

ScriptS++ 1.2 Specification for dev120_180_03262008

Obsah

ScriptS++ 1.2 Specification for dev120_172_03262008.....	1
Úvod.....	2
Novinky ve verzi 1.2.....	2
Syntaxe.....	3
Komentáře.....	3
Hlavička.....	3
Namespace (NS).....	3
Proměnné.....	4
Funkce.....	5
Základní kostra kódu.....	7
Vytváření prvků.....	8
Objekty a procedury.....	8
Komponenty.....	9
Standartní komponenty.....	9
.....	9
.....	9
Speciální komponenty.....	12
Použití templates.....	12
Úvod do templates.....	12
Vytvoření templaty.....	12
Použití v našem kódu.....	12
Použití includes.....	12
Speciální možnosti jazyka.....	12
Příklady skriptů.....	12
Rejstřík.....	12

Úvod

Jazyk ScriptS++ byl vytvořen pro jednodušší tvorbu aplikací s GUI. Sám o sobě NENÍ určen pro programování nových aplikací. Obsahuje též pár vestavěných funkcí, které se budou rozrůstat.

Původní motivací, proč jsem napsal tento jazyk, byla zkouška toho co již umím. Vlastně napsat vlastní jazyk byl můj sen již od mého programátorského počátku. Proti původním představám se syntaxe jazyka změnila velmi mnoho a postupně se zjednodušovala tak jak jsem zjišťoval že jde o zbytečnou složitost či nevím jak bych to napsal. Vlastně jsem zjistil že navrhnout jazyk není nic jednoduchého a může to být těžší i než jen jeho implementace. Postupně jsem zjistil že nevím jak vytvořit binární soubor, tak jsem zvolil nejjednodušší variantu a to skript. Tento jazyk je navržěn tak aby vyhovoval mě, což znamená že se ostatním nemusí líbit.

Mezi hlavní výhody tohoto jazyka patří jeho jednoduchost a nenáročnost. Na začátku své existence byl jen návrhářem formulářů ale neumožňoval pracovat s procedurami, volat akce atd. Jediné co uměl vykreslit okno tak jak bylo napsáno ve zdrojovém souboru. Postupně ale přibýly možnosti jak ovlivnit okno již za jeho běhu. Nová verze přináší spoustu novinek které opět rozšiřují jeho možnosti.

Novinky ve verzi 1.2

- Nová upravená kostra kódu
- Uzamčení do **namespace**
- Práce s proměnnými a podpora datových typů
- Nové funkce a jejich používání
- Nové knihovny funkcí **\$system** a **\$math**
- Nové komponenty
- Vylepšení tlačítka **SPP_START**
- Interní proměnné **\$script** a **\$application**
- Revize a zjednodušení syntaxe scriptu
- Podpora pro *templates*

Syntaxe

Tento programovací jazyk je určen pro psaní „okének“, proto snahou autora je co nejvíce jednoduchá a snadno použitelná. Program obsahuje možnost využít i speciálních komponent, které jsou složeny z několika prvků tzv templates.

Komentáře

Tento jazyk podporuje dva druhy komentářů:

- jednořádkový uvozený znakem *****
- více řádkový začínající **/*** a končící ***/**

Hlavička

Každý script musí být uvozen hlavičkou. Před hlavičkou smí být jen komentáře. Pomocí hlavičky pozná kompilátor jak má daný kód interpretovat a zda jej podporuje.

```
#PROJECT REF="SPP_SCRIPT" VERSION="1.2"
```

#PROJECT	udává počátek hlavičky	
REF	Varianta kompilátoru	
	<i>SPP_SCRIPT_V1</i>	starší verze, není doporučena
	<i>SPP_SCRIPT</i>	nová DOPORUČUJI
VER	Verze kompilátoru	
	<i>1.0</i>	<i>v současnosti nejstarší, sloužila jen jako ukázka</i>
	<i>1.1</i>	<i>již funkční verze, její nevýhodou byla ne mnoho funkcí</i>
	<i>1.1.1</i>	<i>oprava předchozí verze, přidání includes</i>
	<i>1.2</i>	<i>současná verze !POZOR stále jen development versioon !</i>

Namespace (NS)

V překladu jmený prostor. NS se musí shodovat s názvem souboru bez přípony tj. pokud máme například soubor **sample.sso** tak se NS musí jmenovat **sample**. Tato část je v nové verzi mírně upravená proti všem předchozím. Začátek jmeného prostoru se píše za hlavičku skriptu a existují dva druhy NS:

script new_project; *nebo* **tempalate** new_project;

script určuje jmený prostor skriptu

template určuje jmený prostor šablony (o jejím využití později). Ty nejsou spustitelné!

Pro ukončení NS se před daný typ píše znak **!** Ovšem bez názvu prostoru př.:

!script *nebo* **!tempalate**

Pokud použijete template nezapomeňte prosím že nelze samostatně bez vložení do skriptu spustit!

Proměnné

Scriptovací jazyk podporuje dva typy proměnných systémové a uživatelské. Ve starších verzích kompilátoru bylo nutné tyto typy předem určit slovy **var** a **sysvar**. Od této verze si kompilátor projede seznam systémových proměnných a pokud ji tam nenalezne nastaví danou proměnnou jako uživatelskou a není již tak nutné uvádět její typ. Díky tomu je i syntaxe zcela pozměněna, přesto však konstrukce zápisu zůstala podobná.

Syntaxe:

```
$VAR {  
    VARNAME = VALUE;  
}
```

Mezi významné novinky jistě patří i podpora DATOVÝCH typů, které píšeme za hodnotu proměnné:

```
VARNAME = VALUE as DATATYPE;
```

Podporované datové typy:

String	Textový řetězen
Integer	Celočíselný datový typ
Boolean	Booleanovská funkce True, False

Systémové proměnné v současnosti dostupné

__APPLICATIONTITLE__	Titulek programu na liště
__FORMCOLOR__	Výchozí barva formulářů
__FORM3DSTYLE__	Prostorový vzhled okna
__HINTTIME__	Doba zobrazení popisku
__HINTCOLORINT__	Barva popisku

Pozn. Speciálním typem sys. proměnné je __SYSMSG__ která zobrazí svůj obsah při spuštění skriptu

V programu existují i tzv. Interní proměnné (IP). Ty slouží jako výstup pro jednotlivé prvky přímo do vlastnosti. V současné době jsou podporovány IP: **\$script** a **\$application**.

Syntaxe: **\$script.vlastnost**;

Proměnná **\$script** – též lze zapsat jako **\$_NAMESPACE_** kde **_NAMESPACE_** je název jmeného prostoru skriptu - slouží k získání informací o skriptu. V současné době podporuje následující vlastnosti:

Namespace	Název jmeného prostoru
Filename	Plný název souboru skriptu
Cmp	Je skupinou informací o kompilátoru. Vlastnost „filename“ obsahuje úplnou cestu ke kompilátoru a vlastnost „version“ pak udává jeho verzi.

Promnná **\$application** obsahuje informace o samotném kompilátoru

Exename	Plná cesta k souboru kompilátoru
Title	Titulek na liště

Mezi proměnné též zahrnujeme i **\$system** která ověšem obsahuje přístup k funkcím kompilátoru a nahrazuje tak v současné době používaný znak **@** a zjednodušuje tak jejich použití protože existuje jako jednotné rozhraní jejich použití. O fukncích bude pojednávat následující kapitola této dokumentace.

Funkce

Jak již bylo zmíněno, skriptovací jazyk obsahuje i nějaké funkce, které nám zjednoduší práci. O některých, jako \$script a \$applications, již byla řeč v části Proměnné. V této kapitole si povíme o systémové proměnné \$system ve které se právě nachází rozhraní pro práci s funkcemi.

Syntaxe: `$system.__FUNCTION__` respektive `$system.__FUNCTION__.subfunction`

kde __FUNCTION__ je název funkce a je uveden spolu s krátkým popisem v záhlaví tabulky a v prvním sloupci s názvem podfunkce a druhém sloupci jeho popis či nějaké možnosti této funkce.

DATETIME – práce se datem a časem	
Now	Zobrazí datum a čas
Date	Zobrazí datum
Time	Zobrazí čas
WINVER – Informace o verzi Windows	
Name	Název verze Windows
Code	Kód verze Windows
Build	Označení verze windows
Info	Zobrazí další informace o verzi windows
ToString	Zobrazí název, kód a další informace o verzi Windows
HARDWARE – Informace o hw počítače	
-> MEMORY	
Phystotal	Celková fyzická pamět
Physnow	Fyzická pamět nyní
Virttotal	Celková virtuální pamět
Virnow	Virtuální pamět nyní
Pftotal	Celková stránkovací pamět
pfnnow	Stránkovací pamět nyní
Memload	Celková zabraná pamět
Memlen	Velikost paměti
-> CPU	
Count	Počet procesorů / jader
Type	Typ procesoru
SPPC – Informace kompilátoru ScriptS++	
Version	Informace o verzi kompilátoru
Develinfo	Interní označení verze kompilátor
Author	Autor komoilátoru
Name	Název kompilátoru

Kromě knihovny **\$system** je součástí jednotka **\$math** ve které jsou uloženy matematické funkce.

Syntaxe: **\$Math.__FUNCTION__** respektive **\$math.__FUNCTION__.subfunction**

ln(x)	Přirozený logaritmus z x
log(x,y)	Logaritmus y při základu x Možno zadat jen log(x) program použije základ 10
sqr(x,y)	Umocní x na y Možno zadat jen sqr(x) program umocní číslo na druhou
sqrt(x,y)	Y-tá Odmocnina z x Možno zadat jen sqrt(x) program odmocní číslo x druhou odmocninou
Const – sada konstant	
Random(x)	Vypíše náhodné číslo do x
Infinity	Nekonečno
!Infinity	- Nekonečno
Pi	Vypíše číslo Pi
E	Vypíše Eulerovo číslo
Trigonometric – sada goniometrických funkcí	
Sin(x)	Funkce sinus
Cos(x)	Funkce cosinus
Tan(x)	Funkce tangent
Cotg(x)	Funkce cotg
Sinh(x)	Hyberbolický sinus
Cosh(x)	Hyberbolický cosinus
Tanh(x)	Hyberbolický tan
Cotgh(x)	Hyberbolický cotg
Arcsin(x)	Arcus sinus
Arccos(x)	Arcus cosinus

Základní kostra kódu

Základní konstrukce programu obsahuje několik základních částí, které musí být obsaženy v kódu skriptu. Některé z nich jsme si již ukázaly, některé si ukážeme dalších kapitolách této dokumentace. Následuje kostra kódu programu:

```
#PROJECT REF="SPP_SCRIPT" VERSION="1.2"
```

```
script sample;
```

```
    $VAR {
```

```
    }
```

```
    $ITEMS {
```

```
        new TForm [Form1];
```

```
    }
```

```
    $OBJECTS {
```

```
    }
```

```
    $CODE {
```

```
    }
```

```
!script
```

\$VAR Místo kam se zadávají proměnné viz přechází kapitoly

\$ITEMS Zde vytváříme komponenty nebo nastavujeme použití z templates

\$OBJECTS Část zdrojového kódu kde se nastavuje vizuální podoba a chování komponent

\$CODE Zde zapisujeme funkční kód skriptu

Odstavec první této kapitoly však není docela přesný. Kontainer pro deklaraci proměnných \$var a funkční část skriptu \$code totiž pro minimální skript nejsou nutné. Proto minimální kód pro spuštění skriptu vypadá:

```
#PROJECT REF="SPP_SCRIPT" VERSION="1.2"
```

```
script sample;
```

```
    $ITEMS {
```

```
        new TForm [Form1];
```

```
    }
```

```
    $OBJECTS {
```

```
    }
```

```
!script
```

Vytváření prvků

Kostra kódu, kterou jsme si ukázaly o kapitulu více, je sice zajímavá ale bez prvků které by tam něco dělaly tam chybí. Ty musíme vytvořit pro což slouží kontainer **\$ITEMS** kam se vkládají konstrukтуры pro jednotlivé prvky (komponenty). Kontainer se zapisuje takto:

```
$ITEMS {  
}
```

Zápis konstruktorů je následující:

Syntaxe: **create Tcomponent [Name] ;**

nebo od verze 1.2 doporučené:

new Tcomponent [Name] ;

Tcomponent je název komponenty, kterou chceme vytvořit

Name název komponenty, který se bude používat pro práci s prvkem

Základním prvkem každého skriptu je komponenta Tform. Ta je povinou součástí každého skriptu a musí být přítomna v kódu.

Objekty a procedury

Comming soon

Komponenty

Bez komponent by aplikace byla jen prázdným oknem na v operačním systému. Jsou to stavební kameny každého programu. Kompilátor pro S++ jich obsahuje značné množství. Základní komponentou je Tform, samostatný formulář. Jeho vytvoření je popsáno v části Vytváření prvků. V této kapitole si popíšeme nastavení jeho vlastností a událostí.

Pro zápis vlastností a událostí komponent slouží kontainer **\$OBJECTS** tak se zapisují ve formátu:

```
COMPONENT_NAME {  
    property name=value;  
    event name=value;  
}
```

Některé komponenty mohou obsahovat i další funkce, o těch se budeme bavit vždy u jednotlivých komponent.

Příklad zápisu vlastností prvku Form1 popisující jeho titulek z názvu skriptu a formulář bude červený:

```
$OBJECTS {  
    Form1 {  
        property title=$script.filename;  
        property color=clred;  
    }  
}
```

Standartní komponenty

Společné vlastnosti

Top	Horní pozice prvku
Left	Pozice okna odleva
Width	Šířka prvku
Height	Výška prvku
Hint	Popisek prvku
+Hint	Přidá k současněmu popisku
Showhint	Zobrazit prvek (true, false)
Enable	Dostupnost prvku (true, false)
Visible	Viditelnost prvku (true, false)

Společné události

OnClick	Kliknutí na prvek
OnDbClick	Dvojitě kliknutí na prvek
OnMouseDown	Při stisknutí tlačítka myši dolů
OnMouseUp	Při stisknutí tlačítka myši nahoru
OnMouseMove	Při pohybu myši
OnMouseEnter	Při vstupu kurzoru na prvek
OnMouseLeave	Při opuštění prvku kurzorem

Tform

Prvek typu formulář je základní prvkem v programu.

Vlastnosti:

title	Horní titulek okna
+title	Přidá k současnému titulku
color	Barva okna
color.i	Barva zadaná jako číslo
font.color	Barva písma
font.color.i	Barva zadaná jako číslo
font.name	Název písma
font.size	Velikost písma
autosize	Automatická velikost okna dle rozložení prvků
autoscroll	Zobrazí scrollbarů pokud je nějaký prvek umístěn mimo rozměry okna
transparent	Průhlednost okna (true, false)
transparentcolor	Barva, která bude průhledná
center	Nastaví okno doprostřed
alpha	Průhlednost celého okna číslo 0..255 kde 255 je zcela neprůhledný
type	Určuje typ okna (ovládací prvky) fsnone ... žádné okraje fsdialog ... jen základní prvky fstoolwin ... nástrojové okno fssizetoolwin ... nástrojové okno s změnou rozměrů fssingle



Tlabel

Prvek Tlabel je textový popisek.

Vlastnosti:

title	Horní titulek okna
+title	Přidá k současnému titulku
color	Barva okna
color.i	Barva zadaná jako číslo
font.color	Barva písma
font.color.i	Barva zadaná jako číslo
font.name	Název písma
font.size	Velikost písma
autosize	Automatická velikost okna dle rozložení prvků

This is label

Speciální komponenty

Použití templates

Úvod do templates

Vytvoření templaty

Použití v našem kódu

Použití includes

Speciální možnosti jazyka

Příklady skriptů

Rejstřík