

# Vývoj webovského rozhraní aplikace

V této kapitole je na příkladu firemní nástěnky vysvětlen postup při vývoji uživatelského rozhraní aplikace, která je provozována prostřednictvím Internetu, webserveru a internetových prohlížečů (browserů).

Vysvětlíme si, jak ve vývojovém prostředí WinBase602 navrhnout dynamicky generované HTML stránky. Uvedeme také důležité informace o bezpečnosti při provozování aplikací s webovským rozhraním.

Kapitola je rozdělena na tři části: v první, pojmenované *Teoretický úvod*, se seznámíme s tím, jakým způsobem se data z databáze zpřístupní na Internetu.

Ve druhé části, nazvané *Začínáme*, se naučíme, jak vytvářet jednoduché šablony.

V poslední části se budeme zabývat pokročilými aspekty vývoje webovského rozhraní, jako např. přístupová práva k WWW objektům, bezpečnost při provozu aplikací, rychlost atd.

## Teoretický úvod

Dříve než se budeme věnovat vývoji WWW objektů, vysvětlíme si krátce princip, podle kterého jsou generovány HTML stránky podle existujících WWW objektů.

Technické podrobné informace o parametrech a konfigurování jsou uvedeny v *Průručce správce*, kapitola *Přístup pomocí Internet klienta*. Popis a syntaxi jazyka Internet klientů naleznete v elektronické nápovědě k WinBase602 Internet klientům (WBCGI.HLP).

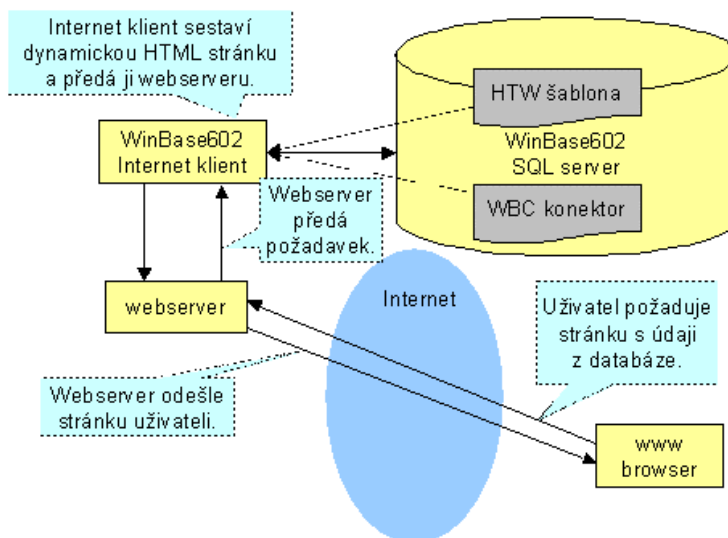
Při vytváření dynamické HTML stránky spolupracuje několik aplikací:

1. **www browser** - s touto aplikací pracuje uživatel, browser odesílá požadavky webserveru a zobrazuje výslednou stránku
2. **webserver** - komunikuje s browserem, předává uživatelský požadavek WinBase602 Internet klientovi, přebírá od něj sestavenou stránku a odesílá ji browseru
3. **WinBase602 Internet klient**
  - přebere od webserveru požadavek na sestavení dynamické HTML stránky
  - vyhledá a přečte WBC KONEKTOR (jeho název je obsažen v požadavku, který Internet klient obdržel od webserveru) a provede operace s databází, které jsou popsány

v tomto konektoru (tj. typicky sestaví a provede SQL příkazy a v případě příkazu SELECT přečte ty údaje z databáze, které vyhovují podmínce v příkazu SELECT)

- vyhledá a přečte HTW ŠABLONU a podle ní sestaví výslednou HTML stránku, přičemž podle příkazů v šabloně zformátuje přečtené údaje z databáze do výsledné stránky
  - výslednou stránku předá webserveru
4. **WinBase602 SQL server** - spravuje data, se kterými pracuje daná aplikace a provádí SQL příkazy, které sestavil **Internet klient**; uchovává WBC konektory a HTW šablony

Bližší představu o mechanismu sestavování dynamické HTML stránky Vám dá následující obrázek:



Struktura požadavku, který vysílá www browser a který obdrží **Internet klient**, je následující:

```
http://www.server.cz/cgi-bin/wbcgi.exe/databáze/aplikace/konektor.wbc
```

- a) komunikační protokol    b) název serveru    c) adresář se skripty    d) Internet klient    e) název databáze    f) název aplikace    g) název konektoru

Zároveň s požadavkem www browser vysílá také seznam tzv. formulářových proměnných a jejich hodnot. Jsou to údaje, které uživatel zadal v HTML formuláři (tag <FORM>),

respektive v jednotlivých prvcích formuláře (tagy <INPUT>). **Internet klient** je volán typicky v okamžiku, kdy uživatel stiskne tlačítko SUBMIT (tj. <INPUT TYPE="SUBMIT" VALUE="Odeslat">) a **Internet klient** zpracuje zadané údaje (které dostane prostřednictvím formulářových proměnných) a vygeneruje HTML stránku, kterou následně browser zobrazí.

**WinBase602 Internet klienti** jsou realizováni formou skriptů, které rozšiřují možnosti webserveru. Z tohoto důvodu je nutno v požadavku uvést adresář se skripty webserveru a také název skriptu (tj. **Internet klienta**), který má vyřídit tento požadavek (protože samotný webserver neumí sestavovat dynamické HTML stránky s údaji z databáze **WinBase602**).

Místo názvu konektoru lze uvést název HTW šablony, která však musí obsahovat tzv. WBC sekci, což je vlastně konektor ve zcela stejné podobě, v jaké by byl v samostatném souboru.

## WBC konektory

WBC konektor je buď samostatný objekt kategorie WWW objekty v databázi **WinBase602**, nebo část HTW šablony. Jsou v něm popsány jazykem WBC konektorů akce, které je třeba vykonat s daty v databázi při sestavení výsledné HTML stránky.

Název WBC konektoru je obsažen v požadavku, který převezme **Internet klient** od webserveru.

Jednotlivé akce, které budou vykonány při generování HTML stránky, se označují termínem KOMPONENTY. Mezi nejdůležitější typy komponent patří SQL příkazy (SELECT, INSERT apod.) a programy ve vnitřním jazyku **WinBase602**.

## HTW šablony

Je to databázový objekt kategorie WWW objekty, který obsahuje jednak text ve formátu HTML, jednak speciální HTW příkazy, kterými je řízeno formátování dat z databáze do HTML textu. **Internet klient** provádí toto formátování dat do HTML textu podle výše zmíněných HTW příkazů. Výsledkem práce **Internet klienta** je text pouze ve formátu HTML, tj. bez HTW příkazů.

Šablona typicky v sobě obsahuje i WBC konektor. Při generování výsledné HTML stránky **Internet klient** provede akce obsažené ve WBC konektoru, přečte požadovaná data (např. výsledky SQL příkazů SELECT) a podle HTW příkazů zformátuje získaná data do HTML textu.

## Začínáme

V této části se zaměříme na základní postupy při vývoji webovského rozhraní aplikace. Vysvětlíme, které funkce pro vývoj webovského rozhraní nám nabízí Vývojové prostředí WinBase602, naučíme se s nimi pracovat a vytvoříme jednoduché šablony pro dynamické generování HTML stránek. Programátor, který chce přidat své aplikaci webovské rozhraní, by měl, mimo dobré znalosti Vývojového prostředí WinBase602, také alespoň zčásti znát jazyk HTML a být obeznámen s mechanismem fungování webserverů.

Uvedená témata vysvětlíme na následujícím velice jednoduchém modelovém příkladu:

*Firma Doprava603 by chtěla na své vnitrofiremní (intranetové) síti provozovat firemní vývěsku. Zprávy přidávané do této vývěsky budou ukládány do tabulky Zprávy s následující strukturou:*

```
Nadpis: String[50]
Vytvořeno kým: CSISString[30]
Zpráva: Text
Datum vytvoření: Date4
Čas vytvoření: Time
Platnost dní: Short
```

Pomocí webovského rozhraní chceme zaměstnancům umožnit zobrazení existujících zpráv na vývěsce a vytvoření nové zprávy.

### Vytváříme tabulku

Nejdříve založíme novou aplikaci, pojmenujeme ji např. **Vývěska**. V ní vytvoříme tabulku **Zprávy** s výše uvedenými sloupci. U sloupce `Datum vytvoření` nastavíme implicitní hodnotu `CURRENT_DATE` a u sloupce `Čas vytvoření` implicitní hodnotu `CURRENT_TIME`. Postup při vytváření aplikace a tabulky je popsán v *Příručce uživatele*.

Dále v aplikaci vytvoříme roli **Web\_user** a obsadíme do ní uživatele `__web` (jméno, kterým se přihlašují **Internet klienti** - viz *Příručka správce*, kapitola *Přístup pomocí Internet klienta* a kapitola *Správa uživatelů*). Pokud uživatel `__web` není založen, je nutné ho vytvořit.

Pomocí této role nastavíme práva k jednotlivým objektům aplikace tak, aby uživatel `__web` s těmito objekty mohl pracovat. Práva k objektům přidělíme této roli a výše uvedeného uživatele do této role obsadíme.

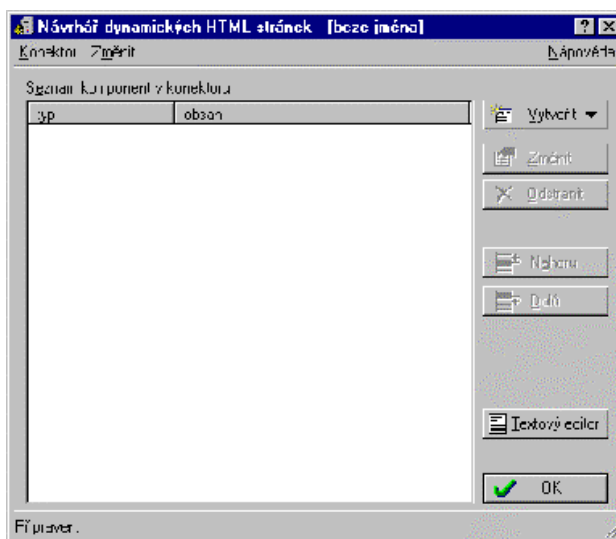
Nastavíme tato práva roli **Web\_user** k tabulce **Zprávy**:

- použití objektu
- čtení hodnot všech sloupců
- vkládání nových záznamů

## Zobrazení zpráv na vývěsce

Chceme vytvořit WWW objekt, který bude zobrazovat zprávy, které jsou momentálně na vývěsce.

V naší aplikaci na ovládacím panelu Vývojového prostředí si zobrazíme kategorii WWW objekty a provedeme akci **Vytvořit**. V dialogu **Vyberte druh nového objektu** zvolíme šablonu. Objeví se okno Návrhář dynamických HTML stránek.

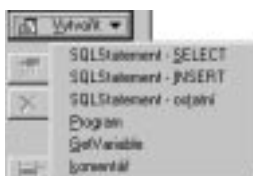


Jak už víme, HTW šablona obsahuje v sobě také WBC konektor. Pomocí tohoto Návrháře vytváříme právě WBC konektor.

Chceme zobrazit zprávy, které jsou na naší vývěsce, převedeno do databázové terminologie: chceme zobrazit ty záznamy z tabulky Zprávy, které v daném časovém okamžiku ještě neztratily svou platnost. Tj. záznamy, pro které platí

```
`datum vytvoření` + `platnost dní` >= CURRENT_DATE
```

Vytvoříme tedy komponentu typu SELECT. Stiskneme tlačítko Vytvořit a v menu vybereme typ komponenty: **SQLStatement – SELECT**



V dialogu **Vlastnosti komponenty SQLStatement – SELECT** zadáme pojmenování výsledku SQL příkazu `SELECT`. Je to jméno, pomocí kterého se budeme v šabloně odkazovat na záznamy, které vyhovují tomuto příkazu `SELECT`. Zadáme např. "zprávy". Dále ponecháme nastavenou volbu načíst všechny záznamy, které vyhovují dotazu.

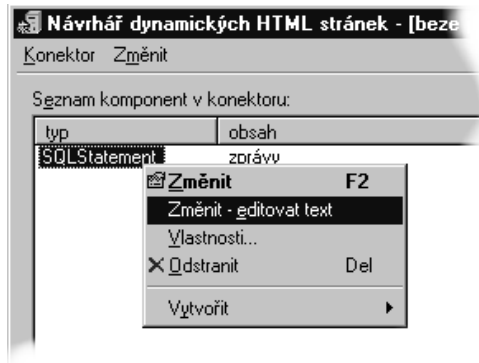


Stisknutím klávesy **OK** přejdeme do prostředí **Návrháře dotazů**. V něm vytvoříme dotaz, který vybere ty záznamy z tabulky `Zprávy`, které vyhovují výše uvedené podmínce. Dále chceme, aby dotaz uspořádal vyhovující záznamy podle data (tj. podle hodnot sloupce ``datum vytvoření``) sestupně a podle času vytvoření (podle hodnot sloupce ``čas vytvoření``).

Výsledný dotaz zapsaný v jazyku SQL bude vypadat takto:

```
SELECT *
FROM Zprávy
WHERE (`datum vytvoření`+`platnost dní`>=CURRENT_DATE)
ORDER BY Zprávy.`datum vytvoření` DESC,
        Zprávy.`čas vytvoření` DESC
```

Tento zdrojový kód dotazu uvidíte (poté, co uložíte nově vytvořený dotaz v **Návrháři dotazů** a vrátíte se zpět do prostředí **Návrháře dynamických HTML stránek**), když zvolíte příkaz "Změnit – editovat text" z kontextového menu nově vytvořené komponenty.



Další akce s daty v databázi nechceme provádět, přejdeme tedy k návrhu šablony. V menu **Konektor** zvolíme příkaz **Editovat šablonu**. Než budeme moci editovat šablonu, systém nás vyzve k zadání jména nově vytvořené šablony. Pojmenujme ji **Seznam**.

Po zadání jména nového WWW objektu bude zobrazen textový editor se zdrojovým textem naší šablony. Nyní tato šablona obsahuje pouze zdrojový text právě vytvořeného WBC konektoru. Naším úkolem je přidat do šablony jednak HTML kód stránky, a jednak HTW příkazy, kterými vložíme záznamy, které vyhovují SQL příkazu `SELECT`, do stránky.

Nejdříve doplníme šablonu o HTML příkazy:

```
<%wbc%>
SQLStatement: zprávy
+ SELECT *
+ FROM Zprávy
+ WHERE (`datum vytvoření`+'platnost dní`>=CURRENT_DATE)
+ ORDER BY Zprávy.`datum vytvoření` DESC,
+ Zprávy.`čas vytvoření` DESC
<%/wbc%>
```

```
<html>
<head>
<title>Vývěska</title>
</head>
```

```
<body>
<h1>Vývěska firmy Doprava603</h1>
```

Naše vývěska obsahuje celkem --zde bude doplněn počet zpráv-- zpráv.

```
<p>
--zde bude seznam zpráv--
<p>
</body>
</html>
```

Pokud bychom nyní zprovoznili tuto šablonu, budou sice přečteny záznamy z databáze, ale vygenerovaná HTML stránka bude vždy stejná. Aby HTML stránka obsahovala aktuální údaje z databáze, je třeba na označená místa zdrojového textu šablony doplnit:

```
<%zpravy#%>
a
<dl>
<%begindetail zpravy%>
<dt>od <%`vytvoreno kym`%>: <b><%nadpis%></b>
(<%`datum vytvoreni`%>)
<dd><%zprava%>
<%enddetail%>
</dl>
```

Nyní vysvětlíme význam jednotlivých příkazů. Nejdříve si všimněme, že všechny HTW příkazy jsou uzavřeny v závorkách <% a %> (tzv. HTW závorky).

Počet zpráv na  
vývěsce

V prvním příkazu chceme vypsát na dané místo stránky počet záznamů ve výsledku SQL příkazu SELECT, který jsme si pojmenovali "zpravy". K tomuto účelu slouží operátor #, jehož hodnotou je právě tento údaj. Jakýkoliv výraz vypíšeme do sestavované stránky tak, že tento výraz jednoduše uzavřeme do HTW závorek (citace hodnoty výrazu).

Seznam zpráv na  
vývěsce

Další skupina příkazů vypíše do sestavované HTML stránky ty zprávy, které jsou v odpovědi na SQL příkaz SELECT, který jsme pojmenovali "zpravy". K tomuto účelu jsou použity HTW příkazy begindetail a enddetail, souhrnně NAZÝVANÉ DETAIL SEKCE. V hlavičce detail sekce (v příkazu begindetail) uvádíme název SQL příkazu SELECT, který má být použit.

V těle detail sekce (mezi oběma příkazy) popisujeme, jak vygenerovat jeden záznam z SQL příkazu SELECT. Každý záznam v příkazu SELECT bude vygenerován podle těla detail sekce. Dále jsou zde pouze citace hodnot sloupců SELECTu.

**Příklad:** V tabulce **Zprávy** jsou následující dva záznamy:

Název	Vytvořeno kým	zpráva	datum vytvoření	čas vytvo- ření	Platnost dní
Příspěvek 1	Jan Novák	zpráva 1	1.1.1999	15:50:00	14
Příspěvek 2	Jiří Stříbrný	zpráva 2	8.1.1999	10:51:00	14

a HTML stránku generujeme např. 9.1.1999 (aby oba záznamy nepozbyly své platnosti).

Pak z detail sekce vzniknou následující HTML příkazy:

```
<dl>
<dt>od Jiří Stříbrný: <b>příspěvek 2</b> (8.1.1999)
```



```

<dd>zpráva 2
<dt>od Jan Novák: <b>příspěvek 1</b> (1.1.1999)
<dd>zpráva 1
</dl>

```

Další podrobnosti o citaci výrazů a o detail sekcích naleznete v elektronické nápovědě k Internet klientům (WBCGL.HLP).

Před provedením šablony

Dříve, než zkusíte šablonu vykonat **Internet klientem**, je třeba přidělit roli **Web\_user** právo použití WWW objektu **Seznam**.

Dále doporučujeme vložit do tabulky **Zprávy** alespoň několik záznamů, aby bylo co zobrazovat.

Popis konfigurace webserveru pro běh **Internet klientů** naleznete v *Příručce správce* v kapitole *Přístup pomocí Internet klientů*.

## Vkládání nové zprávy

Nyní vytvoříme šablonu, pomocí které budou uživatelé vkládat nové zprávy do naší vývěsky. Nakonec propojíme obě šablony mezi sebou.

Vkládání nových zpráv

Vytvoříme šablonu se jménem **Formulář**, ve které bude formulář pro vytvoření nové zprávy na vývěsce. Na ovládacím panelu provedeme akci **Vytvořit** a vybereme druh WWW objektu šablona. Návrháře dynamických HTML stránek (resp. Návrháře konektoru) ukončíme příkazem **Editovat šablonu** z menu **Konektor**, protože nepotřebujeme vytvořit žádnou komponentu konektoru. Objeví se textový editor s nově vytvořenou šablonou.

V něm vytvoříme HTML stránku s formulářem, pomocí kterého budou uživatelé zadávat své jméno, nadpis zprávy, text zprávy a dobu expirace zprávy. Stránka může vypadat např. takto:

```

<html>

<head>
<title>Formulář</title>
</head>

<body>

<table border="1" cellspacing="2" cellpadding="4">
<tr>
<td valign="TOP" bgcolor="#000080"><font color="#FFFFFF"
size="+1">Nová zpráva</font></td>
</tr>
<tr>
<td bgcolor="#C0C0C0">

```

```

<form action="<%wb_url%>/přidej.htw" method="POST">
<table border="0" cellspacing="4" cellpadding="0">
<tr>
    <td valign="top">autor:</td>
    <td valign="top"><input type="Text" name="autor" size="30"
maxlength="30"></td>
</tr>
<tr>
    <td valign="top">nadpis:</td>
    <td valign="top"><input type="Text" name="nadpis" size="30"
maxlength="50"></td>
</tr>
<tr>
    <td valign="top">text zprávy:</td>
    <td valign="top"><textarea name="text" cols="30" rows="10"
wrap="VIRTUAL"></textarea></td>
</tr>
<tr>
    <td valign="top">platnost dní:</td>
    <td valign="top"><input type="Text" name="platnost" value="14"
size="10" maxlength="10"></td>
</tr>
<tr>
    <td valign="top">&nbsp;</td>
    <td valign="top"><input type="Submit" value="Odeslat"></td>
</tr>
</table>
</form>
</td>
</tr>
</table>

</body>
</html>

```

Jak lze vidět z výpisu, vložené údaje bude zpracovávat šablona **Přidej**.

Proměnná  
wb\_url

Všimněme si údaje v parametru ACTION v tagu FORM:

```
<%wb_url%>/přidej.htw
```

Vidíme, že při zpracování této šablony bude do tohoto parametru vložena hodnota `wb_url`. `wb_url` je proměnná, kterou definuje Internet klient a která obsahuje alias adresáře s CGI skripty, název použitého Internet klienta, název databáze a název aplikace. Pokud bychom pro zpracování této šablony použili CGI klienta, pokud by alias adresáře s CGI skripty byl "cgi-bin" a pokud by se databáze jmenovala "Doprava603", pak proměnná `wb_url` bude mít hodnotu

```
/cgi-bin/wbcgi.exe/Doprava603/Vývěska
```

a hodnota parametru ACTION bude

```
/cgi-bin/wbcgi.exe/Doprava603/Vývěska/přidej.htm
```

Výhodou tohoto způsobu zápisu je to, že proměnná `wb_url` je nastavována podle aktuálních údajů o aliasu, Internet klientovi, názvu databáze a názvu aplikace, a tedy tuto šablonu v nezměněné podobě lze provozovat najednou pomocí různých Internet klientů, dokonce pomocí různých webserverů. Po přejmenování aplikace, či po importu aplikace do jiné databáze, lze provozovat šablonu v této nezměněné podobě. Podrobnosti o proměnných nastavovaných Internet klientem naleznete v elektronické nápovědě.

### Šablona Přidej

Nyní vytvoříme šablonu, která bude zpracovávat údaje zadané do výše uvedeného formuláře.

Vytvoříme nový WWW objekt šablonu a vytvoříme novou komponentu typu **SQLStatement – INSERT** (chceme vložit do tabulky **Zprávy** údaje zadané ve vstupním formuláři **Formulář**). Pro vytvoření příkazu `INSERT` máme k dispozici Průvodce. V jednotlivých krocích Průvodce postupně zadáme:

- nový záznam bude příkaz `INSERT` vytvářet v tabulce **Zprávy**
- příkaz `INSERT` bude nastavovat hodnoty sloupců **Nadpis**, **Vytvořeno kým**, **Platnost dní** a **Zpráva**
- hodnoty jednotlivých sloupců budou následující:

Nadpis	"%nadpis%"
Vytvořeno kým	"%autor%"
Platnost dní	str2int("%platnost%")
Zpráva	"%text%"

- dále zadáme, že chceme přečíst nově vytvořený záznam pod jménem "nový", přičemž pro vyhledání nového záznamu mají být použity sloupce "datum vytvoření" a "čas vytvoření".

Tím je komponenta `INSERT` vytvořena. Pomocí příkazu **Editovat šablonu** přejdeme k editaci šablony. V šabloně pouze necháme vypsát nově vytvořenou zprávu a požádáme uživatele, aby se vrátil do seznamu všech zpráv na vývěsce. Text šablony může vypadat např. takto:

```
<%wbc%>
SQLStatement: nový(`datum vytvoření`,`čas vytvoření`)
+ INSERT INTO Zprávy(nadpis,`platnost dní`,`vytvořeno kým`,zpráva)
+ VALUES ("%nadpis%",str2int("%platnost%"),"%autor%", "%text%")
<%/wbc%>

<html>
<head>
<title>Nová zpráva vytvořena</title>
</head>
```

```
</body>
<h1>Nová zpráva</h1>

<dl>
<dt>od <%nový.`vytvořeno kým`%>: <b><%nový.nadpis%></b>
(<%nový.`datum vytvoření`%>)
<dd><%nový.zpráva%>
</dl>
<p>
byla vytvořena.
<p>
Přejděte do <a href="<%wb_url%>/seznam.htm">seznamu zpráv na
vývěsce</a>.
</body>
</html>
```

#### Další informace

Zde jsme uvedli jenom základní informace o vývoji webovského rozhraní aplikace, navíc v některých případech velmi stručně. Podrobnější popis všech příkazů, které lze použít v šablonách, a všech komponent konektoru můžete nalézt v elektronické nápovědě k **Internet klientům** a k **WinBase602**. V ní jsou tyto údaje zpracovány do podoby referenční příručky, přičemž každé téma je popsáno do všech podrobností. V nápovědě jsou také části, které podrobně popisují určité programátorské obraty, např. použití komponenty INSERT s následným přečtením nově vytvořeného záznamu.

Na distribučním médiu naleznete také příklady aplikací s webovským rozhraním. U každé aplikace je uveden její stručný popis. Další příklady a informace lze najít na Internetu v Knihovně informací WinBase602 na webserveru firmy **Software602 a.s.**, který má adresu <http://www.software602.cz>.

## Informace pro pokročilé

V této části kapitoly jsou uvedena témata, jejichž znalost není nutná pro vývoj jednoduchého webovského rozhraní aplikace. Při vývoji složitějšího rozhraní se však s těmito tématy musíme seznámit.

Jsou zde soustředěny poznámky o použití schématu aplikace, o editaci WWW objektů externím editorem, informace o právech a bezpečnosti a popis WBS selektorů – WWW objektů, které umožňují umístit do HTML formuláře více tlačítek typu SUBMIT.

## Využití schémat s WWW objekty

V rozsáhlejších aplikacích lze získat přehled o vztazích mezi WWW objekty pomocí schémat. Ve schématech jsou mezi WWW objekty zobrazovány dva druhy vztahů:

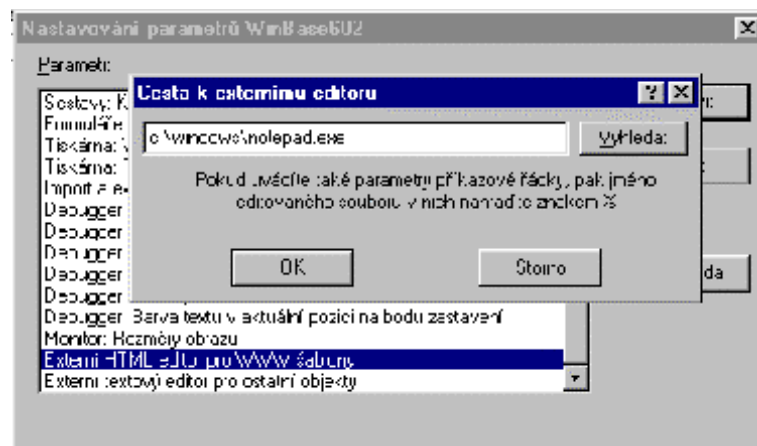
1. černou šipkou je zachycen vztah mezi konektorem a šablonou, která je z tohoto konektoru volána, přičemž šipka vychází z konektoru a končí v šabloně
2. modrou šipkou je zachycen vztah mezi šablonou a WWW objektem, který udává, že z dané šablony lze volat daný WWW objekt (daný WWW objekt je uveden v tagu `<FORM ACTION="...">`, nebo v tagu `<A HREF="...">`); tak lze získat přehled o tom, které WWW objekty jsou volány z dané šablony; šipka vychází z šablony a končí ve WWW objektu

## Editace externím editorem

Po editaci zdrojových textů WWW objektů je k dispozici interní textový editor **WinBase602**, který mimo jiné umí vybarvit jednotlivá slova podle syntaxe dané části textu. Další možností je pro editaci zdrojových textů WWW objektů použít libovolný textový editor – tato možnost se označuje jako externí editace, nebo editace externím editorem.

Nastavení  
externího editoru

Externí editor je třeba nastavit v globálních parametrech WinBase602 – položka **Externí HTML editor pro WWW šablony**. Tento dialog vyvoláte příkazem **Parametry** z menu **Nástroje** a výběrem výše uvedeného nastavení v seznamu všech parametrů.

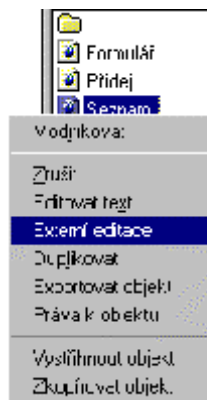


Je třeba, aby externí editor vytvářel text v kódování ASCII bez jakýchkoliv formátovacích příznaků (např. Notepad – Poznámkový blok). Externí editace probíhá tak, že **WinBase602** vytvoří v adresáři TEMP dočasný soubor s příponou HTM a do něj zkopíruje zdrojový text vybraného WWW objektu. Následně **WinBase602** spustí externí editor a předá mu na příkazové řádce jméno dočasného souboru. V externím editoru upravujete

zdrojový text, v této době je **WinBase602** nepřístupná, nelze s ní pracovat. Až ukončíte editaci textu v externím editoru, uložíte změněný text do stejného dočasného souboru a ukončíte externí editor. Poté **WinBase602** zaktualizuje zdrojový text daného WWW objektu, smaže dočasný soubor a zpřístupní hlavní okno pro další práci.

**Vyvolání editace**

Editaci WWW objektu externím editorem vyvoláte z Řídícího panelu příkazem **Externí editace** v kontextovém menu daného WWW objektu.



## Práva a bezpečnost

Aplikace s webovským rozhraním jsou provozovány pomocí **WinBase602 Internet klientů** (podrobnosti v *Příručce správce* v kapitole *Přístup pomocí Internet klienta*). Internet klienti provádějí veškeré akce s databází pomocí uživatele **\_\_web** (dvě podtržítka na začátku).

**Uživatel \_\_web**

Tento uživatel musí v databázi existovat (v nově vytvořené databázi neexistuje, správce jej musí vytvořit). Jeho heslo může být libovolné, může být také libovolně často měněno.

Tento uživatel musí mít alespoň následující práva:

**Práva k objektům**

- právo používat WWW objekty v dané aplikaci
- právo používat objekty typu Program, které jsou využívány ve WWW objektech
- právo používat databázové objekty, se kterými se pracuje v procedurách a funkcích obsažených v objektech typu Program, které jsou využívány ve WWW objektech
- právo používat tabulky, se kterými manipulují SQL příkazy obsažené ve WWW objektech
- odpovídající práva k datům v těchto tabulkách (v případě SQL příkazu SELECT se jedná o právo číst dané sloupce, v případě INSERTu je to právo vytvářet nové záznamy, v případě příkazu UPDATE musí být nastaveno právo přepisovat dotčené sloupce, při použití příkazu DELETE musí mít uživatel nastaveno právo mazat záznamy z tabulky)

Standardní způsob, jak přidělit uživateli `__web` tato práva, je založen na využití mechanismu rolí. V aplikaci vývojář vytvoří roli (objekt typu role), např. `Web_user`. Dále vývojář přidělí této roli všechna práva, která jsou potřebná pro provoz webovského rozhraní jeho aplikace. Protože je role exportována a importována spolu s ostatními objekty aplikace, správce provozní databáze musí po importu aplikace pouze obsadit uživatele `__web` do výše uvedené role.

### Bezpečnost

Potenciální narušitel, který chce získat údaje uložené v určité databázi v aplikaci, která má webovské rozhraní, ví pouze to, že v této databázi existuje uživatel `__web`, který má všechna práva pro provedení operací popsanych ve WWW objektech.

Aby se mohl přihlásit do databáze pomocí tohoto uživatelského účtu, musel by znát heslo tohoto uživatele. Tento uživatel je však vytvářen správcem dané databáze, který také vybírá a nastavuje heslo tohoto uživatele. Heslo uživatele `__web` tedy kromě správce databáze nezná žádný jiný uživatel. Správce navíc může libovolně často měnit heslo tohoto uživatele `__web`, dokonce za plného provozu.

Tyto okolnosti znemožňují využití uživatelského účtu `__web` k provádění nepovolených operací s daty v databázi.

## Použití selektorů

V určitých situacích může být užitečné mít ve svém HTML formuláři více tlačítek typu SUBMIT. Na druhou stranu jeden formulář provede po stlačení tlačítka typu SUBMIT vždy tu samou akci, spustí ten samý WBC konektor. Zde si vysvětlíme, jak umístit do formuláře více tlačítek typu SUBMIT a jak zajistit, aby se po jejich stlačení provedly různé akce v závislosti na tom, které tlačítko bylo stlačeno.

Abychom zajistili výše zmíněné chování formuláře, musíme tlačítka typu SUBMIT ve formuláři pojmenovat stejným jménem. Tzn. všechny tagy `<INPUT TYPE="SUBMIT">` musí mít v parametru `NAME` stejné jméno. Toto jméno musí být složeno z písmen (i s háčky a čárkami), z číslic a ze znaku podtržení `'_'`.

WBS selektor je databázový objekt kategorie WWW objekty s následujícím obsahem:

```
[Submit]
SubmitName=jméno-tlačítek-typu-submit
[WBClist]
hodnota-tlačítka-číslo-1=název-WBC-konektoru-číslo-1
hodnota-tlačítka-číslo-2=název-WBC-konektoru-číslo-2
...
hodnota-tlačítka-číslo-n=název-WBC-konektoru-číslo-n
```

*Jméno-tlačítek-typu-submit* je ono jméno, kterým jsme pojmenovali všechna tlačítka typu SUBMIT v našem formuláři. Ve formuláři máme *n* tlačítek typu SUBMIT, *hodnota-tlačítka-číslo-i* je hodnota parametru `VALUE` *i*-tého tlačítka typu SUBMIT (tzn. to, co je na tlačítku zobrazeno v WWW browseru). *Název-WBC-konektoru-číslo-i* je název WBC konektoru, který má být použit, pokud uživatel stlačí *i*-té tlačítko. Může být zde uveden

pouze WBC konektor z té samé aplikace, ve které je tento WBS selektor. Kromě konektoru lze zde uvést i název HTW šablony (opět ze stejné aplikace). Stejný WBC konektor můžeme uvést u více tlačítek (i když tato situace odporuje tomu, proč jsme multitlačítkový formulář vytvářeli).

V parametru `ACTION` tagu `FORM` našeho multitlačítkového formuláře uvedeme odkaz na námi vytvořený WBS selektor. Syntaxe tohoto odkazu se řídí identickými pravidly, jako syntaxe odkazu na WBC konektor. V tomto ohledu se můžeme na WBS selektory dívat jako na speciální typ konektoru, který obsahuje pouze jakousi tabulku s přiřazením tlačítek typu `SUBMIT` a WBC konektorů, či HTW šablon.

Návrhář  
selektorů

K vytváření a změnám WBS selektorů používáme Návrhář selektorů. Popis jednotlivých ovládacích prvků je uveden v nápovědě k jednotlivým prvkům i k celému dialogu.