

Programátorské konvence .NET 2.0, 3.0 a 3.5

Upozornění

- **2007 Siemens IT Solutions and Services spol. s r.o., Praha, Česká republika**

Tento dokument obsahuje informace, které Siemens IT Solutions and Services spol. s r.o. považuje za součást svého obchodního tajemství ve smyslu §17 Obchodního zákoníku a je určen pouze pro vnitřní potřebu České správy sociálního zabezpečení. Jakékoliv zveřejnění v dokumentu obsažených informací, jejich předání třetí straně nebo využití pro potřebu subjektu, který není mateřskou, sesterskou nebo dceřinou společností Siemens Business Services spol. s r.o. bez jejího písemného souhlasu bude považováno za nekalou soutěž ve smyslu §51 Obchodního zákoníku. Osobní data použitá v dokumentu slouží k prokázání referencí, znalostí a zkušeností zaměstnanců Siemens Business Services a jsou určena výhradně pro potřebu tohoto projektu. Veškeré osobní údaje jsou uveřejněny se souhlasem zaměstnanců dle zákona č. 101/2000 Sb., o ochraně osobních údajů.

Obsah

PROGRAMÁTORSKÉ KONVENCE .NET 2.0, 3.0 A 3.5.....	1
OBSAH.....	2
ÚVOD	4
ZDROJOVÝ KÓD	4
Architektura	4
Projekty	4
Struktura projektů	5
Unit testy	5
Source Control	5
Struktura projektu v source control	6
Jmenné konvence pro subprojekty	6
Dokumenty	7
Jmenné konvence	7
Jmenné konvence pro proměnné	8
Jmenné konvence pro „Visual Controls“	8
Namespaces	8
Jmenné konvence pro assemblies	9
Jmenné prostory XML	9
Jmenné konvence pro aplikace	9
Obecná pravidla	9
Podporované programovací jazyky .NET	9
Kódování.....	9
Formátování položek.....	9
Proměnné	9
Konfigurace	10
GlobalsData	10
Značka Region	10
Desktopové aplikace (aplikace typu tlustý klient a smart klient)	10
Hlavní formulář	10
Desktopové aplikace bez GUI	11
Aplikace typu WindowsService.....	11
Webové aplikace	11
Webové služby	11
Ošetření výjimek.....	12
Databáze	13
Logování	13
Úrovně logování	14
Konfigurace	14

Kryptografické algoritmy	14
Komprimační algoritmy	15
Znovupoužitelné (sdílené) komponenty	15
Komponenty třetích stran	15
Dokumentace kódu	15
Příklad použití při vytváření kódu	16
Programátorská dokumentace	16
Verzování	16
Strong name	16
PŘÍLOHY	17
Seznam vyhrazených slov .NET	17
Doporučené pojmenování visual controls	17

Úvod

Tento dokument je určen pro programátory .NET. Jeho cílem je standardizovat kód, pro zvýšení kvality vytvářených aplikací, kódu, snadnější zaškolení uživatelů (jednotné prostředí), snadnější převzetí kódu, rychlou orientaci v kódu při vyhledávání chyb a řešení problémů.

Zdrojový kód

Konvence pro psaní zdrojového kódu vycházejí z

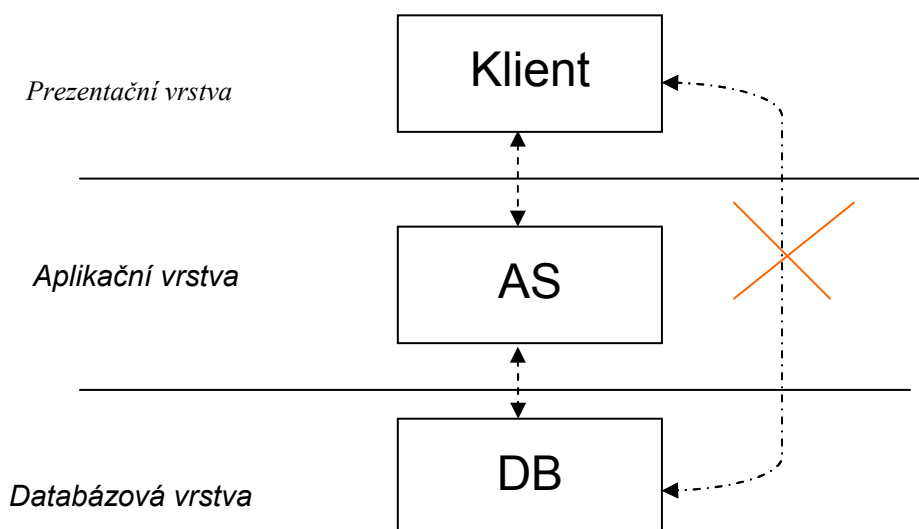
MSDN - Design Guidelines for Class Library Developers

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpgenref/html/cpconNETFrameworkDesignGuidelines.asp>

Zde jsou uvedeny ty nejdůležitější, detaily jsou uvedeny přímo v dokumentaci MSDN.

Architektura

Konvence jsou připraveny pro tří-vrstvou architekturu, s tlustým klientem na prezentační vrstvě. Veškeré systémy musí být vytvářeny tak, že aktuální vrstva smí komunikovat pouze se sousední vrstvou.



Projekty

Řešení je možné spravovat v jednom solution (slh) či ve více solutions, která se mohou i překrývat (jeden projekt může být v několika solutions). Pro předání musí být definována

- Solutions s unit testy
- Solutions bez unit testů a bez projektů, které se nekompilují (pouze produkční kód)

Pro jednotlivá solutions musí být popsáno vývojové prostředí pro úspěšné otevření všech souborů a kompilaci solution. Definovány musí být zejména

- Verze frameworku (.NET 2.0, 3.0, 3.5)
- Verze OS (Windows Server 2003, Windows XP, Vista,)
- SP level (Windows Server 2003 SP1, Windows XP SP2, ...)
- Jazyková verze (ENG, CZ) a regionální nastavení systému (ENG) a uživatele (CZ)
- Verze Visual Studia (Visual Studio 2005, 2008) – Visual Studio 2008 preferováno
- Řada Visual Studia
 - o Visual Studio Standard
 - o Visual Studio Professional
 - o Visual Studio Tools For Office
 - o Visual Studio Team Edition For Database Professionals
 - o Visual Studio Team Edition For Software Architects
 - o Visual Studio Team Edition For Software Developers
 - o Visual Studio Team Edition For Software Testers
 - o Visual Studio Team Suite
 - o Visual studio Team Suite
 - VSTS Architect edition
 - VSTS Developement edition
 - VSTS Test edition
 - VSTS Databáze edition
- SP level Visual Studia
- Seznam dalších požadovaných aplikací (SQL Server 2005 Developer, Visual Studio Team Foundation Server, BizTalk Server 2006 atd.)
- Seznam add-inů včetně přesného určení verzí

Použití jakýchkoliv add-inů pro vytváření musí být předem schváleno Odborem rozvoje a údržby IS ČSSZ.

Struktura projektů

Struktura odpovídá struktuře [Struktura projektu vsource control](#) od adresáře source níže.

Unit testy

Unit testy musí být spustitelné z Visual Studio Team Edition For Software Testers nebo Visual Studio 2008 professional.

Source Control

- Veškerý kód ukládaný do source control musí být sestavitelný (zbuildovatelný) a spustitelný
- Při odstranění starého kódu, je nutné před vložením do SourceSafe kód odstranit (ne pouze zakomentovat)

- Projekt se do source safe zakládá ihned po vytvoření
- Každý vývojář si před započítím práce zamkne objekty, které upravuje

Struktura projektu v source control

Projekty jsou do source control ukládány v následující struktuře:

- Jméno projektu
 - Source
 - Subprojekt
 - Ico
 - Resources
 - Package
 - Bin
 - Tools

Kde:

Adresář	Obsah
Source	Zdrojové kódy
Subprojekt	Dle pojmenování Namespace
<i>Podadresáře v adresáři source/subprojekt</i>	
Ico	Obrázky, ikony
Resources	xml, xsd, číselníky
Package	Dle pojmenování Namespace
Bin	Adresář pro umístění sdílených knihoven
Tools	Pomocné soubory (dávky pro import klíče, registrace komponent, instalace do GAC apod.)

Jmenné konvence pro subprojekty

Prefix	Význam
ws	Webová služba
ui	Uživatelské rozhraní
Lib	Knihovna
ca	Dávková aplikace
srv	Služba
Err	Chyby a výjimky

Sdílené knihovny jsou umísťovány do adresáře bin v kompilované podobě a **MUSÍ** být součástí projektu v source control tak, aby předávané zdrojové kódy byly zkompileovatelné bez nutnosti dohledávání komponent či dalších zdrojových kódů. Sdílené knihovny musí být referencovány jako kompilované soubory (dll) z adresáře BIN či GAC (Global Assembly Cache).

V předávací dokumentaci řešení musí být uvedeny přesné verze a kompletní informace o umístění zdrojového kódu sdílených komponent tak, aby bylo možné při předání ověřit dostupnost ev. zkompilevat reference a následně zkompilevat řešení.

Dokumenty

Dokumenty k jednotlivým projektům i k celému řešení musí být umístěny do složek Documents Team Foundation Serveru.

Jmenné konvence

Všechny objekty jsou pojmenovávány v angličtině, to znamená že veškerý kód, kromě komentářů, je v anglickém jazyce.

V žádném případě nesmí být v názvu samostatně použita vyhrazená slova pro .NET, Oracle, MS SqlServer!

Seznam vyhrazených slov .NET je v příloze tohoto dokumentu.

Pro pojmenovávání jsou používány tyto notace:

Jméno	Popis	Příklad
Pascal	První písmeno velké, ostatní malá, pokud obsahuje více slov další slovo navazuje bez mezery a začíná opět velkým písmenem	BackColor
camelCase	První slovo malými písmeny, pokud obsahuje více slov další slovo navazuje bez mezery a začíná opět velkým písmenem	getName
UPPERCASE	Všechna písmena velká, obvykle používáno u velmi krátkých názvů (jedno nebo dvou znakových)	IO

Identifikátor	Case	Příklad
Class	Pascal	AppDomain
Enum type	Pascal	ErrorLevel
Enum values	Pascal	FatalError
Event	Pascal	ValueChanged
Exception class	Pascal	WebException Poznámka: Vždy je ukončen sufixem Exception
Read-only static field	Pascal	RedValue
Interface	Pascal	IDisposable Poznámka: Vždy začíná prefixem I
Method	Pascal	ToString
Namespace	Pascal	System.Drawing
Parameter	Pascal	TypeName
Constants	Uppercase	TABLENAME

Jmenné konvence pro proměnné

Proměnné jsou pre-fixovány hodnotou dle typu proměnné:

Prefix	Typ proměnné	Příklad
	Globální (Pascal)	Color Poznámka: Proměnná musí být umístěna ve třídě GlobalsData.
m	Lokální v modulu nebo formuláři (camelCase)	mColor

Jmenné konvence pro „Visual Controls“

Visual controls jsou pojmenovávány dle notace „Camel Case“, před popisné jméno controlu je předřazen popis dle následující tabulky. Podřízené záznamy (například Menu Item v Menu Strip jsou pojmenovány dle nadřazeného + i + název nadřazeného **mnuiFileSave**). Výjimku z pojmenovávání tvoří control label – dle konvencí jsou pojmenovávány pouze ty, které jsou fyzicky používány (například se mění nadpis).

Doporučené konvence pro jednotlivé controls jsou v příloze tohoto dokumentu.

Inspiraci je možno nalézt zde:

<http://support.microsoft.com/kb/q110264/>

Namespaces

Každá třída musí být umístěna ve správném namespace. Namespaces se pojmenovávají dle následujícího klíče:

AMCSSZ.JmenoProduktu.Vrstva.package

AMCSSZ	Pevný text AMCSSZ
JmenoProjektu	Jméno projektu dle katalogu projektů <ul style="list-style-type: none"> Komponenty jsou umístěny v projektu Components
Vrstva	Ui – Klient aplikace
	Ws – webová služba
	Web - aplikace ASP .NET 2.0, 3.0, 3.5
	Ca – uživatelská aplikace bez GUI
	Srv – systémová aplikace bez GUI (služba)
	Lib – knihovna
	Err – chyby a výjimky
Package	Jméno package dle logických celků aplikace

Poznámka: Při vytváření projektu je potřeba vyplnit správný Root namespace.

Jmenné konvence pro assemblies

Jména assemblies dll knihoven musí vycházet z obdobného schématu jako namespace dané assembly:

AMCSSZ.JmenoProduktu.Vrstva.package

Například: **AMCSSZ.ProductName.Lib.InternalSchemas.dll**

Jmenné prostory XML

Jmenné prostory XML musí odpovídat následujícímu schématu

<http://schemas.cssz.cz/ProductName/ProductVersion/SchemaName/SchemaVersion>

Bližší viz Popis jmenných prostorů XML v dokumentu Standard metodiky vývoje.

Jmenné konvence pro aplikace

Aplikace (exe soubory) musí být pojmenovávány dle následujícího schématu:

AMCSSZ.JmenoProduktu

Obecná pravidla

- Kód musí splňovat doporučené zásady formulované jako množina pravidel pro statickou analýzu kódu. Statickou analýzu lze provádět pomocí nástrojů ve Visual Studio 2005 Team Developer nebo nástrojem Microsoft FxCop.

Podporované programovací jazyky .NET

- C#
- Visual Basic

Kódování

Veškeré vytvářené programy, musí pracovat v kódování **UTF-8**

Formátování položek

Je řešeno v datovém katalogu

Proměnné

- Statické texty ve výkonném kódu nesmí být „hardcodovány“, dle potřeby musí být umístěny do proměnných, globálních proměnných nebo resources. Toto omezení se netýká logovacích informací. V každém projektu musí být jasná dohoda se zadavatelem o míře využití ressource souborů.
- *Doporučení: deklarace proměnných jsou umísťovány na začátek kódu*

Konfigurace

Pro konfiguraci **musí** být využívány standardní prostředky .net, tj. pro tlustého klienta *app.config* a pro tenkého *web.config*. Položky musí být okomentovány, tak aby byl jasný význam. Pro webové služby je využíván Web.config.

Connection string pro přístup k databázím může obsahovat jméno a heslo v plain textu.

Pro připojení k MS SQL serveru doporučujeme využívat integrované ověření (integrated authentication).

GlobalsData

Pokud jsou v projektu globální proměnné musí být umístěny v třídě **Globals** (soubor Globals.cs) Případně GlobalsData (GlobalsData.vb)).

Značka Region

Pro zpřehlednění kódu je používána značka Region, značky jsou pojmenovávány v angličtině. Je potřeba využívat doporučené regiony a dále vytvářet regiony dle potřeby dané úlohy.

Region	Popis
Imports	Importy knihoven
Variables and properties	Proměnné a vlastnosti
Windows Form Designer Generated Code	Automaticky generovaný kód z Visual Studia
Initialization	Inicializace formuláře, konstruktory tříd ...
Event handling	Řízení událostí (tlačítka, menu, toolbar ...)

Desktopové aplikace (aplikace typu tlustý klient a smart klient)

- Desktopové aplikace nesmí při práci s daty vytvářeny za běhu (dočasné soubory, soubory logu, dump atd.) či s daty určenými pro uchování stavu mezi jednotlivými spuštěními (soubory s uživatelskými preferencemi apod.) přímo přistupovat k souborovému systému a registry, místo toho musí používat koncept IsolatedStorage v .NET frameworku.
- Pro načítání vstupních dat musí desktopové aplikace používat standardní dialogy pro výběr souboru či složky s výchozím adresářem určeným odpovídající položkou konfigurace aplikace
 - System.Windows.Forms.OpenFileDialog
 - System.Windows.Forms.FolderBrowserDialog
- Pro zápis výstupních dat musí desktopové aplikace používat standardní dialogy pro určení souboru či složky s výchozím adresářem určeným odpovídající položkou konfigurace aplikace
 - System.Windows.Forms.SaveFileDialog
 - System.Windows.Forms.FolderBrowserDialog
- Aplikace nesmí používat přímý přístup do registry

Hlavní formulář

Každá aplikace musí mít hlavní formulář pojmenovaný **frmMain**. Tento formulář se spouští při spuštění aplikace.

Desktopové aplikace bez GUI

- Pro načítání vstupních dat, pro zápis výstupních dat, při práci s daty vytvářenými za běhu či s daty určenými pro uchování stavu mezi jednotlivými spuštěními musí aplikace používat adresáře či soubory určené odpovídající položkou konfigurace aplikace
- Aplikace nesmí používat přímý přístup do registry

Aplikace typu WindowsService

- Pro načítání vstupních dat, pro zápis výstupních dat, při práci s daty vytvářenými za běhu či s daty určenými pro uchování stavu mezi jednotlivými spuštěními musí aplikace používat adresáře či soubory určené odpovídající položkou konfigurace aplikace
- Aplikace nesmí používat přímý přístup do registry

Webové aplikace

- Pro načítání vstupních dat, pro zápis výstupních dat, při práci s daty vytvářenými za běhu či s daty určenými pro uchování stavu mezi jednotlivými spuštěními musí aplikace používat adresáře či soubory určené odpovídající položkou konfigurace aplikace
- Aplikace nesmí používat přímý přístup do registry
- Aplikace musí být otestovány na dvou či více serverech v režimu NLB (network load ballancing) pro použití session ve všech následujících režimech
 - ASP.NET Session State (StateServer)
 - SQL Session State

Webové služby

Vytvářené služby musí splňovat:

- Musí volány asynchronně
- Musí podporovat UDDI
- Musí mít návratovou hodnotu
- V případě chyby musí vracet jasný popis chyby
- Veškeré chyby a varování musí být logovány
- Musí být autorizovány
- V definici rozhraní webových služeb není bez předchozího schválení dovoleno používat netypové předávání strukturovaných informací (XML) s použitím XmlDocument, String či Object.
- V definici rozhraní webových služeb není doporučeno z důvodu problematické přenositelnosti napříč platformami používat DataSet.
- Webové služby musí být bezstavové, mezi jednotlivými voláními neudržují stav (session).
- **Webové služby musí vždy definovat strukturu požadavků request/response. Nesmí se jednat o primitivní datový typ (string, int apod.), ale musí to být třída či struktura zapouzdřující jednotlivé členy (např. Struct {string result, int code})**
- Target namespaces webových služeb musí obsahovat číslo verze.

Namespace rozhraní

Namespace rozhraní je textový řetězec jednoznačně identifikující dané rozhraní, je uložen v atributu `targetNamespace` elementu `wsdl:definitions`. Atribut bude ve tvaru:

`targetNamespace = [namespaceRozhraní]/[majorVerze].[minorVerze].[release]`

kde:

`[namespaceRozhraní]` je string ve tvaru

`namespaceRozhraní = http://interfaces.cssz.cz/Product_Name/\[Package name\]Schema name/Schema Version`

Blíže viz Popis jmenných prostorů XML v dokumentu Standard metodiky vývoje.

Dále `soapaction` bude ve tvaru : `soapAction="[targetNamespace]/[jménoOperace]`.

Takto bude dosaženo stavu - pokud bude klient komunikovat s nesprávnou verzí webové služby dostane odpověď HTTP 500 s textem výjimky : *SoapException: Server did not recognize the value of HTTP Header SOAPAction.*

Verze rozhraní

Verze rozhraní je řetězec ve tvaru `[majorVerze].[minorVerze].[release]` ,kde `[majorVerze]`,`[minorVerze]`,`[release]` jsou celá nezáporná čísla. Při změně definice rozhraní je doporučeno následující

- přidá/ubere se z rozhraní metoda nebo se změní její název - zvednout major verzi
- změni/přidá se datová struktura - zvednout minor verzi
- změni se název datové položky - zvednout minor verzi
- minimální změna primitivního typu, která nebude mít příliš důsledků ve změně kódu např: `int` na `long` – zvednout release
- jakákoliv jiná změna (za změnu je považováno že se soubor `wsdl` změní) – zvednout jakoukoliv verzi, dle uvážení závažnosti změny

příklad verze : 1.0.2 není doporučeno používat zaznamenání verze ve formátu 001.000.002

Webové služby musí být bezstavové, mezi jednotlivými voláními neudržují stav (session). V rámci optimalizace výkonu mohou webové služby využívat cache, pokud ji využívají, tak pouze implementace dostupné v rámci namespace `System.Web.Caching`.

Ošetření výjimek

- Všechny hlavní metody a procedury musí být ošetřeny pomocí bloku `try` (, `catch`, `end try`)
 - Odchytyvány musí být pouze výjimky, jež lze ošetřit (`CryptographicException`, `IOException` atd.), `SystemException` pouze ve výjimečných případech a každé odchytení obecné výjimky musí být zaneseno v dokumentaci
- Výjimky nesmí být potlačovány, všechny musí být zobrazeny (v případě klienta s GUI) a zapsány do logu dle konfigurace (všechny aplikace – tenký klient, tlustý klient, aplikace provádějící dávková zpracování, web aplikace, služba/server atd.)

- Popisy a kódy všech výjimek musí umísťovány do konstant (viz.: [Proměnné](#))
- Popisy výjimek (návratové hodnoty) musí být v českém jazyce, srozumitelné koncovému uživateli
- Pro zobrazení výjimek musí být vždy použit jednotný formulář, s chováním dle grafického manuálu (komponenta – Chybový formulář) (neošetřené výjimky musí být odchyceny a zobrazeny tímto formulářem také)
- Pro základní kontroly – maska, algoritmizovatelný formát, a podobně musí být použity standardní .NET validátory
- Třídy pro ošetření výjimek jsou umísťovány do namespace Err
- Každý projekt musí obsahovat soubor **Exceptions.vb** (případně Exceptions.cs.), do tohoto souboru jsou umísťovány třídy pro definici výjimek.
- Všechny výjimky musí být logovány viz. [Logování](#)
- Výjimky musí podporovat katalog chyb, zobrazované výjimky vygenerují chybový kód a odpovídající hlášku dotáhnou z katalogu chyb (katalog je možné/nutné replikovat do projektu do odpovídajícího (potřebného) formátu). Zobrazuje se ve formátu „Číslochyby – Popis“ v detailních informacích ErrorFormu je možné zobrazit ostatní položky z katalogu.

Databáze

Pro připojení k databázi ORACLE je používán výhradně Oracle Database Provider for .NET (ODP), v konfiguraci připojení musí být pouze záznam z TNSNAMES. Například z connect stringu:

DEV10G =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = 172.20.2.13)(PORT = 1521))

(CONNECT_DATA =

(SERVER = DEDICATED)

(SERVICE_NAME = DEV10G)

)

)

V konfiguraci smí být použit pouze text DEV10G

Pro připojení k databázi MS SQL 2005 je používán MS SQL Server 2005 Native Client.

Pro připojení k databázi MS SQL 2000 je používán MS SQL Server 2000 OLEDB Provider.

Logování

Všechny programy musí logovat pomocí komponenty log4net.

Pro všechny programy platí, že musí logovat veškeré výjimky, které nastanou a musí obsahovat debugovací a informační logy.

Logování musí podporovat katalog chyb, (katalog je možné/nutné replikovat do projektu do odpovídajícího (potřebného) formátu). Loguje se ve formátu „Číslochyby – Popis“.

Úrovně logování

Úroveň	Popis	Použití	Kategorie v katalogu chyb
FATAL	Neošetřené výjimky a chyby infrastrukturního charakteru	Standardní formulář pro zachycení výjimek (komponenta ErrorForm), handler pro odchycení neošetřených výjimek, ...	0
ERROR	Aplikační chyby	Veškeré aplikační výjimky a chyby	1,2 (dle příslušnosti)
WARN	Varování	Veškeré varování aplikace	1,2 (dle příslušnosti)
INFO	Informační hlášení	Veškeré informační hlášky	4
DEBUG	Debugovací a trasovací informace	Do důležitých částí systému, pro prověření běhu programu. Do všech větví programů (if, case,)	N/A

Konfigurace

Konfigurace je vždy prováděna pomocí souboru APP.config/web.config dle metodiky vývoje.

Kryptografické algoritmy

Pro použití jsou schváleny následující implementace kryptografických algoritmů:

Symetrické šifrování

- System.Security.Cryptography.RijndaelManaged AES (Rijndael)
- System.Security.Cryptography.TripleDESCryptoServiceProvider 3DES

Asymetrické šifrování

- System.Security.Cryptography.RSACryptoServiceProvider RSA
- System.Security.Cryptography.DSACryptoServiceProvider DSA

Pozn.: algoritmus Diffie-Hellman není v .NET 2.0 implementován, v případě nutnosti je možné požádat o povolení k použití komponent třetích stran.

Hash

- System.Security.Cryptography.SHA512Managed SHA512
- System.Security.Cryptography.SHA256Managed SHA256
- System.Security.Cryptography.RIPEMD160Managed RIPEMD160
- System.Security.Cryptography.SHA1Managed SHA1
- System.Security.Cryptography.SHA1CryptoServiceProvider SHA1
- System.Security.Cryptography.MD5CryptoServiceProvider MD5

Generování náhodných čísel

- System.Random - je možné použít pro generování náhodných čísel
- System.Security.Cryptography.RNGCryptoServiceProvider - je nutné použít, pokud aplikace vyžaduje generování kryptograficky silných náhodných čísel

Komprimační algoritmy

- System.IO.Compression.GZipStream – implementace algoritmu GZIP

Pozn.: Algoritmy RAR a ZIP nejsou v .NET implementovány. Pro jejich použití je možné požádat o povolení k použití komponent třetích stran.

Znovupoužitelné (sdílené) komponenty

Pro jednodušší dodržování konvencí, grafického manuálu, datového katalogu a dalších, jsou (případně budou) vytvořeny znovupoužitelné komponenty, ty jsou uloženy včetně dokumentace v source control a poskytovány subdodavatelům i internímu vývoji.

Komponenty třetích stran

Knihovny, komponenty a frameworky třetích stran smějí být použity až po schválení Odborem rozvoje a údržby IS ČSSZ.

Jejich použití musí být uvedeno v instalační dokumentaci a v akceptačním protokolu v samostatné kapitole včetně uvedení příslušné licence.

Dokumentace kódu

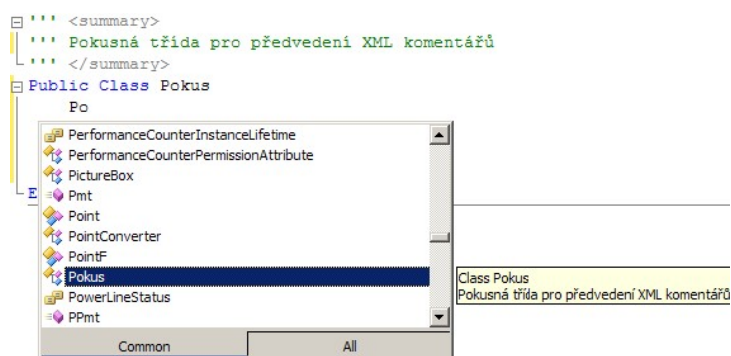
Pro snadnější práci se zdrojovým kódem je **povinné komentovat veškeré třídy, interface, metody pomocí XML dokumentace**, která umožňuje snadnou kompilaci referenční dokumentace kódu.

c	Text, který představuje zdrojový kód
code	Text, který představuje více řádek zdrojového kódu
example	Pro uvedení příkladu použití členu popř. třídy
exception	Slouží k uvedení výjimek, které mohou ve třídě nastat.
include	Umožňuje se referencovat na jiný soubor s XML dokumentací, kde existuje popis k typům v našem zdrojovém kódu.
list	Používá se pro uvedení výčtu hodnot
para	Umožňuje strukturování text. Měl by se nacházet uvnitř tagů summary, remarks nebo returns
param	Slouží k popisu parametru
paramref	Indikuje, že uvedené slovo označuje parametr
permissions	Určuje viditelnost členu
remarks	Umožňuje uvést doplňující informace o typu nebo jeho členovi.

returns	Tato značka by měla uvádět popis návratové hodnoty
see	Definuje odkaz na jiný programový element. Vhodné například pro implementaci interfaců
seealso	Definuje odkaz na jiný programový element. Tento odkaz by měl být uveden v sekci See Also.
summary	Uvádí základní informace o typu nebo jeho členovi.
value	Používá se popisu hodnoty vlastnosti.

Tato dokumentace je používána jak přímo při vytváření kódu:

Příklad použití při vytváření kódu



Tak pro generování programové dokumentace.

Poznámka: Pro VB v VS 2003 je potřebný [add on VB Commenter](#)

Programátorská dokumentace

Je generována pomocí nástroje SandCastle nebo nDoc2005. Pro každý projekt musí být vytvořena konfigurace pro generování. Proto je do každého hlavního projektu v solution vytvořen (nakopírován) konfigurační soubor, ten je pojmenován dle jména projektu a přípona ndoc (projectname.ndoc).

Verzování

Verzování veškerých produktů je popsáno v dokumentu „Pravidla releasu managementu“. Pro knihovny a WinForms jsou verze ukládány jak do „Assembly version“, tak do „File version“.

Strong name

- Klíče podpisu (strong name key) musí být předány spolu se zdrojovými kódy řešení
- Klíče musí být buď referencovány relativně vzhledem k umístění projektu nebo součástí projektu nebo uloženy do containeru a referencovány jménem klíče (v posledním případě musí být v dokumentaci předány klíče a názvy, pod jakými mají být naimportovány)

Přílohy

Seznam vyhrazených slov .NET

AddHandler	AddressOf	Alias	And	Ansi	As	Assembly
Auto	Base	Boolean	ByRef	Byte	ByVal	Call
Case	Catch	CBool	CByte	CChar	CDate	CDec
CDbl	Char	CInt	Class	CLng	CObj	Const
CShort	CSng	CStr	CType	Date	Decimal	Declare
Default	Delegate	Dim	Do	Double	Each	Else
Elseif	End	Enum	Erase	Error	Event	Exit
ExternalSource	False	Finalize	Finally	Float	For	Friend
Function	Get	GetType	Goto	Handles	If	Implements
Imports	In	Inherits	Integer	Interface	Is	Let
Lib	Like	Long	Loop	Me	Mod	Module
MustInherit	MustOverride	MyBase	MyClass	Namespace	New	Next
Not	Nothing	NotInheritable	NotOverridable	Object	On	Option
Optional	Or	Overloads	Overridable	Overrides	ParamArray	Preserve
Private	Property	Protected	Public	RaiseEvent	ReadOnly	ReDim
Region	REM	RemoveHandler	Resume	Return	Select	Set
Shadows	Shared	Short	Single	Static	Step	Stop
String	Structure	Sub	SyncLock	Then	Throw	To
True	Try	TypeOf	Unicode	Until	volatile	When
While	With	WithEvents	WriteOnly	Xor	eval	extends
instanceof	Package	var				

Doporučené pojmenování visual controls

Prefix	Typ objektu	Příklad
ctl	Control (všechny ostatní neuvedené v tabulce)	ctl Other
btn	Button control	btn Search
chk	CheckBox control	chk ReadOnly
lst	ListBox control	lst Name
lbl	Label control	lbl Name
lsv	ListView control	lsv Name
grp	GroupBox control	grp Section
txt	TextBox control	txt Name

tab	TabControl control	tabSections
tst	ToolStrip control	tstMain
cbo	ComboBox control	cboPosition
tre	TreeView control	treEmployees
grd	DataGridView control	grdEmployees
cxm	ContextMenuStrip control	cxmComponents
cdi	ColorDialog control	cdiColorChoose
iml	ImageList control	imlThumbnails
hlp	HelpProvider component	hlpEmployees
dtm	DateTimePicker control	dtmWhenStarted
dup	DomainUpDown control	dupEmployees
flp	FlowLayoutPanel control	flpSection
fbd	FolderBrowserDialog control	fbdSelectTargetFolder
fnt	FontDialog control	fntSelectNormalFont
clb	CheckedListBox control	clbSelectComponents
llb	LinkLabel control	llbCompanyUrl
msb	MaskedTextBox control	msbPhoneNumber
mnu	MenuStrip control	mnuMain
mca	MonthCalendar control	mcaCalendar
nic	NotifyIcon control	nicEmployees
nud	NumericUpDown control	nudRows
ofd	OpenFileDialog control	ofdSystem
pan	Panel control	panSection
pic	PictureBox control	picPhoto
pda	PrintDialog control	pdaPhotoPrint
ppd	PrintPreviewDialog control	ppdPhotoPrint
pgb	ProgressBar control	pgbStart
rbu	RadioButton control	rbuControlType
txr	RichTextBox control	txrDescription
sfd	SaveFileDialog control	sfdExportEmployees
sup	SoundPlayer control	supMusic
spl	SplitContainer control	splSections
sts	StatusStrip control	stsMain
tlp	TableLayoutPanel control	tlpSections
tip	ToolTip component	tipName

trc	TrackBar control	trcVolume
web	WebBrowser control	webConfirmLicence