

Používání 4D Databáze na Webu

Obsah

Obsah

OBSAH.....	I
1. ÚVOD.....	1
1.1 VÍTEJTE.....	1
1.2 JAK PRACOVAT S KURZEM	1
1.3 SCÉNÁŘ.....	1
1.4 KONVENCE	5
1.5 PŘÍKAZY	6
1.6 KONVENCE NÁZVŮ PRO PROMĚNNÉ.....	6
1.7 AKTIVNÍ OBJEKTY.....	7
1.8 KONVENCE NÁZVŮ PRO METODY A PŘÍKAZY A FUNKCE	7
2. SÍŤ A TCP/IP	9
2.1 INTERNET, INTERNET, SÍŤ, INTRANET...UF?	9
2.2 TCP/IP — UNIVERZÁLNÍ JAZYK	9
2.3 OTEVŘENÉ STANDARTY	10
2.4 VŠELIDOVĚ STANOVENÉ STANDARTY (RFC).....	10
2.4 NEDOKONALÝ SOFTWARE	10
3. ZÁKLADY INTERNETU	11
3.1 INSTALACE TCP/IP.....	11
3.2 KONFIGURACE TCP/IP	11
3.4 VÝHODY SLUŽEB NÁZVŮ DOMÉN.....	12
3.5 Co ZNAMENAJÍ ČÍSLA V IP ADRESÁCH?.....	12
3.6 TRVALÉ SPOJENÍ	13
3.7 "V6" BUDOUCNOST IP ADRES.....	14
4. DATABÁZE NA WEB ZA 5-15 MINUT.....	15
4.2 VYTVOŘENÍ JEDNODUCHÉ DATABÁZE	16
4.3 FORMULÁŘE	16
4.4 ZPROVOZNĚNÍ APLIKACE	17
4.5 PŘÍSTUP K DATABÁZI Z WEBOVSKÉHO PROHLÍŽEČE	18
4.6 ZACHÁZENÍ S DATY PŘES PROHLÍŽEČ	20
5. 4D A WEB.....	21
5.1 TRANSAKČNĚ ORIENTOVANÉ APLIKACE	21
5.2 INTEGRACE VĚTŠÍ SÍTĚ.....	21
5.3 INTERAKTIVITA.....	22
5.4 TRADIČNÍ ŘÍZENÍ WEB.....	22
5.5 FRONT END PRO SQL DATABÁZE.....	24
6. PŘEHLED RYSŮ	25
6.1 ÚVOD.....	25
6.2 WEB PŘIPOJENÍ	25
6.3 AUTOMATICKÝ PŘEVOD FORMULÁŘŮ.....	25

Používání 4D Databáze na Webu

Obsah

6.4.	ZPĚTNÉ VOLÁNÍ METOD 4D	25
6.5.	PROVÁZÁNÍ PROMĚNNÝCH	25
6.6.	ZAMYKÁNÍ ZÁZNAMŮ	25
6.7.	AUTOMATICKÉ STRÁNKOVÁNÍ	25
6.8.	EXISTUJÍCÍ HTML	25
6.9.	ODESLÁNÍ HTML NEBO JAVASCRIPT PŘÍMO	26
7.	KONFIGURACE	27
7.1.	ÚVOD.....	27
7.2.	NASTAVENÍ PŘEDVOLEB STARTU WEB SERVERU	27
7.3.	LICENZOVÁNÍ	30
7.4.	SPUŠTĚNÍ WEB SERVER	31
7.5.	CO SE STANE PŘI STARTU WEB SERVERU	32
7.6.	NASTAVENÍ ČASU VYPRŠENÍ PŘIPOJENÍ WEB PROCESU	32
7.7.	MINIMÁLNÍ POŽADAVKY	34
8.	ZÁKLADY ROZHRANÍ WEB	35
8.1.	ÚVOD.....	35
8.2.	PŘIDÁNÍ JEDNODUCHÉHO ROZHRANÍ	35
9.	FORMULÁŘE WEB AWARE.....	39
9.1.	ÚVOD.....	39
9.2.	AUTOMATICKÉ PŘEVODY 4D.....	39
9.3.	HTML FORMULÁŘE.....	39
9.4.	PŘEVODY OBJEKTŮ 4D FORMULÁŘŮ.....	39
9.5.	OMEZENÍ PRO PŘEVODY 4D FORMULÁŘŮ.....	41
9.6.	OMEZENÍ PROHLÍŽEČŮ.....	41
9.7.	UNIVERZÁLNÍ OMEZENÍ	43
9.8.	VYTVOŘENÍ VÝSTUPNÍHO FORMULÁŘE	46
9.9.	VYTVOŘENÍ ZADÁVACÍHO FORMULÁŘE/FORMULÁŘE PRO JEDEN ZÁZNAM	51
9.10.	VÍCESTRÁNKOVÉ FORMULÁŘE	54
9.11.	STYLY TLAČÍTEK	54
10.	AUTOMATICKÉ AKCE A METODY FORMULÁŘE.....	55
10.1.	ÚVOD.....	55
10.2.	AUTOMATICKÉ AKCE	55
10.3.	AKCE TLAČÍTEK, KTERÝM BY JSTE SE MĚLI VYHNOUT	55
10.4.	TISKY	55
10.5.	PODPOROVANÉ METODY.....	56
10.6.	DŮLEŽITÉ DETAILS O METODÁCH OBJEKTŮ A AUTOMATICKÝCH TLAČÍTKÁCH AKCÍ	59
11.	V6: JEDINEČNÝ WEB SERVER	60
11.1.	ÚVOD.....	60
11.2.	CO JE WEB SERVER.....	60
11.3.	CGI.....	60
11.4.	STAV KTERÝ NEUDRŽUJE INFORMACE	61
11.5.	COOKIES	62
11.6.	V6 ŘÍZENÍ STAVU	62
11.7.	KDY POUŽÍT TRADIČNÍ WEB SERVER	62

Používání 4D Databáze na Webu

Obsah

12.	WEB PŘIPOJENÍ PODROBNĚ	63
12.1.	ÚVOD	63
12.2.	SPUŠTĚNÍ PROCESU	65
12.3.	PŘI WEB SPOJENÍ	65
12.4.	ZÁHLAVÍ I	66
12.5.	ID OBSAHU A PODOBSAHU	66
12.6.	NEKLEPAT NA TLAČÍTKO ZPĚT!	67
13.	BEZPEČNOST A UŽIVATELSKÝ PŘÍSTUP	70
13.1.	HESLA	70
13.2.	PŘÍSTUP UŽIVATELE A SKUPIN	71
13.3.	UŽIVATELSKÉ ÚČTY	71
13.4.	UŽIVATELSKÉ STRÁNKY	71
14.	PŘI WEB SPOJENÍ	72
14.1.	ÚVOD	72
14.2.	PŘÍCHÁZEJÍCÍ INFORMACE	72
14.3.	VYTVÁŘENÍ OZVĚNY INFORMACÍ	72
14.4.	HLAVIČKA HTTP: CO TO VŠECHNO ZNAMENÁ ?	74
14.5.	URL	79
14.6.	VYTVÁŘENÍ STRUKTURY ŘÍZENÍ AKCÍ ZALOŽENÝ NA URL	80
14.7.	ZAMYKÁNÍ ZÁZNAMŮ	87
14.8.	POUŽITÍ DAT K AUTOMATIZACI PŘÍSTUPU K DATABÁZI	89
14.9.	ZPĚTNÁ VOLÁNÍ	93
14.10.	JINÉ ÚLOHY PRO METODU PŘI WEB SPOJENÍ	103
15.	PŘÍZPŮSOBNÉ RUTINY PRO WEB	104
15.1.	ÚVOD	104
15.2.	URČITÉ PŘÍKAZY A WEB	104
15.3.	SLEDOVÁNÍ TYPU KLIENTA	107
15.4.	ZMÍNKA O PŘÍKLADU U KADETA.	111
15.5.	PŘÍPRAVA ACÍ VIDEO PRO WEB	111
16.	HTML	112
16.1.	ÚVOD	112
16.2.	HTML: HISTORIE A ÚČEL	112
16.3.	HTML: CO TO NENÍ	112
16.4.	VYTVÁŘENÍ HTML	113
16.5.	SYNTAXE HTML	114
16.6.	TAGY ZNAKOVÝCH STYLŮ	115
16.7.	TAGY ODSTAVŮ	115
16.8.	TAGY DOKUMENTŮ	115
16.9.	TAGY OBRÁZKŮ	116
16.10.	TAGY TABULEK	117
16.11.	URL	118
16.12.	FORMULÁŘE	122
16.13.	ODESÍLÁNÍ FORMULÁŘŮ DO 4D	126
16.14.	AUTOMATICKÉ PROVEDENÍ METODY	128
17.	ODESÍLÁNÍ HTML	129

Používání 4D Databáze na Webu

Obsah

17.1.	SEND HTML FILE	129
17.2.	VNOŘOVANÉ ODKAZY NA SOUBOR	136
17.3.	ZMĚNA CESTY K HTML	139
17.4.	VNOŘENÍ BLOKŮ HTML.....	140
17.5.	VÁZÁNÍ PROMĚNNÝCH	145
18.	UKLÁDÁNÍ HTML V ZÁZNAMECH DATABÁZE	148
18.1.	POZADÍ	148
18.2.	UKLÁDÁNÍ DAT V ZÁZNAMECH	149
18.3.	O METODĚ HTML_TBLOCK	154
18.4.	OPTIMALIZACE RUTINY	155
19.	SPECIÁLNÍ POZNÁMKY KE KÓDOVÁNÍ ZNAKŮ V HTML	156
19.1.	POZNÁMKA O NÁZVECH SOUBORŮ.....	156
20.	PRÁCE S JAVASCRIPTY	158
20.1.	CO JE JAVASCRIPT?	158
20.2.	JE JAVASCRIPT ČÁSTÍ JAVA?	158
20.3.	PROČ UŽÍVAT JAVASCRIPT?.....	158
20.4.	UŽITÍ JAVASCRIPT V SYSTÉMECH 4D WEB	158
20.5.	O TĚCHTO CVIČENÍCH.....	159
20.6.	4D A JAVASCRIPT.....	159
20.7.	ODESLÁNÍ JAVASCRIPT	159
20.8.	JAK VYPADÁ JAVASCRIPT.....	159
20.9.	UKLÁDÁNÍ JAVASCRIPT	160
20.10.	PŘIDÁNÍ FORMULÁŘŮ DO TABULKY.....	160
20.11.	VÁŠ PRVNÍ JAVASCRIPT.....	161
20.12.	RUTINA TVORBY JAVASCRIPT	161
20.13.	PŘIDÁNÍ JAVASCRIPT DO FORMULÁŘE	163
20.14.	TESTOVACÍ RUTINA	164
20.15.	OBJEKTY JAVASCRIPT	165
20.16.	ZMĚNY 4D POLÍ	165
20.17.	ZACHYCENÍ ODESLÁNÍ FORMULÁŘE	167
20.18.	OTEVŘENÍ NOVÉHO OKNA PRO NOVÝ PROCES	167
21.	TRIGGERY	169
20.1.	ÚVOD.....	169
20.2.	CO JE TRIGGER?	169
20.3.	JEDNODUCHÝ TRIGGER	169
20.4.	PSÁNÍ TRIGGERŮ PRO LIBOVOLNÝ INTERFACE	170
20.5.	SPECIFICKÉ POLOŽKY WEB.....	170
22.	SHRNUTÍ PŘÍKAZŮ WEB.....	171
22.1.	SEND HTML FILE (SOUBOR CESTA)	171
22.2.	SET HTML ROOT (CESTA)	171
22.3.	SET WEB DISPLAY LIMITS(MNOŽSTVÍ;STRÁNEK;IKONA)	171
22.4.	SET WEB TIME-OUT (SEKUND)	171
22.5.	START WEB SERVER ()	171
22.6.	STOP WEB SERVER ().....	171
22.7.	MAC TO ISO (MAC ASCII) : LATIN-1, LATIN 2, WIN 1250 KÓDOVÁNÍ TEXTU	171

Používání 4D Databáze na Webu

Obsah

23. DEFINICE TERMÍNŮ.....	172
CGI (COMMON GATEWAY INTERFACE)	172
FTP (FILE TRANSFER PROTOCOL)	172
HTTP (HYPERTEXT TRANSFER PROTOCOL)	172
HTML (HYPERTEXT MARKUP LANGUAGE)	172
IIS (MICROSOFT INTERNET INFORMATION SERVER)	172
INTERNET	172
INTRANET	172
JAVASCRIPT	172
POP3 (POST OFFICE PROTOCOL 3)	172
SMTP (SIMPLE MAIL TRANSFER PROTOCOL)	173
SSL (SECURE SOCKET LAYER)	173
24. KONTAKTNÍ INFORMACE.....	174
DATAWAVE	174
INTERNET TOOLKIT (ITK)	174
NETLINK/4D	174
NETWINGS	174
PDM INTERNET TOOLS	175
TCP TOOLKIT	175
WEB SERVER 4D	175
25. DALŠÍ ZDROJE	176
24.1. KNIHY	176
24.2. ELEKTRONICKÉ PŘÍRUČKY	176

4th Dimension

Příručka školení

4th Dimension a Web

Verze 6.0.5

Kurz připraven:

Jméno	E-Mail	Web
David Adams	dpadams@aloha.net	http://www.aloha.net/~dpadams/
Kent Wilbur	kent@acius.com	

Překlad a upraveno:

Jaroslav Macháček	Inforce@mbox.vol.cz	http://www.inforce.cz
-------------------	--	---

Používání 4D Databáze na Webu

Úvod

1. Úvod

1.1 Vítejte

Tento kurz je navržen pro zkušené programátory, obeznámené již s 4th Dimension (4D). Kurz uvádí technologie specifické pro přístup prohlížečů Web k 4D databázím. Tato možnost znamená, že můžete použít standardní prohlížeče WEB jako alternativu připojení místo klienta (4D Client) v klient/server architektuře buď ve vaší vnitřní síti, nebo na internetu, nebo pro obojí.

1.2 Jak pracovat s kurzem

Kurz je navržen na dva dny vyučovací doby.

Je konán následujícími metodami:

- ❖ Instruktor diskutuje některé rysy a demonstruje jejich použití pomocí promítání z počítače na plátno.
- ❖ Účastníci kurzu provádějí podle této příručky příklady, které pokrývají témata diskutovaná instruktorem.
- ❖ Instruktor odpovídá na dotazy a je-li potřeba věnuje se individuálně jednotlivým účastníkům při řešení příkladů.
- ❖ Pro kontroly jsou k materiálům kurzu jsou přiloženy řešené příklady na disketách nebo CD

1.3 Scénář

Během celé práce budete upravovat zjednodušenou databázi pro distribuci videokazet. Je to táž databáze, která byla vytvořena v předchozích kurzech. Tato databáze obsahuje zákazníky, faktury, položky faktur a seznam všech prodáváných produktů.

Databáze je pojmenována podle fiktivní společnosti „ACI VIDEO“. Na konci tohoto kurzu bude obsahovat následující tabulky a sloupce (pole) uvedené níže.

[zDialogy]		
Název pole	Typ	Vlastnosti
Povinnné	Logické	

Používání 4D Databáze na Webu

Úvod

[Zákazníci]

Název pole	Typ	Vlastnosti
IDZákazníka	Alfa 10	Nutný vstup; Pouze zobrazit; Indexované; Jedinečné
Jméno	Alfa 20	Indexované
Příjmení	Alfa 20	Indexované
Firma	Alfa 25	Indexované; Nutný vstup
Adresa	Alfa 25	
Město	Alfa 20	Indexované
Stát	Alfa 2	Indexované
PSC	Alfa 10	Indexované
Telefon	Alfa 10	
PlátceDaně	Logické	
Důležitý	Logické	
CelkovéProdeje	Real	
DatumVytvoření	Datum	Pouze zobrazit; Neviditelné
ČasVytvoření	Čas	Pouze zobrazit; Neviditelné
DatumÚpravy	Datum	Pouze zobrazit; Neviditelné
ČasÚpravy	Čas	Pouze zobrazit; Neviditelné

[ObecnéInformace]

Název pole	Typ	Vlastnosti
NázevFirmy	Alfa 40	Nutný vstup, Indexované, Jedinečné
Adresa	Text	
Město	Alfa 22	
Stát	Alfa 2	
PSC	Alfa 9	
Telefon	Alfa 10	

[Faktury]

Název pole	Typ	Vlastnosti
IDZákazníka	Alfa 10	Nutný vstup, Neměnné; Indexované; Jedinečné
ČísloFaktury	Long Integer	Indexované; Jedinečné
DatumFaktury	Datum	Indexované
FakturaCelkem	Real	
Placeno	Logické	
ZpůsobPlatby	Alfa 15	
DatumVytvoření	Datum	Pouze zobrazit; Neviditelné
ČasVytvoření	Čas	Pouze zobrazit; Neviditelné
DatumUpravení	Datum	Pouze zobrazit; Neviditelné
ČasUpravení	Čas	Pouze zobrazit; Neviditelné

Používání 4D Databáze na Webu

Úvod

[PoložkyFaktur]

Název pole	Typ	Vlastnosti
ČísloFaktury	Long Integer	Indexované
IDZbožíProdukty	Long Integer	Indexované
Neužito	Logické	Neviditelné
Cena	Real	
Množství	Integer	
JednCena	Real	
CenaCelkem	Real	

[Produkty]

Název pole	Typ	Vlastnosti
IDZboží	Alfa 10	Nutný vstup; Neměnné; Indexované; Jedinečné
Název	Alfa 30	Indexované
Cena	Real	Indexované
Rok	Integer	Indexované
Kategorie	Alfa 15	Indexované
DatumVytvoření	Datum	Pouze zobrazit; Neviditelné
ČasVytvoření	Čas	Pouze zobrazit; Neviditelné
DatumUpravení	Datum	Pouze zobrazit; Neviditelné
ČasUpravení	Čas	Pouze zobrazit; Neviditelné

[zNavracenáČísla]

Název pole	Typ	Vlastnosti
NázevSekvence	Alfa 20	Indexované
Hodnota	Long integer	

[zSekvence]

Název pole	Typ	Vlastnosti
NázevSekvence	Alfa 20	Indexované
Hodnota	Long integer	

[zStruktura]

Název pole	Typ	Vlastnosti
ČísloTabulky	Long Integer	Nutný vstup, Pouze zobrazit, Indexované
ČísloPole	Long Integer	Nutný vstup, Pouze zobrazit, Indexované
PlatnýNázev	Alfa 31	
NázevZástupce	Alfa 31	
PořadíZástupce	Integer	
VložitVlastníHledání	Logické	
VložitVlastníTřídění	Logické	
URL_PřidatPovoleno	Logické	
URL_ZobrazitPovoleno	Logické	
URL_UpraviPovoleno	Logické	

Používání 4D Databáze na Webu

Úvod

[zUživatelé]

Název pole	Typ	Vlastnosti
IDUživatele	Long integer	Nutný vstup, Indexované, Jedinečné
NázevUživatele	Alfa 31	Nutný vstup
PlatnýNázev	Alfa 42	
Adresa	Text	
Město	Alfa 22	
Stát	Alfa 2	
PSČ	Alfa 9	
Telefon	Alfa 10	
DatumPosledníhoPřipojení	Datum	
ČasPosledníhoPřipojení	Čas	
Trvání	Čas	
ČítačTikůPřipojení	Longint	Neviditelné
Připojení	Integer	
Připojen	Logické	
OdvolatPřístup	Logické	
PředvolbyDotazuTabulky	Text	Neviditelné
PředvolbyTříděníTabulky	Text	Neviditelné
VíceZobrazení	Text	Neviditelné

[zHTML Bloky]

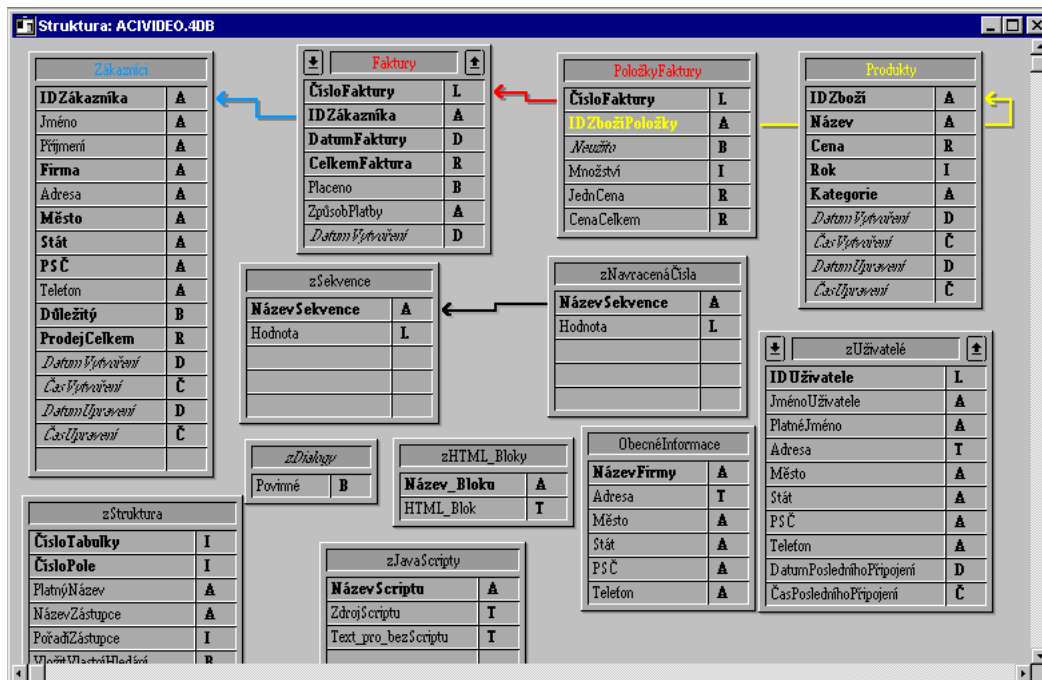
Název pole	Typ	Vlastnosti
Název_Bloku	Alfa (80)	Indexované, Jedinečné, Nutný vstup
HTML_Blok	Text	Nutný vstup

[zJavaScripty]

Název pole	Typ	Vlastnosti
NázevScriptu	Alfa (80)	Indexované, Jedinečné, Nutný vstup
ZdrojScriptu	Text	Nutný vstup
Text pro bezScriptu	Text	

Používání 4D Databáze na Webu

Úvod



1.4. Konvence

- ❖ “Programování” a “tvorba kódu” jsou synonyma pro instrukce, které budete vkládat.

- ❖ To co budete zapisovat je v tomto tvaru.

Příklad: Napište Johnson do pole Příjmení.

- ❖ Speciální klávesy na klávesnici jsou uvedeny následovně:

Stiskněte Enter.

- ❖ Jestliže je požadováno, abyste stiskli více než jednu klávesu současně je tento požadavek zapsán následovně:

Stiskněte CTRL + Shift + 3.

Vysvětlení: Stiskněte současně klávesy CTRL, Shift a 3.

Macintosh™ ekvivalent bude okamžitě následovat po Windows™ v kurzívě, jak je ukázáno níže.
Stiskněte: Ctrl + Shift + 3 (*Command + Shift + 3*)

- ❖ Volba položky z nabídky je popsána následovně:

Zvolte Soubor → Otevřít (Ctrl + O) (*□ + O*)

Vysvětlení: Z nabídky Soubor, zvolte položku Otevřít. Na Windows™ stiskněte Control + O.
Na Macintosh™ stiskněte Command + O.

- ❖ Položky 4D kódu jsou zobrazovány stejně jako jsou v 4D Editoru metod. Například:

Používání 4D Databáze na Webu

Úvod

[Zákazníci]Stát	Pole
ALERT	Příkaz
MyProcedure	Metoda (procedura)

- ❖ Sekce označené “Extra Credit” jsou výběrové. Jestliže skončíte svůj příklad dříve můžete pracovat v sekci “Extra Credit” a naučit se více.
- ❖ HTML a JavaScript příklady jsou formátovány následovně:

```
<SCRIPT LANGUAGE="JavaScript">
<!-- dale kod JavaScript -->
</SCRIPT>

<NOSCRIPT>
<!-- dale zpráva pro prohlizece ne -JavaScript -->
</NOSCRIPT>
```

1.5. Příkazy

V tomto kurzu se naučíte používat následující příkazy z jazyka 4D. Povšimněte si, že některé z těchto příkazů jsou uvedeny pouze v části Extra Credit. Jsou pak určeny pro samostudium.

SEND HTML FILE

SET HTML ROOT

SET WEB DISPLAY LIMITS

SET WEB TIME-OUT

START WEB SERVER

STOP WEB SERVER

Mac to ISO

1.6. Konvence názvů pro proměnné

Pro přehlednost kódu je důležitá konvence názvů proměnných pro metody a funkce. ACI nezavedla žádnou povinnou konvenci, prezentujeme zde však jednu, která je poměrně široce používána v různých publikacích a kurzech. Všechna písmena jsou malá, kromě L označující LongInteger, kde se může plést l a malé I.

Všechny array (pole v paměti) začínají “a”. Dvoudimenzionální array začínají “a2” nebo “aa”.

Proměnné mají následující předpony , které jsou rovněž používána pro array po písmenu “a”.

Používání 4D Databáze na Webu

Úvod

Typ	Proměnná	1D Array	2D Array
String	s	as	a2s
Text	t	at	a2t
Integer	není	ai	a2i
Longint	L	aL	a2L
Real	r	ar	a2r
Date	d	ad	a2d
Time	h	není	není
Boolean	f	af	a2f
Picture	g	ag	a2g
Pointer	p	ap	a2p
BLOB	o	není	není

1.7. Aktivní objekty

Aktivní objekty jsou téměř vždy číselné. Pro odlišení však pro ně používáme vlastní předpony.

Objekt	Předpona
Tlačítko/Button	b
Zaškrťovací políčko/Checkbox	ck
Teploměr/Thermometer	th
Přepínač/Radio Button	rb1; rb2; rb3; sb1; sb2; sb3, etc.

1.8. Konvence názvů pro metody a příkazy a funkce

Vestavěné příkazy jazyka 4D jsou uváděny následovně :

- ❖ Vše velkými písmeny – pro příkazy vykonávající konkrétní akci jako CANCEL (provádí akci storno).
- ❖ Smíšeně velká a malá písmena – pro příkazy, které vrací hodnotu jako např. Uppercase.

Příkazy se smíšenou konvencí jsou vždy vpravo od operátoru přiřazení.

V žargonu programátorů je tato metoda navracující hodnotu nazývána funkce

Většina programátorů volí následující konvenci. Metody jsou uváděny s velkými písmeny na počátku každého slova „MojeProcedura“. Speciální metody nejčastěji používané začínají „aa“, aby v seznamu metod byly na počátku. Další dělení je pro metody zvláštního účelu:

- ❖ “M_” pro metodu spouštěnou z nabídek
- ❖ “P_” pro metodu začínající nový proces
- ❖ “E_” pro metodu, která je navržena k ručnímu spouštění pomocí příkazu nebo nabídky Execute.

Kromě toho jsme vytvořili následující konvenci pro segmentaci kódu. Vše přímo přiřazené některým účelům (moduly, proměnné, názvy metod, formulářů) začíná následujícími předponami:

- ❖ “ALIAS_” pro komponenty, které zacházejí s tabulkami, nebo zastupují pole.
- ❖ “CHO_” pro metody, které jsou pro výběrové seznamy
- ❖ “GEN_” pro metody, které jsou plně generické
- ❖ “IMPEXP” pro komponenty účastné při exportu a importu

Používání 4D Databáze na Webu

Úvod

- ❖ “INITIALIZE” pro komponenty, které inicializují proměnné nebo procesy
- ❖ “KEY_” pro komponenty, které zachytávají klávesy
- ❖ “LOCK_” pro komponenty, které obsahují zamykání a odemykání záznamů
- ❖ “PROCESS_” pro komponenty účastné při řízení procesů
- ❖ “SPE_” pro komponenty zacházející s uloženými procedurami
- ❖ “WEB_” pro komponenty zacházející s WEB rutinami
- ❖ “WIN_” pro komponenty provádějící manipulaci s okny

Příklady:

- ❖ M_Customers
- ❖ P_Customers
- ❖ sConstants
- ❖ GEN_MyMethod

Když píšete vaše vlastní metody, které jsou funkcemi, je často praktické začínat název písmenem, které reprezentuje typ navrácené hodnoty. Tyto písmena jsou pak stejná jako v tabulce výše.

Používání 4D Databáze na Webu

Sít' a TCP/IP

2. Sít' a TCP/IP

Internet se rozšířil podstatně rychleji, než si jeho tvůrci mohli kdy představit a ještě stále se rozšiřuje. Několik faktorů mělo vliv na toto rozšíření a budou uvedeny dále. Zatímco média nazvala Internet „Informační superdálnice“, byla většina prací na rozvoji provedena pro sítě místní, podnikové. Definujme si několik termínů a pak se podívejme proč byl Internet schopen růst tak rychle a snadno.

2.1. Internet, internet, sít', Intranet...uf?

Dnes existuje mnoho slov pro označení sítí. Aby jsme se vyhnuli zmatkům definujme si několik termínů. Kdysi bylo prosté slovo sít'. Sít' je soubor počítačů, které sdílejí protokol a některá fyzická zařízení, jež jim umožňují komunikovat. Nezáleží na tom jaký je to protokol, nebo zařízení, pokud jsou společná všem propojeným počítačům. Pak přišla „mezisít'“ (internetworking). „internet“ je spojení dvou libovolných oddělených sítí, které mezi sebou komunikují pomocí společného protokolu. Fyzická zařízení jako přemostění, routery jsou použita k vytvoření této mezisítě. Opět nezáleží jaký protokol a jaká zařízení, počítače a kabely sítě používají, dokud mají společný protokol komunikace. Dále již přišla Mezisít' - Internet s velkým písmenem. To znamená mezisít', která vyrostla z ARPANet projektu. Všechny sítě na Internetu spolu komunikují přes protokol TCP/IP. Mohou samozřejmě interně komunikovat přes jiné protokoly (IPX/SPX, AppleTalk, Vines), ale musí použít TCP/IP k vzájemné komunikaci. Naposled intranet se objevil v několika posledních letech, je to určitě míry marketingové heslo, které hledá svůj význam. Obecně je používáno pro podnikovou sít', která používá software pro WEB. Prosté slovo sít' by asi mohlo pokrýt tento význam, ale nevypadalo by tak dobře v propagaci. Zde je přehled termínů:

Síťové termíny	
Termín	Definice
Sít' / Network	Skupina počítačů , která sdílí fyzické propojení a komunikační protokol.
internet nebo internetwork	Skupina sítí , která sdílí fyzické propojení a komunikační protokol.
Internet	Skupina sítí, jež vznikla z ARPANet, používající TCP/IP jako komunikační protokol.
Intranet	Podniková sít' používající Web software k přenosu informací.

2.2. TCP/IP — univerzální jazyk

Jak jsme se zmínili počítače a sítě počítačů spolu mohou komunikovat pokud sdílejí společný protokol a nezáleží na tom jaký (IPX, DECNet, AppleTalk, Vines, atd.). TCP/IP je dohoda pro společnou komunikaci vhodnou pro Internet – je to druh univerzálního jazyka nebo počítačové esperanto. Soukromé internety mohou užívat libovolný protokol či protokoly, ale nutně musí použít TCP/IP jestliže chtějí být částí Internetu.

TCP/IP je soubor protokolů, dva z nejdůležitějších jsou TCP (Transmission Control Protocol) a IP (Internet Protocol). Je zde zahrnuto ještě více protokolů a celý tento soubor je dohromady nazýván TCP/IP. Tyto protokoly byly navrženy k tomu, aby určily jak má být strukturována informace v paketech, jak má být rozeznána a znovu sestavena *bez ohledu na určitou fyzickou sít', hardware nebo software*. TCP/IP pracuje na různých typech fyzických sítí (Ethernet, modemy, vlákna, Local talk), různých operačních systémech (VMS, UNIX, OS2, Windows NT, Mac OS) a typech počítačů (VAX, AS400, Alpha, Intel PC's, Macintosh). Tato univerzálnost a přenositelnost podstatně usnadnila zesíťování různých typů počítačů, sítí a operačních systémů do jedné světové sítě.

Používání 4D Databáze na Webu

Sít' a TCP/IP

Největší výhodou TCP/IP, která se hodí obzvláště vývojářům aplikací je, že většina z úloh komunikace je prováděna přímo v TCP/IP. Např. Web server běží nad TCP/IP, takže vývojáři pro Web nemusí pracovat přímo s komunikačním protokolem, ale pouze s tím jak Web server a prohlížeče komunikují s TCP/IP. Tento rys podstatně zjednodušil vytváření aplikací a usnadnil jejich přenos mezi jednotlivými operačními systémy. V tomto smyslu může být TCP/IP považováno za samostatnou platformu.

2.3. Otevřené standarty

Jiným důvodem pro úspěch Internetu je, že software je (většinou) založen na tzv. *otevřených standardech*. TCP/IP není vytvářeno a řízeno žádným konkrétním dodavatelem nebo konsorciem dodavatelů. To platí o všech ostatních nejrozšířenějších službách internetu včetně FTP, Telnet, HTTP, SMTP, NNTP a POP3. Dodavatelé vytvářejí a prodávají specifické implementace na jednotlivé platformy, ale neřídí jejich samotné vytváření. Např. můžete vytvořit HTTP (Web) server bez nutnosti vyžádání si povolení licencovat samotné HTTP. Definice známé jako Request for Comment (RFC) popisují co by váš software měl dělat, ale ne jak by jste to měli udělat. Jestliže zavedete RFC můžete využít výhod jakéhokoliv nástroje, který je pro vaši platformu nabízen. Tento přístup k návrhu software vedl k vysoce modularizovanému chování software, velice rychlému přijetí standartů z platformy na platformu a umožnil počítačům vzájemnou komunikaci na vrchní úrovni.

2.4. Všelidově stanovené standarty (RFC)

Dokumenty stanovující standarty pro Internetový software (RFC) jsou vytvářeny širokou skupinou lidí, nejen počítačovými odborníky. Jak již samotný název napovídá (žádost o připomínku) tyto standarty jsou otevřené a nejsou neměnné. Ve skutečnosti RFC ztrácejí platnost po dvou letech někdy dokonce dříve. RFC jsou dostupné pomocí el. pošty nebo FTP z několika serverů na Internetu.

2.4. Nedokonalý software

Překvapivě jeden z klíčových faktorů v úspěchu Internetu je, že většina software používaného na Síti není tak dokonalá. Posílání pošty (Sendmail) je ohromný zmatek, FTP je pomalý a HTTP je značně omezený a pomalý.

Většina tohoto software vyrostla z chytrých doplňků, které řešili menší problémy, které v softwarech čas od času vznikají. Výhodou toho všeho je, že software pracuje dostatečně, není kontrolován žádným dodavatelem, rychle se rozvíjí a je relativně jednoduše přenositelný na různé platformy.

Když se jedna část software stane populární najde se okamžitě někdo, kdo je ochoten vytvořit ještě lepší implementaci. Např. HTTP potřebuje vyšší rychlost a lepší datovou strukturu, kdyby však HTTP čekalo na všechna tato vylepšení ještě před svým zavedením, Web by pravděpodobně dosud neexistoval!

Používání 4D Databáze na Webu

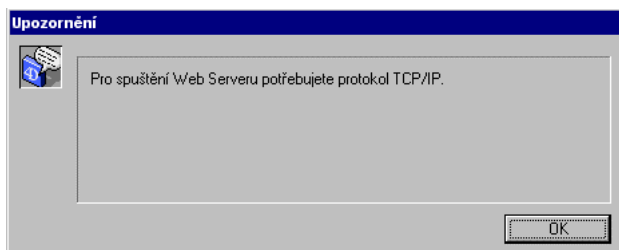
Základy Internetu

3. Základy Internetu

3.1. Instalace TCP/IP

V kursu budeme používat TCP/IP adresy a síť pro simulaci Internet/Intranet. Na počítačích Windows musí být instalovány síťové komponenty (network components) do složky aktivních Windows a musí zde být umístěny ve složce ACI/NETWORK. Na počítačích Macintosh musí být síťové komponenty instalovány přímo do 4D, 4D Server nebo 4D aplikace.

Jestliže se pokusíte nainstalovat služby 4D Web bez běžícího TCP/IP a nebo jestliže nemáte správně nainstalovaný ACI Network component, 4D zobrazí následující upozornění:



Jestliže se objeví tato zpráva proveďte instalaci síťových komponent, jak je popsáno níže nebo prověřte konfiguraci TCP/IP na vašem stroji.

3.1.1. Instalace TCP/IP na počítači Windows

1. Otevřete ve složce 4D Server složku Network components.
2. Poklepejte na Setup.
3. Postupujte dle instrukcí instalace.

3.1.2. Instalace TCP/IP na Macintosh

1. Poklepejte na TCP/IP installer
2. Klepněte na tlačítko Otevřít
3. Najděte příslušný 4D Engine, označte jej a klepněte na tlačítko Instal.
4. Okno zprávy zabliká a zobrazí: Mac TCP v...has been correctly installed.
5. Klepněte na tlačítko Quit.

3.2. Konfigurace TCP/IP

Pokud nemáte nakonfigurováno TCP/IP proveďte tuto konfiguraci, pokud máte stroj napojen přímo na Internet poradte se raději se svým administrátorem. Dále budeme diskutovat pouze nastavení na lokální síti.

Používání 4D Databáze na Webu

Základy Internetu

3.3. Názvy domén

Veškeré výhody Internetu můžete používat bez toho, aniž by jste příliš rozuměli službám názvů domén. Jediné co potřebujete znát je, že název domény odpovídá určité skupině IP adres strojů a že určitý název stroje slouží jako logický odkaz k tomuto stroji. Váš poskytovatel služeb Internetu nebo síťový administrátor vám pomohou vytvořit a udržovat jedinečný název domény. Je třeba vás dopředu upozornit na to, že váš poskytovatel internetu může čas od času z provozních důvodů měnit vaše IP adresy:

- ❖ přemístí váš stroj interně do jiné sítě, která používá odlišné skupiny adres IP
- ❖ změní svůj hlavní počítač Internet a musí změnit všechny IP čísla včetně samozřejmě vašeho stroje

Měli by jste vždy používat názvy domén, místo IP adres. V tomto kursu budeme používat IP adresy, ale doma to radši nezkoušejte.

3.4. Výhody služeb názvů domén

Pokud ještě nejste přesvědčeni o dobrých důvodech pro použití názvu domény, uvedeme si několik dalších výhod užití názvů strojů oproti použití čísel:

- ❖ Lidé si mnohem snadněji pamatují názvy než dlouhé seznamy čísel.
- ❖ Názvy je mnohem jednodušší uhodnout než nějaká čísla. Je běžnou praxí zkoušet hledat společnost na Internetu tím, že zadáte adresu způsobem www.názevspolečnosti.cz.
- ❖ Můžete přemístit server z jednoho stroje na jiný (z jedné adresy na druhou) aniž by jste nutili uživatele učit se nové názvy a čísla. V databázi názvů domén je jednoduché přiřadit názvu jinou IP adresu, takže uživatelé takovouto změnu ani nepostřehnou.
- ❖ Na tentýž stroj může ukazovat několik názvů, stačí je pouze použít v databázi domén. Např. společnost Tříhvězdičky, s. r. o. použije názvy Tříhvězdičky, 3Hvězdičky a pokud uživatel tyto názvy napíše dostane se na jeden a týž Web server Tříhvězdičky, s. r. o..
- ❖ S použitím volání Bind je možné, aby jeden název ukazoval na více strojů. Toto je způsob pro automatizaci distribuovaných procesů. Dobrým příkladem použití je rozdělení vyhledávání na několik oddělených strojů. Rozhodně je to lepší než nechat čekat uživatele ve frontě. Tímto způsobem může být použit řetězec více levných strojů (RAIC - Redundant Array of Inexpensive Computers) místo jednoho či několika drahých pracovních stanic.

3.5. Co znamenají čísla v IP adresách?

Do této doby jste již pravděpodobně viděli mnoho IP adres, co znamenají? Existuje krátká i dlouhá odpověď.

Krátká odpověď

Kdo by se o to staral? Nic vám to neřekne, můžou se změnit a stejně si je nemůžete vymyslet sami.

Používání 4D Databáze na Webu

Základy Internetu

Dlouhá odpověď

IP adresa je jedinečné 32-bitové číslo obvykle seřazené do čtyř skupin: 205.178.15.77. Toto číslo je užíváno k jednoznačné identifikaci zařízení na síti nebo mezi sítěmi jako Internet. Váš poskytovatel Internetu nebo administrátor sítě vám pomůže správně nastavit toto číslo na vašem počítači.

Pokud se chcete dozvědět více, viz např. **TCP/IP Network Administration** od O'Reilly & Associates, Inc.

3.6. Trvalé spojení

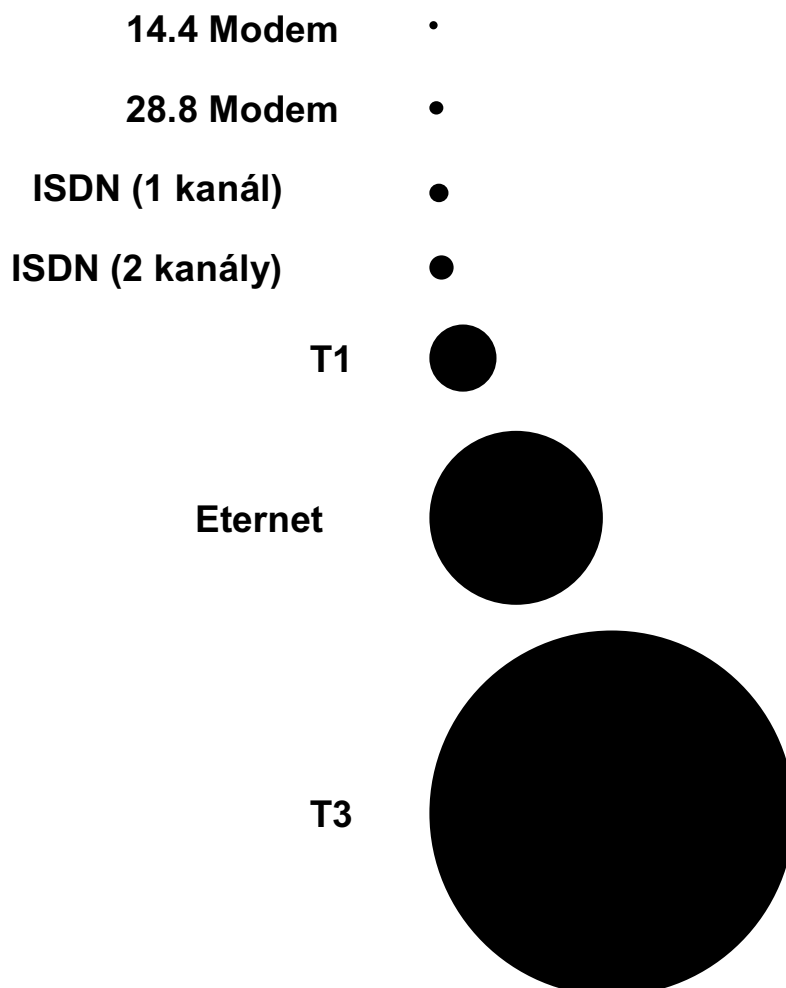
Jestliže chcete být ke svému serveru připojeni neustále, potřebujete pevnou linku nebo jiný druh trvalého připojení. Váš výběr bude pravděpodobně záviset na technologii dostupné od vašeho poskytovatele a ceně, kterou jste ochotni zaplatit. Nejpravděpodobnější výběry jsou ISDN, FrameRelay T1, fractional T1. T1 linka vyžaduje speciální telefonní linku, router na vaší straně a straně poskytovatele. Dá vám to možnost připojení do asi 1 536 mps. Fractional T1 vyžaduje tentýž hardware a nastavení linky, ale může stát méně v závislosti na tom jakou její kapacitu si zakoupíte. Nové možnosti se samozřejmě objevují čím dál rychleji a nebudeme je zde diskutovat např. bezdrátové spojení atd. Jestliže se chystáte najmout si trvalé propojení na internet udělejte následující:

- ❖ Porovnejte nabídky několika dodavatelů.
- ❖ Vyberte tu která vám dovolí zvyšovat kapacitu, až ji budete potřebovat
- ❖ Ujistěte se, že váš kontrakt obsahuje datum instalace a penále za pozdní instalaci.
- ❖ Promluvte si se současnými zákazníky vybraného poskytovatele

Jiná mnohem jednodušší možnost pro zpřístupnění svých stránek je umístit vlastní server v sídle vašeho poskytovatele. V některých případech to může být nejlepší řešení. Přinesete plně konfigurovaný server do sídla vašeho poskytovatele a on jej pro vás pouze připojí na internet. Získáte výhodu vysoce rychlostní linky za poměrně menší cenu, než je udržování spojení do vašeho sídla. Můžete být schopni se připojit až na linky T3, které jsou několikanásobně rychlejší než T1 – 45 mps. Další možnosti jsou propojit se standartním ethernetem (10 mps) nebo rychlým ethernetem (100 mps). Následující diagram ukazuje relativní kapacity několika možností:

Používání 4D Databáze na Webu

Základy Internetu



3.7. "V6" budoucnost IP adres

Tak jako telefonní společnosti mohou dojít telefonní čísla a potřebuje modernější ústřednu, tak i Internetu docházejí jeho IP adresy. Řešení se nazývá V6 (nemá nic společného se 4Dv6). Znamená to pouze IP adresy o šesti pozicích. Naplánované zavedení je na přelomu tisíciletí. Toto řešení umožní celkem $4 \cdot 29$ triliónů adres. Pro vaši představu to znamená několik set adres na čtvereční palec naší planety (možná to bude stačit),

Používání 4D Databáze na Webu

Databáze na Web za 5-15 minut

4. Databáze na Web za 5-15 minut

V této kapitole se budeme zabývat vytvořením databáze na Web za 5 minut. Toto demo se většinou používá na prezentacích. Zde jej budeme probírat ze tří následujících důvodů:

- 1) Chceme demonstrovat jednoduchost publikování databáze 4D pro Web.
- 2) Chceme ukázat základy vytváření databáze pro Web.
- 3) Chceme uvést ty rysy, které budeme na reálných databázích probírat později.

4.1. Zadání TCP/IP adres

Protože je toto pouze kurs pro Web nebudeme databázi skutečně publikovat, ale to nevadí Microsoft i Netscape uvádějí, že prodali stokrát více Web serverů pro užití v lokálních podnikových sítích než pro užití v Internetu. Nástroje Web totiž pomáhají pracovníkům v oblasti informačních technologií zavádět řešení za rozumné ceny. Web obvykle odpovídá Web serveru na Internet, ale není to rozhodně jediné a nejčastější místo, kde jsou nástroje a aplikace pro Web používány.

Vytvoříme si svůj vlastní intranet přímo tady. Ujistíme se nejdříve, že máme jedinečnou IP adresu v naší síti. Protože nepoužíváme služeb serveru názvu domén, budeme používat k přístupu ke 4D databázi přímo IP adresy.

4.1.1. Zadání TCP/IP adresy na Windows

1. Poklepejte na ikonu Tento počítač.
2. Poklepejte na ikonu Ovládací panely.
3. Poklepejte na ikonu Síť.
4. Nalistujte a označte si TCP/IP protokol.
5. Klepněte na tlačítko Vlastnosti.
6. Ověřte si platnost IP adresy, jestliže není platná nebo zadána, napište IP adresu např. 123.45.6.71
7. Prověřte, že maska podstítě je 255.255.255.0
8. Zapamatujte si svoji IP adresu.
9. Uzavřete otevřená okna.

4.1.2. Zadání TCP/IP adresy na počítači Macintosh

1. Otevřete ovládací panel Mac TCP.
2. Ujistěte se, že je vybrána ikona Ethernet.

Používání 4D Databáze na Webu

Databáze na Web za 5-15 minut

3. Prověřte platnost IP adresy. Jestliže není platná zadejte jedinečnou adresu např. 123.45.6.71 a zapamatujte si ji.
4. Klepněte na tlačítko Více.
5. Prověřte, že maska podsítě je 255.255.255.0
6. Klepněte na tlačítko Budiž
7. Uzavřete otevřená okna.

4.2. Vytvoření jednoduché databáze

Vytvoříme velice jednoduchou databázi. Budeme sledovat různé druhy kávy: jejich název, stupeň hodnocení a místo původu. Kdykoliv vytváříme novou databázi s v6 automaticky se nastartují i služby Webu.

4.2.1. Vytvoření struktury

1. Otevřete 4D a vytvořte novou databázi nazvanou Coffee. Datový soubor se bude jmenovat Coffee.4DD
2. Klepněte na Tabulku1 a změňte název na Coffee a to následujícím způsobem vyberte Soubor→Otevřít strukturu→Vlastnosti tabulky...pak stiskněte tlačítko Provést a potom tlačítko Hotovo.
3. Zvolte Struktura→Nové pole... a přidejte následující pole. Po každém zadání nového pole stiskněte tlačítko Provést:

Název	Alpha (40)
Hodnocení	Alpha (20)
Místo	Alpha (40)

4. Stiskněte tlačítko Hotovo

4.3. Formuláře

Místo, abychom nechali 4D vytvořit formuláře automaticky. Vytvoříme své vlastní, které budou na Webu vypadat lépe. Tato část je pro navrhování 4D databází pro Web velice důležitá. HTML jsou mnohem více omezeny než samotná 4D, takže použité formuláře musí být navrženy poněkud jinak než standartní formuláře.

4.3.1. Vytvoření formulářů pro Web

1. Zvolte Návrh→Nový formulář...
2. Zadejte jméno formuláře Vstup
3. Vyberte typ formuláře S obsahem vstupní z nabídky Typ formuláře.
4. Vyberte Web Aware z nabídky Užítý vzor.
5. Vyberte všechna tři pole poklepáním na každé z nich. Vyberte je v pořadí Název, Stupeň, Místo.
6. Klepněte na tlačítko Upravit.

Používání 4D Databáze na Webu

Databáze na Web za 5-15 minut

7. Ujistěte se, že tlačítka jsou zcela nad úrovní Čáry obsahu (O).
8. Uzávěřte formulář.
9. Klepněte na tlačítko Nový v Průzkumníku
10. Pojmenujte formulář Výstup.
11. Z nabídky Typ formuláře vyberte Seznam výstupní.
12. Z nabídky Užítý vzor vyberte Web Aware.
13. Poklepáním na první dvě pole je vyberte.
14. Klepněte Upravit a zkontrolujte formulář.
15. V okně Průzkumníka, klepněte na formulář Vstup a zatrhněte políčko Vstupní form. v pravé dolní části okna.
16. Klepněte na formulář Výstupní a zatrhněte políčko Seznam v pravé dolní části okna.
17. Zavřete okno Průzkumníka.

4.4. Zprovoznění aplikace

Vytvoříme nabídku a přiřadíme k ní několik metod, tak jak vytváříme nabídky pro aplikaci. Skutečně pokud budeme potřebovat budou nabídky použitelné i na úrovni aplikace.

4.4.1. Vytvoření metod

1. Zvolte Návrh→Nová metoda...
2. Napište název metody: M_AddRecords
3. Přidejte následující kód:

```
` Method: M_AddRecords  
C_TEXT ($1)  
  
Repeat  
  ADD RECORD ([Coffee];*)  
Until (OK = 0)
```

4. Zvolte Návrh→Nová metoda...
5. Napište název metody: M_ShowRecords
6. Přidejte následující kód:

```
` Method: M_ShowRecords  
C_TEXT ($1)  
  
ALL RECORDS ([Coffee])  
ORDER BY ([Coffee]Název)  
MODIFY SELECTION ([Coffee]; *)
```

7. Zavřete všechna okna.

4.4.2. Vytvoření záhlaví nabídky a několika položek nabídky.

1. Vyberte Nástroje→ Editor nabídek.
2. Stiskněte tlačítko Přidat v levém dolním rohu a vytvořte tak Nabídku #1
3. Rožšířte nabídku Soubor klepnutím na trojúhelník.

Používání 4D Databáze na Webu

Databáze na Web za 5-15 minut

4. Stiskněte tlačítko Přidat položku.
5. Pojmenujte novou položku nabídky: Přidat záznam.
6. Stiskněte Ctrl+mezerník (CMD+mezerník), dostanete tak na popředí okno Průzkumníka.
7. Najděte metodu M_AddRecords.
8. Potáhněte metodu M_AddRecords a přiřaďte ji tak k položce Přidat záznam.
9. Stiskněte opět tlačítko Přidat položku.
10. Přidejte položku nabídky: Zobrazit záznamy a k této položce přiřaďte metodu M_ShowRecords
11. Stiskněte tabelátor a zavřete Editor nabídek.

4.5. Přístup k databázi z webovského prohlížeče

Všichni jsme již pracovali se 4D databází z jejího vlastního prostředí nebo pomocí 4D Client. Nyní uvidíme jak se taková databáze chová, když se k ní připojíme pomocí webovského prohlížeče. Pro tento účel můžeme použít Netscape Navigator nebo Microsoft Explorer, existuje samozřejmě více webovských prohlížečů vhodných pro tento účel. Ve spojení s 4Dv6 se stává prohlížeč Web klientem databáze na úrovni klient/server (4D Client). Toto je obrovská výhoda pro uživatele, který může nyní užívat 4D z kterékoliv platformy včetně UNIX a OS/2 a ne pouze Mac OS a Windows.

K takovému Web serveru se můžete připojit odkudkoliv z vaší sítě včetně vašeho vlastního stroje, na kterém je Web server spuštěn. Na konci kursu se sešítujeme dohromady což nám umožní připojit se vzájemně na naše databáze a nejenom ke své vlastní.

4.5.1. Připojení k databázi a úvodní stránka

1. Spustíte prohlížeč Web.
2. Když se otevře okno prohlížeče napište IP adresu vašeho počítače a stiskněte klávesu enter. Musíte napsat `http://123.4.56.71`
3. Název okna prohlížeče se změní na Nabídka 1, v okně uvidíte nabídku Soubor své aplikace tak jak je interpretována prohlížečem.

Prohlížeč nekomunikuje přímo se 4D, dokonce ani o 4D neví. Vše co ví je, že komunikuje s Web serverem. Co se týká prohlížeče, Web server klidně může být spuštěn na Mac či UNIX pracovní stanici nebo sálovém počítači IBM 3270. Prohlížeč neví nic o 4D, ale ví jak interpretovat a zobrazit HyperText Markup Language (HTML). 4D v6 ví jak automaticky převádět 4D objekty do HTML formátu. Vaše záhlaví nabídky #1 bylo převedeno do HTML při nastavení spojení a prohlížeč použil HTML, kterou obdržel k vytvoření obrazovky, kterou nyní vidíte před sebou. HTML budeme probírat podrobněji později, i když při použití 4Dv6 nepotřebujete, pokud nechcete, napsat ani řádku HTML kódu. 4D převádí 4D objekty do HTML a to včetně formulářů a nabídek za běhu aplikace v době, kdy jsou prohlížečem požadována tzn., že není potřeba měnit nějaký HTML kód jestliže změníte záhlaví nabídky #1 nebo jestliže změníte některý ze svých formulářů—4D provede za vás tyto úlohy automaticky.

4.5.2. HTML verze nabídky 1

Nyní se podívejme podrobněji co se děje v prohlížeči. Povšimněme si, že 4D odešle název záhlaví nabídky do názvu okna. Rovněž odešle název nabídky Soubor nad jednotlivé položky. Pokud by existovaly další

Používání 4D Databáze na Webu

Databáze na Web za 5-15 minut

nabídky, objevili by se obdobným způsobem pod první nabídkou. První nabídka obsahuje tři položky pod hlavičkou Soubor a ty jsou zobrazeny svými názvy a ne přiřazenými metodami.

4.5.2. Prohlížení a uložení HTML kódu

Nyní se podívejme na HTML, které 4D vytvoří (použijeme Netscape Navigator):

1. Zvolte View → Document Source z nabídky Navigatoru.
2. HTML kód se zobrazí v novém okně.

HTML kód bude vypadat asi následovně:

```
<HTML>

<HEAD>

<TITLE>Záhlaví 1</TITLE>

</HEAD>

<BODY>

<H2>Soubor</H2>

<MENU>

<LI> <A Href="/4Dmenu/1-1-1/$140789744.0">Přidat záznam</A>

<LI> <A Href="/4Dmenu/1-1-2/$140789744.0">Zobrazit záznamy</A>

<LI> <A Href="/4Dmenu/1-1-3/$140789744.0">Konec</A>

</MENU>

</BODY>

</HTML>
```

V prohlížeči najdete odkaz, na který lze klepnout pro každou položku s přiřazenou metodou, např. Přidat záznam. Uvnitř kódu HTML vidíte, že pokud klepnete na položku odkaz použije následující adresu: /4Dmenu/1-1-1/\$140789744.0

Každá část této adresy znamená něco pro 4D:

4Dmenu	Je to objekt nabídky
1-1-1	Záhlaví #1, nabídka #1, položka #1
\$1407890789744	ID obsahu. Toto číslo je použito ve jméně procesu ve 4D, je přiřazeno tomuto spojení WEB.
.0	ID podobsahu. 4D jej užívá pro sledování, kde se uživatel nachází v této sekci Web.

Používání 4D Databáze na Webu

Databáze na Web za 5-15 minut

Povšimněte si, že název metody se neobjeví nikde v HTML kódu, objeví se pouze číselná reference k této položce. Samotné spuštění metody zajistí 4D na základě reference. ID kontextu je generováno 4D a vy je nikdy přímo nepoužijete. Je použito interně samotnou 4D k udržování komunikace se zcela určitým oknem prohlížeče pro tuto určitou sekci. Jestliže otevřete nové okno v Netscape a připojíte se k databázi podruhé, výsledný HTML kód pro záhlaví 1 bude obsahovat jiné ID kontextu.

4.6. Zacházení s daty přes prohlížeč

Až do teď jsme vytvořili pouze dvě velice jednoduché metody, přesto budeme již schopni zacházet s daty databáze Coffee. Díky síle příkazu MODIFY SELECTION, jsme odvedli dobrou práci. Můžeme přidávat záznamy, prohlížet si záznamy a dokonce upravovat existující záznamy.

4.6.1. Přidání několika nových záznamů

1. Klepněte na odkaz Přidat záznam a vyzkoušejte váš vstupní formulář. Jestliže se obrázky reprezentující tlačítka neobjeví, klepněte na prázdné grafické objekty k přenosu obrázků, nebo klepněte na tlačítko "images" v liště nástrojů Netscape.
2. Napište následující:
Kona
Premium
Hawaii
3. Stiskněte tlačítko Přijmout a přidejte další záznamy:
Jamaican Blue Mountain
Supreme
Jamaica

Folgers
Méně je někdy více
USA
4. Stiskněte tlačítko Storno k ukončení smyčky přidat. Navrátíte se do záhlaví 1.

4.6.2.. Prohlížení a úpravy záznamů

1. Klepněte na odkaz Zobrazit záznamy, k vyzkoušení výstupního seznamu.
2. Klepněte na ikonu šipky vlevo od záznamu Folgers, k zobrazení záznamu ve vstupním formuláři.
3. Jestliže jste si rozmysleli hodnocení Folgers, změňte jej.
4. Klepněte na tlačítko Přijmout k uložení vašich změn.
5. Klepněte opět na ikonu šipky vlevo od záznamu Folgers, a uložte změny.
6. Klepněte na tlačítko Doprava k zobrazení dalšího záznamu.
7. Klepněte na tlačítko Přijmout k návratu do seznamu záznamů.
9. Klepněte na tlačítko Hotovo k uložení první stránky.

Přesto, že jsme vytvořili velice jednoduchý příklad, byly jsme již schopni ukázat možnosti přidávání, úprav a vymazání záznamů pomocí prohlížeče Web a to vše v několika minutách. Naše další databáze bude již propracovanější, předtím než se ponoříme hlouběji uvedme stručný přehled toho, co jsme schopni se 4Dv6 udělat.

Používání 4D Databáze na Webu

4D a Web

5. 4D a Web

Díky své univerzálnosti je 4D schopna zastat několik úloh v prostředí sítí, které před několika lety rozhodně nikdo nemohl předpovědět. 4D Server komunikuje s TCP/IP – internetovým esperantem-přímo. 4D na Web je ideální platforma pro vytváření, údržbu, spojení a interakci mezi uživatelem a daty. To co lze ve 4D provést spadá do několika kategorií uvedených níže a později podrobněji rozebraných.

- ❖ Transakčně orientované aplikace
- ❖ Integrace do větší sítě
- ❖ Interaktivita
- ❖ Tradiční řízení Web
- ❖ Front End pro SQL Data

5.1. Transakčně orientované aplikace

Užití Web serveru k odesílání statických stránek je jednoduché. Použití databázi pro generování HTML z obsahu záznamů je rovněž relativně jednoduché. Přinutit prohlížeč Web, aby fungoval jako klient databáze není těžké, těžké je to pouze, když nepoužíváte 4Dv6. Web nebyl navržen pro transakčně orientované aplikace a doposud to byl složitý a časově náročný problém. V6 nabízí jedinečnou integraci funkcí databáze v rozhraní prohlížečů Web. Je dokonce přesnější říci že v6 tak jak používá HTTP je síťovým komponentem než Web serverem. Žádný další produkt na jakékoli platformě nedosáhl takové synergie jakou lze nalézt ve 4Dv6. S 4Dv6 začíná být již výhodné použít Web pro transakčně orientované systémy jako nákupní centra, vkládání dat, uživatelské hledání atd. V jiných prostředích tyto úlohy zaberou neúměrné množství programátorského času, se 4D v6 tomu tak není. Revoluční rysy 4Dv6 jsou tím co se budete učit ve zbytku tohoto kurzu. Ale nebylo by správné nezmínit další způsoby použití 4D s Internetem nebo na vašem intranetu.

5.2. Integrace větší sítě

Sítě spojují *počítače* dohromady. Z důvodů zapojení lidského prvku do Web, e-mail a skupin je poměrně jednoduché zapomenout, že síť spojuje počítače a ne lidi. Tato poznámka byla učiněna jen proto, aby zdůraznila, že v prostředí počítačových sítí by jste měli uvažovat o tom, jak spolu interagují programy a data.

- ❖ Použijete-li Netlink/4D vaše databáze může interagovat s libovolným tradičním Web serverem umístěným na Windows nebo Mac ve vaší lokalitě. Je to výhoda pokud již máte zavedený Web server a nebo chcete dosáhnout bezpečnou úroveň a užít Secure Socket Layer (SSL) Web server pro finanční či jiné transakce.
- ❖ S PDM Internet Tools nástroji pro internet, můžete použít vaši 4D databázi pro čtení a psaní zpráv e-mail. Můžete například automatizovat mailové odpovědi, posílat noční zprávy, ovládat pořadí odesílání a odpovídat takto na nejběžnější otázky zákazníka (odpovědi jsou v databázi).
- ❖ S Internet ToolKit (ITK) můžete vytvořit virtuálně jakýkoliv na TCP/IP založený systém. Můžete vytvořit FTP, mail nebo Web klienta či server uvnitř vaší 4D.

Používání 4D Databáze na Webu

4D a Web

Z důvodů silného jazyka a rozšiřitelné architektury, je poměrně těžké představit si nějakou úlohu pro Internet, kterou by 4D nezvládla.

5.3. Interaktivita

Koncept statické stránky ukazuje sám o sobě na dobu svého vzniku a své stáří. Samotný Web a nástroje jako 4D dospívali v době, kdy lidé již očekávali interaktivitu a komunikativnost, se speciálními rysy jako jsou:

- ❖ Schopnost vyhledat data a klást otázky a provádět dotazy.
- ❖ Personalizaci samotných stránek
- ❖ Schopnost odesílat data k zpracování
- ❖ Podpůrné výpočty a prostředí pro uskutečnění konferencí

4D V6 nabízí kompletní ošetření vašich úloh pro vytvoření interaktivních webovských stránek. Pomocí 4D a Netlink/4D můžete také přidat zákaznické funkce tradičnímu Web serveru.

5.4. Tradiční řízení Web

4DV6 vás nezatěžuje samotným nudným kódováním HTML, jestliže si však přejete psát HTML přímo, 4D je skvělým nástrojem! Použití 4D a jazyka 4D k vytvoření vašich Web stránek je celkem snadno zvládnutelný úkol s poměrně velkými výhodami. Je to ideální řešení pro zákazníky, kteří chtějí být neobvyklou součástí internetu, ale nejsou ochotni zaplatit za vysoce kapacitní internetové linky. 4D můžete použít pro vytvoření kompletního umístění pro upload vašich zákaznických stránek na pronajatá místa na hostitelích na ISP.

HTML tagy zapadají přirozeně přímo do struktury záznamů 4D a systém odkazů internetu zapadá do samotného modelu relační databáze 4D. Web umístění libovolné šíře požaduje nástroj pro řízení strukturovaných dat k tomu, aby zajistil, že odkazy jsou stále platné, indexy k umístění mohou být vytvořeny a že data nejsou ztracena nebo neplatná. Bez použití 4D, vyžaduje údržba umístění libovolné složitosti buď správně naprogramovaný software nebo skupinu pracovníků zabývajících se úpravami HTML.

Se zákaznickou databází 4D, mohou být texty a odkazy abstrahovány, takže konečný uživatel se může soustředit pouze na přispívání do obsahu v databázi aniž by znal cokoliv o HTML. Po poměrně dlouhý čas samotný vnitřní Web systém ACI byl produkován z databáze 4D v3, kde přispěvatelé nemuseli znát nic o HTML. Toto je ideální řešení vašich starostí s udržováním Web. Důvody jsou následující:

- ❖ V době, kdy jste připraveni využít výhod automatického systému Web 4Dv6, jsou vaše data k dispozici.
- ❖ Náklady spojené s údržbou Web umístění jsou výrazně sníženy.
- ❖ Údržba obsahu stránek může být delegována zpět do pracovních skupin tím, že přispívají do původního obsahu.
- ❖ Přispěvatelé se nemusí učit nic nového (HTML) a pouze potřebují aktualizovat obsah 4D databáze.
- ❖ Styly v HTML mohou být použity dle potřeby.

Používání 4D Databáze na Webu

4D a Web

- ❖ Vzhled systému může být radikálně změněn několika jednoduchými kroky.
- ❖ Všechny vnitřní odkazy jsou řízeny procedurami, které vylučují chybovost odkazů.
- ❖ Stránka, text a grafický styl mohou být změněny v jednom místě (proceduře) a změny se projeví automaticky v celém systému.
- ❖ Existující data mohou být publikována bez nového pořizování.
- ❖ Tatáž data mohou být užita pro několik účelů současně, včetně přímého přístupu pomocí 4D nebo 4D Client a mohou publikována pomocí dalších nástrojů (knižně nebo v PDF souborech...)

Následující příklad formuláře je skutečný snímek z obrazovky z řídicí databáze ACI umístění:

User Groups (ACI_US : English)

User Groups

Name	City	State
BMUG DB SIG	Berkeley	CA
Meeting date	Meeting locale	URL
3rd Saturday , 11 am	BMUG offices, 2055 Center St.	
Contact Name	Contact phone	Contact e-mail
Glenn A. Bookout	(510) 549-2684 x225	gbookout@eworld.com

Services

A monthly meeting to share information, ideas, and questions related to databases in general with emphasis on Macs and ACI US' 4th Dimension, and its related products

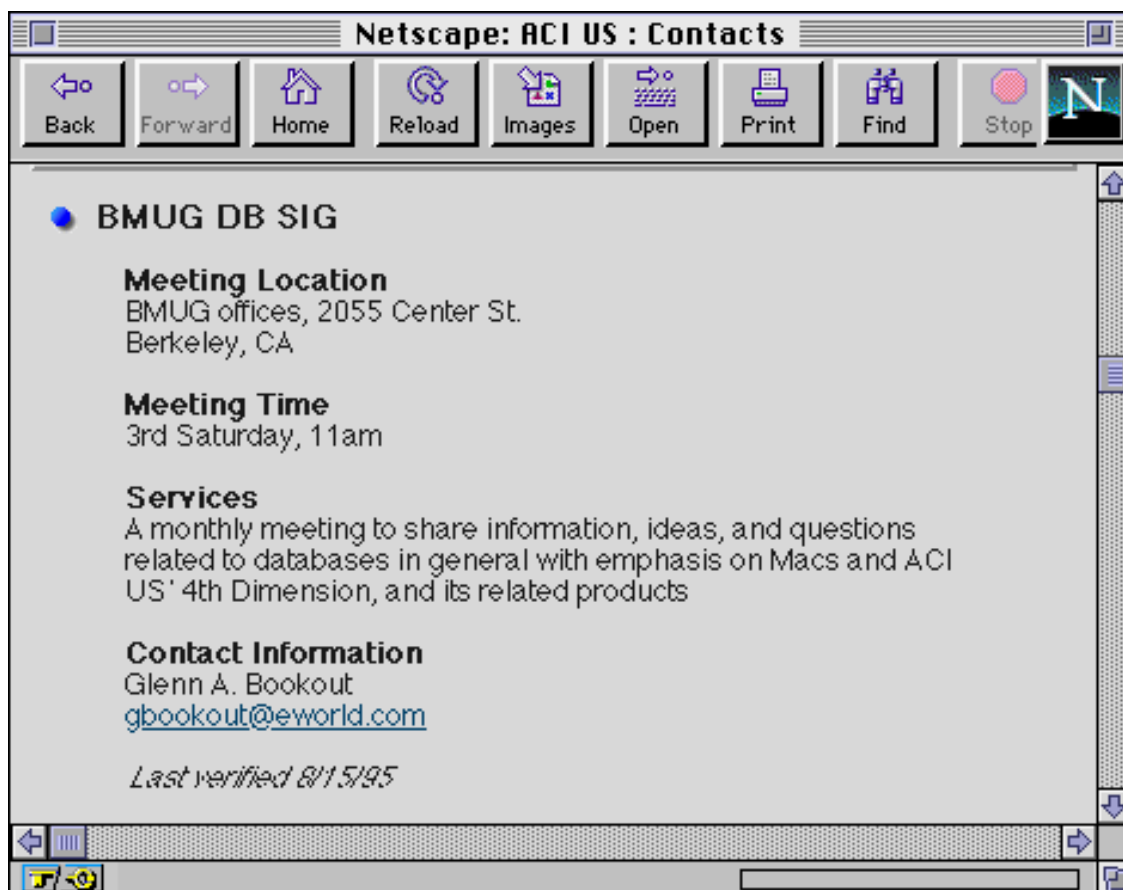
Last Verified

08/15/95

Následující obrazovka z Netscape ukazuje tatáž data na Web po připojení webovským prohlížečem. Data jsou konvertovaná do HTML vnitřní procedurou 4D:

Používání 4D Databáze na Webu

4D a Web



5.5. Front end pro SQL databáze

Potřeba pružných a vícenásobně použitelných nástrojů pro údržbu SQL databází vzrůstá tak, jak se integrace databází do Web stává důležitějším. Samotný SQL neobsahuje žádný koncept pro řetězec nebo stránky což jsou nejzákladnější prvky Webu. Aby mohl být konvertován existující obsah SQL dat do formátu pochopitelných pro Web jsou potřeba samostatné aplikace a překladače dat. Dodavatelé SQL poskytují podporu pro tyto nové požadavky pomocí externích aplikací a nových způsobů volání uložených procedur. Protože tyto úlohy jsou cizí samotné podstatě SQL jsou tato řešení někdy nemotorná a značně pomalá v porovnání s typickými operacemi SQL.

Bohatý jazyk čtvrté generace (4DGL) samotné 4D obsahuje všechny příkazy a datové struktury potřebné pro úplné řízení formátování a uspořádání vašich dat. Konverze SQL dat do HTML je ve 4D jednoduchá, pružná a účinná. Vedle kvality a rysů jazyka, můžete 4D lokálně použít ještě k zkopírování nejčastěji požadovaných SQL dat a dosáhnout lepšího chování než se samotným SQL backend. Ve 4D prováděná podpora jakékoliv ODBC databáze činí z 4D neobvykle konkurenceschopné vývojové prostředí na trhu.

Používání 4D Databáze na Webu

Přehled rysů

6. Přehled rysů

6.1. Úvod

Vytvořili jsme pětiminutovou databázi Web, takže nyní již víte, že se 4Dv6 můžete vaši databázi publikovat na Web téměř okamžitě a použít prohlížeč Web jako klienta databáze. Provedme shrnutí toho co jsme se v příkladové databázi dozvěděli i když to někdy nebylo přímo viditelné.

6.2. Web připojení

4D automaticky řídí Web připojení ve svém vlastním procesu. Udrží stav a obsah tzn., že 4D sleduje co uživatel právě dělá a automatizuje všechny aspekty sekce klient/server. Na podrobnosti tohoto tématu se podíváme později během kursu.

6.3. Automatický převod formulářů

Převod jste viděli, je určitě překvapující a funkční. 4D převádí formuláře do HTML v průběhu připojení a to velice rychle. To není vše. Většina automatických akcí a některé formuláře a události objektů formulářů jsou podporovány tak, že obdržíte vysoce funkční systém Web bez dodatečného programování.

6.4. Zpětné volání metod 4D

Samotné stránky komunikují s 4D tak, že je možné použít přímé odkazy na metody programované ve 4D. Tzn., že můžete váš již existující kód plně využít bez přelišného doprogramování.

6.5. Provázání proměnných

Můžete vytvořit své HTML stránky, které automaticky převádějí své hodnoty do proměnných 4D při jejich odeslání. Jak ? Přidělte poli v HTML ten samý název jako cílové proměnné ve 4D. Opět žádné zvláštní programování není potřeba.

6.6. Zamykání záznamů

4D a 4D Server řídí veškeré zamykání záznamů za vás v případě, že Web client upravuje záznam. V tomto případě je Web client úplnou ekvivalencí 4D Client. Pro využití tohoto rysu není potřeba žádné programování.

6.7. Automatické stránkování

Není praktické posílat stovky či tisíce záznamů přes Web současně. Jestliže váš Web client provádí dotaz a databáze nalezne 300 záznamů. 4D provede automatické odstránkování této sekce. Když zobrazujete záznamy pomocí DISPLAY SELECTION nebo MODIFY SELECTION, 4D odešle první část výběru a automaticky přidá odkazy na stránky pokud uživatel bude požadovat jiné části výběru. V normálních databázích Web tento rys vyžaduje podrobný rozbor a poměrně dlouhé programování. Ve 4D v6 nepřidáte ani jednu řádku kódu.

6.8. Existující HTML

Jestliže již máte své webovské stránky rozhodně je nemusíte vyhodit, ale můžete je všechny umístit do 4D. 4Dv6 obsahuje jednoduché příkazy k odeslání HTML dokumentů umístěných na pevném disku, velice podobných funkcím regulérního Web serveru.

Používání 4D Databáze na Webu

Přehled rysů

6.9. Odeslání HTML nebo JavaScript přímo

Jestliže chcete větší kontrolu vašich stránek můžete je zahrnout přímo do formuláře 4D. Do formulářů 4D můžete zahrnout vlastní HTML včetně JavaScript, do standartních formulářů 4D můžete zahrnout pouze JavaScript. V tomto kursu se naučíme jak používat JavaScript tak, aby interagoval s webovskými formuláři vytvářenými 4D.

Používání 4D Databáze na Webu

Konfigurace

7. Konfigurace

7.1. Úvod

4D V6 je pravděpodobně nejjednodušejí konfigurovatelný Web server na světě. Musíte udělat následující:

1. Otevřete databázi

To je všechno! Teda přesněji řečeno, jak víte musíte udělat ještě několik dalších kroků ale ne moc. Zde je přesnější popis kroků:

1. Ujistěte se, že máte instalovaný TCP/IP Network Component viz výše.
2. Otevřete databázi.
3. Ujistěte se, že databáze je připravena pro publikování na Web.
4. Zapněte Web server pokud již neběží.

A to je skutečně všechno! Pojďme se blíže podívat na některé kroky.

7.2. Nastavení předvoleb startu Web Serveru

7.2.1. Otevřete Předvolby databáze

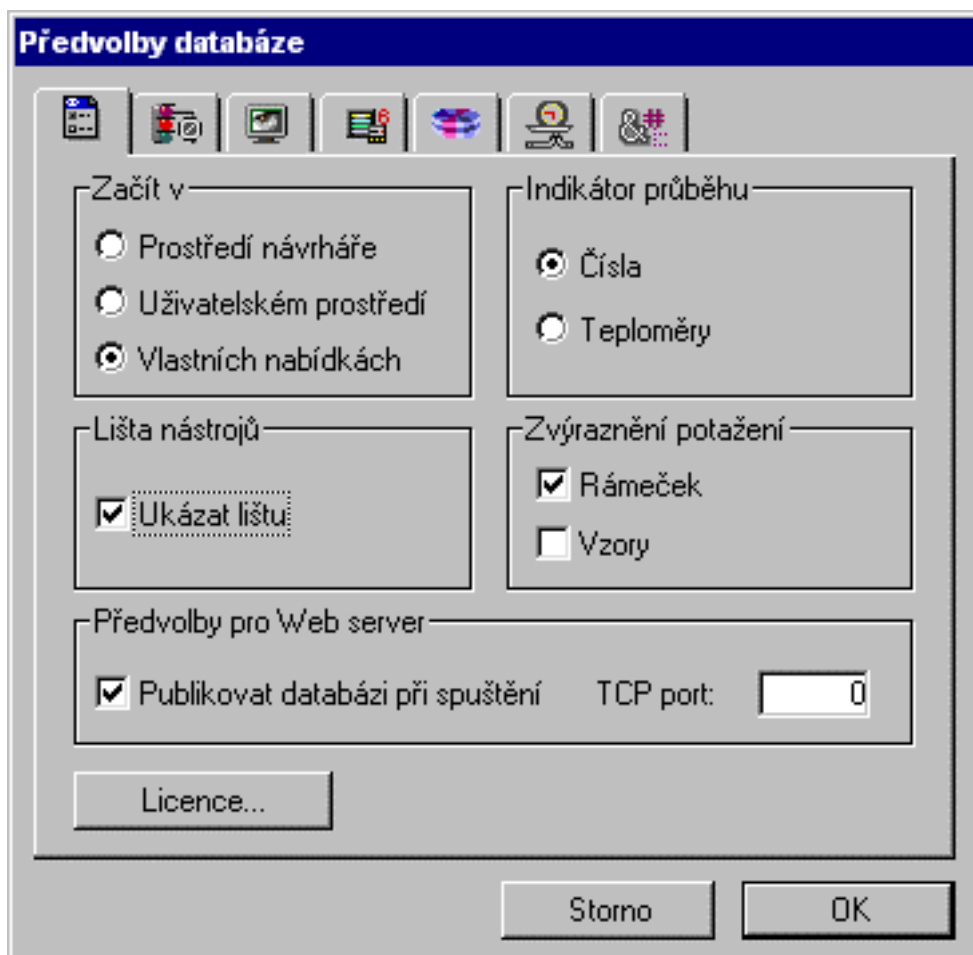
Podívejte na stránku nastavení Web.

1. Otevřete databázi ACI Video.
2. Přejděte do prostředí Návrháře.
3. Zvolte z nabídky Soubor – Předvolby databáze...

Používání 4D Databáze na Webu

Konfigurace

Uvidíte tuto obrazovku:



“Předvolby pro Web Server” vám dovolí určit dvě vlastnosti: zda se Web server **nastartuje** automaticky při spuštění a dále jaký **port** bude použit pro publikování Web služeb.

7.2.2. Publikování Web serveru při startu

Jestliže chcete, aby vaše databáze byla viditelná prohlížeči Web od startu aplikace, zaškrtněte “Publikovat databázi při spuštění”. Jinak můžete Web server startovat a ukončovat z Prostředí uživatele nebo příkazem z procedury a vlastní nabídky, to ale probereme později.

7.2.3. TCP Port

7.2.3.1. Co je TCP Port?

O číslech portů je dobré přemýšlet jako o součásti adresy aplikace a Web serveru. Čísla portů jsou velice využívána v implementacích UNIX, ale jsou součástí TCP/IP všech platform. Různé aplikace běží na různých portech, aby je bylo možno rozeznat. Běžné TCP/IP aplikace mají své stanovené porty. Dobře známá čísla portů umožňují aplikacím nezasahovat si navzájem a umožní klientovi tuto službu nalézt. Zde je seznam dobře známých portů pro běžné služby:

Používání 4D Databáze na Webu

Konfigurace

Služba	Popis	Port
FTP	File Transfer protocol	21
Telnet	Interactive terminal session	23
SMTP	Simple Mail Transfer Protocol	25
whois	Lookup DNS record	43
gopher	The Gopher protocol	70
finger	finger	79
HTTP	HyperText Transfer Protocol	80
NNTP	USENET news using NNTP access	119
NTP	Network Time Protocol	123

Poznámka: Technicky je číslo portu přiřazeno kombinaci služeb a přenosových protokolů, takže tentýž port může být použit pro tutéž službu na různých protokolech. UDS je přenosový protokol, který je částí TCP/IP takže může být využit místo TCP. 4D nepoužívá UDP protokol, takže je to naposled co jsme jej zmínili.

Neexistuje nic co by vás nutilo používat tato čísla pro odpovídající služby, ale jsou to čísla, kde klientův software tyto služby hledá s výchozím nastavením. Například číslo portu pro HTTP (Web Server) je 80. Je automaticky volen stanovíte-li adresu v prohlížeči standardním způsobem:

`http://www.4D-Rules.com/`

prohlížeč ve skutečnosti použije URL:

`http://www.4D-Rules.com:80/`

Váš prohlížeč Vám toto URL nezobrazí, ale rovnou jej interně použije. Můžete si všimnout, že ve 4D je standardní nastavení 0, to znamená výchozí port 80. Jinými slovy je-li napsáno 0 nebo 80 je to totéž.

7.2.3.2. Použití nestandardních čísel portů

Není důvod pro použití nestandardních čísel portů ve vaší databázi a proč by jste se museli nějakým číslem vyhnout. Jsou však důvody proč můžete použít nestandardní čísla portů:

- ❖ Chcete spustit více než jeden WebServer na témže stroji
- ❖ Chcete použít nestandardní port z důvodů bezpečnosti. Pouze uživatelé, kteří jej znají naleznou váš Web Server.
- ❖ Konfigurujete například ne-4D Secure Socket Layer (SSL) nebo SHTTP (Secure HTTP) Web server pro bezpečné transakce. Tyto servery používají odlišné porty. WebSTAR SSL užívá například port 443.

Používání 4D Databáze na Webu

Konfigurace

Pokud uživatel zná číslo portu je možno napsat URL téměř jako normálně. Jestliže provozujete databázi na portu 4444, uživatel napíše:

`http://www.4D-Rules.com:4444/`

Pokud hledáte nestandardní port pak se ujistěte, že:

1. Tento port je na vašem stroji jedinečný.
2. Vyhněte se číslům menším než 1024. Čísla pod 256 jsou rezervována pro známé služby a čísla 256-1024 jsou rezervována pro služby UNIX, spoustu z nich je používáno i na jiných platformách.
3. Řekněte vašim uživatelům, jak se na váš server připojit!

*Poznámka: Všechny další podrobnosti lze nalézt v knize **TCP/IP Network Administration**.*

7.3. Licenzování

7.3.1. Jak pracují licence

Protože 4D a 4D server jsou už založeny na TCP/IP můžete je použít přímo na Internetu nebo další TCP/IP síti (např. firemním intranetu) dokonce bez prohlížeče Web. Můžete použít 4D Client přímo přes Internet, ale pravděpodobně nedosáhnete žádného skvělého chování pokud nebudete mít digitální připojení o vysoké kapacitě. 4D Client nebyl navržen pro natěsně omezené síťové spojení – jako je propojení po telefonních linkách, ale Web prohlížeče ano. Prohlížeče pracují uspokojivě i s připojením s nízkou rychlostí, takže je ideální náhradou 4D Client jestliže takovéto připojení máte. Jestliže užíváte 4D Client přes síť použijte na 4D Server jednu uživatelskou licenci jak ostatně asi očekáváte. Jestliže se uživatel připojí Web prohlížečem není použita standardní licence klienta ale licence Web. Rovněž si pamatujte, že se uživatelé mohou prohlížečem Web připojit rovnou ke 4D a nejen ke 4D Server. Zde uvádíme některé podrobnosti licenzování pro Web:

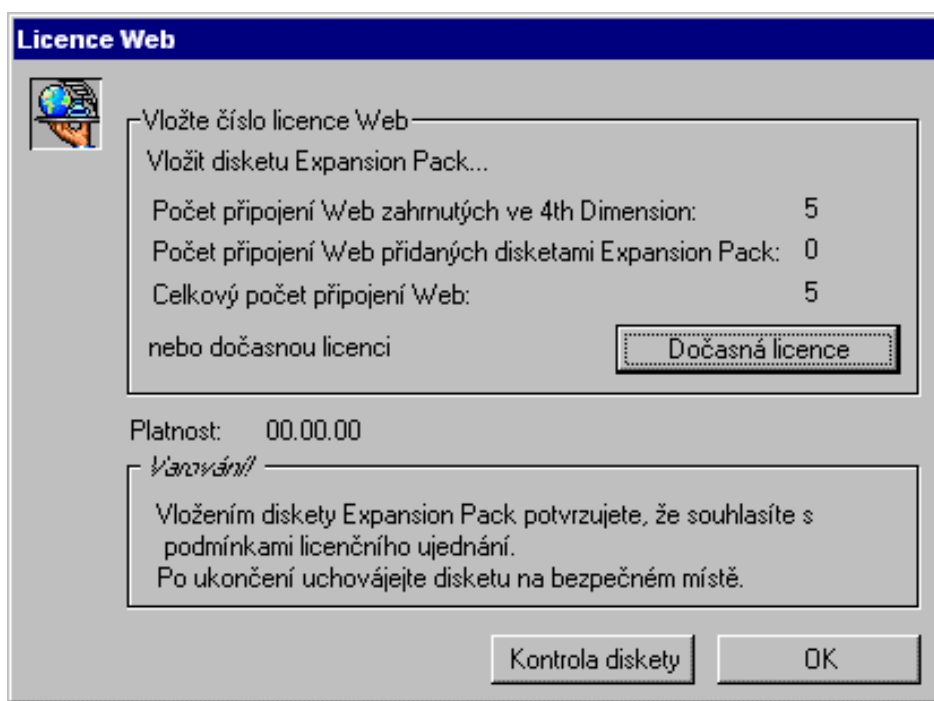
- ❖ Jednouchvatelská 4D zahrnuje licenci pro pět současných připojení Web, pokud potřebujete další licence musí být dokoupeny.
- ❖ 4D Server se poněkud liší. Jestliže vlastní x licenci pro připojení 4D Client pak můžete mít 2x připojení pro Web

7.3.2. Aktualizace licence

Váš počet licenci můžete přidat přes okno Předvolby. V tomto okně klepněte na tlačítko Licence a objeví se následující okno:

Používání 4D Databáze na Webu

Konfigurace



Toto okno funguje stejně jako okno obnovy licencí na 4D Server, ale je oddělené. Licence Web mohou být přidány buď do 4D nebo 4D Server a proto je k přidání použito zvláštní okno.

7.4. Spuštění Web Server

Jestliže váš server právě teď neběží můžete jej nastartovat z prostředí uživatele a nabídky Web server:



Jestliže neexistuje žádný dodatečný problém na síti Web server se spustí a menu se změní.



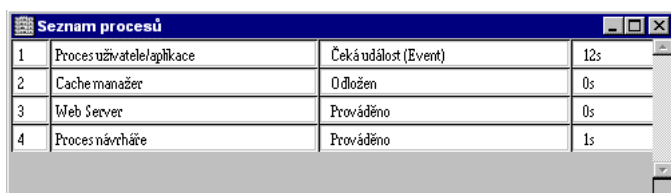
Tato nabídka vám umožní Web server vypnout. V příkazech jazyka existují příkazy, které provádějí totéž. Jsou to START WEB SERVER a STOP WEB SERVER. Tyto příkazy není potřeba komentovat, provádí přesně to co říkají. Jestliže ukončíte Web server za běhu aktivního připojení Web, Web server bude odmítat nová připojení, ale ponechá Web server spuštěný pro běžící připojení. Až se poslední běžící připojení uzavře, Web server se ukončí.

Používání 4D Databáze na Webu

Konfigurace

7.5. Co se stane při startu Web serveru

Když nastartujeme Web server, 4D nebo 4D Server spustí nový proces jádra nazvaný Web server:



	Proces uživatele/aplikace	Čeká událost (Event)	12s
2	Cache manažer	Odložen	0s
3	Web Server	Prováděno	0s
4	Proces návrháře	Prováděno	1s

Tento proces nemůžete krokovat, pozastavit ani shodit z nabídky Procesy v Prostředí návrháře. Tento proces nemá žádné okno a je řízen vnitřně 4D. Pokaždé, když se připojí nový uživatel nastartuje proces Web server nový proces pro sekci uživatele a použije pro něj jednu licenci připojení. Proces uživatele zůstane aktivní dokud uživatel nedokončí všechny úlohy nebo nevyprší čas procesu (dle nastaveného time out). Hodnotu vypršení připojení budete asi chtít nastavovat podle potřeb a případně druhu práce a připojení uživatelů.

7.6. Nastavení času vypršení připojení Web procesu

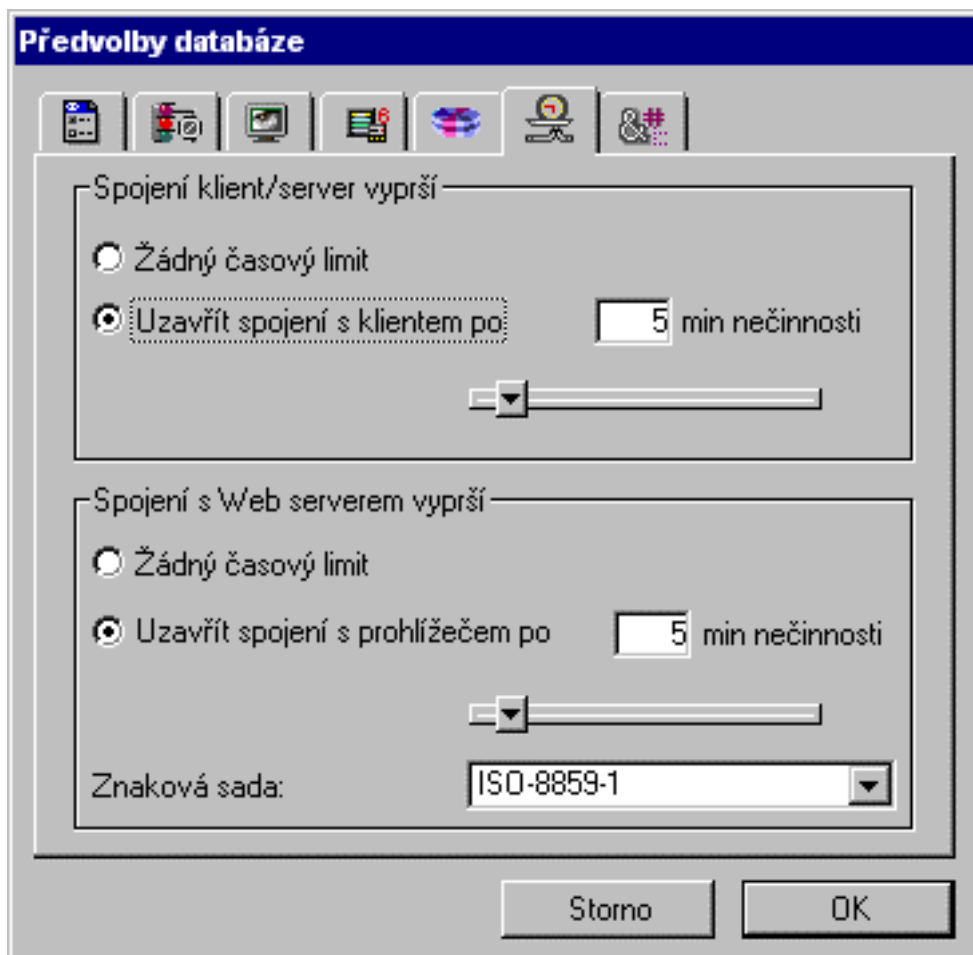
Jestliže potřebujete přenastavit výchozí hodnotu vypršení času připojení pro Web postupujte následovně:

1. Ujistěte se, že máte otevřenu správnou databázi.
2. Přejděte do Prostředí návrháře.
3. Zvolte nabídku Soubor → Předvolby databáze...
4. Otevřete stránku Spojení.

Uvidíte následující obrazovku:

Používání 4D Databáze na Webu

Konfigurace



Oblast Spojení s Web serverem vyprší, vám dovolí stanovit zda používat hlídání času pro vypršení spojení a případně určit po kolika minutách nečinnosti spojení přerušit. Pamatujte si, že když klient Webu upravuje záznam je tento záznam pro všechny ostatní uživatele uzamčen tak jako by jej používal 4D Client.

V příkazech jazyka je příkaz nazvaný SET WEB TIME-OUT, který slouží k ovládání času vypršení procedurálně. Tímto příkazem se zadává čas vypršení v sekundách:

SET WEB TIME-OUT (120) 2 minuty do vypršení

Jestliže chcete vypnout hlídání času použijte nastavení 0 sekund:

SET WEB TIME-OUT (0) Žádný čas do vypršení

Volání tohoto příkazu nezmění trvale nastavení v předvolbách v prostředí Návrháře. Příště při otevření databáze je použito originální nastavení.

Poznámka: Tento příkaz má efekt pouze na další připojení k databázi. Nemá vliv na již běžící připojení. Rovněž je potřeba si pamatovat, že to je příkaz, který bude použit na všechna připojení. Bohužel neexistuje příkaz, který má vliv na běžící připojení a pouze toto jedno připojení.

Používání 4D Databáze na Webu

Konfigurace

7.7. Minimální požadavky

Aby jste se mohli k vaší databázi připojit přes Web potřebujete mít ve vaší databázi přístup do Prostředí aplikace. 4D neaktivuje Prostředí aplikace, dokud nepřidáte alespoň jedno záhlaví nabídek do vašeho projektu, takže potřebujete minimálně jedno záhlaví nabídek aby jste databázi mohli pro Web použít. Dokonce, i když neplánujete používat tuto databázi přes nabídky 4D na Webu musíte alespoň jedno záhlaví vytvořit.

Používání 4D Databáze na Webu

Základy rozhraní Web

8. Základy rozhraní Web

8.1. Úvod

Nyní, když máme databázi správně nakonfigurovanou pro Web jsme schopni použít většinu formulářů metod a nabídek existujících v databázi pro připojení Web. Ujistěte se, že máte spuštěný prohlížeč Web a připojte se k databázi. Pamatujte si, že pokud jste nezměnili výchozí přiřazení portu databázi, nemusíte číslo portu v adrese vašeho stroje uvádět. Po připojení obdržíte HTML překlad záhlaví nabídek #1.

8.2. Přidání jednoduchého rozhraní

Do této doby jsme viděli některé věci, které fungují nebo nefungují zcela korektně při překladech pro Web. Během zbytku tohoto kursu se naučíme jak jednoduché je uzpůsobit existující objekty pro Web a jak vytvořit nové zaměření pro Web. Nyní si vytvoříme jednoduchý interface k našemu předchozímu příkladu. Přidáme nové záhlaví nabídek, určené speciálně pro připojení Web a přidáme jeden řádek kódu do metody databáze Při Web spojení.

8.2.1. Přidání Testovací metody

1. Zvolte Návrh → Nová metoda...
2. Napište název metody: M_WEB_TestAlert
3. Přidejte následující kód:

```
If (False)
  ` Method: M_WEB_TestAlert
  ` ACI University Programming Classes
  ` Created By: David Adams
  ` Date: 3/21/97

  ` Purpose: Ukazuje ALERT na WEB.

  ⋄f_Version6x40 := True
  ⋄fD_Adams := True

End if

ALERT ("Toto je mé upozornění pro Web.")
`End of method
```

4. Zavřete všechna okna.

Poznámka: Do textu upozornění můžete napsat jakoukoliv jinou větu, které je v souladu se zákony vaší země.

8.2.2.. Vytvoření nabídky pro Web

Neexistují žádná speciální omezení pro vytvoření nabídky pro Web. Můžete použít libovolné záhlaví nabídek databáze. Již jsme viděli použité záhlaví 1 - výchozí záhlaví - v akci. Nyní se chystáme přidat nové záhlaví nabídek do databáze a to stejným způsobem jak jsme je přidávali v předchozím příkladu pětiminutové databáze:

1. Vyberte Nástroje → Editor nabídek

Používání 4D Databáze na Webu

Základy rozhraní Web

2. Stiskněte tlačítko Přidat a vytvořte nové záhlaví.
3. Vyberte nabídku Soubor ve vašem novém záhlaví a klepněte na tlačítko Vymazat v pravém rohu obrazovky.
4. Klepněte na tlačítko Přidat nabídku.
5. Pojmenujte novou nabídku Web cvičení.
6. Klepněte na tlačítko Přidat položku v pravém spodním rohu.
7. Pojmenujte položku Test upozornění.
8. Přeneste okno Průzkumníka na popředí.
9. Potáhněte název metody projektu: M_WEB_TestAlert k přiřazení do vaší nové položky nabídky.
10. Stiskněte klávesu tabelátor a uzavřete Editor nabídek.

8.2.3. Úprava metody Při Web spojení

Nakonec musíme 4D říci, aby při připojení uživatele přes Web odeslala naše nové záhlaví nabídek místo záhlaví 1. To nám zabere pouze jednu řádku kódu v metodě databáze Při Web spojení. V dalších částech tohoto kursu strávíme trochu více času při psaní této metody, ale nyní se chystáme změnit pouze záhlaví nabídek, které je automaticky odesíláno, když se připojí Web klient. Proved'te následující:

1. Zvolte Návrh → Upravit Metodu...
2. Objeví se Průzkumník. Klepněte na ikonu Plus vlevo od Metody databáze a rozevřete seznam sedmi metod databáze.
3. Poklepejte na název metody Při Web spojení a otevřete ji.
4. Vyberte typ metody Textová.
5. Přidejte následující kód, kterým 4D sdělíte, aby odeslala nové záhlaví nabídek:

```
If (False)
  ` Database Method: On Web Connection
  ` ACI University Programming Classes
  ` Created By: David Adams
  ` Date: 3/21/97

  ` Purpose: Nastavení prostředí Web

  ⋄f_Version6x40 := True
  ⋄fD_Adams := True

End if

MENU BAR(5)
  ` End of method
```

6. Zavřete všechna okna.

Tento kód sdělí 4D, že má odeslat nové záhlaví nabídek Web klientovi okamžitě po navázání spojení k vyzkoušení položky nabídky.

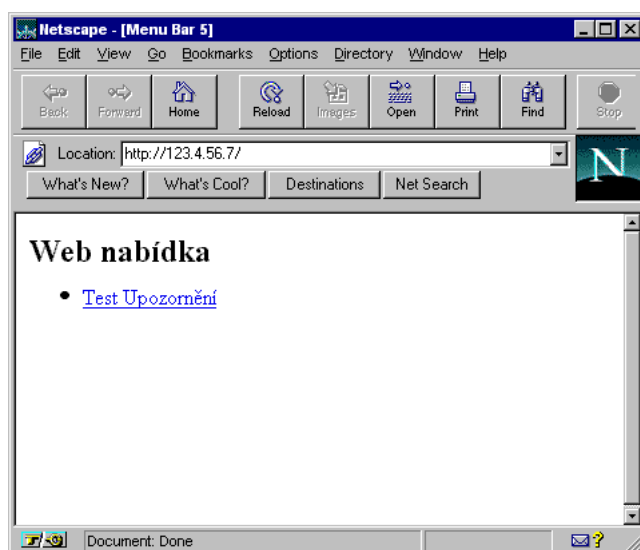
Používání 4D Databáze na Webu

Základy rozhraní Web

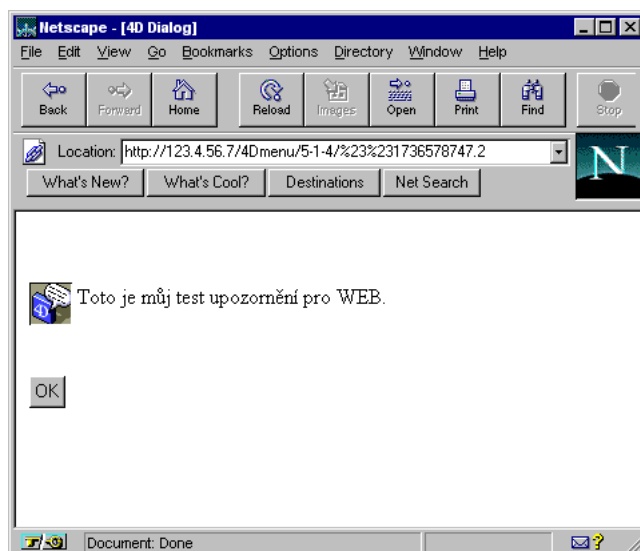
8.2.3. Testování vaší rutiny

Nyní se přesvědčíme, že vše pracuje správně.:

1. Otevřete váš prohlížeč.
2. Po otevření prohlížeče napište IP adresu vašeho počítače a klepněte enter.
3. Měli by jste vidět následující obrazovku:



4. Klepněte na odkaz Test upozornění.
5. Měli by jste vidět následující obrazovku:



6. Klepněte na tlačítko OK a navraťte se do původní obrazovky.

Používání 4D Databáze na Webu

Základy rozhraní Web

Nyní jsme přidali několik jednoduchých prvků do databáze, které postupně rozšíříme v budoucích cvičeních. Naším následujícím krokem bude naučit se více o vytváření formulářů na Web.

Používání 4D Databáze na Webu

Formuláře Web Aware

9. Formuláře Web Aware

9.1. Úvod

4D automaticky převádí své formuláře do HTML. Dříve během pětiminutové Web databáze, jste viděli jak rychle a jednoduše lze vytvořit samotnou 4D databázi, která spolupracuje s prohlížečem Web. Nyní se chystáme naučit jak vytvářet vaše 4D formuláře, aby jste dosáhli co nejlepšího výsledku pro Web. Vytváření speciálních formulářů vyladěných pro klienty Web je přinejmenším stejně důležité jako vytvoření odlišných formulářů pro uživatele Windows a Macintosh.

9.2. Automatické převody 4D.

Když zobrazíte 4D upozornění, potvrzení, chybovou zprávu, dialog nebo záznam ve formuláři, uživatel Web vidí porovnatelné objekty, které fungují v mnohém podobně jako originály ve 4D. Jestliže zobrazíte, přidáte nebo upravujete záznam, 4D převádí vaše formuláře a naplňuje je odpovídajícími daty záznamu. Všechno toto se děje automaticky a vy nepotřebujete vědět nic o HTML, aby jste tyto rysy mohli používat. V tomto kursu se chystáme hovořit o HTML, protože některé znalosti vám pomohou vytvořit lepší rozhraní pro své Web klienty. Nechystáme se hovořit přímo o kódování HTML, ale nyní se zmíníme o omezeních, která HTML mají.





9.3. HTML formuláře

Původně byly všechny HTML zaměřeny pouze a jenom k formátování textu. Jestliže chcete jakýkoliv druh interakce, vkládat oblasti, tlačítka, textové oblasti HTML potřebujete použít formulář HTML. Když 4D převádí formuláře 4D a záznamy převede je do formuláře HTML. HTML formuláře podporují mnohem méně objektů rozhraní než 4D a nezahrnují žádný vestavěný systém maker. Později v tomto kursu budeme hovořit o JavaScriptech a způsobu jak přidávat některá makra do formulářů HTML, ale JavaScript není součástí HTML a rozhodně není podporován všemi prohlížeči.

Můžete převést na Web libovolný formulář 4D, ale některé určité objekty se nemusí převést dobře nebo se nepřevedou vůbec, proto je tak důležité vytvořit speciální formuláře vyladěné pro Web. Existuje několik figlů pro vytvoření dobrých webovských formulářů, které jsme již uváděli (Web Aware).

9.4. Převody objektů 4D Formulářů

Uvedme si přehled objektů 4D, které se převádějí dobře do objektů HTML formulářů. Jak uvidíte existuje jich poměrně mnoho. Obrázky v následující tabulce jsou sejmuty z obrazovek 4D a Netscape.

Typ objektu 4D	4D	Web
Obrázek		
Čára		

Používání 4D Databáze na Webu

Formuláře Web Aware

Statický text	This is some static text.	This is some static text.
Oblast pro zadávání	<input type="text" value="Enterable"/>	<input type="text" value="Enterable"/>
Nedostupná oblast	Non-enterable	Non-enterable
Tlačítko	<input type="button" value="Button"/>	<input type="button" value="Button"/>
Výchozí tlačítko	<input type="button" value="Default Button"/>	<input type="button" value="Default Button"/>
Přepínač	<input checked="" type="radio"/> Radio Button	<input checked="" type="radio"/> Radio Button
Zaškrtnutí políčko	<input checked="" type="checkbox"/> Check Box	<input checked="" type="checkbox"/> Check Box
Nabídka	<input type="text" value="Popup Menu"/>	<input type="text" value="Popup Menu"/>
Rozevírací seznam	<input type="text" value="Drop Down List"/>	<input type="text" value="Drop Down List"/>
Nabídka/Seznam	<input type="text" value="Menu Drop Down ..."/>	<input type="text" value="Menu Drop Down List"/>
Combo Box	<input type="text" value="Combo Box"/>	<input type="text" value="Combo Box"/>
Posuvná oblast	<div>Scrollable Area</div>	<div>Scrollable Area</div>

Jak můžete vidět tlačítka, obrázky, čáry, nabídky všechny druhů i posuvné oblasti jsou převáděny dobře. 4D dokonce převede barevné nastavení objektů, když je to možné. Povšimněte si, že všechny druhy objektů array vyjma posuvné oblasti se převrátí do rozevíracích nabídek v HTML. To je proto, že HTML má pouze jeden druh rozevíracích nabídek a posuvné oblasti. Vytvoříme formulář, který použije některé z těchto objektů a podívejme jak snadno jste se stali odborníky na Web aniž by jste znali cokoliv o HTML.

Používání 4D Databáze na Webu

Formuláře Web Aware

9.5. Omezení pro převody 4D formulářů

9.5.1. Úvod

Mohli jste si všimnout, že z minulého cvičení bylo vynecháno několik objektů 4D formulářů a tabulka. Některé objekty 4D formulářů se díky omezením HTML špatně předkládají. Je zde také několik dalších omezení od různých typů prohlížečů, kterých by jste si při navrhování systému měli být vědomi.

9.5.2. Objekty formulářů 4D, které nepřekládáme do HTML

Zde vidíme seznam objektů formulářů 4D, které se špatně překládají nebo se vůbec nepřekládají a proto by jste následující objekty neměli na svých formulářích pro Web vůbec používat:

Hierarchické nabídky	Hierarchické seznamy	Stránkové řízení
Obrázkové nabídky	Obrázková tlačítka	Obrázkové přepínače
Teploměry	Pravítka	Výseče
Vertikální čáry	Ovály	Obdelníky
3D zaškrtávací políčka	3D Přepínače	
Skupinové posuvné oblasti	Podformuláře	

Je jednoduché pracovat bez většiny výše uvedených objektů. Skutečný nedostatek budeme pociťovat pouze u skupinových posuvných oblastí a podformulářů. Podívejme se všeobecně na omezení jednotlivých prohlížečů a potom na některé specifické prvky které budou mít vliv na vytváření Web systémů se 4D.

9.6. Omezení prohlížečů

9.6.1. Úvod

Prohlížeče HTML začali jako velice jednoduché softwary, které zobrazovali jednoduše formátovatelný text a klepnutelné odkazy. Web se nerozšířil do té doby než se objevil první grafický prohlížeč Mosaic s jednoduchou grafikou a moderním uživatelských rozhraním. Od té doby se objevila řada dalších prohlížečů zvláště Navigator a zavedla nové rysy, které nebyly součástí oficiální HTML definice. V této době je oficiální verze HTML verze 3.2. a její rysy zahrnují téměř všechny novinky uvedené Netscape a Microsoft. Problém je, že ne každý uživatel používá poslední verzi prohlížeče vedoucích prohlížečů, takže není jednoduché říci, které rysy váš uživatel využije. Jediné místo, kde můžete mít kontrolu je vaše vnitřní podniková síť.

Používání 4D Databáze na Webu

Formuláře Web Aware

9.6.2. Rozdíly prohlížečů

Podívejme se na rozdíly u jednotlivých prohlížečů

Browser	Grafika	Formuláře	Tabulky	Rámečky	Cookies	JavaScript
Lynx		√				
Netscape 1.0	√	√				
Netscape 1.1	√	√	√		√	
Netscape 2.0	√	√	√	√	√	√
Netscape 3.0	√	√	√	√	√	√
IE 3.0	√	√	√	√	√	
IE 3.02	√	√	√	√	√	√

Tato tabulka je seznamem několika hlavních rysů. Existují tucty dalších rysů a tagů, které byly přidány během posledních několika let, a které jsou některými prohlížeči podporovány a jinými ne. Je potřeba poznamenat, že dokonce jestliže užíváte poslední verze Netscape nebo Internet Explorer nemusí tyto podporovat všechny rysy. Např. automatické zavádění obrázků a grafiky je často vypínáno lidmi s velice pomalým modemovým spojením tzn., že tito efektivně využívají pouze textovou podporu prohlížeče. Takovýto uživatelé si rovněž mohou vypnout podporu JavaScript v předvolbách. Neexistuje způsob jakým byste se dozvěděli jaké nastavení si uživatel vybral.

9.6.3. Jaké typy prohlížečů lidé používají ?

Na papíře existují stovky různých prohlížečů, ale většina lidí používá Mosaic nebo některý z jeho klonů, Navigator nebo Internet Explorer. Většina lidí v této době nepoužívá poslední verze těchto prohlížečů např. prohlížeč AOL je zcela nemoderní, ale přitom poměrně rozšířený. Pamatujte si rovněž, že pouze textové prohlížeče typu Lynx jsou překvapivě poměrně široce rozšířeny. Lynx je druh textového prohlížeče UNIX často se vyskytující na terminálech veřejných sítí (knihovny, instituce, školy) a je součástí některých sítí.

9.6.4. Dohled nad prohlížeči: Intranet vs. Internet

Jak jste viděli mnoho pokročilých rysů prohlížečů je specifických pro daný prohlížeč. Jestliže se chystáte svůj systém použít na Internetu nemůžete předpokládat, že uživatelé mohou vidět tabulky, rámečky, formuláře nebo obrázky. Také nemůžete předpokládat, že na jejich systému budou pracovat JavaScript. Dokonce uživatelé s prohlížeči, které tyto rysy podporují je nemusí mít zapnutý. Jestliže se chystáte vytvořit informační systém pro síť společnosti a jste v pozici, kde můžete kontrolovat nasazení a použití prohlížečů pak můžete předpokládat, že rysy těchto prohlížečů budou vašim uživatelům dostupné. Jinými slovy můžete vytvořit systém založený na pokročilých rysech prohlížečů, jestliže jej vytváříte pro domácí použití.

Používání 4D Databáze na Webu

Formuláře Web Aware

9.6.5. Speciální stránky pro různé typy prohlížečů

Ještě bych zmínil jednu myšlenku a to, že můžete zasílat formuláře v závislosti na prohlížeči používaném uživatelem. V tomto kurzu to nebudeme vysvětlovat do detailů, ale chceme to jednoduše popsat pro ty, kteří někdy tento rys budou potřebovat. Poněkud později uvidíme jak metoda databáze Při Web spojení vám dovolí určit spoustu informací o uživateli včetně názvu a verze prohlížeče. Tyto informace můžete použít k odesílání různých 4D formulářů nebo vkládaných HTML bloků či vyladění stránek pro konkrétního uživatele. Ve skutečnosti to není obtížné programování ve 4D a je to přitom sofistikovaný přístup při publikování databáze. Hlavní nevýhoda tohoto přístupu je, že zabere mnohem více času a práce při sestavení specializovaných formulářů, souborů a metod pro odlišné prohlížeče.

9.7. Univerzální omezení

9.7.1. Úvod

Již jsme viděli poměrně velký seznam objektů 4D, které nemůžete použít. Existuje jich několik, které si zaslouží speciální poznámku, protože se ve 4D používají běžně a přitom nejsou podporovány žádným prohlížečem HTML: vkládání obrázků, skupinová pole a podformuláře.

9.7.2. Obrázky

Na Web můžete zobrazovat obrázky, již jsme to ostatně viděli. Se 4D můžete zobrazovat statickou grafiku, obrázkové proměnné a obrázková pole bez větších těžkostí. Co však nemůžete je vytvořit formulář, který umožňuje vkládání obrázků. Jediná cesta jak to v současné době uskutečnit je pomocí Web serveru, který podporuje přijímání HTTP souborů (většina Web serverů toto nepodporuje) a pomocí rozhraní Web, které umožní uživateli určit, který soubor nebo soubory z jejich počítače mají být odeslány. Tuto možnost máte jestliže použijete 4D jako CGI (externí program) k Web serveru jako je WebSTAR prostřednictvím Netlink/4D. CGI a Netlink budeme probírat později.

9.7.3. Žádná skupinová pole

9.7.3.1. Úvod

Skupinové posuvné oblasti jsou výborným rysem 4D. Jsou často nejefektivnějším způsobem prezentování informace uživateli, proto jsou také široce využívána. Naneštěstí HTML nemá žádný koncept podobný skupinovému oblaskem. Můžete odeslat z 4D array sdružená do skupin, ale v prohlížeči Web se rozdělí do svých jednotlivých částí. Podívejme se co se stane.

9.7.3.2. Přidání testovací metody

1. Vytvořte novou metodu M_WEB_ChoiceListTest a to následovně:

```
If (False)
  ` Method: M_WEB_ChoiceListTest
  ` ACI University Programming Classes
  ` Created By: David Adams
  ` Date: 3/21/97

  ` Purpose: Ukazuje skupinová pole na WEB.

  ⋄f_Version6x40 := True
  ⋄fD_Adams := True
```

Používání 4D Databáze na Webu

Formuláře Web Aware

End if

CHO_FillCustomerArrays

CHO_DisplayChoiceList

`End of method

4. Zavřete nebo uložte metodu.

9.7.3.3. Přidání položky nabídky

1. Jděte do Prostředí návrháře.
2. Vyberte Nástroje → Editor nabídek.
3. Otevřete vaši nabídku Web cvičení
4. Otevřete nabídku poklepáním na ikonu.
4. Stiskněte tlačítko Přidat v pravém dolním rohu.
5. Pojmenujte novou položku nabídky: Test seznamu
6. Přiřaďte metodu M_WEB_ChoiceListTest k nové položce nabídky.
7. Stiskněte tabelátor a zavřete Editor nabídek.

9.7.3.4. Testování metody ve 4D

1. Jděte do Prostředí uživatele.
2. Vyberte nabídku Zvláštní → Spustit metodu...
3. Vyberte a spusťte vaši novou metodu: M_WEB_ChoiceListTest
4. Měli by jste viděl následující okno:



Tento dialog obsahuje skupinu posuvných oblastí a ukazuje informace o zákaznících. Systém skupin posuvných oblastí je užíván ve 4D alternativně k jejím vestavěným rysům ukazatelů na záznamy. Zde je používáme, abychom viděli co se stane se skupinou array na Web. Nyní uzavřete tento dialog ve 4D.

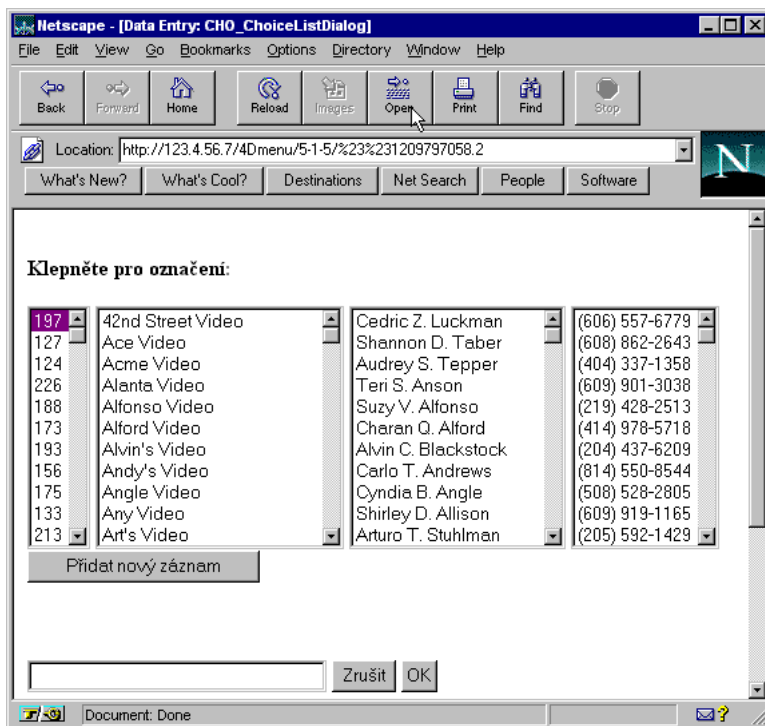
9.7.3.5. Testování metody v prohlížeči Web

1. Přejděte do Web prohlížeče.
2. Napište IP adresu vašeho stroje a klepněte enter.

Používání 4D Databáze na Webu

Formuláře Web Aware

3. Vyberte nový odkaz "Test seznamu".
4. Měli by jste vidět okno podobné tomu na následujícím obrázku:



5. Zmenšíte okno Netscape a sledujte co se stane se skupinou oblastí.

Každé ze zdrojových polí je překonvertováno jako nezávislý HTML objekt. Protože HTML nemají žádný objekt skupiny polí, 4D je nemůže vytvořit.

Poznámka: Jestliže v otevřené 4D po spuštění této rutiny uvidíte prázdné okno, jděte do seznamu procesů v Prostředí návrháře a ukončete běžící připojení Web.

9.7.4. Použití pole o pevné délce doplněné mezerami

Jedním z řešení nepřítomnosti skupinových polí v prohlížeči je překonvertovat řadu polí do jednoho textového pole zobrazeného v písmu se znaky stejné délky. Protože písmo má znaky stejné délky můžete použít mezery k oddělení údajů jednotlivých sloupců a vidět je v prohlížeči správně pod sebou. Nebezpečí tohoto přístupu je, že v HTML neexistuje spolehlivý způsob pro určení písma nebo stylu písma pro posuvné oblasti a uživatel může změnit nastavení písem svého prohlížeče. Např. uživatel změní písmo o znacích pevné délky na jiné písmo, které se mu líbí více a výsledkem bude totální zmatek.

Kromě toho prohlížeče odstraňují vkládané mezery, pokud tyto mezery neuzavřete tagem `<PRE>` `</PRE>`

Pro ty, které zajímá tato technika je ve školících materiálech příkladová rutina nazvaná Pad_string.

9.7.5. Žádné podformuláře

Podformuláře ve 4D jsou výborným způsobem jak s minimálním úsilím zobrazit informace ze vztažených souborů. HTML neobsahuje koncept záznamů natož tedy vztažených záznamů. Proto podformuláře nejsou

Používání 4D Databáze na Webu

Formuláře Web Aware

vůbec převedeny, jestliže chcete simulovat chování podformulářů musíte použít buď pole s vyplňovanými mezerami z předchozího příkladu a často odesílat a znovu zobrazovat formulář nebo napsat pokročilý JavaScript.

9.7.6. Žádné automatické ověřování či kontrola vstupů

Samotné formuláře HTML nemají žádnou inteligenci jsou to pasivní objekty pro vstup a řízení. Jediné vestavěné chování je, že tlačítko Odeslat provede údajost formuláře (odešle jej) a tlačítko Storno vyprázdní všechny hodnoty ze vstupních oblastí. Při použití 4Dv6 můžete automaticky využít výhod mnoha z vašich metod objektu bez jakékoliv speciální práce jestliže, ale chcete provádět ověření polí přímo v prohlížeči Web, musíte do vašeho systému zahrnout JavaScript. Oba tyto způsoby budeme probírat v dalších částech tohoto kursu.

9.8. Vytvoření výstupního formuláře

Nyní, když již máte přehled o tom co můžete a nemůžete zahrnout do HTML formulářů, podívejme se na specifický výstupní formulář pro Web a zblízka se podívejme jak pracuje.

9.8.1. Vytvoření výstupního formuláře

1. Jděte do Prostředí návrháře
2. Vyberte Návrh \rightarrow Nový formulář...
3. Vyberte Zákazníci z nabídky Vytvořit formulář pro tabulku.
4. Obrazovka v Editoru formulářů by měla nyní vypadat takto:

Používání 4D Databáze na Webu

Formuláře Web Aware

Nový formulář

Vytvořit formulář pro tabulku:

Název formuláře:

Typ formuláře:

Užitý vzor:

Dostupná pole:

- Adresa
- ČasUpravení
- ČasVytvoření
- DatumUpravení
- DatumVytvoření
- Důležitý
- Firma
- IDZákazníka
- Jméno

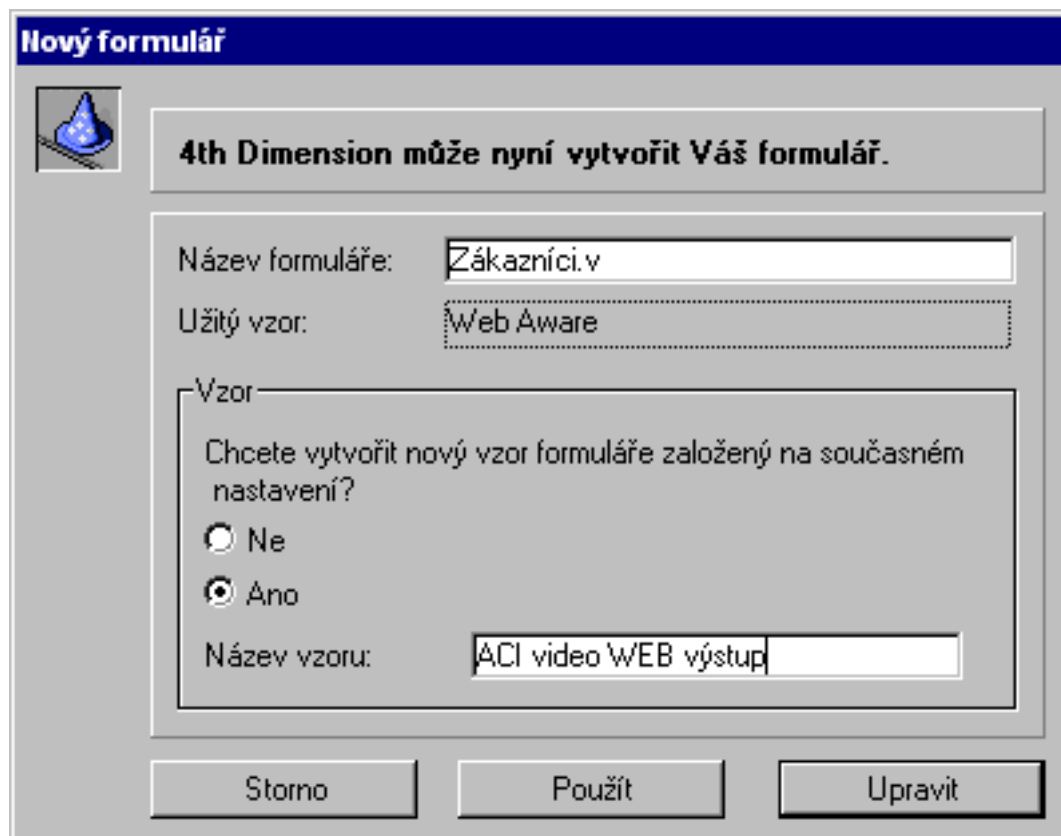
Vybraná pole:

- IDZákazníka
- Jméno
- Příjmení

5. Pojmenujte formulář Zákazníci.ow
6. Vyberte Seznam z nabídky Typ formuláře.
7. Vyberte Web Aware z nabídky Užitý vzor.
8. Přidejte následující pole v tomto pořadí: Společnost, Adresa, Město.
9. Vyberte tlačítko Rozšířit
10. Vyberte stránku Tlačítka
11. Vyberte druh tlačítek Běžná tlačítka.
12. Stiskněte tlačítko OK
13. Klepněte na přepínač ANO k založení nového vzoru.
14. Pojmenujte nový vzor ACI Video Web výstup.

Používání 4D Databáze na Webu

Formuláře Web Aware



15. Klepněte na Použít.
16. Nyní jste v prostředí uživatele. Vyberte z nabídky Vybrat→Všechny záznamy
17. Formulář vypadá dobře a je použitelný pro 4D a ne pouze pro WEB.

9.8.3. Přidání testovací metody

1. Zvolte Návrh → Nová metoda ...
2. Vytvořte novou metodu projektu M_WEB_ShowCustomers následujícím způsobem:

If (False)

```
` Method: M_WEB_ShowCustomers  
` ACI University Programming Classes  
` Created By: David Adams  
` Date: 3/21/97
```

```
` Účel: Vytvorení jednoduché metody k ukázání výberu na WEB
```

```
øf_Version6x40 := True
```

```
øfD_Adams := True
```

End if

```
OUTPUT FORM ([Zákazníci];" Zákazníci.ow")
```

```
ALL RECORDS ([Zákazníci])
```

```
SET WEB DISPLAY LIMITS (5) ` ukaze pouze 5 zaznamu.
```

Používání 4D Databáze na Webu

Formuláře Web Aware

MODIFY SELECTION ([Zákazníci];*)
`End of method

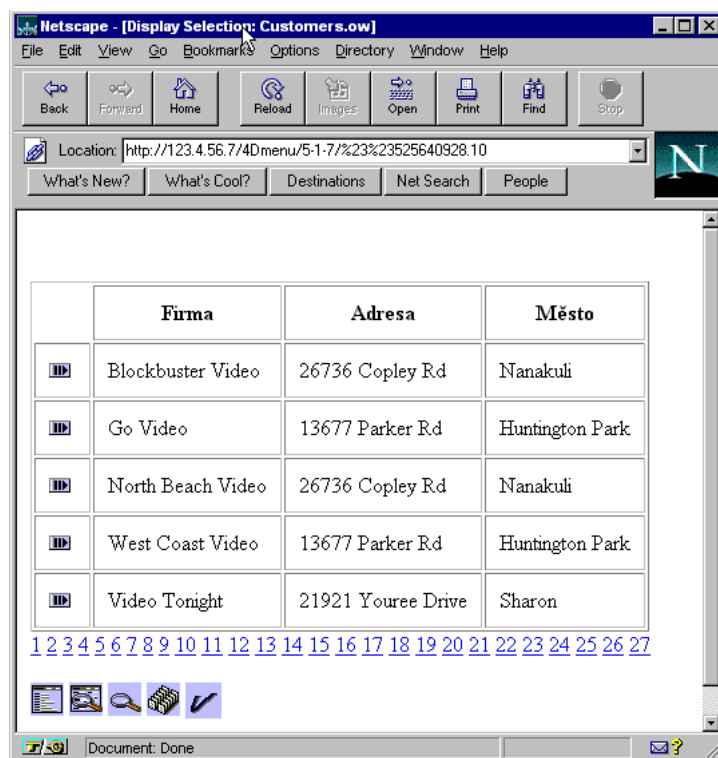
3. Uzavřete nebo uložte metodu.

9.8.4. Přidání položky nabídky

1. Přejděte do Prostředí návrháře.
2. Zvolte Nástroje → Editor nabídek.
3. Otevřete nabídku Web cvičení.
4. Rozšiřte nabídku klepnutím na trojúhelník.
4. Klepněte na tlačítko Přidat položku v pravém rohu.
5. Pojmenujte novou položku: Ukázat zákazníky
6. Přiřaďte metodu M_WebShowCustomers k nové položce.
7. Klepněte na klávesu Tabelátor a uzavřete Editor nabídek.

9.8.5. Testování formuláře v prohlížeči Web

1. Přejděte do prohlížeče.
2. Napište IP adresu svého stroje a klepněte Enter
3. Klepněte na nový odkaz “Ukázat zákazníky”.
4. Měli by jste vidět následující okno, které zobrazuje 5 zákazníků, které vypadají asi následovně:



Používání 4D Databáze na Webu

Formuláře Web Aware

Ponechte toto okno v prohlížeči Web dokud budeme ve 4D upravovat formulář a budeme chtít vidět co se děje v Netscape. Všimněte si, že výstupní formulář v Netscape je skutečně HTML tabulka o čtyřech sloupcích. Sloupce vlevo ukazuje pouze grafiku na kterou lze klepnout, a která obsahuje odkaz na konkrétní záznam, který nám umožní otevřít a upravovat tento záznam. Další tři sloupce jsou jednotlivá pole z databáze. Podívejme se jak vám 4D dovolí řídit tyto objekty.

9.8.6. Nastavení počtu zobrazených záznamů na Web

V tomto příkladu jste použili nový příkaz nazvaný SET WEB DISPLAY LIMITS pro řízení počtu záznamů zaslaných současně webovskému prohlížeči. Všimněte si, že výstup na Web zobrazuje ne více než tento počet záznamů a obsahuje řadu číselných odkazů na další záznamy. To co 4D dělá je automatické řízení stránkování vašeho výběru. Uživatel může volně přecházet uvnitř výběru aniž by jste museli programovat. Proč jsou výstupní zobrazení na Web stránkovány tímto způsobem?

- ❖ Rozhodně není dobrý způsob posílat přes Web mnoho záznamů najednou. Lidé často mají pomalé připojení a prohlížeče HTML nejsou tak rychlé dokonce ani v plně digitalizovaném síťovém připojení.
- ❖ Výstupní zobrazení je formátováno jako HTML tabulky, jak by ostatně i mělo být. Prohlížeč Web nevykreslí žádnou část tabulky dokud neobdrží celou tabulku. Je to proto, že formát tabulky může být změněn kdekoli uvnitř tabulky, takže prohlížeč potřebuje celou tabulku, aby ji byl schopen správně zobrazit (buňky mohou být spojovány a tabulky vkládány dovnitř sebe, 4D toto nedělá, ale prohlížeč to samozřejmě neví). To znamená, že uživatel vidí prázdnou obrazovku dokud není tabulka natažena. S velkými tabulkami by uživatel mohl nabýt dojmu, že systém zatuhl.

SET WEB DISPLAY LIMITS je jednoduše použitelný, prostě zadejte maximální počet záznamů, které chcete zobrazovat najednou a 4D provede zbytek. Výchozí hodnota pokud neřeknete jinak je 20 záznamů.

Poznámka: Tento příkaz můžete taky použít pro nastavení maximální počtu stránek při zobrazení a použít jiný obrázek pro klepnutelný grafický odkaz. Podrobnosti lze nalézt v referenční kapitole na konci této příručky.

Poznámka: Jestliže jste již někdy doopravdy ručně vytvářeli stránkování vašich vlastních HTML stránek pro zobrazení více záznamů, velice dobře víte kolik nezájímavé dlouhodobé práce vám automatické stránkování 4D ušetří. Řekněme hodně a nechme to tak. .

9.8.7. Zvýšení počtu stránek uchovaných v paměti

1. Upravte metodu projektu M_WEB_ShowCustomers následujícím způsobem:

- ❖ SET WEB DISPLAY LIMITS (5;100) ` Ukazuje pouze 5 záznamů., vytváří 100 stránek
2. Uložte metodu M_WEB_ShowCustomers
 3. Otestujte tuto změnu ve vašem Web prohlížeči

9.8.8. Výstupní formuláře Web Aware: podrobnosti

1. Stiskněte grafické tlačítko na spodu seznamu záznamu v prohlížeči Web.
2. Nyní by jste měli být zpět ve vaší nabídce Web cvičení zobrazené v prohlížeči Web.
3. Ve 4D přejděte do Prostředí návrháře
4. Otevřete formulář Zákazníci.ow
5. Vyberte černý rámeček, který rámuje pole Město a jeho popis.

Používání 4D Databáze na Webu

Formuláře Web Aware

6. Potáhněte rámeček pod čáru zápatí (P), ale neměňte žádný jiný objekt.
7. Uložte vaše změny, ale okno formuláře nechte otevřené.
8. Přejděte do prohlížeče Web.
9. Klepněte na odkaz Ukázat zákazníky.
10. Pole Město zmizelo, pokud ne změny provedené ve formuláři jste neuložili.
11. Klepněte na grafické tlačítko hotovo a vraťte se do nabídky Web cvičení.

Obdelníky ve výstupním formuláři jsou to co způsobí, že 4D zná kde má vytvořit sloupec tabulky HTML. Jestliže vložíte dvě pole dovnitř téhož obdelníku budou vykreslena v téže buňce sloupce HTML a jestliže odstraníte obdelník, pole nebude ukázáno vůbec. Toto je jednoduchý a pružný způsob jak můžete řídit co se má zobrazit a jak. Zde je několik pravidel, která by jste měli znát:

- ❖ Jestliže se obdelník dotkne objektu záhlaví pak je sloupec zahrnut a objekt záhlaví je vykreslen.
- ❖ Jestliže se obdelník dotkne oblasti záhlaví, ale ne objektu záhlaví pak je sloupec zahrnut, ale objekt záhlaví není vykreslen.
- ❖ Jestliže se obdelník dotkne pole v oblasti obsahu, je pole zahrnuto. Jinak je sloupec prázdný.
- ❖ Jestliže použijete `MODIFY SELECTION` nebo `DISPLAY SELECTION` jsou kontrolovány oblasti formuláře záhlaví, obsah a zápatí. Objekty v oblasti záhlaví se objeví jednou na vrcholu stránky HTML. Objekty obsahu se opakují jednou pro každý záznam a objekty zápatí se objeví jednou pod posledním záznamem a všemi automatickými odkazy stránkování přidanými 4D.

9.9. Vytvoření zadávacího formuláře/formuláře pro jeden záznam

Vytvoření zadávacího formuláře je zrovna tak jednoduché jako vytvoření výstupního formuláře, již jsme to ve skutečnosti prováděli v našich předchozích cvičeních, ale nyní se chystáme se podívat na tuto práci podrobněji.

9.9.1. Vytvoření formuláře

1. Přejděte do Prostředí návrháře 4D.
2. Zvolte Návrh → Nový formulář...
3. Vyberte Zákazníci z nabídky Formulář pro tabulku.
4. Pojmenujte formulář Zákazníci.iw
5. Vyberte z nabídky Typ formuláře S obsahem (vstupní)
6. Vyberte Web Aware z nabídky Užítý vzor
7. Vyberte následující pole v tomto pořadí: Firma, Adresa, Město, Stát, PSČ, Jméno, Příjmení a Telefon.
8. Okno v Editoru formulářů by mělo vypadat následujícím způsobem:

Používání 4D Databáze na Webu

Formuláře Web Aware

Nový formulář

Vytvořit formulář pro tabulku: Zákazníci

Název formuláře: Zákazníci.iw

Typ formuláře: S obsahem (vstupní)

Užitý vzor: Web Aware Vymazat

Dostupná pole: Vztažené tabulky

- ☒ IDZákazníka
- ☒ Jméno
- ☒ Příjmení
- ☒ Firma
- ☒ Adresa
- ☒ Město
- ☒ Stát
- ☒ PSČ
- ☒ Telefon

Vybraná pole:

- Firma
- Adresa
- Město
- Stát
- PSČ
- Jméno
- Příjmení
- Telefon

Rozšířit... Storno Upravit Použít 🔑

9. Klepněte na tlačítko Rozšířit.
10. Přejděte na stránku Tlačítka.
11. Vyberte typ tlačítka VCR
12. Klepněte na OK.
13. Zvolte přepínač Ano a vytvořte nový vzor.
14. Pojmenujte váš nový vzor ACI Video Web Vstup
13. Klepněte na Použít.
15. Nyní jste v Prostředí uživatele. Vyberte všechny záznamy a na jeden z nich poklepejte.
16. Formulář vypadá dobře a může být přímo použit ve 4D a ne pouze pro Web.

9.9.2. Úpravy testovací metody

1. Upravte metodu M_WebShowCustomers následovně.

Používání 4D Databáze na Webu

Formuláře Web Aware

❖ ...
INPUT FORM ([Zákazníci];" Zákazníci.iw")
OUTPUT FORM ([Zákazníci];" Zákazníci.ow")
ALL RECORDS ([Zákazníci])
SET WEB DISPLAY LIMITS (5;100) `
MODIFY SELECTION ([Zákazníci];*)
...

3. Zavřete všechna okna.

9.9.3. Testování formulářů v prohlížeči Web

1. Přejděte do prohlížeče Web.
2. Klepněte opět na Ukázat zákazníky.
3. Klepněte na tlačítko v levém sloupci u libovolného záznamu.
4. Váš nový vstupní formulář by měl vypadat asi takto:

Netscape - [Data Entry: Customers.iw]

File Edit View Go Bookmarks Options Directory Window Help

Back Forward Home Reload Images Open Print Find Stop

Location: http://123.4.56.7/4Ddisprec/0/%23%23247900413.1

What's New? What's Cool? Destinations Net Search People

Zákazníci

Tento záznam čtete v 03:18:12 dne 16.2.1999 4D času.

Firma Blockbuster Video

Adresa 26736 Copley Rd

Město Nanakuli

Stát HI

PSC 96792

Jméno Celia U.

Příjmení Simensen

Telefon 7087514615

Obnovit čas

Document: Done

Používání 4D Databáze na Webu

Formuláře Web Aware

5. Vyzkoušejte tlačítka navigace záznamů. 4D za vás provádí tuto akci rychle a průhledně. Nemusíte psát žádný speciální kód na hledání, zavádění záznamů či cokoliv jiného. Zkuste si něco podobného vytvořit s jiným řešením Web databáze.
6. Zrušte tento vstupní formulář klepnutím na tlačítko Storno.
7. Klepněte na tlačítko Hotovo ve výstupním formuláři a navraťte se do nabídky Web.

9.10. Vícestránkové formuláře

Na Web můžete použít vícestránkové formuláře. Jestliže zahrnete do formuláře více stránek a tlačítka navigace stránek 4D je za vás bude ovládat automaticky. Nemusíte nic speciálního programovat. Toto je opět specialita 4Dv6, ale je to velice obtížné provést v tradičním prostředí Web databáze.

Poznámka: Tlačítka navigace stránek odešlou formulář, takže pokud je používáte provedou se metody formuláře a triggeru tabulky.

Poznámka: Stránka 0 (master stránka) se neužívá. Jestliže máte opakované prvky musíte je umístit na každou stránku zvlášť.

9.11. Styly tlačítek

4D nabízí několik vestavěných typů tlačítek, zahrnujících různá grafická tlačítka a několik běžných tlačítek. Všechna tato tlačítka jsou převáděna na Web, ale nepřevádějí se stejným způsobem. Jestliže použijete libovolné grafické tlačítko 4D je převede jako objekt Obrázek. Tento vstupní typ nepracuje se staršími prohlížeči a negeneruje událost onSubmit v zdokonalených JavaScript prohlížečích. Pro co nejlepší kompatibilitu je doporučováno, aby jste používali typ tlačítek Běžná tlačítka nebo Běžná tlačítka malá. Tyto typy tlačítek se převádějí do vstupních typů “odeslat” typ tlačítek, který je podporován ve všech druzích dostupných prohlížečů.

Používání 4D Databáze na Webu

Automatické akce a Metody formuláře

10. Automatické akce a Metody formuláře

10.1. Úvod

HTML nezahrnuje žádný vestavěný systém maker. Systém maker jako JavaScript nebo VBScript může být použit, ale většina dostupných prohlížečů, pokud nejsou nejnovější verzí, je nepodporuje. Nejběžnějším způsobem pro přidání maker do Web stránek je poslat je zpět Web serveru, kde extra program (CGI) zpracuje informace pro Web server. Výhodou tohoto přístupu je, že tento způsob pracuje pro všechny uživatele a nejenom pro ty s lepšími prohlížeči. Nevýhoda je, že uživatel musí odeslat formulář k ověření svých vstupů a to tak, že vy k tomu musíte napsat spoustu složitého kódu, aby jste tuto možnost zajistili. O JavaScript budeme hovořit později jako o způsobu přidávání maker do vašich formulářů. Nyní se podíváme na to jak 4D automaticky ovládá objekty formulářů. Některé události formulářů a objektů formulářů pracují, i když formulář je používán přes Web. Kód běží uvnitř 4D, když uživatel klepne na příslušné tlačítko nebo zavádí formulář. Nemusíte přitom psát žádný speciální kód pro ovládání tohoto rysu. Jinými slovy váš regulérní kód 4D, může ošetřovat rovněž Web klienty. Takže tento rys podstatně zjednodušuje uzpůsobení existující databáze pro Web a vytváření nových databází, které budou sloužit odlišnému druhu klientů.

10.2. Automatické akce

Předtím, než se začneme zabývat skripty, pamatujte si, že tlačítka automatických akcí 4D, která mají na Web smysl pracují, jsou to např. navigace záznamů a stránek a nevyžadují psaní speciálního kódu jestliže použijete automatické akce tlačítek ve formulářích.

10.3. Akce tlačítek, kterým by jste se měli vyhnout

4D nabízí velké množství akcí automatických tlačítek a to buď přes automatické skripty anebo tlačítka automatických akcí. Některé z těchto objektů se nechovají správně pokud jsou použity z prohlížečů Web a měli by jste se jim vyhnout.

Výstupní formuláře

Vymazat vybrané

Vytvořit diagram

Třídít

Zprávy

Tisk výběru

Vstupní formuláře

Přidat podzáznam

Upravit podzáznam

Otevřít podzáznam

Tisk záznamu

10.4. Tisky

Vyhněte se všem možnostem tisku ve 4D spouštěným přes Web. Tisk se provede na stroji 4D nebo 4D Server ošetřujícím požadavky Web a může chtít otevřít dialogové okno, které v prohlížeči Web nemůže být zobrazeno. Můžete tisknout přímo z prohlížeče Web, ale nesmíte použít metody 4D.

Používání 4D Databáze na Webu

Automatické akce a Metody formuláře

10.5. Podporované metody

V6 zahrnuje bohatou sadu událostí formulářů, která vám dovolí vytvořit podrobně řízené moderní rozhraní. Většina z těchto akcí není v HTML dostupná a proto nejsou podporovány. Následující události formuláře jsou podporovány:

4D Event	Ve 4D	Ve Web prohlížeči
Při zavedení	Formulář je poprvé zaváděn	Formulář je poprvé zaváděn
Při vyvedení	Formulář je opouštěn	Formulář je opouštěn
Při klepnutí	Tlačítko je klepnuto	Tlačítko je klepnuto

Vidíte něco podobného na těchto dvou sloupcích? Jsou stejné. Tyto události vám umožní řízení vašich internetových stránek bez nutnosti napsání nového kódu nebo učení se JavaScript. Vyzkoušejme to

10.5.1. Testování událostí formuláře

1. Jděte do Prostředí návrháře ve 4D.
2. Otevřete metodu formuláře Zákazníci.iw.
3. Vytvořte metodu typu Textová
4. Přidejte následující kód:

```
If (False)
  ` Form Method: [Zákazníci];" Zákazníci.iw"
  ` ACI University Programming Classes
  ` Created By: David Adams
  ` Date: 3/21/97

  ` Purpose: Představuje užití události formuláře WEB aware

  ⋄f_Version6x40 := True
  ⋄fD_Adams := True
End if

Case of
: (Form event=On Load)
  ALERT("Při zavedení Form Event")

: (Form event=On Clicked)
  ALERT("Při klepnutí Form Event")

: (Form event=On Unload)
  ALERT("Při vyvedení Form Event")

End case
`End of mehtod
```

5. Zavřete všechna okna.
6. Přejděte do prohlížeče Web.

Používání 4D Databáze na Webu

Automatické akce a Metody formuláře

7. Vyberte odkaz Ukázat zákazníky.
8. Upravte libovolný záznam.
9. Předtím než se záznam zavede je převedeno upozornění a posláno vašemu prohlížeči. Poté co toto upozornění potvrdíte zobrazí se formulář.
10. Stiskněte tlačítko Storno.
11. Nyní uvidíte upozornění při události Při klepnutí.
12. Stiskněte tlačítko OK.
13. Nyní uvidíte upozornění při události Při vyvedení.
14. Stiskněte tlačítko OK a vraťte se do seznamu.
15. Stiskněte tlačítko Hotovo a vraťte se to do své Web nabídky.

Tyto jednotlivé metody formuláře jsou samy o sobě zbytečné, ale ilustrují že události formuláře 4D pracují i přes Web. Nyní s nimi uděláme již něco užitečného.

10.5.2. Použití události Při zavedení k úpravě zobrazovaných položek.

1. Jděte do Prostředí návrháře ve 4D.
2. Otevřete metodu formuláře Zákazníci.iw
3. Odstraňte existující kód.
4. Přidejte následující kód:

```
If (False)
  ` Form Method: [Zákazníci];" Zákazníci.iw"
  ` ACI University Programming Classes
  ` Created By: David Adams
  ` Date: 3/21/97

  ` Purpose: Zavádí proměnné pro zobrazení na Web.

  ⋄f_Version6x40 := True
  ⋄fD_Adams := True
End if

Case of
  : (Form event = On Load)
    sWelcome:="Čtete tento záznam v hod. "+String(Current time)+" dne:
    "+String(Current date)+" 4D času."

End case
`konec metody
```

5. Otevřete formulář Zákazníci.iw
6. Vytvořte nedostupnou oblast (proměnnou) nazvanou sWelcome. Tato proměnná zobrazí hodnotu vytvořenou pomocí metody formuláře.
7. Zavřete nebo uložte všechna okna.
8. Přejděte do prohlížeče Web.
9. Klepněte na odkaz Ukázat zákazníky.

Používání 4D Databáze na Webu

Automatické akce a Metody formuláře

10. Upravte libovolný záznam.
11. Všimněte si, že se vaše uvítací hlášení objeví spolu s datumem a časem.
12. Použijte navigační tlačítka záznamů přesunu na jiný záznam.
13. Všimněte si že se čas v uvítací zprávě změnil.
14. Uzavřete záznam.
13. Uzavřete seznam záznamů a vraťte se do vaší Web nabídky.

4D automaticky provede váš kód přesně tak, jako by byl ve 4D a zobrazí výslednou hodnotu. Není to úžasné! Ale to ještě není vše! Nejenom, že máte přístupné události formuláře, ale máte přístupné i metody událostí objektů formuláře bez zvláštního usilí. Metody vašich tlačítek respektují tytéž události jako samotné formuláře. Událost Při klepnutí pracuje pouze pro skutečná tlačítka, ale události Při zavedení a Při vyvedení pracují rovněž s dalšími objekty zahrnujícími pole, posuvné oblasti, nabídky a přepínače. Zkusme tlačítko:

10.5.3. Odezva na událost Při klepnutí.

1. Jděte do Prostředí návrháře 4D.
2. Otevřete formulář Zákazníci.iw.
3. Přidejte nové tlačítko nazvané bMyButton a nazvěte jej Aktualizovat čas.
4. Přidejte následující kód k metodě tlačítka:

```
If (False)
  ` Object Method: bMy Button [Zákazníci];" Zákazníci.iw"
  ` ACI University Programming Classes
  ` Created By: David Adams
  ` Date: 3/21/97

  ` Zavádí proměnné pro zobrazení na Web.

  ⋄f_Version6x40 := True
  ⋄fD_Adams := True
End if

Case of
  : (Form event = On Clicked)
    sWelcome:=" Čtete tento záznam v hod. "+String(Current time)+" dne:
    "+String(Current date)+" 4D času"

End case
`End of method
```

5. Zavřete nebo uložte všechna okna.
6. Jděte do Web prohlížeče.
7. Klepněte na odkaz Ukázat zákazníky.
8. Upravte libovolný záznam.
9. Stiskněte tlačítko Aktualizovat čas.
10. Vaše uvítací hlášení se změní.
14. Stiskněte OK a vraťte se do Web seznamu.

Používání 4D Databáze na Webu

Automatické akce a Metody formuláře

15. Stiskněte tlačítko Hotovo a vraťte se do Web nabídky.

10.6. Důležité detaily o metodách objektů a automatických tlačítkách akcí

Když používáte metodu tlačítka nebo tlačítko automatické akce formulář je odeslán. Váš 4D kód se provádí uvnitř 4D a ne uvnitř vašeho prohlížeče Web to znamená, že jediný způsob jak 4D umí odpovědět na události tlačítka je tehdy, když je formulář odeslán. To rovněž znamená, že změny jsou uloženy a trigger tabulek se provedou jestliže samozřejmě existují. Na toto je potřeba při používání dávat pozor např. u vícestránkového formuláře se provedou metody formulářů a trigger tabulky pokaždé, když se přepnete ze stránky na stránku, protože z pohledu databázového stroje 4D byl formulář odeslán. Jestliže si přejete napsat skript objektu provádějící nějakou akci a neodeslat přitom formulář pak potřebujete použít JavaScript nebo nějaký jiný systém kódu na straně klienta.

Používání 4D Databáze na Webu

V6: Jedinečný Web Server

11. V6: Jedinečný Web Server

11.1. Úvod

Jestliže jste již někdy realizovali projekt integrující Web a databázi, máte samozřejmě mnoho otázek jak 4D pracuje dosahuje-li takových výsledků. Jestliže je to vaše první setkání s integrací Web databáze tak vás asi rozmátlíme. 4D přistupuje na Web odlišně v porovnání s jinými prostředími Web databáze a to již v samotném principu. Jak jste slyšeli dříve je pravděpodobně lepší přemýšlet o tom následujícím způsobem: 4D používá HTTP jako síťový komponent, aby udělala ze standartního Web prohlížeče plného databázového klienta. Nyní se chystáme vyzkoušet zda normální prostředí Web databáze jsou konstruovány tak, že mohou dosáhnout to co dělá 4D a kdy můžete potřebovat přístup bližší tradičnímu.

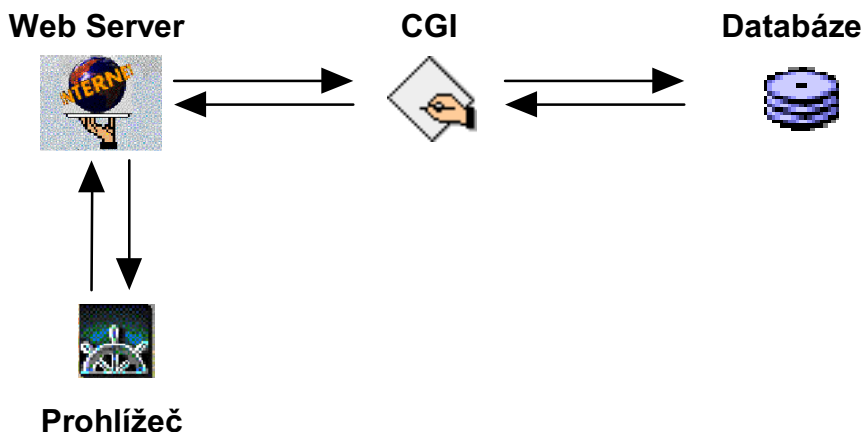
11.2. Co je Web server

Tradiční Web server je jen něco více než server souborů. Prohlížeč požádá o stránku nebo obrázek a Web server mu ji pošle zpět. To je téměř vše co Web server dělá. Normálně neprovádí výpočty a nepamatuje si uživatele od jednoho požadavku ke druhému. Web servery byly navrženy tak, aby byly co nejjednodušší. Jejich jednoduchost je důvodem jejich širokého a rychlého rozšíření na různých platformách (např. existují dva odlišné Web servery pro sálové počítače IBM 3270). Tradiční Web server se spoléhá na další aplikace při provádění zákaznických požadavků a zpracování jako např. u dotazů.

11.3. CGI

11.3.1. Tradiční CGI

The Common Gateway Interface (CGI) je na platformě nezávislý standart definující jak jsou data zasilána externímu programu pro zpracování a jak se výsledná data navracují od externího programu k Web serveru. Externí program je nazýván CGI. Standart neříká nic o tom co nebo jak programy CGI provádějí, říká pouze to jaké informace jsou vyměňovány mezi ním a Web serverem. CGI může být napsáno v libovolném prostředí schopném komunikovat s Web serverem: Perl, C, AppleScript, 4D, MicroFOCUS, Java, SmallTalk, zkrátka cokoliv. Zde vidíme diagram jak klient, server, CGI a databáze interagují:



V tomto tradiční modelu Web server nezná nic o databázi, CGI nekomunikuje s prohlížečem a databáze nezná nic o Webu. V tomto modelu, když uživatel chce přístup k datům databáze, děje se následující:

1. Prohlížeč odešle požadavek na Web server.

Používání 4D Databáze na Webu

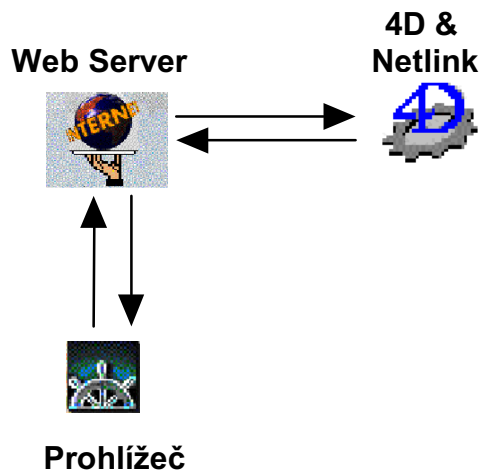
V6: Jedinečný Web Server

2. Web server vidí, že obdržel požadavek na informace z jiného programu.
3. Web server nemá ponětí o samotném požadavku, ale ví kam jej odeslat, takže jej odešle CGI.
4. CGI obdrží požadavek a přečte jej. Předpokládejme, že potřebuje hovořit s databází.
5. CGI se zeptá databáze na informaci.
6. Databáze nezná nic o HTML. Webu nebo čemkoliv jiném, ale jestliže je CGI napsáno správně databáze předá data.
7. CGI zformátuje data a odešle je zpět na Web server.
8. Web server obdrží data, neví vůbec co to je, ale ví, že je má poslat klientovi.
9. Klient obdrží data.

Jak můžete vidět 4D v6 je poněkud jednodušší. Tradiční model se skládá z několika kusů nezávislého software, který ví jak komunikovat. Komunikace probíhá pomocí AppleEvents na Mac OS, DDE na Windows, a stdin a stdout na UNIX. Každé z těchto prostředí implementuje CGI standart odlišně (proměnné prostředí jsou používány na UNIX, záznamy AppleEvents jsou používány na Mac OS atd.) ale celková architektura je tatáž.

11.3.2. Zdokonalené CGI: Netlink/4D

Ve světě 4D vylepšení tohoto modelu je v činnosti již po delší dobu. Použitím Netlink/4D můžete sloučit aspekty CGI a databáze do jednoho 4D programu a to jak na Mac OS tak Windows.



Je to skutečně velké vylepšení oproti většině prostředí CGI, protože úkoly CGI jsou součástí databáze a hlavně mají plný přístup ke kompletnímu 4D jazyku.

11.4. Stav který neudrží informace

Tato část o CGI směřovala k této kapitole. Web servery jsou technicky specifikované jako neudržující stav. To je skvělé, ale co to znamená? Tradiční Web server neudrží jakékoliv stavové informace o klientu mezi jednotlivými spojeními. Když se spojíte s Web serverem a požádáte o stránku, Web server vyhledá stránku, odešle ji a kompletně na vás zapomene. Jestliže požádáte o jinou stránku o sekundu později Web server o vás nic neví, nepamatuje si, že jste právě před chvílí požádali o jinou stránku. Samozřejmě mnoho důležitých implementací požaduje informace o stavu např. aplikace nákupních center, personalizované stránky atd. K zavedení takových systémů programátoři obvykle drží stavové informace na straně serveru

Používání 4D Databáze na Webu

V6: Jedinečný Web Server

v aplikacích CGI nebo databázích. To je rozumná strategie, ale je to spousta práce jestliže máte vytvořit nějaký komplexní systém. Jiná strategie je udržovat informace o stavu na straně klienta a odesílat je serveru při každém připojení. Tato strategie se nazývá použití cookie.

11.5. Cookies

Asi už jste o cookies slyšeli. Jsou tak známé díky vnímanému bezpečnostním riziku, které představují. Cookie by mohli být nazýváno bezpečnostní kartičkou nebo záznamem. Cookie má název, čas vypršení, hodnotu a doménu, které náleží. Když se spojíte se serverem, který vytváří cookie, server pošle informaci vašemu prohlížeči a prohlížeč napíše tuto informaci na váš disk. Příště, když se připojíte k serveru a to během jedné sekundy, minuty nebo o měsíc později – cookie je posláno zpět na server, pokud mu ovšem nevypršela platnost. Toto je rozumný způsob jak udržovat stavovou informaci. Mnoho nákupních serverů používá cookies, aby si udrželi přehled vámi vybíraného zboží do té doby než jej potvrdíte. Tiskové agentury na svých internetových stránkách používají cookies aby si udrželi přehled o článcích, které jste již viděli. Na uživateli záleží jestliž umožní cookies ve svém prohlížeči.

Cookies jsou dobrým řešením pro CGI, které si sami nic moc nepamatují. Jako pro programátory 4D je pro vás snadné vytvořit nástroj pro řízení se strukturovanými daty, který to zajistí. Pro mnoho programátorů není schopnost ukládat a vyhledávat záznamy standardní součástí prostředí. Užívání cookies je pokusem vyřešit situaci těchto programátorů. 4D v6 nepoužívá cookies, protože je nepotřebuje: sleduje všechno sama, což je samozřejmě mnohem elegantnější řešení.

Samozřejmě cookies se 4D můžete použít, ale nepotřebujete je. Pokud používáte Netlink/4D můžete přidat cookies do dat a odeslat Web klientovi. Rovněž můžete použít vlastní JavaScript k zapsání cookies na straně klienta. Můžete číst příchozí cookie z hlavičky požadavku HTTP s pomocí Netlink/4D a nemo v metodě databáze při Web spojení. Nicméně, protože cookies nepotřebujete nebudeme si je zde demonstrovat. Pokud se o ně zajímáte jsou popsány v RFC 2109 HTTP State Management Mechanism, který nahradil původní návrh od Netscape.

11.6. V6 Řízení stavu

Nyní když víme jak 4Dv6 nepracuje, řekněme si jak skutečně pracuje. 4Dv6 je vždy server udržující stav. Používá HTTP k vytvoření sekce mnohem podobnější tradičnímu databázovému připojení. Toto je možné i v jiných nástrojích po určitém programování, ale 4D zde nekončí. 4D rovněž udržuje obsah jak jsme se již zmínili dříve. Místo, aby jste sledovali cestu uživatele skrz databázi, 4D ji sleduje za vás, tak jako kdyby byl regulérním 4D Klientem. Jestliže např. váš uživatel prohlíží záznamy pomocí MODIFY SELECTION a pak upravuje určitý záznam, 4D to všechno sleduje za vás. Nepotřebujete provádět vůbec nic speciálního k udržování výběru záznamů, uzamykání záznamů, udržování informací o úrovni přístupu atd. Jestliže vytváříte databázi nebo transakčně orientované systémy, je kombinace úplného udržování stavu a automatického udržování stavu jedinečná a nadmíru užitečná.

11.7. Kdy použít tradiční Web Server

Jak jsme již řekli, 4D vždy udržuje stav. To znamená, jestliže chcete Web server, který odesílá tisíce statických stránek za hodinu nebude v6 tím správným výběrem. To není kritika 4D v6: pro tento účel prostě nebyla navržena. Jestliže máte potřebu tradičního Web serveru, použijte nějaký a ponechte si 4D pro úlohy pro které se nejlépe hodí. To úžasné na Web systémech je to, že mohou být rozprostřeny přes více strojů kolem celého světa. Můžete mít stroj pouze pro grafiku, jiný pouze pro statické texty a třetí pro databázové úlohy. Takovéto uspořádání nabízí vysokou výkonnost (dokonce bez UNIX strojů) a plně využívá síly každého prvku tohoto systému.

Dost teorie a do práce!

Používání 4D Databáze na Webu

Web připojení podrobně

12. Web Připojení podrobně

12.1. Úvod

Co se stane, když se Web uživatel připojí k vaší databázi? Předpokládejme, že máte alespoň jednu uživatelskou nabídku a Web server je spuštěn. Provede se:

12.1.1. Připojení Web popsané po krocích.

1. Proces Web Serveru zachytí požadavek.
2. Jestliže je databáze chráněná heslem, je odesláno okno hesel. Jestliže je vráceno správné jméno uživatele a heslo, jdeme dál ke kroku 3, jinak se opakuje tento bod.
3. Proces Web serveru zkontroluje, zda nebyla překročena licence připojení.
4. Jestliže bylo povoleno připojení založí Web server dočasný lokální proces pro inicializaci připojení s prohlížečem. Provede se okamžitě a je mimo vaši kontrolu.
5. Nastartuje se nový proces pro tuto sekci Web. Procesu je přiřazen název asi takto: Web Connection# 1764055203. Číselný údaj tohoto názvu je ID obsahu pro sekci, jak jsme se zmínili výše.
6. Jestliže databázová metoda Při Web spojení má kód, je provedena.
7. Po tom, co všechny routiny z metody Při Web spojení jsou dokončeny je převedeno první záhlaví nabídek a stránka odeslána prohlížeči. (Pokud není v prováděném kódu stanoveno jinak)

Podívejme se na některé z těchto kroků podrobněji a prohlédneme si je v akci. Provedme následující:

12.1.2.. Vyzkoušení podrobností Web spojení

1. Přejděte do Prostředí návrháře.
2. Upravte metodu Při Web spojení následovně:

```
If (False)
  ` Database Method: On Web Connection
  ` ACI University Programming Classes
  ` Created By: David Adams
  ` Date: 3/21/97

  ` Purpose: Demonstruje skupiny array na Web.
```

```
  ⋄f_Version6x40 := True
  ⋄fD_Adams := True
```

```
End if
```



```
TRACE
```

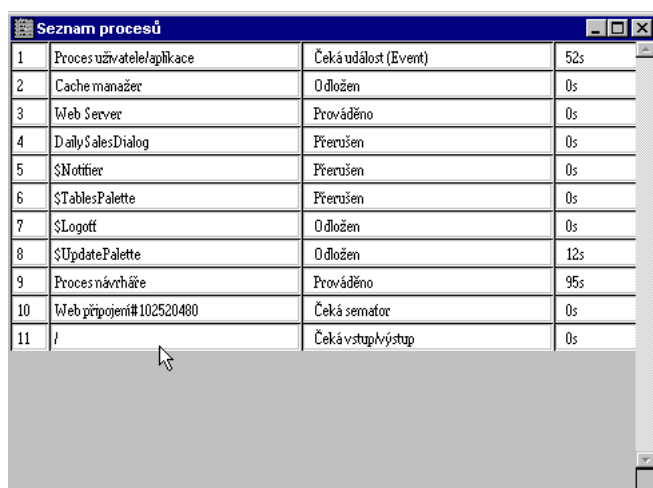
```
ALERT ("Chystám se odeslat záhlaví Web cvičení.")
```

```
MENU BAR(5)
  `Konec metody
```

Používání 4D Databáze na Webu


Web připojení podrobně

3. Uložte a uzavřete metodu.
4. Vyberte Nástroje → Seznam procesů.
5. Jestliže běží nějaký Web proces odstraňte jej. (Nejste schopni odstranit proces Web serveru a ani to nepotřebujete.).
6. Umístěte si okno Seznamu procesů, tak aby jste jej viděli při přechodu do prohlížeče Web.
7. Přejděte do vašeho prohlížeče a změňte velikost okna tak, že okno 4D seznamu procesů bude viditelné.
8. Napište adresu vašeho stroje a klepněte enter.
9. Objeví se krokovací okno 4D, a nové dva procesy se objeví v seznamu procesů:



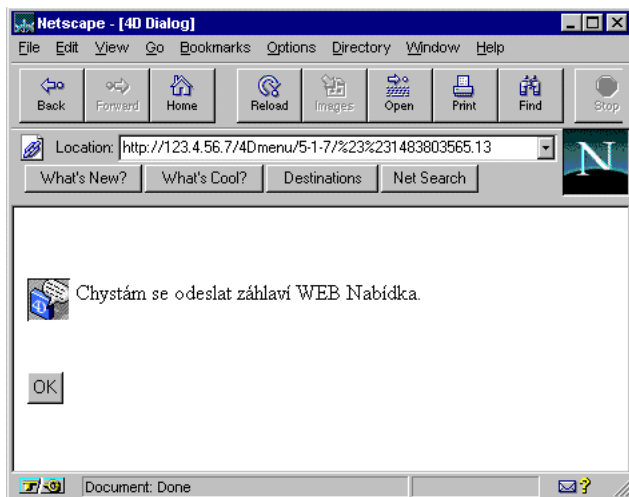
	Proces uživatele/aplikace	Čeká událost (Event)	52s
2	Cache manažer	Odložen	0s
3	Web Server	Prováděno	0s
4	DailySalesDialog	Přerušeno	0s
5	\$Notifier	Přerušeno	0s
6	\$TablesPalette	Přerušeno	0s
7	\$Logoff	Odložen	0s
8	\$UpdatePalette	Odložen	12s
9	Proces návrhář	Prováděno	95s
10	Web připojení#102520480	Čeká semafor	0s
11	/	Čeká vstup/výstup	0s


Proces nazvaný “/” je použit k inicializaci nového Web připojení, číslovaný zde 102520480 (Vaše číslo bude odlišné).

10. Přejděte do okna debuggeru a klepněte na tlačítko  “Krok”, takže se provede řádek kódu s ALERT.
11. Inicializační proces “/” se odstraní. Je nyní hotov.
12. Navraťte se do Web prohlížeče.
13. Nyní vidíte vaše upozornění převedené do HTML.

Používání 4D Databáze na Webu

Web připojení podrobně



14. Klepněte na tlačítko OK v upozornění v prohlížeči.
15. 4D krokovací okno se opět objeví.
16. Klepněte na tlačítko  “Nekrokovat” vlevo okna.
17. Přejděte do prohlížeče Web a podívejte se, že byla odeslána nabídka.

Toto cvičení ukazuje jak 4D řídí příchozí Web připojení, ale není to to nejdůležitější co chceme zdůraznit. Klíčové zjištění, které jsme učinili je, že metoda Při Web spojení se zastaví a čeká dokud dialogy upozornění a formuláře nejsou přesně vyřízeny tak jako kdyby jste pracovali uvnitř samotné 4D. Když provádíte ve 4D rutinu, která otevírá okno, volající metoda se zastaví dokud není okno uzavřeno a pak teprve pokračuje dále. TRACE, DIALOG, ALERT, CONFIRM, MODIFY SELECTION, ADD RECORD and DISPLAY SELECTION jsou běžné příkazy 4D, které se chovají tímto způsobem. Volající metoda čeká dokud okno není vyčištěno, předtím než pokračuje dalším řádkem kódu. V metodě Při Web spojení (nebo v libovolné jiné metodě používané při Web spojení) je uskutečněn tento normální běh řízení 4D, i když klientem je prohlížeč Web. Toto je zásadní část myšlenky, kdy jsme říkali že 4D udržuje obsah a stav. V běžném prostředí Web, když je poslána stránka prohlížeči je volající metoda ukončena. Web systém může být programován k přijetí následných událostí, ale neznám žádné další prostředí, které skutečně čeká na další logickou událost.

12.2. Spuštění procesu

Spouštění procesů Web připojení je plně automatizováno. Jádro procesu Web serveru ošetří kontrolu licence, ziniculuje nový proces a ustanoví obsah. Je to úplný proces 4D s přístupem ke svým vlastním výběrům záznamů, proměnným, sadám a pojmenovaným výběrům. Proces zůstává aktivním dokud nevyprší jeho čas nebo není vypnuta databáze.

12.3. Při Web spojení

Po té, kdy je nastartován proces Web připojení, spustí se metoda Při Web spojení, jestliže je nalezena. V této rutině můžete udělat buď mnoho a nebo málo podle toho jak chcete. Nemusíte ji vůbec použít nebo ji můžete použít k řízení každého aspektu sekce Web klienta. Podíváme se na tuto metodu podrobněji v další kapitole.

Používání 4D Databáze na Webu

Web připojení podrobně

12.4. Záhloví 1

Ať je v metodě Při Web spojení nalezen jakýkoliv kód (pokud vůbec je) a je proveden po jejím skončení je převedeno Záhloví 1 a odesláno prohlížeči Web. V našem případě metoda Při Web spojení posílá Záhloví5.

12.5. ID obsahu a podobsahu

ID obsahu odpovídá procesu Web připojení přiřazenému této sekci Web. Již dříve jsme se dívali na některá HTML, která zahrnovala ID obsahu a podobsahu. ID obsahu je zahrnuto do každé schránky a posláno zpět s každým požadavkem, takže 4D může okamžitě určit ze které sekce Web požadavek přišel. Mapování sekci tímto způsobem je v tradičních prostředích CGI poměrně standartní technika. ID obsahu je jednoduchý a neomylný způsob 4D sledování, o co který klient žádá.

Ale co ID podobsahu? ID podobsahu je celé číslo přidané k ID obsahu 865656880.0 znamená "obsah 865656880" s podobsahem ".0".

Tato inovace 4D je způsob jak 4D synchronizuje akce uživatele v prohlížeči s procesem Web připojení v databázi. Udělejte následující a všimněte si toho jak se ID podobsahu objevuje v poli adresy vašeho Web prohlížeče.

1. Jděte do Prostředí návrháře 4D.
2. Otevřete databázovou metodu Při Web spojení.
3. Upravte tuto metodu, tak, aby již neobsahovala příkazy TRACE nebo ALERT. Metoda by měla vypadat následujícím způsobem:

```
If (False)
  ` Database Method: On Web Connection
  ` ACI University Programming Classes
  ` Created By: David Adams
  ` Date: 3/21/97

  ` Purpose: Demonstruje skupinová pole na Web.

  ⋄f_Version6x40 := True
  ⋄fD_Adams := True

End if

MENU BAR(5)
  `End of method
```

4. Uložte a zavřete tuto metodu.
5. Otevřete a nastavte Okno seznamu procesů tak, že jej uvidíte ze svého Web prohlížeče.
6. Přejděte do svého Web prohlížeče.
7. Upravte a nastavte váš prohlížeč tak, aby jste z něho viděli Okno seznamu procesů.
8. Připojte se k databázi.
9. Měli by jste vidět svou Web nabídku.
10. Všimněte si, ID obsahu ve vašem novém procesu Web připojení v Okně seznamu procesů ve 4D.

Používání 4D Databáze na Webu

Web připojení podrobně

11. Klepněte na odkaz Ukázat zákazníky a všimněte si pole adresy v prohlížeči. Nyní obsahuje URL které končí ID obsahu následované "0".
12. Upravte jeden ze záznamů a všimněte si, že ID podobsahu se nyní změnilo na .1
13. Vraťte se do výstupního seznamu a všimněte si, že se ID podobsahu změnilo na .2

Pokaždé, když klepnete na odkaz ID podobsahu se změní. Zde vidíme příklady typů URL, které by jste měli vidět:

Ukázat Zákazníky [http://192.9.200.181/4Ddataentry/1/20/\\$1233021104.0](http://192.9.200.181/4Ddataentry/1/20/$1233021104.0)

Upravit Zákazníky [http://192.9.200.181/4Dmenu/5-1-4/\\$1233021104.1](http://192.9.200.181/4Dmenu/5-1-4/$1233021104.1)

Storno [http://192.9.200.181/4Ddisprec/0/\\$1233021104.2](http://192.9.200.181/4Ddisprec/0/$1233021104.2)


ID obsahu zůstává stejné, protože všechny vaše akce jsou součástí jednoho a téhož připojení Web nebo-li obsahu, mění se pouze ID podobsahu. Proč 4D přidává ID podobsahu? Pamatujeme si, že 4D čeká než klient Web dokončí práci na dané stránce, předtím než pokračuje dál. Jestliže upravujete záznam uvnitř MODIFY SELECTION 4D čeká než dokončíte práci se záznamem jinak nemůže uzavřít MODIFY SELECTION a pokračovat dále ve vašem kódu. V prostředí 4D to není problém: 4D vás nenechá odskočit z příkazu jako je MODIFY SELECTION bez uzavření dialogu. Web je však zcela odlišné prostředí. Ve skutečnosti norma je podstatně bližší náhodnému přeskokování ze stránky na stránku než strukturovaným úlohám prováděným na serveru v definovaném pořadí. Jinými slovy normální navigace web přes přeskoky kamkoliv nemá žádný smysl pokud používáte prohlížeč jako databázového klienta. Podívejme se na důsledky takovéhoho použití.

12.6. Neklepat na tlačítko Zpět!

Co se stane jestliže uživatel ve svém prohlížeči klepne na tlačítko Zpět nebo přejde někam jinam? Pojďme si to vyzkoušet:

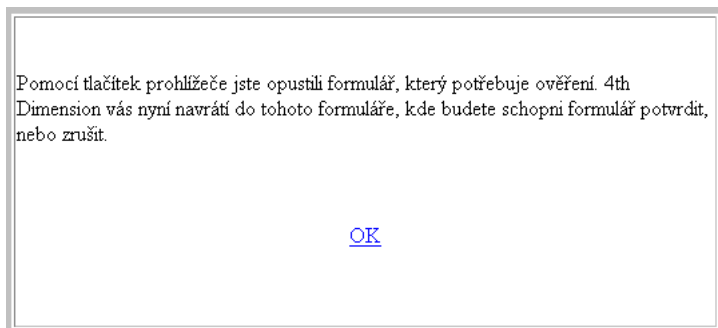
1. Otevřete svůj prohlížeč.
2. Připojte se k databázi.
3. Měli by jste vidět svou nabídku Web.
4. Klepněte na odkaz Ukázat zákazníky ve vašem prohlížeči Web.
5. Upravte jeden ze záznamů.



6. Stiskněte tlačítko  Zpět. Nyní by jste se měli vrátit do seznamu zákazníků.
7. Klepněte na odkaz pro jiného zákazníka.
8. Nyní se vám zobrazí následující chybové hlášení:

Používání 4D Databáze na Webu

Web připojení podrobně



9. Stiskněte OK a nechte 4D ať se pokusí vrátit do správné obrazovky.

Co se stalo? Tlačítko Zpět je normálně zcela bezpečné, ale zde působí problémy. Toto je součást problematiky použití Web prohlížeče jako databázového klienta, která se výrazně liší od klasického prohlížení Web. Tento problém může vadit těm uživatelům, kteří jsou zvyklí používat Web prohlížeče ale nejsou seznámeni s problematikou Web prohlížeče jako databázového klienta. Pojdme si podrobně vyzkoušet co se vlastně stane, své zkušenosti pak budete samozřejmě předávat svým uživatelům. Podívejme se na to jak se jednotlivé kroky, které jsme podnikli na Web projevují ve 4D:

Na Web

Odkaz Ukázat zákazníky

Klepnout na jeden záznam

Dvakrát klepnout na tlačítko Zpět

Opět odkaz Ukázat zákazníky

Ve 4D

Položka nabídky Ukázat zákazníky

provede se MODIFY SELECTION

Úprava jednoho záznamu uvnitř výběru z MODIFY SELECTION

Žádný efekt

Vyvolání druhé MODIFY SELECTION z vstupního formátu záznamu, který upravujete.

Samozřejmě vám nic a nikdo nemůže zabránit v tom, aby jste si do svého vstupního formuláře dali skript, který po odeslání informací o záznamu uzavře MODIFY SELECTION. Ale nemá to smysl. 4D monitoruje tento druh akcí Web klienta jako synchronizační chybu a to vždy, když uživatel nedokončí započatou úlohu.

Když používáte databázi, provádíte úkol, který je zařazen někde v nějaké posloupné řadě úkolů. V dobře navrženém rozhraní máte hodně volnosti, ale struktura je zde stále. Web nemá pod sebou žádnou strukturu. Na dokumenty se odkazujete takovým způsobem jak kdyby byly uspořádány bez jakéhokoli řádu. Protokoly zacházejí s každým dokumentem jako s objektem bez jakýchkoli vztahů s dalšími dokumenty. Je pochopitelné, že systém, který je většinou bez jakéhokoli řádu nemusí mít rozhraní nebo strukturu navigace odpovídající dobře strukturovanému systému. Základní pravidla, které je potřeba si pamatovat zde uvádím:

1. Otevřené úlohy musí být uzavřeny..

Používání 4D Databáze na Webu

Web připojení podrobně

2. Uživatelům se ze začátku může tento systém zdát zmatečný, protože Web je normálně zcela nestrukturovaný.
3. Správné použití systému 4D obsahu přináší značné výhody.
4. Když se uživatel seznámí s tímto systémem je použití velice jednoduché.

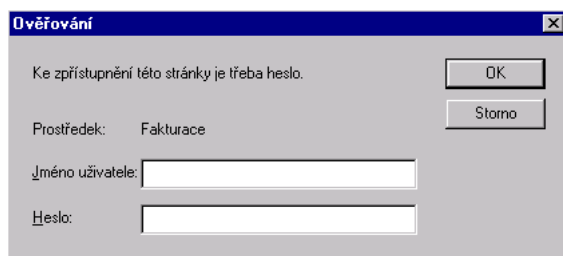
Používání 4D Databáze na Webu

Bezpečnost a uživatelský přístup

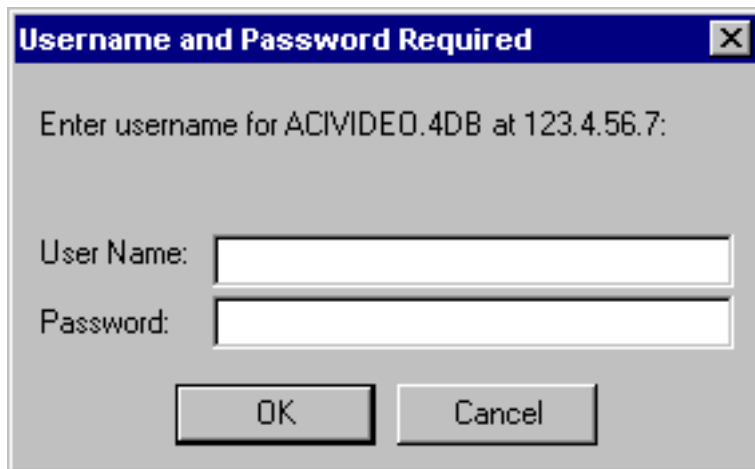
13. Bezpečnost a uživatelský přístup

13.1. Hesla

4D zahrnuje integrovaný systém bezpečnosti a uživatelského přístupu. Tento systém je rozšířen i na Web klienty. Když se Web klient připojí Web prohlížečem k databázi chráněné heslem, 4D odpoví bezpečnostním hlášením HTTP. Toto není formulář 4D, je to speciální HTTP hlavička, která přikazuje prohlížeči, aby požadoval uživatelské jméno a heslo. Záleží na prohlížeči jakým způsobem bude tuto informaci požadovat a jaké informace zobrazí o serveru. Zde můžeme vidět dva dialogy bezpečnostního hlášení ze dvou nejběžnější používaných prohlížečů, běžících na počítači PC a připojených ke stejné databázi:



Internet Explorer



Netscape Navigator

Je zde několik dobrých důvodů proč 4D používá bezpečnostní hlášení prohlížečů raději než své vlastní formuláře:

- ❖ Navracený text je posílán v zakódované formě..
- ❖ Bezpečnostní dialog se objeví v jazyce, který uživatel používá, protože je součástí jeho prohlížeče.
- ❖ Toto rozhraní je známé Web uživatelům, kteří používají chráněné Web stránky.

Můžete si vytvořit svůj vlastní systém hesel založený na 4D formuláři, ale pamatujte si, že informace budou vráceny jako nekódovaný text.

Používání 4D Databáze na Webu

Bezpečnost a uživatelský přístup

13.2. Přístup uživatele a skupin

Systém přístupových práv pro uživatele a skupiny pro objekty 4D zahrnující formuláře, tabulky a nabídky platí i pro Web klienty. Nemusíte dělat žádné nové programování.

13.3. Uživatelské účty

V době, kdy již běží Web připojení byly již splněny všechny požadavky na bezpečné připojení. To znamená, že funkce 4D Current user správně vrátí jméno uživatele příslušného k této sekci Web. Tuto informaci můžete použít k uzpůsobení uživatelského rozhraní, k sledování aktivity a k monitorování uživatelů bez ohledu na předchozí heslo.

13.4. Uživatelské stránky

Stránky uzpůsobené pro jednotlivé uživatele jsou výborný přístup k uživatelskému rozhraní a lidé jim věnují větší pozornost. Poměrně malé úsilí vyžaduje zahrnout do stránky uživatelské jméno nebo odesílat se stránkou novinky, které tento uživatel sleduje (většině lidí se to bude zdát zajímavější, než údaje čítače připojení, který jim řekne jak málo lidí se zatím připojilo na vaše stránky). Stránky uzpůsobené podle uživatele jsou jednou z mnoha věcí, které můžete řešit v metodě Při Web spojení, tématu následující kapitoly.

Používání 4D Databáze na Webu

Při Web spojení

14. Při Web spojení

14.1. Úvod

Nyní jste již zkušený v tom, jak 4D používá formuláře a metody přes Web. 4D automaticky převádí formuláře, spouští metody, udržuje výběry, zamyká záznamy a sleduje obsah. Všeho tohoto lze dosáhnout s minimem programování. V tom co jste se již dozvěděli je 4D jedinečná, žádné další prostředí Web a databáze vám nedovolí takto jednoduše použít prohlížeč jako databázového klienta. Všechno co jsme udělali do této doby je závislé na vestavěných převodních rysech 4D. Můžete však dosáhnout mnohem větší přímou kontrolu svých Web spojení pro určité specifické účely.

14.2. Přicházející informace

Již jste viděli, že při vytváření Web připojení se jako první provede metoda Při Web spojení. To co jsme ještě nezmiňovali je, že tato metoda má dva automatické parametry \$1 a \$2. \$1 obsahuje část URL za adresou stroje a \$2 obsahuje kompletní hlavičku požadavku HTTP od klienta. Tuto informaci můžete použít např. k napsání rutin, které:

- ❖ Volají 4D metody na základě přímého požadavku obsaženého v URL
- ❖ Provádějí speciální akce založené na URL
- ❖ Určují typ prohlížeče uživatele.

14.3. Vytváření ozvěny informací

14.3.1. Přehled

Informační parametry \$1 a \$2 jsou prosté textové bloky. 4D tyto textové bloky nerozebírá. Ale protože jste se zúčastnili tohoto kursu, odnesete si odsud rutiny, které můžete použít v libovolné databázi pro dělení textu. Nejdříve se podívejme co jsou tyto parametry zač. Máme na to výborný nástroj: prohlížeč Web. Zde je co se chystáme udělat:

1. Vytvoříte nový formulář k odeslání příchozí informace zpět k prohlížeči Web.
2. Změníte metodu Při Web spojení tak, aby odesílala zpět přicházející data.
3. Odešleme data.

14.3.2. Vytvoření formuláře

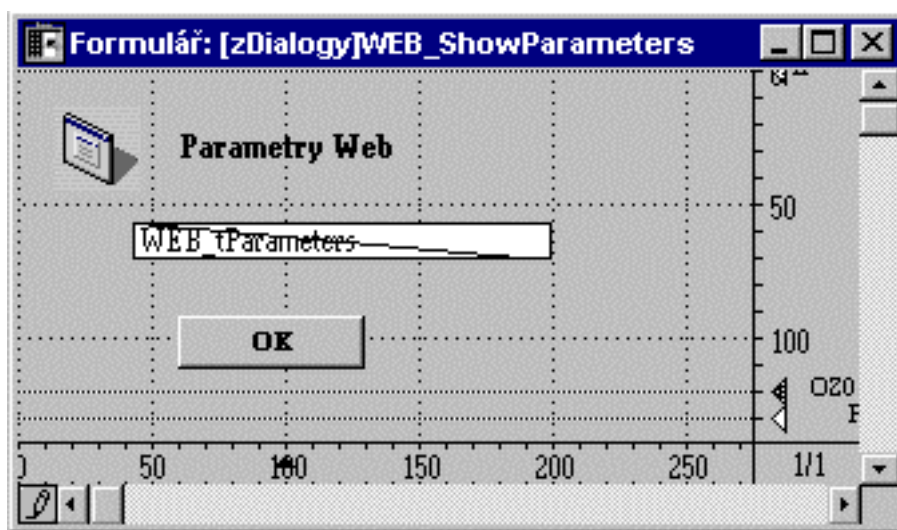
Můžete vytvořit a odeslat vlastní HTML s použitím speciálních rysů v6 nazývaných vnoření HTML. Tento rys si můžete představit jako složenou HTML. Vytvoříme formulář, který použijeme k vytvoření ozvěny informací zpět na prohlížeč.

1. Přejděte do Prostředí návrháře 4D.
2. Vyberte Návrh → Nový formulář...
3. Vyberte tabulku pro formulář zDialogy
4. Pojmenujte formulář WEB_ShowParameters.
5. Vyberte typ formuláře Web Aware
6. Přidejte do formuláře pole Povinné.

Používání 4D Databáze na Webu

Při Web spojení

7. Klepněte na tlačítko Rozšířit.
8. Klepněte na stránku Tlačítka
9. Vyberte typ tlačítek: Běžná tlačítka - malá.
10. Odstraňte všechna tlačítka z formuláře kromě tlačítka OK.
11. Klepněte na tlačítko OK.
12. Zvolte možnost Ano a vytvořte nový vzor.
13. Pojmenujte jej ACI Video Web Dialog.
14. Klepněte na tlačítko Upravit.
15. Vyberte a odstraňte pole Povinné a jeho popisku.
16. Změňte statický text na vrchu formulář z textu "zDialogy" na "Parametry Web".
15. Vložte do formuláře nedostupnou proměnnou a nazvěte ji WEB_tParameters. Není důležité jakou bude mít velikost, ale umístěte ji nad tlačítko OK.
16. Uložte formulář.
17. Formulář by měl vypadat takto:



Formulář je hotov. Je to příklad jednoduchého dialogu, který zobrazí, jakoukoli HTML, kterou umístíte ve WEB_tParameters. Tlačítko OK dovolí uživateli Web odstranit dialog.

14.3.3. Úpravy metody Při Web spojení

Nyní potřebujeme upravit metodu Při Web spojení, abychom využili nový dialog. Všechno co potřebujeme udělat je získat informace z parametrů \$1 a \$2 a tyto informace korektně zformátovat jako vnořenou HTML.

1. Jděte do Prostředí návrháře 4D.
2. Nahraďte metodu Při Web spojení, následující metodou a potom importujte textový soubor Echo URL and HTTP Header pomocí textového editoru:

```
If (False)
  ` Database Method: On Web Connection
```

Používání 4D Databáze na Webu

Při Web spojení

```
` ACI University Programming Classes  
` Created By: David Adams  
` Date: 3/21/97
```

```
` Purpose: Demonstruje skupinová pole na WEB.
```

```
◊f_Version6x40 := True  
◊fD_Adams := True
```

```
End if
```

```
C_TEXT(WEB_tParameters)
```

```
WEB_tParameters:=Char(1)
```

```
WEB_tParameters:=WEB_tParameters+"<B>URL:</B>"+$1+"<P>"  
WEB_tParameters:=WEB_tParameters+Char(Carriage return)+Char(Carriage return)  
WEB_tParameters:=WEB_tParameters+"<B>HTTP Header Information:</B><BR>"  
WEB_tParameters:=WEB_tParameters+Char(Carriage return)+$2
```

```
DIALOG([zDialogs];"WEB_ShowParameters")
```

```
MENU BAR(5)
```

3. Uložte metodu Při Web spojení.

Metoda Při Web spojení nyní obsahuje několik řádek kódu, které kombinují proměnné \$1 a \$2 do textové proměnné nazvané WEB_tParameters. Je to proměnná, kterou jsme použili v předchozím dialogu a zde bude naplněna. Je nutno si povšimnout dvou věcí, proměnná WEB_tParameters začíná znakem Char (1), to oznamuje 4D, že hodnota proměnné by měla být interpretována jako HTML a ne regulérní text. Ti z vás, kteří již vědí něco více o HTML rozeznají tag **Tučně**. Všimněte si, že zde již nejsou žádné další HTML tagy. 4D vygeneruje celkový dokument tj. hlavičku a tag obsahu, pokud je použit 4D formulář.

14.3.4. Odeslání dat

Nyní nejjednodušší část. S novou metodou Při Web spojení pokaždé, když se uživatel připojí, je mu odeslána zpět informace o URL a hlavička HTTP, kterou odeslal jeho prohlížeč. Samozřejmě to není prakticky a reálný systém, ale je to výborný nástroj pro ladění. Vyzkoušejme si jej:

1. Opusťte Prostředí návrháře.
2. Přejděte do Web prohlížeč.
3. Otevřete nové spojení s vaší databází.
4. Nyní uvidíte jaká je vaše URL a že je zde mnoho HTTP informací.

14.4. Hlavička HTTP: Co to všechno znamená ?

14.4.1. Přehled

Co znamenají všechny údaje v hlavičce HTTP? Hlavička HTTP, kterou jste si prohlédli byla vytvořena vaším prohlížečem nikoli 4D. Různé prohlížeče umísťují do záhlaví různé informace. Jestliže používáte cookies nebo připojené knihovny JavaScript (nepodporované ve v6) rovněž se objeví v hlavičce. Hlavička

Používání 4D Databáze na Webu

Při Web spojení

je užívána k tomu, aby Web serveru sdělila několik věcí, včetně toho, jakému druhu dokumentu prohlížeč rozumí, adresu stroje klienta, název prohlížeče a verzi HTTP, které prohlížeč rozumí. Zde jsou příklady hlaviček z Netscape a Internet Explorer připojených k téže databázi v tutéž dobu:



Netscape Navigator

GET / HTTP/1.0

Connection: Keep-Alive

User-Agent: Mozilla/3.0Gold (Macintosh; I; PPC)

Host: 192.9.200.181

Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
/



MS Internet Explorer

GET / HTTP/1.0

Accept: */*, q=0.300

Accept: audio/wav

Accept: audio/x-wav

Accept: audio/aiff

Accept: audio/x-aiff

Accept: audio/basic

Accept: multipart/x-mixed-replace

Accept: text/url

Accept: text/plain

Accept: text/html

User-Agent: Mozilla/2.0 (compatible; MSIE 3.0b1;
Mac_PowerPC)

Accept-Language: en

Connection: Keep-Alive

Host: 192.9.200.181

14.4.2. User-Agent

Některé z těchto informací jsou stejné a jiné trochu odlišné. Pole User-Agent určuje samotný prohlížeč. V oficiální definici HTTP prohlížečů se samotný prohlížeč nazývá “user agent” takže pole v záhlaví zachovává tutéž terminologii. Mozilla je kódové označení pro Navigator. Tuto informaci si můžete přečíst vaším kódem a použít ji např. ke změně formulářů metod nebo vnořených HTML bloků podle připojeného prohlížeče.

14.4.3. Rozložení HTTP hlavičky

Rozložení hlavičky HTTP do polí vám umožní lépe přečíst jednotlivé položky hlavičky. Není to těžká úloha, ale samozřejmě není příliš zajímavá. Naštěstí nemusíte tento kód psát, přidejme do databáze generickou rutinu, která úlohu provede.

1. Přejděte do Prostředí návrháře 4D.

Používání 4D Databáze na Webu

Při Web spojení

2. Vytvořte novou metodu projektu WEB_Text2Array a nainportujte textový soubor WEB_Text2Array s použitím textového editoru.

```
If (False)
  ` Metoda: WEB_Text2Array (Text;->Array;String (10))
  ` WEB_Text2Array (Text to parse;->Array to parse into;Delimiter)
  `
  ` ACI University Programming Classes
  ` Created By: David Adams
  ` Date: 3/21/97

  ` Purpose:
  ` Tato rutina vezme textovy blok a rozdeli jej na prvky array.
  ` Textovy blok muze byt promenna nebo pole, pokud je to promenna.
  ` uz musi existovat. Array muze byt string nebo text array
  ` ale pokud pouzijete string array ujistete se, ze kazdy prvek se muze vejít do prvku
  array.
  ` Array uz musí existovat. Polozky jsou pridany na konec array.
  ` **jiz existujici prvky se nemeni **.

  ` 4D nepodporuje ukazatele na lokalni promenne nebo arrays, takze libovolna
  promenna nebo
  ` array musi byt procesu nebo meziproceni.

  ` promenné klicovych slov ve 4D Insider:
  ⋄fD_Adams := True
  ⋄f_Version6x40 := True
  ⋄fGeneric := True
End if

  ` Deklaruje parametry
C_TEXT($1;$tSourceText)
C_POINTER($2;$pArrayPointer)
C_STRING(10;$3;$sDelimiter)

  ` Deklaruje lokalni promenne
C_TEXT($tCopyOfTextForItemCheck)
C_LONGINT($LPosition)
C_LONGINT($LStartingElement)
C_LONGINT($LItemsToAdd)
C_LONGINT($LLastElement)
C_LONGINT($LDelimiterLength)
C_LONGINT($LLastCharacter )

$tSourceText := $1
$pArrayPointer := $2
$sDelimiter := $3

$tSourceTextLength := Length($tSourceText)
$LDelimiterLength := Length($sDelimiter)

  ` Tato rutina pridava na konec array, takze naleznete
  ` velikost array a pridejte 1 prvek.
```

Používání 4D Databáze na Webu

Při Web spojení

```
$LStartingElement := Size of array($pArrayPointer->)+1

` Kolikrat je oddelovac ve zdrojovem textu? To rekne rutine
` kolik prvku pridat do array.
$stCopyOfTextForItemCheck := Replace string($stSourceText;$sDelimiter;"")
$LItemsToAdd:=Length($stSourceText)-Length($stCopyOfTextForItemCheck)

` Potrebne deleni, pri vkladani nekolikanasobneho oddelovace
$LItemsToAdd := $LItemsToAdd/$LDelimiterLength

` Pridani polozek jako blok lepsi nez jedne nekolikrat.
INSERT ELEMENT($pArrayPointer->,$LStartingElement;$LItemsToAdd)

` Jake bude cislo posledniho prvku v array s novou velikosti?
$LLastElement := Size of array($pArrayPointer->)

` Smycka pres zdrojovy text kopirujici kazdy radek do spravneho
` prvku array.
For ($Element;$LStartingElement;$LLastElement)

    $LPosition := Position($sDelimiter;$stSourceText)
    $pArrayPointer->{$Element} := Substring($stSourceText;1;$LPosition-1)
    $stSourceText := Substring($stSourceText;$LPosition+$LDelimiterLength)

End for

If ($stSourceText # "") ` Něco zbyva na konci
    $LLastElement := $LLastElement+1
    INSERT ELEMENT($pArrayPointer->,$LLastElement;1)
    $pArrayPointer->{$LLastElement} := $stSourceText
End if

` Zjisteni posledniho.
$LLastElement := Size of array($pArrayPointer->)
$LLastCharacter := Length($pArrayPointer->{$LLastElement} )
if ($pArrayPointer->{$LLastElement} <= $LLastCharacter <= "?")
    $pArrayPointer->{$LLastElement} := Substring ($pArrayPointer->{$LLastElement}
;1; $LLastCharacter -1)
End if
` Konec metody
```

3. Uložte metodu.

Právě jste přidali do své databáze generickou rutinu, která vezme blok textu a rozdělí jej do jednotlivých řádek. Každá řádka je vložena do svého vlastního prvku pole. Použijeme tuto rutinu k rozložení přicházejících URL a hlavičky HTTP.

4. Nahraďte metodu databáze Při Web spojení a importujte textový soubor Echo s použitím textového editoru:

```
If (False)
    ` Database Method: On Web Connection
    ` ACI University Programming Classes
```

Používání 4D Databáze na Webu

Při Web spojení

` Created By: David Adams

` Date: 3/21/97

` Purpose: Demonstrování skupinových polí WEB.

◊f_Version6x40 := True

◊fD_Adams := True

End if

` Inicializuje array uzivane WEB_Text2Array:

ARRAY TEXT(WEB_atURL;0)

ARRAY TEXT(WEB_atHTTPHeader;0)

` Prikaz substring nalezne prvni lomitko, které je vzdy

` pritomno a nezajima nas.

WEB_Text2Array (Substring(\$1;2);->WEB_atURL;"/")

WEB_Text2Array (\$2;->WEB_atHTTPHeader;Char(10))

` Odesle promennou uzivateli:

DIALOG([zDialogs];"Web_ShowParameters")

` OK, ukaz menu:

MENU BAR(5)

` End of method

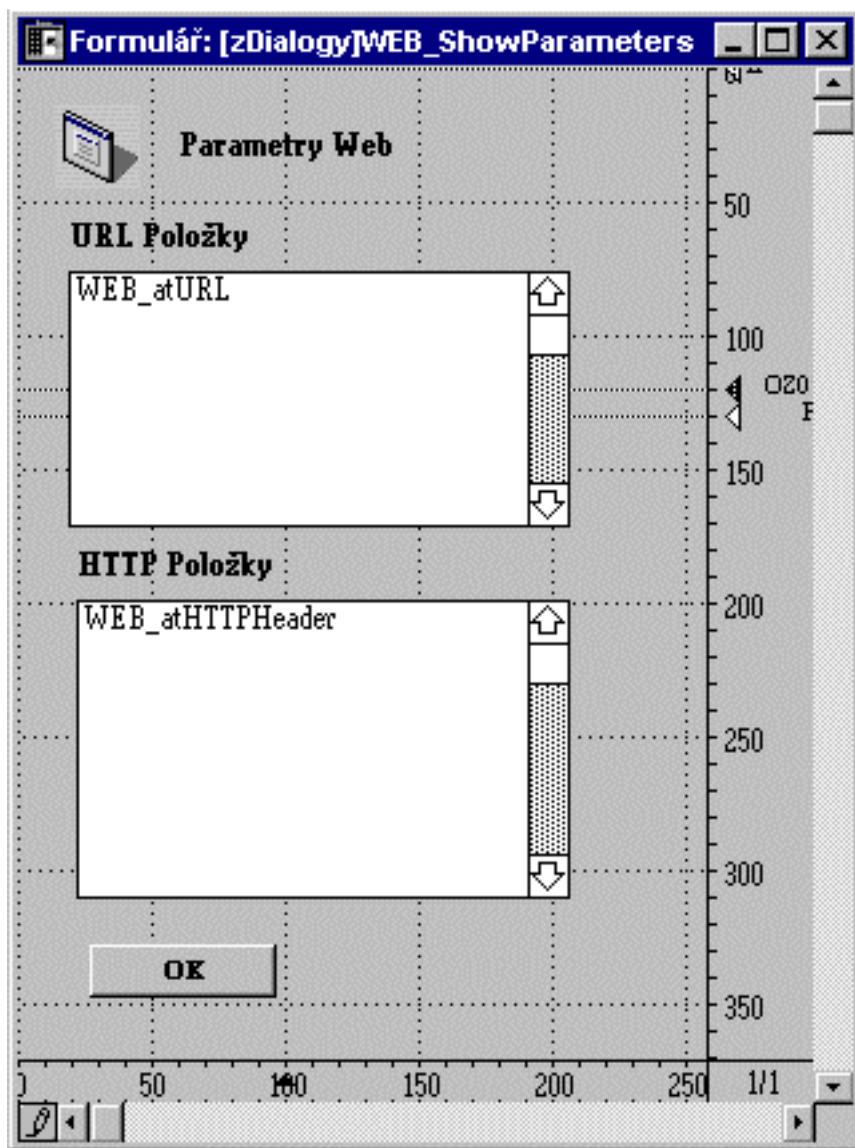
Odstranili jste veškerý kód, který vytvářel proměnnou statické HTML a nahradil ji kódem k rozložení přicházející informace od array. Rutina WEB_Text2Array vezme každý parametr a rozloží ho. Oddělovač mezi prvky URL je "/" a oddělovač uvnitř hlavičky je návrat vozíku (ASCII 10). 4D nezavedla tyto oddělovače, jsou standartní části HTTP a HTML užívané všemi prohlížeči a Web servery.

OK, nyní vraťme array zpět uživateli místo předchozího textového bloku.

5. Otevřete formulář Web_ShowParameters.
6. Odstraňte proměnnou WEB_tParameters.
7. Přemístěte tlačítko OK dolů o několik centimetrů.
8. Přidejte dvě posuvné oblasti: WEB_atURL a WEB_atHTTPHeader.
9. Umístěte posuvné oblasti tak, aby se nedotýkali a jedna byla nad druhou.
10. Přidejte statický text nad každou posuvnou oblast k jejich rozlišení: "URL položky" a "HTTP položky"
11. Uložte formulář.
12. Formulář by měl vypadat následovně:

Používání 4D Databáze na Webu

Při Web spojení



13. Opust'te Prostředí návrháře.
14. Otestujte tyto změny v prohlížeči Web.

Nyní přejděte do prohlížeče Web a otevřete nové spojení uvidíte, že informace z URL a hlavičky HTTP jsou rozloženy v posuvné oblasti. Hodnoty jednotlivých prvků jsou takové, že tyto informace můžete lehce v kódu využít.

14.5. URL

Takže jak jsme nyní viděli informace v URL odeslaná zpět, byla prázdná a to proto, že 4D sleduje tu část URL za adresou stroje. Zde vidíme jak je URL reportováno:

URL v prohlížeči

4D vrací URL 4D

Používání 4D Databáze na Webu

Při Web spojení

`http://www.4D-Rules.com/`

`/`

`http://www.4D-Rules.com/Bob/Home.html`

`/Bob/Home.html`

Vyzkoušejme si to nyní.

14.5.1. Testování rutiny rozložení URL.

1. Jděte do svého prohlížeče Web.
2. Napište název svého počítače, `/My/Special/Stuff` a klepněte return.
3. 4D vrátí ozvěnou zpět co jste zadali. Prvky array URL jsou nyní tři: My, Special a Stuff.
4. Stiskněte OK.
5. Nyní uvidíte svou nabídku Web.

Poznámka: Protože nemáme nastavený time out musíme po vyzkoušení Web připojení, přejít do seznamu procesů v Prostředí návrháře a odstranit proces Web připojení.

Nezáleží na tom, kde je URL napsána, může být napsána přímo v oblasti adresy prohlížeče nebo může být vložena do klepnutelného odkazu. 4D rozdíl nepozná, totéž platí ostatně o všech Web serverech. Povšimněte si, že dělení URL a kód dialogu běží předtím, než se objeví nabídka. To je důležité, protože nás to upozorňuje na to jak má být strukturována metoda Při Web spojení. A to dále znamená, že do metody Při Web spojení můžeme implementovat speciální chování. Podívejme se jak to udělat.

14.6. Vytváření struktury řízení akcí založený na URL

14.6.1. Nastavení

Ve 4D je jednoduché vytvořit podporu akcí založených přímo na požadavku v URL. Můžete např. přidat podporu pro standartní akce, pro některé či všechny vaše tabulky:

Akce	Tabulka	URL
Add	Zákazníci	<code>/Add/Zákazníci/</code>
Add	Produkty	<code>/ Add /Produkty/</code>
Modify	Zákazníci	<code>Modify/ Zákazníci /</code>
Modify	Produkty	<code>Modify / Produkty /</code>

Zkusme to udělat jednodušším způsobem:

1. Jděte do Prostředí návrháře ve 4D.
2. Nahraďte databázovou metodu Při Web spojení importem z textového souboru URL Action Handler 1:

```
If (False)
` Database Method: On Web Connection
` ACI University Programming Classes
` Created By: David Adams
```

Používání 4D Databáze na Webu

Při Web spojení

```
` Date: 3/21/97

` Ucel: Pusobi pri spusteni Web a ovlada akce.

◇f_Version6x40 := True
◇fD_Adams := True

End if

C_TEXT($1;$2)

C_TEXT($URL)

` Inicializuje arrays pouzite s WEB_Text2Array:
ARRAY TEXT(WEB_atURL;0)
ARRAY TEXT(WEB_atHTTPHeader;0)

` Prikaz substring nalezne prvni lomitko, které je vzdy
` pritomno a nezajima nas.
WEB_Text2Array (Substring($1;2);->WEB_atURL;"/")
WEB_Text2Array ($2;->WEB_atHTTPHeader;Char(10))

` Zasle uzivateli promennou k prohlednuti:
` DIALOG([zDialogs];"Web_ShowParameters")

$URL:=$1

Case of
: ($URL="/Add@") ` Pozadavek na pridani zaznamu.

Case of
: ($URL="/Add/Zákazníci@")
INPUT FORM([Zákazníci];" Zákazníci.iw")
ADD RECORD([Zákazníci];*)

: ($URL="/Add/Produkty@")
INPUT FORM([Produkty];" Produkty.iw")
ADD RECORD([Produkty];*)
End case

: ($URL="/Modify@") ` Pozadavek na upravy.
Case of
: ($URL="/Modify/ Zákazníci @")
INPUT FORM([Zákazníci];" Zákazníci.iw")
OUTPUT FORM([Zákazníci];" Zákazníci.ow")
ALL RECORDS([Zákazníci])
MODIFY SELECTION([Zákazníci])

: ($URL="/Modify/ Produkty @")
INPUT FORM([Produkty];" Produkty.iw")
OUTPUT FORM([Produkty];" Produkty.ow")
ALL RECORDS([Produkty])
MODIFY SELECTION([Produkty])
```

Používání 4D Databáze na Webu

Při Web spojení

End case

End case

```
` OK, ukaz Web nabidku:  
MENU BAR(5)  
`End of method
```

3. Vytvořte vstupní formulář pro Produkty a pojmenujte jej Produkty.iw
4. Vytvořte výstupní formulář pro [Produkty] a pojmenujte jej Produkty.ow
5. Uložte všechny nové metody a formuláře a otestujte všechny změny.

Dostali jsme nyní strukturu, která se stará o ošetření několika konkrétních příchozích URL. Přejděte zpět do vašeho prohlížeče Web a napište adresu vašeho stroje a přidejte /Add/Zákazníci:

<http://192.9.200.181/Add/Zákazníci>

Objeví se vám vstupní zadávací formulář pro zákazníky pro nový záznam. Tento systém můžete rozšířit tak, aby ošetřoval libovolný počet akcí a ovládal soubory genericky.

14.6.2. Vylepšení tohoto systému

Existuje několik způsobů vylepšení systému, který jste právě viděli. Výraz case je blokový a může být již příliš velký a nepřehledný při několika akcích a třeba i dvou tabulkách. Jestliže používáte standartní názvy formulářů a standartní akce tabulek. Můžete odstranit většinu tohoto kódu. Např. když máte vstupní a výstupní formuláře Web pro každou tabulku a chcete je použít na Web, použijte stejné jméno jako u tabulky:

Formulář	Syntaxe	Příklad
Vstupní	Název tabulky + ".iw"	Zákazníci.iw
Výstupní	Název tabulky + ".ow"	Zákazníci.ow

Jestliže se jste se zúčastnili kursu pro středně pokročilé jste již určitě zběhlí v používání generických názvů a metod. Nyní se podívejme na jednoduchý systém pro přidávání a úpravy napsaný genericky a založený na URL. Chystáme se přidat tři nové metody a opět modifikovat Při Web spojení.

1. Přejděte do Prostředí návrháře 4D.
2. Vytvořte vstupní a výstupní formuláře Web pro tabulku Produkty s použitím vzoru ACI Video WEB, který jsme si již vytvořili.
3. Vytvořte novou metodu projektu URL_TablePointer a naimportujte text ze souboru URL_TablePointer.

```
If (False)  
` URL_TablePointer (Table name)  
` ACI University Programming Classes  
` Created By: David Adams  
` Date: 3/21/97
```


Používání 4D Databáze na Webu

Při Web spojení

```
`
` Purpose: This routine turns a table name into a table
` pointer. A Nil pointer is returned if the table name is unknown.
`
` Note: You can make this more efficient by loading arrays with table
` names and table pointers at startup. Then, instead of looping
` through tables a simple Find in array is enough.
`
` 4D Insider keyword variables:
`fD_Adams := True
`f_Version6x40 := True

End if

C_POINTER($0;$pTablePointer)
C_TEXT($1)
C_STRING(31;$sTableName)

C_LONGINT($LNumberOfTables)
C_LONGINT($LTableNumber)

` Substring? The URL might contain *anything*, even something
` too long. The Substring function makes sure we don't use too
` many character.
`sTableName := Substring($1;1;31)

$LNumberOfTables := Count tables

For ($LTableNumber;1;$LNumberOfTables)
  If (Table name($LTableNumber)= $sTableName)
    ` It's a match.
    $pTablePointer:=Table($LTableNumber) ` Create the pointer.
    $LTableNumber:=$LNumberOfTables+1 ` Get out of the loop.
  End if
End for

$0:=$pTablePointer

` End of method.
```

4. Vytvořte novou metodu projektu URL_AddRecords, naimportujte text ze souboru **URLAddRecords**

```
If (False)
  ` URL_AddRecords (Table name)
  ` ACI University Programming Classes
  ` Created By: David Adams
  ` Date: 3/21/97
`
` This routine adds record to table based on a call
` from a URL.
`
```

Používání 4D Databáze na Webu

Při Web spojení

```
` 4D Insider keyword variables:
♦fD_Adams := True
♦f_Version6x40 := True

End if

C_TEXT($1)

C_POINTER($pTablePointer)

` Turn the passed name into a table pointer.
` A Nil pointer is returned if the table is not known.
$pTablePointer := URL_TablePointer ($1)

If (Not(Nil($pTablePointer))) ` It's a good table pointer.
  $sTableName := Table name($pTablePointer) ` Get the table's name.
  INPUT FORM($pTablePointer->;$sTableName+".iw")
  ADD RECORD($pTablePointer->;*)
End if
` End of method
```

5. Vytvořte novou metodu projektu URL_ModifyRecords a naimportujte textový soubor URL_ModifyRecords.

```
If (False)
  ` URL_ModifyRecords (Table name)
  ` ACI University Programming Classes
  ` Created By: David Adams
  ` Date: 3/21/97

  ,

  ` Purpose: This routine modifies records in a table based
  ` on a call from a URL.
  ,

  ` 4D Insider keyword variables:
  ♦fD_Adams := True
  ♦f_Version6x40 := True

End if

C_TEXT($1)

C_POINTER($pTablePointer)

` Turn the passed name into a table pointer.
` A Nil pointer is returned if the table is not known.
$pTablePointer:=URL_TablePointer ($1)

If (Not(Nil($pTablePointer))) ` It's a good table pointer.
  $sTableName := Table name($pTablePointer) ` Get the table's name.
  INPUT FORM($pTablePointer->;$sTableName+".iw")
  OUTPUT FORM($pTablePointer->;$sTableName+".ow")
  QUERY($pTablePointer->)
```

Používání 4D Databáze na Webu

Při Web spojení

```
        MODIFY SELECTION($pTablePointer->,* )  
    End if  
  
    `End of method
```

Používání 4D Databáze na Webu

Při Web spojení

6. Nahradíte databázovou metodu Při Web spojení importem ze souboru URL Action Handler 2 s použitím textového editoru:

```
If (False)
  ` Database Method: On Web Connection
  ` ACI University Programming Classes
  ` Created By: David Adams
  ` Date: 3/21/97

  ` Purpose: Acts as Web startup and action handler

  ⋄f_Version6x40 := True
  ⋄fD_Adams := True

End if

C_TEXT($1;$2)

C_TEXT($URL)

  ` Initialize arrays used with WEB_Text2Array:
ARRAY TEXT(WEB_atURL;0)
ARRAY TEXT(WEB_atHTTPHeader;0)

  ` Substring gets rid of the first "/", which is always there and
  ` does not interest us.
WEB_Text2Array (Substring($1;2);->WEB_atURL;"/")
WEB_Text2Array ($2;->WEB_atHTTPHeader;Char(10))

  ` Send the variable to the user to see:
  ` DIALOG([zDialogs];"Web_ShowParameters")

$URL:=$1

  ` How many items are in the URL?
$URL_elements := Size of array(WEB_atURL)

If ($URL_elements>1)
  ` There are at least two URL elements, the minimum we need for a special action.

  Case of
  : (WEB_atURL{1}="Add")
    URL_AddRecords (WEB_atURL{2})

  : (WEB_atURL{1}="Modify")
    URL_ModifyRecords (WEB_atURL{2})
  End case

End if

  ` OK, now show the regular Web menu:
MENU BAR(5)
```

Používání 4D Databáze na Webu

Při Web spojení

`End of method

Dlouhý blok case užitý v předchozím byl nahrazen několika jednoduchými testy a voláním rutin. Nyní můžete ošetřit libovolný počet tabulek v databázi pomocí této struktury.

Rutiny, které jste přidali jsou zcela jednoduché, ale jsou rovněž dostatečně generické k ošetření libovolné tabulky v jakékoliv připravované databázi. Jestliže již máte generické rutiny, jako tyto ve svých vlastních systémech, můžete je použít ve všech vašich kódech. Tyto rutiny byly zvoleny jako příklad. Podívejme se jak pracují:

Rutina `URL_AddRecords` používá podrutinu `URL_TablePointer` k nalezení známé tabulky. Jestliže existuje pak `URL_AddRecords` použije standardní vstupní formulář a začne s vkládáním záznamů. To je vše.

`URL_ModifySelection` je podobná `URL_AddRecords`. Všimněte si, že dotaz `QUERY` je použit, aby dal uživateli možnost vybrat si prohlížené záznamy. 4D Editor dotazů může být použit i přes Web nebo si samozřejmě můžete vytvořit svůj vlastní editor dotazů používaný přes Web.

`URL_TablePointer` začíná názvem tabulky — předpokládaným z `URL` — a kontroluje jestli odpovídá nějakému existujícímu názvu tabulky. Tato rutina může být mnohem rychlejší pokud dopředu zavedete do paměti řadu arrays s názvy tabulek a ukazateli. Velice jednoduše můžete zavést i systém zástupců, aby jste dovolili uživateli vkládat zástupce na názvy tabulek místo samotných občas nepochopitelných názvů tabulek.

7. Uložte všechny tyto opravené metody.
8. Vraťte se do vašeho Web prohlížeče a otestujte tyto změny.
9. Vyzkoušejte další tabulky ze své databáze jako [zStruktura].

14.7. Zamykání záznamů

Jak jsme se zmínili, 4D za vás ovládá veškeré zamykání záznamů, dokonce i tehdy, když je klientem prohlížeč Web. Zamykání záznamů je úloha centralizovaná databázovým strojem, takže ve skutečnosti nezáleží na tom jaký druh klienta používáte. Podívejme se na tuto akci a otestujme si náš nový systém akcí `URL` spolu s ním:

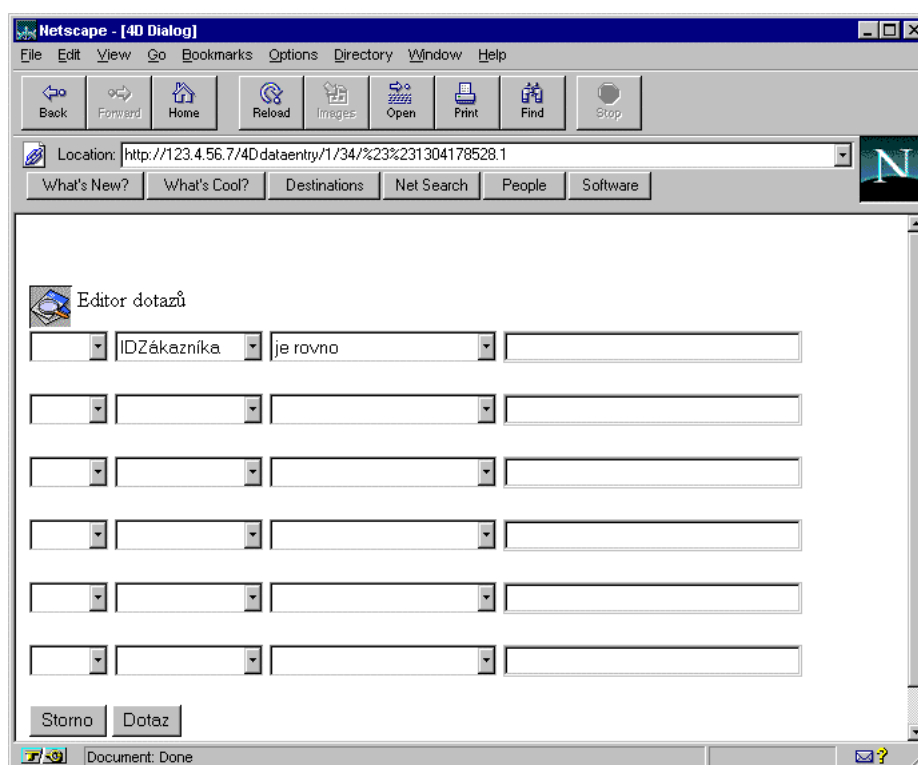
1. Přejděte do prohlížeče Web.
2. Připojte s k vaší databázi pomocí adresy a akce `URL`

`http://192.9.200.181/Modify/Zákazníci`

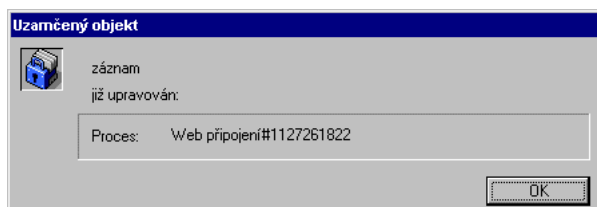
Používání 4D Databáze na Webu

Při Web spojení

3. Když se objeví Editor dotazů, zadejte “název není rovno” a ponechte vstupní pole prázdné. (může se stát, že budete potřebovat změnit velikost okna prohlížeče, aby se vám dobře zobrazil Editor dotazů).



4. Klepněte na tlačítko Dotaz.
5. 4D vám odešle první stránku seznamu záznamů, které nalezne.
6. Upravte první záznam zákazníka.
7. Přejděte do Prostředí aplikace ve 4D.
8. Vyberte Zákazníci a klepněte na tlačítko Všechny záznamy.
9. Upravte první záznam.
10. Uvidíte následující upozornění:

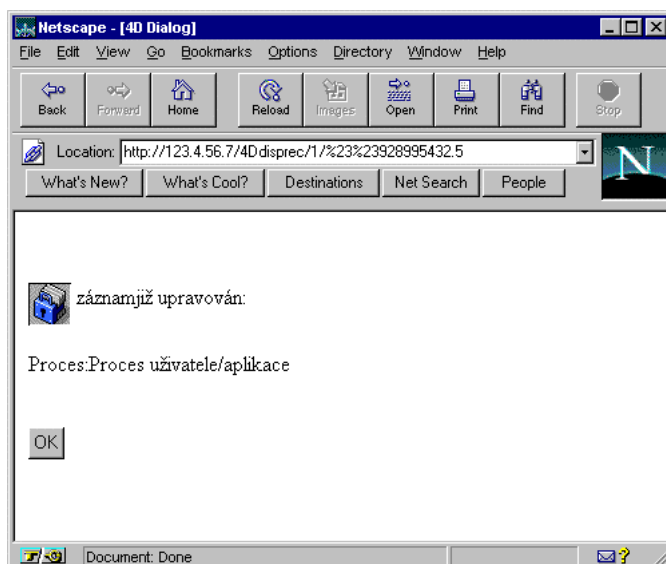


11. Web klient zamknul záznam. Klepněte OK.
12. Přejděte na další záznam ve 4D.
13. Ponechte tento záznam otevřený — uzamčený — a vraťte se do prohlížeče Web.
14. Klepněte na tlačítko Další záznam ve stránce prohlížeče

Používání 4D Databáze na Webu

Při Web spojení

15. Uvidíte následující upozornění:



16. Klepněte OK.
17. Klepněte Storno a vraťte se do seznamu záznamů.
18. Klepněte na tlačítko Hotovo a navraťte se do Web nabídky.
19. Ve 4D přejděte do Prostředí uživatele.
20. Stornujte zobrazený záznam zákazníka.

4D ovládá uzamykání záznamů v obou prostředích. Ovládání uzamykání záznamů pro Web klienty je v jiných systémech poněkud náročné.

14.8. Použití dat k automatizaci přístupu k databázi

V předchozím příkladě jsme k správné funkci přidávání a úprav záznamů přes Web použili standardizovaná jména vstupním a výstupním formulářů jednotlivých tabulek. Ale co když nechcete přístup přes Web ke všem tabulkám? K tomuto účelu můžete použít jednoduchý soubor, který bude obsahovat předvolby pro podporované akce jednotlivých tabulek. V této databázi již existuje soubor pro ukládání standardních a uživatelských informací o tabulkách:

Používání 4D Databáze na Webu

Při Web spojení

zStruktura	
ČísloTabulky	I
ČísloPole	I
PlatnýNázev	A
NázevZástupce	A
PořadíZástupce	I
VložitVlastníHledání	B
VložitVlastníTřídění	B
URL_PřidatPovoleno	B
URL_UpravitPovoleno	B
URL_ZobrazitPovoleno	B

Můžete přidat nová pole pro všechny URL akce, které jste se rozhodli podporovat. Když URL požadavek přichází, nemusíte testovat pouze to která tabulka je známá a existuje, ale také které akce jsou přes Web podporovány a které ne. To vám dovolí velmi rychle zapínat a vypínat přístup k tabulkám přes Web, dokonce i v kompilované databázi.

14.8.1. Omezení přístupu k tabulkám.

1. Přidejte následující pole do [zStruktura]

Název pole	Typ
URL_PřidatPovoleno	Boolean
URL_UpravitPovoleno	Boolean
URL_ZobrazitPovoleno	Boolean

2. Vypněte vlastnost povinné z tabulky [zStruktura]Číslo tabulky
3. Otevřete vstupní formulář [zStruktura].
4. Přidejte tři nová pole do formuláře.
5. Jděte do Prostředí uživatele.
6. Proveďte metodu E_Set2ReadWrite
7. Přepněte se do tabulky [zStruktura].
8. Proveďte dotaz pro záznamy, kde je číslo tabulky 0.
9. Změňte pole, které začínají URL_ jak požadujete.
10. Vytvořte novou metodu projektu URL_DisplayRecords importem z textového souboru URL_DisplayRecords s použitím textového editoru.

```
If (False)
  ` URL_DisplayRecords (Table name)
```


Používání 4D Databáze na Webu

Při Web spojení

```
` ACI University Programming Classes
` Created By: Kent Wilbur
` Date: 3/21/97

`
` Purpose: This routine displays records in a table based
` on a call from a URL.
`
` 4D Insider keyword variables:
⋄fK_Wilbur := True
⋄f_Version6x40 := True

End if

C_TEXT($1)

C_POINTER($pTablePointer)

` Turn the passed name into a table pointer.
` A Nil pointer is returned if the table is not known.
$pTablePointer:=URL_TablePointer ($1)

If (Not(Nil($pTablePointer))) ` It's a good table pointer.
  $sTableName := Table name($pTablePointer) ` Get the table's name.
  INPUT FORM($pTablePointer->,$sTableName+".iw")
  OUTPUT FORM($pTablePointer->,$sTableName+".ow")
  QUERY($pTablePointer->)
  DISPLAY SELECTION($pTablePointer->*)
End if

`End of method
```

11. Nahraďte databázovou metodu Při Web spojení, pak naimportujte textový soubor URL Action Handler 3 s použitím textového editoru:

```
If (False)
  ` Database Method: On Web Connection
  ` ACI University Programming Classes
  ` Created By: David Adams
  ` Date: 3/21/97

  ` Purpose: Acts as Web startup and action handler

  ⋄f_Version6x40 := True
  ⋄fD_Adams := True

  Modified: To verify table action validity
  ⋄fK_Wilbur := True
End if

C_TEXT($1;$2)

C_TEXT($URL)
```

Používání 4D Databáze na Webu

Při Web spojení

```
` Initialize arrays used with WEB_Text2Array:
ARRAY TEXT(WEB_atURL;0)
ARRAY TEXT(WEB_atHTTPHeader;0)

` Substring gets rid of the first "/", which is always there and
` does not interest us.
WEB_Text2Array (Substring($1;2);->WEB_atURL;"/")
WEB_Text2Array ($2;->WEB_atHTTPHeader;Char(10))

` Send the variable to the user to see:
` DIALOG([zDialogs];"Web_ShowParameters")

$URL:=$1

` How many items are in the URL?
$URL_elements := Size of array(WEB_atURL)

If ($URL_elements>1)
    ` There are at least two URL elements, the minimum we need for a special action.

    QUERY( [zStructure];[zStructure]TableNumber = 0;*)
    QUERY( [zStructure]; & ;[zStructure]AliasName = WEB_atURL {2})

    Case of
        : (Records in selection ([zStructure]) = 1 )
        Case of
            : (WEB_atURL {1}="Add") & ([zStructure]URL_AddAllowed)
                URL_AddRecords (WEB_atURL {2})

            : (WEB_atURL {1}="Modify") & ([zStructure]URL_ModifyAllowed)
                URL_ModifyRecords (WEB_atURL {2})

            : (WEB_atURL {1}="Display") & ([zStructure]URL_DisplayAllowed)
                URL_DisplayRecords (WEB_atURL {2})

        Else
            ALERT ("Action chosen not recognized!")
        End case
    End case

End if

` OK, now show the regular Web menu:
MENU BAR(5)
` End of method
```

12. Uložte změny.
13. Otestujte změny ve vašem Web prohlížeči.

Používání 4D Databáze na Webu

Při Web spojení

14.9. Zpětná volání

14.9.1. Podstata

Vedle přímého ošetřování akcí, existuje ještě další důvod k monitorování přicházejících URL to jsou zástupci metod. Můžete navrhnout svůj systém tak, že můžete vložit volání na metody a jejich případné parametry přímo v URL. Podíváme se na to podrobně.

Poznámka: 4D rovněž podporuje přímé zpětné volání metod. Pro tento účel nepotřebujete monitorovat URL, protože se o to postará 4D. Na tento předmět se podíváme později při přímém vytváření HTML.

14.9.2. Experiment

Přidejme si pro náš experiment testovací metodu:

1. Jděte do Prostředí návrháře.
2. Vytvořte novou metodu projektu WEB_CountObjects importem z textového souboru WEB_CountObjects s použitím textového editoru.

```
If (False)
  ` WEB_CountObjects ({Table number})
  ` ACI University Programming Classes
  ` Created By: David Adams
  ` Date: 3/21/97
  ,
  ` Purpose: Toto je demonstrační rutina, která něco počítá a potom zobrazí
  ` výsledky v upozornění. Co se počítá závisí na parametrech:
  ` Zadání parametry:      Počítá tabulky v databázi
  ` Číslo tabulky:        Počítá pole, záznamy v tabulce a záznamy ve výběru.
  ` Špatné číslo tabulky: Counts parameters passed.
  ,
  ` Klíčová slova proměnné ve 4D Insider:
  ⋄fD_Adams := True
  ⋄f_Version6x40 := True

End if

  ` Deklaruje parametry
C_TEXT($1;$tTableNameFromURL)

  ` Deklaruje lokální proměnné
C_STRING(80;$sAlertMessage)
C_LONGINT($LTableName)
C_POINTER($pTablePointer)

$sAlertMessage:=""

  ` Clear out stray characters from URL, if any.
If (Count parameters>0)
  $tTableNameFromURL := Replace string($1;"?";"")
End if
```

Používání 4D Databáze na Webu

Při Web spojení

```
If (Count parameters<1) ` No parameters, count tables.
    $sAlertMessage := "Tables in database:"+Char(Carriage return)
    $sAlertMessage := $sAlertMessage+String(Count tables)

Else                                     ` There's a parameter, check if it's a good table
number.

    $LTableNumber := Num($tTableNumberFromURL)

    If ($LTableNumber <= Count tables) & ($LTableNumber>0)           ` OK, it's a valid
table number.
        $pTablePointer := Table($LTableNumber)
        $sAlertMessage := "Table Count"+Char(Carriage return)
        $sAlertMessage := $sAlertMessage+"#:"
        "+String($LTableNumber)+Char(Carriage return)
        $sAlertMessage := $sAlertMessage+"Name  "+Table name($LTableNumber)+
            Char(Carriage return)
        $sAlertMessage := $sAlertMessage+"Fields: "+String(Count
fields($LTableNumber))+
            Char(Carriage return)
        $sAlertMessage := $sAlertMessage+"Recs:  "+String(Records in table
($pTablePointer->))+Char(Carriage return)

    Else                                     ` Not a valid table number, count parameters we
received.
        $sAlertMessage := "Web_CountObjects received this many
parameters:"+Char(Carriage return)
        $sAlertMessage := $sAlertMessage+String(Count parameters)
    End if

End if

ALERT($sAlertMessage)

` Konec metody.
```

3. Uložte a uzavřete metodu.

Web_CountObjects je rutina, která posílá rozdílná hlášení v závislosti na tom kolik parametrů ji dodáte. Jestliže nezadáte žádný parametr, spočítá tabulky v databázi, jestliže ji zadáte číslo tabulky, vrátí statistiku o tabulce a jestliže uděláte chybu, řekne vám kolik parametrů jste zadali. Pokud chceme použít tuto metodu na Web, musíme ji přidat do Web nabídky a nebo jako zpětné volání napsat v URL. Použijeme druhou možnost.

4. Změňte databázovou metodu Při Web spojení následujícím způsobem:

```
...
: (WEB_atURL{1}="Display") & ([zStructure]URL_DisplayAllowed)
  URL_DisplayRecords (WEB_atURL{2})

Else
  ALERT ("Action chosen not recognized!")
```

Používání 4D Databáze na Webu

Při Web spojení

```
❖ End case
❖
❖ : (WEB_atURL{1}="Callback")
❖ If (WEB_atURL{2}="Web_CountObjects")
❖     Web_CountObjects
❖ End if
```

End case

...

5. Nyní by jste měli mít čtyři položky v příkaze case: Add, Modify, Display a Callback.
6. Uložte změny.

OK, zkusme to ve Web prohlížeči.

7. Jděte do Web prohlížeče.
8. Napište adresu vašeho počítače, Callback a Web_CountObjects
`http://192.9.200.181/Callback/Web_CountObjects/`

Měli by jste vidět upozornění, které vám říká kolik tabulek je ve vaší databázi.

14.9.3. Parametry

Co když chcete něco zjistit o konkrétní tabulce? Rutina Web_CountObjects přijímá parametry. Jak je dodat? Je to jednoduché:

1. Jděte do Prostředí Návrháře.
2. Změňte databázovou metodu Při Web spojení následujícím způsobem:

```
❖ ...
❖ : WEB_atURL{1}="Callback")
❖ If (WEB_atURL{2}="Web_CountObjects")
❖     If (Size of array(WEB_atURL)=2) ` No optional parameters
❖         Web_CountObjects
❖     Else ` An optional parameter was included in the
URL
❖         Web_CountObjects (WEB_atURL{3})
❖     End if
❖ End if
```

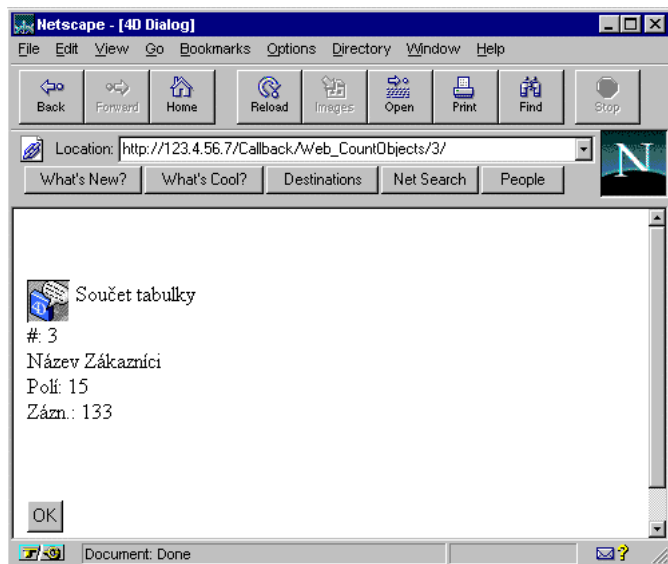
Nyní budeme testovat a uvidíme kolik položek bylo v URL a pokud tam byla nějaká navíc předáme ji metodě Web_CountObjects. Nyní si to zkusme ve Web prohlížeči

3. Uložte změny.
4. Jděte do Web prohlížeče.
5. Napište adresu vašeho počítače, Callback a Web_CountObjects
`http://192.9.200.181/Callback/Web_CountObjects/3/`

Zobrazí se upozornění která vám dá informace o tabulce 3- tabulce Zákazníci:

Používání 4D Databáze na Webu

Při Web spojení



14.9.4. Praktičtější příklad

Rutina `WEB_CountObjects` není úplně praktická. Ve skutečnosti nechcete dát uživateli všechny detaily o svém databázovém systému. Ale jak nechat uživatele upravit jeden určitý záznam.

1. Vytvořte novou metodu projektu `URL_TestValueType` importem z textového souboru `URL_TestValueTyoe` s použitím textového editoru.

```
If (False)
  ` Method: TestValueType(Longint;string)
  ` ACI University Programming Classes
  ` Created by: Kent Wilbur
  ` Date: 3/17/97

  ` Purpose: Test for legitimate value type before using in query

  ⋄f_Version6x40 := True
  ⋄fK_Wilbur := True

End if

  ` Declare Parameters
  C_BOOLEAN($0)           ` Is value valid
  C_LONGINT($1;$LFieldType) ` Field type
  C_TEXT($2;$tQueryValue)  ` Value to query

  ` Reassing parameters to local variables
  $LFieldType := $1
  $tQueryValue := $2

  $0 := False              ` Set default return value

Case of
```

Používání 4D Databáze na Webu

Při Web spojení

```
: ($LFieldType = Is LongInt)      ` Exact number type does not matter
  If (Num($tQueryValue) = 0))      ` Is the value zero
    If ($tQueryValue = "0")        ` Is the data to check actually zero
      $0 := True
    End if
  Else
    $0 := True
  End if

: ($LFieldType = Is Date)
  If (Date($tQueryValue) = !00/00/00!)) ` Is the date empty
    If ($tQueryValue = "00/00/00")      ` Is the data to check actually zero
      $0 := True
    End if
  Else
    $0 := True
  End if

: ($LFieldType = Is Time)
  If (Date($tQueryValue)=†00:00:00†)
    If ($tQueryValue="00:00:00")        ` Is the data to check actually zero
      $0 := True
    End if
  Else
    $0 := True
  End if

: ($LFieldType = Is Boolean)
  If
    (($tQueryValue="Y") | ($tQueryValue="T") | ($tQueryValue="N") | ($tQueryValue="F")
    | ($tQueryValue="Yes") | ($tQueryValue="True") | ($tQueryValue="No") |
    ($tQueryValue="False"))
    $0 := True
  End if

End case
`End of method
```

2. Vytvořte novou metodu projektu URL_BuildQuery importem z textového souboru URL_BuildQuery s použitím textového editoru.

```
If (False)
  ` Method: URL_BuildQuery(pointer;pointer;string;string)
  ` ACI University Programming Classes
  ` Created by: Kent Wilbur
  ` Date: 3/17/97

  ` Purpose: Builds a Query

  ⋄f_Version6x40 := True
  ⋄fK_Wilbur := True

End if
```

Používání 4D Databáze na Webu

Při Web spojení

```
` Declare Parameters
C_POINTER($1;$pTable) ` Table to query
C_POINTER($2;$pField) ` Field to query
C_STRING(5;$3;$sBuildPosition) ` Position in built query
C_TEXT($4;$tQueryValue) ` Value to query

` Declare Local Variables
C_LONGINT($LFieldType)

` Reassign Parameters to Locals
$pTable := $1
$pQueryField:= $2
$sBuildPosition := $3
$tQueryValue := $4

` What is the field type
$LFieldType := Type($pQueryField->)
Case of
: ($sBuildPosition="First")
Case of
: ($LFieldType=Is Alpha Field) | ($LFieldType=Is Text)
QUERY($pTable->;$pQueryField->=$tQueryValue+"@",*)

: ($LFieldType=Is Real) | ($LFieldType=Is LongInt) | ($LFieldType=Is Integer)
If (URL_TestValueType (Is LongInt;$tQueryValue))
QUERY($pTable->;$pQueryField->=Num($tQueryValue);*)
End if

: ($LFieldType=Is Date)
If (URL_TestValueType (Is Date;$tQueryValue))
QUERY($pTable->;$pQueryField->=Date($tQueryValue);*)
End if

: ($LFieldType=Is Boolean)
If (URL_TestValueType (Is Boolean;$tQueryValue))
QUERY($pTable->;$pQueryField->= ($tQueryValue<1>="T");*)
QUERY($pTable->; | ;$pQueryField->= ($tQueryValue<1>="Y");*)
End if

: ($LFieldType=Is Time)
If (URL_TestValueType (Is Time;$tQueryValue))
QUERY($pTable->;$pQueryField->=Time($tQueryValue);*)
End if
End case

: ($sBuildPosition="Last")
Case of
: ($LFieldType=Is Alpha Field) | ($LFieldType=Is Text)
QUERY($pTable->; | ;$pQueryField->=$tQueryValue+"@")

: ($LFieldType=Is Real) | ($LFieldType=Is LongInt) | ($LFieldType=Is Integer)
If (URL_TestValueType (Is LongInt;$tQueryValue))
```


Používání 4D Databáze na Webu

Při Web spojení

```
        QUERY($pTable->; | ;$pQueryField->=Num($tQueryValue))
    Else
        QUERY($pTable->)
    End if

: ($LFieldType=Is Date)
    If (URL_TestValueType (Is Date;$tQueryValue))
        QUERY($pTable->; | ;$pQueryField->=Date($tQueryValue))
    Else
        QUERY($pTable->)
    End if

: ($LFieldType=Is Boolean)
    If (URL_TestValueType (Is Boolean;$tQueryValue))
        QUERY($pTable->; $pQueryField->= ($tQueryValue≤1≥="T");*)
        QUERY($pTable->; | ;$pQueryField->= ($tQueryValue≤1≥="Y"))
    Else
        QUERY($pTable->)
    End if

: ($LFieldType=Is Time)
    If (URL_TestValueType (Is Time;$tQueryValue))
        QUERY($pTable->; | ;$pQueryField->=Time($tQueryValue))
    Else
        QUERY($pTable->)
    End if
End case

: ($sBuildPosition="Only")
Case of
: ($LFieldType=Is Alpha Field) | ($LFieldType=Is Text)
    QUERY($pTable->;$pQueryField->=$tQueryValue+"@")

: ($LFieldType=Is Real) | ($LFieldType=Is LongInt) | ($LFieldType=Is Integer)
    If (URL_TestValueType (Is LongInt;$tQueryValue))
        QUERY($pTable->;$pQueryField->=Num($tQueryValue))
    Else
        REDUCE SELECTION($pTable->;0)
    End if

: ($LFieldType=Is Date)
    If (URL_TestValueType (Is Date;$tQueryValue))
        QUERY($pTable->;$pQueryField->=Date($tQueryValue))
    Else
        REDUCE SELECTION($pTable->;0)
    End if

: ($LFieldType=Is Boolean)
    If (URL_TestValueType (Is Boolean;$tQueryValue))
        QUERY($pTable->;$pQueryField->= ($tQueryValue≤1≥="T");*)
        QUERY($pTable->; | ;$pQueryField->= ($tQueryValue≤1≥="Y"))
    Else
        REDUCE SELECTION($pTable->;0)
```

Používání 4D Databáze na Webu

Při Web spojení

```
End if

: ($LFieldType=Is Time)
  If (URL_TestValueType (Is Time;$tQueryValue))
    QUERY($pTable->; $pQueryField->=Time($tQueryValue))
  Else
    REDUCE SELECTION($pTable->;0)
  End if
End case

Else
Case of
: ($LFieldType=Is Alpha Field) | ($LFieldType=Is Text)
  QUERY($pTable->; | ;$pQueryField->=$tQueryValue+"@";*)

: ($LFieldType=Is Real) | ($LFieldType=Is LongInt) | ($LFieldType=Is Integer)
  If (URL_TestValueType (Is LongInt;$tQueryValue))
    QUERY($pTable->; | ;$pQueryField->=Num($tQueryValue);*)
  End if

: ($LFieldType=Is Date)
  If (URL_TestValueType (Is Date;$tQueryValue))
    QUERY($pTable->; | ;$pQueryField->=Date($tQueryValue);*)
  End if

: ($LFieldType=Is Boolean)
  If (URL_TestValueType (Is Boolean;$tQueryValue))
    QUERY($pTable->; $pQueryField->=($tQueryValue≤1≥="T");*)
    QUERY($pTable->; | ;$pQueryField->=($tQueryValue≤1≥="Y");*)
  End if

: ($LFieldType=Is Time)
  If (URL_TestValueType (Is Time;$tQueryValue))
    QUERY($pTable->; | ;$pQueryField->=Time($tQueryValue);*)
  End if
End case

End case
`End of method
```

3. Nahrad'te metodu projektu URL_ModifyRecords potom importujte textový soubor REV_ModifyRecords s použitím textového editoru:

```
If (False)
` URL_ModifyRecords (Table name)
` ACI University Programming Classes
` Created By: David Adams
` Date: 3/21/97
,
` Purpose: This routine modifies records in a table based
` on a call from a URL.
,
```

Používání 4D Databáze na Webu

Při Web spojení

```
` 4D Insider keyword variables:
♦fD_Adams := True
♦f_Version6x40 := True

` Modified to Query on third parameter
♦fK_Wilbur := True
End if

C_TEXT($1)

C_POINTER($pTablePointer)
C_POINTER($pFieldPointer)
C_LONGINT($i)
C_LONGINT($LTableNumber)
C_LONGINT($LRecordsInSelection)

` Turn the passed name into a table pointer.
` A Nil pointer is returned if the table is not known.
SpTablePointer:=URL_TablePointer ($1)

If (Not(Nil($pTablePointer))) ` It's a good table pointer.
  $sTableName := Table name($pTablePointer) ` Get the table's name.
  INPUT FORM($pTablePointer->,$sTableName+".iw")
  OUTPUT FORM($pTablePointer->,$sTableName+".ow")

If (Size of array(WEB_atURL)=3)
  $LTableNumber := Table ($pTablePointer)
  QUERY([zStructure];[zStructure]TableNumber = $LTableNumber ;*)
  QUERY([zStructure];&[zStructure]IncludeCustomQuery= True)
  $LRecordsInSelection := Records in selection([zStructure])
  Case of
    :($LRecordsInSelection > 2)
      SpFieldPointer := Field($LTableNumber ;[zStructure]FieldNumber)
      URL_BuildQuery($pTablePointer ; $pFieldPointer ;"First";WEB_atURL{3})
      For($i;2;$LRecordsInSelection - 1)
        NEXT RECORD([zStructure])
        SpFieldPointer := Field($LTableNumber ;[zStructure]FieldNumber)
        URL_BuildQuery($pTablePointer ; $pFieldPointer ;"";WEB_atURL{3})
      End for
      NEXT RECORD([zStructure])
      SpFieldPointer := Field($LTableNumber ;[zStructure]FieldNumber)
      URL_BuildQuery($pTablePointer ; $pFieldPointer ;"Last";WEB_atURL{3})

    :($LRecordsInSelection = 2)
      SpFieldPointer := Field($LTableNumber ;[zStructure]FieldNumber)
      URL_BuildQuery($pTablePointer ; $pFieldPointer ;"First";WEB_atURL{3})
      NEXT RECORD([zStructure])
      SpFieldPointer := Field($LTableNumber ;[zStructure]FieldNumber)
      URL_BuildQuery($pTablePointer ; $pFieldPointer ;"Last";WEB_atURL{3})

    :($LRecordsInSelection = 1)
      SpFieldPointer := Field($LTableNumber ;[zStructure]FieldNumber)
```

Používání 4D Databáze na Webu

Při Web spojení

```
URL_BuildQuery($pTablePointer ; $pFieldPointer ; "Only"; WEB_atURL{3})

Else
    QUERY($pTablePointer-> )
End case

Else
    QUERY($pTablePointer-> )
End if

MODIFY SELECTION($pTablePointer->;*)
End if
`End of method
```

4. Uložte změny.
5. Jděte do Web prohlížeče.
6. Napište adresu vašeho počítače, Callback a Web_CountObjects
<http://192.9.200.181/Modify/Customers/Blockbuster%20Video/>
<http://192.9.200.181/Modify/Customers/CA/>
<http://192.9.200.181/Modify/Customers/102/>
<http://192.9.200.181/Modify/Products/Star/>
<http://192.9.200.181/Modify/Products/19.95/>

14.9.5. Pro co se hodí a omezení tohoto systému

Metoda Při Web spojení je spuštěna v době, kdy se navazuje spojení a pak již ne. To znamená, že obdržíte pouze *počáteční* URL a ne každé URL, které je užito uvnitř jednoho konkrétního připojení. Z tohoto důvodu je zpětné volání (callback) užitečné, když zahajujete nový proces. Rovněž se v tomto kurzu dále budeme zabývat *4D automatickým callback*.

14.9.6. Důležité poznámky k použití

Struktura Case, kterou jsme použili pro zpětné volání není automatická, není vůbec generická a je použita v hlavní proceduře Při Web spojení. Pokud se rozhodnete použít tento systém, postupujte podle následujícího:

- 1) Přemístěte tento kód z metody Při Web spojení do samostatné rutiny nebo souboru rutin.
- 2) Neprogramujte tuto rutinu genericky.

Bod je dostatečně srozumitelný, vaše metoda Při Web spojení by při přítomnosti této metody byla značně nepřehledná, takže stylisticky je rozhodně přijatelnější odložit speciální kód do zvláštní metody. Výsledná rutina je přehlednější, lépe se čte a metoda Při Web spojení je zaváděna rychleji. Proč? Kterákoliv metoda, když je zaváděna musí se zavést celá, takže je rozhodně mnohem rychlejší nalézt např. čtyři 8k bloky RAM než např. 32k blok RAM.

Bod 2 zní jako kacířství a zaslouží si tedy vysvětlení. 4D nezahrnuje ukazatele na metody. Jediná generická technika pro volání metod je příkaz EXECUTE. Použití tohoto příkazu by vždy mělo být pečlivě hlídáno a o to více když je váš klient připojen přes prohlížeč Web. Proč? Příkaz EXECUTE provede cokoliv mu

Používání 4D Databáze na Webu

Při Web spojení

řeknete, takže asi není dobré nechat uživateli takovou volnost. Můžete si představit, že by uživatel provedl následující příkazy:

```
http://192.9.200.181/Callback/DELETE SELECTION([Zákazníci])/
```

```
http://192.9.200.181/Callback/QUIT 4D/
```

Nekontrolované použití EXECUTE dá uživateli příkazový řádek pro manipulaci s celou vaší databází. Uživatelé nemusí mít patřičné znalosti nebo touhu způsobit problém, ale co kdyby. Jiným důvodem pro vyloučení EXECUTE je že při nesprávném použití, nejpravděpodobněji způsobí chybu. EXECUTE se provede s tím co uživatel napíše a váš systém může zamrznout nebo vygenerovat chyby, jestliže se splete v syntaxe. Takže by to mohlo vést k přerušení vašeho systému a obtěžovat vaše uživatele. Nedělejte to. Můžete také dovolit administrátorovi kontrolovaný přístup k EXECUTE, ale ani to se nedoporučuje.

Doporučená technika pro zavedení uživatelské metody k ovládání metod je vytvořit oddělenou rutinu, která zkontroluje explicitně název metody, zkontroluje kolik parametrů, jaké a jestli vůbec byly dodány. Tato technika je bezpečná a nebude generovat chyby.

14.10. Jiné úlohy pro metodu Při Web spojení

Při Web spojení je užitečná z několika dalších důvodů. Je to správné místo k inicializaci libovolných stavových proměnných, které vaše rutiny potřebují, aby se chovali správně pro Web (toto je jedno z témat příští kapitoly). U metody Při Web spojení je také místo pro odeslání uživateli nabídky, tak jak jsme to již dělali. Ve svém systému určitě budete používat nejenom jedno uživatelské záhlaví nabídek, můžete např. použít odlišné záhlaví v závislosti na jazyku používaném uživatelem nebo v závislosti na přístupových právech. Metoda Při Web spojení vám rovněž dá příležitost odeslat uživateli HTML nebo vůbec externí soubor HTML. Tento způsob může být použit pro prezentaci stylizované domovské stránky vašeho serveru nebo k prezentaci uzpůsobeného rozhraní pro vaše uživatele.

Používání 4D Databáze na Webu

Přizpůsobení rutin pro Web

15. Přizpůsobení rutin pro Web

15.1. Úvod

HTML a prohlížeče Web nenabízejí ty samé rysy jako 4D. Již jsme viděli, že pouze omezená sada objektů 4D rozhraní, může být do HTML převedena. Mnoho 4D příkazů se chová rozdílně a některé nemohou být použity vůbec. V praxi, omezení na to co nemůžete dělat není příliš velké, ale potřebujete vědět co dělat můžete a co ne. To je předmětem této kapitoly. Rovněž se podíváme na to jak sledovat jaký typ klienta právě užívá vaše rutiny, takže je budete moci s rozlišením typů klientů uvnitř téměř celé znovu použít i pro Web. Nejdříve se podívejme na příkazy, které se chovají odlišně nebo, které by vůbec neměly být na Web použity.

15.2. Určité příkazy a Web

15.2.1. Úvod

Většina příkazů pracuje bez úprav uvnitř připojení Web. Existují však nějaké výjimky. Určité příkazy, které zahrnují zacházení se soubory nebo složité 4D Editory. Následující informace vycházejí z dokumentace 4D Jazyka. Oproti době vzniku dokumentace přibyli na Web některé další příkazy. Např. můžete použít ORDER BY (třídít) v kódu, ale nemůžete zobrazit na Web dialog Třídít. Pravděpodobně se v pozdějších verzích 4D i tento rys změní. Editor dotazů např. je již implementován a může být ve Web prohlížeči zobrazen.

15.2.2. Příkazy, které dobře pracují

Název příkazu	Komentář
ADD RECORD	Automatický překlad formulářů, podpora vícestránkových formulářů.
ALERT	Automatický převod dialogového okna.
CONFIRM	Automatický převod dialogového okna..
DIALOG	Automatický převod formulářů, podpora vícestránkových formulářů.
DISPLAY SELECTION	Automatický převod formulářů, vestavěný mechanismus stránkování.
MODIFY RECORD	Automatický převod formulářů, podpora vícestránkových formulářů.
MODIFY SELECTION	Automatický převod formulářů, vestavěný mechanismus stránkování.
QUERY	Podpora standartního dialogového okna Editoru dotazů.
QUERY BY EXAMPLE	Automatický převod formulářů, podpora vícestránkových formulářů.
Request	Automatický převod dialogového okna.

15.2.3. Příkazy, které by se měli používat opatrně

Následující příkazy se provádějí lokálně na stroji Web serveru. Např. můžete spustit tisk výběru z prohlížeče Web, avšak tisk bude proveden na počítači Web serveru. Kromě toho, když budou zahrnuty

Používání 4D Databáze na Webu

Prizpůsobení rutin pro Web

komponenty uživatelského rozhraní, objeví se na stroji Web serveru. Příkazy, které otevírají dialogy tiskárny nebo souboru jako Open document("") nebo PRINT LABELS (""), otevrou tento dialog nástroji Web serveru. Prohlížeč pak čeká dokud dialogové na stroji serveru není uzavřeno, což se obvykle nemusí stát. Bezpečné je používat tyto příkazy pokud se neobjevují dialogová okna. Můžete např. otevřít určité dokumenty, které víte že existují bez libovolného problému, že bude inicializován na akce uživatelského rozhraní.

Název příkazu	Komentář
Append document	OK, když není spuštěno žádné dialogové okno souboru.
BEEP	Bezcenné, Web server pípá.
Create document	OK, když není spuštěno žádné dialogové okno souboru.
DISPLAY RECORD	Bezcenné, nic nedělá.
EXPORT DIF	OK, když není spuštěno žádné dialogové okno souboru.
EXPORT SYLK	OK, když není spuštěno žádné dialogové okno souboru.
EXPORT TEXT	OK, když není spuštěno žádné dialogové okno souboru.
IMPORT DIF	OK, když není spuštěno žádné dialogové okno souboru.
IMPORT SYLK	OK, když není spuštěno žádné dialogové okno souboru.
IMPORT TEXT	OK, když není spuštěno žádné dialogové okno souboru.
LOAD SET	OK, když není spuštěno žádné dialogové okno souboru.
LOAD VARIABLES	OK, když není spuštěno žádné dialogové okno souboru.
MESSAGE	Zpráva se objeví na Web serveru.
Open document	OK, když není spuštěno žádné dialogové okno souboru.
Open external window	Okno se otevře na Web serveru.
Open resource file	OK, když není spuštěno žádné dialogové okno souboru.
PLAY	Bezcenné, zvuk je přehrán na Web serveru.
PRINT FORM	OK, když není spuštěno žádné dialogové okno tisku.
PRINT LABELS	OK, když není spuštěno žádné dialogové okno tisku.
PRINT RECORD	OK, když není spuštěno žádné dialogové okno tisku.
PRINT SELECTION	OK, když není spuštěno žádné dialogové okno tisku.

Používání 4D Databáze na Webu

Prizpůsobení rutin pro Web

QUIT 4D	Je podporována, můžete vzdáleně uzavřít Web server (4D nebo 4D Server).
SAVE SET	OK, když není spuštěno žádné dialogové okno souboru.
SAVE VARIABLES	OK, když není spuštěno žádné dialogové okno souboru.
SELECT LOG FILE	OK, když není spuštěno žádné dialogové okno souboru.
SET CHANNEL	OK, když není spuštěno žádné dialogové okno souboru (dokumentu)
TRACE	Okno debugger se zobrazí na Web serveru

15.2.4. Příkazy, kterým je nutné se vyhnout

Název příkazu	Komentáře
ADD DATA SEGMENT	Nevolejte tento příkaz z procesu Web připojení. Tento příkaz není navržen pro použití na Web.
ADD SUBRECORD	Nevolejte tento příkaz z procesu Web připojení. Tento příkaz není navržen pro použití na Web.
CHANGE ACCESS	Nevolejte tento příkaz z procesu Web připojení. Tento příkaz není navržen pro použití na Web.
EDIT ACCESS	Nevolejte tento příkaz z procesu Web připojení. Tento příkaz není navržen pro použití na Web. Okno hesel se objeví na stroji Web serveru. Prohlížeč čeká dokud není okno uzavřeno.
GRAPH TABLE	Nevolejte tento příkaz z procesu Web připojení. Tento příkaz není navržen pro použití na Web.
MODIFY SUBRECORD	Nevolejte tento příkaz z procesu Web připojení. Tento příkaz není navržen pro použití na Web.
Open window	Okno se otevře na stroji Web serveru.
ORDER BY	Podporováno pouze programově. Standartní dialogové okno třídění není doposud na Web podporováno.
PRINT SETTINGS	Nevolejte tento příkaz z procesu Web připojení. Dialogové okno tisku se objeví na Web serveru. Prohlížeč bude čekat dokud se dialogové okno nezavře.
REPORT	Nevolejte tento příkaz z procesu Web připojení. Okno Rychlých zpráv se objeví na Web serveru. Prohlížeč bude čekat dokud se dialogové okno nezavře.

Používání 4D Databáze na Webu

Prizpůsobení rutin pro Web

15.2.5. Doporučení

Informace v těchto tabulkách jsou užitečné a je jednoduché s nimi pracovat. Avšak nejjednodušší je používat známé příkazy, aniž bychom si museli pamatovat že pracují ve 4D a na Web pracují jinak, špatně nebo vůbec. Zde jsou některá pravidla, která by jste měli dodržovat:

- ❖ Otestujte své Web formuláře a rutiny s Web prohlížečem.
- ❖ Otestujte své Web formuláře a rutiny s Web prohlížečem.
- ❖ Otestujte své Web formuláře a rutiny s Web prohlížečem.
- ❖ Pamatujte si, že nesmíte používat Otevřít okno. To je ta nejjednodušší past, do které můžete spadnout.
- ❖ Vyvarujte se rutin tisku a manipulace se soubory, které mohou otevřít okna, která se objeví na Web server a znemožní prohlížeči pokračovat v práci.
- ❖ Nepoužívejte metody s QUIT 4D jestliže skutečně nechcete ukončit databázi. Jestliže používáte nabídku s QUIT 4D pod 4D Server, tento příkaz ukončí 4D Server a to asi nebude to co chcete. Této položce věnujte speciální pozornost, protože je zcela běžné vkládat do nabídek položku Konec. Uživatel potom aniž ví co dělá, klepne na odkaz a ukončí server.
- ❖ Pečlivě projděte kód a odstraňte všechny příkazy TRACE. Samozřejmě TRACE nemá žádný efekt v kompilované databázi.

15.3. Sledování typu klienta

15.3.1. Úvod

Neexistuje žádný důvod, aby jste svůj kód nepoužili znovu i pro Web klienty. Všechno co potřebujete je vyhnout se příkazům, které nemohou být na Web použity. Aby jste to mohli provést musíte znát typ klienta tj. že konkrétní rutinu používá prohlížeč Web. 4D neobsahuje žádnou vnitřní proceduru, která vám řekne, že daný klient je prohlížeč Web. Výborná věc ve 4D je, že i když v příkazech něco chybí můžete si takový příkaz vytvořit sami. V následujícím se chystáme přidat příkazy a funkce, které umožní určit typ klienta. Toto je část kódu, kterou můžete přidat do všech svých aplikací, které mohou být používány Web prohlížečem. Nemusíte samozřejmě používat konkrétně tyto rutiny, ale používejte rozhodně něco podobného.

15.3.2. Přidání metod

Přidejme rutiny a uvidíme co pro nás udělají.

1. Jděte do prostředí návrháře ve 4D.
2. Vytvořte novou metodu projektu GEN_SetCurrentConnectionType importem z textového souboru GEN_SetCurrentConnectionType s použitím textového editoru.

```
If (False)
  ` GEN_SetCurrentConnectionType (String (10))
  ` ACI University Programming Course
  ` Created By: David Adams
  ` Date: 3/20/97
```

Používání 4D Databáze na Webu

Přizpůsobení rutin pro Web

```
` Purpose: This routine is used to keep track of the current client connection typ
` Type values are:
`      "" (empty string)
`      "4D"
`      "Web"

` Why? This property effects the behavior of various routines, which can check
` on the current connection type with a call to the GEN_sCurrentConnectionType
` function.
`
⋄f_Version6x40:=True
⋄fD_Adams:=True
⋄fGeneric:=True
```

End if

C_STRING(10;\$1)

```
` This variable is set *only* in this function, and read *only* in
` GEN_sCurrentConnectionType
sCurrentConnectionType := $1
```

` End of method.

3. Vytvořte novou metodu projektu GEN_sCurrentConnectionType importem z textového souboru GEN_sCurrentConnectionType s použitím textového editoru.

```
If (False)
` GEN_sCurrentConnectionType () : String (10)
` ACI University Programming Course
` Created By: David Adams
` Date: 3/20/97

` Purpose: This function returns the current connection type description.
` This value is set by a call to GEN_SetCurrentConnectionType.
`
` Why use these routines instead of reading and writing the variable used to
` track this value directly? Well, this method makes it *much* easier to change
` how you track this information, and helps prevent the variable getting written
` to inappropriately.

⋄f_Version6x40:=True
⋄fD_Adams:=True
⋄fGeneric:=True
```

End if

C_STRING(10;\$0)

```
If (Undefined(sCurrentConnectionType))      ` Make sure there is something if
interpreted
sCurrentConnectionType := ""
```

Používání 4D Databáze na Webu

Prizpůsobení rutin pro Web

End if

\$0 := sCurrentConnectionType

` End of method.

4. Uložte metody.

15.3.3. O metodách

Metody GEN_SetCurrentConnectionType a GEN_SCurrentConnectionType pracují společně, aby sledovali, který typ klienta vytvořil daný proces. GEN_SetCurrentConnectionType vrátí řetězec, který popisuje typ připojení. Očekávané argumenty jsou:

Argument	Význam
4D nebo prázdný řetězec	4D nebo 4D Client
Web	Web prohlížeč

GEN_SetCurrentConnectionType uloží tuto hodnotu v proměnné a GEN_SCurrentConnectionType pro vás tuto proměnnou přečte. Ve svých rutinách nyní můžete napsat kód, který mění své chování v závislosti na typu provedeného připojení:

```
If (GEN_SCurrentConnectionType # "Web")
    $LWindowID := WIN_LNewWindow (-1;-1;Upper Centered;Plain Window)
End if
```

To znamená, že okno můžete otevřít nebo ne, užít rozdílné formuláře, potlačit dialog třídění a cokoliv dalšího potřebujete. Jestliže jste vytvořili svůj systém pomocí generických centralizovaných rutin, pak přizpůsobení vašeho systému pro Web je jednoduché. Např v tomto systému, jestliže přidáte test jako bylo právě ukázáno v rutině WIN_LnewWindow, pak rutiny, které ji volají nemusí být měněny.

15.3.4. Inicializace Typu klienta

Když se uživatel připojí prohlížečem Web, spustí se metoda Při Web spojení jak ostatně bylo řečeno dříve. To je správné místo pro registraci typu připojení. Vložte následující řádek kódu nahoru do své metody Při Web spojení:

1. Upravte metodu databáze Při Web spojení tak, aby první řádek metody byl následující:

❖ *GEN_SetCurrentConnectionType ("Web")*

2. Uložte metodu Při Web spojení.

Nyní libovolná a všechny rutiny vašeho systému, mohou volat GEN_sCurrentConnectionType tak, aby určily jestli jsou prováděny ve prospěch klienta Web.

15.3.5. Návrhy provedení

Někteří z vás asi uvažují nad tím proč nový příkaz a funkce jsou používány jenom na čtení a psaní proměnné nazvané sCurrentConnectionType. Proč pouze nečíst a psát proměnnou přímo? Samozřejmě

Používání 4D Databáze na Webu

Přizpůsobení rutin pro Web

můžete číst a psát proměnnou přímo jestliže chcete, ale je několik dobrých stylistických důvodů proč to nedělat:

- ❖ Názvy metod se čtou dobře v kódu.
- ❖ Můžete změnit metodu sledování této informace, bez změny čehokoli jiného kromě těchto dvou rutin. Jestliže se např. rozhodnete použít array k sledování několika podmínek, můžete tak jednoduše udělat, přepsáním těchto dvou metod.
- ❖ Je dobrým programátorským zvykem, izolovat místa, kde se globální proměnné (procesu nebo meziprocení) ve 4D nastavují. To znamená pro kód, že se mnohem jednodušeji udržuje, zvláště jestliže někdo jiný musí zdědit vaši práci.

Je to již dávný zvyk a vy by jste měli implementovat tuto funkci takovým způsobem, který by pro vás byl pochopitelný a komfortní. Jestliže používáte metody místo přímo proměnných mám pro vás další návrh: přidejte speciální funkce pro různé stavy tak, že se váš kód stane jasnější. Např. všechny následující funkce mohou být konstruovány jednoduše pomocí volání GEN_sCurrentConnectionType:

1. Vytvořte novou metodu projektu GEN_fThisIsAWebConnection importem z textového souboru GEN_fThisIsAWebConnection s použitím textového editoru.

```
If (False)
  ` GEN_fThisIsAWebConnection -> Boolean
  ` ACI University Programming Course
  ` Created By: David Adams
  ` Date: 3/20/97

  ` Purpose: This routine returns true if the Current connection
  ` is a WEB connection

  ⋄f_Version6x40:=True
  ⋄fD_Adams:=True
  ⋄fGeneric:=True

End if

C_BOOLEAN($0)

$0 := (GEN_sCurrentConnectionType ="Web")
`End of method
```

2. Vytvořte novou metodu projektu GEN_fThisIsNotAWebConnection importem z textového souboru GEN_fThisIsNotAWebConnection s použitím textového editoru.

```
If (False)
  ` GEN_fThisIsNotAWebConnection -> Boolean
  ` fACI University Programming Course
  ` fCreated By: David Adams
  ` fDate: 3/20/97

  ` fPurpose: This routine returns true if the Current connection
  ` fis a WEB connection
```

Používání 4D Databáze na Webu

Přizpůsobení rutin pro Web

```
    ◇f_Version6x40:=True
    ◇fD_Adams:=True
    ◇fGeneric:=True

End if

C_BOOLEAN($0)

$0 := Not(GEN_fThisIsAWebConnection )
`End of method
```

To vám umožní napsat velmi čtivý kód:

```
If (GEN_fThisIsAWebConnection )
    ` Do your Web specific code here.
Else
    ` Do your regular code here.
    $LWindowID := WIN_LNewWindow (-1;-1;Upper Centered;Plain Window)
End if
```

15.4. Zmínka o příkladu U kadeta.

U kadeta je již existující databáze. Je na platformě nezávislá a Web Aware. Podívejme se na to jak pracuje a jak je vytvořena

15.5. Příprava ACI Video pro WEB

1. Vytvořte a prostudujte následující metody projektu z odpovídajících textových souborů.

```
CUSTOMER_Initialize
GEN_ModifySelection
GEN_SetFormName
MENU_SetMenuBar
PRELIM_CustomQueryOrder
PROCESS_LUserLimit
WIN_LNewWindow
```

2. Uložte tyto metody.
3. Otevřete nabídku #5 a přidejte do horní část nabídky Web cvičení následující položky:

Položka	Metoda
Zákazníci	P_Customers
Produkty	P_Products

4. Zkuste tyto nové položky nabídky.

Používání 4D Databáze na Webu

HTML

16. HTML

16.1. Úvod

Jsme již pěkně daleko v tomto kursu a stále jsme ještě nemluvili o psaní HTML. Ve v6 nemusíte přímo programovat HTML. Ale někdy může chtít napsat HTML nebo můžete chtít zahrnout HTML dokumenty vytvořené někým jiným. Probereme si některé základy HTML a vysvětlíme jak integrovat HTML do 4D. Tento kurs není zaměřen jako úvod do HTML, ani to není kompletní příručka HTML. Na Webu je k dispozici několik výborných a bezplatných příruček a je k dispozici řada knih o HTML (HTML je dobrý systém pro dokumentaci sám o sobě). Současná verze HTML je 3.2., ale většina prohlížečů podporuje doposud HTML 2.0 nebo 3.0.

16.2. HTML: Historie a Účel

16.2.2. Cíle HTML

Tim Berners-Lee – švýcarský fyzik – více či méně vymyslel Web. Navrhl HTTP a HTML, HyperText jako mechanismus pro vytváření hypertextových výzkumných zpráv. V tomto schématu v zápatí ve výzkumné zprávě se stalo klepnutelným odkazem na zdroj původní citace. Účel HTML je integrovat data z různých zdrojů a typu médií do jednoho strukturovaného dokumentu. Zde je vyjádření Tim Berners-Lee v definici HTML (RFC 1866).

The HyperText Markup Language (HTML) je jednoduchý jazyk označování, užívaný k vytváření hypertextových dokumentů, které jsou na platformě nezávislé. HTML dokumenty jsou určeny pro představení informací z domén různého původu. HTML označení může reprezentovat hypertextové zprávy, poštu, dokumentaci a hypermedia; nabídky možností, výsledky databázových dotazů, jednoduché strukturované dokumenty s čárovou grafikou a hypertextovou reprezentaci dalších informací.

16.2.3. SGML

HTML je něco jako podmnožina SGML (Standard Generalized Markup Language). SGML bylo vytvořeno IBM jako obecný způsob strukturování jejich dokumentace. IBM určila tento systém jako standart a je v určitých oblastech široce zaveden. SGML je podstatně složitější a účinnější než HTML. SGML je dostatečně pružné k definování typu dokumentů, které mají složité chování a pravidla. Definiční dokument SGML je nazýván Document Type Definition nebo DTD. Technicky všechny HTML dokumenty jsou výskytem jednoho SGML DTD (existují stovky různých DTD). Z tohoto důvodu někdy uvidíte v HTML dokumenty, které začínají následujícím řádkem:

```
<!doctype html public "-//IETF//DTD HTML//EN">
```

To je SGML deklarace DTD, která definuje chování dokumentu. V každém případě můžeme očekávat že o DTD a SGML uslyšíme ještě v budoucnu. Dále již přímo o HTML.

16.3. HTML: Co to není

HTML není programovací jazyk. Je to řada kódů, která je interpretována prohlížečem. Tagy jsou určeny tak, aby nezávisely na platformě. Dokonce nepředpokládají, že čtenář může vidět grafiku nebo má instalované jakékoliv určité písmo (grafika např. může být automaticky nahrazena názvy v pouze textových prohlížečích).

Používání 4D Databáze na Webu

HTML

HTML jsou určeny pro představení strukturované dokumentace, jak byly původně výzkumné zprávy. Jejich struktura je užívána několika způsoby. Včetně automatického indexování HTML není typograficky orientovaný jazyk. Styly jsou reprezentovány odlišným způsobem v odlišných prohlížečích na různých platformách. HTML má velmi málo společného se systémy grafických studií, protože jeho účel je velmi odlišný. Adobe PDF (Acrobat) formát je typograficky orientovaná alternativa k HTML. Kaskádové styly (CSS1 a CSS2) jsou poměrně nové rysy HTML 3.2., které vylepšují schopnosti HTML, ale nejsou ještě příliš rozšířeny.

16.4. Vytváření HTML

16.4.1. Automatické převody ve 4D

Je několik způsobů k vytvoření HTML. Využití 4D k automatickému vytváření HTML je nejjednodušší. Jestliže někdy skončíte tak, že budete psát HTML ručně je v každém případě dobré začít příklady HTML, která za vás napíše 4D. Zopakujme si jak to udělat se stránkou Web nabídky:

1. Ujistěte se že vaše 4D Web server databáze je zapnuta.
2. Ujistěte se, že máte zapnutý Web prohlížeč.
3. Přejděte do prohlížeče Web.
4. Připojte se k vaší databázi.
5. 4D pošle zpátky stránku s vaší Web nabídkou.
6. Vyberte View → Document Source v Netscape.
Vyberte View → Source v Internet Explorer.
7. Objeví se stránka HTML generovaná 4D.

Nyní můžete tento text uložit a upravovat jak budete chtít.

16.4.2. Editor HTML

HTML Editory jsou čím dál lepší. Poslední verze Adobe PageMil a Claris HomePage, Microsoft FrontPage jsou plny nových rysů a poměrně příjemné. Ještě před rokem bych řekl, aby jste pro vaše HTML stránky použili Textový editor. Dnes doporučuji aby jste použili poslední HTML Editor a Textový editor používejte jen v nejnnutnějších případech. Moderní HTML Editory konečně zredukovali chyby obsažené v ručně kódovaných HTML.

16.4.3. Textový editor

Po tom všem, co bylo řečeno, Textový editor může stále být prospěšný. HTML editory nemusí zahrnovat všechny rysy, které potřebujete, nebo je nemusí dovolovat a nebo nedovolí do určitých tagů přidat všechny vlastnosti, které potřebujete. Není špatné rozumět HTML, pokud chcete v případě potřeby upravovat HTML přímo. A jenom tak můžete nalézt co se stalo, když některé z vašich stránek nevytvořených ve 4D nepracují správně.

16.4.4. 4D: Vlastní výroba stránek

Jestliže zjistíte, že potřebujete množství statických stránek správně produkovaných a reprodukováných. 4D je skvělým prostředím pro tvorbu stránek a údržbu umístění. Vysoce strukturované tagy HTML pracují dobře s nástroji řízení 4D pro strukturovaná data. HTML jsou pěkně ukecané a skutečně vhodné pro automatizaci produkce.

Používání 4D Databáze na Webu

HTML

Nyní se podívejme na nějaké HTML.

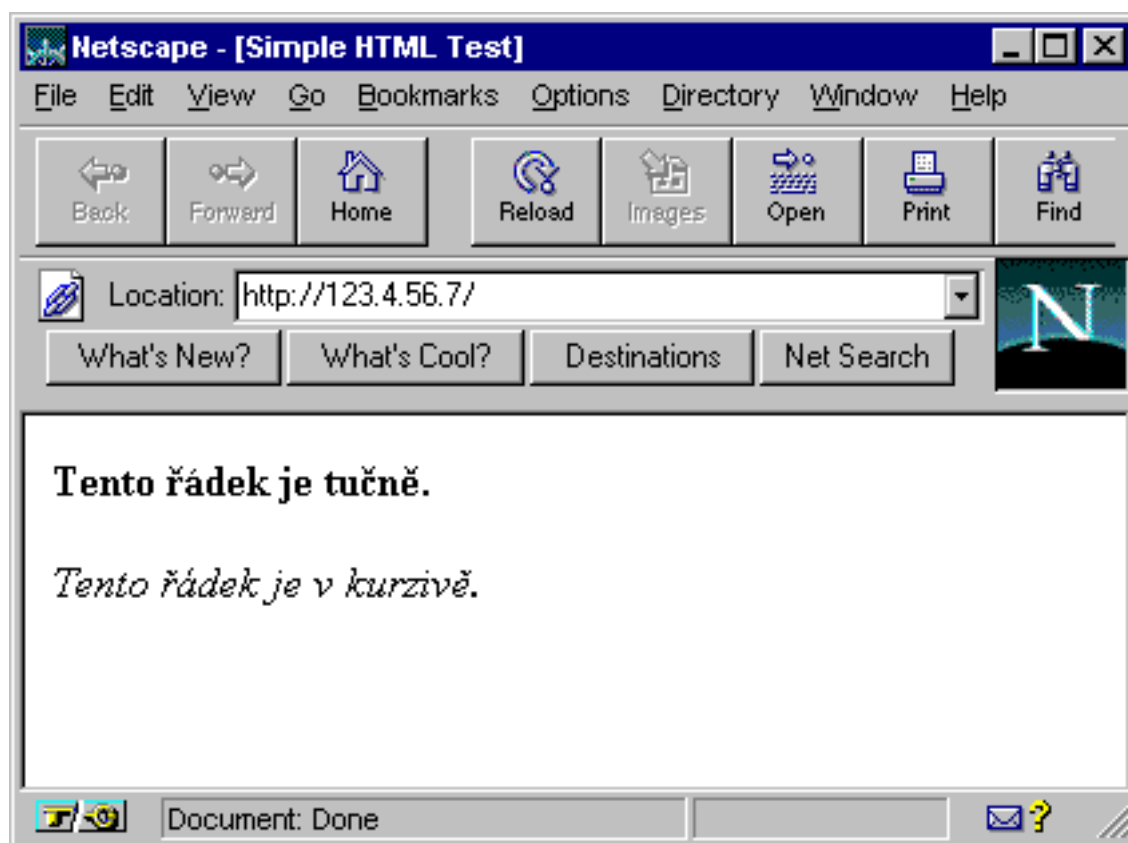
16.5. Syntaxe HTML

HTML tagy jsou vytvořené textově pro interpretaci v prohlížečích, ale jsou pochopitelné i pro člověka. Některé tagy ošetřují formální příkazy, které říkají prohlížeči jak interpretovat následující textový blok nebo instruují jak indexovat dokument.

HTML syntaxe je obecně zcela jednoduchá, obvykle je vyjádřena v podobném kódu:

```
<TAG VLASTNOSTI=Hodnota>Text ovlivně tagem.</TAG>
```

Zde je jednoduchý příklad:



Prohlédněte si dokument "Simple.html" ve vašich školících materiálech.

```
<HTML>
<HEAD>
<TITLE>Simple HTML Test</TITLE>
</HEAD>

<BODY>
<B>Tento řádek je tučně.</B><P>
<I>Tento řádek je v kurzivě.</I>
```


Používání 4D Databáze na Webu

HTML

</BODY>
</HTML>

Většina tagů je ohraničena počátečním a koncovým tagem – jako ty které jsou ukázány výše - , který určuje kde styl nebo vlastnost začíná a končí

Prohlédněte si dokument “HTML_Tags.html” z vašich školících materiálů.

16.6. Tagy znakových stylů

Styly jednotlivých znaků a textových řetězců mohou být určovány řadou tagů. Pamatujte si, že tyto tagy jsou interpretovány různě v různých prostředích. Některé z názvů tagů jako “strong” a “emphasis” nám připomínají, že tagy nejsou absolutním indikátorem konečného vzhledu textu, ale pouze jejich relativního vzhledu k dalšímu textu.

	Tučně	
<I>	Kurziva	</I>
	Zvyýraznění textu	
	Strong je často zobrazen jako Tučně	
<!--	Řetězce	-->

16.7. Tagy odstavů

Několik tagů je používáno pro formátování a rozeznávání odstavců. Protože HTML nemá žádný koncept nebo “pravítko” nebo “ohraničení” jsou tyto tagy jedny z nejužitečnějších:

<P>	Vloží prázdný řádek	
 	Konec řádku	
<HR>	Přidává horizontální pravítko	
<PRE>	Formát na písmo se stejnou délkou znaků	</PRE>
<BLOCKQUOTE>	Vloží blok	</BLOCKQUOTE>

16.8. Tagy dokumentů

Některé HTML tagy slouží jako formální označení dokumentu nebo sekce a nebo poskytují strukturu dokumentu.

<HTML>	Značí celý dokument	</HTML>
<HEAD>	Značí hlavičku dokumentu.	</HEAD>
<TITLE>	Značí nadpis uvnitř hlavičky.	</TITLE>

Používání 4D Databáze na Webu

HTML

<BODY>

Značí tělo dokumentu.

</BODY>

HTML nemá SGML section tag, ale to se může změnit. Nyní máme pořadové seznamy a šest úrovní hlaviček.

16.9. Tagy obrázků

Obrázky jsou pouze jiné externí zdroje HTML, jiný druh kotvy. V nejjednodušším typu reference obrázku užívá tuto syntaxu:

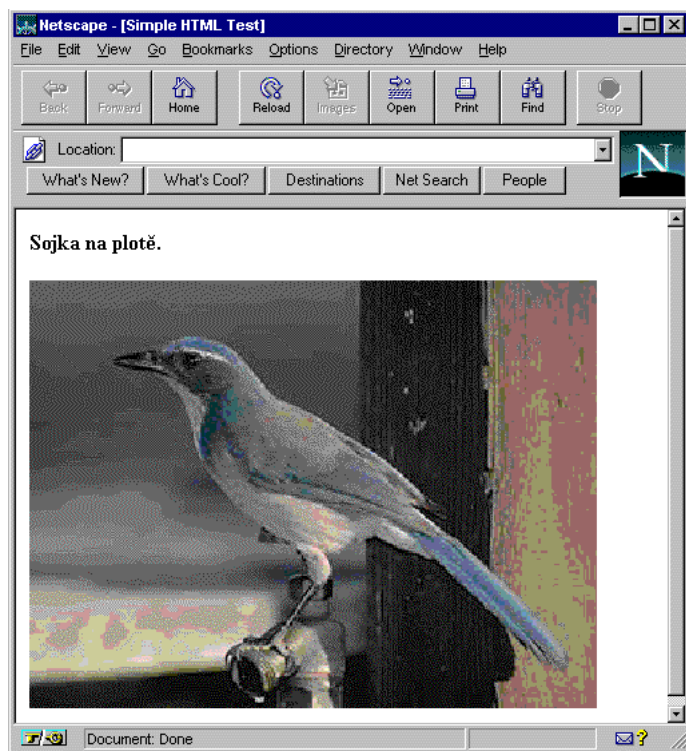
```
<IMG SRC="Path/to/image/file.gif">
```

Nemáme žádný závěrečný tag pro obrázky, IMG tag je zcela osamocen. Tento tag má několik výběrových vlastností včetně ALIGN, HEIGHT, WIDTH a ALT. Do svého obrázku vždy zahrňte tag ALT. ALT znamená “alternate” a zobrazuje textovou zprávu, když samotný obrázek nemůže být zobrazen. Je několik důvodů proč obrázek nemůže být nalezen:

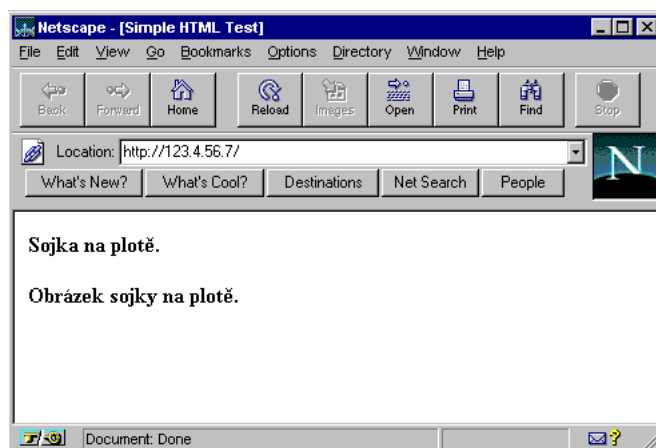
- ❖ Cesta SRC je špatná a to z jakéhokoli důvodu.
- ❖ Uživatel vypnul grafiku k zlepšení chování.
- ❖ Uživatel používá pouze textové prohlížeče jako je Lynx.
- ❖ Uživatelé používají software pro čtení textu, protože jsou zrakově postižení. Vážně zrakově postižení uživatelé nejsou na Internetu neobvyklou záležitostí. Pro převážně textovou zaměřenost el. pošty, pracovních skupin a samotného Web. Čtečky textu dovolí těmto lidem komunikovat jednoduše uvnitř své komunity přes počítače.

```
<IMG SRC="scrubjay.jpg" ALT="Obrázek sojky na plotě.">
```

Používání 4D Databáze na Webu HTML



Správně zobrazený obrázek. Tag ALT není vidět.



ALT tag je použit, když není zobrazen obrázek.

16.10. Tagy tabulek

Tabulky jsou vysoce účinný způsob pro organizaci a prezentaci dat. V naší společnosti jsou lidé zvyklí většinu informací znázorňovat v tabulkové formě. Databázové informace se v tabulkové formě reprezentují snadno, protože tabulky jsou vnitřní přirozeností databáze. Již jste viděli jak 4D využije HTML tabulky pro zobrazení skupiny záznamů k napodobení výstupního seznamu ve 4D. Základní tagy tabulek jsou:

<TABLE>

Značí celou tabulku

</TABLE>

Používání 4D Databáze na Webu

HTML

<CAPTION>	Legenda tabulky, jako část tabulky	</CAPTION>
<TH>	Hlavička tabulky, zvláštní druh buňky	</TH>
<TR>	Řádek tabulky	</TR>
<TD>	Data tabulky, buňka	</TD>

Tyto tagy nejsou podporovány ve starších prohlížečích, ale nalezneme je v Netscape 2.0. a vyšší a Internet Explorer 3.0. a vyšší.

HTML jsou vždy ukecané a v případě tabulek je tato charakteristika jazyka obzvláště platná. Tabulky jsou ideální místo pro automatizaci. Po ručním kódování několika tabulek, určitě oceníte automatické konstruování tabulek ve 4D.

Poznámka: Jestliže chcete vytvářet HTML tabulky mimo 4D, procedurálně viz. ACI US Technical Note #96-14 "Converting Arrays to HTML Tables".

16.11. URL

16.11.1. Standardní syntaxe

Uniform Resource Locators — URL — jsou tahounem, který posunuje Web dopředu. Web byl vytvořen k podpoře několika existujících zdrojových typů a aby měl dostatek prostoru pro další neočekávané datové typy. Existuje standardní syntaxe pro URL a několik důležitých odlišností v závislosti na zdrojovém typu zdůrazníme.

Zcela základní úroveň URL se skládá ze dvou částí::

<schéma>:<schéma-specifikace-část>

Schéma obsahuje typ služby a schéma specifikace část obsahuje instrukce příslušné k tomuto schématu. Zde uvádím několi příkladů platných URL:

`http://www.bigcompany.com/We_Are_Huge.html`

`ftp://server1@bigcompany.com/pubs/bigfile.hqx`

`mailto:The_Boss@BigCompany.com`

Jak můžete vidět syntaxe se liší v závislosti na schématu. Za rámec toho co zde bylo ukázáno, některá schemata jako FTP dovolí zadat jméno uživatele a heslo jako část URL (pro doplnění viz RFC 1738 nebo příručky HTML).

16.11.2. Poznámka k portům

Mnoho schémat včetně HTTP, dovolí zadat číslo portu jako část hostitelovi adresy následujícím způsobem:

<schema>://<server>:<port>/<cesta>

`http://www.bigcompany.com:80/Home.html`

Používání 4D Databáze na Webu

HTML

Jestliže nezahrnete číslo portu do URL, prohlížeč kontaktuje pro toto schéma tzv. “dobře známý port”. O těchto portech jsme již hovořili v části o konfigurování 4D na Web. Pro URL různých typů nemusíte zadávat číslo portu, jestliže daná služba je provozována na dobře známém portu. Jinými slovy, jestliže zadáváte URL pro FTP soubor potřebujete zadat číslo portu pouze tehdy jestliže FTP server běží na jiném portu než 23. Zde vidíme tabulku dobře známých čísel portu v nejběžněji užívaných schemech Internetu:

Služba	Popis	Port
FTP	File Transfer protocol	21
Telnet	Interactive terminal session	23
SMTP	Simple Mail Transfer Protocol	25
whois	Lookup DNS record	43
gopher	The Gopher protocol	70
finger	finger	79
HTTP	HyperText Transfer Protocol	80
NNTP	USENET news using NNTP access	119
NTP	Network Time Protocol	123

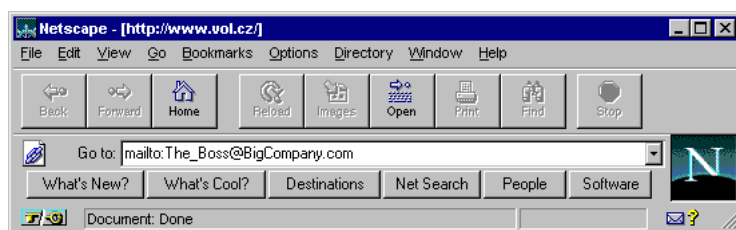
16.11.3. URL v HTML

Syntaxe URL jsme již viděli:

`<schema>://<server>:<port>/<cesta>`

je dostačující, když píšete přímo do pole adres vašeho prohlížeče.

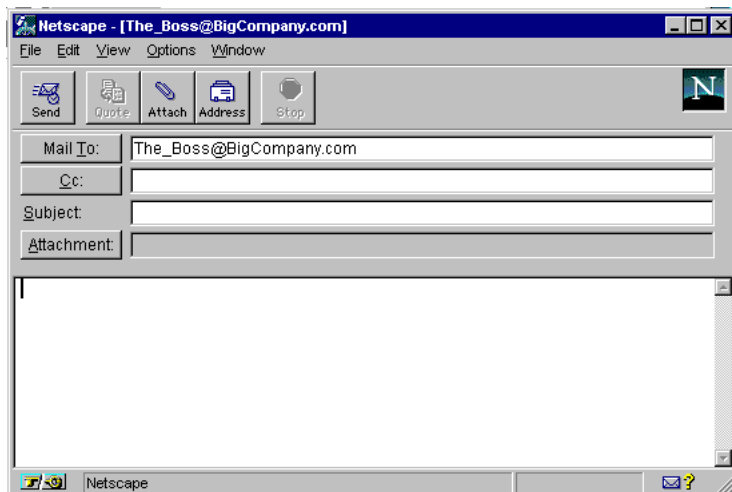
1. Přejděte do prohlížeče Web.
2. Napište: The_Boss@BigCompany.com do pole adresy vašeho prohlížeče.



3. Objeví se okno pošty ve vašem prohlížeči.

Používání 4D Databáze na Webu

HTML



4. Jestliže nemáte v prohlížeči nastavenou e-mail adresu bude generována chyba. Pokud bude generována chyby zkuste nakonfigurovat své poštovní služby a opakujte bod 3. Vaše e-mail adresa je např. v Netscape na tomto místě:

Option → Mail and News Preferences... Identity page

Pole adres vašeho prohlížeče je dobrým místem pro testování, zda je daná URL platná, správně formátovaná, ale není to dostatečné pro URL vnořené ve vašich HTML dokumentech. Abyste vnořili URL do HTML musíte použít tag kotvy `<A>` a ``. Tento tag má dvě komponenty: reference a štítek. Štítek je to co se objeví uživateli jako aktivní odkaz a reference je to na co daný odkaz odkazuje:

`Text štítku na který uživatel klepne.`

Zde jsou tři příklady:

`Visit our home page.`

`Download the file you need.`

`E-mail the boss`

Prohlédněte si dokument "URL_Examples.html" ve vašich materiálech.

16.11.4. Výhody ovládání URL pomocí 4D

Je jednoduché napsat špatnou URL, zvláště když ji zahrnujete dovnitř stránky HTML. Musíte umístit uvozovky na správné místo, nezapomenout správné vlastnosti a tagy. Nejednodušší je napsat URL s pomocí HTML editoru. Existují rovněž výhody pro psaní URL pomocí 4D databáze:

- ❖ Můžete automatizovat jinak nudnou ruční konstrukci URL.
- ❖ Můžete použít externí příkazy k periodickému ověřování URL např. pomocí ITK, můžete rychle ověřit jestli katalog URL stále ukazuje na existující adresy.
- ❖ Můžete vytvořit rozhraní, které zabrání uživateli vytvářet neplatné URL jestliže máte databázi obsahující odkazy.

Používání 4D Databáze na Webu

HTML

- ❖ Jestliže se rozhodnete upravit nebo obohatit způsob zobrazení URL potřebujete změnu pouze v jednom místě kódu.
- ❖ Jestliže je tatáž URL používána na několika stránkách, je příjemné když máte skutečnou URL uloženou v jednom záznamu.

Následující řádky kódu uvádějí jednoduchou rutinu pro vytváření URL FTP:

```
If (False)
  ` Method: URL_Make_FTP_URL (URL;Label) —» HTML block
  ` $0 Text Resulting HTML statement.
  ` $1 Text FTP address
  ` $2 Text {Label for FTP address. Uses $1 if $2 not found.}
  `
  ` 4D Insider keyword variables:
  ⋄fD_Adams:=True
  ⋄f_Version6x40:=True
  ⋄fGeneric:=True

End if

C_TEXT($0;$Result)
C_TEXT($1;$URL)
C_TEXT($2;$Label)

C_LONGINT($Length)
C_STRING(1;$Quote)

$URL := $1

$Result:=""
$Quote := Char(34) ` Literal quotation mark
$Length := Length($URL)

If ($Length>0)                                ` Don't bother if parameter is empty...

  If ($Length< 7)
    $URL := "ftp://" + $URL
  End if

  If ($URL # "ftp://@")
    $URL := "ftp://" + $URL
  End if

  If (Count parameters>1)
    $Label:=$2
  Else
    $Label:=$URL
  End if

  ` E.g: <A HREF="ftp://server1@big.com/pubs/bigfile.hqx">FTP</A>
  $Result:="<A HREF="+$Quote+$URL+$Quote+"/>"+$Label+"</A>"
```

Používání 4D Databáze na Webu

HTML

```
End if ` ($Length>0)

` ...in case user input some //'s into the FTP name...
$Result:=Replace string($Result,"//","/")

$0 := $Result
```

Poznámka: *Tuto metodu lze nalézt ve vašich materiálech..*

Vytváření správně formátované FTP URL je nyní takto jednoduché:

```
$My_FTP_URL:=URL_Make_FTP_URL ([Links]Reference:[Links]Label)
```

16.12. Formuláře

HTML obsahuje tagy, které vám umožní vytvořit pro uživatele zadávací formuláře. Tyto formuláře mohou obsahovat vstupní pole typů integer nebo text, posuvné textové oblasti, výběrové nabídky, přepínače, zaškrtačací okénka a seznamy. Formuláře jsou normálně používány k odeslání dat na serveru pro zpracování pomocí CGI. V našem případě, žádná vrstva CGI neexistuje, zasíláme formuláře přímo do 4D. Dále si uvedeme popis jak jsou formuláře definovány:

FORMULÁŘE (METODY, AKCE)

Tag formuláře zachycuje celý formulář. Formuláře nemůžete vnořovat, ale můžete mít mnoho formulářů na téže stránce. Libovolný formulář generovaný 4D bude vždy první formulář na stránce, a to i tehdy jestliže vnoříte do formuláře 4D další formuláře. Do formulářů Netscape mohou být zahrnuty tabulky, ale jejich chování se liší v závislosti na tom co do nich vložíte.

Metoda Může být použit libovolný podporovaný typ, typicky POST nebo GET. GET zahrne všechny informace odesílané do 4D do URL. URL může obsahovat informaci do 1K (1024 znaků). Formuláře odesílané pomocí GET budou prověřovány vámi napsanou metodou prověření URL v metodě Při Web spojení. 4D normálně očekává, že uživatelské formuláře budou odeslány pomocí POST. POST může obsahovat všechna pole formuláře včetně textových oblastí. Aby POST fungoval správně, HTML dokument by měl být nejdříve ošetřen ve 4D.

Akce Akce je normálně cesta k CGI. V našem případě je to typicky název metody, kterou chcete ve 4D provést.

Zde vidíme fragment formuláře, který odesílá žádost o akci prostřednictvím systému zákaznického zpětného volání, které jsme vytvořili dříve:

```
<FORM NAME="myForm" METHOD="GET"
ACTION="/Callback/Web_CountObjects/3/">
<!-- Input fields, buttons and so on get defined here. -->
</FORM>
```

Zde vidíme fragment formuláře, který říká 4D, aby automaticky provedla zmíněnou metodu (pracuje automaticky tehdy, když je originální stránka ošetřena ze 4D):

Používání 4D Databáze na Webu

HTML

```
<FORM NAME="myForm" METHOD="POST" ACTION="/4DMETHOD/methodName">
<!-- Input fields, buttons and so on get defined here. -->
</FORM>
```

Samotné tagy formulářů jsou dostatečně jednoduché. To co nás zajímá jsou objekty uvnitř formulářů. Podívejme se na vše co můžete umístit do formuláře a pak si vytvoříme ručně formulář, který interaguje se 4D.

1. Přejděte do prohlížeče Web.
2. Otevřete soubor Form_objects.htm z vašich materiálů.


Můžete vidět výběrové nabídky, přepínače, posuvné oblasti, tlačítka, zaškrťovací okéna a několik druhů posuvných oblastí. Poznamenejme, že speciální typ vstupu “hesla” vytváří vstupní oblasti, které nezobrazují na obrazovku to co uživatel píše.

1. Otevřete Form_objects.htm ve vašem textovém editoru (můžete ji současně nechat otevřenu prohlížečem).

Tato HTML poskytuje dobrou představu o tom jak nastavovat každý z těchto objektů formuláře. Až budete po absolvování tohoto kurzu vytvářet uživatelské formuláře HTML, můžete tento soubor použít jako příklad pracujících objektů. Zde máme shrnutí toho co můžete dělat. Všechny tagy a ilustrace v této tabulce jsou převzaty z dokumentu, na který se díváte.

Používání 4D Databáze na Webu

HTML

Vstupní typ	Příklad	Netscape
text	<code><Input type=text name="myText" value=""></code>	<input type="text"/>
hesla	<code><Input type=password name="myPassword" value="Password"></code>	<input type="password"/>
int	<code><Input type=INT name="myInteger" value="3"></code>	<input type="text" value="3"/>
přepínač	<code><Input type=radio name=rb_sex value=Male checked>Male
</code> <code><Input type=radio name=rb_sex value=Female">Female
</code> <code><Input type=radio name=rb_sex value=None>None of the above
</code>	<input checked="" type="radio"/> Male <input type="radio"/> Female <input type="radio"/> None of the above
Zaškrtávací políčko	<code><Input type=checkbox name=cb_longForm value="yes" checked>Include details</code>	<input checked="" type="checkbox"/> Include details
skrytý	<code><Input type=hidden name="myHiddenValue" value="Whatever"></code>	
odeslat	<code><Input type=SUBMIT value="Search"></code>	<input type="button" value="Search"/>
storno	<code><Input type=RESET value="Start Over"></code>	<input type="button" value="Start Over"/>
grafika	<code><Input type=IMAGE name="done" SRC="Submit.gif" BORDER=0></code>	

Používání 4D Databáze na Webu

HTML

Struktura

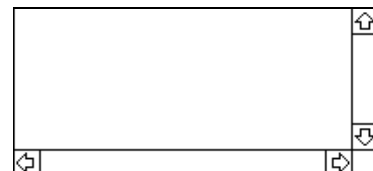
Příklad

Netscape

TEXTOVÁ OBLAST

```
<TEXTAREA rows=6 cols=30 wrap="soft">

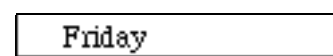
</TEXTAREA>
```



SEZNAM

velikost = 1

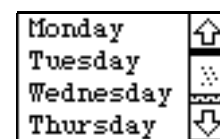
```
<SELECT name=myPopup size=1>
<OPTION>Monday
<OPTION>Tuesday
<OPTION>Wednesday
<OPTION>Thursday
<OPTION SELECTED>Friday
</SELECT>
```



SEZNAM

velikost > 1

```
<SELECT name=myScrollableArea size=4>
<OPTION>Monday
<OPTION>Tuesday
<OPTION>Wednesday
<OPTION>Thursday
<OPTION SELECTED>Friday
</SELECT>
```



Jak jste viděli máme pouze tři způsoby vytváření objektů formulářů:

Struktura

Objekty

INPUT

Vstupní oblasti a tlačítka všech druhů.

TEXTAREA

Posuvné textové oblasti.

SELECT

Výběry a jejich posuvné oblasti.

Zde máme podrobnější definici těchto struktur.

INPUT (NÁZEV, TYP, HODNOTA, VELIKOST, MAX.DÉLKA, VYBRÁNO)

Název

Odpovídá názvu proměnné. Názvy polí by měli být jedinečné s výjimkou skupin přepínačů, které musí mít stejné jméno a různé hodnoty.

Typ

Platné vstupní typy jsou text, password, hidden, int, radio, checkbox, submit, reset, image a file. Hidden vytváří pole, které nevidí uživatel, ale 4D ano. To je dobrý způsob jak odesílat zpět zvláštní hodnoty. V samotné 4D není v současné době podporován File (pouze pomocí přídatný modulů).

Hodnota

Můžete naplnit pole výchozí hodnotou. To je zvlášť užitečné jestliže vytváříte formulář za běhu aplikace jako odpověď na předchozí odeslaný formulář.

Používání 4D Databáze na Webu

HTML

<i>Velikost</i>	Toto je charakteristika zobrazení a měla by být spíše nazvána “šířka”.
<i>Max.délka</i>	Tato vlastnost omezuje počet znaků, které lze vložit do pole.
<i>Vybráno</i>	Zahrnuje slovo CHECKED jestliže chcete, aby výběrový vstupní typ (přepínač a zaškrtnutí políčko) byl vybrán při zobrazení formuláře.

TEXTAREA (NÁZEV, ŘADY, SLOUPCE, ZOBRAZENÍ)

Tag <TEXTAREA> je ukončen </TEXTAREA> tagem. Jestliže chcete naplnit pole výchozí hodnotou, vložte ji mezi tyto dva tagy. Je to odlišný mechanismus pro dosažení téhož efektu jako při vytváření INPUT pomocí VALUE.

<i>Název</i>	Totéž jako pro INPUT.
<i>Řady</i>	Výška posuvné textové oblasti.
<i>Sloupce</i>	Šířka posuvné textové oblasti
<i>Zobrazení</i>	Co řídí způsob zobrazení a posuvníky v prohlížečích Navigator 2.0. a vyšší. Možné hodnoty jsou OFF (výchozí), SOFT a HARD.

SELECT (NÁZEV, VELIKOST, VÍCE)

<SELECT>...</SELECT> struktura značí řadu výrazů, které definují nabídku nebo posuvnou oblast. Řada prvků OPTION definuje hodnoty polí.

<i>Název</i>	Totéž jak pro INPUT.
<i>Velikost</i>	Tato vlastnost řídí vykreslení struktury. Jestliže je velikost 1 je zobrazena výběrová nabídka/seznam. Jestliže je větší než 1 je zobrazena posuvná oblast o velikosti prvků uvedených v hodnotou <i>velikost</i> .
<i>Více</i>	Zahrnuje slovo MULTIPLE, jestliže chcete, aby šlo v posuvné oblasti vybrat více jak jeden prvek.

OPTION (VYBRANÉ)

OPTION je užívána uvnitř struktury SELECT a předchází hodnotu. Nemá uzavírací tag.

<i>Vybrané</i>	Znamená vyber tento prvek v nabídce nebo posuvné oblasti.
----------------	---

16.13. Odesílání formulářů do 4D

Nyní, protože máte již informace o formulářích HTML, podívejme se jak vytvořit uživatelský formulář HTML, který volá 4D.

16.13.1. Použití formuláře HTML na volání 4D

1. Otevřete ve vašem textovém editoru nový dokument.

Používání 4D Databáze na Webu

HTML

2. Vyvolejte si dokument Call4D.htm a uložte jej do stejné složky, ve které máte vaši databázi (ne do složky školících materiálů).
3. Vložte následující text (pouze místo adresy počítače zde uvedené vložte adresu svého počítače).

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Call 4D</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
<!-- NOTE: Put your machine address in place of 192.9.200.181 -->
```

```
<FORM NAME="call4DForm" METHOD="POST"
```

```
ACTION="http://192.9.200.181/Callback/Web_CountObjects/3/">
```

```
<INPUT TYPE=submit VALUE="Call 4D!">
```

```
</FORM>
```

```
</BODY>
```

```
</HTML>
```

Tento formulář má jedno tlačítko a nedělá nic jiného než, že volá metodu Web_CountObjects , kterou jsme přidali již dříve.

4. Přejděte do svého prohlížeče.
5. Otevřete soubor Call4D.htm.
6. Měli by jste vidět obrazovku s jedním tlačítkem "Call4D".
7. Klepněte na tlačítko.
8. Měli by jste vidět upozornění, které popisuje tabulku Zákazníci.

Právě jste viděli, že je možné volat metodu 4D ze souboru HTML otevřenou přímo 4D prohlížečem. Normálně to samozřejmě nebudete chtít provádět. Pokaždé, když se připojíte na 4D tímto způsobem, použijete jiný proces Web připojení (licenci). Jestliže chcete využít výhody 4D systému řízení stavu měli by být vaše formuláře obsluhovány ze 4D a ne pomocí jiného Web serveru nebo ze souborů.

16.13.2. Sledování Web připojení

1. Ujistěte se, že máte viditelný seznam procesů 4D.
2. Jděte do svého prohlížeče Web.
3. Otevřete z vašeho disku soubor Call4D.htm.
4. Klepněte na tlačítko ve stránce.
5. Klepněte na "zpět" ve vašem prohlížeči a přepněte se tak zpět na původní soubor Call4D.htm
6. Minimálně pětkrát opakujte kroky 4 a 5.
7. Všechna vaše Web připojení jsou vyčerpána!
8. Přejděte do Prostředí návrháře ve 4D a odstraňte všechny procesy Web připojení.

Používání 4D Databáze na Webu

HTML

16.13.3. Rozšíření formuláře, získání více automatických akcí

1. Rozšířte váš formulář HTML Call4D.htm k provedení následujících akcí pomocí tlačítek:

Přidat Produkty

Přidat Zákazníky

Dotaz na Produkty

16.14. Automatické provedení metody

Jiným důvodem pro ošetřování vašich zákaznických formulářů z 4D je využití výhod automatického vázání proměnných a systému zpětného volání metod. 4D zahrnuje automatické provádění metod založené na URL bez speciální práce na procedurálním prověřování URL. Tento rys očekává, že formuláře jsou obsluhovány z 4D a odesílány pomocí metody formuláře POST. Předtím, než budeme moci využít automatický systém 4D Zpětné volání procedury, potřebujeme se dozvědět více o obsluze stránek či o vnořování HTML ze 4D.

Používání 4D Databáze na Webu

Odesílání HTML

17. Odesílání HTML

17.1. SEND HTML FILE

17.1.1. Odesílání souboru

4D zahrnuje jednoduchý způsob obsluhování souborů HTML, tak jak regulární Web server. 4D nebo 4D Server může obsluhovat dokumenty nalezené na tomtéž stroji, na kterém běží 4D aplikace. Příkaz, který toto zajišťuje je SEND HTML FILE. Výchozí nastavení 4D je takové, že HTML soubory jsou hledány ve složce, kde je umístěna struktura databáze. Do HTML můžete zahrnout cokoliv potřebujete včetně odkazu na obrázky. Vyzkoušejme si to:

1. V textovém editoru vytvořte nový dokument.
2. Nazvěte tento dokument Home.htm a uložte jej do složky k databázové struktuře.
3. Napište do dokumentu následující HTML:

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Welcome!</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
Dobrý den a vítějte na naší webovské stránce. Stránka se právě upravuje, ale brzy nás opět navštívte!<P>
```

```
</BODY>
```

```
</HTML>
```

4. Otevřete metodu Při Web spojení.
5. Na konci metody přidejte následující řádek kódu nad řádek, kde je zasíláno záhlaví nabídek:

```
SEND HTML FILE ("Home.htm")
```

6. Uložte změny v metodě Při Web spojení.
7. Ujistěte se, že je viditelný seznam procesů.
8. Přejděte do prohlížeče.
9. Otevřete nové spojení s vaší databází.
10. Nové Web připojení se naskartuje.
11. Objeví se stránka [Home.htm](#) zaslaná 4D.
12. Všimněte si, že na ní nejsou žádná tlačítka nebo odkazy.
13. Stiskněte opět enter.
14. Nastartuje se další nový proces a stránka se znovu zobrazí.

Něco nám zde chybí...

Používání 4D Databáze na Webu

Odesílání HTML

17.1.2. Jak sdělit 4D, aby vyčistila stránku

4D nemá žádný problém v obsluhování vašich stránek, ale tato vaše Web stránka nezahrnuje způsob jak sdělit 4D, aby pokračovala. Vaše metoda Při Web spojení čeká na potvrzení ze stránky, že může provádět něco dalšího nebo odeslat záhlaví nabídek. Opravme nyní tento problém:

1. Jděte do Prostředí návrháře.
2. Vytvořte novou metodu projektu WEB_SendEmptyPage a to následovně:

```
If (False)
  ` Method: WEB_SendEmptyPage
  ` ACI University Programming Classes
  ` Created By: David Adams
  ` Date: 3/21/97

  ` Purpose: Clears 4D's HTML mode

  ⋄f_Version6x40 := True
  ⋄fD_Adams := True

End if

SEND HTML FILE ("" )
`End of method
```

3. Uzavřete a uložte metodu.
4. Otevřete okno seznamu procesů a odstraňte všechny procesy Web připojení.
5. Ponechte seznam procesů viditelný.
6. Otevřete vaši Home.htm textovým editorem.
7. Přidejte do definice formuláře tlačítko, které 4D řekne, aby odeslala prázdnou stránku:

```
<FORM ACTION="/4DMethod/WEB_SendEmptyPage/" Method="POST">
<INPUT TYPE="submit" NAME="bDone" VALUE="Done">
</FORM>
```

9. Home.htm by nyní měla vypadat následovně:

```
<HTML>
<HEAD>
```

```
<TITLE>Welcome!</TITLE>
```

```
<BODY>
```

Dobrý den a vítějte na naší webovské stránce. Stránka se právě upravuje, ale brzy nás opět navštívte!<P>

```
<FORM ACTION="/4DMethod/WEB_SendEmptyPage/" Method="POST">
<INPUT TYPE="submit" NAME="bDone" VALUE="Done">
</FORM>
```


Používání 4D Databáze na Webu

Odesílání HTML

</BODY>

</HTML>

10. Uložte změny.
11. Jděte do svého prohlížeče.
12. Otevřete nové připojení s vaší databází.
13. Nastartuje se nový proces Web připojení.
14. Opravená [Home.htm](#) se objeví obsloužena 4D (jestliže se nezměnila ujistěte se, že jste vaši práci v textovém editoru uložili a krok opakujte).
15. Klepněte na tlačítko, které jste do této stránky přidali.
16. Objeví se nabídka Web.
17. Nenastartuje se žádný nový proces Web připojení – jste stále uvnitř jednoho procesu, který jste nastartovali a to díky úpravě [Home.htm](#).

17.1.3. Vždy zahrnovat odkaz nebo tlačítko

Protože 4D je vždy Web server “udržující stav” potřebujete vždy zahrnout odkaz nebo tlačítko k vyčištění stránky. V předchozím jste viděli jak to provést pomocí formuláře a tlačítka. Nyní se podívejme na syntaxe pro odkaz:

1. V textovém editoru otevřete Home.htm.
2. Přidejte následující HTML nad definici fomuláře:

<P>

Chci nabídku Web!

<P>

3. Home.htm by měla vypadat následovně:

<HTML>

<HEAD>

<TITLE>Vítejte!</TITLE>

<BODY>

Dobrý den a vítejte na naší webovské stránce. Stránka se právě upravuje, ale brzy nás opět navštivte!<P>

<P>

Chci nabídku Web!

<P>

Používání 4D Databáze na Webu

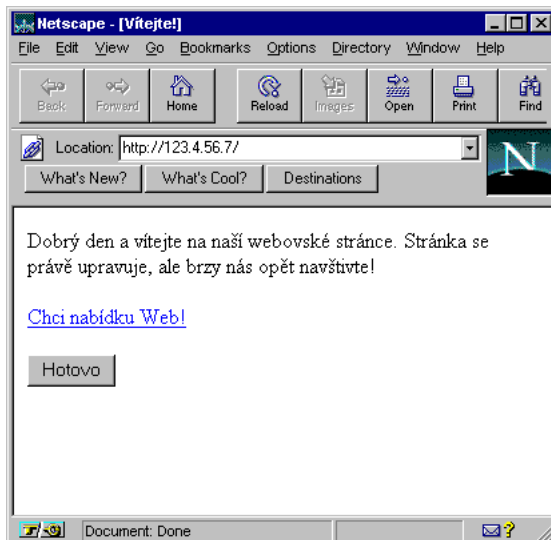
Odesílání HTML

```
<FORM ACTION="/4DMethod/WEB_SendEmptyPage/" Method="POST">
<INPUT TYPE="submit" NAME="bDone" VALUE="Hotovo">
</FORM>

</BODY>

</HTML>
```

4. Uložte změny.
5. Přejděte do prohlížeče Web.
6. Vytvořte nové připojení s vaší databází.
7. Měli by jste vidět následující stránku:



8. Klepněte na odkaz Chci nabídku Web!
9. Objeví se nabídka Web.

Nezáleží na tom zda použijete odkaz nebo tlačítko, obojí pracuje dobře. To co jsme právě viděli byl příklad automatického zpětného volání ošetřeného ve 4D. 4D čte přicházející URL a vytvoří pro /4D METHOD/ klíčové slovo a provede metodu jestliže ji zde nalezne. Proč se objeví nabídka Web? Podívejme se nyní do metody Při Web spojení. Povšimněme si, že na spodu poslední dvě řádky kódu jsou:

```
SEND HTML FILE("Home.htm")
```

```
` OK, now show the regular Web menu:
MENU BAR(5)
```

Když se uživatel poprvé připojí SEND HTML FILE odešle z disku vaši jednoduchou domovskou stránku. Metoda Při Web spojení se v tomto připojení zastaví dokud není "HTML mód" vypnut – vyčištěn. Jinými slovy čeká dokud nevyčistíte stránku a pak provede další instrukce zde odeslání nabídky. Účelem tlačítka nebo odkazu je odeslat 4D instrukci k vyčištění stránky. Když je stránka vyčištěna metoda Při Web spojení pokračuje.

Používání 4D Databáze na Webu

Odesílání HTML

17.1.5. Uživatelské formuláře se zpětným voláním 4DMETHOD

Nyní máme za sebou úvod v integraci uživatelských formulářů HTML se 4D. Podívejme se na toto téma podrobněji:

1. Jděte do Prostředí návrháře ve 4D.
2. Vytvořte novou metodu projektu WEB_SendCallback a to následovně:

```
If (False)
  ` Method: WEB_SendCallback
  ` ACI University Programming Classes
  ` Created By: David Adams
  ` Date: 3/21/97

  ` Purpose: Test calling a 4D method form HTML

  ⋄f_Version6x40 := True
  ⋄fD_Adams := True

End if

SEND HTML FILE ("CountObjects.htm")
`End of method
```

3. Uložte a uzavřete tuto metodu.
4. Otevřete vaše záhlaví nabídek Web a přidejte novou položku nazvanou “Spočti objekty”.
5. Přiřaďte metodu Web_SendCallback této položce nabídky.
6. Uzavřete Editor nabídek.
7. Otevřete metodu databáze Při Web spojení.
8. Nahradejte řádku SEND HTML FILE následovně:

Web_SendCallback

9. Uložte všechny změny.
10. Přejděte do textového editoru.
11. Vytvořte nový soubor pojmenovaný CountObjects.htm a uložte jej do té samé složky jako strukturu databáze.
12. Do souboru přidejte následující HTML :

<HTML>

<HEAD>

<TITLE>Vítejte!</TITLE>

<BODY>

Podívejme se, jak vytvořit řádek který bude volat naši vlastní metodu<P>

<P>

Spočti objekty

Používání 4D Databáze na Webu

Odesílání HTML

<P>

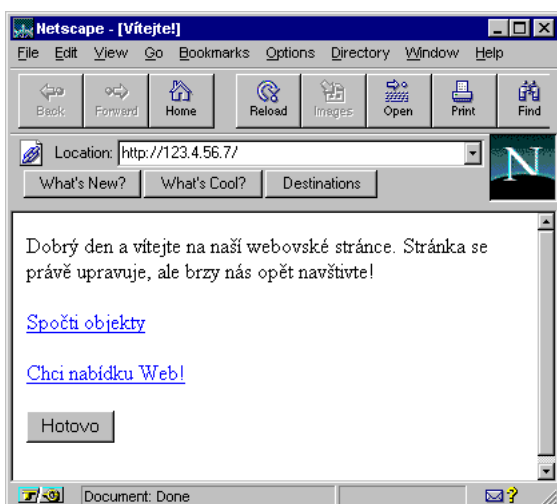
Chci Web nabídku!

<P>

</BODY>

</HTML>

13. Uložte tento soubor.
14. Přejděte do prohlížeče.
15. Otevřete nové připojení k databázi.
16. Nastartuje se nový proces Web připojení.
15. Objeví se stránka CountObjects.htm:



16. Klepněte na odkaz Spočti objekty.
17. Objeví se upozornění popisující třetí tabulku.
18. Přejděte do vašeho textového editoru a změňte odkaz Spočti objekty tak, aby žádal o informaci o tabulce 2 místo tabulky 3:
- 19.

Spočti objekty

19. Uložte změny.
20. Jděte do prohlížeče.
21. Klepněte OK k vyčištění dialogu popisujícího tabulku 3.
22. Objeví se nová verze stránky CountObjects.htm.
23. Klepněte na odkaz Spočti objekty.
24. Objeví se upozornění popisující tabulku 2.

Používání 4D Databáze na Webu

Odesílání HTML

25. Klepněte OK.

Během všech předchozích kroků byl použit pouze jeden proces Web připojení. Tajemství zpětného volání 4D je, zahrnuje zpětná volání pouze do dokumentů ošetřovaných ze 4D.

17.1.6. Ukončení HTML módu

Systém klíčových slov HTML 4D METHOD velmi zjednodušuje integraci zákaznických HTML a 4D routines. Pokud již odešlete soubor pomocí SEND HTML FILE, 4D pokračuje v HTML módu dokud zpětně nezavoláte metodu, která řekne SEND HTML FILE, aby odeslal prázdný znak. Toto je důvod pro zpětné volání WEB_SendEmptyPage používané v těchto zákaznických HTML dokumentech.

17.1.7. Upozornění

4D provede jakoukoliv metodu, kterou do ní zpětně odešlete. Je to velmi silný nástroj, ale nebezpečný. Jednodušší systém zpětného volání, který jsme vytvořili na začátku kursu vyžaduje prověrku URL a dovoluje vám potlačit nebo odmítnout požadavky, které jsou nebezpečné nebo nesprávné. Jestliže chcete zahrnout podobnou úroveň bezpečnosti do automatického zpětného volání 4D METHOD, potřebujete testovat zda je prováděné připojení Web připojením. V předchozím jsme se již zabývali sledováním typu připojení, takže umíme tento test jednoduše zahrnout. Zde vidíme příkladovou metodu, která toto provádí:

```
C_POINTER($Current_table)
C_LONGINT($LTableNumber)

If (GEN_fThisIsAWebConnection )
    ALERT("Z Web klienta nemůžete použít metodu 'Delete_all_data_and_quit' ")
Else

    READ WRITE(*)

    For ($LTableNumber;1;Count tables)

        $Current_table:=Table($Current_table->)

        ALL RECORDS($Current_table)
        DELETE SELECTION($Current_table)

        While (Records in set("LockedSet") # 0)
            USE SET("LockedSet")
            DELETE SELECTION($Current_table->)
        End while

    End for

    QUIT 4D

End if
```

Poněkud méně dramatický ale bližší normálnímu použití je následující příklad:

```
` Method: QUIT_DATABASE ()
`
```

Používání 4D Databáze na Webu

Odesílání HTML

```
` Do cleanup tasks and quit.  
  
If (GEN_fThisIsAWebConnection )  
    ALERT("Tuto databázi nemůžete ukončit z Web připojení.")  
Else  
    QUIT_CLEANUP_TASKS  
    QUIT 4D  
End if
```

Vaše testy můžete vylepšit dotazem na hesla, prověřením příslušnosti k patřičným bezpečnostním skupinám nebo prověřovat i jiné věci pomocí uživatelských účtů uložených v záznamech databáze.

17.2. Vnořované odkazy na soubor

4D vám poskytuje ještě jiný způsob jak odeslat HTML stránky. Můžete vložit odkaz na soubor dovnitř pole statického textu ve formuláři. Když odešlete formulář 4D automaticky odešle i HTML soubor. Nyní to provedme:

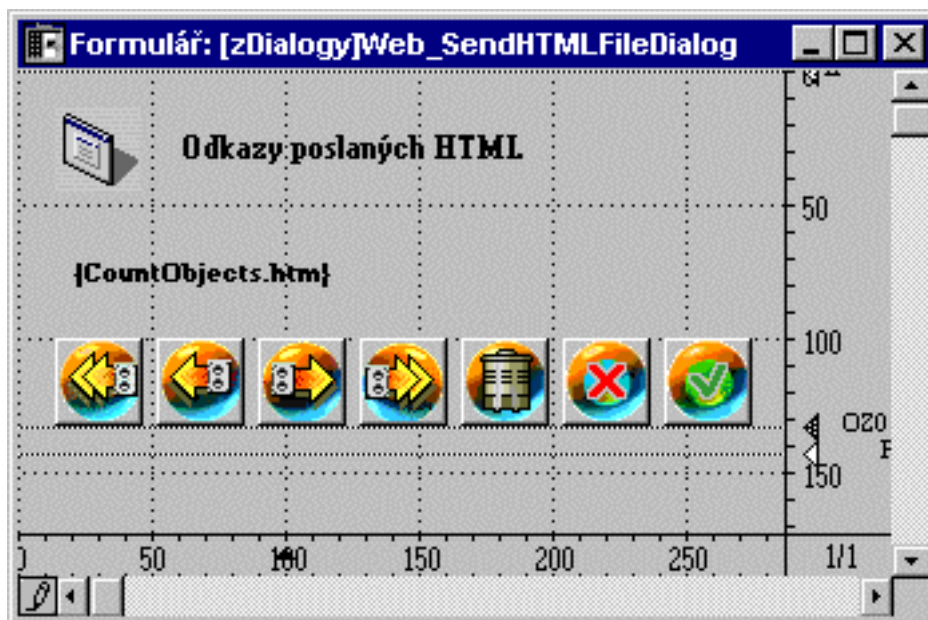
1. Přejděte do Prostředí návrháře 4D.
2. Do tabulky zDialogy přidejte nový vstupní formulář Web Aware nazvaný Web_SendHTMLFileDialog.
3. Ujistěte se, že jste přidali pole Povinné.
4. Upravte formulář.
5. Odstraňte pole Povinné popisku ponechte.
6. Ponechte tlačítka i grafiku na místě.
7. V popisce pole povinné napište ve složených závorkách následující referenci:

{CountObjects.htm}

8. Syntaxe {Cesta soubor} říká 4D, aby hledala soubor a neposílala samotný text.
9. Změňte statický text dialogu na "Odkazy poslaných HTML".
10. Formulář by měl vypadat následovně:

Používání 4D Databáze na Webu

Odesílání HTML



11. Uložte a uzavřete formulář.
12. Vytvořte novou metodu projektu WEB_SendHTMLFileDialog a to následovně:

```
If (False)
  ` Method: WEB_SendHTMLFileDialog
  ` ACI University Programming Classes
  ` Created By: David Adams
  ` Date: 3/21/97

  ` Ucel: Demonstruje vnoreni souboru HTML do formulare

  ⚡_Version6x40 := True
  ⚡D_Adams := True

End if

DIALOG([zDialogs];"Web_SendHTMLFileDialog")
  `Konec metody
```

14. Uložte a uzavřete metodu.
15. Přidejte novou položku nabídky do Web záhlaví nabídek nazvanou Poslat soubor.
16. Přiřaďte metodu Web_SendHTMLFileDialog k nové položce.
17. Uložte změny a uzavřete Editor nabídek.

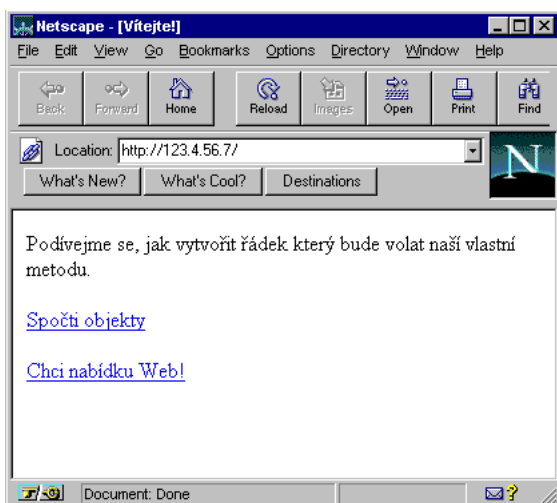
Nyní jsme přidali dialog, kterým se chystáme odeslat ten samý soubor, se kterým jsme pracovali v několika posledních příkladech, když vyberete patřičnou položku z nabídky Web. Vyzkoušejme si to:

18. Přejděte do prohlížeče.
19. Vytvořte nové spojení s vaší databází.

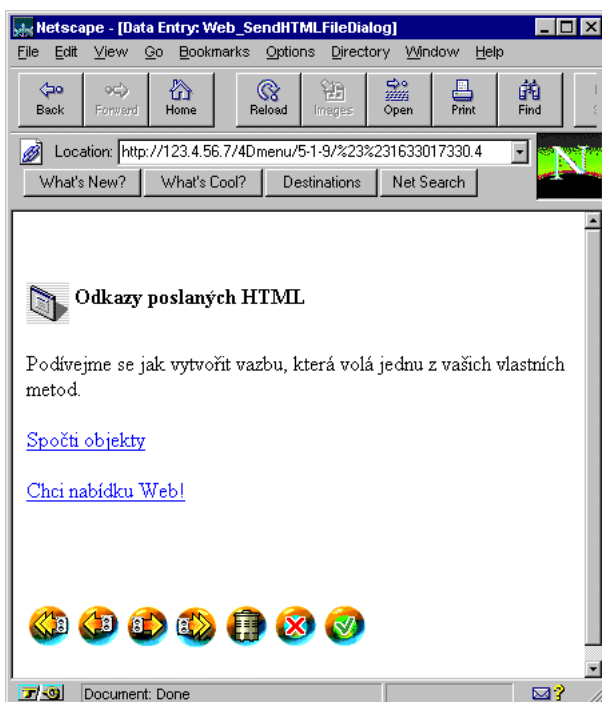
Používání 4D Databáze na Webu

Odesílání HTML

20. Objeví se soubor CountObjects.htm. Je to proto, že jsme nezměnili metodu Při Web spojení, zde nepoužíváme nově vytvořený dialog. Všimněte si, že tato stránka nemá tlačítka 4D, má pouze odkazy, které jste napsali do HTML.



21. Klepněte na odkaz Chci nabídku Web!
22. Objeví se Web nabídka..
23. Klepněte na odkaz Poslat soubor..
24. Spustí se vaše metoda Web_SendHTMLFileDialog , která říká 4D, aby zobrazila váš nový dialog, který uvnitř řekne 4D, aby odeslala soubor CountObjects.htm
25. Soubor se objeví, ale všimněte si že je integrovaný do 4D formuláře:



Používání 4D Databáze na Webu

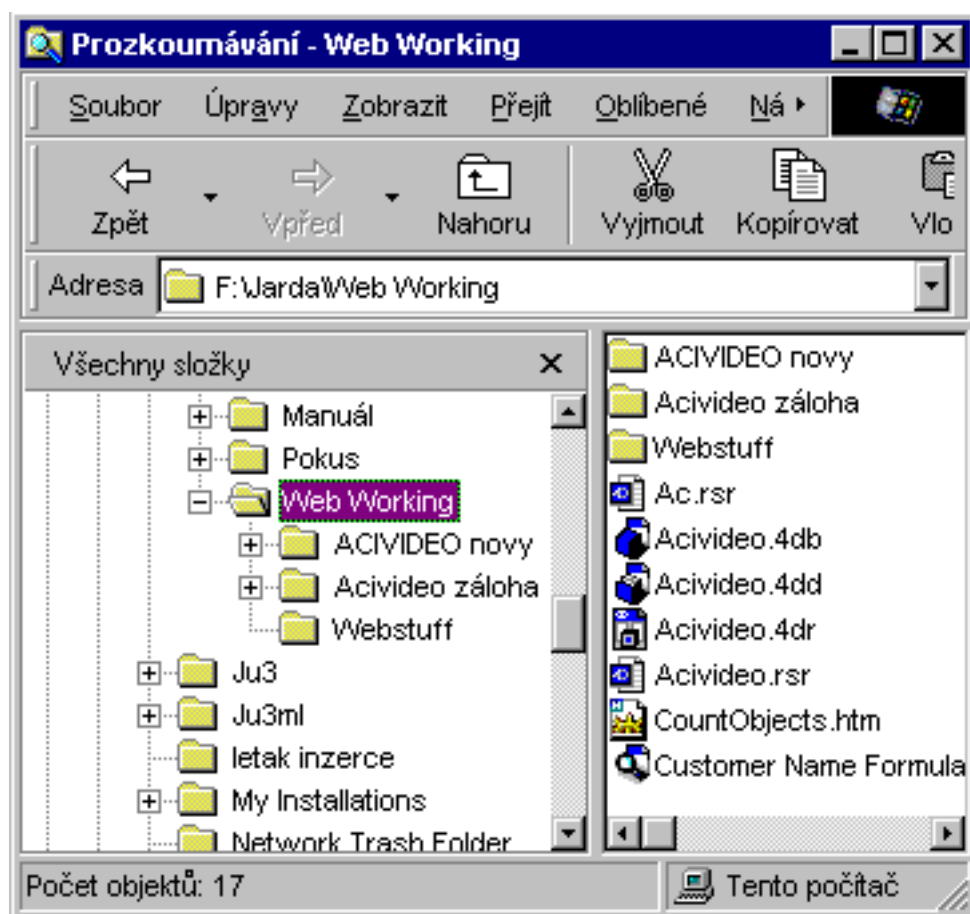
Odesílání HTML

Není to ohromující! Je to jeden z nejlepších rysů v6 pro Web. Můžete vkládat uživatelské HTML soubory dovnitř zákaznických formulářů bez jakéhokoliv úsilí. 4D přečte zdroj vašeho HTML souboru a odstraní tagy, které by mohli způsobit konflikt. Např. stránka HTML nesmí mít více než jednu sadu následujících tagů struktury: HTML, HEAD a BODY. Podívejme se na to:

26. Přejděte do prohlížeče Web.
27. Zobrazte si zdroj (source) této stránky.
28. Můžete vidět, že je zcela odlišný od zdroje originálního souboru CountObjects.htm, ale stále zahrnuje všechny prvky uvedené v struktuře body.
29. 4D to za vás všechno ošetří automaticky.

17.3. Změna cesty k HTML

Jak jste již viděli 4D hledá HTML soubory v téže složce, ve které je umístěna struktura databáze. Můžete určit cestu relativně k umístění struktury databáze jestliže v ní máte podsložky/subdirectories:



Jestliže máte složku Statické můžete požadovat dokumenty následujícím způsobem.

SEND HTML FILE("Statické/CountObjects.htm")

Používání 4D Databáze na Webu

Odesílání HTML

“/” je standartní oddělovač pro URL nezávislý na platformě, pracuje stejně na Windows i Mac.

Jestliže přemístíte všechny vaše dokumenty HTML na speciální místo, můžete přenastavit celé výchozí hledání cesty. To znamená, že 4D bude hledat a odesílat HTML soubory ze složky, kterou nastavíte a ne ze složky struktury databáze. Zde máme příklad:

```
SET HTML ROOT("Statické/")  
SEND HTML FILE("CountObjects.htm")
```

Tento příkaz odešle soubor HTML z místa:

Složka umístění struktury /Statické/CountObjects.htm

Běžné je nastavovat cestu HTML při startu databáze. Poznámka: Tento příkaz má efekt na všechny procesy Web připojení a ne pouze na proces, kde byl příkaz použit.

17.3.1. Změna výchozí cesty

1. Vytvořte ve složce ACIVideo podsložku s vlastním názvem.
2. Přemístěte všechny soubory HTML do této složky.
3. Do své metody Při Web spojení přidejte příkaz SET HTML ROOT s názvem podsložky, kterou jste vytvořili. .
4. Otestujte tyto změny v prohlížeči Web.

17.4. Vnoření bloků HTML

17.4.1. Princip odeslání HTML v proměnné formuláře

V dřívější kapitole, kde jsme odesílali ozvěnu URL a hlavičky HTTP z metody Při Web spojení, jsme se již stručně zmínili jak vnořovat HTML pomocí 4D proměnné. Vnoření HTML včetně JavaScript do 4D formulářů je jednoduché:

1. Umístěte do formuláře nedostupnou textovou proměnnou.
2. Použijte metodu k vložení znaku ASCII 1 do proměnné.
3. Přidejte k proměnné jakékoliv HTML, které chcete.
4. Odešlete formulář přes Web pomocí DIALOG, ADD RECORD, MODIFY SELECTION nebo DISPLAY SELECTION.

17.4.2. Příklad odeslání HTML v proměnné formuláře

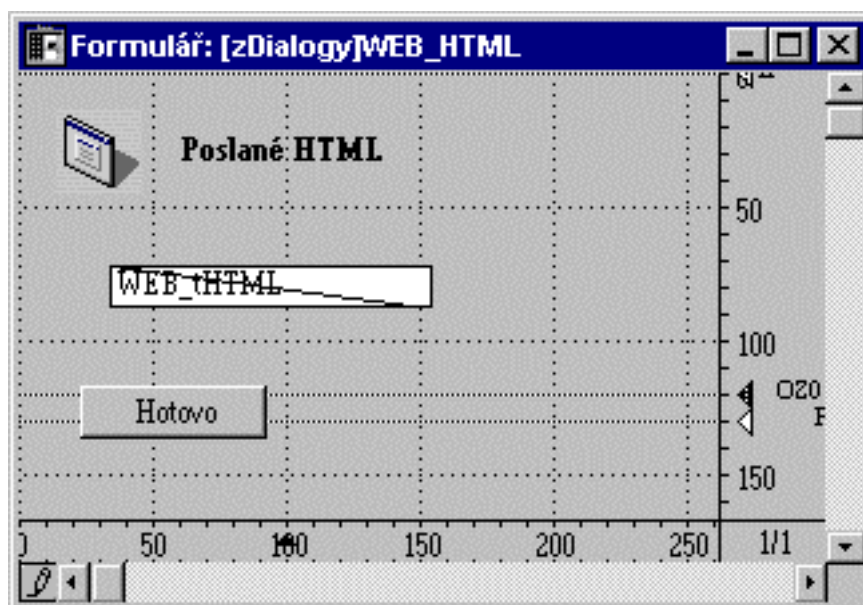
Když takovýto formulář použijete ve Web procesu. Vloží se HTML blok do zdroje HTML automaticky, podívejme se na příklad takovéto akce.

1. Přejděte do Prostředí návrháře 4D.
2. Přidejte nový vstupní formulář Web Aware do tabulky zDialogy nazvaný WEB_HTML.
3. Ujistěte se, že jste přidali pole Povinné.
4. Užijte vzor ACI Video Web vytvořený dříve.
5. Upravte formulář.

Používání 4D Databáze na Webu

Odesílání HTML

6. Odstraňte pole Povinné i s popiskou.
7. Přidejte nedostupnou oblast nazvanou WEB_tHTML a umístěte ji nad tlačítko Hotovo.
8. Změňte titul formuláře na Poslané HTML.
9. Změňte text tlačítka na Hotovo.
10. Formulář by měl vypadat následovně:



11. Uložte a uzavřete tento formulář.
12. Uložte novou metodu projektu WEB_SendHTML následovně:

```
If (False)
  ` Method: WEB_SendHTML
  ` ACI University Programming Classes
  ` Created By: David Adams
  ` Date: 3/21/97

  ` Purpose: Demonstruje vnoření HTML textu ve formuláři.

  ⋄f_Version6x40 := True
  ⋄fD_Adams := True

End if

C_TEXT(WEB_tHTML)

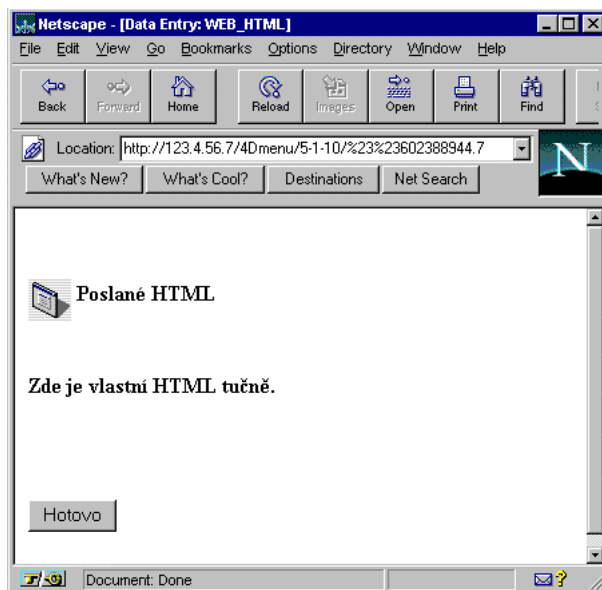
WEB_tHTML := Char(1)
WEB_tHTML := WEB_tHTML+"<B>Zde je vlastní HTML tučně.</B>"

DIALOG([zDialogs];"WEB_HTML")
  `End of method
```

Používání 4D Databáze na Webu

Odesílání HTML

13. Uložte a uzavřete tuto metodu.
14. Otevřete metodu Při Web spojení.
15. Odstraňte řádek kódu, který volá `Web_SendCallback`. Chceme, aby se nabídka Web objevila jakmile se připojíme k tomuto příkladu.
16. Uložte vaše změny do metody Při Web spojení.
17. Přidejte novou položku nabídky do nabídky Web nazvanou Vnořené HTML.
18. Přiřaďte metodu `Web_SendHTML` této nové položce.
19. Uložte změny v nabídce a uzavřete Editor nabídek.
20. Přejděte do prohlížeče.
21. Spojte se s vaší databází.
22. Nabídka by se měla objevit bez jakéhokoliv předchozího dialogu.
23. Klepněte na nový odkaz vnořené HTML v nabídce.
24. Měli by jste vidět následující stránku:



25. Klepněte tlačítko Hotovo a vraťte se do nabídky Web.

Poznámka: Jestliže se vaše HTML zobrazí jako zadávací pole HTML, znamená to, že vaše proměnná `WEB_tHTML` je definována ve 4D jako dostupná. Přejděte zpět a změňte tuto proměnnou na nedostupnou a vyzkoušejte ji.

4D automaticky umístí vaše HTML do formuláře. Zkusme změnit HTML:

17.4.3. Odeslání více HTML v proměnné formuláře.

1. Upravte metodu `Web_SendHTML` následovně:

```
...  
C_TEXT(WEB_tHTML)
```

Používání 4D Databáze na Webu

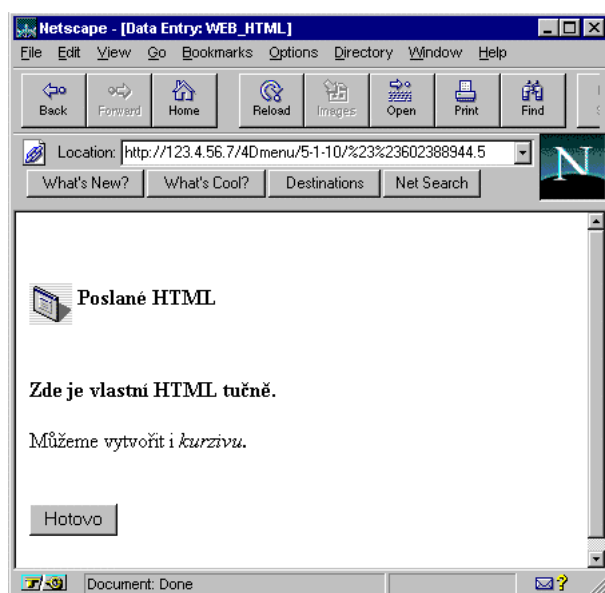
Odesílání HTML

❖ WEB_tHTML := Char(1)
WEB_tHTML := WEB_tHTML+"Zde je vlastní HTML tučně.<P>"+
Char(Carriage return)

❖ WEB_tHTML := WEB_tHTML+"Můžeme vytvořit i <I>kurzívu.</I>"+Char(Carriage
return)

DIALOG([zDialogs];"WEB_HTML")

2. Uložte a zavřete tuto metodu.
3. Jděte do prohlížeče.
4. Klepněte na odkaz Vnořené HTML v nabídce Web.
5. Uvidíte něco takového:



Poznamenejme, že 4D zobrazí jakékoliv HTML, které mu dáte. Další prvky formulář jako titulek okna a tlačítko Hotovo, jsou příkladem čehokoliv dalšího co jste zahrnuli.

17.4.2. Nevnořené HTML

Když vnoříte HTML do formuláře, 4D je automaticky uzavře do standartní stránky formuláře 4D HTML. Což je omezení toho co můžete zahrnovat tímto způsobem.

1. Přejděte do prohlížeče a prohlédněte si zdroj stránky, kterou jste právě obdrželi.
2. Uvidíte něco takového:

<HTML>

<HEAD>

<TITLE>Data Entry: WEB_HTML</TITLE>

</HEAD>

Používání 4D Databáze na Webu

Odesílání HTML

```
<BODY>

<FORM Method="POST" Action="/4Ddataentry/1/26/$310799296.5">

<BR>

<BR>

<IMG Src="/4Dimg/4Dlayout/1101699245/$310799296.1101699245" Width="32"
Height="32" Align="middle">

<B><FONT color="#000000">Vnořené HTML</B></FONT><BR>

<BR>

<FONT color="#000000"><B>Zde je uživatelské HTML, tučně.</B><P>
Můžeme <I>kurzivu</I> rovněž použít.
</FONT><BR>

<BR>

<FONT color="#000000"><INPUT Type="submit" Name="4DVar.bValidate" Value="Hotovo">

</FONT></FORM>

</BODY>

</HTML>
```

Vaše uživatelské HTML jsou umístěna uvnitř vytvořeného formuláře 4D v sekci body definované 4D:

```
<HTML>

<HEAD>

<TITLE>Vstup dat: WEB_HTML</TITLE>

</HEAD>

<BODY>

<FORM Method="POST" Action="/4Ddataentry/1/26/$310799296.5">

Zde vaše HTML.

</FORM>

</BODY>

</HTML>
```

Používání 4D Databáze na Webu

Odesílání HTML

Několik sekcí HTML nemůže být tímto způsobem použito (vše co chcete poslat beze změny může být odesláno přes SEND HTML FILE). Zde je seznam položek, které nelze vnořit:

Položka	Poznámka
HEAD	Nemůžete zahrnout jakýkoliv prvek, který patří do hlavičky HEAD. Včetně TITLE a META tagu.
Frames	Frame a frameset (dělení do několika nezávislých stránek v jednom okně) nejsou normální HTML dokumenty. Pár tagů FRAMESET je používán místo páru tagů BODY. Podpora pro vnořování rámců Frame bude zahrnuta až v budoucích verzích 4D. Rámy Frame můžete obsluhovat pomocí SEND HTML FILE, ale každý samostatný rám volá 4D a nastartuje nový proces Web připojení. Rámy Frame nejsou v současné době ve 4D doporučeny.
Forms	Formuláře HTML nemohou být vnořovány samy do sebe. Můžete jich mít více na jedné stránce, ale ne jeden uvnitř druhého. Protože 4D již definovala danou stránku jako formulář, nemůžete do ní vložit jinou. Můžete přidávat prvky formuláře jako vstupy a textové oblasti, kdykoliv si přejete.

17.4.3. Používání dočasných souborů

Vše co nemůžete zahrnout pomocí vnoření proměnných, můžete ovládat přes externí HTML soubor a příkaz SEND HTML FILE. Jestliže objevíte, že váš Web systém požaduje dynamicky vytvářené HTML, které nemůžete vnořit do HTML proměnné přímo, pak můžete vytvořit dočasný soubor, který obsahuje nezbytné HTML a po použití jej vymazat. Citlivý způsob k zahrnutí tohoto rysu je vytvoření speciální složky pro tyto vymazatelné soubory, který vyčistíte vždy, když otevíráte nebo zavíráte databázi. Protože používáte databázi, užíjte meziprocesní proměnné chráněné semaforem k ukládání dalšího dočasného čísla dokumentu a generickou rutinu, k vytvoření samotného dokumentu.

17.5. Vázání proměnných

Již víte, jak vytvořit uživatelský formulář, který pošlete pomocí SEND HTML FILE. Rovněž víte jak zahrnout do těchto formulářů volání 4D metod. Co ještě musíme přidat k těmto znalostem, je jak definovat proměnné v HTML, které odpovídají proměnným ve 4D. Tento rys 4D vám dovolí vytvořit váš vlastní HTML formulář jednoduše a s velkou pružností.

17.5.1. Příklad na vázání proměnných

1. Ve vašich školících materiálech je soubor nazvaný Variable.binding.htm.
2. Přemístěte tento soubor do téže složky jako databázovou strukturu — tj. do složky, kde 4D hledá HTML dokumenty.
3. Otevřete tento dokument vaším textovým editorem.
4. Zde je co uvidíte:

<HTML>

<HEAD>

<TITLE>Vázání proměnných</TITLE>

Používání 4D Databáze na Webu

Odesílání HTML

```
<BODY>

<FORM NAME="variableBindingForm" METHOD="POST"
ACTION="/4DMETHOD/WEB_VariableBindAlert/">

<B>Vložte sem text a klepněte tlačítko Test.</B><BR>
<Input type=text name="WEB_sVariable" value="">
<P>

<INPUT TYPE=SUBMIT NAME="testButton" VALUE="Test">

</FORM>

<P>

<A HREF="/4DMETHOD/WEB_SendEmptyPage/">Chci Web nabídku!</A>

</BODY>

</HTML>
```

5. Přečteme-li si tento dokument můžeme vidět, že formulář by měl ve 4D pracovat správně a že zpětně volá novou metodu nazvanou WEB_VariableBindAlert.
6. Vytvořte novou metodu projektu WEB_VariableBindAlert a to následovně:

```
If (False)
  ` Method: WEB_VariableBindAlert
  ` ACI University Programming Classes
  ` Created By: David Adams
  ` Date: 3/21/97

  ` Purpose: Demonstruje vázání HTML proměnných

  ⋄f_Version6x40 := True
  ⋄fD_Adams := True

End if

ALERT(WEB_sVariable)
  `End of method
```

7. Všimněte si, že název vstupního pole v HTML formuláře je WEB_sVariable.
8. Uložte tuto metodu.
9. Otevřete vaši metodu Při Web spojení.
10. Nad příkaz nabídky, přidejte následující řádky kódu:

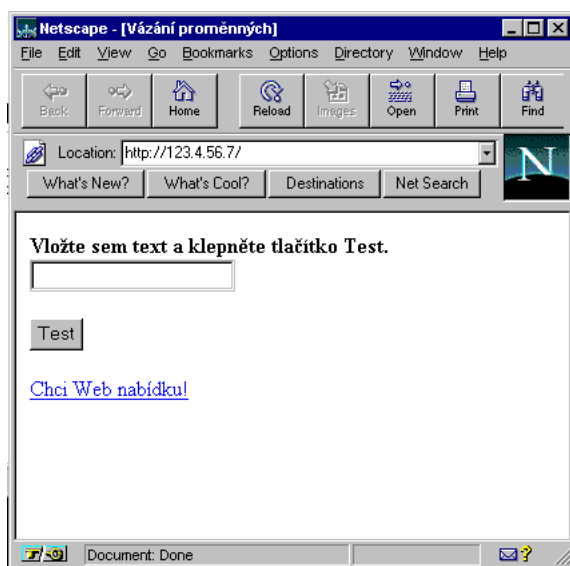
```
WEB_sVariable := ""
SEND HTML FILE("Variable.binding.htm")
```

11. První řádka kódu inicializuje WEB_sVariable uvnitř 4D a druhá řádka odesílá určený soubor.

Používání 4D Databáze na Webu

Odesílání HTML

12. Uložte změny do metody Při Web spojení.
13. Jděte do vašeho prohlížeče Web.
14. Připojte se k databázi.
15. Uvidíte následující obrazovku:



16. Vložte cokoli do vstupní oblasti a klepněte na tlačítko Test.
17. Uvidíte upozornění, které zpětně odráží co jste napsali.
18. Vraťte se do prostředí návrháře.
19. Otevřete metodu Při Web spojení.
20. Odstraňte tyto řádky kódu, které jsme napsali pro toto cvičení:

```
WEB_sVariable:=""  
SEND HTML FILE("Variable.binding.htm")
```

17.5.2. Pravidla zjednodušující vázání proměnných

Následuje jednoduchý příklad, který ilustruje mechanismu tohoto silného rysu. Pamatujte si následující:

1. Hodnoty z HTML polí můžete do 4D proměnných převádět **automaticky**.
2. Trik spočívá v tom, že pojmenujete pole HTML stejně jako 4D proměnné.
3. Pracujte s proměnnými alfanumerických typů.
4. Ujistěte se, že proměnné jsou definovány a inicializovány ve 4D (je to normální a správný programátorský postup).
5. Odešlete a přijměte tuto stránku pomocí 4D.
6. To je vše!

V tomto příkladu jsme pro snadnější pochopení použili pouze jednu proměnnou, můžete však použít tolik proměnných kolik potřebujete, pokud budete dodržovat pravidla, která jsme stanovili výše.

Používání 4D Databáze na Webu

Ukládání HTML v záznamech databáze

18. Ukládání HTML v záznamech databáze

18.1. Pozadí

Pomocí vestavěných automatických nástrojů v6, můžete vytvořit Web systém bez napsání řádky HTML. Existuje však spousta běžných důvodů, pro které budete vytvářet své vlastní HTML. Jak jsme již viděli, HTML je dost upovídaný jazyk. To jej činí velice dobře čitelným, ale poměrně nesnadno ovladatelným. Obvyklá cesta při vytváření HTML stránek je sestavování kompletních HTML po jednotlivých řádcích. Jako např.:

C_TEXT (\$tHTML_Block)

```
$tHTML_Block := "<HTML><HEAD><TITLE>Nazdar světe!</TITLE></HEAD>"
$tHTML_Block := $tHTML_Block + "<BODY><H1>Text obsahu zde.
</H1></BODY>"
$tHTML_Block := $tHTML_Block + "</HTML>"
```

Nevýhody této praxe jsou poměrně zřejmé:

- ❖ Vytváření kódu tímto způsobem, je zdlouhavé nudné a obvykle chybové.
- ❖ Hlídaní všech uvozovek vpravo apod. je urážkou lidské inteligence.
- ❖ Vaše HTML se stanou nepružné, protože je je nesnadné upravit.
- ❖ Při každé změně v HTML potřebujete vždy překompilovat databázi.
- ❖ Nemůžete jednoduše zkontrolovat pravopis textů použitých v kódu.
- ❖ Bude obtížné takovéto stránky testovat
- ❖ Celková změna bloků HTML způsobí, že budete muset upravit množství metod.
- ❖ Lokalizace takového projektu bude obtížná.
- ❖ Budete potřebovat přepsat metody, pokud bude nutno podporovat jiný jazyk tagů jako např. QuarkXPress tagy nebo FrameMaker Interchange Format.
- ❖ Konečným uživatelům není dána žádná kontrola přes výstup HTML.
- ❖ Vnitřní textové bloky, jsou rekonstruovány pokaždé, když jsou použity, což je pomalé.

Všechny tyto problémy můžete vyřešit jednou tabulkou a několika metodami. Celou tuto práci provedete velice rychle během půldne.

Používání 4D Databáze na Webu

Ukládání HTML v záznamech databáze

18.2. Ukládání dat v záznamech

18.2.1. Struktura záznamu

4D záznamy jsou logickým místem k uložení opakovaně používaných HTML bloků:

zHTML_Bloky	
Název_Bloku	A
HTML_Blok	T

Pro zjednodušení úprav můžete vaše HTML bloky uložit v textových polích. Můžete vytvořit HTML v Editoru HTML, otestovat je v prohlížeči Web a potom vložit do 4D, když jste již spokojeni s výsledkem. Můžete uložit definice formulářů, bloky textů se styly a dokonce celé stránky. Nyní můžete přidat uvozovky a nové řádky do HTML bez nebezpečí pro své duševní zdraví nebo zdraví svých spolupracovníků.

18.2.2. HTML bloky v akci

1. Jděte do Prostředí návrháře 4D.
2. Zvolte Nástroje \rightarrow Struktura databáze.
3. Přidejte novou tabulku a pojmenujte ji zHTML_Bloky
4. Přidejte alfa pole nazvané Název bloku o vlastnostech, 80 znaků, indexované, povinné, jedinečné.
5. Přidejte textové pole nazvané HTML_Block.
7. Přidejte novou metodu nazvanou HTML_tBlock.
8. Otevřete soubor HTML_tBlock z vašich školících materiálů a překopírujte obsah do nové metody.
9. Uložte tuto metodu.
10. Vytvořte novou metodu projektu WEB_SendHTMLBlock následujícím způsobem:

```
If (False)
  ` Method: WEB_SendHTMLBlock
  ` ACI University Programming Classes
  ` Created By: David Adams
  ` Date: 3/21/97

  ` Purpose: Demonstruje posílání HTML bloků.

  ⋄f_Version6x40 := True
  ⋄fD_Adams := True

End if

C_TEXT(WEB_tHTML)

WEB_tHTML := Char(1)
```

Používání 4D Databáze na Webu

Ukládání HTML v záznamech databáze

```
WEB_tHTML:=HTML_tBlock ("Test Block")
```

```
DIALOG([zDialogs];"WEB_HTML")  
`End of method
```

11. Uložte a uzavřete tuto metodu.
12. Přidejte novou položku nabídky do nabídky Web nazvanou Poslat HTML Blok.
13. Přiřaďte metodu Web_SendHTMLBlock položce této nabídky.
14. Přejděte do Prostředí uživatele a nechte 4D vytvořit formuláře pro tuto tabulku.
15. Vraťte se do Prostředí návrháře a otevřete formuláře [zHTML_Bloky];Vstup.
16. Upravte velikost textové oblasti podle formuláře.
17. Přiřaďte textové oblasti posuvníky.
18. Jděte do Prostředí uživatele.
19. Proveďte metodu E_Set2ReadWrite.
20. Přidejte nový záznam do zHTML_Bloky a pojmenujte jej Test block a vložte následující text ze souboru Test block s použitím textového editoru:

```
<B>Pokus s text blokem</B>  
<P>
```

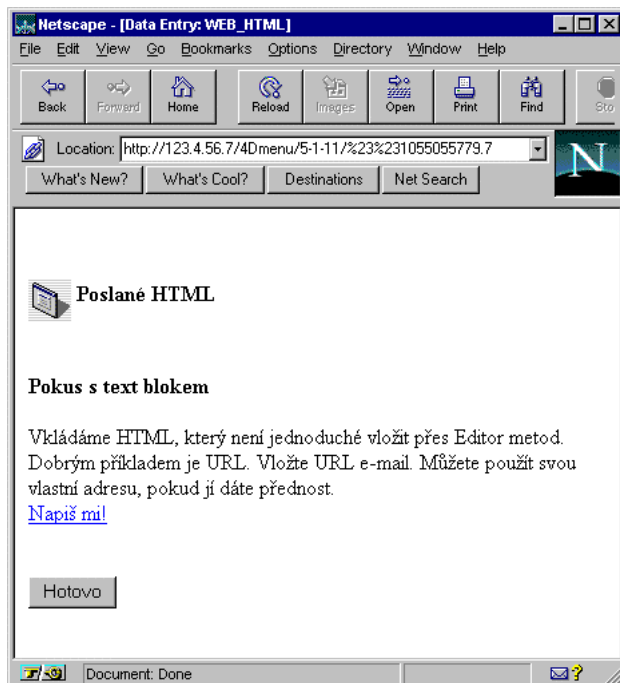
Vkládáme HTML, který není jednoduché vložit přes Editor metod. Dobrým příkladem je URL. Vložte URL e-mail. Můžete použít svou vlastní adresu, pokud jí dáte přednost.

```
<BR>  
<A HREF="mailto:inforce@mbox.vol.cz">Napiš mi!</A>
```

21. Uložte tento záznam.
22. Přejděte do prohlížeče.
23. Otevřete nové spojení s databází.
24. Měla by se objevit nabídka Web.
25. Klepněte na odkaz Poslat HTML blok.
26. Měli by jste vidět následující stránku:

Používání 4D Databáze na Webu

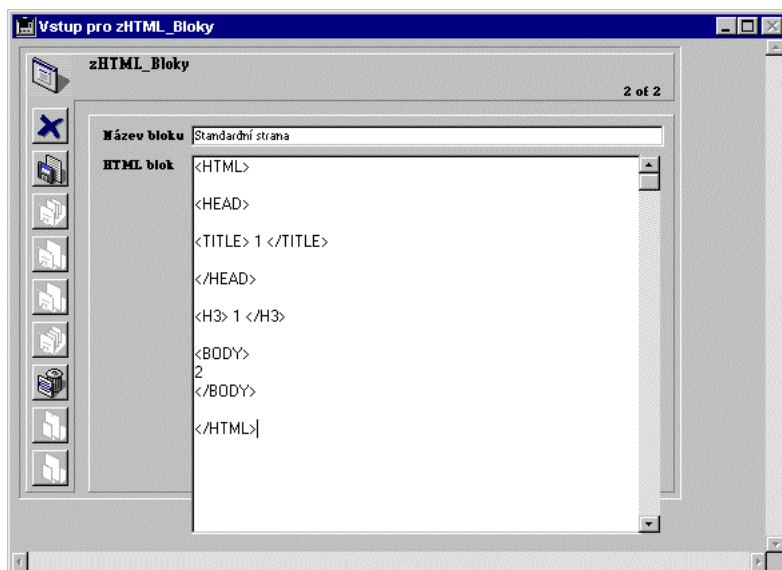
Ukládání HTML v záznamech databáze



Není to jednodušší než psát všechny HTML přes Editor metod?

18.2.3. Vzory

Jedna z nejužitečnějších položek k uložení HTML bloků je vzor. Rutina HTML_tBlock je navržena pro generické zacházení se vzory. Jestliže máte společný návrh pro skupin prvků, uložte si vzor a pak podle potřeby vytvořte jednotlivé prvky. Ne každý potřebuje vzory, takže se nebudeme zabývat použitím vzorů. Ale vysvětlíme si jak pracují a jak pracuje HTML_tBlock.



Používání 4D Databáze na Webu

Ukládání HTML v záznamech databáze

Zde je generický vzor stránky HTML obsahující místa k vložení nadpisu stránky a textu obsahu. Metoda `HTML_tBlock` je použita tak, aby našla blok a vložila jakýkoliv řetězec znaků, který si přejete. Zde jsou dvě odlišná volání a výsledky, které vytvoří:

```
$HTML_Text:=HTML_tBlock("Standard Page";"Zdravím!";"Toto je text obsahu.")
```

```
<HTML>
<HEAD>
<TITLE>Zdravím!</TITLE>
<BODY>
<H3>Zdravím!</H3>
<BODY>
Toto je text obsahu
</BODY>
</HTML>
```

```
$HTML_Text:=HTML_tBlock("Standard Page";"Moje skvělá stránka;"<B>4D je skvělá!</B>.")
```

```
<HTML>
<HEAD>
<TITLE>Moje skvělá stránka</TITLE>
<BODY>
<H3> Moje skvělá stránka </H3>
<BODY>
<B>4D je skvělá!</B>
</BODY>
</HTML>
```

Výhoda použití vzorů zmíněným způsobem vynikne pokud se rozhodnete pro určité standardní formátování HTML bloků. Nebo čehokoli co zde může být obsaženo. Jediné co pak musíte změnit jsou textová pole v příslušných záznamech. Např. jestliže chcete přidat odkaz mailto na tuto stránku, pouze změníte původní záznam vzoru:

```
<HTML>

<HEAD>
<TITLE>^1</TITLE>
</HEAD>

<H3>^1</H3>

<BODY>
^2

<A HREF="mailto:inforce@mbox.vol.cz">Otázky? Ptejte se</A>
</BODY>

</HTML>
```

Nyní pokaždé, když budete generovat stránku s tímto vzorem, bude zahrnut odkaz mailto.

Používání 4D Databáze na Webu

Ukládání HTML v záznamech databáze

18.3.4. Nahrazování řetězců (Replace string)

Několik poznámek k funkci nahrazování řetězců – Replace string:

- ❖ Replace string je rychlé. Je rychlejší nahradit několik řetězců v textovém bloku, než sestavovat celý blok znovu. Jako výchozí Replace string nahrazuje více výskytů téhož řetězce ve zdrojovém bloku.
- ❖ Jako výchozí Replace string nahrazuje více výskytů téhož řetězce ve zdrojovém bloku. Např. ^1 se vyskytuje ve zdrojovém bloku dvakrát a bude nahrazena v obou případech jedním voláním Replace string. Toto chování Replace string můžete změnit čtvrtým volitelným parametrem.
- ❖ Můžete napsat systém, který používá přímé odkazy na znaky nebo řetězce, jestliže chcete vylepšit chování.

Ve dvou případech volání jsme ilustrovali způsob, kde nahrazované hodnoty jsou řetězce písmen zahrnuté ve 4D kódu (v parametrech procedury) ve skutečné praxi, můžete dodávat hodnoty z:

- ❖ proměnných z formuláře HTML
- ❖ vkládané uživatelem.
- ❖ jiných záznamů z [zHTML_Bloky].
- ❖ jako výsledek libovolného počtu výpočtů.
- ❖ 4D funkcí jako string (Current date).
- ❖ diskových souborů.
- ❖ libovolných záznamů ve vašem systému 4D.
- ❖ jako výsledek dotazů SQL do jiných systémů.

Zobecnění kódu a vašich dat podstatně zjednoduší údržbu systému a usnadní experimentování.

18.3.5. Generické vkládání

Příkladový vzor používá generické řetězce jako ^1 a ^2 jako cíl pro REPLACE STRING. To zobecňuje chování generického vkládání. Tyto řetězce jsou zástupci dynamických hodnot, které vložíte při vytváření konečného bloku. Symbol “^” je nepovinný můžete použít jakýkoli jiný znak nebo řetězec, který vám umožní zacílit vaši náhradu řetězců přesně a jednoznačně. Dále je uveden kód kompletní metody HTML_tBlock a nebo se můžete podívat do 4D Editoru metod.

```
If (False)
` Method: HTML_tBlock (Name; String 1; ...; {String n}) —> Block
` ACI University Programming Classes
` Created By: David Adams
` Date: 3/21/97

` Purpose: Merges data with variables to write HTML used by 4D
`
```

Používání 4D Databáze na Webu

Ukládání HTML v záznamech databáze

```
` 4D Insider keyword variables:
⋄fD_Adams:=True
⋄f_Version6x40:=True

End if
` Declare parameters and their labels.
C_TEXT($0;$Result_block)
C_STRING(80;$1;$Block_name)

` Parameter indirection in declaration supports n parameters
` of same type.
C_TEXT(${2}) ` Insertion block 1 through n passed in $2 - $n

` Declare local variables
C_LONGINT($Current_parameter) ` Current input parameter in loop
C_LONGINT($Insertions_to_Make) ` Number of insertion strings passed

` nput parameter reassignment
$Block_name := $1

$Result_block:=""

QUERY([zHTML_Blocks];[zHTML_Blocks]Block_Name=$Block_name)

` Empty string if no [HTML_Blocks] found.
$Result_block := [zHTML_Blocks]HTML_Block

If ($Result_block # "") ` No need to substitute on an empty string.

` How many insert parameters were passed?
$Insertions_to_make := Count parameters -1
For ($Current_Parameter;1;$Insertions_to_make)
` Huh? Imagine that we're processing $2, the next line would effectively be:
` $Result_block:= Replace string($Result_block;"^1";$2)
` In other words, it looks in $Result_block for the string ^1 and replaces it
` with the value from $2.
$Result_block := Replace string($Result_block;"^"+
String($Current_Parameter);${$Current_Parameter+1})
End for

End if ` ($Result_block # "")

UNLOAD RECORD([zHTML_Blocks])

$0 := $Result_block

` End of method.
```

18.3. O metodě HTML_tBlock

Metoda HTML_tBlock využívá s výhodou **přesměrování** pomocí parametrů. 4D vám dovolí vložit neznámý počet parametrů téhož typu do metody. Tyto parametry musí být na konci v seznamu parametrů,

Používání 4D Databáze na Webu

Ukládání HTML v záznamech databáze

funkce Count parameters určí jak mnoho parametrů bylo předáno, a na každý se odkazujeme syntaxí pro nepřímé parametry:

`${číslo parametru}`

Tento výraz je obvykle nazýván generický parametr. Tak `${1}` odpovídá parametru 1, atd. Protože počet může být stanoven do proměnné, dovolí vám to vytvořit smyčku přes všechny předané parametry, což je to co se děje v metodě HTML_tBlock. Když používáte proměnnou ve smyčce syntaxe vypadá hlubokomyslnější, ale není víc než:

`${Tento_parametr}`

`${cokoliv}` znamená “vezmi parametr tohoto čísla”. Číslo zde přichází z čítače smyčky.

18.4. Optimalizace rutiny

Optimalizovat lze systém HTML_Block umístěním nejužívanějších bloků do array při spuštění. Pak když je blok požadován, zkontrolujete array před dotazem na záznam z disku. Jestliže máte dostatek RAM pro podporu tohoto rysu, můžete vylepšit chování při zavádění bloků v řádech.

18.5. Praktický příklad

Nyní praktický příklad. Použijme HTML bloky k vlastnímu účelu.

1. Otevřete formulář [Produkty];"Produkty.iw"
2. Vymazejte všechna pole a popisky z formuláře.
3. Přidejte do formuláře nedostupnou proměnnou.
4. V události Při zavedení použijte HTML bloky k vytvoření zajímavého textového popisu, ukazujícího všechna data z původního formuláře a odložte všechna data do proměnné téhož názvu jako v kroku 3 výše.
5. Podívejte se na výsledek ve vašem prohlížeči WEB.

Používání 4D Databáze na Webu

Speciální poznámky ke kódování znaků HTML

19. Speciální poznámky ke kódování znaků v HTML

19.0.1. Popis

HTML může být použita s libovolnou sadou z několika znakových sad, včetně dvou-bytových sad jako je japonština. Nejběžnější sadou je dnes "Latin-1", která je založena na standardu ISO 8859-1.

Je to sada 256 znaků schopných zobrazit text z libovolného jazyka západní Evropy a některých z východní Evropy.

V českém verzi je tato sada upravena v tzv. Latin2 standardu ISO 8859-2. a další použitou sadou je Win1250.

19.0.2. Latin, Win1250 a 4D

Když jste uvnitř originální 4D a Web připojení 4D automaticky převádí text z a do Latin1, přizpůsobená česká 4D konvertuje do a z Win 1250. Na požádání lze však dodat i Latin2. Jestliže potřebujete převést statickou stránku má 4D pro tento účel na MAC příkaz **Mac to ISO**, který převádí do vnitřně nastavené výše zmíněné sady.

19.1. Poznámka o názvech souborů

4D je tolerantní ke znakům, které můžete použít v názvech souborů obsluhovaných SEND HTML FILE. Všechny další Web Servery jsou striktnější. Jestliže používáte 4D k vytvoření statického HTML dokumentu pro jiný Web server musíte zakódovat speciální znaky v názvech souborů do Hex. Speciální znaky zahrnují mezeru, lomítka, znak dolaru atd. Prohlížeči řeknete, že URL je v hex, když předchází znak procent. Následující tabulka ukazuje některé znaky kódované do hex. Je zde správné hexadecimální kódování, které musí být použito pro jiné servery:

Znak	Kódování	Znak	Kódování
Tab	%09	}	%7D
Space	%20		%7C
"	%22	~	%7E
<	%3C	%	%25
>	%3E	/	%2F
[%5B	#	%23
]	%5D	\$	%24
\	%5C	?	%3F
^	%5E	=	%3D
`	%60	&	%26

Používání 4D Databáze na Webu

Speciální poznámky ke kódování znaků HTML

{	%7B		
}	%7D		

Tabulka uzpůsobena z Wiederspan and Shotton (1996) str 66-67.

Používání 4D Databáze na Webu

Práce s JavaScripty

20. Práce s JavaScripty

20.1. Co je JavaScript?

JavaScript je skriptovací jazyk vytvořený Netscape Communications Corporation pro jejich prohlížeče a servery. Microsoft implementoval podporu pro JavaScripty ve svých novějších verzích prohlížečů Internet Explorer. Microsoft rovněž nabízí VBScript formát v jeho prostředí Visual Basic jako alternativu pro JavaScript.

20.2. Je JavaScript částí Java?

JavaScript není formou jazyka Java, není jeho částí a není v něm ani napsán. Je lepší přemýšlet o JavaScript jako "NetscapeScript". Původně byl JavaScript nazýván LiveScript jakou součást produktové řady Netscape's LiveWire. Název JavaScript je více výsledkem marketingového výběru než technické podobnosti mezi JavaScript a Java. Netscape spolupracoval se Sun, tvůrcem jazyka Java na uzpůsobení LiveScript k jazyku Java. To je důvod proč JavaScript sdílí mnoho slov a mnoho ze Syntaxe z jazyka Java.

20.3. Proč užívat JavaScript?

JavaScript hraje velmi důležitou roli v systémech Web: dává vám schopnost provádět kód v prohlížeči. Tradiční prohlížeče byly zcela pasivní zařízení, nechali vás pouze požádat o informaci a prohlédnout si ji. Všechno uživatelské zpracování a kód museli být ošetřovány servery někde na síti. Tento model pracuje dobře pro statické stránky, ale je zcela nevhodný pro databázové aplikace. Bez JavaScript nebo jiného skriptovacího systému nemohou standartní HTML ověřit zda požadovaná pole byla vyplněna, zkontrolovat, že vložené hodnoty jsou uvnitř správného rámce nebo interagovat s uživatelem. Uživatel musí odeslat formulář na server, který jej ověří a pak čekat na výsledné případné chybové hlášení. Představte si, že vaši uživatelé 4D by museli vyplnit celý vstupní formulář, klepnout na tlačítko enter, předtím než by se dozvěděli, že vyplnily svá pole nekompletně nebo nesprávně (a nakonec uviděli zprávu "Spojení odmítnuto, server je zaneprázdněn, zkuste později!") toto je neúčinné uspořádání a vede k frustraci uživatelů.

20.4. Užití JavaScript v systémech 4D Web

Ve vašich 4D Web aplikacích nepotřebujete používat JavaScript a to z toho důvodu, že 4D určí proces pro každou sekci Web připojení a uživatel se nemusí obávat, že server bude zaneprázdněn, pokud nebudou čekat příliš dlouho s odpovědí. Když uživatel odešle formulář, budou metody 4D formuláře a jeho objektů provedeny uvnitř 4D a bude uskutečněna jakákoliv akce, kterou si přejete včetně ověření hodnot polí. To je funkční řešení v celku bez práce, ale jak jsme již poukázali není to ideální rozhraní pro ověřování dat rutin. Ve svých 4D Web systémech můžete použít jednoduché JavaScripty pro:

- ❖ Vytvoření speciálních efektů, které nemají nic společného se 4D.
- ❖ Provedení kontroly mezních hodnot polí.
- ❖ Zajištění vstupu do požadovaných polí.
- ❖ Ověření formuláře před odesláním.
- ❖ Interakci s uživatelem.

Používání 4D Databáze na Webu

Práce s JavaScripty

V6 všechno toto lehce umožní, podívejme se jak.

20.5. O těchto cvičeních

Tento kurs není kursem JavaScript a my si nečiníme nárok být považováni za experty v JavaScript. JavaScript je úplný jazyk a jestliže jej chcete používat vyžaduje plné studium. V současnosti neexistují žádné podrobné příručky pro JavaScript. Jednou z dobrých příruček je **JavaScript Sourcebook** od Gordon McComb, vydaná Wiley and Sons. Mnoho užitečných informací a učebnic najdete také na Web.

V těchto cvičeních se chystáme předvést jak můžete JavaScript vkládat do vašich 4D formulářů a co s nimi můžete dělat. Zahrnuli jsme sem několik jednoduchých JavaScriptů, ale hlavním účelem těchto cvičení je ukázat vám co je možné.

20.6. 4D a JavaScript

Automatizovaná podpora pro JavaScript bude pravděpodobně vzrůstat s novými verzemi 4D. V tomto okamžiku můžete přidat vlastní JavaScript do libovolného formuláře, ale samotný JavaScript musíte napsat. Když do 4D formuláře zahrnete JavaScript, 4D pro vás stále tento formulář automaticky převede. Přidáním JavaScript se nevzdáte žádného z automatických rysů 4D. Tento JavaScript nemá žádný efekt v samotném prostředí 4D, ale přidá nové funkce do HTML formulářů vytvářených 4D, samozřejmě pokud je používat s prohlížeči podporujícími JavaScript.

Poznámka: 4D programátoři musí mít na paměti, že JavaScript je citlivý na rozdíl mezi velkými a malými písmeny, na rozdíl od samotné 4D. Pokud si to včas neuvědomíte může vám to způsobit značné problémy. Samozřejmě si určitě nyní myslíte, že se vám to nemůže stát, ale o takových případech víme...

20.7. Odeslání JavaScript

Zahrnutí JavaScript do 4D formuláře je jednoduché. Je to pouze jedno použití vnoření HTML. Zde máme jednotlivé kroky:

1. Vložte do formuláře nedostupnou textovou proměnnou.
2. Použijte metodu k vložení znaku ASCII (1) do této proměnné.
3. Přidejte do proměnné jakýkoliv JavaScript nebo HTML.
4. Pošlete formulář přes Web pomocí, DIALOG, ADD RECORD, MODIFY SELECTION nebo DISPLAY SELECTION.

Jestliže používáte tento formulář v procesu Web. JavaScript je automaticky vložen do zdrojového textu HTML. Hovořme nyní trochu o JavaScript a pak se podívejme na příklad použití.

20.8. Jak vypadá JavaScript

HTML syntaxe pro zahrnutí JavaScript je jednoduchá. Zde máme příklad:

```
<SCRIPT LANGUAGE="JavaScript">
<!--Zde napište kód JavaScript -->
</SCRIPT>

<NOSCRIPT>
<!--Zde napište zprávu pro prohlížeče nepodporující JavaScript -->
</NOSCRIPT>
```

Zde máme malý příklad:

Používání 4D Databáze na Webu

Práce s JavaScripty

```
<SCRIPT LANGUAGE="JavaScript">
<!--Vyhněte se provedení tohoto kódu ve starém prohlížeči.
function buttonAlert (whatToSay) {
    alert (whatToSay)
}
// Hotovo uschovíme skript před starými prohlížeči. -->
</SCRIPT>

<NOSCRIPT>
<P><I>Potřebujete prohlížeče podporující JavaScript, aby jste využili zvláštní rysy této stránky
</I><P>

</NOSCRIPT>
```

JavaScripts je pro váš dokument zahrnut uvnitř tagů `<SCRIPT></SCRIPT>`. Tagy `<NOSCRIPT></NOSCRIPT>` vám umožní uložit zprávy pro uživatele, kteří mají prohlížeče nepodporující JavaScript. Ujistěte se, že jste z důvodů dobrých mravů zahrnuli NOSCRIPT a zabránili tak zmatení některých svých uživatelů.

Neobávejte se zde příliš syntaxe JavaScript v několika minutách si naimportujeme předem připravenou rutinu, která se postará o sestavování tagů JavaScript automaticky.

20.9. Ukládání JavaScript

Nejúčinnější způsob pro ukládání vnořených JavaScript je vytvořit si novou tabulku, která uloží vaše JavaScript. JavaScript je psán v normálním textu, takže jeho uložení v textovém poli 4D záznamu je jednoduché. Vložení JavaScript do záznamů vám umožní je jednoduše měnit, exportovat, tisknout a používat. Rovněž to zjednoduší vytvoření metody pro umístění JavaScript uvnitř správných HTML tagů. Přidejte následující pole a vlastnosti:

[zJavaScripty]		
Název pole	Typ	Vlastnosti
NázevScriptu	String (80)	Indexované, Jedinečné, Povinné
ZdrojScriptu	Text	Povinné
Text pro bezScriptu	Text	

Tato tabulka ukládá vaše JavaScripty a výběrově zprávy pro uživatele, kteří nemají prohlížeče podporující JavaScript.

20.10. Přidání formulářů do tabulky

Jděte pro Prostředí uživatele a nechte 4D aby automaticky vytvořila formuláře. Tyto formuláře nepotřebujete Web Aware, protože v tomto kurzu budeme všechny úpravy provádět uvnitř 4D. Ani neexistuje důvod k tomu, aby jste mohli upravovat vaše JavaScript přes Web. Nyní se vraťte zpět do Prostředí návrháře a upravte vstupní formulář.

Používání 4D Databáze na Webu

Práce s JavaScripty

20.10.1 Vytvoření a úpravy výchozích formulářů

1. U obou textových polí zobrazte posuvníky.
2. Každé z těchto textových polí udělejte 400 bodů široké a 100 bodů vysoké.
3. Umístěte obě textová pole tak, aby se nepřekrývala.

Jste hotovi s úpravou formuláře!

20.11. Váš první JavaScript

Nyní přidejte JavaScript do tabulky.

20.11.1 "Napsání" JavaScript

1. Jděte do Prostředí uživatele.
2. Přidejte nový záznam do tabulky zJavaScript.
3. Pojmenujte JavaScript "Ukázat prvky formuláře" (bez uvozovek..
4. Otevřete textový soubor JS_Show z vašich školících materiálů.
5. Překopírujte obsah JS_Show a vložte jej do pole Skript_zdroj.
6. Uložte záznam.

Právě jste vložili jednoduchý JavaScript, který automaticky napíše popis formuláře generovaného 4D na patřičnou stránku Web. Za chvíli ho uvidíme v akci.

20.12. Rutina tvorby JavaScript

Máte tabulku pro uložení JavaScript a svůj JavaScript. Nyní potřebujeme rutinu, která převede data z této tabulky do správně formátovaného JavaScript.

20.12.1. Úpravy databáze a odeslání JavaScript

1. Vytvořte novou metodu projektu WEB_MakeJavaScript importem z textového souboru WEB_MakeJavaScript použitím textového editoru.

```
If (False)
  ` WEB_MakeJavaScript (String (80)) : Text
  ` WEB_MakeJavaScript (String (80)) : JavaScript block
  `
  ` This routine loads a [zJavaScripts] record and formats it
  ` s a JavaScript script.
  `
  ` Note: You can easily rewrite this routine with *parameter indirection* to
  ` combine any number of JavaScripts records into single script.
  `
  ` 4D Insider keyword variables:
  <>fD_Adams:=True
  <>f_Version6x40:=True

End if
```

Používání 4D Databáze na Webu

Práce s JavaScripty

```
C_TEXT($0;$tScript)
C_STRING(80;$1;$sJavaScriptName)

$sJavaScriptName := $1

` Open the script. This line produces a statement like this:
` <-- JavaScript Definitions: -->
` <SCRIPT LANGUAGE="JavaScript">
$tScript := $tScript+"<!-- JavaScript Definitions: -->"&Char(13)
$tScript := $tScript+"<SCRIPT
LANGUAGE="&Char(34)&"JavaScript"&Char(34)&">"&Char(13)
$tScript := $tScript+"<!-- Avoid having this code displayed in old browsers."&Char(13)
MESSAGES OFF
QUERY([zJavaScripts];[zJavaScripts]Script_name=$sJavaScriptName)
MESSAGES ON

If (Records in selection([zJavaScripts])=0)
    $tScript := $tScript+"// JavaScript ""&$sJavaScriptName&"" not found."
Else
    $tScript := $tScript+[zJavaScripts]Script_source
End if

` Now close the script:
$tScript := $tScript&Char(13)&"// Done hiding script from old browsers. -->"&Char(13)
$tScript := $tScript+"</SCRIPT>"&Char(13)&Char(13)

` Now add the NOSCRIPT section for browsers without JavaScript enabled or
support
$tScript := $tScript+"<NOSCRIPT>"&Char(13)

If ([zJavaScripts]No_script_text="")
    ` Empty field or no current record, add the default NOSCRIPT message:
    $tScript := $tScript+"You need a JavaScript enabled browser to use the JavaScript on
this page."
Else
    ` Add the custom NOSCRIPT message from this script definition:
    $tScript := $tScript+[zJavaScripts]No_script_text
End if

$tScript := $tScript&Char(13)&"</NOSCRIPT>"&Char(13)

UNLOAD RECORD([zJavaScripts])

$0 := $tScript

` End of method
```

2. Uložte tuto metodu.

K vytvoření JavaScript je potřeba pouze rutina WEB_MakeJavaScript s názvem jednoho z JavaScript ve vašem záznamu zJavaScripts

Používání 4D Databáze na Webu

Práce s JavaScripty

```
$My_JavaScript := Web_MakeJavaScript("Ukázat prvky formuláře")
```

20.13. Přidání JavaScript do formuláře

Nyní přidejme JavaScript do formuláře. První věc, kterou chceme je formulář Web Aware. Budeme pracovat s tabulkou Zákazníci a to ve všech dalších cvičeních.

20.13.1. Úprava formuláře Zákazníci pro JavaScript

1. Vytvořte nový vstupní formulář Web aware pro tabulku Zákazníci nazvaný Zákazníci.js
2. Zahrňte následující pole v tomto pořadí (pořadí je důležité)

Firma

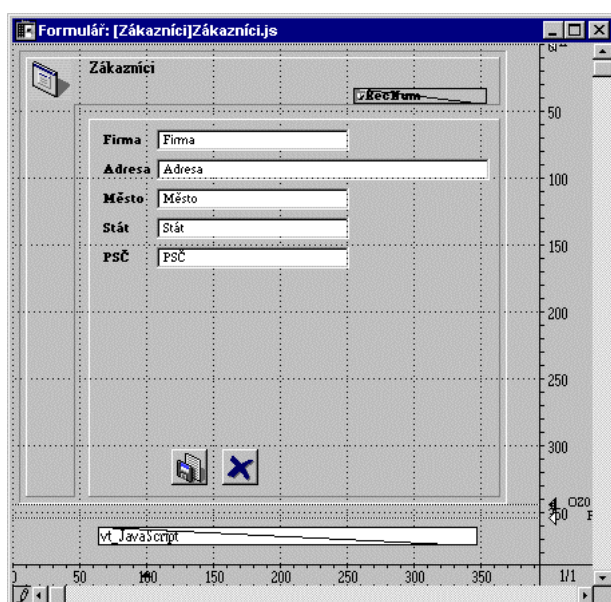
Adresa

Město

Stát

PSČ

3. Užijte typ tlačítka Běžná tlačítka – malá a zahrňte pouze tlačítka OK a Storno.
4. Upravte formulář.
5. Přidejte nedostupnou oblast a nazvěte ji vt_JavaScript.
6. Formulář by měl vypadat následovně:



Nyní můžete použít tento formulář s a nebo bez JavaScript a s libovolný JavaScriptem, který potřebujete. Před odesláním formuláře Web klientovi, přidáte JavaScript do proměnné vt_JavaScript. Nastavme testovací rutinu a vyzkoušejme.

Používání 4D Databáze na Webu

Práce s JavaScripty

20.14. Testovací rutina

Použijeme jednoduchou testovací rutinu, která naplní JavaScript ze záznamu, pro všechny další příklady

1. Jděte do Prostředí návrháře ve 4D.
2. Otevřete metodu M_Web_ShowCustomers.
3. Přidejte následující řádky kódu navrch metody:

```
vt_JavaScript:=Char (1)
vt_JavaScript:=vt_JavaScript+WEB_MakeJavaScript ("Ukázat prvky formuláře")
```

4. Změňte vstupní formulář, přiřazením našeho formuláře s JavaScript:

```
INPUT FORM([Zákazníci];"Zákazníci.js")
```

5. Zbytek metody ponechte nezměněný.
6. Metoda by měla vypadat následovně:

```
vt_JavaScript:=Char (1)
vt_JavaScript:=vt_JavaScript+WEB_MakeJavaScript ("Ukázat prvky formuláře ")

INPUT FORM([Zákazníci];" Zákazníci.js")
OUTPUT FORM([Zákazníci];" Zákazníci.ow")
ALL RECORDS([Zákazníci])
SET WEB DISPLAY LIMITS(5) ` Show only 5 records at a time.
MODIFY SELECTION([Zákazníci];*)
```

7. Uložte metodu.

Obzvlášť důležitý je počáteční znak Char (1), pokud není přítomen, 4D interpretuje vkládaný text jako data a váš JavaScript nebude pracovat. Nyní můžeme otestovat JavaScript na existujícím systému zobrazení zákazníků z Web menu.

8. Přejděte do prohlížeče Web.
9. Zobrazte Web nabídku.
10. Vyberte odkaz Ukázat zákazníky
11. Upravte jeden záznam zákazníků.
12. Nyní by jste měli vidět váš nový vstupní formulář a text napsaný JavaScript popisujícím prvky HTML formuláře vytvořeného 4D.
13. Zrušte tento záznam zákazníků a vraťte se do hlavní nabídky Web.

Informace na spodu obrazovky popisuje formulář tak, jak jej vidí JavaScript. Všimněte si, že první pole je očíslováno 0 nikoli 1. JavaScript, který jste zahrnuli, může být použit k popsání libovolného 4D formuláře tímto způsobem. Jestliže jste se rozhodli zahrnout do svého systému JavaScript, tento jednoduchý systém utilit vám ušetří mnoho času. Číslo prvků zobrazená nahoře jsou klíčem ke psaní JavaScript, které interagují s vaším 4D formulářem.

Poznámka: Jestliže skript nepracuje nebo vidíte zprávu NOSCRIPT, pak nemusíte mít umožněny JavaScript. Proto, aby tento příklad pracoval potřebujete prohlížeč, který pracuje s JavaScript jako Netscape 3.0. a pozdější a JavaScript musí být zapnutý.

Používání 4D Databáze na Webu

Práce s JavaScripty

20.15. Objekty JavaScript

Jestliže chcete pracovat s JavaScript je několik věcí, které by jste měli vědět:

- ❖ Pro každý dokument (stránku) JavaScript udržuje array popisující každý formulář
- ❖ Pro každý formulář JavaScript udržuje array popisující každý prvek. To je to co skript, který jsme právě vytvořili zobrazoval.
- ❖ JavaScript array jsou číslovány od 0 ne od 1.
- ❖ Formulář generovaný 4D je formulář číslo 0, protože je to první formulář.

Nyní se podívejme na několik praktických ukázek využití této informace.

20.16. Změny 4D polí

Jedno jednoduché užití JavaScript je k prověřování dat ve 4D polích. Podívejme se jak přidat JavaScript, který zajišťuje, že pole adresy a města mají hodnoty a že pole státu je velkými písmeny.

20.16.1. Testování JavaScript-ověřování polí

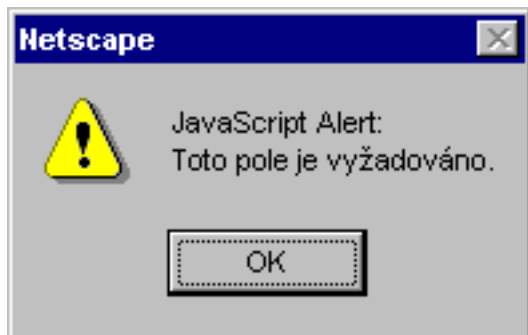
1. Jděte do Prostředí uživatele.
2. Přidejte nový záznam do tabulky zJavaScript.
3. Pojmenujte skript "Ověření polí" (nezahrnujte uvozovky).
4. Otevřete textový soubor nazvaný Customer field validation ze školících materiálů.
5. Překopírujte jeho obsah do pole zdrojového textu.
6. Uložte záznam.
7. Přejděte do Prostředí návrháře.
8. Upravte rutinu M_Web_ShowCustomers tak aby žádala o JavaScript "Ověření polí"

```
vt_JavaScript:=Char(1)
vt_JavaScript:=vt_JavaScript+WEB_MakeJavaScript ("Ověření polí")
```

9. Uložte metodu.
10. Přejděte do prohlížeče.
11. Zobrazte a upravte záznam zákazníka.
12. Vymažte pole adresy a stiskněte tabelátor.
13. Uvidíte upozornění od JavaScript a kursor se vrátí do pole, které jste opustili.

Používání 4D Databáze na Webu

Práce s JavaScripty



14. Stiskněte OK. Právě jste viděli ekvivalent 4D příkazu REJECT.
15. Napište něco do pole adresy.
16. Tabeľátorem přeskočte na pole Stát a vložte hodnotu malými písmeny.
17. Stiskněte tabeľátor.
18. Hodnota v poli se změní na velká písmena.
19. Zrušte tento záznam a vraťte se zpět do nabídky Web.

S tímto JavaScript nepotřebujete odesílat daný formulář do 4D pro provádění rutin na úrovni poli zajišťující ověřování. Podívejme se stručně na část tohoto JavaScript.

```
function errorIfEmpty (formNumber, fieldNumber) {  
    if (document.forms[formNumber].elements[fieldNumber].value == "") {  
        alert('Toto pole je vyžadováno.')        document.forms[formNumber].elements[fieldNumber].focus()  
    }  
}  
  
function checkAddress () {  
    errorIfEmpty (0, 1)  
}  
  
// Attach event handlers to some of the fields.  
// NOTE: Direct event handler calls only works with Netscape 3.0 or later.  
document.forms[0].elements[1].onblur=checkAddress    // Address
```

Sekce “function errorIfEmpty” kódu definuje standartní rutinu nazývanou errorIfEmpty, kterou mohou volat další funkce JavaScript. Všechny rutiny JavaScript se nazývají funkce dokonce i když nenavracují hodnotu. Funkce checkAddress volá funkci errorIfEmpty pomocí určitého čísla formuláře a čísla prvku. Pamatujte si, že formulář 4D je vždy číslo 0 a že prvek pole je vždy pozice pole na formuláři -1. Výraz `document.forms[0].elements[1].onblur=checkAddress` přiřazuje funkci checkAddress jako hodnotu, která může jít do vlastnosti onBlur druhého pole z prvního formuláře. Jinými slovy přiřazuje checkAddress k adresovému poli jako akci, kdykoliv uživatel vychází z tohoto pole tabeľátorem, dokonce i tehdy jestliže nemění data. Neobávejte se těchto podrobností jdou rozhodně za rámec tohoto kurzu. Jenom jsme se na ně podívali a stručně je okomentovali, aby to pomohlo těm, kteří budou chtít po skončení kursu integrovat JavaScript do vašeho systému.

Používání 4D Databáze na Webu

Práce s JavaScripty

20.17. Zachycení odeslání formuláře

Občas můžete chtít zachytit odeslání formuláře k provedení celkového testu několika polí. Nyní použijeme JavaScript, který pracuje s odesláním. Nedělá nic moc užitečného, ale ukazuje jak se to dělá.

20.17.1. Testování odeslání formuláře v JavaScript

1. Jděte do Prostředí uživatele.
2. Přidejte nový záznam do tabulky zJavaScript.
3. Pojmenujte skript “Odeslání” (bez uvozovek).
4. Otevřete textový soubor Customer onSubmit z vašich školících materiálů.
5. Překopírujte obsah souboru Customer onSubmit do pole Skript_zdroje
6. Uložte záznam.
7. Jděte do Prostředí návrháře.
8. Upravte rutinu M_Web_ShowCustomers tak aby žádala o JavaScript “Odeslání”.

```
vt_JavaScript:=Char(1)
vt_JavaScript:=vt_JavaScript+WEB_MakeJavaScript ("Odeslání")
```

9. Uložte metodu.
10. Přejděte do prohlížeče.
11. Ukažte zákazníky a jednoho upravte.
12. Uvidíte tentýž formulář jako v předchozím příkladě, pouze skripty polí jsou pryč a tlačítka Odeslat a Storno mají skripty.
13. Zkuste stisknout tlačítko Storno a zobrazí se vám hlášení JawaScriptu.
14. Uzavřete toto upozornění a vraťte se do nabídky Web.

Můžete přerušit odesílání záznamu, zastavit jej nebo dovolit pokračování, vše s užitím JavaScript.

Poznámka: K přiřazení akcí Při odeslání potřebujete skutečný odesílací objekt, když užíváte skutečná 4D tlačítka, jsou odesílací objekty vytvářeny a vše pracuje dobře, když použijete obrázková tlačítka 4D, jsou vytvořena grafická tlačítka HTML, která nepřijmou metodu Při odeslání. Jestliže chcete zachycovat vstupy do formuláře, ujistěte se, že používáte běžná tlačítka ve 4D formulářích.

20.18. Otevření nového okna pro nový proces

Jiná užitečná věc pro JavaScript je otevření nového okna pro nový 4D proces.

20.18.1. Použití JavaScript k otevření nového okna pro nový proces

1. Jděte do Prostředí uživatele.
2. Přidejte nový záznam do tabulky zJavaScript.
3. Pojmenujte skript “Otevři okno” (bez uvozovek).
4. Otevřete textový soubor nazvaný Open About ve vašich školících materiálech.
5. Zkopírujte jeho obsah a vložte jej do pole Skript_zdroj.
6. Uložte záznam.

Používání 4D Databáze na Webu

Práce s JawaScripty

7. Jděte do Prostředí návrháře.
8. Upravte rutinu M_Web_ShowCustomers tak, aby žádala o JavaScript“ Otevři okno ”.

```
vt_JavaScript:=Char(1)
vt_JavaScript:=vt_JavaScript+WEB_MakeJavaScript ("Otevři okno ")
```

9. Uložte metodu.
10. Jděte do prohlížeče.
11. Zobrazte a upravte záznam jednoho zákazníka.
12. Ve vašem prohlížeči se objeví nové okno s informacemi o databázi.
13. Zavřete okno s informacemi o databázi.
14. Když jste hotovi uzavřete záznam a vraťte se do Web nabídky.

Pokaždé, když otevřete nové okno v prohlížeči pomocí JavaScript a zavoláte 4D metodu, vytvoří se nová sekce Web (to znamená je použito další Web připojení). Všechny vlastnosti nového okna můžete řídit přes JavaScript, výšku, šířku, řídicí prvky atd. S pojmenovaným oknem vytvořeným JavaScript jako bylo toto, můžete mít dokonce okna, která spolu navzájem interagují. Tuto techniku můžete např. použít pro zobrazení palet nástrojů.

Používání 4D Databáze na Webu

Triggery

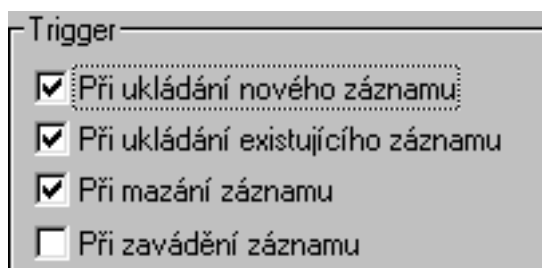
21. Triggery

20.1. Úvod

V tomto kursu se nebudeme triggery nijak podrobně zabývat. Co se zde chystáme vysvětlit je jak vytvářet vaše triggery s respektováním Web připojení.

20.2. Co je trigger?

Trigger je metoda přiřazená tabulce, která se provádí při událostech spojených s úpravami dat. Tyto události jsou běžně “vložit, upravit a vymazat”. 4D má dělení o něco podrobnější na čtyři události:



Triggery jsou umožňovány pomocí vlastností tabulky. Když je trigger umožněn a příslušná událost se stane z jakéhokoliv důvodu, trigger se spustí. To znamená, že jestliže máte umožněn trigger Při ukládání nového záznamu a novým záznam je ukládán, provede se kód triggeru. Nezáleží na tom jaký formulář uživatel používá, nezáleží na tom zda je připojen přes Web prohlížeče, 4D Client nebo přes 4D Open. Triggery jsou tím správným místem, kde soustřeďovat všeobecná pravidla. Jinými slovy: triggery jsou správné místo pro zajištění integrity dat. Správné použití triggerů zahrnuje:

- ❖ Ověření jedinečnosti.
- ❖ Ověření zda byly vloženy povinné údaje.
- ❖ Aktualizace výpočtových dat.
- ❖ Přidání, úpravy a vymazání vztažených záznamů.

Triggery nejsou správným místem pro řízení určitých prvků rozhraní, to je úloha pro vaše metody formulářů a objektů formulářů.

20.3. Jednoduchý trigger

Triggery jsou prostředkem pro zajištění pravidel pro vaše data nikoli rozhraní. Když vytváříte vaše triggery, ujistěte se, že je píšete tak, že budou podporovat libovolnou platformu. V předchozích příkladech jsme odstranili trigger tabulky Zákazníci, nyní jeden přidáme zpět.

1. Otevřete trigger tabulky Zákazníci.
2. Upravte existující kód, aby zahrnoval převod pole společnosti do velkých písmen:

` Customers Trigger

Používání 4D Databáze na Webu

Triggery

C_LONGINT(\$0;\$Error_code) ` Error code.

\$Error_code:=0 ` Default to no error condition.

If (Database event=Save New Record Event) | (Database event=Save Existing Record Event)
[Zákazníci]Společnost:=Uppercase([Zákazníci]Společnost)
End if

\$0:=\$Error_code

3. Uložte tuto metodu.
4. Přejděte do prohlížeče Web.
5. Zobrazte seznam zákazníků.
6. Upravte jeden záznam.
7. Uložte záznam.
8. Otevřete tentýž záznam znovu.
9. Uvidíte, že název společnosti je nyní velkými písmeny.

Povšimněte si, že událost, kterou jsme kontrolovali je událost databáze ne formuláře. Triggery by vždy explicitně měli testovat, události databáze.

Poznámka: Nikdy neměňte platný záznam tabulky v triggeru tzn. nesmíte v platné tabulce provádět dotazy, push, cancel, nebo save na platný záznam.

20.4. Psaní triggerů pro libovolný interface

Protože triggery mohou být prováděny v závislosti na akcích libovolného klienta nebo metody. Je důležité vyhnout se prvkům rozhraní, které jsou nevhodné. Jestliže instalujete metodu ON ERR CALL k zachycení chyb generovaných triggerem, je to bezpečné užití i na Web. Jestliže použijete vaše vlastní dialogové okno a otevřete okno ujistěte se, že neotevíráte okno, když uživatel je klient Web, protože toto okno se otevře na stroji se 4D. To je jiný důvod proto, že musíte neustále sledovat typ prováděného spojení.

20.5. Specifické položky Web

Pamatujte si, že všechna tlačítka, která odesílají vaše formuláře, spouštějí patřičný trigger. Jedno místo, kde jsme to zmínili a které se uvnitř 4D neprovádí jsou změny stránek formulářů. Jak již bylo řečeno, když měníte stránku formulářů ve formuláři Web, daný formulář a jeho záznam je odesílán. 4D provede trigger pro uložení záznamu, zrovna tak jak kdyby záznam byl přijat. Jestliže to není to co chcete, musíte kontrolovat ve vašem triggeru, kdy jej provádí uživatel připojený přes Web a jestli to není pouhé klepnutí na tlačítko přepnutí stránky.

Používání 4D Databáze na Webu

Shrnutí Příkazů Web

22. Shrnutí Příkazů Web

Toto je krátký přehled použitých příkazů pro Web ve v6. Každý z těchto příkazů byl probrán v odpovídající sekci této příručky, a dále je popsán v příručce 4D Language reference.

22.1. SEND HTML FILE (Soubor cesta)

Tento příkaz odešle příslušný soubor prohlížeči. Cesta je relativní k umístění struktury databáze, pokud ji nezměníte příkazem SET HTML ROOT. Symbol pro oddělování složek je “/” nezávisle na platformě.

22.2. SET HTML ROOT (Cesta)

Tento příkaz nastaví výchozí cestu k HTML. Cesta je určována relativně k databázové struktuře. Symbol k použití pro oddělovače je “/” nezávisle na platformě.

22.3. SET WEB DISPLAY LIMITS(Množství;Stránek;Ikona)

Tento příkaz vám dovolí nastavit několik možností Web, které působí na MODIFY SELECTION a DISPLAY SELECTION. Když jsou záznamy odesílány jsou automaticky stránkovány tak, že síť ani uživatel není nadměrně zatížen. Množství určuje počet záznamů na jedné stránce, Stránek určuje jak mnoho stránek je drženo v paměti, Ikona určuje grafiku, která bude použita k odkazu na konkrétní záznam. Jestliže určíte množství 20 a stránek 10, může uživatel volně přecházet mezi 200 záznamy ve výběru. Odkaz v ikoně je vnitřní obrázkový odkaz 4D na obrázkovou knihovnu, a to číslem obrázku.

22.4. SET WEB TIME-OUT (Sekund)

Nastaví čas vypršení spojení — v sekundách — pro nový Web proces. Užijte 0 pro ‘žádný limit’.

22.5. START WEB SERVER ()

Nastartuje proces jádra WEB serveru, jestliže je databáze nastavena pro služby WEB.

22.6. STOP WEB SERVER ()

Znemožní Web serveru startovat nová připojení. Existující procesy pokračují dokud neukončí práci a nebo nevyprší čas připojení. Po odpojení posledního připojení se ukončí i Web Server.

22.7. Mac to ISO (Mac ASCII) : Latin-1, Latin 2, Win 1250 kódování textu

Konvertuje Mac kódovaný ASCII text do sad užívaných na WEB a to ISO-8859-1 (Latin-1), ISO-8859-2 (Latin-2) nebo Win 1250 podle uzpůsobení verze.

Používání 4D Databáze na Webu

Definice termínů

23. Definice termínů

CGI (Common Gateway Interface)

“CGI” nebo “ACGI” (Asynchronní CGI) je externí aplikace, se kterou komunikuje HTTP server. CGI standart definuje jak si server a CGI vyměňují data, ale neříká nic o tom co externí program má nebo může dělat. Externí program tj. v našem případě 4D, může dělat cokoli co naprogramujete a to do doby než vrátíte výsledky způsobem srozumitelným HTTP serveru.

FTP (File Transfer Protocol)

Toto je nejpoužívanější mechanismus pro přenos souborů na Internetu.

HTTP (HyperText Transfer Protocol)

Tento protokol definuje jak si Web server a Web klient vyměňují data.

HTML (HyperText Markup Language)

HTML je na platformě a zobrazení nezávislý systém, založený na tags pro strukturované dokumenty. Jazyk obsahuje tagy stylů, které jsou interpretovány každým prohlížečem, způsobem vhodným pro daný počítač a operační systém.

IIS (Microsoft Internet Information Server)

The Microsoft Internet Information Server je součástí standartního balíku Windows NT Server. IIS obsahuje rysy FTP, Gopher a HTTP server. IIS má úplný přístup do registru NT. Nový API nazvaný ISAPI byl publikován pro IIS, aby dodavatelé mohli vytvářet DLL, které se zahrnou do IIS. ISAPIDLL od Netlink je produktem založeným na ISAPI.

Internet

Internet (s velkým “I”) globální síť slučující sítě komunikující pomocí TCP/IP. “internet” (s malým “i”) je libovolná síť slučující sítě komunikující přes jakýkoli sdílený protokol. Společnosti např. mohou mít soukromý NetWare internet.

intranet

“intranet” je nová odvozenina posledních let. Znamená služby založené na Webu běžící na soukromé síti společnosti. Jestliže používáte 4Dv6 v soukromé síti k publikování databázových dat, vytvořili jste intranet.

JavaScript

Skriptovací jazyk Netscape Communications Corporation přidávající inteligenci Web prohlížečům.

POP3 (Post Office Protocol 3)

Populární e-mail standart užívaný na Internetu.

Používání 4D Databáze na Webu

Definice termínů

SMTP (Simple Mail Transfer Protocol)

Silný a široce rozšířený mail protokol užívaný na Internetu.

SSL (Secure Socket Layer)

Jeden ze standartů pro bezpečné transakce přes Web. Tento standard získává stále větší podporu a nyní existuje v tržních produktech.

Používání 4D Databáze na Webu

Kontaktní informace

24. Kontaktní informace

Následuje seznam ACI partnerů a 4D vývojářů, zmíněných v tomto kurzu, kteří nabízejí :

- ❖ Rozšíření, které přidávají Internet nebo World Wide Web služby do 4D
- ❖ 4D databáze, které fungují jako Web, mail, nebo Internet servery

Produkty jsou řazeny abecedně podle názvů produktů

DataWave

DataWave od DataCraft přidává řadu nástrojů do libovolné databáze 4D a dovoluje tak publikovat obsah databáze na WEB v několika minutách bez programování.

<http://www.datacraft-inc.com/>

Internet ToolKit (ITK)

ITK zahrnuje více než 50 rutin, které provádějí nízkourovňový přístup k TCP/IP a UDP, rutiny se vztahem ke konverzi dat jako HTML do textu, PICT do GIF, RFC kódování datumů, BinHex soubory, MacBinary, FTP, SMTP, POP3 atd. JCA Telematique's Web system je celý založen na 4D. Tento produkt je distribuován v české republice Inforce s.r.o.

<http://www.inforce.cz>

JCA Telematique

<http://www.jca.fr/jca/itk/>

NetLink/4D

NetLink/4D dovolí 4D pracovat jako CGI s jinými Web servery na Windows a MacOS. Speciální připojení je dostupné pro vylepšení chování s WebSTAR, IIS a Netscape Web server pro Windows. Netlink ošetřuje veškerou komunikaci s Web serverem, dělení zpráv, podporuje více procesů, povolujících simultánní přístup k databázi od uživatelů pracujících s Web serverem a zahrnuje knihovny třídy Java k zjednodušení vytváření FrontEndJava k 4D datum. Netlink ovládá všechny HTTP hlavičky, podporuje všechny HTML tagy, automatizuje produkty cookie a může konvertovat obrázky do formátů PICT, GIF nebo JPEG.

Foresight Technology, Inc.

<http://www.fsti.com/productinfo/netlink.htm>

NetWings

NetWings je na 4D založený Web server s kompletní podporou SMTP, POP3, FTP, HTTP a více.

NetWings, Inc.

<http://www.netwings.com>

Používání 4D Databáze na Webu

Kontaktní informace

PDM INTERNET Tools

PDM Internet Tools podporuje nízkoúrovňové volání TCP/IP jako zasílání a přijímání řad TCP paketů, rovněž protokoly vyšší úrovně včetně FTP, SMTP, POP3, NTP a další běžně užívané UNIX protokoly. PDM Internet Tools poskytuje kompletní prostředí pro databáze, které potřebují odesílat a přijímat e-mail, přenášet soubory nebo ovládat přímou komunikaci přes TCP/IP.

Pacific Data Management, Inc.

<http://www.pdm-inc.com/>

TCP ToolKit

TCP ToolKit provádí procedurální kontrolu pro FTP, SMTP, POP3, a Finger. Rovněž podporuje protokoly jako HTTP, RPC, a telnet. TCP ToolKit rovněž podporuje aktivní a pasivní socket a odesílání a přijímání textu z celých souborů. Web server, který byl kompletně vytvořen se 4D a TCP ToolKit najdete na <http://www.careermosaic.com>. Tento nástroj bude od příští verze integrovanou částí 4D.

Dynamic Information Systems, Inc

tcpsales@dynamic-info.com

Web Server 4D

Web Server 4D je 4D-založený Web server, který nabízí kompletní Web server prostředí. Web. Server 4D ošetřuje Web stránky a podporuje klepnutelné mapy, zprávy, MIME podporu, 30 současných uživatelů, 4D database integraci, e-mailování atd.

MDG Computer Services, Inc

<http://www.mdg.com>

Publishing a 4D Database on the Web

Další zdroje

25. Další zdroje

24.1. Knihy

- ❖ **Planning and Managing Web Sites on the Macintosh** od Jon Wiederspan a Chuck Shotton
Chuck Shotton je autorem MacHTTP a WebSTAR.
- ❖ **HTML3 Manual of Style** od Larry Aronson
- ❖ **The Visual Display of Quantitative Data** od Edward Tufte
- ❖ **Envisioning Information** od Edward Tufte
- ❖ **Designing Large-Scale Web Sites** od Darrell Sand
- ❖ **Multimedia and Hypermedia, the Internet and Beyond** od Jakob Nielsen
- ❖ **Managing Internet Information Services** by Liu, Peek,
- ❖ **TCP/IP Network Administration** od Craig Hunt (O'Reilly & Associates, Inc.)
- ❖ **JavaScript Sourcebook** od Gordon McComb

24.2. Elektronické příručky

- ❖ The World Wide Web Consortium
<http://www.w3.org/pub/WWW/>
- ❖ Jon Wiederspan stránky o Internet software a standardech
<http://www.comvista.com/net/Directory.htm>.
- ❖ Pokud vyvíjíte s NetLink, Foresight's NetLink-Talk navštivte jejich stránku
(<http://www.fsti.com/productinfo/netlink.htm>).