



GORDIC

GINIS 2020

Modul GSAWST02

Klientské API pro webové služby XRG

Základní příručka uživatele

Od verze 4.84

Modul GSAWST02

Klientské API pro webové služby XRG

Základní příručka uživatele
Od verze 4.84

GINIS - Klientské API pro webové služby XRG

Žádná část této publikace nesmí být publikována a šířena žádným způsobem a v žádné podobě bez výslovného souhlasu společnosti GORDIC spol. s r.o.

Značky GORDIC® a GINIS® jsou ochrannými známkami a pobírají ochranu dle příslušných právních předpisů.

Vydal: GORDIC spol. s r.o., Erbenova 4, 586 01 Jihlava; IČ: 47 90 37 83.

Tisk: V Jihlavě 2020.

Obsah

Část 1	Úvod	7
1.1	Seznámení	7
1.2	Používané termíny a zkratky	7
1.3	Požadavky na prostředí	8
1.4	Instalace	8
Část 2	Popis programového rozhraní	10
2.1	Popis knihoven <i>Gordic.Tools.WsClientProxy</i> a <i>Gordic.Tools.WsClientProxy_x64</i>	10
2.1.1	Seznam parametrů knihovny <i>Gordic.Tools.WsClientProxy</i> a <i>Gordic.Tools.WsClientProxy_x64</i>	10
2.1.2	Popis funkce <i>CallWebMethod</i>	12
2.1.3	Popis funkce <i>GetError</i>	13
2.1.4	Popis funkce <i>GetErrorProperties</i>	13
2.1.5	Popis funkce <i>GetParameter</i>	14
2.1.6	Popis funkce <i>SetParameter</i>	15
2.1.7	Popis funkce <i>GetSupportedInterfaces</i>	15
2.1.8	Popis funkce <i>GetSupportedMethods</i>	16
2.2	Popis komponenty <i>Gordic.Tools.WsClient</i>	16
2.2.1	Vlastnosti komponenty <i>Gordic.Tools.WsClient</i>	17
2.2.2	Popis metody <i>CallWebMethod</i>	19
2.2.3	Popis metody <i>GetSupportedInterfaces</i>	19
2.2.4	Popis metody <i>GetSupportedMethods</i>	20
2.2.5	Popis metody <i>GetPasswordOption</i>	20
2.3	Testovací nástroj <i>Gordic.Tools.WsTest32</i> a <i>Gordic.Tools.WsTest64</i>	21
2.4	Testovací nástroj <i>Gordic.Tools.WsTest</i>	23
Část 3	Postupy a návody	26
3.1	Příklad použití knihovny <i>Gordic.Tools.WsClientProxy</i> a <i>Gordic.Tools.WsClientProxy_x64</i>	26
3.2	Příklad použití komponenty <i>Gordic.Tools.WsClient</i>	27
3.3	Použití konfiguračního souboru aplikace	27

Část 4	Dodatky a změny	29
4.1	O společnosti GORDIC spol. s r. o.	29
4.2	O informačním systému GINIS	29
4.3	Novinky	30

Předmluva

Dámy a pánové, vážení uživatelé,

dostává se Vám do rukou příručka pro práci v modulu Klientské API pro webové služby XRG, který je součástí systému GINIS - Gordic Integrovaný Informační Systém.

Tato příručka si klade za cíl poskytnout Vám potřebné informace pro plnohodnotnou práci s modulem, tak aby Vaše práce s programem byla maximálně efektivní a pro Vás příjemná.

Příručka je členěna do několika základních částí:

Úvod a obecné zásady práce v aplikaci

- úvodní část je věnována elementárnímu seznámení s modulem a ukazuje základní vazby v rámci systému GINIS[®], dále pak popisuje obecné zásady práce v aplikaci, typy ovládacích prvků a obecnou terminologii.

Popis práce v aplikaci, jak na to

- tato část tvoří jádro celé příručky a v maximální možné míře, při zachování čitelnosti a orientace v textu charakterizuje, jak se v modulu orientovat a jak s ním efektivně pracovat.

Dodatky a změny

- část obsahuje dodatky, doplnění, změny. Tato část představuje novinky v modulu a slouží pro rychlou orientaci a seznámení se s novými funkcionalitami.

Věříme, že tato příručka Vám bude platným pomocníkem při Vaší práci.

Hodně úspěchů přeje tým pracovníků GORDIC[®].

V Jihlavě, únor 2020.

1 Úvod

GSAWST02 ver. 4.84

Klientské API pro webové služby XRG je distribuováno jako doplněk informačního systému [GINIS](#). Toto API je primárně určeno pro vývojáře aplikací založených na platformě Win32. S úspěchem ho však lze využít i v případě klientských aplikací překládaných do prostředí Microsoft .NET. Jedná se o sadu sdílených knihoven, nástrojů a zdrojových kódů určených pro použití v aplikacích instalovaných a provozovaných výhradně v souladu s podmínkami provozu rozhraní XRG. Podmínky provozu rozhraní XRG lze nalézt na internetové adrese <https://robot.gordic.cz/xrg>.

Některé části příručky, zejména, nikoliv však výlučně, obrázky, se mohou v souvislosti s individuální konfigurací systému [GINIS](#) na každém projektu/implementaci lišit od grafického uživatelského rozhraní (GUI) zobrazené aplikace. Příručka je vytvořena s použitím demonstrační databáze nastavené v obecném režimu. Pro doplnění uvádíme, že zobrazení konkrétních tlačítek, ikon a dalších ovládacích prvků, včetně jejich funkcionalit, je kromě uvedeného nastavení uživatelských parametrů závislé na stavu zpracovávaného dokladu a na případných dalších okolnostech.

Podporované verze jednotlivých komponent provozního prostředí, zejména pak operačního systému, frameworků, komponent, doplňků a aplikací třetích stran zmíněných v dokumentaci se řídí samostatným dokumentem "Compatibility List - Technologie". Provozoschopnost informačního systému GINIS je podmíněna dodržením podmínek deklarovaných v dokumentu "Compatibility List - Technologie" odpovídající verzi informačního systému.

Datum poslední aktualizace dokumentace: **10.02.2020**.

1.1 Seznámení

Klientské API pro volání webových služeb XRG je distribuováno jako doplněk informačního systému [GINIS](#). Toto API je primárně určeno pro vývojáře aplikací založených na platformě Win32. S úspěchem ho však lze využít i v případě klientských aplikací překládaných do prostředí Microsoft .NET. Jedná se o sadu sdílených knihoven, nástrojů a zdrojových kódů určených pro použití v aplikacích instalovaných a provozovaných výhradně v souladu s podmínkami provozu rozhraní XRG. Podmínky provozu rozhraní XRG lze nalézt na internetové adrese <https://robot.gordic.cz/xrg>.

Pomocí klientského API pro volání webových služeb XRG lze relativně snadným způsobem provést spuštění jednotlivých webových metod rozhraní, předat jim patřičně zabezpečená vstupní data a v odpovědi získat požadované výstupní údaje. Pro prvotní otestování dostupnosti webových služeb XRG je společně s klientským API distribuován jednoduchý testovací nástroj.

1.2 Používané termíny a zkratky

GORDIC - společnost [GORDIC](#) spol. s r. o.

- GINIS** - informační systém GINIS vyvíjený a dodávaný společností [GORDIC](#) spol. s r. o.
- XRG** - zkratka pro XML rozhraní informačního systému GINIS, více informací o tomto rozhraní lze nalézt na internetové adrese <https://robot.gordic.cz/xrg>
- API** - zkratka anglického termínu Application Program Interface, což lze volně přeložit jako aplikační programové rozhraní
- WSE** - zkratka pro komponentu Microsoft Web Service Enhancements

1.3 Požadavky na prostředí

Klientské API pro volání webových služeb XRG je určeno výhradně pro provozování na počítačích s instalací některého z níže uvedených operačních systémů :

- Microsoft Windows 10 Anniversary Update a vyšší
- Microsoft Windows 8.1
- Microsoft Windows 7 s aktualizací SP1
- Microsoft Windows Server 2019
- Microsoft Windows Server 2016
- Microsoft Windows Server 2012 R2
- Microsoft Windows Server 2012
- Microsoft Windows Server 2008 R2 s aktualizací SP1

Klientské API pro volání webových služeb XRG pro svoji korektní funkčnost dále vyžaduje předchozí instalaci následujícího software :

- | | | |
|--|---|--|
| Microsoft .NET Framework 4.8 | - | instalace je zdarma ke stažení na internetové adrese https://dotnet.microsoft.com/download/ |
| Microsoft WSE 3.0 Runtime | - | instalace je zdarma ke stažení na internetové adrese https://www.microsoft.com/en-us/download/ |
| Microsoft Visual C++ 2015 Redistributable (x86) | - | instalace je zdarma ke stažení na internetové adrese https://www.microsoft.com/en-us/download/ |

1.4 Instalace

Klientské API pro volání webových služeb XRG je distribuováno prostřednictvím komprimovaného zip archivu dostupného ke stažení na adrese <https://robot.gordic.cz/xrg> pomocí příslušného odkazu dostupného po přihlášení do portálu. Po jeho dekomprimaci standardními prostředky je do cílového adresáře vytvořen adresář **@GINADR@** s podadresářem **WST02**, který obsahuje následující redistribuovatelné komponenty a soubory:

- | | | |
|----------------------------------|---|---|
| Gordic.Tools.WsClient.dll | - | .NET komponenta obsahující veřejné vlastnosti a metody klientského API pro volání webových služeb XRG |
|----------------------------------|---|---|

Gordic.Tools. WsClientProxy. dll	- sdílená nativní knihovna obsahující exportované funkce klientského API pro volání webových služeb XRG (32-bitová verze)
Gordic.Tools. WsClientProxy_ x64.dll	- sdílená nativní knihovna obsahující exportované funkce klientského API pro volání webových služeb XRG (64-bitová verze)
Gordic.Tools. WsClientProxy. Export.h	- hlavičkový soubor s deklarací exportovaných metod sdílených knihoven Gordic.Tools.WsClientProxy.dll a Gordic.Tools.WsClientProxy_x64.dll určený pro usnadnění vývoje aplikací s využitím klientského API pro volání webových služeb XRG.
Gordic.Tools. WsClientProxy. lib	- knihovna objektů pro Gordic.Tools.WsClientProxy.dll určená pro prostředí Microsoft Visual C++ 2015
Gordic.Tools. WsClientProxy_ x64.lib	- knihovna objektů pro Gordic.Tools.WsClientProxy_x64.dll určená pro prostředí Microsoft Visual C++ 2015
Gordic.Tools. WsTest.exe	- jednoduchý konzolový nástroj pro prvotní testování dostupnosti webových služeb XRG
Gordic.Tools. WsTest.exe. config	- konfigurační XML soubor pro utilitu Gordic.Tools.WsTest.exe
Gordic.Tools. WsTest32.exe	- jednoduchý nástroj pro prvotní testování dostupnosti webových služeb XRG (32-bitová verze)
Gordic.Tools. WsTest32.exe. config	- konfigurační XML soubor pro utilitu Gordic.Tools.WsTest32.exe
Gordic.Tools. WsTest64.exe	- jednoduchý nástroj pro prvotní testování dostupnosti webových služeb XRG (64-bitová verze)
Gordic.Tools. WsTest64.exe. config	- konfigurační XML soubor pro utilitu Gordic.Tools.WsTest64.exe
GSAWST02.chm	- soubor s nápovědou klientského API pro webové služby XRG
GSAWST02.pdf	- soubor s dokumentací klientského API pro webové služby XRG
GSAWST02.tst	- kontrolní soubor instalace klientského API pro webové služby XRG
ostatní soubory s příponou dll	- kolekce .NET komponent s podpůrnou funkcí

2 Popis programového rozhraní

2.1 Popis knihoven *Gordic.Tools.WsClientProxy* a *Gordic.Tools.WsClientProxy_x64*

Knihovny *Gordic.Tools.WsClientProxy* a *Gordic.Tools.WsClientProxy_x64* jsou sdílenými nativními knihovnami obsahující exportované funkce klientského API pro volání webových služeb XRG. Pomocí této funkcionality lze provést programové spuštění všech webových metod implementovaných v rozhraní XRG. Vlastnosti volání jednotlivých webových metod se přitom ovlivňují pomocí nastavení hodnot [parametrů](#) knihovny *Gordic.Tools.WsClientProxy* nebo *Gordic.Tools.WsClientProxy_x64*, a to vždy ještě před jejich vlastním spuštěním. Názorně je tento přístup demonstrován na [příkladu](#) použití knihovny *Gordic.Tools.WsClientProxy*, resp. *Gordic.Tools.WsClientProxy_x64*.

Seznam exportovaných funkcí knihovny *Gordic.Tools.WsClientProxy* a *Gordic.Tools.WsClientProxy_x64* :

- [CallWebMethod](#) - Spuštění webové metody na základě [parametrů](#) knihovny nastavených pomocí exportované funkce [SetParameter](#).
- [GetError](#) - Získání textu poslední běhové výjimky, která nastala při volání některé z exportovaných funkcí knihovny *Gordic.Tools.WsClientProxy*, resp. *Gordic.Tools.WsClientProxy_x64*.
- [GetErrorProperties](#) - Získání textu včetně dodatečných informací o poslední běhové výjimce, která nastala při volání funkce [CallWebMethod](#) exportované z knihovny *Gordic.Tools.WsClientProxy*, resp. *Gordic.Tools.WsClientProxy_x64*.
- [GetParameter](#) - Získání aktuální hodnoty [parametru](#) knihovny *Gordic.Tools.WsClientProxy*, resp. *Gordic.Tools.WsClientProxy_x64*.
- [SetParameter](#) - Nastavení hodnoty [parametru](#) knihovny *Gordic.Tools.WsClientProxy*, resp. *Gordic.Tools.WsClientProxy_x64*.
- [GetSupportedInterfaces](#) - Získání seznamu podporovaných typů webových rozhraní.
- [GetSupportedMethods](#) - Získání seznamu podporovaných webových metod v daném rozhraní.

2.1.1 Seznam parametrů knihovny *Gordic.Tools.WsClientProxy* a *Gordic.Tools.WsClientProxy_x64*

Parametrem knihovny *Gordic.Tools.WsClientProxy* a *Gordic.Tools.WsClientProxy_x64* se rozumí uspořádaná dvojice řetězců název - hodnota, jejichž pomocí lze ovlivnit chování klientského API webových služeb XRG při volání konkrétní webové metody. Aktuální hodnotu jednotlivých parametrů lze získat voláním funkce [GetParameter](#) nebo naopak nastavit je pomocí funkce [SetParameter](#). Hodnoty všech požadovaných parametrů je nicméně potřeba nastavit ještě před vlastním voláním funkce [CallWebMethod](#).

Seznam parametrů knihovny *Gordic.Tools.WsClientProxy*:

- Url** - URL adresa instalace webové služby na serveru. Povinný

	parametr.
Method	- Název požadované webové metody. Povinný parametr. Podporované názvy webové metod lze získat pomocí funkce GetSupportedMethods
User	- Uživatelské jméno. Nepovinný parametr.
Pwd	- Heslo. Nepovinný parametr.
InputFile	- Cesta ke XML souboru se vstupními daty webové služby. Nepovinný parametr. Vstupní data webové služby lze alternativně předat pomocí parametru RequestData . <i>Poznámka: při volání více webových metod po sobě se vstupními daty předávanými pomocí parametru InputFile je nutné zabezpečit před každým zavoláním další metody nastavení parametru RequestData na hodnotu prázdný řetězec.</i>
OutputFile	- Cesta ke XML souboru pro uložení výstupních dat webové služby. Nepovinný parametr. Výstupní data webové služby lze alternativně získat pomocí parametru ResponseData .
TicketFile	- Cesta k souboru se vstupenkou do systému. Nepovinný parametr. <i>Poznámka: vstupenka do systému je v současnosti již zastaralý institut a je uváděna výhradně z důvodu zpětné kompatibility dřívějších rozhraní. U nově vytvářených aplikací by proto již neměla být využívána.</i>
Proxy	- URL adresa proxy serveru. Nepovinný parametr.
Timeout	- Klientský časový limit pro dokončení zpracování v sekundách. Nepovinný parametr. Výchozí hodnota je 100 sekund, hodnota menší než 1 znamená volání webové služby bez časového limitu.
MaxMessageSize	- Maximální velikost zprávy v Bytech. Nepovinný parametr. Lze použít rovněž speciální hodnoty 0 = výchozí systémová hodnota a -1 = velikost zprávy bez omezení. <i>Poznámka: má význam pouze při volání WCF služeb.</i>
AllowUntrustedSslCertificate	- Příznak povolení komunikace pomocí neověřeného SSL certifikátu. Nepovinný parametr. Výchozí hodnota je ne. <i>Poznámka: má význam pouze při volání WCF služeb.</i>
WseVersion	- Požadovaná verze knihovny Microsoft WSE. Nepovinný parametr. Přípustná hodnota je výhradně 3.0 .
PasswordSecurity	- Styl zabezpečení hesla v odesílaných datech. Nepovinný parametr. Přípustné hodnoty jsou Hashed nebo Plain-text . V prvním případě je se vstupními daty webové službě namísto hesla odesílán pouze kontrolní součet, zatímco v druhém případě je heslo odesíláno ve formě prostého textu. Výchozí hodnotou parametru je Hashed.
RequestData	- Vstupní XML data webové služby ve formě textového řetězce. Nepovinný parametr. Vstupní data webové služby lze alternativně předat pomocí parametru InputFile . <i>Poznámka: hodnota tohoto parametru je při volání webové metody se vstupními daty předávanými pomocí parametru InputFile plněna automaticky. Před zavoláním další webové</i>

služby podle stejného scénáře je proto nutné ji dříve nastavit na hodnotu prázdný řetězec.

- ResponseData** - Výstupní XML data webové služby ve formě textového řetězce. Nepovinný parametr. Výstupní data webové služby lze alternativně získat pomocí parametru **OutputFile**.
- TicketData** - Data vstupenky do systému ve formě textového řetězce. Nepovinný parametr.
- Poznámka: vstupenka do systému je v současnosti již zastaralý institut a je uváděna výhradně z důvodu zpětné kompatibility dřívějších rozhraní. U nově vytvářených aplikací by proto již neměla být využívána.*
- RequestBinaryFile** - Cesta k souboru s binárními daty. Nepovinný parametr. Hodnota parametru je využívána pouze pokud daná webová metoda na svém vstupu očekává tento typ dat.
- CertificateFile** - Cesta k souboru s klientským certifikátem pro ověření uživatelské identity na serveru. Nepovinný parametr. Pokud je pro přístup k privátnímu klíči tohoto certifikátu vyžadováno heslo, předává se pomocí parametru **Pwd**.
- CertificateThumbprint** - Otisk (tj. miniatura) certifikátu v uložišti certifikátů, který má být použit pro ověření uživatelské identity na serveru. Nepovinný parametr. Pokud je pro přístup k privátnímu klíči tohoto certifikátu vyžadováno heslo, předává se pomocí parametru **Pwd**.
- SecurityProtocol** - Typ protokolu pro zabezpečení komunikace. Nepovinný parametr. Přípustné hodnoty jsou Ssl3, Tls, Tls11, Tls12 nebo Tls13. V případě nevyplnění hodnoty bude použito výchozí nastavení operačního systému.

2.1.2 Popis funkce CallWebMethod

Definice :

```
bool CallWebMethod ( );
```

Popis :

Spuštění webové metody na základě [parametrů](#) knihovny nastavených pomocí exportované funkce [SetParameter](#).

Vstupní parametry :

Tato funkce nemá žádné vstupní parametry.

Návratová hodnota :

Funkce v případě úspěšného provedení požadované akce vrací hodnotu **true**, naopak v případě výskytu běhové výjimky je vrácena hodnota **false**. Text této výjimky je možno získat následným voláním funkce [GetError](#) nebo [GetErrorProperties](#).

2.1.3 Popis funkce *GetError*

Definice :

```
bool GetError (  
    char * buffer,  
    long  bufferLength  
);
```

Popis :

Získání textu poslední běhové výjimky, která nastala při volání některé z exportovaných funkcí knihovny Gordic.Tools.WsClientProxy.

Vstupní parametry :

buffer

ukazatel na buffer pro uložení textu chyby

bufferLength

velikost bufferu v Bytech

Návratová hodnota :

Funkce v případě úspěšného získání textu výjimky vrací hodnotu **true**, v opačném případě je vrácena hodnota **false**.

2.1.4 Popis funkce *GetErrorProperties*

Definice :

```
bool GetErrorProperties (  
    char * messageBuffer,  
    long  messageBufferLength,  
    char * typeBuffer,  
    long  typeBufferLength,  
    int * code,  
    int * extendedCode,  
    char * techMessageBuffer,  
    long  techMessageBufferLength,  
    char * userMessageBuffer,  
    long  userMessageBufferLength  
);
```

Popis :

Získání textu včetně dodatečných informací o poslední běhové výjimce, která nastala při volání funkce [CallWebMethod](#) exportované z knihovny Gordic.Tools.WsClientProxy.

Vstupní parametry :

messageBuffer

ukazatel na buffer pro uložení textu chyby

messageBufferLength

velikost bufferu pro uložení textu chyby v Bytech

typeBuffer

ukazatel na buffer pro uložení typu chyby. Typ chyby přitom představuje konstantní řetězec umožňující bližší kategorizaci běhových výjimek. možné typy chyb jsou následující : aplikacni-chyba, technologicka-chyba, zaznam-uzamcen, zaznam-nenalezen nebo jina-chyba.

typeBufferLength

velikost bufferu pro uložení typu chyby v Bytech

code

ukazatel na proměnnou pro uložení čísla chyby

extendedCode

ukazatel na proměnnou pro uložení doplňkového čísla chyby

techMessageBuffer

ukazatel na buffer pro uložení technologicky zaměřeného popisu chyby

tecMessageBufferLength

velikost bufferu pro uložení technologicky zaměřeného popisu v Bytech

userMessageBuffer

ukazatel na buffer pro uložení uživatelsky orientovaného textu chyby

userMessageBufferLength

velikost bufferu pro uložení uživatelsky orientovaného textu popisu v Bytech

Návratová hodnota :

Funkce v případě úspěšného získání textu výjimky vrátí hodnotu **true**, v opačném případě je vrácena hodnota **false**.

2.1.5 Popis funkce GetParameter

Definice :

```
bool GetParameter (
    const char * parameterName,
    char * buffer,
    long bufferLength
);
```

Popis :

Získání aktuální hodnoty [parametru](#) knihovny Gordic.Tools.WsClientProxy.

Vstupní parametry :

parameterName

název [parametru](#) knihovny Gordic.Tools.WsClientProxy

buffer

ukazatel na buffer pro uložení hodnoty parametru

bufferLength

velikost bufferu v Bytech

Návratová hodnota :

Funkce v případě úspěšného provedení požadované akce vrací hodnotu **true**, naopak v případě výskytu běhové výjimky je vrácena hodnota **false**. Text této výjimky je možno získat následným voláním funkce [GetError](#).

2.1.6 Popis funkce *SetParameter*

Definice :

```
bool SetParameter (  
    const char * parameterName,  
    const char * parameterValue  
);
```

Popis :

Nastavení hodnoty [parametru](#) knihovny Gordic.Tools.WsClientProxy.

Vstupní parametry :

parameterName

název [parametru](#) knihovny Gordic.Tools.WsClientProxy

parameterValue

požadovaná hodnota [parametru](#) knihovny Gordic.Tools.WsClientProxy. Vložení prázdného řetězce je hodnota parametru vymazána.

Návratová hodnota :

Funkce v případě úspěšného provedení požadované akce vrací hodnotu **true**, naopak v případě výskytu běhové výjimky je vrácena hodnota **false**. Text této výjimky je možno získat následným voláním funkce [GetError](#).

2.1.7 Popis funkce *GetSupportedInterfaces*

Definice :

```
bool GetSupportedInterfaces (  
    char * buffer,  
    long bufferLength  
);
```

Popis :

Získání seznamu podporovaných typů webových rozhraní. Seznam je uveden v jednom řetězci, přičemž názvy jednotlivých podporovaných rozhraní v něm jsou odděleny čárkami.

Vstupní parametry :

buffer

ukazatel na buffer pro uložení seznamu podporovaných typů webových rozhraní

bufferLength

velikost bufferu v Bytech

Návratová hodnota :

Funkce v případě úspěšného provedení požadované akce vrací hodnotu **true**, naopak v případě výskytu běhové výjimky je vrácena hodnota **false**. Text této výjimky je možno získat následným voláním funkce [GetError](#).

2.1.8 Popis funkce *GetSupportedMethods*

Definice :

```
bool GetSupportedMethods (
    const char * interfaceName,
    char * buffer,
    long bufferLength
);
```

Popis :

Získání seznamu podporovaných webových metod v daném rozhraní. Seznam je uveden v jednom řetězci, přičemž názvy jednotlivých podporovaných webových metod v něm jsou odděleny čárkami.

Vstupní parametry :

interfaceName

název webového rozhraní získaného pomocí funkce [GetSupportedInterfaces](#)

buffer

ukazatel na buffer pro uložení seznamu podporovaných webových metod

bufferLength

velikost bufferu v Bytech

Návratová hodnota :

Funkce v případě úspěšného provedení požadované akce vrací hodnotu **true**, naopak v případě výskytu běhové výjimky je vrácena hodnota **false**. Text této výjimky je možno získat následným voláním funkce [GetError](#).

2.2 Popis komponenty *Gordic.Tools.WsClient*

Komponenta *Gordic.Tools.WsClient* je .NET assembly obsahující sadu veřejných vlastností a metod klientského API webových služeb XRG. Zmíněné vlastnosti a metody jsou ve zdrojovém kódu dostupné prostřednictvím instance třídy **Gordic.Tools.WsClient.GWsClient**. Pomocí zde obsažené funkcionality lze provést programové spuštění všech webových metod implementovaných v rozhraní XRG. Atributy volání jednotlivých webových metod se přitom ovlivňují pomocí nastavení hodnot zveřejněných [vlastností](#) objektu, a to vždy ještě před jejich vlastním spuštěním. Názorně je tento přístup demonstrován na [příkladu](#) použití komponenty

Gordic.Tools.WsClient.

Seznam veřejných vlastností komponenty Gordic.Tools.WsClient :

viz. kapitola [vlastnosti](#) komponenty Gordic.Tools.WsClient

Seznam veřejných metod komponenty Gordic.Tools.WsClient :

- [CallWebMethod](#) - Spuštění webové metody s dříve nastavenými [vlastnostmi](#)
- [GetSupportedInterfa
ces](#) - Získání seznamu podporovaných typů webových rozhraní.
- [GetSupportedMethod
s](#) - Získání seznamu podporovaných webových metod v daném rozhraní.
- [GetPasswordOption](#) - Převod řetězce na výčtový typ stylu zabezpečení hesla v odesílaných datech a naopak.

2.2.1 Vlastnosti komponenty Gordic.Tools.WsClient

Všechny vlastnosti komponenty Gordic.Tools.WsClient jsou dostupné prostřednictvím instance třídy **Gordic.Tools.WsClient.GWsClient**. Tyto vlastnosti jsou zpravidla určeny jak pro čtení, tak i pro zápis. Pomocí nastavení jejich hodnot se ovlivňují atributy volání jednotlivých webových služeb metodou [CallWebMethod](#).

Seznam veřejných vlastností komponenty Gordic.Tools.WsClient :

- Url** - URL adresa instalace webové služby na serveru. Povinná vlastnost typu řetězec znaků.
- Name** - Název požadované webové metody. Povinná vlastnost typu řetězec znaků. Podporované názvy webové metod lze získat pomocí metody [GetSupportedMethods](#)
- User** - Uživatelské jméno. Nepovinná vlastnost typu řetězec znaků.
- Pwd** - Heslo. Nepovinná vlastnost typu řetězec znaků.
- InputFile** - Cesta ke XML souboru se vstupními daty webové služby. Nepovinná vlastnost typu řetězec znaků. Vstupní data webové služby lze alternativně předat pomocí vlastností **RequestNode** nebo **RequestData**.
*Poznámka: při volání více webových metod po sobě se vstupními daty předávanými pomocí vlastnosti **InputFile** je nutné zabezpečit před každým zavoláním další metody nastavení vlastnosti **RequestNode** na hodnotu null.*
- OutputFile** - Cesta ke XML souboru pro uložení výstupních dat webové služby. Nepovinná vlastnost typu řetězec znaků. Výstupní data webové služby lze alternativně získat pomocí vlastností **ResponseNode** nebo **ResponseData**.
- TicketFile** - Cesta k XML souboru se vstupenkou do systému. Nepovinná vlastnost typu řetězec znaků.
Poznámka: vstupenka do systému je v současnosti již zastaralý institut a je uváděna výhradně z důvodu zpětné kompatibility dřívějších rozhraní. U nově vytvářených

	<i>aplikací by proto již neměla být využívána.</i>
Proxy	- URL adresa proxy serveru. Nepovinný parametr typu řetězec znaků.
Timeout	- Klientský časový limit pro dokončení zpracování v sekundách. Nepovinný parametr typu celé číslo. Výchozí hodnota je 100 sekund, hodnota menší než 1 znamená volání webové služby bez časového limitu.
MaxMessageSize	- Maximální velikost zprávy v Bytech. Nepovinný parametr. Lze použít rovněž speciální hodnoty 0 = výchozí systémová hodnota a -1 = velikost zprávy bez omezení. <i>Poznámka: má význam pouze při volání WCF služeb.</i>
AllowUntrustedSslCertificate	- Příznak povolení komunikace pomocí neověřeného SSL certifikátu. Nepovinný parametr. Výchozí hodnota je ne. <i>Poznámka: má význam pouze při volání WCF služeb.</i>
WseVersion	- Požadovaná verze knihovny Microsoft WSE. Nepovinná vlastnost typu řetězec znaků. Přípustná hodnota je výhradně 3.0 .
PasswordSecurity	- Styl zabezpečení hesla v odesílaných datech. Nepovinná vlastnost typu řetězec znaků. Přípustné hodnoty jsou Hashed nebo Plain-text . V prvním případě je se vstupními daty webové službě namísto hesla odesílán pouze kontrolní součet, zatímco v druhém případě je heslo odesíláno ve formě prostého textu. Výchozí hodnotou parametru je Hashed.
RequestNode	- Vstupní data webové služby. Nepovinná vlastnost typu System.Xml.XmlNode. Vstupní data webové služby lze alternativně předat pomocí vlastností InputFile nebo RequestData . Je-li vlastnost nastavena na hodnotu různou od null, mají zde uvedená data při zpracování přednost před daty získanými ze souboru se vstupními daty. <i>Poznámka: hodnota této vlastnosti je při volání webové metody se vstupními daty předávanými pomocí vlastnosti InputFile plněna automaticky. Před zavoláním další webové služby podle stejného scénáře je proto nutné ji dříve nastavit na hodnotu null.</i>
ResponseNode	- Výstupní data webové služby. Nepovinná vlastnost typu System.Xml.XmlNode. Výstupní data webové služby lze alternativně získat pomocí vlastností OutputFile nebo ResponseData .
TicketNode	- Data vstupenky do systému. Nepovinná vlastnost typu System.Xml.XmlNode. Je-li vlastnost nastavena na hodnotu různou od null, mají zde uvedená data při zpracování přednost před daty získanými ze souboru se vstupenkou. <i>Poznámka: vstupenka do systému je v současnosti již zastaralý institut a je uváděna výhradně z důvodu zpětné kompatibility dřívějších rozhraní. U nově vytvářených aplikací by proto již neměla být využívána.</i>
RequestData	- Vstupní XML data webové služby. Nepovinná vlastnost typu řetězec znaků. Tato vlastnost má shodnou funkci

	jako vlastnost RequestNode .
ResponseData	- Výstupní XML data webové služby. Nepovinná vlastnost typu řetězec znaků. Tato vlastnost má shodnou funkci jako vlastnost ResponseNode .
TicketData	- XML data vstupenky do systému. Nepovinná vlastnost typu řetězec znaků. Tato vlastnost má shodnou funkci jako vlastnost TicketNode .
RequestBinaryFile	- Cesta k souboru s binárními daty. Nepovinná vlastnost typu řetězec znaků. Hodnota vlastnosti je využívána pouze pokud daná webová metoda na svém vstupu očekává tento typ dat.
CertificateFile	- Cesta k souboru s klientským certifikátem pro ověření uživatelské identity na serveru. Nepovinná vlastnost typu řetězec znaků. Pokud je pro přístup k privátnímu klíči tohoto certifikátu vyžadováno heslo, předává se pomocí vlastnosti Pwd .
CertificateThumbprint	- Otisk (tj. miniatura) certifikátu v uložisti certifikátů, který má být použit pro ověření uživatelské identity na serveru. Nepovinná vlastnost typu řetězec znaků. Pokud je pro přístup k privátnímu klíči tohoto certifikátu vyžadováno heslo, předává se pomocí vlastnosti Pwd .

2.2.2 Popis metody CallWebMethod

Definice :

```
public void CallWebMethod ( );
```

Popis :

Spuštění webové metody na základě dříve nastavených [vlastností](#).

Vstupní parametry :

Tato metoda nemá žádné vstupní parametry.

Návratová hodnota :

Tato metoda nemá návratovou hodnotu.

2.2.3 Popis metody GetSupportedInterfaces

Definice :

```
public string GetSupportedInterfaces ( );
```

Popis :

Získání seznamu podporovaných typů webových rozhraní. Seznam je uveden ve výstupním řetězci, přičemž názvy jednotlivých podporovaných rozhraní v něm jsou odděleny čárkami.

Vstupní parametry :

Tato metoda nemá žádné vstupní parametry.

Návratová hodnota :

Seznam názvů podporovaných webových rozhraní oddělený čárkami.

2.2.4 Popis metody *GetSupportedMethods***Definice :**

```
public string GetSupportedMethods (
    string interfaceName
);
```

Popis :

Získání seznamu podporovaných webových metod v daném rozhraní. Seznam je uveden ve výstupním řetězci, přičemž názvy jednotlivých podporovaných webových metod v něm jsou odděleny čárkami.

Vstupní parametry :

interfaceName

název webového rozhraní získaného pomocí metody [GetSupportedInterfaces](#)

Návratová hodnota :

Seznam názvů podporovaných webových metod rozhraní oddělený čárkami.

2.2.5 Popis metody *GetPasswordOption***Definice :**

```
public string GetPasswordOption (
    Gordic.Tools.WsClient.GWsClient.PasswordOption passwordOption
);
```

```
public Gordic.Tools.WsClient.GWsClient.PasswordOption GetPasswordOption (
    string passwordOption
);
```

Popis :

Převod řetězce na výčtový typ stylu zabezpečení hesla v odesílaných datech a naopak.

Vstupní parametry :

passwordOption

styl zabezpečení hesla v jednom z následujících tvarů :

- hodnota výčtového typu **Gordic.Tools.WsClient.GWsClient.PasswordOption**

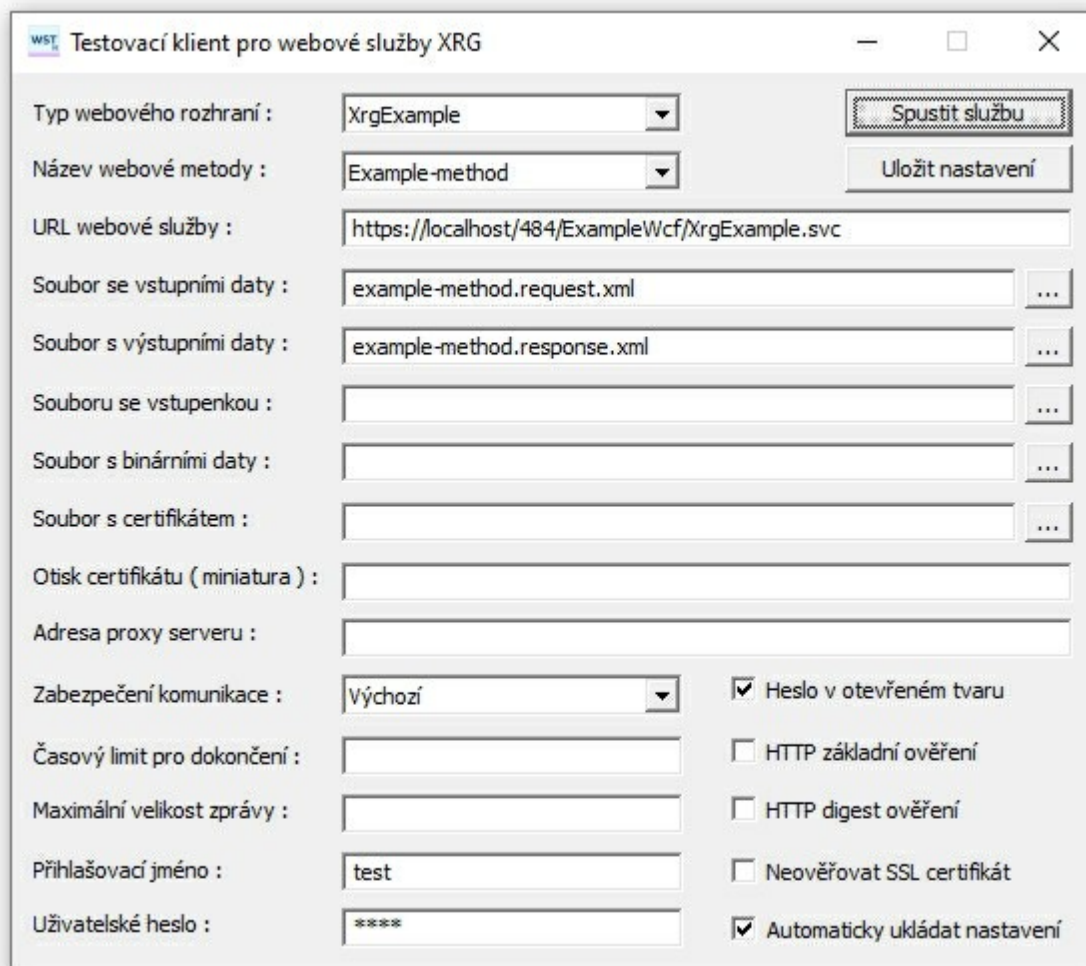
- textový řetězec s přípustnými hodnotami **Hashed** nebo **Plain-text**.

Návratová hodnota :

Konvertovaná hodnota stylu zabezpečení hesla v odesílaných datech.

2.3 Testovací nástroj Gordic.Tools.WsTest32 a Gordic.Tools.WsTest64

Nástroj Gordic.Tools.WsTest32, resp. Gordic.Tools.WsTest64, je dodáván společně s klientským API pro volání webových služeb XRG a je určen výhradně pro testování dostupnosti webových služeb XRG. Tento nástroj je vytvořen v prostředí Microsoft Visual C++ 2015, využívá knihovny MFC a je kompilován pro platformu Windows, a to buď v 32 nebo v 64 bitové verzi. Po jeho spuštění je zobrazeno hlavní okno, jehož příklad je vidět na následujícím obrázku :



Popis ovládacích prvků hlavního okna testovacího nástroje Gordic.Tools.WsTest32, resp. Gordic.Tools.WsTest64 :

Pole Typ - Slouží pro výběr názvu požadovaného webového rozhraní. Jedná se o

webového rozhraní	povinný údaj.
Pole Název webové metody	- Slouží pro výběr názvu spouštěné webové metody. Jedná se o povinný údaj.
Pole URL webové služby	- URL adresa instalace webové služby na serveru. Jedná se o povinný údaj.
Pole Soubor se vstupními daty	- Slouží pro zadání cesty k XML souboru se vstupními daty webové služby. Jedná se o nepovinný údaj.
Pole Soubor s výstupními daty	- Slouží pro zadání cesty k XML souboru pro uložení výstupních dat webové služby. Jedná se o nepovinný údaj.
Pole Soubor se vstupenkou	- Slouží pro zadání cesty k XML souboru se vstupenkou do systému GINIS. Jedná se o nepovinný údaj.
Pole Soubor s binárními daty	- Slouží pro zadání cesty k souboru s binárními daty pro případ, že webová služba na svém vstupu očekává tento typ dat. Jedná se o nepovinný údaj.
Pole Soubor s certifikátem	- Slouží k zadání cesty k souboru s klientským certifikátem pro ověření uživatelské identity na serveru. Pokud je pro přístup k privátnímu klíči tohoto certifikátu vyžadováno heslo, zadává se do pole Uživatelské heslo. Jedná se o nepovinný údaj.
Pole Otisk certifikátu	- Slouží k zadání otisku, neboli miniaturní, certifikátu v uložení certifikátů, který má být použit pro ověření uživatelské identity na serveru. Pokud je pro přístup k privátnímu klíči tohoto certifikátu vyžadováno heslo, zadává se do pole Uživatelské heslo. Jedná se o nepovinný údaj.
Pole Adresa proxy serveru	- Slouží k zadání URL adresy proxy serveru. Jedná se o nepovinný údaj.
Pole Zabezpečení komunikace	- Slouží pro výběr typu protokolu pro zabezpečení komunikace. Přípustné hodnoty jsou Ssl3, Tls, Tls11, Tls12 nebo Tls13. Jedná se o nepovinný údaj. V případě ponechání výchozí hodnoty bude použit protokol dle aktuálního nastavení operačního systému.
Pole Časový limit pro dokončení	- Slouží k zadání časového limitu pro dokončení volání webové služby v sekundách. Výchozí hodnota je 100 sekund, nastavení hodnoty na hodnotu menší než 1 značí volání webové služby bez časového limitu.
Pole Maximální velikost zprávy	Slouží k zadání maximální přípustné velikosti datové zprávy v Bytech pro WCF služby. Lze použít rovněž speciální hodnoty: 0 ... výchozí systémová hodnota -1 ... velikost zprávy bez omezení
Pole Přihlašovací jméno	- Slouží pro zadání uživatelského jména. Jedná se o nepovinný údaj.
Pole Uživatelské heslo	- Slouží pro zadání hesla. Jedná se o nepovinný údaj.
Pole Heslo v otevřeném	- Slouží k zadání požadavku na odeslání hesla na server ve formě prostého textu. Není-li toto pole zaškrtnuto, je se vstupními daty

tvaru	webové službě namísto hesla odesílán pouze jeho kontrolní součet. Pokud je toto pole zaškrtnuto, nelze použít zaškrtnutí u polí HTTP základní ověření ani HTTP digest ověření.
Pole HTTP základní ověření	- Slouží k odeslání jména a hesla na server ve tvaru pro ověření uživatelské identity metodou HTTP Basic. Pokud je toto pole zaškrtnuto, nelze použít zaškrtnutí u polí Heslo v otevřeném tvaru ani HTTP digest ověření.
Pole HTTP digest ověření	- Slouží k odeslání jména a hesla na server ve tvaru pro ověření uživatelské identity metodou HTTP Digest. Pokud je toto pole zaškrtnuto, nelze použít zaškrtnutí u polí Heslo v otevřeném tvaru ani HTTP základní ověření.
Pole Neověřovat SSL certifikát	- Slouží k zadání příznaku povolení komunikace pomocí neověřeného SSL certifikátu pro WCF služby.
Pole Automaticky ukládat nastavení	- Slouží k zadání požadavku na automatické ukládání nastavení aplikace do systémového registru při každém spuštění webové služby.
Tlačítko Spustit službu	- Provede vlastní volání webové služby na základě aktuálního nastavení hodnot v jednotlivých polích okna.
Tlačítko Uložit nastavení	- Uloží aktuální nastavení aplikace do systémového registru. Nastavení se ukládá do klíče HKCU\Software\Gordic\Gordic.Tools.WsTest32. Upozornění: vyplněné heslo je do registru ukládáno v otevřeném tvaru.
Tlačítka ...	- Slouží ke zobrazení dialogu pro výběr souboru.

2.4 Testovací nástroj Gordic.Tools.WsTest

Nástroj Gordic.Tools.WsTest je dodáván společně s klientským API pro volání webových služeb XRG a je určen výhradně pro testování dostupnosti webových služeb XRG. Jedná se o konzolovou aplikaci určenou pro běhové prostředí Microsoft .NET Framework. Po jeho spuštění bez parametrů je zobrazena nápověda k použití programu tak, jak ukazuje následující obrázek :

```

C:\WINDOWS\system32\cmd.exe
Použití: wstest.exe -a <url-služby>
                -n <název-metody>
                [ -ns <jmenný-prostor-služby> ]
                [ -i <vstupní-xml-soubor> ]
                [ -o <výstupní-xml-soubor> ]
                [ -u <uživatel> ]
                [ -p <heslo> ]
                [ -t <xml-soubor-se-vstupenkou> ]
                [ -d <soubor-s-binárními-daty> ]
                [ -c <soubor-s-přihlašovacím-certifikátem> ]
                [ -m <miniatura-přihlašovacího-certifikátu> ]
                [ -x <url-proxy-serveru> ]
                [ -e 3.0 ]
                [ -s hashed | plain-text | http-basic | http-digest ]
                [ -l <časový-limit> ]
                [ -v <velikost-zprávy> ]
                [ -uc true | false ]
                [ -sp ssl3 | tls | tls11 | tls12 | tls13 ]

Parametr -e udává požadovanou verzi WS Enhancements, verze 3.0 je defaultní.
Parametr -s udává styl zabezpečení odesílaného hesla, defaultní je hashed.
Parametr -l udává časový limit v sekundách, výchozí hodnota je 100,
             hodnota menší než 1 znamená volání bez časového limitu.
Parametr -v udává maximální velikost zprávy v Bytech, výchozí hodnota je 65536,
             hodnota menší než 1 znamená volání bez omezení velikosti zpráv.
Parametr -uc nastavuje příznak povolení neověřeného SSL certifikátu.
Parametr -sp nastavuje typ protokolu pro zabezpečení komunikace.

Press any key to continue . . .

```

Význam jednotlivých parametrů testovacího nástroje Gordic.Tools.WsTest je následovný :

- a <url-webové-služby>** Slouží pro zadání URL adresy instalace webové služby na serveru. Jedná se o povinný parametr.
- n <název-webové-metody>** Slouží pro zadání názvu spouštěné webové metody. Jedná se o povinný parametr.
- ns <jmenný-prostor-služby>** Slouží k explicitní specifikaci jmenného prostoru služby. Jedná se o nepovinný parametr.
- i <vstupní-xml-soubor>** Slouží pro zadání cesty ke XML souboru se vstupními daty webové služby. Jedná se o nepovinný parametr.
- o <výstupní-xml-soubor>** Slouží pro zadání cesty ke XML souboru pro uložení výstupních dat webové služby. Jedná se o nepovinný parametr.
- u <uživatel>** Slouží pro zadání uživatelského jména. Jedná se o nepovinný parametr.
- p <heslo>** Slouží pro zadání hesla. Jedná se o nepovinný parametr.
- t <xml-soubor-se-vstupenkou>** Slouží pro zadání cesty ke XML souboru se vstupenkou do systému GINIS. Jedná se o nepovinný parametr.
- d <soubor-s-binárními-daty>** Slouží pro zadání cesty k souboru s binárními daty pro případ, že webová služba na svém vstupu očekává tento typ dat. Jedná se o nepovinný parametr.
- c <soubor-s-přihlašovacím-certifikátem>** Slouží k zadání cesty k souboru s klientským certifikátem pro ověření uživatelské identity na serveru. Pokud je pro přístup k privátnímu klíči tohoto certifikátu vyžadováno heslo, zadává se do pole Uživatelské heslo. Jedná se o nepovinný parametr.
- m <miniatura-přihlašovacího-certifikátu>** Slouží k zadání otisku, neboli miniatury, certifikátu v uložišti certifikátů, který má být použit pro ověření uživatelské identity na serveru. Pokud je pro přístup k privátnímu klíči tohoto

- certifikátu vyžadováno heslo, zadává se do pole Uživatelské heslo. Jedná se o nepovinný parametr.
- x <url-proxy-serveru>** Slouží k zadání URL adresy proxy serveru. Jedná se o nepovinný údaj.
- e 3.0** Slouží k zadání požadované verze knihovny WS Enhancements, přičemž verze 3.0 je defaultní a jediná podporovaná. Jedná se o nepovinný parametr.
- s <typ-ověření-identity>** Slouží k zadání typu ověření klientské identity. Možná je jedna z následujících voleb:
- hashed** - ověření klientské identity pomocí protokolu WS Security s heslem předávaným ve formě kontrolního součtu
 - plain-text** - ověření klientské identity pomocí protokolu WS Security s heslem předávaným ve formě prostého textu
 - http-basic** - ověření klientské identity pomocí metody HTTP Basic
 - http-digest** - ověření klientské identity pomocí metody HTTP Digest
- Jedná se o nepovinný parametr.
- l <časový-limit>** Slouží pro zadání klientského časového limitu pro dokončení zpracování v sekundách. Výchozí hodnota je 100 sekund, hodnota menší než 1 znamená volání webové služby bez časového limitu. Jedná se o nepovinný parametr.
- v <velikost-zprávy>** Maximální velikost zprávy v Bytech pro WCF služby. Lze použít rovněž speciální hodnoty 0 = výchozí systémová hodnota a -1 = velikost zprávy bez omezení. Jedná se o povinný parametr.
- uc <příznak-povolení>** Příznak povolení komunikace pomocí neověřeného SSL certifikátu pro WCF služby. Jedná se o nepovinný parametr. Výchozí hodnota je false.
- sp <typ-protokolu>** Typ protokolu pro zabezpečení komunikace. Přípustné hodnoty jsou ssl3, tls, tls11, tls12 nebo tls13. Jedná se o nepovinný parametr. V případě neuvedení bude použito výchozí nastavení operačního systému.

3 Postupy a návody

3.1 Příklad použití knihovny

Gordic.Tools.WsClientProxy a

Gordic.Tools.WsClientProxy_x64

Následující příklad v syntaxi jazyka C++ demonstruje spuštění webové metody ws-info z imaginární URL adresy `http://localhost/example/example.asmx` pomocí exportovaných metod knihovny `Gordic.Tools.WsClientProxy`, resp. `Gordic.Tools.WsClientProxy_x64`. Vstupní data jsou načtena ze souboru `c:\tmp\request.xml` a výstup webové metody je uložen do souboru `c:\tmp\response.xml`. V případě úspěšného vykonání webové metody je na standardní výstup vypsán text `ok`, v opačném případě je vypsán text běhové výjimky.

```
#include "Gordic.Tools.WsClientProxy.Export.h"

void main() {
    bool result = false;
    char text[1024] = "ok";

    do {

        // nastavení URL adresy webové služby
        if(! SetParameter("Url","http://localhost/example/example.asmx")) break;

        // nastavení názvu požadované webové metody
        if(! SetParameter("Method","ws-info")) break;

        // nastavení cesty k souboru se vstupními daty
        if(! SetParameter("InputFile","c:\\tmp\\request.xml")) break;

        // nastavení cesty k souboru pro uložení výstupních dat
        if(! SetParameter("OutputFile","c:\\tmp\\response.xml")) break;

        // vlastní spuštění webové metody
        result = CallWebMethod();

    } while(false); // end while

    // získání textu případné běhové výjimky
    if(result == false) GetError(text,sizeof(text));

    // vypsání výsledku
    cout << text << endl;

} // end main
```

3.2 Příklad použití komponenty Gordic.Tools.WsClient

Následující příklad v syntaxi jazyka C# demonstruje spuštění webové metody ws-info z imaginární URL adresy `http://localhost/example/example.asmx` pomocí exportovaných metod knihovny `Gordic.Tools.WsClientProxy`. Vstupní data jsou načtena ze souboru `c:\tmp\request.xml` a výstup webové metody je uložen do souboru `c:\tmp\response.xml`. V případě úspěšného vykonání webové metody je na standardní výstup vypsán text `ok`, v opačném případě je vypsán text běhové výjimky.

```
using System;
using Gordic.Tools.WsClient;

static void Main(string[] args) {
    try {

        // vytvoření instance třídy Gordic.Tools.WsClient.GWsClient
        GWsClient l_oWsClient = new GWsClient();

        // nastavení vlastností pro volání webové metody
        l_oWsClient.Name = "ws-info";
        l_oWsClient.Url = "http://localhost/example/example.asmx";
        l_oWsClient.InputFile = @"c:\tmp\request.xml";
        l_oWsClient.OutputFile = @"c:\tmp\response.xml";

        // vlastní spuštění webové metody
        l_oWsClient.CallWebMethod();
        Console.WriteLine("ok");

    } // end try

    catch(Exception e) {
        Console.WriteLine(e.Message);
    } // end catch

} // end method
```

3.3 Použití konfiguračního souboru aplikace

Použití konfiguračního souboru volající aplikace je nezbytné v případě, že volaná webová služba je nakonfigurována na zabezpečení výstupních dat pomocí bezpečnostní známky s heslem předávaným formou kontrolního součtu. Není-li v takovém případě při volání webové služby příslušný konfigurační soubor aplikace nalezen, je při ověření výstupních dat generována běhová výjimka.

Konfigurační soubor je standardní XML konfigurační soubor pro prostředí Microsoft .NET Framework. Tento soubor musí být umístěn ve stejném adresáři jako exe soubor volající aplikace a musí mít s tímto exe souborem shodný název, doplněný navíc o příponu `.config`. Má-li tedy spustitelný soubor aplikace volající webovou službu název např. `MojeAplikace.exe`, bude název konfiguračního souboru `MojeAplikace.exe.config`. Obsah konkrétního konfiguračního souboru se může v závislosti na typu aplikace lišit, avšak pro ověření zabezpečení výstupních dat z webové služby XRG je zapotřebí uvést minimálně následující údaje:

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
```

```
<configSections>
  <section name="microsoft.web.services3"
    type="Microsoft.Web.Services3.Configuration.
WebServicesConfiguration,
    Microsoft.Web.Services3,
    Version=3.0.0.0, Culture=neutral,
    PublicKeyToken=31bf3856ad364e35" />
</configSections>
<microsoft.web.services3>
  <security>
    <securityTokenManager>
      <add localName="UsernameToken"
        type="Gordic.Tools.WsClient.GWseTokenManager3, Gordic.Tools.
WsClient"
        namespace="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" />
    </securityTokenManager>
  </security>
</microsoft.web.services3>
</configuration>
```

4 Dodatky a změny

4.1 O společnosti GORDIC spol. s r. o.

Firma GORDIC spol. s r. o. vznikla v Jihlavě v roce 1990 jako soukromá softwarová firma. Jádro firmy tvoří výkonný programátorský a analytický tým, převážně složený z bývalých pracovníků státní správy a bankovníctví, doplněný o softwarové a hardwarové specialisty na vysoké odborné úrovni.

Firma se zaměřila na vývoj, tvorbu a prodej flexibilního, legislativně závislého software pro oblast státní správy a samosprávy. V této oblasti se stal tento software velmi oblíbený a užívá jej několik tisíc uživatelů, mimo jiné resort MO ČR, magistrátní úřady města Prahy, Brna a Ostravy včetně podřízených organizací, 55 okresních úřadů, městské a obecní úřady, další rozpočtové i příspěvkové organizace. Postupně došlo k rozšíření i mimo oblast státní správy a samosprávy.

Péče o zákazníka poskytovaná firmou je komplexní. K nejdůležitějším výhodám, které zákazník získá, je spolehlivost, rychlost, variabilita a legislativní závislost programového vybavení, podpora prostřednictvím poskytovaných konzultací a hotline a v neposlední řadě metodická podpora a podpora spočívající v systematické odborné přípravě pracovníků uživatele.

Více informací o společnosti GORDIC spol. s r. o. naleznete na stránkách www.gordic.cz.

4.2 O informačním systému GINIS

GORDIC GINIS představuje komplexní řešení informačního systému organizace. Zahrnuje ekonomické agendy, řízení oběhu dokumentů prostřednictvím spisové služby, řadu registrů a správních agend včetně softwarového řešení vedení správního řízení. Systém je vyvíjen od samého počátku s ohledem na legislativní prostředí veřejné správy s vysokými nároky na bezpečnost a autentičnost a disponuje řadou modulů pro podporu specifických činností státních a samosprávních úřadů. Jeho součástí jsou také aplikace pro podporu e-governmentu, včetně moderních portálových komponent. Také díky tomu je GINIS v současnosti nejpoužívanějším komplexním informačním systémem v této oblasti.

Základní komponenty informačního systému GINIS

- Jádro systému
- Subsystem Ekonomika
- Subsystem Spisová služba
- Subsystem Personalistika
- Subsystem Registry
- Subsystem Správní agendy

Informační systém GINIS a technologie

Společnost GORDIC spol. s r.o. pokračuje v nastoupeném směru vývoje aplikací informačního systému GINIS v technologii .NET. Orientace na tuto platformu byla zvolena zejména pro

moderní pojetí její technologické architektury, která umožňuje okamžitě řešit většinu požadavků kladených v současnosti na softwarové produkty GORDIC. Kromě toho obsahuje programový model .NET kompaktní a ucelenou sadu nástrojů a komponent, jež po všech stránkách představuje přirozené pokračování ve vývoji informačního systému GINIS.

Výraznou modernizací IS GINIS je změna architektury systému z modelu klient-server do tzv. n-vrstvé architektury. Jedná se o rozdělení původně monolitických aplikací do více logických aplikačních celků. V této souvislosti přitom můžeme označit současnou etapu budování systému za přechod od vývoje tzv. tlustých klientů na vytváření tzv. distribuovaných aplikací.

Jednou z výrazných vlastností systému GINIS je jeho otevřenost umožňující kontrolovatelnou a bezpečnou výměnu dat s dalšími IS v organizaci i mimo ni. Integrovaná platforma GINIS zahrnuje komplexní soubor technologií, zaměřených na procesní i datovou integraci heterogenních softwarových systémů a aplikací.

Integrační platforma GINIS

- sjednocuje integrační technologie do společné platformy založené na používání mezinárodních otevřených průmyslových standardů,
- umožňuje bezproblémovou implementaci nezávisle na organizačním uspořádání, či fyzickém rozdělení subjektu do více geograficky oddělených lokalit,
- poskytuje zabezpečenou datovou komunikaci uvnitř organizace i mimo ni, k čemuž přednostně využívá existující IT infrastrukturu subjektu,
- podporuje alternativní přístup směřující k zavedení systémové integrace s progresivní architekturou orientovanou na služby, která je obecně známá pod termínem Services Oriented Architecture (SOA),
- přináší rozsáhlou množinu standardně dodávaných a provozem ověřených komponent, které vznikly na základě praktických požadavků z již realizovaných integračních projektů u subjektů z oblasti veřejné správy,
- garantuje vysokou flexibilitu, otevřenost a snadnou rozšiřitelnost, která vychází přímo ze základní koncepce integr. platformy GINIS, tedy orientace na použití standardizovaných technologií spojených s využitím jazyka XML,
- nabízí příležitost pro inovaci a optimalizaci stávajících procesů v organizaci s ohledem na snížení celkových nákladů na vlastnictví IT infrastruktury.

4.3 Novinky

Novinky ve verzi 4.84

V této verzi nejsou žádné novinky.

Novinky ve verzi 4.82

V této verzi nejsou žádné novinky.