1.1. Revize dokumentu

Verze dokumentu	Verze SW	Autor	Datum vydání	Popis změn
1.00	1.00	Cr	21.01.2004	Úprava dokumentu dle ISO9000

1.2. Účel dokumentu

Tento dokument slouží jako popis knihovny RString, která je součástí systémových knihoven firmy SofCon.

1.3. Rozsah platnosti

Určen pro programátory a uživatele programového vybavení SofCon.

1.4. Související dokumenty

Pro čtení tohoto dokumentu není potřeba číst žádný další manuál, ale je potřeba rientovat se v používání programového vybavení SofCon.

Popis formátu verze knihovny a souvisejících funkcí je popsán v manuálu LibVer.

2. Termíny a definice

Používané termíny a definice jsou popsány v samostatném dokumentu Termíny a definice.

3. Úvod

3.1. Účel knihovny RString

Knihovna RString je pomocná knihovna pro práci s lokalizovatelnými resource stringy. Knihovna podporuje vytváření vícejazyčných aplikací s možností přepínání jazyků přímo za chodu aplikace.

Resource string je textový řetězec, ke kterému existuje jednoznačný identifikátor. Lokalizovaná aplikace se neodkazuje přímo na tento textový řetězec nýbrž na jeho identifikátor. Resource stringy jsou udržovány v tabulkách a přístup k nim je možný pouze pomocí jejich identifikátoru.

Aby aplikace mohla využít tuto knihovnu je potřeba nadefinovat identifikátory potřebných textových řetězce (viz. kapitola 3.2.1) a vytvořit tabulky textových řetězcu alespoň pro jednu jazykovou mutaci aplikace. Tato tabulka je implementovaná pomocí funkce s prototypem **TRSGetter** (viz. kapitola 3.3.1), kterou je nutné zaregistrovat pomocí funkce **RSRegister** (viz. kapitola 3.4.11).

Mezi jednotlivými jazyky lze za chodu aplikace přepínat pomocí funkce **RSSetLanguage** (viz. kapitola 3.4.9).

Aplikace může za běhu využívat až 16 variant lokalizovaných řetězců.

3.2. Konstanty

3.2.1. Konstanty sid_ a _rsp

Konstanty s prefixem sid_ a _rsp jednoznačně identifikují resource string. Používání těchto prefixů je pouze doporučení. V této knihovně nejsou definovány žádné konstanty s tímto prefixem.

např.

```
const
  sidDeviceParameters    = 1;
  sidTemperature         = 2;
```

Hodnota konstant sid_ může být v rozsahu 1 až 65279 ($FEFF). Hodnoty 65280 až 65535 jsou rezervovány pro účely vizualizačních knihoven.

Ke každé konstantě s prefixem sid_ je možné definovat konstantu s prefixem rsp_. Konstanta s prefixem _rsp je speciální ukazatel, který má v segmentové části vždy hodnotu 0 a v ofsetové části hodnotu identifikátor resource stringu sid_. Konstantu s tímto prefixem je možné použít při volání funkce s parametrem **PRString** nebo **TRString**.

např.

```
const
  rspDeviceParameters   = PRString( Ptr( 0, sidDeviceParameters ) );
  rspTemperature        = PRString( Ptr( 0, sidTemperature ) );
```

Knihovny s konstantami sid_ a rsp_ lze automatizovaně vytvářet pomocí aplikace RSCreator.

3.2.2. Konstanty rsl_

Konstanty rsl_ jednoznačně identifikují jazyk resource stringu. V knihovně RString jsou definovány následující konstanty. V případě potřeby dalšího jazyka, neuvedeného v tabulce níže, je možné vytvořit další identifikátor rsl_. Hodnoty identifikátorů mohou být v rozsahu 0 až 15.

Identifikátor	Kód	Popis
rslCZ	0	Čeština
rslEN	1	Angličtina
rslDE	2	Němčina
rslRU	3	Ruština
rslPL	4	Polština
rslSK	5	Slovenština
rslHU	6	Maďarština

3.3. Typy

3.3.1. TRSGetter

Procedurální typ **TRSGetter** definuje prototyp funkce vracející ukazatel na resource string podle identifikátoru.

```
TRSGetter = function( Id: Word ): PString;
```

K vytvoření funkce s tímto prototypem, lze použít aplikaci pro tvorbu resource stringů RSCreator, nebo lze tuto funkci vytvořit ručně, např.:

```
function AppResCZ( Id: Word ): PString; far;
begin
  case Id of
    sidDeviceParameters: MyStrings := StrPtr( 'Parametry zařízení' );
    sidTemperature     : MyStrings := StrPtr( 'Teplota' );
  else
    MyStrings := nil;
  end;
end;


function AppResEN( Id: Word ): PString; far;
begin
```

```
  case Id of
    sidDeviceParameters: MyStrings := StrPtr( 'Device parameters' );
    sidTemperature     : MyStrings := StrPtr( 'Temperature' );
  else
    MyStrings := nil;
  end;
end;
```

Takto vytvořené funkce je potřeba zaregistrovat pomocí funkce **RSRegister** (viz. kapitola 3.4.11).

Např.

```
RSRegister( rslCZ, AppResCZ );
RSRegister( rslEN, AppResEN );
```

3.3.2. TRString

Typ **TRString** je typ používaný v parametrech funkcí, které umožňují zpracovat odkaz na resource string. Typ **TRString** je definován následovně:

```
PRString =  ^TRString;
TRString = string;
```

3.4. Funkce

3.4.1. Funkce RS

Funkce **RS** provádí konverzi identifikátoru resource string (sid_) na ukazatel na resource string (rsp).

```
function RS( Id: Word ): PRString;
```

Parametry:

Id	Identifikátor resource stringu (konstanta s prefixem sid_).

Návratové hodnoty:

Funkce vrací hodnotu ekvivalentní výrazu PRString(Ptr(0, Id)).

3.4.2. Funkce RSToStr

Funkce **RSToStr** převádí ukazatel na resource string na ukazatel na obyčejný string.

```
function RSToStr( P: PRString ): PString;
```

Parametry:

P Ukazatel na string nebo resource string.

Návratové hodnoty:

Pokud je parametrem funkce ukazatel na string, pak funkce vrací tento ukazatel nezměněn. Pokud je parametrem funkce ukazatel na resource string, pak funkce vrací ukazatel na textový řetězec odpovídající tomuto ukazateli v aktuálně nastaveném jazyce.

3.4.3. Funcke IsRS

Funkce **IsRS** zjišťuje zda předaný ukazatel je typu string nebo resource string.

```
function IsRS( P: PRString ): Boolean;
```

Parametry:

P Ukazatel na string nebo resource string.

Návratové hodnoty:

Funkce vrací hodnotu True, pokud je předaný ukazatel typu resource string.

Poznámky:

Hodnota funkce odpovídá výrazu Seg(P^) = 0.

3.4.4. Funkce RSAlloc

Funkce **RSAlloc** vytvoří kopii textového řetězce na hromadě a vrátí na ni ukazatel.

```
function RSAlloc( const S: string ): PRString;
```

Parametry:

S Textový řetězec.

Návratové hodnoty:

Funkce vrací ukazatel na textový řetězec alokovaný na hromadě.

Poznámky:

Takto alokovaný řetězec lze z hromady uvolnit pomocí funkce **RSFree** (viz. kapitola 3.4.5).

3.4.5. Procedura RSFree

Procedura **RSFree** uvolní z hromady kopii řetězce alokovanou pomocí funkce **RSAlloc**, **RSCopy** nebo **RSDeepCopy**.

```
procedure RSFree( P: PRString );
```

Parametry:

P	Ukazatel na string nebo resource string..

Poznámky:

Funkce zjišťuje, zda řetězec není typu resource string, tj. zda segmentová část předaného ukazatele je nenulová. Pokud se jedná o resource string, pak procedura neprovádí žádné uvolnění.

3.4.6. Funkce RSCopy

Funkce **RSCopy** vytváří kopii textového řetězce.

```
function RSCopy( P: PRString ): PRString;
```

Parametry:

P	Ukazatel na string nebo resource string..

Návratové hodnoty:

Pokud předaný ukazatel je typu string, pak funkce vrátí ukazatel na string alokovaný na heapu. Pokud je předaný ukazatel typu resource string, funkce vrátí předaný ukazatel.

Poznámky:

Funkce alokuje prostor na heapu pouze tehdy, jestliže parametr P obsahuje ukazatel na obyčejný string.

3.4.7. Funkce RSDeepCopy

Funkce **RSDeepCopy** vytváří kopii textového řetězce.

```
function RSDeepCopy( P: PRString ): PRString;
```

Parametry:

P	Ukazatel na string nebo resource string..

Návratové hodnoty:

Funkce vrátí ukazatel na string alokovaný na heapu.

Poznámky:

Na rozdíl od funkce RSCopy, funkce RSDeepCopy vždy alokuje na heapu kopii řetězce bez ohledu na to, zda se jedná o obyčejný string nebo resource string.

3.4.8. Funkce StrPtr

Funkce **StrPtr** je pomocná inline funkce, vracející ukazatel na textovou konstantu.

```
function StrPtr( const S: String ): PString;
```

Parametry:

S	Textová konstanta.

Návratové hodnoty:

Funkce vrátí ukazatel na textovou konstantu, tj. ukazatel do kódového segmentu.

3.4.9. Procedura RSSetLanguage

Procedura **RSSetLanguage** slouží k nastavení aktuálního jazyka aplikace.

```
procedure RSSetLanguage( ALanguage: Integer );
```

Parametry:

ALanguage	Identifikátor jazyka, tj. jedna z konstant rsl_ (viz. kapitola 3.2.2).

Poznámky:

Nastavení aktuálního jazyka ovlivňuje chování funkcí **RSDeepCopy** a **RSToStr**.

3.4.10. Funkce RSGetLanguage

Funkce **RSGetLanguage** vrací identifikátor aktuálně nastaveného jazyka aplikace.

```
function RSGetLanguage: Integer;
```

Parametry:

Funkce nemá žádné parametry.

Návratové hodnoty:

Funkce vrací jednu z konstant rsl_ (viz. kapitola 3.2.2).

3.4.11. Funkce RSRegister

Procedura **RSRegister** slouží k zaregistrování tabulky resource stringů.

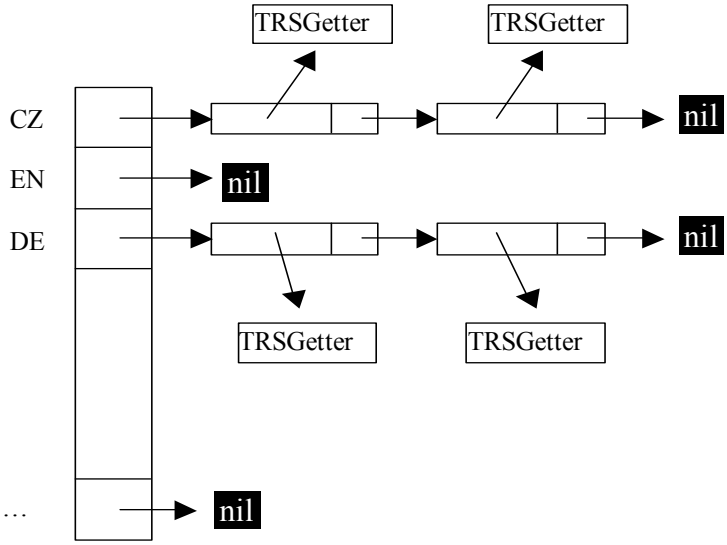
```
procedure RSRegister( ALanguage: Integer; AGetter: TRSGetter );
```

Parametry:

ALanguage	Identifikátor jazyka, tj. jedna z konstant rsl_ (viz. kapitola 3.2.2).
AGetter	Funkce vracející ukazatel na textové řetězce odpovídající příslušných identifikátorům resource stringu v jazyce specifikovaném parametrem ALanguage.

Poznámky:

V rámci jednoho jazyka lze zaregistrovat více tabulek resource stringů pomocí opakovaného volání této procedury. Zaregistrované tabulky (resp. funkce vracející resource string) jsou uloženy ve spojovém seznamu jak ilustruje následující obrázek. Pro každou jazykovou variantu existuje vlastní spojový seznamu.



3.4.12. Funkce RSUnregister

Procedura **RSUnregister** slouží k odregistrování tabulky resource stringů dříve zaregistrované pomocí procedury **RSRegister**.

```
procedure RSUnRegister( ALanguage: Integer; AGetter: TRSGetter );
```

Parametry:

ALanguage	Identifikátor jazyka, tj. jedna z konstant rsl_ (viz. kapitola 3.2.2).
AGetter	Odkaz na funkci předaný při volání metody **RSRegister**.