



enTeam

Příručka

Workflow Manager

Tvorba workflow

Workflow Manager – Tvorba workflow

Leviom data s.r.o.

Jungmannova 17/3
110 00 Praha 1

Telefon +420 603 902 475
E-mail helpdesk@leviom.cz
Internet www.leviom.cz

V případě problému kontaktujte zákaznickou podporu:

Tel: +420 608 608 340
helpdesk@leviom.cz

1 Obsah

1	OBSAH.....	3
2	REJSTŘÍK.....	9
3	TVORBA WORKFLOW.....	13
3.1	CO JE TO WORKFLOW	13
3.1.1	DEFINICE ZÁKLADNÍCH POJMŮ.....	13
3.1.2	NÁVRH GRAFU WORKFLOW	15
	Rozvrh grafu	15
	Konstrukce grafu workflow	15
	Pravidla pro hrany	15
	Mechanismus zpracování workflow	16
3.2	DEFINICE PROMĚNNÝCH.....	17
3.2.1	DEFINOVÁNÍ JEDNOTLIVÝCH PRACOVNÍCH KROKŮ	17
3.2.2	URČENÍ, KDO BUDE ZPRACOVÁVAT JAKÝ KROK.....	19
3.2.3	SKRIPTY PRO HRANY	19
3.2.4	TESTOVÁNÍ WORKFLOW	19
	Testování masky	19
	Testování skriptu	19
	Testování konzistence workflow	20
3.2.5	ZMĚNY WORKFLOW.....	20
4	JAZYK SKRIPTŮ	20
4.1	CO JE SKRIPT	20
4.2	TERMINOLOGIE A KONVENCE.....	20
4.3	SYMBOLY	21
4.3.1	REZERVOVANÁ SLOVA	21
4.3.2	NÁZVY	22
4.3.3	KONSTANTY STRING.....	22
4.4	PRVKY JAZYKA SKRIPTŮ	22
4.4.1	TYPY DAT.....	22
4.4.2	KONSTANTY	22
4.4.3	PROMĚNNÉ.....	22
	Lokální proměnné.....	22
	Workflow proměnné	23

Proměnné typu list	23
Reference proměnných	23
4.4.4 OPERÁTORY	23
Aritmetické operátory	23
Boolean operátory	24
Seznamové operátory	24
String operátory	24
Relační operátory	24
Srovnání	25
4.4.5 VÝRAZY	25
Hierarchie operací	25
Syntaxe výrazů	25
4.4.6 VOLÁNÍ FUNKCÍ	26
Volání externích funkcí	26
Interní volání skriptů	26
Volání masek	26
4.4.7 PŘÍKAZY	26
4.4.8 JEDNODUCHÉ PŘÍKAZY	27
Přidělení hodnoty	27
Prováděcí příkaz	28
Příkaz návratu	28
Zobrazení dialogu	28
Spouštění procedur	28
4.4.9 STRUKTUROVANÉ PŘÍKAZY	28
Podmíněné příkazy	28
Příkazy cyklu	29
Příkaz while	29
Příkaz for	29
Příkaz foreach	29
4.4.10 OBJEKTY	30
Přehled objektů	30
Implicitní objekt muser	30
Objekt user	31
Objekt wkf	32
Objekt document	32
Objekt date	33
Rozšiřující metoda Count	35
5 STANDARDNÍ KONSTRUKCE SKRIPTU	35
5.1 PŘEHLED	35
5.2 DEKLARACE SKRIPTU	35
5.2.1 EXTERNÍ DEKLARACE	35
5.2.2 TĚLO SKRIPTU	36
Typ skriptu	36
Blok	36
Deklarace proměnných	36
Výsledek skriptu	37
5.2.3 KOMENTÁŘ	37
5.3 PŘÍKLADY	37
5.3.1 ZÁSADY TVORBY WORKFLOW	37

5.3.2	VSTUPNÍ SKRIPTY	38
	Skript Act_User.....	38
	Skript User_template	38
	Skript User_department.....	38
	Skript pracovnik.....	38
	Skript ukol_pracovnik.....	38
	Skript TeamBridge	38
5.3.3	SKRIPT HRANY	39
	Skript HranaSCHVALENO.....	39
5.3.4	PROVÁDĚCÍ SKRIPTY	39
	Skript Vypracovani_ukolu.....	39
	Skript odpoved_na_prihlasku.....	39
	Skript test_DBconnect.....	39
	Skript KH_schvaleni_faktury.....	41
5.3.5	ESKALAČNÍ SKRIPTY	41
	Skript zakaznik_nezaplatil	41
	Skript pridat_reditele	42
	Skript provest_krok	42

6 WORKFLOW DESIGNER 42

6.1	K ČEMU SLOUŽÍ WORKFLOW DESIGNER?	42
6.2	OBECNÉ FUNKCE	43
6.2.1	PŘIHLÁŠENÍ.....	43
6.2.2	ODHLÁŠENÍ.....	43
6.2.3	PRACOVNÍ PLOCHA.....	44
6.3	ADMINISTRACE WORKFLOW.....	44
6.3.1	SPRÁVA WORKFLOW.....	44
	Dialog administrace.....	44
	Vytvoření nového workflow	45
	Otevření workflow.....	45
	Uzavření workflow	45
	Uložení workflow.....	46
	Mazání workflow.....	46
6.3.2	MIGRACE WORKFLOW	46
	Export workflow	46
	Import workflow.....	47
	Konflikty objektů	48
6.3.3	OPRÁVNĚNÍ	48
	Dialog pro nastavení práv.....	49
	Oprávnění role.....	49
	Oprávnění uživatele	49
	Povolení a odepření přístupu	49
6.3.4	VERZE WORKFLOW.....	50
	Prohlížení verzí workflow	50
	Mazání verzí workflow	50
	Aktualizace verze workflow.....	50
6.4	GRAF WORKFLOW	50
6.4.1	KONSTRUKCE GRAFU WORKFLOW	51
	Počáteční uzel.....	51
	Uzly	51

Vnořené workflow	51
Koncový uzel	51
Hrany	51
Mřížka	52
Navigátor	52
6.4.2 ÚPRAVY GRAFU WORKFLOW	52
Přesouvání jednotlivých komponent	52
Vkládání stejných komponent	53
Mazání komponent	53
Zrušení akce	53
6.4.3 ŘÍZENÍ PRŮBĚHU PROCESU	53
Podmíněné větvení a paralelita	53
Synchronizační uzly	53
Zpětné hrany	53
6.4.4 KONTROLA SPRÁVNOSTI WORKFLOW	53
Seznam nalezených chyb	54
6.4.5 UKLÁDÁNÍ WORKFLOW	54
Ukládání nekonzistentního workflow	55
6.4.6 TISK	55
6.5 VLASTNOSTI WORKFLOW	55
6.5.1 OBECNÉ VLASTNOSTI WORKFLOW	56
Popis workflow	56
Grafická nastavení workflow	56
Instanciace workflow	56
6.5.2 PROMĚNNÉ VE WORKFLOW	57
Přidávání proměnných	57
Nastavení proměnných	58
Aktualizace slovníku dat	58
6.6 VLASTNOSTI KOMPONENT WORKFLOW	58
6.6.1 VLASTNOSTI PRACOVNÍHO KROKU	59
Název uzlu	59
Jméno	59
Popis	59
Nápověda	59
Ikona	60
Vstupní skript	60
Uživatel	60
Prováděcí skript	60
Provedení	61
Prováděcí maska	61
Externí operace	62
Maximální doba trvání	62
Eskalace	63
Aktivace	65
Podpis	65
Přílohy	65
6.6.2 VLASTNOSTI HRANY	66
Název hrany	66
Jméno	66
Směr toku hrany	67
Zobrazení hrany (Kreslit)	67
Skript	67

Popiska hrany (Textbox)	67
Kreslit zkráceně	67
6.6.3 VLASTNOSTI VNOŘENÉHO WORKFLOW	68
Název vnořeného workflow	68
Jméno	68
Popis	68
Nápověda	68
Ikona	68
Workflow	69
6.7 ADMINISTRACE SKRIPTŮ	69
6.7.1 VYTVOŘENÍ NOVÉHO SKRIPTU	69
Pojmenování skriptu	69
Typ skriptu	70
6.7.2 EDITACE SKRIPTU	70
Ukotvení okna editoru skriptu	71
Záložky okna editoru skriptu	71
Předdefinované vzory kódu	71
Značky ve skriptu	72
6.7.2.1.1.1 Vytvoření značky	72
6.7.2.1.1.2 Smazání značky	72
6.7.2.1.1.3 Hledání podle značek	72
Výběr proměnných	72
Vyhledávání v textu skriptu	73
Panel nástrojů editoru skriptu	73
6.7.2.1.1.4 Nápověda	73
6.7.2.1.1.5 Nastavení	73
6.7.2.1.1.6 Zpět	74
6.7.2.1.1.7 Schránka	74
6.7.2.1.1.8 Tisknout skript	74
6.7.2.1.1.9 Uložení textu skriptu	74
6.7.3 VYHODNOCENÍ SKRIPTU	74
Překlad skriptu	74
Test skriptu	75
6.7.4 UKLÁDÁNÍ SKRIPTU	75
Uložení skriptu	75
Uložení skriptu s chybami	75
6.7.5 MAZÁNÍ SKRIPTU	75
6.7.6 MIGRACE SKRIPTU	76
Export skriptu	76
Import skriptu	76
6.8 NÁSTROJE	76
6.8.1 NASTAVENÍ	77
Záložka Nastavení	77
Záložka Testování	77
6.8.2 ADMINISTRACE	78
Vzory dokumentů	78
Slovník dat pro definici proměnných	79
Termíny	80
6.8.2.1.1.1 Jméno a Popis	81
6.8.2.1.1.2 Doba	81
Návrhář masek	82
Externí operace	82

Seznam skriptů	82
Knihovna ikon	82
6.8.3 PŘEHLED IKON PANELU NÁSTROJŮ	83

7 NÁVRHÁŘ MASEK **84**

7.1 POUŽITÍ NÁVRHÁŘE MASEK	84
7.1.1 PŘIHLÁŠENÍ A ODHLÁŠENÍ	84
Přihlášení	85
Odhlášení	85
7.2 PROSTŘEDÍ NÁVRHÁŘE MASEK	85
7.2.1 EDITOR VLASTNOSTÍ	86
7.2.2 SEZNAM POLÍ	86
7.3 SPRÁVA MASEK	86
7.3.1 FUNKCE PRO SPRÁVU MASEK	86
Okno pro správu masek	86
Vytvoření nové složky	87
Smazání složky	87
Uzavření okna pro správu masek	88
7.3.2 FUNKCE PRO PRÁCI S MASKOU	88
Otevření masky	88
Export masky	88
Smazání masky	88
7.3.3 VERZE MASKY	89
Prohlížení verzí masky	89
Nastavení aktuální verze	89
Export verze	89
Import verze	90
Smazání verze	90
7.4 TVORBA MASEK	90
7.4.1 NEŽ ZAČNETE	90
Proměnné	90
Slovník dat	90
7.4.2 NOVÁ MASKA	90
Vytvoření masky	90
Editace masky	91
Testování masky	91
Uložení masky	92
Uzavření masky	93
7.4.3 EDITOR VLASTNOSTÍ MASKY	93
7.4.4 NASTAVENÍ VLASTNOSTÍ MASKY	93
Titul	93
Stavový řádek	93
Měnit jméno dok.	94
Standardní jméno dokumentu	94
Povolit klíče	94
Pozice okna	94
Šířka a výška	94
OnClose, OnOpen	94
Klávesa OK, klávesa Zrušit	95
Standardní tlačítka	95

Stavová lišta.....	95
7.5 OBJEKTY MASKY	95
7.5.1 POSTUP VKLÁDÁNÍ OBJEKTŮ.....	96
7.5.2 SPOLEČNÉ VLASTNOSTI OBJEKTŮ.....	96
Vlastnost Jméno	96
Vlastnost Text.....	97
Vlastnost Proměnná	97
Vlastnost Kontrola	97
Vlastnost Editace	100
Vlastnost Standardní hodnota.....	101
Vlastnosti Sloupec a řádek	101
Vlastnosti Šířka a výška	101
Vlastnost Barva.....	101
7.5.3 POPIS OBJEKTŮ	101
Nápis.....	101
Vstupní pole	101
Memo (Text).....	102
Datumové pole	102
Zaškrťovací pole.....	103
Rozbalovací seznam	103
Rámeček.....	104
Skupina přepínačů.....	105
Tlačítko	106
Grid	107
Webový prohlížeč.....	109

2 Rejstřík

A

administrace skriptů · 63
 administrace workflow · 38
 aktivace · 59
 aktualizace slovníku dat · 52
 aktualizace verze · 44
 aritmetické operátory · 17

B

binární aritmetické operace · 18
 blok · 30
 boolean operátory · 18

D

definice aspektů · 74
 definice databázových proměnných · 74

definice pracovního kroku · 11
 definice workflow · 7
 definice zpracovatele kroku · 13
 deklarace proměnných · 30
 deklarace skriptu · 29
 dialog administrace · 38

E

editace masky · 85
 editace skriptu · 64
 editor vlastností · 80
 editor vlastností masky · 87
 eskalace · 57
 eskalační skript · 35
 export masky · 82

- export proměnných · 74
- export skriptu · 70
- export verze masky · 83
- export workflow · 40
- export z editoru skriptů · 70
- export ze Seznamu skriptů · 70
- externí deklarace · 29
- externí operace · 56, 76
- F
- filtr proměnných · 74
- funkce else · 9
- funkce pro práci s maskou · 82
- funkce pro správu masek · 80
- G
- graf workflow · 44
- H
- hierarchie operací · 19
- hrana · 45
- I
- ikona pracovního kroku · 54
- ikona uzlu s vnořeným workflow · 62
- import skriptu · 70
- import verze masky · 84
- import workflow · 41
- indexové proměnné · 73
- instance workflow · 7
- instanciace workflow · 50
- J
- jednoduché příkazy · 21
- jednoduché výrazy · 19
- jméno hrany · 60
- jméno nového skriptu · 63
- jméno pracovního kroku · 53
- jméno uzlu s vnořeným workflow · 62
- K
- knihovna ikon · 76
- komentář · 31
- koncový uzel · 45
- konflikty objektů · 42
- konstanty · 16
- konstanty string · 16
- konstrukce grafu workflow · 9, 45
- kontrola pro typ Číslo · 94
- kontrola pro typ Datum · 93
- kontrola pro typ Grid · 92
- kontrola pro typ Řetězec · 92
- kontrola správnosti workflow · 47
- L
- lokální proměnná · 16
- M
- maximální doba trvání · 56
- mazání komponent · 47
- mazání skriptu · 69
- mazání verzí · 44
- mazání workflow · 40
- menu Administrace · 72
- menu Nastavení · 71
- menu Nástroje · 70
- migrace skriptu · 70
- migrace workflow · 40
- mřížka · 46
- mřížka masky · 85
- N
- nápověda editoru skriptu · 67
- nápověda k pracovnímu kroku · 53, 62
- nastavení aktuální verze masky · 83
- nastavení editoru skriptu · 67
- nastavení práv · 43
- nastavení proměnných · 52
- nastavení vlastností masky · 87
- nástroje editoru skriptu · 67
- navigátor · 46
- návrh grafu workflow · 9
- návrhář masek · 76
- název hrany · 60
- název vnořeného workflow · 62
- názvy · 16
- nekorektní skript · 69
- node ID · 53
- nová maska · 84
- O
- obecné funkce · 37
- obecné vlastnosti workflow · 50
- objekt document · 26
- objekt muser · 24
- objekt Nápis · 95
- objekt user · 25
- objekt Vstupní pole · 95
- objekt wkf · 26
- objekty · 24
- odhlášení · 37
- odhlášení z Návrháře masek · 79
- okno verzí masky · 83
- operátory · 17
- oprávnění · 42
- oprávnění role · 43
- oprávnění uživatele · 43
- orámování ikon uzlů · 50
- otevření masky · 82
- otevření workflow · 39
- P
- panel nástrojů · 77
- počáteční uzel · 45
- podmíněné příkazy · 22
- podmíněné větvení · 47
- podpis pracovního kroku · 59
- popis objektů · 95
- popis pracovního kroku · 53
- popis uzlu s vnořeným workflow · 62
- popis workflow · 50
- popiska hrany · 61
- popup menu vnořeného workflow · 63
- pořadí objektů pro editaci · 91
- postup vkládání objektů · 90
- použití Návrháře masek · 78
- povolení přístupu · 43
- pracovní krok · 7
- pracovní plocha · 38
- pravidla pro hrany · 9
- prázdné znaky · 15
- prázdný string · 16
- prohlížení verzí · 44
- prohlížení verzí masky · 83
- proměnná typu list · 17
- proměnné · 16, 84
- proměnné procesu s řídicími funkcemi · 74

- proměnné ve workflow · 11
- prostředí Návrháře masek · 79
- prováděcí maska · 55
- prováděcí příkaz · 22
- prováděcí skript · 33, 54
- prvky jazyka skriptů · 16
- překlad skriptu · 68
- přesouvání komponent · 46
- přidání přílohy k pracovnímu kroku · 60
- přidávání proměnných · 51
- přidělení časové třídy · 74
- přidělení hodnoty · 21
- přihlášení · 37
- přihlášení a odhlášení · 78
- přihlášení do Návrháře masek · 79
- příkaz for · 23
- příkaz foreach · 23
- příkaz návratu · 22
- příkaz while · 23
- příkazy · 20
- příkazy cyklu · 23
- příklady · 31
- přílohy pracovního kroku · 59
- R**
- reference proměnné · 17
- relační operátory · 18, 19
- rezervovaná slova · 15
- rozměry objektu masky · 95
- rozvrh grafu workflow · 9
- Ř**
- řetězce · 14
- řídící znaky · 15
- řízení průběhu procesu · 47
- S**
- seznam chyb workflow · 48
- seznam polí · 80
- seznam skriptů · 76
- seznamové operátory · 18
- skript · 61
- skript hrany · 33
- skript hrany workflow · 13
- slovník dat · 84
- slovník pro definici proměnných · 73
- smazání masky · 82
- smazání složky · 81
- smazání verze masky · 84
- směr hrany · 61
- souhrnný export workflow · 40
- souhrnný import workflow · 41
- společné vlastnosti objektů · 90
- spouštění procedur · 22
- správa masek · 80
- správa workflow · 38
- srovnání jednoduchého typu · 19
- srovnání stringů · 19
- standartní export workflow · 40
- standartní import workflow · 41
- string operátory · 18
- strukturované příkazy · 22
- symboly · 15
- synchronizační uzel · 47
- syntaxe výrazů · 19
- T**
- tělo skriptu · 30
- term · 19
- termíny · 74
- test skriptu · 69
- test workflow · 13
- testování masky · 85
- tisk grafu workflow · 49
- tisk skriptu · 68
- tvorba masek · 84
- tvorba workflow · 7
- typ skriptu · 64
- typy dat · 16
- U**
- ukládání nekonzistentního workflow · 49
- ukládání skriptu · 69
- ukládání workflow · 48
- ukotvení okna editoru skriptu · 65
- uložení masky · 86
- uložení skriptu · 69
- uložení workflow · 40
- umístění objektu masky · 95
- unární aritmetické operace · 18
- úpravy grafu workflow · 46
- uzavření masky · 87
- V**
- verze masky · 83
- verze workflow · 44
- vkládání stejných komponent · 47
- vlastnost Jméno · 90
- vlastnost Provedení · 55
- vlastnost Uživatel · 54
- vlastnosti komponent workflow · 52
- vlastnosti pracovního kroku · 53
- vlastnosti vnořeného workflow · 62
- vlastnosti workflow · 49
- vložení vnořeného workflow · 63
- vnořené workflow · 45
- volání externích funkcí · 20
- volání funkcí · 20
- volání masek · 20
- volání skriptů · 20
- vstupní skript · 32, 54
- výběr proměnných ve skriptu · 66
- vyhledávání ve skriptu · 67
- vyhodnocení skriptu · 68
- výrazy · 19
- výsledek skriptu · 31
- vytvoření masky · 84
- vytvoření nové složky · 81
- vytvoření skriptu · 63
- vytvoření workflow · 39
- vzory dokumentů · 72
- vzory kódu · 65
- W**
- workflow proměnná · 17
- Z**
- záložka Nastavení · 71
- záložka Proměnné · 51
- záložka Testování · 71
- záložky okna editoru skriptu · 65
- zarovnání objektů · 85
- zavření workflow · 39
- změna workflow · 14

značky ve skriptu · 66
zobrazení dialogu · 22
zobrazení grafu · 39
zobrazení hrany · 61
zpětná hrana · 47
zrušení kroku editace · 47

objekt Datumové pole · 96
objekt Grid · 101
objekt Memo · 96
objekt Rámeček · 98
objekt Rozbalovací seznam · 97
objekt Skupina přepínačů · 99

objekt Tlačítko · 100
objekt Zaškrťovací pole · 97
objekty masky · 89
okno pro správu masek · 80
uzavření okna pro správu masek · 82
vlastnost Barva · 95
vlastnost editace · 94
vlastnost kontrola · 91
vlastnost proměnná · 91
vlastnost standardní hodnota · 95
vlastnost Text · 91
vlastnosti hrany · 60

3 Tvorba workflow

3.1 Co je to workflow

Workflow realizuje automatizaci části nebo celého podnikového procesu, během kterého jsou dokumenty, informace nebo úlohy předávány podle sady určitých procedurálních pravidel ke zpracování od jednoho účastníka k druhému (definice podle mezinárodní organizace Workflow Management Coalition Terminology & Glossary).

Workflow v systému M/TeamBridge je vhodné implementovat tehdy, pokud...

- firemní proces zpracovává více pracovníků
- je třeba hlídat termíny
- je jasně dán pracovní postup, který je nutné dodržovat
- je nutné zavést „pořádek“ a evidenci činností

Příklad:

Cílem je připravit workflow pro řešení reklamace výrobku nebo služby. Proces bude probíhat ve třech pracovních krocích:

- Příjem reklamace
- Rozhodnutí o reklamaci
- Řešení reklamace

Ve druhém kroku se bude workflow větvit podle oprávněnosti reklamace. Třetí krok bude probíhat jen v případě oprávněné reklamace.

3.1.1 Definice základních pojmů

- **definice workflow:**

Návrh workflow, který je zpracován pomocí programu WorkflowDesigner. Např. workflow „Faktura přijatá“

- **instance workflow:**

Jde o konkrétní běh jednoho workflow dle definice workflow.

např. „Faktura přijatá č. FP-0047/2008“

- **pracovní krok:**

Asi nejdůležitějším pojmem a také nejsložitějším je „Pracovní krok“ - část workflow, která je zpracovávána v jeden časový okamžik jedním zpracovatelem kroku. V grafické podobě workflow jde o jeden čtvereček v grafu. Každý krok je vždy přidělen konkrétnímu zpracovateli, tzn. není možné jeden krok přidělit skupině uživatelů (např. účetním) a čekat, kdo krok zpracuje. Pokud je jedno, kdo z nějaké skupiny uživatelů krok provede, pak je nutné využívat tzv. virtuální uživatele.

U každého pracovního kroku je nutné si promyslet a nadefinovat:

1. kdo bude daný krok provádět
2. jaká činnost bude v kroku prováděna
3. jak se bude krok jmenovat
4. jaké masky se budou v kroku zpracovateli zobrazovat
5. jaký je termín na dokončení kroku a jaká akce se má provést, pokud je tento termín překročen

- **hrana:**

Spojnice mezi jednotlivými kroky. U hrany lze definovat podmínky, za jakých bude hrana otevřená, případně zavřená, tzn. kterou z cest se bude workflow ubírat.

- **maska:**

Formulář, ve kterém jsou uživatelům zobrazeny informace. V masce je možné nadefinovat, které informace se budou uživatelům zobrazovat pouze pro čtení a které bude moci editovat. Dále zde mohou být prázdná pole, kam uživatel zadá nové informace. Se zadanými daty ve formě proměnných pak pracuje skript, ze kterého byla maska vyvolána. Na tato pole mohou být v masce aplikovány kontroly zadávaných dat, které se řídí podle typu proměnné použité pro vstup dat.

- **skript:**

Skripty obsahují zápis činností ve workflow v jazyce skriptů.

- Skripty pracovního kroku nebo hrany nabývají návratové hodnoty, podle níž se řídí další průběh workflow (typ Boolean).
- Vstupní skript pracovního kroku pro vyhodnocení zpracovatele kroku. Návratovou hodnotou je uživatel nebo skupina uživatelů, ze které bude zpracovatel kroku vybrán (typ Usercollection).
- Skripty pro zápis činnosti, která má být provedena v případě překročení zadaného termínu pro pracovní krok (eskalační skripty). Tyto skripty nemají návratovou hodnotu (typ Void)

- **globální proměnná:**

Každá definice workflow obsahuje seznam proměnných. Proměnné jsou pak zobrazovány zpracovatelům v maskách. Proměnné jsou uloženy ve Slovníku dat – databázi pro správu proměnných M/TeamBridge. Každá instance pak tyto proměnné využívá a nesdílí je s jinou instancí, tzn. že jestliže např. proměnná `cislo_faktury` je použita pro dva procesy s workflow pro dvě různé faktury, proměnná prvního procesu obsahuje příslušné číslo faktury a není druhým procesem nijak ovlivněna a naopak.

- **lokální proměnná:**

Při zpracování kroku, kdy jsou použity informace z masek uložené v globálních proměnných, je možné nadefinovat proměnné pro pomocné výpočty v rámci kroku. Tyto proměnné se neukládají do Slovníku dat a nemohou být použity mimo krok, kde jsou definovány.

- **vlastník procesu:**

Pracovník firmy, který o daném pracovním procesu ví nejvíce a je schopný spolupracovat na tvorbě workflow procesu.

- **zpracovatel kroku:**

Účastník definovaný v systému správy uživatelů M/User, který je při běhu workflow vyhodnocen podle zadaných podmínek ke zpracování kroku.

- **účastník procesu:**

Uživatel procesu přidáný do něj libovolným způsobem (vyhodnocením ve workflow pro zpracování některého kroku, přímým přidáním jiným účastníkem, jako zástupce jiného účastníka procesu).

- **virtuální uživatel:**

V případě, že u kroku nezáleží na tom, kdo jej bude jmenovitě provádět, ale je to dáno např. profesí, lze nadefinovat uživatele se jménem např. „Účetní“ a ostatním uživatelům spadajícím do této skupiny nastavit zastupování tohoto virtuálního uživatele. Pak může krok virtuálního uživatele zpracovat kdokoliv z této skupiny (kdo se toho „ujme“) aniž by musel uživatel předchozího kroku vybírat konkrétního účastníka.

3.1.2 Návrh grafu workflow

Po ujasnění základních pojmů můžete přistoupit k vlastnímu návrhu workflow

Základem správného fungování workflow je návrh grafu workflow, který splňuje všechny požadavky kladené na pracovní proces. V ideálním případě by měl počítat i s možnými změnami pracovních postupů a personálního obsazení v budoucnu.

Rozvrh grafu

Postup práce:

- načrtnout si na papír potřebné uzly (pracovní kroky) se zřetelem na personální obsazení uzlů a termíny splnění činnosti pracovních kroků
- spojit uzly hranami včetně podmínek jejich průchodnosti

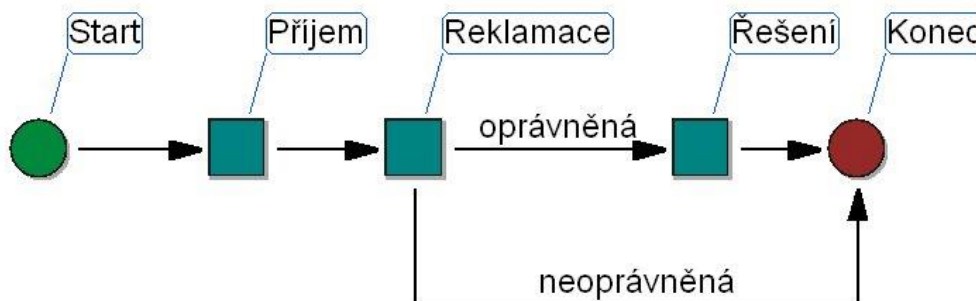
Tento rozvrh se obvykle provádí v součinnosti s vlastníkem workflow.

Konstrukce grafu workflow

Po promyšlení výše uvedených skutečností lze přistoupit ke konstrukci grafu workflow v prostředí WorkflowDesigneru.

Příklad:

Podle výše stanovených podmínek pro workflow Reklamací je vytvořeno toto schéma...



Pravidla pro hrany

Rozhodující pro průběh workflow je výsledek vyhodnocení skriptu hrany.

- Pokud je výsledkem hodnota *true*, může workflow pokračovat dál.
- Pokud je výsledkem hodnota *false*, není možné hranou pokračovat.
- Mohlo by dojít k situaci, že by žádná cesta nebyla průchodná a workflow by bylo přerušeno. Z toho důvodu je někdy nutné použít hranu *else*.

Funkce else

V rámci zajištění konzistence workflow je nutné zajistit, aby nebylo nikde přerušeno. To znamená, že musí být neustále k dispozici hrana, kterou by mohlo workflow pokračovat. K tomuto účelu slouží klíčové slovo *else*. Je nutné dodržet následující pravidla:

Pokud vede od nějakého uzlu pouze jedna hrana, není možné jí přiřadit žádný skript.

Toto pravidlo je založeno na předpokladu, že skript s návratovou hodnotou *boolean* může mít výsledek *false* a nebylo by tedy možné pokračovat dál. Pokud není hraně přiřazen žádný skript, nemůže nastat situace přerušení workflow.

Pokud vede od jednoho uzlu workflow více hran, které všechny obsahují skript, musí mezi nimi být právě jedna hrana *else*.

Hrana *else* je hrana, kterou workflow pokračuje, pokud neexistuje žádná jiná hrana s hodnotou *true*. Pouze v tomto případě je hrana *else* aktivována.

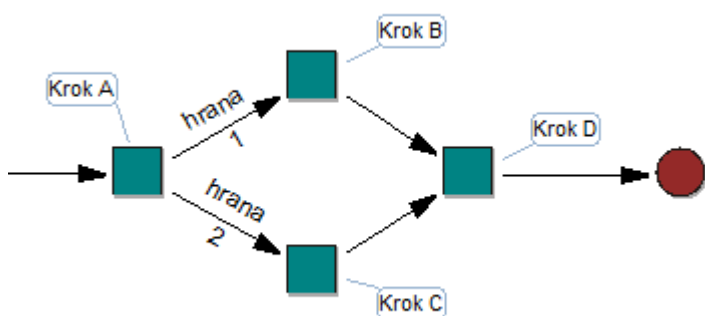
Mechanismus zpracování workflow

Důležitým bodem při tvorbě prvního workflow je pochopení, jak systém vyhodnocuje podmínky na hranách a jaké jsou návaznosti mezi kroky v grafu.

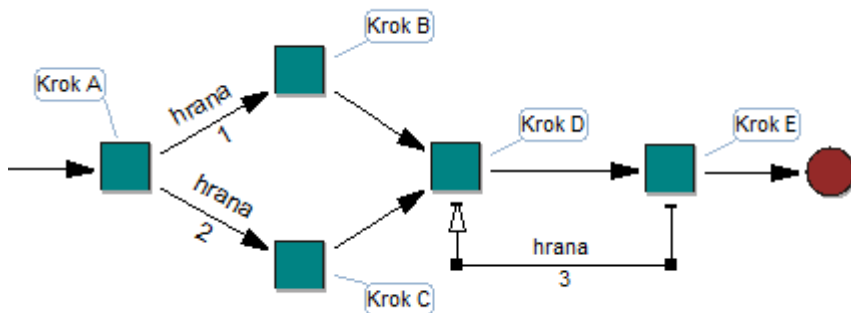
Nejjednodušší situace je taková, kdy workflow má pouze sériové kroky, tzn. kroky následující jeden za druhým a nejsou zde žádná větvení. Pak systém po dokončení jednoho kroku automaticky připraví krok k vyřízení. (Stav vyřízení je takový, kdy krok může být přiděleným uživatelem spuštěn. Někdy také říkáme, že krok je instanciován.)

Drobná komplikace nastane, pokud se workflow větví, tak jak je uvedeno na následujícím obrázku. Pokud na žádné hraně není skript, pak krok D bude ve stavu k vyřízení až tehdy, když bude dokončen krok B a C. Pokud je ale např. na hraně 1 podmínka, která hranu uzavřela, tak se krok D dostane do stavu k vyřízení ihned po dokončení kroku C.

Dalo by se tedy stanovit pravidlo: Krok je instanciován (ve stavu k vyřízení) tehdy, pokud do něj přichází všechny hrany, které do něj vedou.



Ukažme si však nyní příklad, kdy by toto nefungovalo:



Po uplatnění předchozího pravidla by totiž krok D čekal také na krok E, který by ale určitě ještě nebyl dokončen. Proto můžeme některé hrany označit jako **Zpětné** (jde o vlastnost na hraně, graficky je to znázorněno neplnou šipkou). Předchozí pravidlo pak rozšíříme na: Krok je instanciován tehdy, pokud do něj přichází všechny hrany, které do něj vedou, avšak krok s instanciováním nečeká na hrany zpětné.

3.2 Definice proměnných

Velmi důležité je ještě před začátkem návrhu činnosti pracovních kroků, což reprezentuje ve většině případů i návrh masek k přijímání informací od uživatele, nadefinovat globální proměnné, které budou k tomu potřeba. To se provádí ve Slovníku dat (popis v kapitole Slovník dat pro definici proměnných na straně 79), do kterého je přístup z prostředí Návrháře masek i Workflow Designeru (pokud máte přidělena příslušná práva). Ve Workflow Designeru najdete datový slovník v menu NÁSTROJE/ADMINISTRACE/DATOVÝ SLOVNÍK.

V dialogu EDITOVAT/VLASTNOSTI WORKFLOW ve Workflow Designeru je pak nutné tyto proměnné přiřadit k právě editovanému workflow.

K proměnným procesům můžete přistupovat ze skriptu, z příloh (makro v MSWORD) nebo přes OLE rozhraní aplikace. Tyto proměnné (tzv. globální, ve skriptu s předponou <w $_{$ wf.>) si zachovávají hodnotu z jednoho kroku i po přechodu do dalšího kroku téže instance workflow. Naopak v různých instancích téže workflow se tyto proměnné stejného jména nijak neovlivňují.

Důvod, proč se proměnná definuje a zvláště přidává k danému workflow je ten, že je možné jednu definici proměnné používat v různých definicích workflow. Existuje pak např. pouze jedna proměnná ID_dodavatele a ta je použita ve WKF Smlouva i WKF Faktura přijatá.

Možné typy proměnných a jejich standardní hodnoty:

Tabulka typů a hodnot pro definici proměnných		
Typ	Popis	Standardní hodnota
Řetězec	string, řetězec znaků	
Číslo	integer, celé číslo	0
Boolean	log. hodnota Ano/Ne	Ne
Datum	date	
Měna	currency	
Seznam	list všechny typy	

Kromě globálních proměnných existují i lokální proměnné, které lze nadefinovat přímo ve skriptu pro pomocné výpočty. Mají platnost pouze v rámci pracovního kroku, kde je tento skript použit. Mohou být stejného typu jako globální proměnné kromě typu Seznam (list).

Příklad:

Pro workflow Reklamace budou zapotřebí minimálně tyto proměnné:

- pro název reklamovaného výrobku nebo služby (např. nazev, typ string)
- pro rozhodnutí o oprávněnosti reklamace (např. opravnena, typ boolean)
- pro potvrzení vyřešení reklamace (např. vyresena, typ boolean)

V praxi bude samozřejmě proměnných daleko více, např. v případě sledování termínů pro zpracování kroků to budou proměnné typu date, pro cenu výrobku nebo služby a cenu předpokládané opravy typ currency atd.

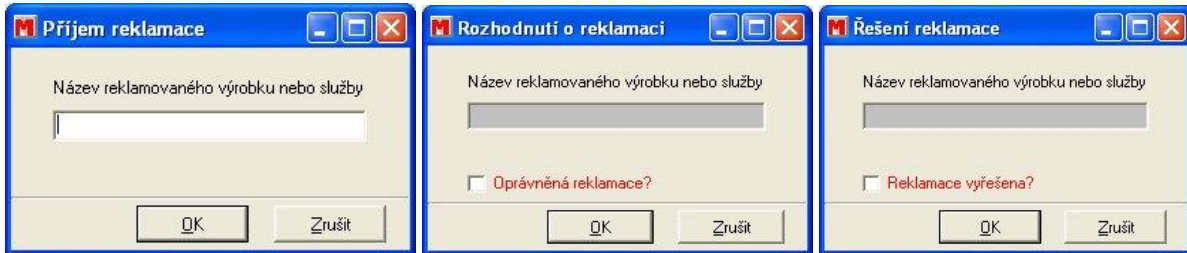
Navržené proměnné nyní nadefinujte ve Slovníku dat. Nezapomeňte je pak také přidat do workflow ve Vlastnostech workflow.

3.2.1 Definování jednotlivých pracovních kroků

Činnost pracovních kroků je určena použitím skriptů a masek. Ve většině případů je pro činnost skriptu třeba získat nejprve od uživatele nějaké informace ve formě hodnot proměnných, což se děje prostřednictvím masky. Lze tedy vycházet z masky, kterou vytvoříte pomocí Návrháře masek (popis na straně 84), a ve které použijete globální proměnné nadefinované předem. O syntaxi jazyka skriptů se více dozvíte v kapitole Jazyk skriptů na straně 20.

Příklad:

Pro tři pracovní kroky workflow Reklamace to mohou být tyto masky (pro zjednodušení byla použita standardní tlačítka, výhodnější je vytvořit tlačítka vlastní, která lépe vystihují další postup. Výhodné bývá tlačítko Zavřít, které umožní uložit zadané hodnoty proměnných, na rozdíl od standardního tlačítka Zrušit, které hodnoty neukládá):



Následuje vytvoření skriptu s voláním této masky (případně postupně více masek) pomocí funkce **idx** a s dalším zpracováním a vyhodnocením hodnot proměnných získaných z masky. Další běh workflow pak je řízen těmito hodnotami.

Důležité je ve skriptu vyhodnotit logickou návratovou hodnotu skriptu (result), která určuje, zda workflow může pokračovat na další krok (true).

Příklad:

Nejjednodušší prováděcí skript pro první krok workflow Reklamace může vypadat takto:

```
script reklamace_prijem (): Boolean
begin
    result:=idx("reklamace_prijem")
end
```

Pro druhý krok může skript vypadat obdobně, protože vyhodnocení proměnné “opravnena” se provádí ve skriptu hrany (viz dále).

Ve třetím pracovním kroku již není proměnná “vyresena” použita pro větvení workflow, ale např. pro zápis textu do procesu:

```
script reklamace_reseni (): Boolean
begin
    result:=idx("reklamace_reseni")
    if wkf.vyresena then
        document.newnote ("Reklamace vyřešena.")
    else
        document.newnote ("Reklamace nevyřešena.")
    end
end
```

Pro činnosti, které nelze realizovat přímo prostředky Jazyka skriptů jsou k dispozici přídatné knihovny dll. Často používaná je knihovna ActualDocument.dll, kde najdete celou řadu funkcí v kategoriích:

- Práce s textem
- Datum a čas
- Práce se soubory
- Komunikace s MS Office
- Práce s uživateli
- Práce se strukturou složek
- Práce s celými procesy

a mnoho dalších. Tato knihovna se stále vyvíjí a doplňuje o nové funkce podle potřeby uživatelů workflow.

Hotový skript pak připojíte k pracovnímu kroku pomocí Editoru vlastností jako Prováděcí skript.

Méně využívanou možností je připojit pouze masku do vlastnosti Masky. Pak už nelze připojit prováděcí skript.

3.2.2 Určení, kdo bude zpracovávat jaký krok

Přidělení zpracovatele kroku probíhá dvěma způsoby v Editoru vlastností pracovního kroku (viz kap. Vlastnosti pracovního kroku na straně 60):

- je vybrán přímo konkrétní uživatel ze seznamu účastníků nadefinovaných ve správě uživatelů M/User do vlastnosti Uživatel pracovního kroku. Tento způsob se hodí pro virtuální uživatele. Pro konkrétního uživatele dle jména není tento způsob příliš vhodný z důvodu možných personálních změn ve firmě.
- do vlastnosti Vstupní skript pracovního kroku je připojen předem připravený skript, který je typu Usercollection a určuje zpracovatele např. podle role.

Příklad:

V prvním kroku workflow Reklamacce můžete jako vstupní použít obecný skript Act_User, pokud workflow může spustit každý, kdo má k tomu oprávnění.

```
script Act_User () : UserCollection
begin
  result := actuser
end
```

Pokud se ve vstupním skriptu odkážete na uživatele `actuser`, pak tímto uživatelem je člověk, který zpracovával krok předchozí. To je způsobeno tím, že vstupní krok se provádí při dokončování kroku předchozího a to z důvodu, že když je krok dokončen, tak systém musí zjistit, komu má workflow předat dál, komu tedy vlastně instanciovat další krok.

Pro další kroky pak můžete vytvořit virtuálního uživatele nebo skripty podle role (např. "Vedoucí"):

```
script reklamace_rozhoduje () : UserCollection
begin
  result := getuserlistbytemplate ("Vedouci")
end
```

3.2.3 Skripty pro hrany

Pokud dochází k větvení grafu workflow, pak podle pravidel pro hrany některé hrany obsahují skripty typu Boolean. Podle vyhodnocení skriptu pak je hrana průchodná nebo neprůchodná. Pokud hraně není přiřazen žádný skript, pak je hrana vždy průchodná. Pokud je hraně přiřazen nějaký skript, pak je hrana neprůchodná a ve skriptu je nutné hranu způchodnit. To se provede tak, že pokud je splněna nějaká podmínka, tak se nastaví `result := True`


Příklad: Ve workflow Reklamacce dochází k větvení ve druhém pracovním kroku na základě vyhodnocení proměnné "opravnena". Proto hrana pro oprávněnou reklamaci bude mít skript...

```
script reklamace_hrana () : Boolean
begin
  result := wkf.opravnena
end
```


Hrana pro neoprávněnou reklamaci pak musí mít podle pravidel skript else.


3.2.4 Testování workflow

Testování masky

Masku můžete otestovat přímo při její tvorbě v Návrhářci masek po uložení změn pomocí ikony . Můžete ověřit vzhled masky, kontroly hodnot polí a výslednou logickou hodnotu masky.

Testování skriptu

Skript můžete v Editoru skriptů zkontrolovat po formální stránce tlačítkem . Zde se projeví chyby typu nesprávný tvar jazyka skriptů, chybějící definice proměnných atd.

Fungování skriptu lze v omezené míře (nenaplněné proměnné z předchozích kroků workflow) vyzkoušet tlačítkem . Zde je důležitá výsledná logická hodnota skriptu. Pozor na to, že funkce z knihovny

ActualDocument.dll, které mají jako jeden z parametrů klíčové slovo „actualDocument“ při testování nebudou fungovat.

Tip: Pozor na uložení změn před testováním, dojde-li např. k zacyklení nebo přerušení programu, změny nebudou uloženy.

Testování konzistence workflow

U vytvořeného workflow můžete provést takzvaný test správnosti workflow (test konzistence), který přezkouší vnitřní logiku workflow. Případné chyby jsou vypsané. Workflow, které neprošlo testem konzistence lze uložit jako nekonzistentní workflow nebo jako konzistentní, pokud jste si jisti, že nekonzistence nebude na závadu fungování workflow.

3.2.5 Změny workflow

Každá instance workflow má uloženo schéma workflow včetně skriptů a masek v okamžiku jejího spuštění. Proto není možné měnit zásadně skripty použité v uzlech workflow, ale vytvořit skript s jiným jménem a ten začlenit do příslušného uzlu (jedná se hlavně o přidání nových globálních proměnných). Od této chvíle budou nové instance workflow používat nový skript. Nedokončené instance workflow spuštěné před změnou budou pokračovat v běhu s původními skripty.

4 Jazyk skriptů

4.1 Co je skript

Při práci s M/TeamBridge můžete mimo jiné zpracovávat procesy založené na workflow. Workflow se vytváří a upravuje pomocí programu WorkflowDesigner. Workflow je nadefinovaný proces, ve kterém se vykonávají předem definované aktivity v pořadí určeném jednotlivými pracovními kroky.

Pro chod workflow se využívají skripty, ve kterých se například rozhoduje, který uživatel bude krok vyřizovat a jaká konkrétní aktivita se v tomto kroku bude provádět. Skripty se tvoří pomocí jazyka skriptů.

Skripty si můžete kdykoliv prohlédnout v programu WorkflowDesigner, není však možné je zde zpracovávat, pokud nemáte pro tuto činnost přidělena specifická práva.

V této příručce se Vám pokusíme jazyk skriptů přiblížit.

4.2 Terminologie a konvence

Řetězce jsou jednotky textu skriptu. Řetězec může být symbol, název, číslo nebo konstanta *string*.

Skript se skládá z řetězců a oddělovačů. Oddělovačem může být prázdný znak nebo komentář. Dva po sobě následující řetězce, které obsahují rezervovaná slova, názvy nebo čísla, musí být vzájemně odděleny více oddělovači. U konstanty *string* nesmí být oddělovač součástí řetězce.

Jazyk skriptů je “case-sensitive”, tj. rozlišuje velká a malá písmena.

Symbole a rezervovaná slova jsou řetězce, které mají ustálený význam (případně více významů).

4.3 Symboly

Symboly mohou být písmena, čísla, prázdné znaky...

- Písmena A až Z, a až z
- Čísla 0 až 9
- Prázdné znaky...
- Mezera (ASCII 32) a všechny řídicí znaky (ASCII 0 až 32) včetně odřádkování a mazání (ASCII 13) patří mezi prázdné znaky.

Následující znaky řadíme mezi symboly...

+ - * / = <> : . > | & , _

Následující skupiny znaků také považujeme za symboly...

<= >= () // := += -= // /* */ +=

4.3.1 Rezervovaná slova

Rezervovaná slova nemohou být znovu nadefinována. Je možné je psát malými i velkými písmeny.

V následující tabulce najdete seznam rezervovaných slov užívaných ve skriptech...

actuser	false	of
adduser	for	or
and	foreach	pointer
as	function	procedure
begin	getdate	reg
boolean	getday	result
break	getmonth	return
case	getusersbyattribute	script
cdecl	getuserbyid	setdate
char	getuserlistbydepartment	setdatemask
continue	getuserlistbytemplate	setday
count	getyear	setmonth
currency	idx	setyear
date	if	string
declare	in	then
dll	ini	to
do	integer	true
document	newnote	user
else	nil	usercollection
end	not	var
env	not_in	void
errornumber	notecount	while

4.3.2 Názvy

Každý skript má název. Názvy se používají i pro proměnné, procedury a funkce. Název může být libovolně dlouhý a začíná vždy písmenem nebo podtržítkem (`_`). Název nesmí obsahovat žádné prázdné znaky, může obsahovat písmena, čísla a podtržítka.

4.3.3 Konstanty string

String je skupina žádného nebo více znaků rozšířené ASCII tabulky znaků (libovolný znak kromě uvozovek), která se nachází v jednom řádku. *String* se píše do uvozovek (`""`).

String, který nemá v uvozovkách žádný znak, se nazývá prázdný string.

Příklad:

```
"Toto je string"
```

4.4 Prvky jazyka skriptů

4.4.1 Typy dat

V jazyce skriptů se užívají jednoduché a strukturované typy.

boolean	má hodnotu <i>true/false</i>	Jednoduché typy
currency	desetinná čísla	
date	datum, udání času	
integer	má formát 32 bitů včetně znaménka	
string	skládá se z libovolného počtu znaků	
void	prázdná proměnná	
user	objekt, který v aplikaci M/User reprezentuje uživatele	Strukturované typy
usercollection	objekt, který reprezentuje seznam objektů typu user	

4.4.2 Konstanty

Jazyk skriptů obsahuje konstantní výrazy...

Příklad:

Nejjednodušším příkladem konstantního výrazu je třeba konstanta 100.

Další příklady:

```
256 - 1, "jazyk" + "skriptů"
```

4.4.3 Proměnné

Proměnná je název pro hodnotu, kterou lze měnit.

Lokální proměnné

Deklarace proměnných je seznam názvů, ve kterém jsou dány nové proměnné a jejich typ. Deklarace začíná zadáním klíčového slova **var**.

```
var seznam_promennych : typ
```

Proměnné v seznamu jsou odděleny čárkami. Platí pouze v rámci daného skriptu.

Příklad deklarace proměnných:

```
var Today: Date
I, J, K: Integer
```

Workflow proměnné

Workflow proměnné jsou předdefinovány ve Slovníku dat a mohou být přiřazeny k workflow v dialogu Vlastnosti workflow. Mají charakter globálních proměnných, tj. jejich hodnoty lze předávat mezi kroky instance workflow.

Ve skriptu můžete s těmito proměnnými pracovat pomocí předpony <w_{kf}>.

S těmito proměnnými pracují také masky.

Příklad:

Ve Slovníku dat je definována proměnná `cislo_faktury`. Ve skriptu bude zapsána jako:

```
wkf.cislo_faktury
```

Proměnné typu list

Pokud je proměnná nadefinována ve Slovníku dat jako typ list, může obsahovat více hodnot (seznam, list). Ke zjištění počtu hodnot uložených v proměnné lze využít metodu `count` (viz Rozšiřující metoda Count na straně 35):

```
pocet := wkf.VariableName.count
```

S jednotlivými hodnotami proměnné se pracuje pomocí indexu v hranatých závorkách za jménem proměnné. Index začíná hodnotou 0 pro první uloženou hodnotu proměnné.

```
prvni_hodnota := wkf.VariableName[0]
```

Index lze nahradit proměnnou typu integer a pracovat v cyklu postupně se všemi uloženými hodnotami, resp. přidávat nové.

Příklad:

```
for i:=0 to k do
  wkf.VariableName[i] := hodnota
end
```

Upozornění:

Nové hodnoty do proměnné typu list je třeba přidávat vzestupně bez vynechání za poslední uloženou hodnotu.

Reference proměnných

Jednoduchou referenci představují lokální proměnné a proměnné workflow.

Strukturovanou referenci reprezentují proměnné s předponou <actuser.> pro aktuálního uživatele a <user.> pro libovolného uživatele. Tyto reference jsou read only, tj. nelze je ve skriptu měnit.

Příklad:

```
s := actuser.userID
```

4.4.4 Operátory

V tomto odstavci jsou popsány operátory používané v jazyce skriptů.

Aritmetické operátory

Následující tabulky obsahují typy operandů a výsledky binárních a unárních aritmetických operací...

Binární aritmetické operace			
Operátor	Operace	Typ operace	Výsledek operace
+	sčítání	integer/currency	integer/currency
-	odčítání	integer/currency	integer/currency
*	násobení	integer/currency	integer/currency
/	dělení	integer/currency	integer/currency
Unární aritmetické operace			
Operátor	Operace	Typ operandu	Typ výsledku
+	pozitivní znaménko	integer/currency	integer/currency
-	změna znaménka	integer/currency	integer/currency

Boolean operátory

Následující tabulka obsahuje operandy a typy výsledků operací boolean...

Operátor	Operace	Typ operandu	Typ výsledku
not	obrácení	boolean	boolean
and	logické A	boolean	boolean
or	Logické NEBO	boolean	boolean

Výsledky operací boolean vyplývají z běžné logiky. Například *A and B* je *true* pouze tehdy, pokud *A* i *B* jsou *true*.

Seznamové operátory

Operátor	Operace	Typ operandu	Typ výsledku
+	propojení	user	usercollection
	propojení	usercollection	usercollection

String operátory

Následující tabulka obsahuje operandy a typy výsledků operací string...

Operátor	Operace	Typ operace	Typ výsledku
+	propojení	string	string

Relační operátory

Následující tabulka obsahuje operandy a typy výsledků relačních operací...

Operátor	Operace	Typ operandu	Typ výsledku
=	rovnost	kompatibilní jednoduché typy, typy string	boolean
<>	nerovnost	kompatibilní jednoduché typy, typy string	boolean
<	menší než	kompatibilní jednoduché typy, typy string	boolean
>	větší než	kompatibilní jednoduché typy, typy string	boolean
<=	menší nebo rovno	kompatibilní jednoduché typy, typy string	boolean
>=	větší nebo rovno	kompatibilní jednoduché typy, typy string	boolean

Operátor	Operace	Typ operandu	Typ výsledku
<=	částečné množství	kompatibilní jednoduché typy, typy string	boolean
>=	maximální množství	kompatibilní jednoduché typy, typy string	boolean
in	příslušnost	User Attribut ze seznamu typů	boolean
not_in	nepříslušnost	User Attribut ze seznamu typů	boolean

Srovnání

- srovnání jednoduchého typu**
 Pokud jsou operandy operátorů =, <>, <, >, >= nebo <= jednoduchého typu, musí být tyto typy vzájemně kompatibilní.
- srovnání stringů**
 Relačními operátory =, <>, <, >, >= a <= mohou být stringy srovnávány na základě rozšířené ASCII tabulky znaků.

4.4.5 Výrazy

Hierarchie operací

Výrazy se skládají z operandu a operátoru. Většina operátorů je binárních, to znamená, že se skládají ze dvou operandů. Zbylé mají pouze jeden operand.

Operátory	Hierarchie	Kategorie
not	1. stupeň	unární operátory
*, /, and, &	2. stupeň	multiplikační operátory
+, -, or,	3. stupeň	sčítací operátory
=, <>, <, >, <=, >=, in, not_in	4. stupeň	relační operátory

Pro určení hierarchie operací existují tři základní pravidla...

- Operand mezi dvěma operátory různé hierarchie patří k operátoru vyšší hierarchie.
- Operand mezi dvěma rovnocennými operátory patří k operátoru více vlevo.
- Výrazy v závorkách se vyhodnocují prioritně.

Syntaxe výrazů

Pravidla hierarchie výrazů vyplývají ze syntaxe výrazů. Výrazy se skládají z faktorů, termů a jednoduchých výrazů.

V termech se využívají multiplikační operátory na faktory...

* / and

V jednoduchých výrazech se pro termy užívají sčítací operátory...

+ - or

Ve výrazu se pro jednoduché výrazy používají relační operátory...

< > <= >= = <> in not_in

Příklad:

```
i > 0
i > 0 and (i > j)
```

4.4.6 Volání funkcí

Volání externích funkcí

Spuštění aktivuje funkci, která je zadána pomocí názvu funkce. Pokud odpovídající deklarace funkce obsahuje seznam formálních parametrů, musí být při spuštění funkce tento seznam k dispozici seznamu aktuálních parametrů. Přitom zaujme každý aktuální parametr místo odpovídajícího formálního parametru.

Příklad:

```
GetState (state)
```

Poznámka:

Funkce musí být před spuštěním deklarovány (viz odstavec „Externí deklarace“)

Interní volání skriptů

Interní volání ve skriptu se spouští uvedením názvu skriptu následovaného závorkami ()

Příklad:

```
M_UserAdmin ()
```

Volání masek

Syntaxe:

```
idx (maskName: string) : Boolean
```

Příklad:

```
result := idx ("TestMask")
```

Jedná se o předdefinovanou funkci. Zavolá masku zadaného jména z aplikace Mask Designer. Návrátová hodnota je *true*, pokud účastník zavře masku tlačítkem OK, jinak má hodnotu *false* (při použití standardních tlačítek masky).

4.4.7 Příkazy

Rezervovaná slova pro příkazy:

Begin	Zahajuje blok příkazů.
break	Ukončuje příkaz opakování.
continue	Přejde na začátek příkazu cyklu.
declare	Deklaruje externí funkci před jejím použitím ve skriptu.
else	Následující příkaz je použit, když podmínka příkazu if není splněna.
end	Ukončuje blok příkazů.
exec	Volá program.

for	Opakované provedení příkazu dokud kontrolní proměnná nenabude stanovené hodnoty.
foreach	Opakuje příkaz pro každý prvek v objektu seznamu.
if	Rozhoduje, zda po vyhodnocení podmínky bude proveden příkaz.
return	Příkaz návratu
then	Následující příkaz bude proveden při splnění podmínky v příkazu if.
while	Řídí opakování skupiny příkazů.
write	Vypíše zadanou hodnotu do dialogu.

Příkaz (statement) je algoritmus, který je proveditelný. Existují jednoduché a strukturované příkazy.

4.4.8 Jednoduché příkazy

Jednoduchý příkaz je příkaz, který neobsahuje žádné další příkazy. Má tyto formy:

- přidělení hodnoty
- prováděcí příkaz
- příkaz návratu
- zobrazení dialogu
- spuštění procedury

Přidělení hodnoty

Přidělení nahrazuje současnou hodnotu proměnné novou hodnotou, která je výsledkem výrazu. Pro všechny typy operandů se používá operátor :=

V následující tabulce najdete další operátory užívané pro přidělení...

Operátor	Operace	Typ operandu	Typ výsledku
+=	přidělení + sčítání	integer	integer
		currency	currency
-=	přidělení + odčítání	integer	integer
		currency	currency
+=	propojení	string	string
Operátor	Operace	Typ operandu	Typ výsledku
+=	přidání uživatele do seznamu	user	usercollection
		usercollection	usercollection
-=	vymazání uživatele ze seznamu	user	usercollection
++=	přidání uživatele do seznamu na první pozici	user	usercollection
		usercollection	usercollection

Příklad:

```
i:=0
Seznam++=actuser
```

Prováděcí příkaz

Prováděcí příkaz spustí program se specifickými parametry uvedený v konstantách string.

Příklad:

```
Exec ("c:\Program Files\Microsoft Office\WINWORD.EXE")
```

Příkaz návratu

Příkaz návratu **return** ukončí skript a přeskočí zpět.

Zobrazení dialogu

Tento příkaz zapíše výraz ve formě dialogu. Používá se pouze pro ladění skriptu. Komunikace s uživatelem skriptu je řešena formou masek a dialogových funkcí z dll knihoven.

Příklady:

```
write (i+1)
write ("meruňky")
```

**Spuštění procedur**

Spuštění procedury aktivuje proceduru určenou zadaným názvem. Pokud deklarace procedury obsahuje seznam formálních parametrů, musí být při spuštění procedury k dispozici seznam aktuálních parametrů (parametry deklarace se označují jako formální, parametry spuštění procedury jako aktuální parametry). Při spuštění jsou aktuální parametry předány formálním parametrům.

Příklad:

```
Find (Name, Address)
```

Poznámka:

Procedury musí být před spuštěním deklarovány.

4.4.9 Strukturované příkazy

Strukturované příkazy se skládají z příkazů, které se buď opakují nebo se provádí v závislosti na určité podmínce.

Podmíněné příkazy

Podmíněný příkaz provede po vyhodnocení podmínky některé nebo taky žádné z dílčích příkazů.

Podmínka **if** výraz **then** příkaz **end** nebo
 if výraz **then** příkaz **else** příkaz **end**

Příklad:

```
if X<1,5 then
  Z:=X+Y
else
  Z:=1,5
end
```

Výsledek výrazu musí mít typ *boolean*. Pokud bude výsledkem *true*, bude proveden příkaz následující po rezervovaném slovu *then*. Pokud je výsledkem *false* a existuje větev *else*, provede se příkaz následující po rezervovaném slovu *else*. Pokud ale větev *else* chybí, program pokračuje příkazem, který následuje.

Příkazy cyklu

Příkazy cyklu se skládají z příkazů, které se mají provádět opakovaně.

- příkaz **while**
- příkaz **for**
- příkaz **foreach**

Příkazy **break** a **continue** můžete průběh opakovacích příkazů ovlivnit. **continue** pokračuje další iterací.

Příkaz while

Příkaz **while** řídí přes výraz opakované provádění příkazu.

while výraz **do** příkaz **end**

Příklad:

```
i:=0
while i<10 do
  i+=1
end
```

Výraz řídicí opakování musí být typu *boolean*. Výraz je vyhodnocen před provedením příkazu. Pokud je výsledkem *true*, provádí se příkaz stále dokola. Pokud je ale výsledkem výrazu *false*, nebude příkaz proveden ani jednou.

Příkaz for

Příkaz **for** řídí opakování příkazu podle měnící se hodnoty parametru až do dosažení stanovené hodnoty parametru. Počáteční hodnotu parametru nastaví příkaz1.

for příkaz1 **to** výraz **do** příkaz **end**

Příklad:

```
for i:=0 to 9 do
  write(i)
end
```

Příkaz foreach

Příkazem **foreach** se příkaz vykonává opakovaně, zatímco kontrolní proměnná *user* projde buď všechny uživatele databáze nebo uživatele, kteří jsou specifikováni v patřičném seznamu (*user in ...*).

foreach parametr **do** příkaz **end**

Příklad:

```
foreach user do
  if user.name="Novák" then
    result+=user
  end
end
```

end

4.4.10 Objekty

Přehled objektů

Objekt	Popis
muser	Implicitní objekt M/User
actuser	Objekt actuser je typu user . Představuje právě přihlášeného uživatele.
user	Představuje uživatele. Objekt user má řadu stálých vlastností a libovolný počet variabilních vlastností (atributů), na které se lze dotázat pomocí jazyka skriptů.
usercollection	Seznam uživatelů.
wkf	Používá se pro workflow-proměnné (globální) instance workflow.
date	Objekt typu datum.
document	Objekt document umožňuje přístup k procesu. Lze např. přidat nového účastníka nebo poznámky k běžící instanci workflow.

Implicitní objekt muser

Objekt **muser** představuje seznam všech uživatelů definovaných v M/User.

Metody používané pro objekt **muser**...

Metoda	Popis
getuserbyid	Uživatel je vybrán podle ID. Syntaxe: <code>getuserbyid (key: string) : user</code>
getuserlistbydepartment	Vrací seznam uživatelů z oddělení uvedeného jako první parametr. Druhý parametr určuje, zda budou zahrnuta do výběru i pododdělení (true). Syntaxe: <code>getuserlistbydepartment (department: string, rekursiv: boolean) : usercollection</code>
getuserlistbytemplate	Vrací seznam uživatelů s rolí uvedenou jako parametr. Syntaxe: <code>getuserlistbytemplate (template: string) : usercollection</code> Poznámka: Lze použít též tvar <code>getuserlistbyrole</code>

Metoda	Popis
getusersbyattribute	<p>Vrací seznam uživatelů s atributem a jeho hodnotou uvedené jako parametry.</p> <p>Syntaxe: <code>getusersbyattribute (attributeName:string, attributeValue:string) :usercollection</code></p> <p>script UzivatelePodleAtributu() :usercollection <code>begin</code> <code> result:=getusersbyattribute ("Odd", "Nákup")</code> <code>end</code></p>

Objekt user

Stálé vlastnosti

Následující tabulka obsahuje stálé vlastnosti pro objekt user, actuser a usercollection...

Stálá vlastnost	Typ	Význam
company	string	firma
department	string	oddělení
firstname	string	křestní jméno
location	string	umístění
LogonID	string	LogonID
name	string	jméno
reportstoguid	user	ID nadřízeného
room	string	číslo místnosti
shortname	string	zkratka (jméno, pod kterým se uživatel zobrazuje)
reportstoshortname	string	zkratka nadřízeného
telephone	string	telefon
UserID	string	jednoznačný klíč uživatele

Atributy

Na všechny ostatní atributy přidělené uživateli se můžete dotázat.

Poznámka:

U těchto atributů by měla být uvedena kompletní cesta (viz Příručka M/User). Pokud je atribut jednoznačný, nemusí být cesta uvedena.

Uživatel má například následující atributy...

- AttributeRights.Muser.General.MuserStartGui
- Attribute.b001.Limit
- Attribute.b002.Limit

U prvního atributu je cesta jednoznačná a není proto nutné ji uvádět.

Příklad:

```
begin
  foreach user do
    if user.MuserStartGui then
      result+=user
    end
  end
end
```

Atribut Limit není jednoznačný, proto je nutné cestu uvést:

Příklad:

```
begin
  foreach user do
    if user.Attribute.b001.Limit>10000 then
      result+=user
    end
  end
end
```

Pro flexibilní přístup může být část cesty nahrazena proměnnými typu *string*.

Příklad:

```
var s:string
begin
  s:=wkf.ucetni_okruh //obsahuje b001 nebo b002
  foreach user do
    if user.Attribute.s.Limit>10000 then
      result += user
    end
  end
end
```

Objekt wkf

Vlastnosti objektu wkf...

Vlastnost	Popis
<VariableName>	Vlastnost <VariableName> je stejného typu jako proměnná, kterou reprezentuje a představuje hodnotu proměnné.
errornumber	Vlastnost errornumber je typu integer. Slouží k předání čísla chyby ve workflow procesu.
errorstring	Vlastnost errorstring je typu string. Slouží k předání chybového hlášení ve workflow procesu.

Objekt document

Objekt **document** umožňuje přístup k procesu.

Vlastnosti objektu document

Vlastnost	Popis
notes []	Vlastnost notes vrací poznámku v procesu podle její pozice. Index začíná od 0, tj. první poznámka má index 0, druhá 1 atd. Syntaxe: <code>document.notes[Index:integer]:string</code>

Vlastnost	Popis
notecount	Vlastnost notecount vrací počet poznámek v procesu. Syntaxe: <code>document.notecount:integer</code>
users []	Vlastnost users vrací ID účastníka procesu podle pořadí. Index začíná od 0, tj. první účastník má index 0, druhý 1 atd. Syntaxe: <code>document.users[Index:integer]:string</code>
usercount	Vlastnost usercount vrací počet účastníků v procesu. Syntaxe: <code>document.usercount:integer</code>

Metody pro objekt document

Metoda	Popis
adduser	Metoda adduser přidá do procesu nového účastníka. Syntaxe: <code>document.adduser(UserGUID:string)</code> nebo <code>document.adduser(User:usercollection)</code>
newnote	Metoda newnote přidá do procesu poznámku. Syntaxe: <code>document.newnote(text:string)</code>

Objekt date

Objekt date představuje datum. Metody pro objekt **date**...

Metoda	Popis
getdate	Metoda getdate vrací datum jako string z proměnné typu date nebo string obsahující datum. Syntaxe: <code><variable>.getdate(): string</code>
getday	Metoda getday vrací den jako string z proměnné typu date nebo string obsahující datum. Syntaxe: <code><variable>.getday(): string</code> Příklad: <pre>var s : String d : Date begin d := "10.11.2030" s := d.getday() //s="10" end</pre>
getmonth	Metoda getmonth vrací měsíc jako string z proměnné typu date nebo string obsahující datum. Syntaxe: <code><variable>.getmonth(): string</code> Příklad: <pre>var s : String d : Date</pre>

Metoda	Popis
	<pre>begin d := "10.11.2030" s := d.getmonth() //s="11" end</pre>
getyear	<p>Metoda getyear vrací rok jako string z proměnné typu date nebo string obsahující datum.</p> <p>Syntaxe: <code><variable>.getyear(): string</code></p> <p>Příklad:</p> <pre>var s : String d : Date begin d := "10.11.2030" s := d.getyear() //s="2030" end</pre>
setday	<p>Metoda setday nastaví den v proměnné typu date nebo string obsahující datum.</p> <p>Syntaxe: <code><variable>.setday(day: string)</code></p> <p>Příklad:</p> <pre>var d : Date begin d := "10.11.2030" d.setday("15") //d=15.11.2030 end</pre>
setmonth	<p>Metoda setmonth nastaví měsíc v proměnné typu date nebo string obsahující datum.</p> <p>Syntaxe: <code><variable>.setmonth(month: string)</code></p> <p>Příklad:</p> <pre>var d : Date begin d := "10.11.2030" d.setmonth("10") //d=10.10.2030 end</pre>
setyear	<p>Metoda setyear nastaví rok v proměnné typu date nebo string obsahující datum.</p> <p>Syntaxe: <code><variable>.setyear(year: string)</code></p> <p>Příklad:</p> <pre>var d : Date begin d := "10.11.2030" d.setyear("2009") //d=15.11.2009 end</pre>

Rozšiřující metoda Count

Metoda count je použitelná pro více objektů...

Metoda	Popis
count	<p>Metoda count vrací počet položek v atributu typu list</p> <p>Syntaxe např.:</p> <pre><user>.Attribute.<attribute path>.count: integer</pre> <p>Příklad:</p> <pre>var i: Integer s: String begin for i:=0 to actUser.Attribute.StrList.count-1 do s:= actUser.Attribute.StrList[i] end end</pre>

5 Standardní konstrukce skriptu

5.1 Přehled

Skript se skládá z těchto prvků:

- klíčové slovo
- název
- dvojtečka
- typ skriptu (návrátové hodnoty)
- blok s vlastní definicí a výsledkem skriptu

Jednotlivé prvky jsou odděleny tzv. "bílymi znaky", buď mezerou (blank, prázdný znak) nebo tabelátorem (Tab).

5.2 Deklarace skriptu

Deklarace skriptu se skládá ze dvou základních komponent...

- externí deklarace
- tělo skriptu

5.2.1 Externí deklarace

Externí deklarace, které se nachází mimo tělo skriptu, se definují podle následujícího vzoru...

declare procedure název **dll** "název knihovny" (parametry)

declare function název **dll** "název knihovny" (parametry):typ

Při deklaraci externích funkcí a procedur se rozlišuje při zadávání formálních parametrů mezi předáním **per value** a **per reference**.

Typ **string** se předává vždy **per reference**. Všechny ostatní typy se předávají **per value**.

Předání **per reference** dosáhnete pomocí klíčového slova **var**. U typu **string** si také musíte uvědomit, že standardně je k dispozici pouze 255 znaků.

Za deklarací procedury nebo funkce může být použit parametr **nounload**, který zajistí, že dll knihovna nebude uvolněna z paměti a ušetří se tak čas potřebný pro její znovunačtení.

Příklad:

```
declare function Test1 dll "myLibrary" (String, String): Boolean nounload
```

5.2.2 Tělo skriptu

Tělo skriptu se skládá z následujících prvků...

script název () :typ blok

Příklad:

```
script faktura_vydana () : boolean
var i: boolean // začátek bloku
    cislo_faktury: string
begin
    result := idx("Maska_faktury_vydane")
    cislo_faktury := "Číslo faktury " + wkf.CisFak
    document.newnote(cislo_faktury)
end // konec bloku
```

Typ skriptu

Typ skriptu odpovídá definovaným typům dat. Nejčastěji se používají skripty typu

- Boolean (prováděcí skripty a skripty hran)
- User/Usercollection (vstupní skripty)
- Void (eskalační skripty).

Blok

Blok je vlastní definice procesu, odpovídá následujícímu vzoru...

deklarace proměnných

begin

příkazy

end

Deklarace proměnných

Začíná klíčovým slovem **var**, následují jména proměnných podle typů, oddělená čárkami. Po dvojtečce se zapíše typ proměnných.

Příklad:

```
var i,j,k: Integer
    chyba,error: String
```

Výsledek skriptu

Používá se klíčové slovo **result** ve významu proměnné, která nabývá hodnoty podle konkrétního typu skriptu. Pokud je skript typu **void**, není **result** definován.

5.2.3 Komentář

V jazyce skriptů se užívají následující komentáře...

```
//Libovolný text mezi dvojitým lomítkem a koncem řádku  
/*Libovolný text, který neobsahuje lomítka společně s hvězdičkou*/
```

Zadané názvy podskriptů je možné proklepávat pomocí klávesy **Ctrl + levé tlačítko myši**.

5.3 Příklady

5.3.1 Zásady tvorby workflow

Z praxe vyplynula potřeba stanovit některé zásady a pravidla při tvorbě workflow. Důvodem je koordinace a spolupráce v pracovním týmu (zastupitelnost pracovníků vytvářejících workflow atd.). Viz příklad prováděcího skriptu KH_schvaleni_faktury:

- Maska se jmenuje stejně jako skript. V názvu jsou dvě písmena firmy, pro které se WKF implementuje. Za podtržítkem pak popis pracovního kroku. př. "KH_schvaleni_faktury"
- Masky je umístěna ve složce odpovídající názvu workflow. Při importu se dává vše do rootu složek a je nutné masky umístit do složky ručně
- Vlastnosti masky – pozice okna: "Střed obrazovky"
- Vlastnosti masky – vyplnění Titul
- Vlastnosti masky – pokud nejde o masku pouze se dvěma tlačítky, pak pole Standardní tlačítka a Stavová lišta jsou nastavena na "Ne"
- Vlastnosti masky – pokud je přímo z masky volána nějaká funkce z DLL, tak vlastnost Povolit klíče je nastavena na "Pouze klíče masky"
- Ovládací prvky určené pouze pro čtení jsou vybarveny šedivě.
- Tlačítka jsou popsána v jednom slovním tvaru (obvykle infinitiv)
- Řídící tlačítka jsou taková, která určují směr workflow. Tato budou vždy umístěna ve spodní řadě a to v pořadí zleva: HELP (malé tlačítko, pouze pokud je help vyžadován), tlačítko odpovídající standardnímu průchodu WKF (v tomto případě Schválit), dále další tlačítka dle potřeby. Vpravo pak Uložit a Zavřít, které ukládá hodnoty proměnných a úplně vpravo tlačítko Zrušit, které hodnoty proměnných neukládá.
- Vlastnosti každého tlačítka jsou nastaveny na Zavřít dialog = ANO , Výsledek dialogu = ANO
- Při stisknutí tlačítka se provádí nastavení řídicí proměnné wkf.jakDal na hodnoty odpovídající názvům tlačítek (bez diakritiky). V tomto případě help, schvalit, neschvalit, zavřít a zrusit
- Ovládací prvky jsou zarovnány do bloků
- Je nastaven rozumný chod kurzoru na tlačítko Tab (viz str. 86)
- U kalendářových polí je obvykle nastavena výchozí hodnota na aktuální datum
- U Skupiny přepínačů (RadioButton) a Rozbalovacího seznamu (ComboBox), kde se hodnoty načítají ze souboru, který byl vygenerován ve skriptu, je tento umístěn v adresáři TEMP aktuálně přihlášeného uživatele.

5.3.2 Vstupní skripty

Skript Act_User

```
script Act_User (): usercollection
begin
  result += actuser
end
```

Skript **Act_User** dodá seznam uživatelů (aktuálního uživatele).

Skript User_template

```
script User_template (): usercollection
begin
  result := getuserlistbytemplate ("Vyroba")
end
```

Skript dá k dispozici všechny uživatele, kteří mají roli **Vyroba**.

Skript User_department

```
script User_Department (): usercollection
begin
  foreach user do
    if user.department = "Vyroba" then
      result += user
    end
  end
end
```

Skript dá k dispozici všechny členy oddělení **Vyroba**.

Skript pracovnik

```
script pracovnik ():usercollection
begin
  result += getuserbyid(wkf.pracovnik)
end
```

Skript **pracovnik** najde uživatele, jehož ID souhlasí s obsahem proměnné **wkf.pracovnik**.

Skript ukol_pracovnik

```
script ukol_pracovnik (): usercollection
begin
  if wkf.pracovnik<>"" then
    result:=getuserbyid(wkf.pracovnik)
  else
    result:=actuser
  end
end
```

Pro případ, že by **wkf** proměnná neobsahovala ID pracovníka, je k předchozímu skriptu přidána možnost nastavit do výsledku skriptu aktuálního účastníka.

Skript TeamBridge

```
script Teambridge (): usercollection
begin
  result := getuserbyid("_teambridge_")
end
```

5.3.3 Skript hrany

Skript HranaSCHVALENO

```
script HranaSCHVALENO (): Boolean
begin
  if wkf.JakDal="schvalit" then
    result := true
  end
end
```

Skript **HranaSCHVALENO** vrací hodnotu **boolean** podle výsledku vyhodnocení obsahu proměnné **wkf.JakDal**. Pokud tato obsahuje zadaný text, je výsledek **true**. Tento skript lze použít jako skript hrany (otvírá nebo zavírá hranu podle výsledku skriptu).

5.3.4 Prováděcí skripty

Skript Vypracovani_ukolu

```
Declare procedure QuestionDialog dll "actualDocument" (string,string)
Declare procedure SetComplete dll "actualDocument" (string,pointer)
```

```
script Vypracovani_ukolu (): Boolean
var vystup,guid : string
begin
  vystup := "Potvrzení"
  QuestionDialog("Potvrdit dokončení úkolu?",vystup)
  if vystup = "ANO" then
    result := true
    document.newnote("Potvrzují dokončení úkolu.")
    guid := actuser.userID
    SetComplete(guid,actualdocument)
  end
end
```

Prováděcí skript, kde je místo masky použit pouze rozhodovací dialog realizovaný deklarovanou procedurou **QuestionDialog**. Procedura **SetComplete** umožní uživateli dokončit krok workflow.

Skript odpoved_na_prihlasku

```
script odpoved_na_prihlasku (): Boolean
var text : String
begin
  result := idx("Maska_odpoved_na_prihlasku")
  if wkf.zaplmeno = 0 then
    text += "Potvrzují Vaši účast na semináři."
  end
  if wkf.zaplmeno = 1 then
    text += "Bohužel seminář je již zaplněn."
  end
  if wkf.zaplmeno = 2 then
    text += "Ještě se Vám ozveme."
  end
  document.newnote(text)
end
```

Prováděcí skript s použitím volání masky, ve které je nastavena proměnná **wkf.zaplmeno**. Podle toho je vybrán text, který bude přidán do procesu a může být následně odeslán mailem ap.

Skript test_DBconnect

```
declare function TB_ParamsAddStr dll "dynOLEEng" (integer,string): integer
declare function TB_CreateParams dll "dynOLEEng" ():integer
declare function DB_ConnectWithParams dll "dynOLEEng" (integer,string,string):integer
declare function DB_CreateSQLQuery dll "dynOLEEng" (string):integer
```

```

declare Procedure DB_SQLQueryNext dll "dynOLEEng" (Integer)
declare procedure DB_OpenSQLQuery dll "dynOLEEng" (integer,integer,var Boolean)
declare Procedure DB_SQLQueryEof dll "dynOLEEng" (Integer,var boolean,var boolean)
declare Function DB_QueryFieldByName dll "dynOLEEng" (Integer,string,var Boolean):integer
declare Function TB_GetResultStr dll "dynOLEEng" (Integer): string
Declare procedure DB_ExecutesSQLQuery dll "dynOLEEng"(Integer,Integer,var Boolean)
Declare procedure DB_FreeSQLQuery dll "dynOLEEng"(Integer)
Declare procedure DB_Disconnect dll "dynOLEEng"(integer)
declare procedure GetItemFromMTBini dll "actualDocument" (string,string,string)

script test_DBconnect (): Boolean
var DBname, DBtype : String
  DBlogin, DBpass : String
  SQLCOL, s : String
  conn, sqlID, columnID, iParamsId : Integer
  eof, success : Boolean
begin
  // standardní uživatel pro Firebird
  DBlogin := "sysdba"
  DBpass := "masterkey"

  GetItemFromMTBini("M/TEAMBRIDGE","server",DBname) // označení databáze
  // pro Firebird zde bude načten typ INTERBASE
  GetItemFromMTBini("M/TEAMBRIDGE","server",DBtype) iParamsId:=TB_CreateParams()

  TB_ParamsAddStr(iParamsId,DBtype)
  TB_ParamsAddStr(iParamsId,DBname)
  TB_ParamsAddStr(iParamsId,"ANCZECH")
  TB_ParamsAddStr(iParamsId,"")
  TB_ParamsAddStr(iParamsId,"3")
  conn:=DB_ConnectWithParams(iParamsId,DBlogin,DBpass)
  SQLCOL := "select * from muuser" // SQL dotaz nebo příkaz
  sqlID:=DB_CreateSQLQuery(SQLCOL)
  DB_OpenSQLQuery(sqlID, conn,success) // provede SQL dotaz
  //Pokud se má provádět SQL příkaz (UPDATE,INSERT), tak se místo OpenSQLQuery
  //použije příkaz DB_ExecutesSQLQuery
  DB_SQLQueryEof(sqlID,eof,success)
  while not eof do //prochází jednotlivé řádky výsledku
    columnID:=DB_QueryFieldByName(sqlID,"loginID",success)
    s:=TB_GetResultStr(columnID)
    //..... zde může následovat dotaz na další sloupce z řádku výsledku.
    //Pro každý sloupec to budou dva příkazy, tak jako u loginID.
    DB_SQLQueryNext(sqlID)
    write(s)
    DB_SQLQueryEof(sqlID,eof,success) //testuje, zda ještě není konec seznamu výsledků
  end
  DB_FreeSQLQuery(sqlID)
  DB_Disconnect(conn)
end

```

Skript demonstruje možnosti napojení skriptovacího jazyka M/TeamBridge na SQL databáze. V příkladu je uvedeno napojení na databázi TeamBridge, kde její umístění je načteno z INI souboru. Proměnná DBtype může obsahovat hodnoty INTERBASE, ORACLE a nebo MSSQL. Po napojení jsou načtena loginID všech uživatelů z tabulky z tabulky MUUSER.

Skript KH_schvaleni_faktury

The screenshot shows a Windows-style dialog box titled "Schválení faktury". It has a blue title bar with standard window controls. The main area is light gray and contains the following elements:

- A text box for "Číslo faktury" containing the character "0".
- A text box for "Číslo objednávky" which is empty.
- A "Načíst" button to the right of the "Číslo objednávky" box.
- A large empty text area for "Poznámka".
- A row of buttons at the bottom: "Help", "Schválit", "Neschválit", "Uložit a zavřít", and "Zrušit".

```

Declare procedure MessageBox dll "actualDocument" (string, string)
script KH_schvaleni_faktury (): Boolean
begin
  wkf.jakDal := "maska"
  while wkf.jakDal = "maska" do
    wkf.jakDal := "zrusit"
    idx("KH_schvaleni_faktury")

    //blok podmínek
    if wkf.jakDal = "neschvalit" and wkf.poznamka = "" then
      MessageBox("V případě neschválení dopište poznámku.", "w")
      wkf.jakDal := "maska"
    end

    if wkf.jakDal = "schvalit" then
      document.newnote("Fakturu schvaluji")
      result := true
    end

    if wkf.jakDal = "neschvalit" then
      document.newnote("Fakturu neschvaluji")
      result := true
    end

    if wkf.jakDal = "zavrit" then
      result := true
    end

    if wkf.jakDal = "zrusit" then
      result := false
    end

  end //while
end

```

5.3.5 Eskalační skripty

Skript zakaznik_nezaplatil

```

Declare function AddTextToProcessLikeUserTeamBridge dll
"actualDocument" (string, string): boolean

```

```

script zakaznik_nezaplatil (): Void
var text:string
begin
    text:="Kontaktovat zákazníka - nezaplatil fakturu."
    AddTextToProcessLikeUserTeamBridge (wkf.DD_PO_docID, text)
end

```

Po uplynutí nastavené doby přidá do procesu text.

Skript pridat_reditele

```

script pridat_reditele (): Void
var guid:string
begin
    guid:=getuserlistbytemplate ("reditel")
    document.adduser (guid)
end

```

Po uplynutí nastavené doby přidá do procesu účastníka.

Skript provest_krok

```

declare procedure ExecuteStepInActualProcess dll "actualDocument"
(string, pointer)
script provest_krok (): Void
begin
    ExecuteStepInActualProcess ("Výběr termínu", actualDocument)
end

```

Po uplynutí zadané doby provede krok v aktuálním procesu. Jméno tohoto kroku je třeba zadat jako parametr procedury.

Další možnosti jsou :

- založit proces
- předat krok jinému účastníkovi atd.

Poznámka:

Tyto skripty se používají pro klasické eskalace tvořené přes vlastnosti pracovního kroku. Další možnost je vytvářet eskalace procesů pomocí procedur a funkcí z knihovny dll actualdocument (funkce SetEscalation). V tom případě mohou být skripty pro eskalace typu **boolean**.

6 Workflow designer

6.1 K čemu slouží Workflow Designer?

Program slouží k definici nových workflow. Společně se skripty zde můžete nadefinovat workflow pro standardizované procesy M/TeamBridge.

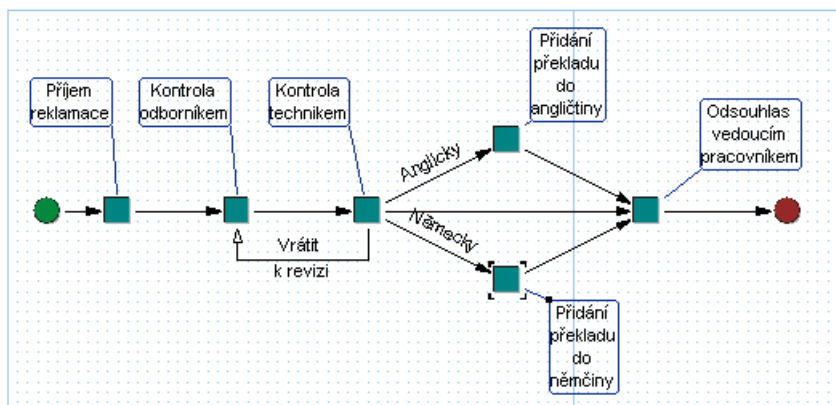
Workflow tvoří předem definovaný proces.

Představuje strukturu a postup procesu a tudíž ukazuje...

- jednotlivé pracovní kroky
- jejich sled
- osoby, které budou jednotlivé kroky provádět

Workflow se zobrazí pomocí takzvaného grafu workflow, který může být lineární, kruhový nebo hvězdicový.

Nebo může vypadat například takto...



Jednotlivé kroky se nazývají uzly. Každé workflow je omezeno počátečním a koncovým uzlem. Šipky, které jsou v tomto kontextu označovány jako hrany, ukazují směr postupu procesu. Popis uzlů a hran informuje o obsahu pracovního kroku.

6.2 Obecné funkce

6.2.1 Přihlášení

Přihlašovací dialog se zobrazí pouze tehdy, nemáte-li puštěný žádný jiný produkt SERIE/M...

Spustit Workflow-Designer mohou pouze ti uživatelé, kteří mají v aplikaci M/User u atributu „MTBAllowWKFD“ nastavenou hodnotu *true*.

6.2.2 Odhlášení

Pokud máte otevřené workflow, objeví se před zavřením programu dialog, zda si přejete vytvořené nebo změněné workflow uložit.

6.2.3 Pracovní plocha

Obsahuje následující prvky...

- lištu menu s nabídkami
- panel nástrojů
- okno WORKFLOW pro vytváření nových workflow a zobrazení skriptů
- okno EDITOR VLASTNOSTÍ, ve kterém se přiřazují vlastnosti uzlům nebo hranám
- okno PRŮZKUMNÍK OBJEKTŮ (alternativně s oknem editor vlastností na záložce) se stromovou strukturou objektů workflow
- okno NAVIGÁTOR, které slouží k jednoduššímu zacházení s workflow.


6.3 Administrace workflow

6.3.1 Správa workflow

Vytvořené workflow jsou ukládány v databázi M/TeamBridge a je možné s nimi dále pracovat.

Dialog administrace

Pro otevření dialogu ADMINISTRACE WORKFLOW...

- klepněte na ikonu  nebo
- vyberte v nabídce menu SOUBOR možnost OTEVŘÍT nebo
- použijte klávesovou zkratku <Ctrl> + <O>.

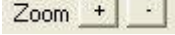


Zobrazení grafu workflow

Pro zobrazení grafu aktuálně označeného workflow klepněte na tlačítko




Tip:


Nad grafem workflow se nachází tlačítka , která umožňují zvětšení nebo zmenšení grafu. Pokud graf zmenšíte na určitou úroveň, zmizí popisky jednotlivých uzlů. Po zvětšení grafu se popisky opět zobrazí.

Vytvoření nového workflow

Pro vytvoření nového workflow...

- klepněte na tlačítko  nebo
- v menu na SOUBOR/NOVÝ nebo
- klávesovou zkratkou <Ctrl> + <N>
- Otevře se nová pracovní plocha pro vytváření grafu workflow.

Otevření workflow

- V dialogu ADMINISTRACE WORKFLOW poklepáním na požadované workflow nebo vybráním workflow a klepnutím na tlačítko  se graf workflow otevře na pracovní ploše.

Tip:

Je možné označit a otevřít i několik workflow zároveň.

Poznámka:

Při otevření workflow je proveden test konzistence, pokud je tato volba zaškrtnuta v menu NÁSTROJE/NASTAVENÍ na kartě Testování.

Uzavření workflow

Pro uzavření workflow...

- vyberte v nabídce menu SOUBOR možnost ZAVŘÍT nebo
- v popup menu možnost ZAVŘÍT nebo

- klávesu <Ctrl> + <F4>




Pokud jste neprovedli v procesu žádné změny, bude workflow uzavřeno bez dalšího komentáře. V opačném případě se zobrazí dotaz na uložení změn.

Uložení workflow

Ukládání workflow je svázáno s prováděním testu konzistence. Test správnosti workflow (test konzistence) přezkouší vnitřní logiku workflow.

Test se automaticky spustí při uložení workflow, pokud je to nastaveno v menu NÁSTROJE/NASTAVENÍ na záložce Nastavení.

Máte dvě možnosti, jak zjistit výsledek testu...

- Provést test správnosti klepnutím na ikonu . Po provedení testu správnosti se zobrazí okno s poznámkou, zda je workflow konzistentní. Pokud tomu tak není, obsahuje dialog i seznam nalezených chyb.
- Navíc se pod oknem s workflow zobrazuje automatické testování - světélko s komentářem. Pokud svítí zeleně  Připraven, je workflow po logické stránce v pořádku, svítí-li červeně  Připraven, nachází se ve workflow chyba. V takovém případě se podívejte do seznamu nalezených chyb.
- **Poznámka:** Tato kontrola není standardně nastavena.

Nastavení automatického testování

Pro nastavení automatického testování...

- vyberte z nabídky menu NÁSTROJE možnost NASTAVENÍ
- na záložce Testování zaškrtněte možnost AUTOMATICKÉ TESTOVÁNÍ.

Mazání workflow

Pro smazání workflow označte workflow v seznamu ADMINISTRACE WORKFLOW a zadejte příkaz WORKFLOW/SMAZAT WORKFLOW

Pro mazání můžete také využít možnost SMAZAT WORKFLOW v popup menu.

6.3.2 Migrace workflow

Export workflow

Pro export workflow existují dvě možnosti...

- standardním způsobem je exportován pouze model workflow
- při souhrnném exportu workflow je exportován model workflow a všechny připojené objekty

Standartní export workflow

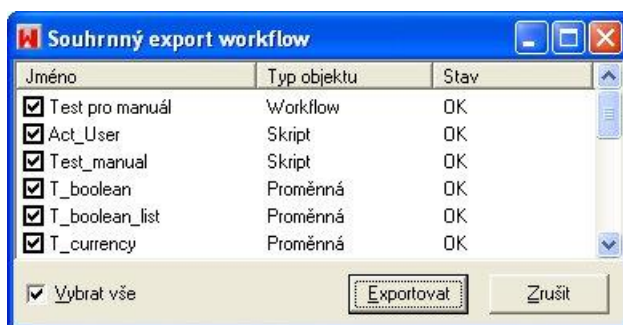
- V otevřené workflow zadejte z menu příkaz SOUBOR/EXPORT/STANDARTNÍ EXPORT . V dialogu pro export vyberte adresář, do kterého má být workflow exportováno, zadejte název a uložte.

Souhrnný export workflow

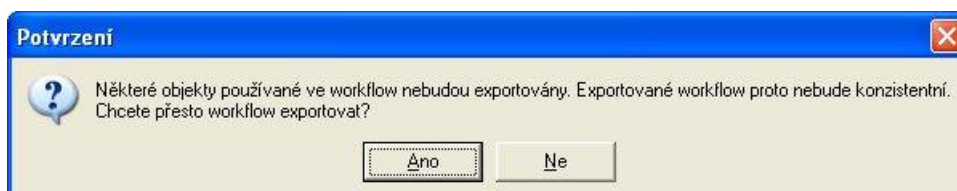
Souhrnný export workflow zahrnuje mimo modelu workflow také všechny související objekty, jako proměnné, termíny, skripty, masky, ikony a vzory dokumentů.

Pro souhrnný export workflow...

- V otevřeném workflow zadejte z menu příkaz SOUBOR/EXPORT/SOUHRNNÝ EXPORT, otevře se seznam všech exportovaných prvků...



- Klepněte do kontrolních políček v seznamu u těch položek, které nechcete exportovat (exportovány budou pouze položky se zaškrtnutými políčky).
- **Tip:** Pokud chcete exportovat jen několik málo prvků, deaktivujte jednoduše možnost VYBRAT VŠE dole v dialogu. Pak můžete zaškrtnout pouze požadované prvky.
- **Pozor:** Interně proběhne test konzistence a v případě nekonzistence budete na tento problém upozorněni. Na přání můžete poté export přerušit.




- Po potvrzení dialogu vyberte složku, do které má být workflow exportováno, zadejte název a potvrďte. Kompletní workflow bude uloženo jako zip-soubor.

Import workflow

Opět jsou k dispozici dvě varianty...

- standardním způsobem je importován pouze model workflow
- při souhrnném importu workflow je importován model workflow a všechny připojené objekty

Standartní import workflow

- Klepněte na ikonu , v dialogu ADMINISTRACE WORKFLOW označte složku, do které má být workflow importováno a zadejte z menu příkaz SLOŽKA/IMPORT
- Otevře se dialog pro import, ve kterém jsou zobrazeny soubory typu **.wkf**. Vyberte workflow, které má být naimportováno, zadejte název a potvrďte.

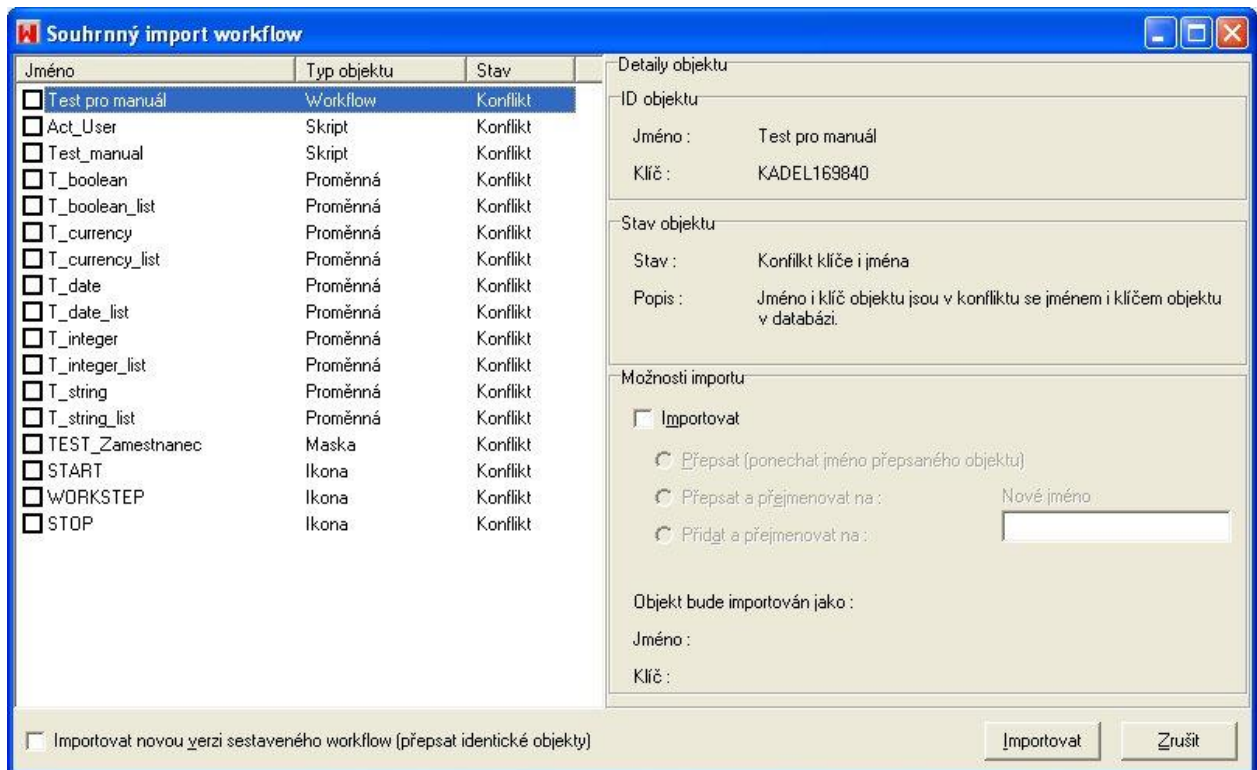
Workflow je naimportováno do vybraného adresáře a je k dispozici pro zpracování v programu Workflow Designer.

Souhrnný import workflow

Souhrnný import workflow zahrnuje mimo modelu workflow také všechny související objekty, jako proměnné, termíny, skripty, masky, ikony a vzory dokumentů.

Souhrnný import se používá u workflow, které byly exportovány jako soubor **.zip** a mají být přeneseny do jiné databáze.

- Zadejte z nabídky menu příkaz SOUBOR/OTEVŘÍT. V nabídce typ souboru vyberte **.zip** soubory a vyberte soubor, který budete importovat. Zobrazí se dialog pro souhrnný import...



- Vlevo je seznam jednotlivých objektů workflow. Na pravé straně dialogu lze provádět nastavení jednotlivých objektů pro import. Pokud jsou všechna nastavení správně, klepněte na tlačítko Importovat.

Konflikty objektů

Pokud některé z importovaných objektů workflow už v databázi existují, vzniká konflikt. Kontrolní pole před názvem objektu není zaškrtnuté a objekt tak nebude zahrnut do importu.

V takovém případě máte na výběr z následujících možností...

- neimportovat objekty, které jsou v konfliktu
- klepnout na možnost IMPORTOVAT NOVOU VERZI SESTAVENÉHO WORKFLOW (PŘEPSAT IDENTICKÉ OBJEKTY). Tak budou přepsány novou verzí i ty objekty, které jsou již uloženy v databázi.
- nastavit import pro objekty v konfliktu individuálně

Nastavení importu pro jednotlivé objekty

Pro individuální nastavení importu u objektů v konfliktu označte objekt v seznamu. V pravé části okna se zobrazí detaily označeného objektu. V části MOŽNOSTI IMPORTU máte na výběr tři varianty importu objektu.

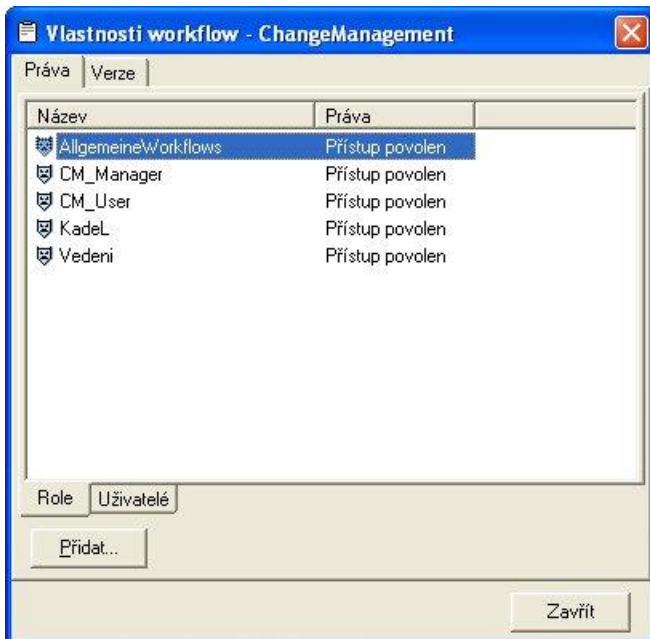
1. přepsat objekt v databázi importovaným objektem
 2. přepsat objekt v databázi a zadat nový název tohoto objektu
 3. uložit objekt do databáze rovnou pod novým jménem, které zadáte do pole. Stávající objekt tak nebude přepsán.
- Pokud má být projekt nainportován, zaškrtněte kontrolní pole před názvem objektu.

6.3.3 Oprávnění

Ve Workflow Designeru je cíleně řízeno, kdo má nebo nemá přístup k workflow. Standardně nemá žádný uživatel oprávnění spouštět workflow.


Dialog pro nastavení práv

- V dialogu ADMINISTRACE WORKFLOW označte workflow a v jeho popup menu vyberte možnost VLASTNOSTI. Zobrazí se dialog se záložkami PRÁVA a VERZE





- Zde můžete přidělit rolím nebo jednotlivým uživatelům právo spouštět workflow.




Oprávnění role

- V dolní části dialogu se přepněte na záložku ROLE, pokud již není aktivní, klepněte na tlačítko . Otevře se okno se seznamem rolí, kde vyberte požadovanou roli (více rolí pomocí klávesy <Ctrl>). Výběr potvrďte tlačítkem OK.

Poznámka:

Uživatelé role, které bylo přiděleno oprávnění, jsou automaticky přiřazeni do seznamu uživatelů na druhé záložce a jsou označeni symbolem , případně  pokud už měli oprávnění podle jména.

Oprávnění uživatele

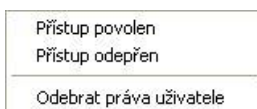
- V dolní části dialogu se přepněte na záložku UŽIVATELÉ, klepněte na tlačítko . Otevře se okno se seznamem uživatelů M/Teambridge. Vyberte požadovaného uživatele a výběr potvrďte. Uživatel bude označen symbolem  nebo  pokud už měl oprávnění podle role.

Poznámka:

Seznam uživatelů, kteří mají oprávnění pro dané workflow, najdete na druhé záložce. Jsou zde i uživatelé, kterým bylo oprávnění přiděleno prostřednictvím některé role. Nastavení zde můžete měnit individuálně pro každého jednotlivého uživatele.

Povolení a odepření přístupu

Po označení role nebo uživatele v seznamu můžete změnit oprávnění v popup menu.



Vybráním možnosti PŘÍSTUP ODEPŘEN znemožníte označenému uživateli nebo uživateli označené role spouštět workflow. Uživatel ale zůstává v seznamu.

Odebrání práva uživateli

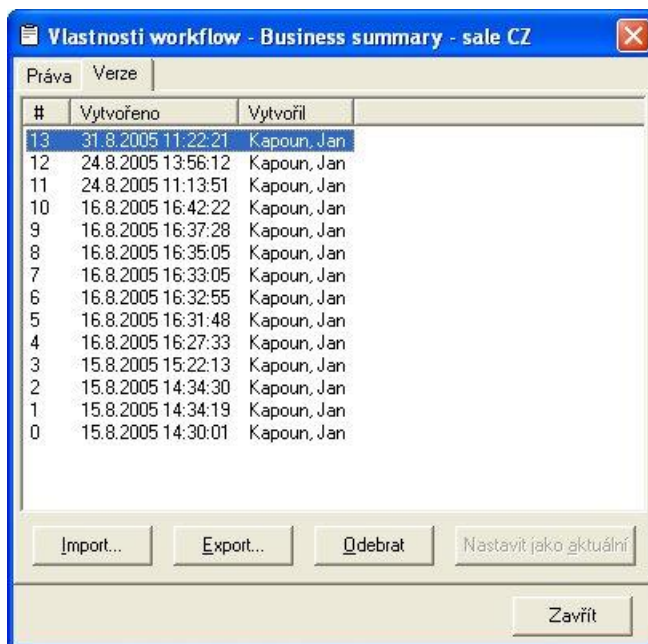
Pomocí této možnosti uživateli odeberete právo spouštět workflow a definitivně ho odstraníte ze seznamu.

6.3.4 Verze workflow


Všechny verze modelu workflow jsou ukládány. Kdykoliv se tak můžete k jednotlivým verzím vrátit, prohlédnout si je nebo aktivovat. Správa verzí je organizována jako vlastnost workflow.

Prohlížení verzí workflow

- V dialogu ADMINISTRACE WORKFLOW označte workflow a v jeho popup menu vyberte možnost UKÁZAT VERZE. Na záložce VERZE vidíte seznam všech verzí...



Mazání verzí workflow

Pro smazání některé z verzí workflow označte verzi, kterou chcete smazat a klepněte na tlačítko .

Poznámka:

Číslování verzí zůstává zachováno, aby bylo i později prokazatelné, že na tomto místě byla smazána verze.

Aktualizace verze workflow

Označte verzi, na kterou chcete aktualizovat a stiskněte tlačítko .

Poznámka:

Verze se zkopíruje na nejvyšší pozici, ale zároveň zůstává na svém místě.

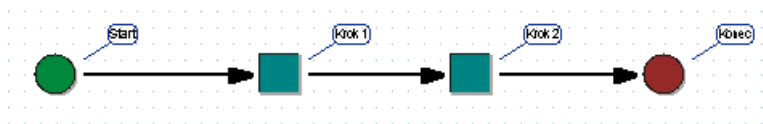
6.4 Graf workflow

Graf workflow je grafické znázornění standardizovaného procesu. Skládá se z počátečního a koncového uzlu, jednotlivých uzlů (*nodes*) a takzvaných hran (*flows*). Komponenty umísťujete do grafu pomocí ikon z panelu nástrojů nebo z nabídky menu OBJEKTY.


6.4.1 Konstrukce grafu workflow

Konstrukce grafu workflow je prvním krokem při vytváření workflow. Uspořádání grafu by mělo být co nejvíce přehledné. Je dobré si předem rozmyslet, kteří uživatelé a v jakém pořadí budou součástí workflow, kde je vhodný zpětný krok atd.


Workflow může v podstatě probíhat lineárně, hvězdicovitě nebo v kruhu. Můžete také vytvořit libovolnou kombinaci těchto forem. Struktura workflow závisí pouze na Vašich přáních a potřebách.



Počáteční uzel

Workflow vždy začíná počátečním uzlem. Klepněte nejdříve na ikonu pro umístění počátečního uzlu  a klepnutím do okna WORKFLOW vyberte pozici počátečního uzlu..


Uzly

Vyberte ikonu pro pracovní krok  a poté klepněte na místo, kde má být uzel umístěn v okně WORKFLOW.


Poznámka:

Pokud chcete vložit uzel mezi dva existující uzly, které jsou již spojeny hranou, umístěte uzel jednoduše na tuto hranu. Hrana se automaticky rozdělí a nový uzel se začlení do grafu.

Vnořené workflow

K workflow je možné připojit další, tzv. vnořené workflow. Vložíte je pomocí této ikony z panelu nástrojů . Postupujte stejným způsobem jako u jakékoliv jiné komponenty.


Koncový uzel

Pokud vytvoříte všechny potřebné uzly, umístěte na konec grafu koncový uzel. Použijte následující ikonu  a klepněte na pozici, kde má být koncový uzel umístěn.

Poznámka:

Z koncového uzlu není možné vést žádné další hrany!

Hrany

Uzly se spojují hranami. Klepněte na ikonu pro vytváření hran  a poté na uzel, ze kterého má hrana vést a se stisknutým levým tlačítkem myši pokračujte požadovaným směrem k uzlu, se kterým má být tento uzel spojen. Pro vytvoření dalších hran tento postup opakujte. S vytvářením hran začněte u počátečního uzlu a samozřejmě nezapomeňte připojit i koncový uzel.

Tip:

Pokud při vytváření hrany tlačítko myši pustíte hned na uzlu, ze kterého vycházíte, vytvoří se zpětná hrana kolem jednoho uzlu.


Poznámka:

Není možné vytvářet zpětné hrany směrem ke koncovému uzlu.

Mřížka

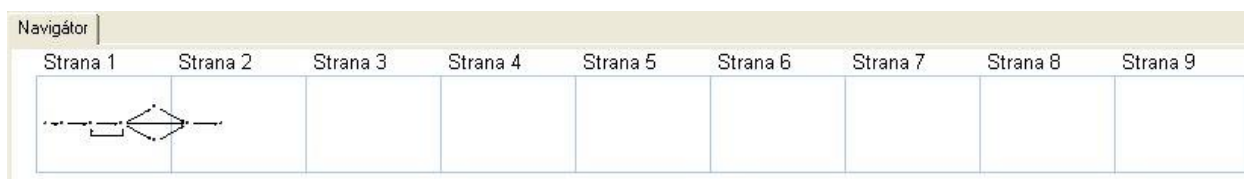
Jako pomoc při umísťování komponent do grafu workflow Vám slouží mřížka. Můžete si ji nechat zobrazit nebo zrušit pomocí možnosti NASTAVENÍ z nabídky menu NÁSTROJE. Standardně je nastaveno, aby se mřížka zobrazovala.

Pro zobrazení nebo zrušení mřížky zadejte z menu příkaz NÁSTROJE/NASTAVENÍ. Na první záložce NASTAVENÍ zaškrtněte nebo zrušte první možnost POUŽÍT MŘÍŽKU

Klepnutím na tlačítko  si můžete nastavit velikost mřížky podle vlastní potřeby:

Navigátor

Dole na obrazovce vidíte okno NAVIGÁTOR. Jedná se v podstatě o několik stran se zmenšeninami workflow, které vidíte paralelně s grafem vytvářeného workflow.



Obrázek znázorňuje jednotlivé komponenty workflow ve zmenšené formě. Pro komplikované workflow máte k dispozici až 15 stran. Slouží k jednodušší orientaci mezi jednotlivými stranami dlouhých workflow, které již není možné zobrazit celé.

Navigátor však neslouží pouze pro přehlednost, ušetří Vám také listování při hledání požadovaného úseku workflow. Jednoduše klepněte v okně navigátor na hledanou pozici a stránka se zobrazí v okně GRAF WORKFLOW.

Pokud nechcete okno NAVIGÁTOR zobrazit, klepněte v nabídce menu na možnost ZOBRAZENÍ/NAVIGÁTOR.

6.4.2 Úpravy grafu workflow

Přesouvání jednotlivých komponent

Všechny komponenty workflow můžete v grafu různě přesouvat nebo nově propojovat.

Tip:

Komponenty grafu workflow se při přesouvání orientují podle mřížky. To platí i pro body jednotlivých hran, kterými je možné měnit tvar hrany.

Přesunutí uzlu

- Přetáhněte objekt se stisknutým levým tlačítkem myši do požadované pozice.

Uzel se přesune a hrany, které od něho nebo k němu vedou se pohybují automaticky s uzlem.

Přesunutí hrany


- Označte hranu klepnutím, aby se zobrazily body. Přetáhněte počáteční, případně koncový bod hrany se stisknutým levým tlačítkem myši k novému uzlu.

Hrana se sváže s novým uzlem, ke kterému byla přesunuta.

Tip:

Při označení hrany se kromě počátečního a koncového bodu zobrazí ještě dva body uprostřed hrany. Pomocí nich můžete měnit tvar hrany a libovolně je přesouvat.

Vkládání stejných komponent

Pro opakované vložení stejné komponenty stiskněte klávesu <Shift> a současně klepněte na ikonu vybrané komponenty, poté už jen klepnutím umístíte komponenty do požadovaných pozic. Vkládání ukončíte stiskem .

Mazání komponent

- Označte myší komponentu, klepněte na tlačítko <Delete> nebo klepněte pravým tlačítkem myši a vyberte možnost SMAZAT. Vybraná komponenta bude z grafu smazána.

K mazání posledních změn můžete použít také možnost ZPĚT z nabídky menu EDITOVAT.

Zrušení akce

- Zadejte z menu příkaz EDITOVAT/ZPĚT nebo použijte klávesovou zkratku <Ctrl> + <Z>

Tato funkce vrací zpět všechny provedené aktivity v opačném pořadí než byly provedeny, tedy od poslední změny.

6.4.3 Řízení průběhu procesu

Podmíněné větvení a paralelita

Z uzlu v grafu workflow může vycházet několik hran. Pomocí skriptů, které přiřadíte jednotlivým hranám, můžete nadefinovat individuální podmínky. V průběhu zpracování workflow se rozhoduje prostřednictvím vyhodnocování stávající situace, kterou větví bude workflow pokračovat. Je možné postupovat několika větvemi zároveň, aktivity obsažené v jednotlivých krocích budou tedy prováděny paralelně.

Synchronizační uzly

Pokud k jednomu uzlu vede několik hran, jedná se o synchronizační uzel. Synchronizační uzel je aktivován teprve tehdy, jsou-li vyřízeny všechny předchozí uzly, od kterých vede hrana k tomuto uzlu.

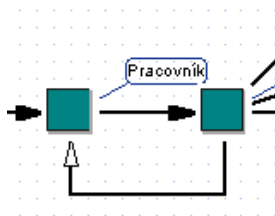
Přezkoušení předcházejících uzlů probíhá tak, že se kontrolují všechny hrany sbíhající se k synchronizačnímu uzlu až do počátečního uzlu. Při této kontrole není brán zřetel na zpětné hrany, protože následující uzel ještě nemusí být vyřízen nebo workflow nemuselo do následujícího kroku ještě dojít a synchronizační uzel, který je současně cílovým uzlem zpětného větvení, by nikdy nebyl aktivován.

Zpětné hrany

Zpětné hrany umožňují kontrolu jednotlivých uzlů modelu workflow. Aktivity nebo celé větve procesu se tak mohou v případě potřeby provádět opakovaně.


Protože není možné technicky rozpoznat, která hrana má být interpretována jako dopředná a která zpětná, musí se při vytváření modelu workflow její směr specifikovat.



V grafu jsou zpětné hrany označeny bílou (nevyplněnou) šipkou směřující zpět, jako vidíte na následujícím obrázku...



6.4.4 Kontrola správnosti workflow

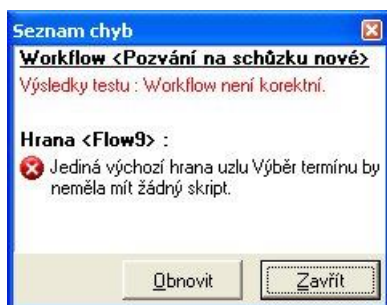
U každého vytvořeného workflow můžete provést takzvaný test správnosti workflow (test konzistence), který přezkouší vnitřní logiku workflow.

Test spustíte příkazem KONTROLA SPRÁVNOSTI z nabídky menu ZOBRAZENÍ nebo pomocí následující ikony . Máte dvě možnosti, jak zjistit výsledek testu:

- Pod oknem s workflow se zobrazí světélko s komentářem  **Připraven.**. Pokud svítí zeleně, je workflow po logické stránce v pořádku, svítí-li červeně  **Připraven.**, nachází se ve workflow chyba. V tomto případě se podívejte do seznamu nalezených chyb. Pokud není světélko zobrazeno, provedete jeho zapnutí v menu NÁSTROJE/NASTAVENÍ na kartě Testování zatržením volby Automatické testování.
- Po provedení testu se zobrazí okno s poznámkou, zda je workflow konzistentní. Pokud tomu tak není, zobrazí se seznam nalezených chyb.

Seznam nalezených chyb

Program Vás přesně informuje o tom, na jakém místě a jaká chyba se ve workflow objevila, takže ji můžete bez problémů odstranit.



Pokud jste provedli patřičné úpravy workflow, aktualizujte test klepnutím na tlačítko OBNOVIT v okně SEZNAM CHYB.

Pokud byly opraveny všechny chyby, zobrazí se pod grafem workflow symbol zeleného světla a v okně poznámka... Výsledek testu : Workflow je konzistentní.


Názvy objektů, které test používá, odpovídají názvům uzlů a hran zadaným v okně EDITOR VLASTNOSTÍ. Pokud jste nezadali žádná jména, užívá program označení (node+ID, flow+ID).

Poznámka:

Je nutné si uvědomit, že test je schopen přezkoušet pouze smysluplnost stavby komponent workflow. Jejich obsah v tomto testu není brán v úvahu.

6.4.5 Ukládání workflow

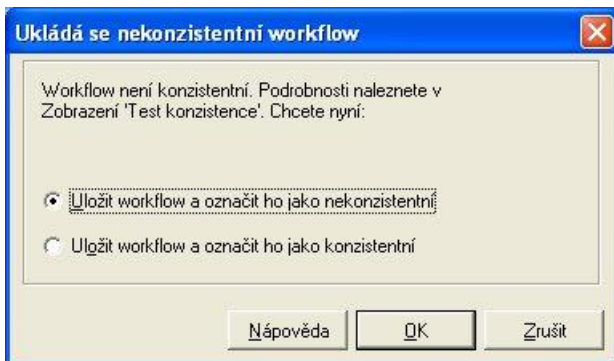
Pokud jste dokončili graf workflow, existuje několik způsobů, jak workflow uložit...

- Klepněte na ikonu s disketou  na levé straně panelu nástrojů nebo
- vyberte v nabídce menu možnost SOUBOR/ULOŽIT.

V každém případě se zobrazí standardní dialog, ve kterém zvolíte, kde chcete workflow uložit.

Ukládání nekonzistentního workflow

Pokud se pokoušíte uložit workflow, které neobstálo v testu konzistence, budete varováni následujícím hlášením...




Přestože není workflow konzistentní, není Vám v zásadě bráněno workflow uložit.

Máte na výběr...


- Uložit workflow jako konzistentní. V tomto případě si je administrátor pravděpodobně jistý, že workflow funguje i když nejsou splněny interní podmínky. Workflow může být v M/TeamBridge užíváno i přes zdánlivou nekonzistentnost.
- Uložit workflow jako nekonzistentní. Uživatelé si workflow mohou prohlížet, není však možné ho instanciovat. Pokud se o to přesto pokusíte, objeví se hlášení chyby.

Pokud chcete nekonzistentní workflow ještě před uložením opravit, klepněte na tlačítko ZRUŠIT. Proved'te test správnosti workflow, který spustíte příkazem z menu ZOBRAZENÍ/KONTROLA SPRÁVNOSTI nebo pomocí

následující ikony z panelu nástrojů . Podle výsledku testu vyhledejte a opravte chyby. Poté aktualizujte test klepnutím na tlačítko OBNOVIT v okně SEZNAM CHYB.

Pokud byly opraveny všechny chyby, zobrazí se pod grafem workflow symbol zeleného světla a v okně poznámka... Výsledky testu : Workflow je konzistentní.

6.4.6 Tisk

Chcete-li si vytisknout graf workflow, detailní informace k workflow a integrovaných skriptech, klepněte na ikonu s tiskárnou  nebo vyberte v nabídce menu SOUBOR možnost TISK.

Zobrazí se standardní dialog, ve kterém můžete...

- vybrat tiskárnu, na které chcete tisknout
- určit počet kopií
- nastavit vlastnosti tisku

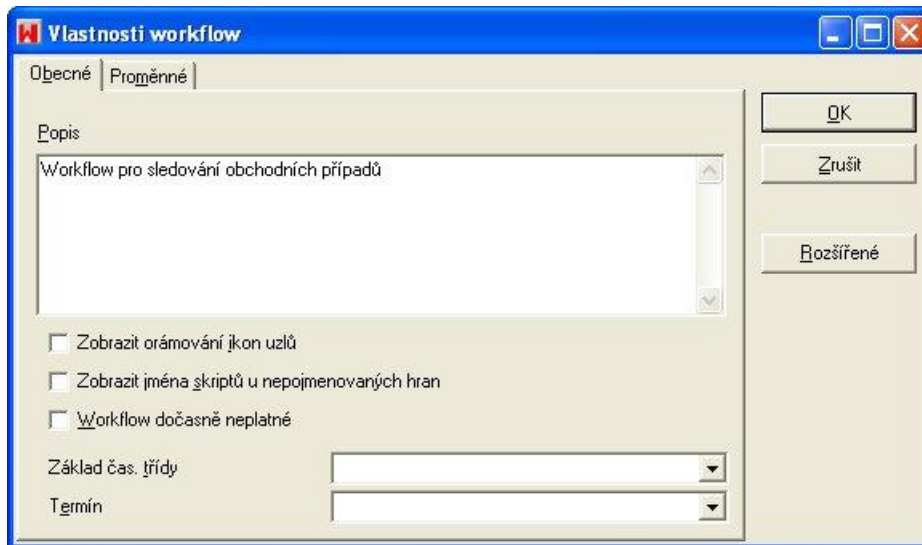
Dále můžete pomocí možnosti NASTAVENÍ TISKU z nabídky menu SOUBOR určit, co přesně budete tisknout...

- graf workflow
- informace týkající se workflow, jako například abecední seznam proměnných, uzlů a hran včetně jejich vlastností
- skripty, které se mohou na různých místech workflow vyskytovat

6.5 Vlastnosti workflow

Obecné vlastnosti workflow jsou speciální charakteristiky, které byly workflow jako celku přiřazeny. Přitom například nadefinujete vybrané proměnné, které mohou být po spuštění workflow aktivovány.

Vlastnosti workflow si můžete zobrazit pomocí klávesy <F12> nebo pomocí příkazu VLASTNOSTI WORKFLOW z nabídky menu EDITOVAT. Zobrazí se následující okno:



Obsahuje záložky **Obecné** a **Proměnné**. Tlačítkem **Rozšířené** zobrazíte dialog se záložkami Práva a Verze, který byl popsán v kapitolách Oprávnění na straně 48 a Verze workflow na straně 50.

Položky **Základ čas. třídy** a **Termín** slouží pro interní potřebu, doporučuje se je ponechat prázdné.

6.5.1 Obecné vlastnosti workflow

Jsou zadávány na záložce **Obecné** ve výše zobrazeném dialogu.

Popis workflow

Popis workflow v horní části dialogu slouží k informovanosti uživatele. Bez jakýchkoliv formálních údajů zde můžete popsat obsah workflow. Kromě názvu a grafu tak máte další možnost jednoznačně identifikovat workflow. Tato možnost se využívá například tehdy, pokud workflow vytváří více osob.

Grafická nastavení workflow

Orámování ikon uzlů

Zaškrtnutím této možnosti **Zobrazit orámování ikon uzlů** ovlivníte vzhled grafu workflow – ikony uzlů budou mít modrý rámeček.

Zobrazení jména skriptu u nepojmenovaných hran

Pokud hrana nemá jméno a obsahuje skript, pak zaškrtnutím **Zobrazit jména skriptů u nepojmenovaných hran** se jméno skriptu zobrazí u hrany.

Instancie workflow

Pod popisem workflow najdete možnost **Workflow dočasně neplatné**. Pokud tuto možnost zaškrtnete, nemůže být workflow v aplikaci M/TeamBridge využíváno. Workflow není aktivní.

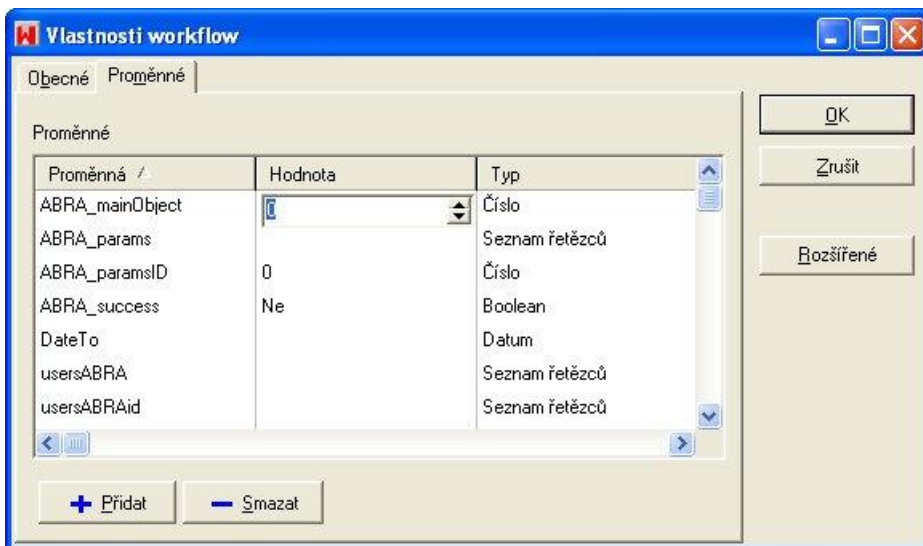
Tato možnost se využívá například v případech, kdy workflow ještě není dokončeno nebo není-li některý z účastníků workflow dlouhodobě přítomen.

Kdykoliv můžete workflow znovu aktivovat odškrtnutím této možnosti.

6.5.2 Proměnné ve workflow

K proměnným procesu můžete přistupovat ze skriptu, z příloh (makro v MSWORD) nebo přes OLE rozhraní aplikace. Ve skriptu mají tyto proměnné před jménem předponu <wkf.>, tj. tzv. globální proměnné. Lokální proměnné jsou bez předpony, můžete si je nadefinovat přímo ve skriptu.

Na záložce PROMĚNNÉ dialogu VLASTNOSTI WORKFLOW můžete přiřazovat workflow konkrétní proměnné a počáteční hodnoty.




Název a typ proměnné ve skriptu a definici proměnné musí přesně souhlasit. V definici skriptu si můžete ověřit, které proměnné se ve skriptu vyskytují.

Poznámka:

V rámci jednoho workflow se může proměnná vyskytovat několikrát. Při instanci stávajícího workflow nebo při jeho opakovaném spuštění jsou vícečetné proměnné ignorovány.

Přidávání proměnných

Chcete-li přidat novou proměnnou do seznamu, klepněte nejdříve na tlačítko . Otevře se okno, které zobrazuje seznam proměnných a jejich typ ze Slovníku dat řazených v abecedním pořadí (viz Slovník dat pro definici proměnných na straně 79).



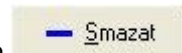
Vyberte z nabídky proměnnou (více proměnných vyberete pomocí klávesy <Ctrl>)

Hodnotu ve druhém sloupci můžete zadat jako počáteční hodnotu proměnné ve workflow.

Tabulka typů a hodnot pro definici proměnných		
Typ	Popis	Standardní hodnota

Řetězec	string, řetězec znaků	
Číslo	integer, celé číslo	0
Boolean	log. hodnota Ano/Ne	Ne
Datum	date	
Měna	currency	
Seznam	list všechny typy	

Pokud chcete nějakou proměnnou ze seznamu smazat, označte ji a použijte tlačítko



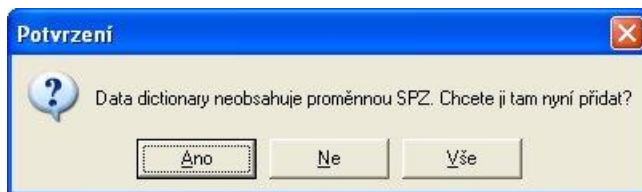
Se sloupci seznamu proměnných můžete pracovat obvyklým způsobem, tj. upravovat šířku sloupců, klepnutím na záhlaví sloupce seřadit hodnoty vzestupně a opakovaným klepnutím sestupně.

Nastavení proměnných

Hodnoty proměnných lze nastavit ručně ve sloupci HODNOTA. Lze také zvolit, zda se mají přepsat hodnoty existujících proměnných při startu workflow (zaškrťovací pole pod seznamem proměnných), tj. pokud už má proměnná nějakou hodnotu z předešlého workflow nebo nastavenou ručně, zda se má přepsat nebo ne.

Aktualizace slovníku dat

Pokud naimportujete workflow s proměnnými, které dosud nejsou ve slovníku dat, pak po zobrazení VLASTNOSTI WORKFLOW se objeví upozornění, že proměnná ještě nebyla nadefinována.



Nyní je možné:

- rozhodnout se pro převzetí proměnné do slovníku dat
- zamítnout převzetí
- převzít vše bez dalšího dotazování

Následně se zobrazí dialog VLASTNOSTI WORKFLOW, proměnné jsou převzaty do seznamu proměnných a zároveň do slovníku dat. Před aktivací workflow je možné u nich zadat ještě počáteční hodnoty. Od té chvíle jsou Vám tyto proměnné k dispozici.

6.6 Vlastnosti komponent workflow

Dosud jsme vysvětlovali pouze strukturu workflow. K tomu je nutné doplnit konkrétní obsah jednotlivých komponent, jinými slovy je nutné přiřadit uzlům a hranám vlastnosti.

Musíte například určit, kdo bude který pracovní krok vyřizovat, jak přesně budou jednotlivé kroky vypadat, lhůty dokončení kroků atd.

6.6.1 Vlastnosti pracovního kroku

Editor vlastností	
Node1	
Jméno	
Popis	
Nápověda	
Ikona	WORKSTEP
Vstup. script	
Uživatel	
Prov. script	
Provedení	Client
Prov. maska	
Externí operace	
Max. doba	
Eskalace	
Aktivace	
Podpis	Ne
Přílohy	

Pro přidělení konkrétních vlastností nejdříve vyberte požadovaný pracovní krok.

Editor vlastností si zobrazíte označením pracovního kroku a...

- klávesou <F11> nebo
- pomocí položky menu EDITOVAT/VLASTNOSTI

Název uzlu

Nahoře v prvním řádku vidíte název uzlu. Uzly jsou automaticky číslovány (Node1, Node2...). Jedná se o označení, které program interně přiděluje podle pořadí, ve kterém byly jednotlivé pracovní kroky vytvořeny.

Tyto názvy není možné měnit, pouze prohlížet. Klepněte na šipku na konci pole s názvem pracovního kroku a zobrazí se seznam všech dosud použitých názvů uzlů (Node...) a hran (Flow...). Pokud označíte některý z použitých názvů, bude automaticky označen příslušný objekt v grafu workflow.

Jméno

Je možné zvolit libovolné jméno pracovního kroku, neexistují žádné podmínky pro jeho formu. Berte ale v úvahu, že název bude zobrazen jako popisek daného uzlu v grafu workflow, proto se nedoporučuje vybírat příliš dlouhé názvy.

Pomocí nabídky menu NÁSTROJE/NASTAVENÍ na záložce Nastavení **Skrýt texty** můžete zaškrtnutím nastavit, zda se mají popisky zobrazovat v grafu workflow nebo ne. Ve stejném dialogu také nastavíte, jestli se mají popisky u hran umísťovat v textboxu jako u uzlů **Používat textbox pro hrany**.

Popis

Stručný popis pracovního kroku slouží pro informaci při vytváření workflow. Uživatel workflow v M/TeamBridge není k dispozici. Tato možnost se využívá například pokud workflow vytváří více osob nebo pro popis od zadavatele.

Nápověda

Obsáhlejší popis pracovního kroku slouží pro informaci při vytváření workflow. Uživatel workflow v M/TeamBridge není k dispozici.

Zde můžete popsat činnost pracovního kroku. Tato možnost se využívá též pokud workflow vytváří více osob nebo pro popis od zadavatele. Nelze psát přímo do pole vlastnosti, po klepnutí do tohoto pole a stisku se otevře okno pro zápis textu...



Ikona

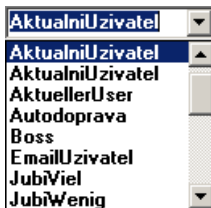
Vzhled pracovního kroku v grafu workflow můžete změnit výběrem ze seznamu dostupných ikon




Chcete-li vidět i náhled ikony, stiskněte  na konci pole. Zobrazí se Knihovna ikon viz strana 82.

Vstupní skript

Nejdříve je nutné pomocí skriptu vybrat osoby, které budou pracovní krok provádět nebo o něm mají být informovány. Když klepnete v EDITORU VLASTNOSTÍ do pole vstupní skript a dále na šipku, zobrazí se seznam skriptů pro výběr účastníka:



Jedná se o skripty s návratovou hodnotou **UserCollection**. Vpravo se nachází ještě další dvě ikony. Definici vybraného skriptu si můžete prohlédnout nebo upravit klepnutím na toto tlačítko .

Druhé tlačítko  umožňuje vybrat skript ze seznamu skriptů a prohlédnout si jeho definici. V tomto dialogu nelze skripty upravovat.

Výsledkem skriptu je přidělení hodnoty **UserCollection** pracovnímu kroku, což znamená výběr konkrétního účastníka workflow, který bude provádět pracovní krok. Pokud se výběr nepovede (žádný účastník nevyhověl podmínkám nastaveným ve skriptu) je zobrazeno upozornění a workflow je přerušeno. Tuto situaci je vhodné ošetřit např. výběrem aktuálního uživatele v případě nenalezení jiného vhodného uživatele.

Uživatel

Místo vstupního skriptu je možno určit uživatele pracovního kroku přímo výběrem ze seznamu účastníků definovaných v M/USER.

Upozornění:



Pozor na ztrátu univerzálnosti workflow při personálních změnách uživatelů workflow.

Je možné použít pouze jednu z obou možností. Pokud byl již nastaven vstupní skript a provedete výběr uživatele, název vstupního skriptu se vymaže a naopak.

Prováděcí skript

Dostáváme se k vlastní činnosti, která má být v rámci pracovního kroku vykonána.

Pro výběr prováděcího skriptu máte k dispozici seznam skriptů s návratovou hodnotou *boolean*. Pokud klepnete na šipku na konci pole v kolonce PROVÁDĚCÍ SKRIPT, můžete vybrat skript podle názvu. Následně si definici

vybraného skriptu můžete prohlédnout nebo i upravit klepnutím na toto tlačítko . Druhé tlačítko  umožňuje vybrat skript ze seznamu skriptů a prohlédnout si jeho definici. V tomto dialogu nelze skripty upravovat.


```

script JubizweigGS: boolean
begin
If wkf.JubizJahre >10 then
    result:= True
else
    result:= False
end
end
end

```

Prováděcí skripty se vztahují přímo k výsledkům workflow. Pokud je například návratová proměnná result ve skriptu nastavena na true, je také celkový výsledek pracovního kroku true.

Provedení

Tato vlastnost má význam při použití 3-vrstvé verze M/User. Lze určit, zda pracovní krok by se měl provádět na straně serveru nebo klienta.  Obvyklé je ponechat první možnost **Client**.


Prováděcí maska

Místo prováděcího skriptu je možné pracovnímu kroku přiřadit pouze masku. Všechny činnosti týkající se masek jsou popsány v samostatné příručce Návrhář masek M/Teambridge.



V pracovním kroku může být použita buď jedna nebo druhá vlastnost (**Prováděcí skript** nebo **Prováděcí maska**).

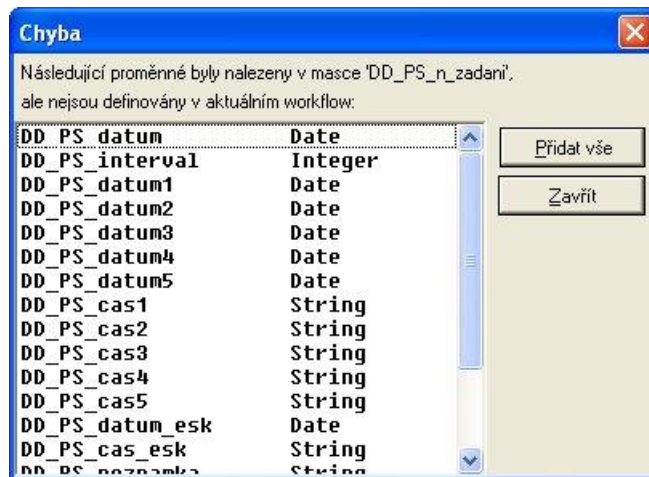
Klepnutím do pole vlastnosti **Prováděcí maska** se zobrazí ikony v pravé části pole.


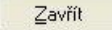



- Klepnutím na ikonu  se zobrazí seznam dostupných masek, ze kterých vyberete požadovanou masku...



- Po potvrzení tlačítkem  se název masky vloží do pole vlastnosti.
- Pomocí ikony  můžete masku otestovat na přítomnost proměnných nedefinovaných v masce ve workflow. Pokud některá z proměnných masky dosud není ve workflow začleněna, objeví se upozornění...

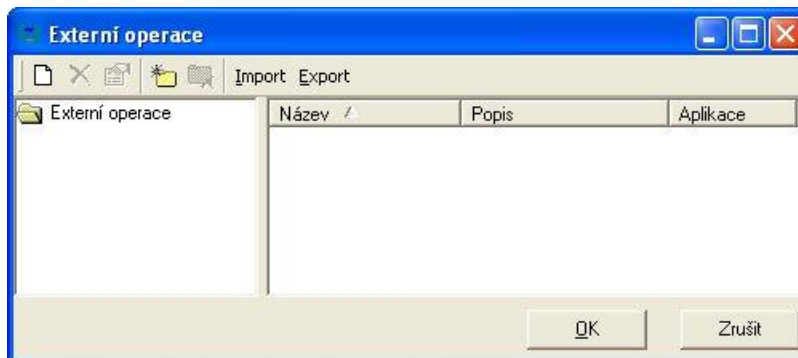


- Pak můžete tlačítkem  začlenit proměnné do workflow nebo tlačítkem  proměnné do workflow nepřidávat a vyřešit situaci např. vybráním jiné masky nebo...
- Ikonou  otevřete Návrhář masek a v něm vybranou masku. Můžete nyní masku editovat, vybrat jiné proměnné atd. Můžete také vytvořit novou masku a po jejím uložení ji vybrat do pole vlastnosti.

Externí operace


Pracovní krok by mohl vyžadovat takzvané externí operace, tj. činnosti, které jsou vykonávány ve speciální aplikaci. M/Teambridge poskytuje speciální aplikace s nezbytnými daty a prostředky.

Všechny informace, které speciální aplikace vyžaduje k vykonání pracovního kroku jsou zadávány v dialogu...



Administrace externích operací je prováděna v levé části okna ve stromu složek a v pravé části okna v seznamu aktivit uložených ve vyznačené složce.

Maximální doba trvání

Zde máte možnost nastavit určitý časový limit pro vyřízení pracovního kroku. Klepnutím na toto tlačítko  si zobrazíte následující dialog:

Dobu lze určit...


- jako časový interval. Pomocí šipek u polí DNY, HODINY a MINUTY můžete nastavit přesnou lhůtu provedení pracovního kroku. Do všech tří polí lze zadat údaj i ručně.
- nadefinovanou časovou třídou. Pomocí šipky na konci pole zvolíte příslušnou časovou třídu, kterou lze nadefinovat v menu NASTAVENÍ/ADMINISTRACE/TERMÍNY (kapitola Vytvoření nové časové třídy na straně 80)
- proměnnou, která obsahuje příslušný časový údaj

Dále je ještě nutné určit počáteční událost. Vybíráte ze dvou možností

- Krok přidělen
- Krok dokončen

Podle toho se pak řídí výběr uzlu, pro první možnost se obvykle volí krok, ve kterém je nastavení prováděno, pro druhou možnost předcházející krok.

U účastníka procesu se nastavení max. doby projeví v okně procesu jako upozornění:

 Zpracujte prosím prac. krok 'krok2' do 29.08.2008 16:07

Eskalace


Pokud je nastavena maximální doba, lze nastavit eskalaci, tzn. v případě, že krok není v zadané době zpracován, provést potřebné činnosti prostřednictvím eskalačních skriptů, které jsou typu Void. Eskalační úrovně pracovních kroků spravuje notifikační server, který spouští eskalační opatření jakmile je dosaženo požadovaného času.

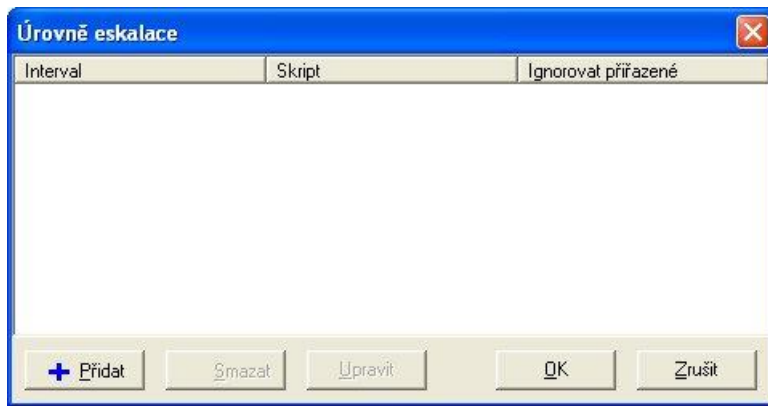
Příklady:

- předání pracovního kroku jinému účastníku
- spuštění externích programů
- modifikace globálních proměnných
- zapsání textu do procesu

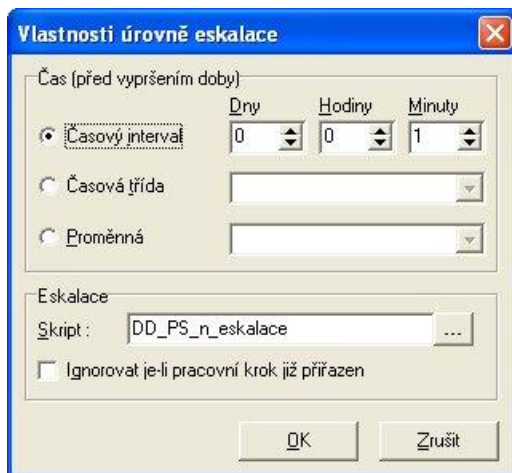
Zvláštní případ:

Může se stát, že pracovní krok, pro který bylo eskalační opatření aktivováno, nebude proveden. V tom případě bude mít eskalační opatření přednost při dalším průchodu.

Klepnutím do pole vlastnosti a stiskem  vyvoláte okno...



Tlačítkem  se otevře dialog...





Zde je nutno zadat čas před vypršením zadané max. doby kdy se má eskalace spustit.

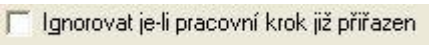
Upozornění:

Pokud má eskalace nastat až v okamžiku nastavené max. doby, doporučuje se zadat alespoň 1 minutu.

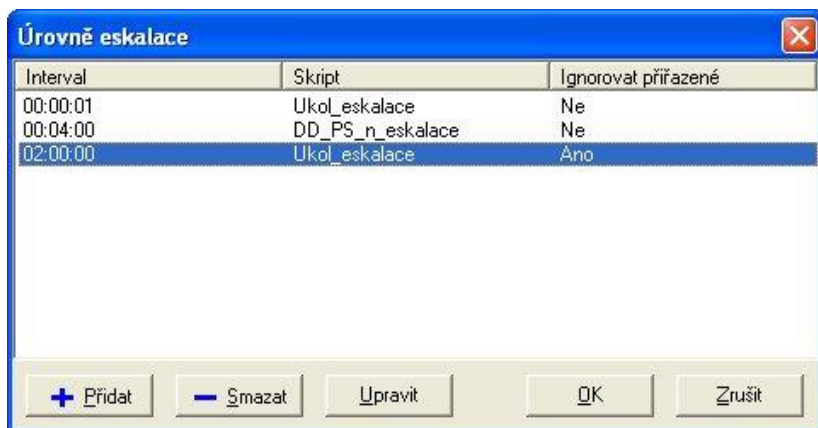
K zadání času jsou k dispozici tři možnosti stejně jako u vlastnosti Maximální doba trvání na straně 62.

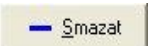

Ve spodní části dialogu je k dispozici pole pro výběr eskalačního skriptu. Stiskem  se otevře dialog pro výběr skriptu. Jsou zobrazeny pouze skripty typu Void (viz příručky M/User a Administrace M/Teambridge)

Vyberte požadovaný skript a potvrďte tlačítkem  vlevo dole.

Následuje zaškrtnuté pole , kterým potvrdíte, zda chcete, aby se eskalace neprováděla v případě přiřazení pracovního kroku jinému účastníkovi.


Můžete nastavit více úrovní eskalace pro pracovní krok...



Tlačítkem  můžete vybranou úroveň eskalace zrušit. Tlačítkem  vyvoláte opět okno Vlastnosti úrovně eskalace a můžete provést úpravy.

Aktivace

Zde můžete pro pracovní krok nastavit datum, kdy se má proces pro zpracovávajícího účastníka objevit v seznamu procesů. To je umožněno nastavením výchozího filtru v M/Teambridge (viz Příručka pro uživatele M/TeamBridge). Filtr je nastaven tak, aby nezobrazoval procesy, u kterých dosud nenastalo datum aktivace.

Klepnutím do pole vlastnosti a stiskem  se objeví okno...




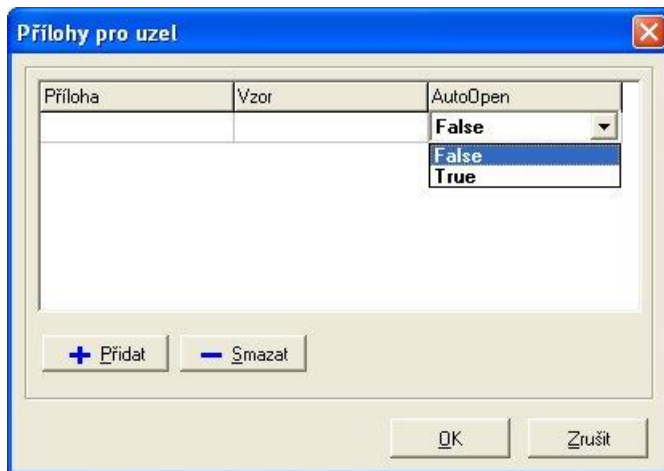
- Klepnutím na šipku rozbalíte na prvním řádku časové třídy. Zvolíte příslušnou časovou třídu, kterou lze nadefinovat v menu nastavení/administrace/termíny (kapitola Vytvoření nové časové třídy na straně 80).
- Na řádku Základ aktivace vyberete pomocí šipky datumovou proměnnou, která bude v době provádění workflow obsahovat údaj datumu, od kterého se bude počítat údaj z časové třídy. Pokud základ aktivace nezvolíte, bude se uvažovat datum startu workflow.
- Komentář bude obsažen v proměnné, kterou vyberete ze seznamu proměnných typu string.

Podpis

Standardně není podpis pracovních kroků nutný. Můžete však nastavit, aby byl podpis u některých konkrétních kroků požadován. Uživatel, který daný krok provádí, bude vyzván, aby pracovní krok podepsal prostřednictvím zadání určitého hesla.




Přílohy

K jednotlivým uzlům lze přidávat informace formou příloh. Použijte následující tlačítko , které se zobrazí po klepnutí do pole v kolonce přílohy. Zobrazí se následující dialog:

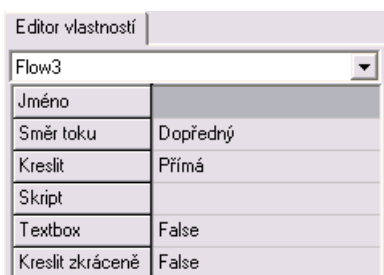


Přidání přílohy k pracovnímu kroku

Klepněte na tlačítko  a postupujte následujícím způsobem...

- zadejte do pole PŘÍLOHA název nově generovaného dokumentu. V rámci daného workflow musí být název jednoznačný.
- vyberte šablonu dokumentu. Seznam šablon si zobrazíte klepnutím na tlačítko  v poli Vzor.
- Označte vybranou šablonu a výběr potvrďte tlačítkem .
- nastavte, zda se má příloha automaticky otevřít. Standardně je nastavena hodnota *false*, příloha se tedy automaticky neotevře.
- dialog potvrďte tlačítkem  a příloha bude k pracovnímu kroku přidána.

6.6.2 Vlastnosti hrany



Stejně jako při definování vlastností pracovních kroků máte i pro vlastnosti hran k dispozici EDITOR VLASTNOSTÍ:

Název hrany

Nahoře v prvním řádku vidíte název hrany. Hrany jsou automaticky číslovány (Flow0, Flow1...). Jedná se o označení, které program interně přiděluje podle pořadí, ve kterém byly jednotlivé hrany vytvořeny.

Tyto názvy není možné měnit, pouze prohlížet. Klepněte na šipku na konci pole s názvem hrany a zobrazí se seznam všech dosud použitých názvů uzlů (Node...) a hran (Flow...). Pokud označíte některý z použitých názvů, bude automaticky označen příslušný objekt v grafu workflow.

Jméno

Můžete zvolit libovolný název hrany. Kvůli přehlednosti doporučujeme vybrat takový název, který naznačuje funkci této hrany. Například název „Spokojen?“ prozrazuje, že v dalším kroku očekáváme odpověď nebo rozhodnutí.

Pomocí nabídky menu NÁSTROJE/NASTAVENÍ na kartě NASTAVENÍ/SKRÝT TEXTY můžete zaškrtnutím nastavit, zda se mají popisky zobrazovat v grafu workflow nebo ne. Ve stejném dialogu také nastavíte, jestli mají být popisky užívány i u hran.

Směr toku hrany

U každé hrany je nutné nastavit její směr toku. Máte na výběr ze dvou možností...



Šipka hrany v grafu workflow udává její směr. Pro větší rozlišení je hrot šipky hrany směřující dopředu černý a hrot zpětné hrany bílý.

Směr hrany můžete kdykoliv obrátit v dialogu EDITOR VLASTNOSTÍ nebo v nabídce popup menu...



Zobrazení hrany (Kreslit)

Tvar hrany je libovolně nastavitelný. Standardně je nastaveno, aby hrana spojovala dva uzly přímo.

Ze seznamu si ale můžete vybrat jiný tvar...



Tvary můžete dále individuálně upravovat. Pokud na některou hranu klepnete, označí se několika černými body. V místě bodu je možné tvar hrany upravovat a posouvat je libovolným směrem.

Skript

Tento bod Vám nabízí výběr skriptu s návratovou hodnotou boolean stejně jako u vlastností PROVÁDĚCÍ SKRIPT. Pokud klepnete na šipku na konci pole v kolonce, můžete vybrat skript podle názvu. Následně si definici vybraného skriptu můžete prohlédnout nebo zároveň upravit klepnutím na toto tlačítko

Druhé tlačítko umožňuje vybrat skript ze seznamu skriptů a prohlédnout si jeho definici. V tomto dialogu nelze skripty upravovat.

Popiska hrany (Textbox)

Zde nastavíte, jak se mají zobrazovat popisky hran – textem na hraně (*false*) nebo v bublině (*true*). V druhé kolonce tohoto řádku vidíte standardně nastavenou hodnotu *true* nebo *false*. Nastavení můžete pro každou konkrétní hranu změnit.

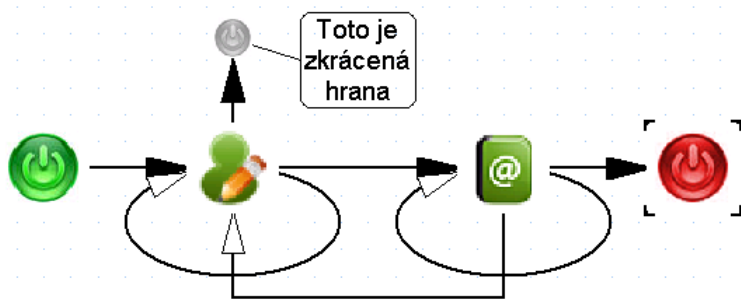
Standardní hodnotu lze změnit v dialogu NÁSTROJE/NASTAVENÍ na kartě Nastavení



při vytvoření workflow pro všechny hrany najednou. Název hrany se pak zobrazí v textboxu.

Kreslit zkráceně


Používá se tehdy, pokud nějaká hrana vede přes velkou část grafu workflow (obvykle u větších workflow hrany vedoucí z některého z prvních uzlů do koncového uzlu) a workflow ztrácí na přehlednosti. Při této volbě není hrana vykreslena až do uzlu, do kterého vede, ale je zkrácena a na jejím konci je "odkaz" na uzel, do kterého vede.



6.6.3 Vlastnosti vnořené workflow

Vnořené workflow je již dříve nadefinované workflow, které je možné připojit k jinému workflow.



V grafu workflow má následující symbol: . Aby bylo vnořené workflow funkční, je nutné nastavit jeho vlastnosti v EDITORU VLASTNOSTÍ.



Vnořené workflow se bude provádět v rámci celého workflow, aniž by bylo v grafu zobrazeno. Můžete si ho prohlédnout pomocí možnosti ZOBRAZIT z popup menu vnořené workflow.

Název vnořené workflow

Vnořené workflow má přiděleno interní číslo, je tedy označeno stejně jako hrany a uzly.

Jméno

Dále můžete samozřejmě zadat libovolné jméno. Pokud jméno nezadáte, bude doplněno podle názvu vkládaného workflow.

Popis

Stručný popis vnořené workflow slouží pro informaci při vytváření workflow. Uživatelé workflow v M/TeamBridge není k dispozici. Tato možnost se využívá například pokud workflow vytváří více osob nebo pro popis od zadavatele.



Nápopvěda

Obsáhlejší popis vnořené workflow slouží pro informaci při vytváření workflow. Uživatelé workflow v M/TeamBridge není k dispozici.

Ikona

Vzhled uzlu s vnořeným workflow v grafu workflow můžete změnit výběrem ze seznamu dostupných ikon (viz kapitola Vlastnosti pracovního kroku na straně 60).

Workflow

Nakonec vyberte workflow z nabídky, která se zobrazí po klepnutí na tlačítko s tečkami . Zobrazí se dialog OTEVŘÍT WORKFLOW, ve kterém workflow vyberete a potvrdíte tlačítkem .

Popup menu vnořeného workflow

Pokud klepnete na ikonu vnořeného workflow pravým tlačítkem myši, máte k dispozici následující popup menu...

Smazat	Del
Zobrazit	
Rozpustit	
Zrušit	Esc
Zavřít	

- ZOBRAZIT. Zobrazí graf vnořeného workflow na novou stranu
- ROZPUSTIT. Zobrazí graf vnořeného workflow přímo vložený v grafu celého workflow.
Upozornění: Počáteční a koncový uzel se rozpustí jako běžný pracovní krok bez nastavených vlastností.
- ZAVŘÍT. Zavře celé workflow. Před uzavřením se zobrazí kontrolní dialog, zda si přejete uložit provedené změny.



6.7 Administrace skriptů

Skripty jsou prováděcí moduly workflow. Jsou zapsány v Jazyce skriptů, spouštějí se v rámci pracovního kroku nebo hrany. Nabývají návratové hodnoty, podle níž se řídí další průběh workflow.

Ke skriptům získáte přístup několika způsoby:

- z menu nástroje/administrace/seznam skriptů
- z vlastnosti pracovního kroku Vstupní skript (Usercollection)
- z vlastnosti pracovního kroku Prováděcí skript a hrany Skript (Boolean)
- z vlastnosti pracovního kroku Eskalace (Void)
- v modulu M/User Serie M/...

V první a poslední možnosti se zobrazí nejdříve Seznam skriptů, ze kterého lze teprve vyvolat editor skriptu (viz kapitola Seznam skriptů na straně 82).

V ostatních případech přístupu ke skriptu přes vlastnosti máte kromě možnosti zobrazení seznamu ikonou  také možnost přímo vyvolat editor skriptu ikonou .

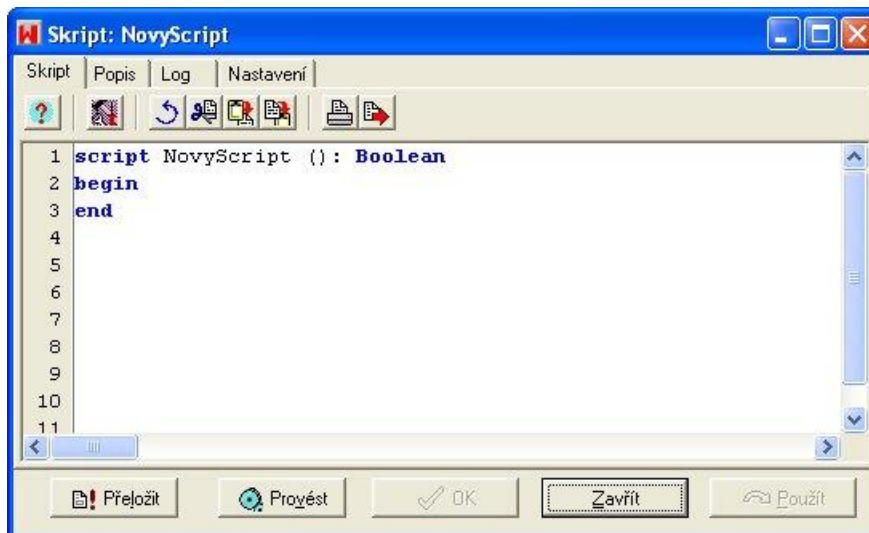
6.7.1 Vytvoření nového skriptu

V Seznamu skriptů vyberte umístění nového skriptu, případně vytvořte novou složku pro uložení skriptu.

V popup menu složky vyberte možnost Nový skript. V okně editoru skriptu se Vám zobrazí základní struktura skriptu, kterou rozpracujete do konečné podoby skriptu (s využitím práce se schránkou pro kopírování textů z jiných skriptů ap.)

Pojmenování skriptu

Skript pojmenujete nahrazením jména NovyScript v základní struktuře skriptu. Při nejbližším uložení skriptu bude toto jméno akceptováno a objeví se ve stromové struktuře skriptů ve složce, kterou jste vybrali pro uložení skriptu.



Skript nelze klasickým způsobem přejmenovat ve stromové struktuře skriptů (popup menu, klepnutí na jméno), musí být vždy přejmenován v otevřeném okně editoru skriptů změnou jména v parametru script.

Typ skriptu


V editoru skriptů se standardně zobrazí typ (návrátová hodnota) skriptu jako **Boolean**. Pokud vytváříte vstupní skript pracovního kroku, musíte jej změnit na typ **Usercollection**. Pro eskalační skripty musí být uveden typ **Void**.

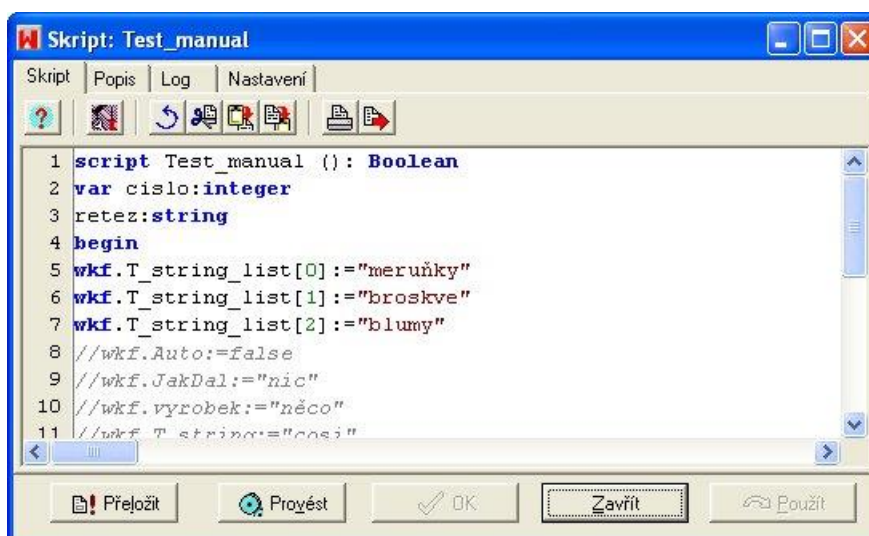
Pro další práci se skriptem můžete použít všechny postupy, které jsou popsány v následující kapitole o editaci skriptu.

6.7.2 Editace skriptu

Editor skriptu otevřete v Seznamu skriptů

- poklepaním na jméno skriptu nebo
- v popup menu skriptu vyberete Editovat skript

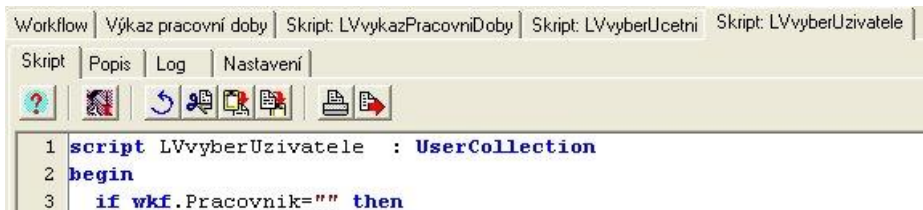
V otevřené workflow v Editoru vlastností pracovního kroku nebo hrany otevřete editor stiskem ikony  u příslušné vlastnosti (Vstupní, Prováděcí skript)



Nyní lze provádět ve skriptu změny při dodržení zásad Jazyka skriptů.

Ukotvení okna editoru skriptu


Pro přehlednější práci s okny skriptů si okno můžete ukotvit v okně pro graf workflow. Tažením okna za titulkový pruh směrem vlevo nahoru se okno v určitém okamžiku rozšíří na rozměr okna pro graf a v horní části vznikne záložka s názvem skriptu.

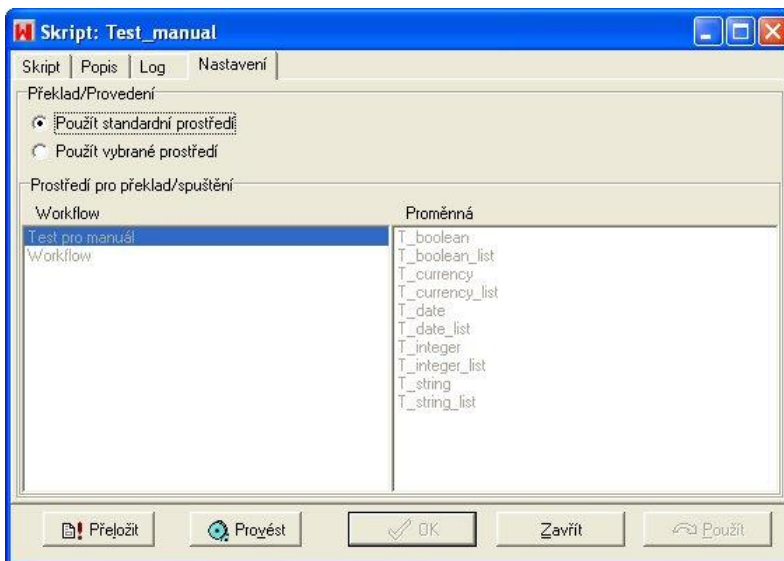


Takto si můžete otevřít více skriptů i workflow najednou a pak mezi nimi přepínat. Takto můžete ukotvit i skripty otevřené přes Seznam skriptů do editačního režimu (poklepáním na název).

Záložky okna editoru skriptu

Kromě hlavní záložky pro skript jsou k dispozici další záložky:

- **Popis** s již vyplněným názvem a typem (návratovou hodnotou) skriptu, lze doplnit popis skriptu
- **Log** pro zápis průběhu činnosti
- **Nastavení**, kde můžete při volbě vybraného prostředí vyzkoušet skript v jiném workflow, které vyberete ze seznamu otevřených workflow v levé části okna. Napravo jsou pak zobrazeny globální proměnné vybraného workflow. Po stisku klávesy  **Přeložit** zjistíte, zda skript bude funkční i v tomto prostředí, nebo zda je nutné nadefinovat chybějící proměnné, masky atd.



Předdefinované vzory kódu

Jako pomůcku při zapisování kódu skriptu můžete využít kombinaci kláves <Ctrl><J>, která zobrazí výběr z těchto textů:

```

boolean
declare procedure dll " " ()
foreach user in do end
foreach user do end
getuserbyid
getuserlistbydepartment
getuserlistbytemplate
getusersbyattribute
if then end
if then else end
integer
result
string
usercollection
write()

```

Výběrem ze seznamu se text přeneso do skriptu, kde jej upravíte nebo doplníte.

Značky ve skriptu

Značky (bookmarks) slouží k orientaci v textu skriptu při tvorbě rozsáhlejších skriptů.

6.7.2.1.1.1 Vytvoření značky

Značky mohou být přidány do textu skriptu a následně lze podle nich v textu vyhledávat.

```

1 script Test_manual (): Boolean
2 var cislo:integer
3 retez:string
4 begin
5 wkf.T_string_list[0] := "meruňky"
6 wkf.T_string_list[1] := "broskve"
7 wkf.T_string_list[2] := "blumy"

```

Značka se vytvoří kombinací kláves <Ctrl K 0 až 9>. Nepoužívejte číslice na numerické klávesnici. Při použití kláves s číslicemi z alfanumerické klávesnice není nutné použít klávesu <Shift>.

6.7.2.1.1.2 Smazání značky

Značka se smaže opakovaným použitím stejné kombinace kláves jako k vytvoření značky.

6.7.2.1.1.3 Hledání podle značek

Podle značek se vyhledává kombinací kláves <Ctrl A 0 až 9>. Pro použití kláves s číslicemi platí stejné zásady jako u vytvoření značky.

Poznámka:

Aby byly značky viditelné, je třeba mít zapnutou lištu (viz kapitola Panel nástrojů editoru skriptu na straně 73).

Výběr proměnných

Další pomůckou je výběr globálních proměnných ze Slovníku dat. Při zapsání předpony <wkf.> se zobrazí seznam proměnných, který se zápisem dalších znaků zpřesňuje.

T_boolean	boolean
T_boolean_list	booleanList
T_currency	currency
T_currency_list	currencyList
T_date	date

Klepnutím na vybraný název proměnné se tento přeneso do skriptu ve správném tvaru jak je ve Slovníku dat (tj. můžete např. psát malými písmeny začátek názvu s velkými písmeny a po výběru se Váš zápis přemaže správným tvarem).

Vyhledávání v textu skriptu

Vyhledávací dialog vyvoláte z popup menu v otevřeném editoru skriptu volbou první položky **Najít** nebo kombinací kláves <Ctrl F>

Panel nástrojů editoru skriptu

6.7.2.1.1.4 Nápověda



Tato ikona zobrazí stručnou nápovědu:

```
Key shortcuts for script editor

Find text :
    <CTRL><F> or "Find" item in popup menu

Find next :
    <F3>

Bookmarks :
    <CTRL><K>[1..9]   toggle bookmark 1 .. 9
    <CTRL><A>[1..9]   go to bookmark 1 .. 9

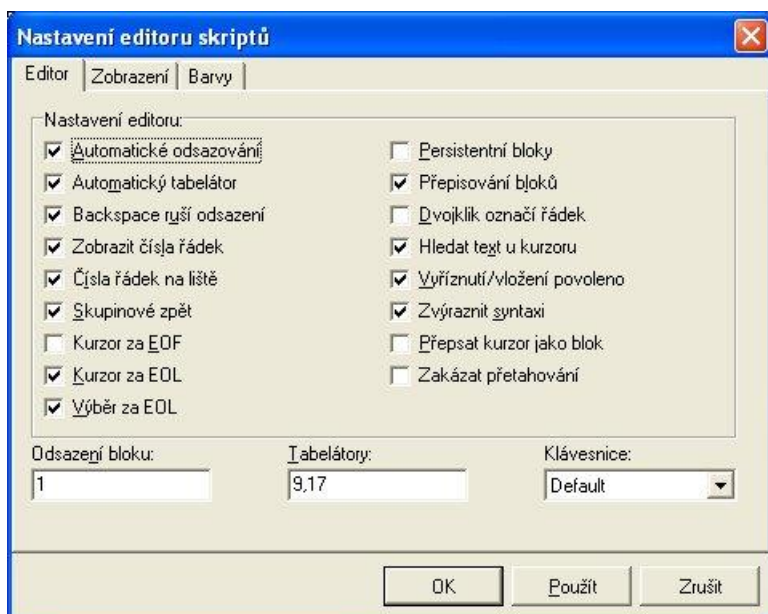
Code templates :
    <CTRL><J>   show code templates combo

Workflow variable hint:
    wkf<.>   show selection with dictionary variables
```

6.7.2.1.1.5 Nastavení




V dialogu máte celou řadu možností nastavení prostředí editoru skriptů na záložkách **Editor**, **Zobrazení** a **Barvy**:



Na záložce **Zobrazení** volíte viditelnost lišty na pravém okraji okna editoru skriptu a linie ohraničující okraj skriptu na pravé straně.

Na záložce **Barvy** volíte vzhled jednotlivých komponent textu skriptu.

6.7.2.1.1.6 Zpět

 Ikona pro vrácení činnosti o krok zpět.

6.7.2.1.1.7 Schránka


Pro práci se schránkou jsou použity netypické ikony

 Vyříznout


 Vlepit

 Kopírovat

6.7.2.1.1.8 Tisknout skript


 Po stisku ikony se objeví nejprve dialog pro nastavení tisku skriptu, ve kterém můžete stiskem tlačítka Setup vyvolat dialog pro nastavení tiskárny nebo tlačítkem OK spustit přímo tisk na standardní tiskárně.

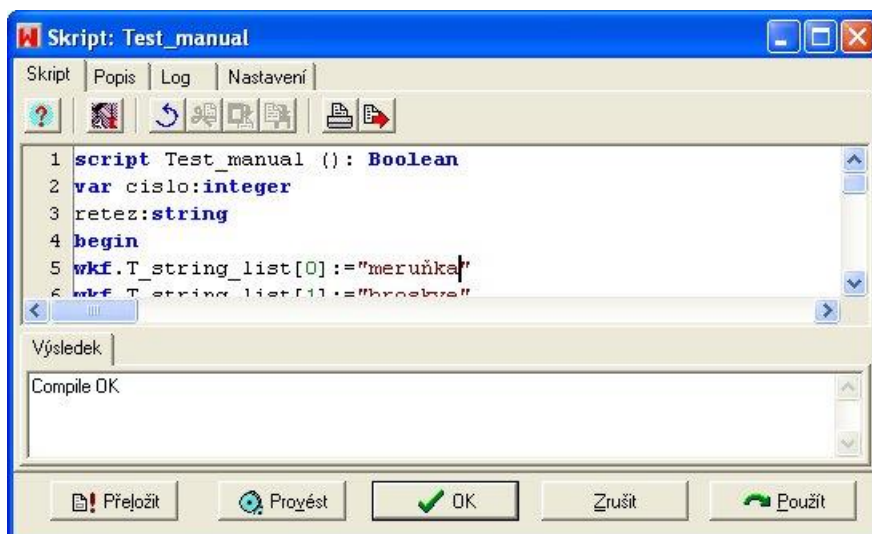
6.7.2.1.1.9 Uložení textu skriptu

 Pomocí standardního dialogu pro uložení souboru uložíte skript ve formě souboru **.txt** nebo **.rtf** nebo **.html** do vybrané složky na disku.

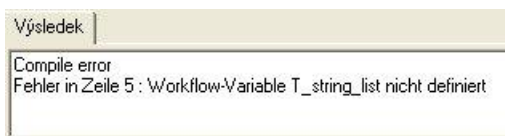
6.7.3 Vyhodnocení skriptu

Překlad skriptu

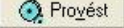
Pokud jste již ukončili zápis nebo opravu textu skriptu, můžete otestovat jeho formální správnost (tj. dodržení zásad Jazyka skriptů, existenci použitých proměnných, masek a funkcí dll knihoven). Stiskem tlačítka  provedete kontrolu, jejíž výsledek se objeví v oblasti výsledku, která se zobrazí ve spodní části okna.



Pokud byla zjištěna chyba ve skriptu, je popsána v poli výsledku:



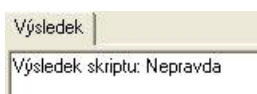
Test skriptu

Je-li překlad skriptu bez chyby, můžete otestovat jeho činnost tlačítkem .

Upozornění:

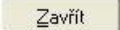
Nelze samozřejmě docílit stejných výsledků jako při zapojení skriptu do workflow (nenaplněné proměnné z předcházejících kroků atd.). Některé procedury a funkce z dll knihoven využívané ve skriptech nelze při tomto testu použít.




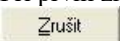
Otestovat lze vzhled masek, pokud jsou ve skriptu vyvolány, funkce na nich umístěných tlačítek a nakonec návratovou hodnotu skriptu.



Pokud nebylo před tímto testem docíleno formální správnosti skriptu, jsou i tyto chyby oznámeny.

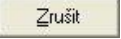
6.7.4 Ukládání skriptu

Než provedete ve skriptu první změnu, je přístupné pouze tlačítko , kterým skript beze změn uzavřete.

Při první změně v textu skriptu se zpřístupní tlačítka  a . Tlačítko  se změní na .

Uložení skriptu

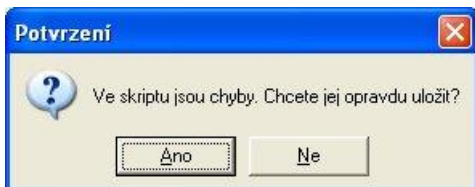
Tlačítko  uloží změny a zavře okno editoru,  uloží změny a nechá okno otevřené.

Po stisku tlačítka  je položen dotaz na uložení změn:



Uložení skriptu s chybami

Pokud skript neprošel překladem bez chyby a přesto má být uložen, je položen dotaz:

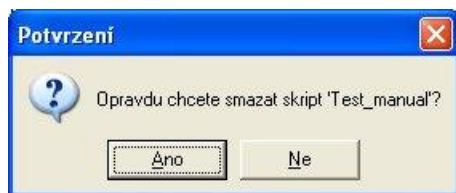


Při potvrzení dotazu tlačítkem **Ano** je skript uložen i s chybami, po stisku **Ne** zůstává okno editoru otevřeno.

6.7.5 Mazání skriptu

V Seznamu skriptů vyberte skript, který chcete smazat. V popup menu vyberte možnost Smazat skript...

Bude položen následující dotaz



6.7.6 Migrace skriptu


Skripty mohou být exportovány do několika typů souborů. Importovány mohou být pouze skripty ve formě **.xml** souborů a ve speciálním tvaru **.kws**.

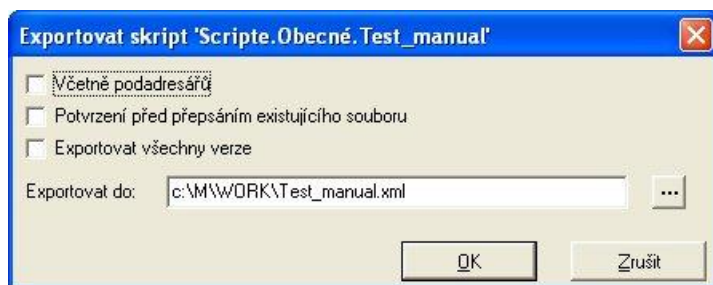
Export skriptu

Export z editoru skriptu

Skript je exportován jako text ve formě souboru **.txt**, **.rtf** nebo **.html**. Tyto soubory slouží pro dokumentační účely, nedají se zpětně importovat. Více viz kapitola Uložení textu skriptu na straně 74.

Export ze Seznamu skriptů

V popup menu vybraného skriptu v Seznamu skriptů zvolte možnost Exportovat skript... Stiskem tlačítka  otevřete standardní dialog pro vyhledání požadované složky na disku pro uložení exportovaného skriptu ve formě souboru **.xml**



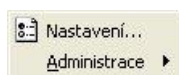
Tyto soubory lze použít pro import skriptů, např. pro přenos do jiné databáze.

Import skriptu

V popup menu složky v Seznamu skriptů zvolte možnost Importovat skript... Ve standardním dialogu vyhledejte soubor typu **.xml** nebo **.kws** a potvrďte tlačítkem Otevřít. Skript bude naimportován do vybrané složky Seznamu skriptů.

6.8 Nástroje

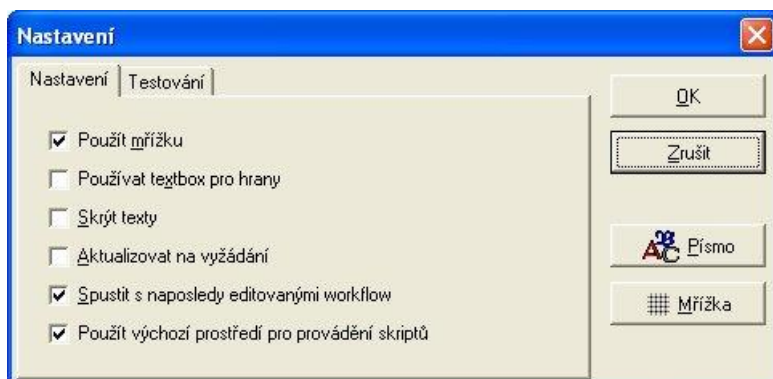
V menu Nástroje jsou k dispozici tyto volby...





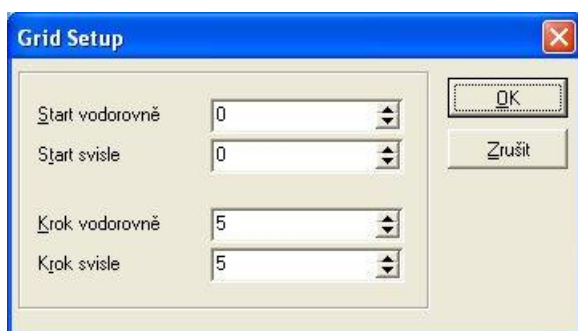
6.8.1 Nastavení

Záložka Nastavení

Na záložce Nastavení můžete nastavit tyto možnosti:



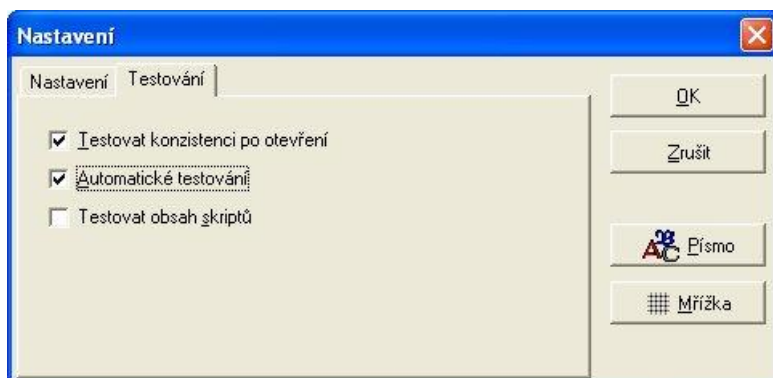
Tlačítkem  nastavíte písmo (standardní dialog), tlačítkem  vyvoláte okno...



kde si můžete individuálně nastavit rozměry mřížky.

Záložka Testování

Na záložce Testování lze nastavit...



6.8.2 Administrace

Slouží pro práci s objekty databáze M/TeamBridge



Vzory dokumentů

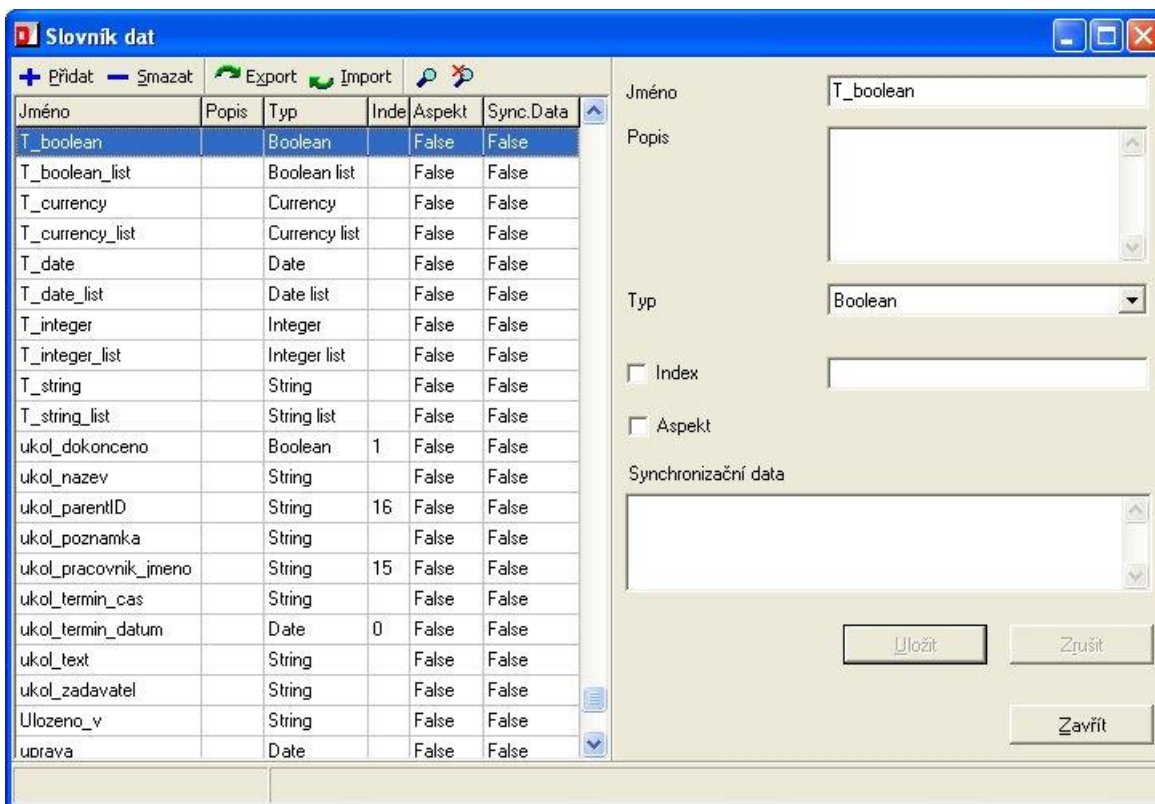
Administrace šablon dokumentů. Po vybrání této možnosti se zobrazí okno se složkami vzorů dokumentů.



Poklepáním na vybraný dokument zobrazíte informace o vlastnostech daného dokumentu.

Slovník dat pro definici proměnných

Slovník dat je databáze, ve které jsou spravovány veškeré proměnné. Pokud máte v aplikaci M/USER příslušná práva (atribut MTBAdminDD) pro přístup ke slovníku dat, můžete jej spustit z menu NÁSTROJE/ADMINISTRACE/DATOVÝ SLOVNÍK.



Vlevo je abecední seznam nadefinovaných proměnných spolu s popisem a typem, vpravo formulář pro vytvoření nových proměnných.

Chcete-li nadefinovat novou proměnnou, klepněte na tlačítko **+ Přidat** nad seznamem proměnných. V tabulce se přidá nový řádek nad naposledy označenou proměnnou a kurzor se objeví v poli JMÉNO.

Do následujícího pole doplňte popis, který slouží k jednodušší identifikaci proměnné.

Typ proměnné musíte vyplnit, máte na výběr následující možnosti (+ Date list a Currency list):



- Indexové proměnné

Typy proměnné s přídomkem list jsou proměnné s indexem, který se ve skriptu zapisuje za jméno proměnné jako celé číslo (integer) v hranatých závorkách (začíná od nuly). Tyto proměnné jsou vždy globální (nelze je definovat ve skriptu jako lokální).

```
//nastaveni voleb wkf
wkf.ukol_nastaveni[0] := "OutlookCalendar=ANO"
wkf.ukol_nastaveni[1] := "HTMLprehled=ANO"
```

- Definice databázových proměnných (kritéria)

Takto označené proměnné slouží k filtrování procesů nebo k jejich řazení.

Jejich použití je podmíněno nastaveným **unikátním** indexem, který Datový slovník sám doplňuje.

Další možnost využití je provázání různých procesů podle společné proměnné (např. Smlouva a faktura podle společného ID. Takovéto proměnné je možné skrýt pomocí volby **Skrytý**. Skryté proměnné jsou vidět pouze v rozhraní Datového slovníku nebo v náhledu proměnných procesu, pokud jsou k tomuto přiřazena uživateli práva.

V seznamu procesů v aplikaci M/TeamBridge se zobrazují databázové proměnné daného workflow v kolonce KRITÉRIA vedle kolonek ÚČASTNÍCI a PŘÍLOHY. Databázové proměnné mohou být využívány také jako vyhledávací kritéria.

- Definice aspektů


Pokud má proměnná sloužit jako argument vyhledávání v M/DMS, je třeba zaškrtnout aspekt v definici proměnné.

- Proměnné procesu s řídicími funkcemi

Existují proměnné, které mohou v M/TeamBridge přebírat určité řídicí funkce. Proměnné musí být nejdříve přidány ze slovníku dat do proměnných workflow. Tyto proměnné začínají obvykle MTB...

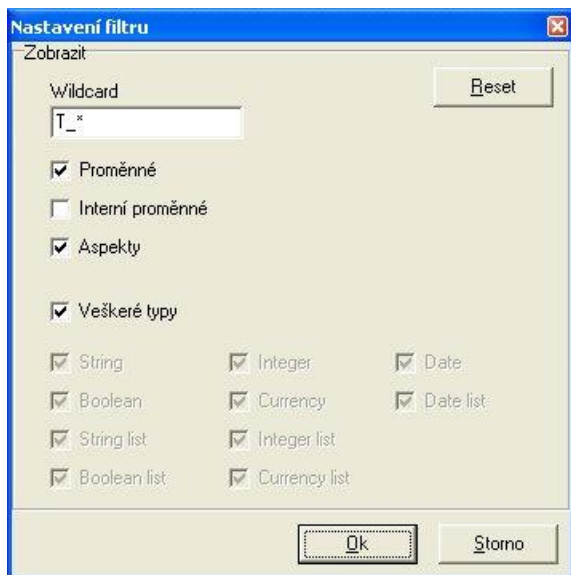
Často používaná je proměnná MTBCaseTitle. Pokud je tato proměnná typu string nadefinována, stává se její hodnota nastavená ve skriptu názvem procesu.


- Export a import proměnných

Na panelu nástrojů dialogu slovníku dat jsou tlačítka pro export a import proměnných . Výsledkem exportu je soubor s příponou **.dad**. Používá se v případě přenosu proměnných do jiné databáze.

- Filtr proměnných ve slovníku dat

Výbornou pomůckou při práci se slovníkem dat je filtr proměnných . Pro nastavení slouží dialog...



Do pole můžete vepsat hledaný řetězec v názvu proměnné s využitím náhradních znaků * ?. Další parametry vyhledávání určíte zatržením požadovaných vlastností výběru. Tlačítko  slouží k výmazu pole a nastavení zatržitek do standardního stavu. Výsledkem výběru po stisku **OK** je požadovaný seznam proměnných zobrazený v okně slovníku.

Termíny

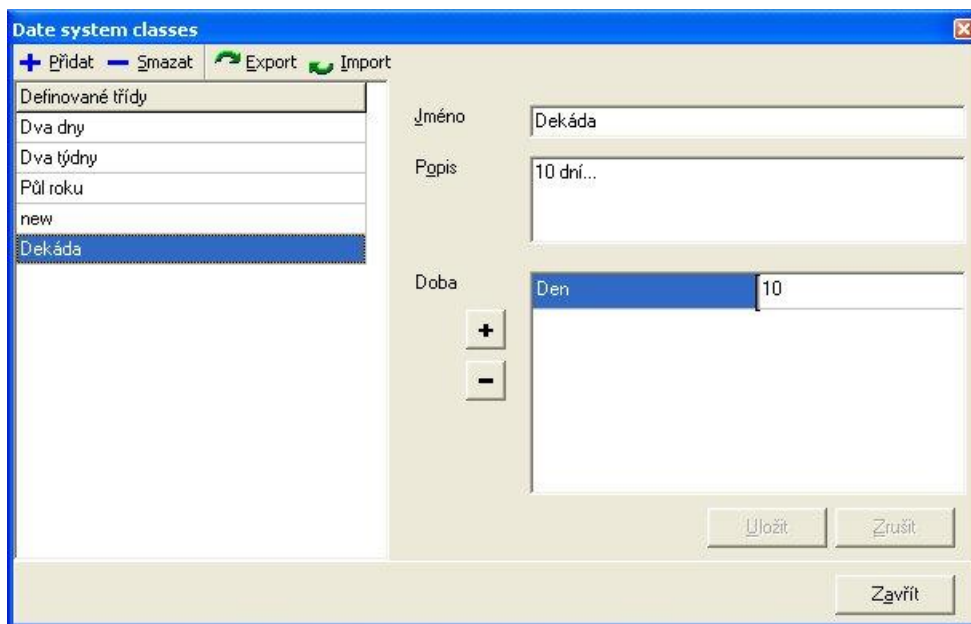
- Přidělení časové třídy

Workflow Designer nabízí možnost přidělit workflow nadefinovanou časovou třídu. Ta Vám umožní časově omezit plnění jednotlivých pracovních kroků. Pomocí workflow tak můžete řídit i dlouhodobé aktivity. V rámci M/TeamBridge řídí časové třídy termíny dokončení pracovního kroku.

Můžete použít některou z již nadefinovaných tříd nebo si vytvořit vlastní.

- Vytvoření nové časové třídy

Pokud si ze seznamu časových tříd nevyberete, můžete si vytvořit třídu podle vlastních potřeb. Otevřete dialog TERMÍNY



Vlevo se nachází seznam již nadefinovaných tříd. Pokud na název některé z nich klepnete, vpravo se zobrazí její details, jako například popis, přesná délka trvání apod.

Pro vytvoření nové třídy použijte toto tlačítko . Poté je nutné vyplnit údaje o nové třídě:


6.8.2.1.1 Jméno a Popis

Zadejte jméno nové třídy a popis.


6.8.2.1.2 Doba

A nakonec doplňte přesnou délku trvání. V levém políčku vyberte z možností časovou jednotku.



Do pravého políčka zadejte počet pomocí šipek, které se tam po klepnutí zobrazí . Šipkou směřující dolů je možné zadat i zápornou hodnotu. Tímto způsobem můžete zadat například lhůtu jeden měsíc bez 3 dní:




Pokud si nevyberete žádnou z nabízených časových jednotek, je možné tyto jednotky kombinovat. Zadejte první jednotku a potom klepněte na tlačítko se znaménkem plus  a vyberte další:



Poznámka:

Každou jednotku času můžete použít pouze jednou.

Po vyplnění všech údajů nové časové třídy potvrďte zadání následujícím tlačítkem .

Pokud se rozhodnete zadání smazat ještě před uložením, použijte tlačítko .

Návrhář masek

Spustí se program imDesigner.exe (návrhář masek). Práce s maskami viz příručku Návrhář masek M/TeamBridge.

Externí operace

M/Teambridge poskytuje speciální aplikace s nezbytnými daty a prostředky.

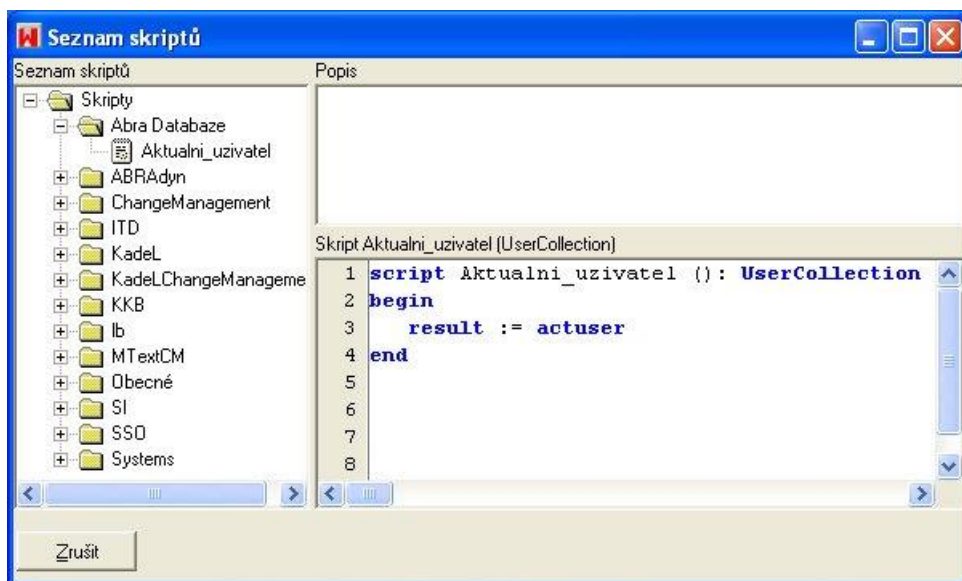
Všechny informace, které speciální aplikace vyžaduje k vykonání pracovního kroku jsou zadávány v dialogu...



Administrace externích operací je prováděna v levé části okna ve stromu složek a v pravé části okna v seznamu aktivit uložených ve vyznačené složce.

Seznam skriptů

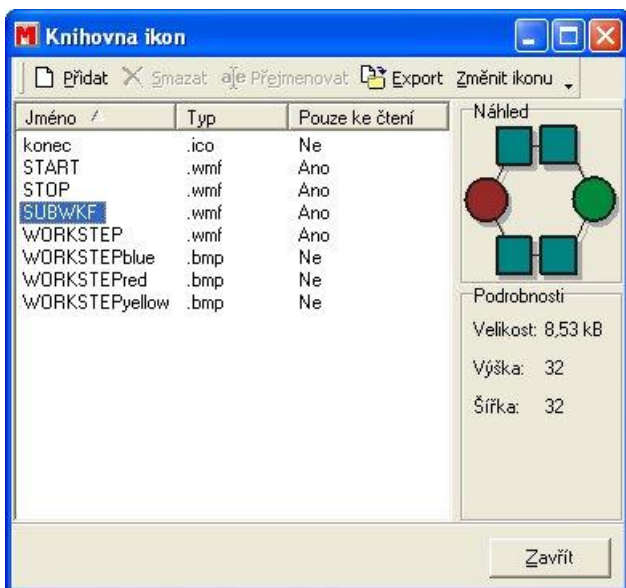
Vyvolá dialog pro správu skriptů, které jsou uspořádány do složek ve stromové struktuře.



Skripty lze pouze prohlížet, do režimu editace skriptů se dostanete poklepáním na název požadovaného skriptu ve stromové struktuře, v popup menu názvu skriptu výběrem možnosti Editovat skript nebo kombinací kláves.<Ctrl + Enter>.

Knihovna ikon

Zde jsou soustředěny dostupné ikony pro zobrazení uzlů workflow...












Zde můžete použitím tlačítek na panelu nástrojů...












- přidat novou ikonu (typy souborů **.jpg**, **.bmp**, **.wmf**)
- smazat ikonu
- přejmenovat ikonu
- exportovat ikonu do souboru **.ico**
- změnit ikonu

Mazání a přejmenování se týká pouze ikon, které mají ve sloupci "Pouze ke čtení" uvedeno "Ne". Čtyři standardní ikony START, STOP, SUBWKF A WORKSTEP nelze smazat ani přejmenovat, pouze změnit (vybrat jiný obrázek).

Ikonu vyberete tlačítkem  nebo akci zrušíte tlačítkem .

6.8.3 Přehled ikon panelu nástrojů

-  nové workflow
-  otevře již existující workflow
-  uloží workflow nebo provedené změny
-  vytiskne všechna okna nastavená v dialogu NASTAVENÍ TISKU
-  aktualizace
-  zavře program
-  slouží ke změnám v grafu workflow
-  vloží počáteční uzel workflow
-  vloží koncový uzel workflow

-  vloží pracovní krok
-  vloží vnořené workflow
-  vytvoří hranu mezi uzly
-  provede test správnosti workflow (test konzistence)
-  zvětší graf workflow
-  zmenší graf workflow
-  upraví velikost grafu workflow na 100%
-  upraví velikost grafu workflow na stránku
-  zobrazí nebo skryje okno Navigátor
-  zobrazí nebo skryje okno Editor vlastností
-  zobrazí nebo skryje okno Průzkumník objektů

7 Návrhář masek

7.1 Použití Návrháře masek

Používá se jako nástroj pro vytváření masek. Pomocí masek jsou proměnným přiřazeny hodnoty. Jedná se o globální proměnné uložené ve Slovníku dat. Lokální proměnné definované ve skriptu nelze v maskách použít.

Masky jsou využívány jako pracovní prostředek k řízení běhu workflow


Masky obsahují objekty, např. vysvětlující texty, pole proměnných, ovládací prvky. Výsledkem zpracování masky je logická hodnota, kterou lze vyhodnotit ve skriptu workflow.

7.1.1 Přihlášení a odhlášení

Návrhář masek lze spustit samostatně (imDesigner.exe) nebo z WorkflowDesigneru v menu NÁSTROJE/ADMINISTRACE.

Přihlášení

Samostatné spuštění Návrháře masek

- spustit imDesigner.exe ve složce M\bin reprezentovaný ikonou  a přihlásit se dle nastavení v M\User. Přihlášení se nevyžaduje, pokud je spuštěn jiný modul Serie M/... kde přihlášení již proběhlo.



- zadat JMÉNO a HESLO a potvrdit tlačítkem OK.


Spuštění z WorkflowDesigneru

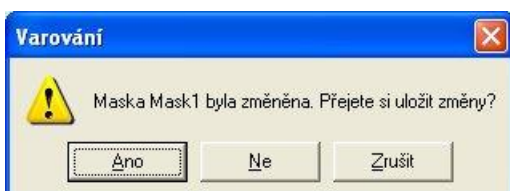
V menu NÁSTROJE/ADMINISTRACE WorkflowDesigneru volbou *Návrhář masek*



Přihlášení není třeba, je platné z WorkflowDesigneru.

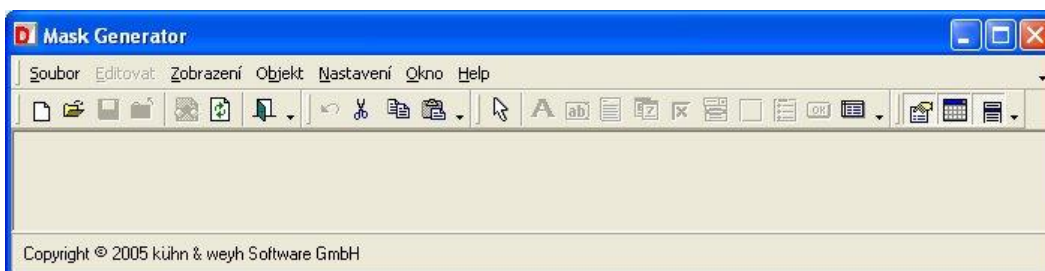
Odhlášení

Ukončení práce s Návrhářem masek provedete ikonou  na panelu nástrojů Standard, v menu SOUBOR/KONEC nebo křížkem v pravém horním rohu okna Návrháře. Pokud nebyly dosud uloženy poslední změny, je položen dotaz

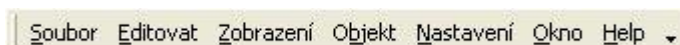


7.2 Prostředí Návrháře masek

Po spuštění Návrháře masek se objeví okno ...




Pracovní plocha slouží pro práci s maskami. Nad pracovní plochou je **menu**



a ovládací panely




7.2.1 Editor vlastností

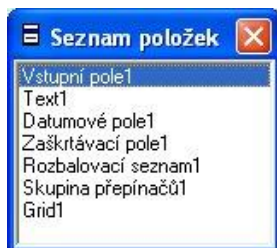
Ikonou  na panelu nástrojů Speedview nebo v menu ZOBRAZIT/EDITOR VLASTNOSTÍ se editor vlastností zapíná (případně i vypíná)



Editor vlastností se vztahuje vždy k vybranému objektu, tj. k samotné masce nebo k objektům na ní uloženým. Jsou zde soustředěny všechny vlastnosti objektu, které lze měnit buď vyplněním hodnot do editoru nebo přímo nastavením objektu pomocí myši (např. rozměry a umístění objektu).

7.2.2 Seznam polí

Seznam polí (nebo též položek, proměnných) se zapíná/vypíná ikonou  na panelu nástrojů Speedview nebo v menu ZOBRAZIT/SEZNAM PROMĚNNÝCH




Obsahuje seznam všech objektů umístěných v masce. Objekty jsou reprezentovány přiděleným jménem (název typu objektu+pořadové číslo tohoto typu v masce). Pořadí proměnných v seznamu určuje pořadí, ve kterém budou tyto proměnné k dispozici uživateli při vyplňování masky s použitím klávesy <Tab>

7.3 Správa masek

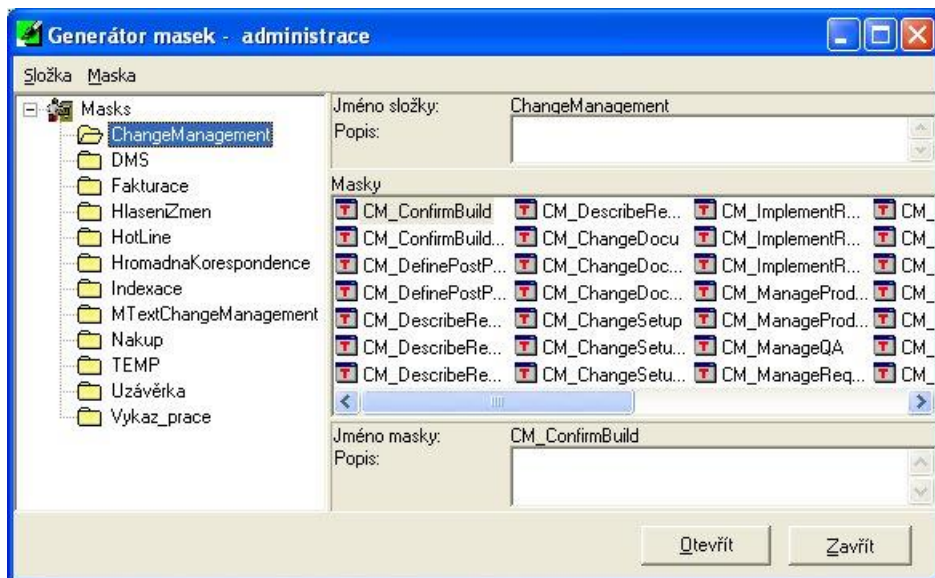
7.3.1 Funkce pro správu masek

Okno pro správu masek

Masky definované v systému jsou pro lepší přehled uloženy ve stromové struktuře. Okno pro zobrazení lze vyvolat několika způsoby:

- Ikonou  na panelu nástrojů Standard nebo
- Kombinací kláves <Ctrl+O> nebo
- Z menu SOUBOR/OTEVŘÍT nebo
- Z menu SOUBOR/ADMINISTRÁTOR MASEK

Otevře se okno pro správu masek...



kde je obsažena...

- stromová struktura složek k uložení masek
- popis označené složky
- seznam masek, které leží v označené složce
- popis označené masky

Vytvoření nové složky

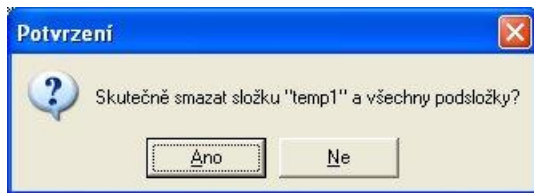
V okně pro správu masek...

- Vyznačte složku, ve které budete vytvářet podsložku.
- V menu vyberte složka/nová složka nebo v popup menu nová složka
- Přepište standardní jméno v rámečku u vytvořené složky požadovaným jménem složky a potvrďte <Enter>
- Do pole vpravo nahoře můžete vepsat popis složky.

Smazání složky

V okně pro správu masek...

- Vyznačte složku, kterou chcete smazat.
- V menu vyberte složka/smazat složku nebo v popup menu smazat složku
- Objeví se dialog pro potvrzení smazání



- Po potvrzení Ano se složka smaže včetně všech podsložek a masek, které obsahovala.

Uzavření okna pro správu masek


Okno pro správu masek uzavřete...

- tlačítkem Zavřít vpravo dole nebo
- v menu složka/konec


7.3.2 Funkce pro práci s maskou

Otevření masky

Masku otevřete v Návrhářii masek...

- ikonou  na panelu nástrojů Standard nebo
- v menu soubor/otevřít... nebo
- kombinací kláves <Ctrl O>

Objeví se okno pro správu masek (viz výše), kde vyberete složku a v ní požadovanou masku, kterou otevřete...

- tlačítkem  nebo
- poklepáním na název masky

Upozornění:

Pokud vyberete pomocí klávesy <Ctrl> více masek, otevře se jen první vybraná maska.

Export masky

Masku můžete exportovat do souboru .scr na disk v otevřeném okně pro správu masek (viz výše). Nejprve masku vyberte. Export provedete...

- v menu maska/exportovat masku nebo
- v popup menu masky volbou exportovat masku

Otevře se standardní okno pro uložení souboru na disk, kde zadáte jméno souboru a vyberete místo uložení na disku.

Uložíte tlačítkem .

Smazání masky

Vybranou masku smažete v otevřeném okně pro správu masek...

- v menu maska/smazat masku nebo
- v popup menu masky volbou smazat masku

Vybraná maska bude smazána ze seznamu masek.

Více informací v kapitole Uložení masky.

7.3.3 Verze masky

V systému jsou uloženy všechny verze masky. Můžete se vrátit do požadovaného časového bodu a aktivovat předchozí verzi masky.

Prohlížení verzí masky

Vyberte masku v otevřeném okně pro správu masek. Okno zobrazení verzí vyvoláte...

- v menu maska/ukázat verze nebo
- v popup menu masky volbou ukázat verze nebo
- kombinací kláves <Ctrl V>

Otevře se okno verzí masky...

Číslo	Datum	Uživatel
23	19.8.2008 12:54:26	Vorlová, Libuše
22	19.8.2008 12:52:17	Vorlová, Libuše
21	14.8.2008 16:39:51	Vorlová, Libuše
20	14.8.2008 15:16:41	Vorlová, Libuše
19	14.8.2008 15:15:38	Vorlová, Libuše
18	14.8.2008 14:29:05	Vorlová, Libuše
17	14.8.2008 14:27:12	Vorlová, Libuše
16	14.8.2008 14:16:48	Vorlová, Libuše

Popis workflow

Zavřít

Verze masky jsou seřazeny sestupně.

V seznamu je uvedeno...

- pořadové číslo
- datum a čas uložení
- uživatel, který uložil verzi

Nastavení aktuální verze

Obvykle je platná nejnovější verze masky. Můžete však nastavit jako platnou některou dřívější verzi masky. V otevřeném okně verzí masky...

- vyznačte požadovanou verzi
- vyvolejte pravým tlačítkem myši popup menu
- vyberte možnost nastavit jako aktuální

Verze se stane aktuální verzí na nejvyšší pozici v seznamu verzí. Údaje Datum a Uživatel budou aktualizovány. Zároveň zůstane verze na původním řádku seznamu.

Export verze

Libovolnou verzi masky můžete také exportovat jako soubor .scr a uložit na disk. V okně verzí masky vyberte verzi a zvolte v popup menu verze EXPORTOVAT VERZI

Otevře se standardní okno pro uložení souboru na disk. Po vyhledání umístění souboru, zadání jeho jména a potvrzení tlačítkem **Uložit** bude soubor uložen.

Import verze

Novou verzi masky můžete také naimportovat z dříve exportovaného souboru .scr uloženého na disku. V okně pro správu masek vyberte masku a pro import nové verze...

- v menu volte maska/přidat verzi nebo
- v popup menu masky zvolte přidat verzi

Otevře se standardní okno pro vyhledání souboru ve složkách na disku. Po vyznačení souboru a potvrzení tlačítkem Otevřít bude verze přidána.

Smazání verze

Verzi masky můžete také smazat. V okně verzí masky vyberte požadovanou masku a v popup menu zvolte SMAZAT VERZI. Verze masky bude smazána.

Poznámka:

Číslování verzí zůstane zachováno, takže je patrné, kde chybí smazaná verze.

7.4 Tvorba masek

7.4.1 Než začnete

Proměnné


Objekty používané v masce obsahují proměnné.

Doporučení:

Je vhodné definovat předem všechny proměnné, které budou pro masku potřebné. Jedná se o globální proměnné (ve skriptech **wkf.jmeno_promenne**), které jsou obsaženy ve **Slovníku dat** (Data dictionary).

Při vytváření masky v okamžiku, kdy je třeba umístit do masky objekt reprezentovaný touto proměnnou, musí být tato proměnná již k dispozici ve slovníku, jinak není dovoleno masku s objektem bez proměnné uložit.

Slovník dat

- Z prostředí Návrháře masek lze spustit Slovník dat z menu ZOBRAZIT/DATA DICTIONARY.
- Z prostředí WorkflowDesigneru spustíte Slovník z menu NÁSTROJE/ ADMINISTRACE/ DATOVÝ SLOVNÍK – ikona .

Podrobný popis zadávání proměnných do Slovníku dat naleznete v kapitole Slovník dat pro definici proměnných na straně 79.


Poznámka:

Proměnné nadefinované pro účely masek je nutné začlenit do workflow, kde bude maska použita (pomocí dialogu v menu EDITOVAT/VLASTNOSTI WORKFLOW ve WorkflowDesigneru)

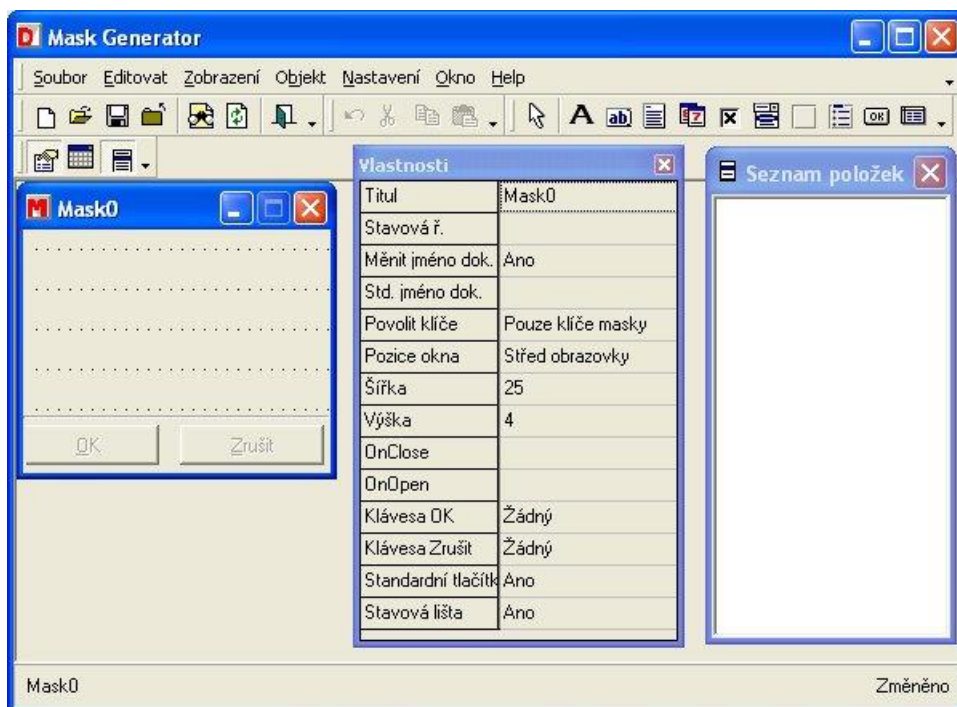
7.4.2 Nová maska

Vytvoření masky

Tvorba nové masky se zahájí...

- klepnutím na ikonu  na panelu nástrojů Standard nebo
- v menu soubor/nový. nebo

- kombinací kláves <Ctrl+N>



Vlevo je okno nové masky. Editor vlastností se vztahuje k masce (řádek Titul). Seznam položek je prázdný.

Editace masky

Objekty můžete v masce přesouvat a měnit jejich rozměry myší nebo můžete tyto údaje zadávat číselně do editoru vlastností objektu.

Pro editaci masky máte k dispozici tyto pomůcky:

Panel nástrojů

Obsahuje ikonu pro návrat o krok zpět a ikony pro práci se schránkou. Pomocí schránky lze kopírovat objekty a to i z masky do jiné masky. Pozor při tom na duplicitu názvů objektů.



Mřížka

Mřížka určuje řádky a sloupce v masce, ke kterým se objekty zarovnávají. Mřížku lze vypnout v menu ZOBRAZENÍ/MŘÍŽKA.

Zarovnání objektů


Tento panel se zapíná/vypíná ikonou  na panelu Speedview nebo v menu ZOBRAZENÍ/PALETA ZAROVNÁNÍ.



Používá se pro 2 a více objektů, které musí být všechny vyznačeny (pomocí klávesy <Ctrl>). Horní řádek panelu posunuje objekty vodorovně, dolní svisle.

Testování masky

Po vytvoření masky máte možnost ji otestovat. K otestování masky...

- stiskněte ikonu  na panelu nástrojů Standard nebo
- volte v menu soubor/test masky nebo
- použijte funkční klávesu <F6>

Otevře se maska v podobě, jak ji uvidí uživatel při průběhu workflow. Můžete zde otestovat nastavené kontroly proměnných v objektech (viz kapitola Vlastnost Kontrola na straně 97)




a návratovou logickou hodnotu masky

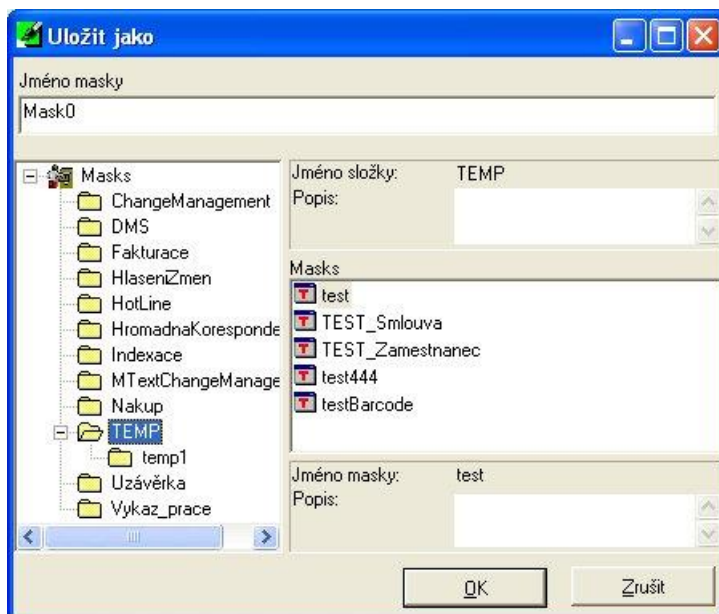


Uložení masky

Po skončení editace a testování masky, je třeba ji uložit...

- stiskem ikony  v menu Standard nebo
- v menu Soubor/uložit nebo
- kombinací kláves <Ctrl S>

Otevře se okno pro uložení masky...




Do horního řádku vyplníte jméno masky a vyberete složku, kam má být maska zařazena. Po stisku tlačítka OK je maska uložena.

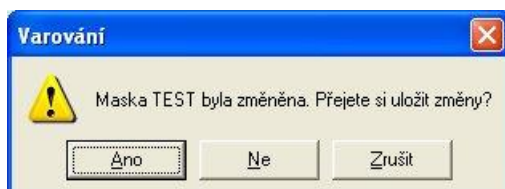
Při ukládání změn ve stávající masce postupujete stejně, ale okno pro uložení se neotvírá.

Pokud chcete vytvořit kopii masky pod jiným jménem, použijte volbu SOUBOR/ULOŽIT JAKO... Otevře se stejné okno jako pro první uložení masky, kde zadáte nové jméno a kopii masky uložíte do požadované složky.

Uzavření masky


Masku uzavřete...

- stiskem ikony  na panelu nástrojů Standard nebo
- v menu soubor /zavřít nebo
- křížkem v pravém horním rohu masky
- Pokud nebyly provedeny od posledního uložení masky žádné změny, maska se uzavře.
- V případě změn se objeví dotaz...



- Ano – maska je uzavřena a změny jsou uloženy
- Ne – maska je uzavřena a změny nejsou uloženy
- Zrušit – maska zůstává otevřena

7.4.3 Editor vlastností masky

Pokud není editor viditelný, ikonou  na panelu nástrojů Speedview nebo v menu ZOBRAZIT/EDITOR VLASTNOSTÍ se editor zapíná (případně i vypíná)

Vlastnosti	
Titul	Zaměstnanec
Stavová ř.	osobní údaje
Měnit jméno dok.	Ano
Std. jméno dok.	
Povolit klíče	Pouze klíče masky
Pozice okna	Střed obrazovky
Šířka	25
Výška	4
OnClose	
OnOpen	
Klávesa OK	Žádný
Klávesa Zrušit	Žádný
Standardní tlačítka	Ne
Stavová lišta	Ano

7.4.4 Nastavení vlastností masky

V okně editoru vlastností lze nyní nastavit vlastnosti masky.

Titul

Tento text bude v záhlaví okna masky (měl by vystihovat, jaké údaje se v masce zadávají).

Stavový řádek

Další vysvětlující text. Zobrazen bude na dolním okraji okna, ale jen v případě volby Ano na posledním řádku
Stavová lišta

Měnit jméno dok.

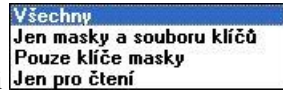


Zde jsou na výběr možnosti **Ano**, **Ne**, **Automaticky**. Slouží pro vnitřní potřebu, doporučuje se ponechat **Ano**.

Standardní jméno dokumentu

Slouží pro vnitřní potřebu, doporučuje se ponechat prázdné.

Povolit klíče



Zde jsou na výběr možnosti **Všechny**, **Jen masky a souboru klíčů**, **Pouze klíče masky**, **Jen pro čtení**. Obvykle **Pouze klíče masky**.

Pozice okna


Určení, kde se okno objeví uživateli. Obvykle je nastaven střed obrazovky.

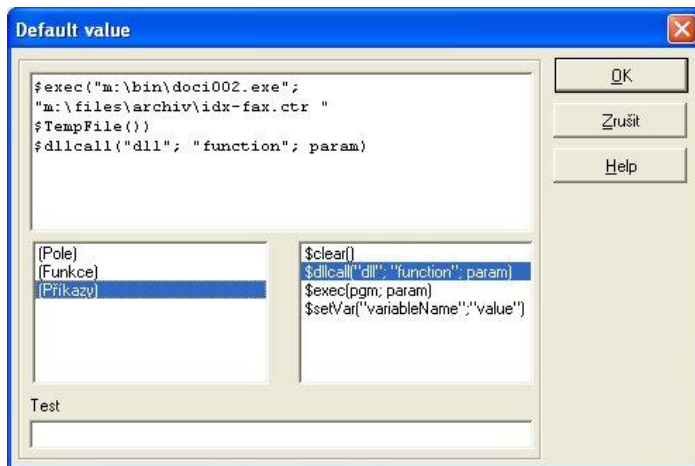


Šířka a výška

Rozměry okna lze měnit tažením myši nebo přímo vepsat. Výška se udává počtem řádků, šířka se udává v bodech (sloupcích).

OnClose, OnOpen

Zde je možnost při otevření (OnOpen) nebo uzavření (OnClose) masky volat externí program nebo funkci z dll knihovny. Stiskem tlačítka  na konci pole se zobrazí dialog pro zadání potřebných příkazů.



Poklepáním na vybrané pole, funkci nebo příkaz se syntaxe zobrazí v horním okně, kde doplníte konkrétní údaje.

Příklady:

- Volání programu

```
$exec("docislct.exe"; "control-file"
header-file)
```

- Volání funkce

```
$dllcall("dll name"; "function
name"; parameters)
```

- Pro volané dll platí následující pořadí vyhledávání:
 - složka produktů Serie M/...
 - aktuální složka
 - složka Windows nebo
 - složka specifikovaná cestou (path)

Klávesa OK, klávesa Zrušit

Zde můžete pro standardní tlačítka **OK** a **Zrušit** zvolit funkční klávesy nebo <Enter> či <Esc>, které potom přeberou funkci těchto tlačítek. Lze použít i pokud je zvolena možnost **Standardní tlačítka Ne**.

Standardní tlačítka

Tyto objekty budou viditelné pouze při volbě Ano



Uživatel stiskem **OK** potvrdí uložení proměnných, maska se uzavře a vrátí logickou hodnotu TRUE, která se vyhodnocuje pro další běh workflow. Místo stisku tlačítka lze použít podržené písmeno jako HotKey (O).

Po stisku **Zrušit** (nebo HotKey Z) se zjišťuje, zda došlo ke změně hodnot proměnných od posledního uložení. V tom případě je položen dotaz



Při stisku **Zrušit** se maska neuzavře a je možno pokračovat v práci.

Volbou **Ano** se maska uzavře, ale hodnoty proměnných se neuloží a je vrácena hodnota FALSE.

Upozornění:

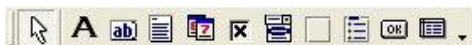
Pokud nastavíte u vlastnosti Standardní tlačítka hodnotu **Ne**, pak musíte zajistit závěrečné vyhodnocení masky jiným způsobem (např. nastavením vlastností **Klávesa OK** a **Klávesa Zrušit**, použití objektu **Tlačítko** – viz dále)

Stavová lišta

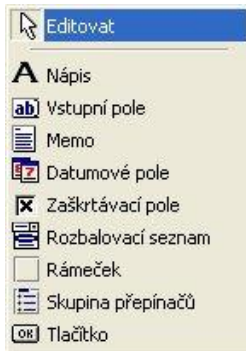
Bude viditelná pouze při volbě Ano.

7.5 Objekty masky

Pro vkládání objektů do masky se používá panel nástrojů **Objekty**




nebo menu OBJEKT

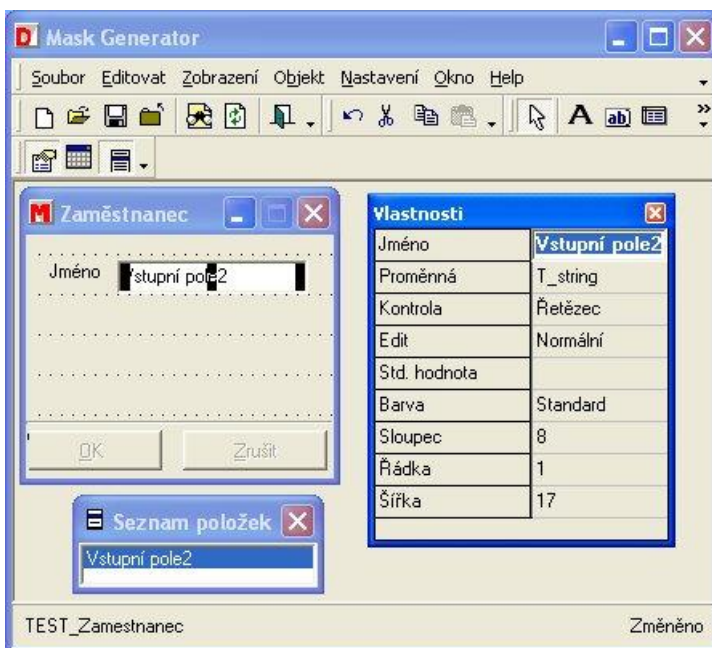


7.5.1 Postup vkládání objektů

Po klepnutí na ikonu požadovaného objektu přesuňte kurzor do okna masky. Kurzor se změní na symbol křížku. Klepnutím do masky vytvoříte objekt. Myši jej můžete posouvat v rámci masky. Po uvolnění tlačítka myši se objekt přichytí na nejbližší sloupec a řádek. Rozměry můžete upravovat tažením za úchyty v rozích a na hranách objektu, pokud to typ objektu dovoluje.

Pokud nechcete objekt zakreslovat, klepněte na ikonu Editovat  na panelu nástrojů nebo v menu. Kurzor se změní opět na šipku a můžete editovat masku.

Při zakreslení objektu (a také kdykoliv je objekt vybrán) se v Editoru vlastností objeví vlastnosti objektu.



7.5.2 Společné vlastnosti objektů

Některé vlastnosti se vyskytují u více objektů, proto budou popsány společně.

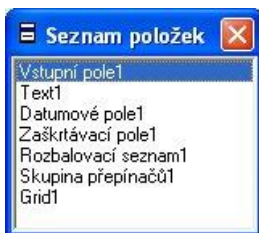
Vlastnost Jméno

Jméno je přidělováno automaticky podle typu objektu s přidáním pořadového čísla. Obvykle není nutno jméno měnit, pouze v případě duplicity je vyžadováno zadání jiného jména



Duplicita vzniká např. při kopírování objektů přes schránku z jiné masky.

Objekty, které mají ve vlastnostech položku **Jméno** (všechny kromě **Nápisu**, **Tlačítka** a **Rámečku**) se pod tímto jménem objevují v okně seznamu položek



Pořadí objektů můžete měnit tažením myši, což nemá vliv na vzhled masky, ale na pořadí, v jakém jsou objekty dávány k dispozici uživateli k editaci při použití klávesy <Tab>.

Vlastnost Text

Objekty **Tlačítko** a **Nápis** nemají vlastnost **Jméno**, jejich název s pořadovým číslem je automaticky umístěn do pole **Text** ve vlastnostech, kde jej změníte podle potřeby.

Objekty **Zaškrťovací pole** a **Skupina přepínačů** mají ve vlastnostech pole **Jméno** i **Text**, kam je automaticky umístěn stejný název s pořadovým číslem. V poli **Jméno** jej ponecháte (pokud není důvod jej měnit, např. duplicita) a do pole **Text** umístíte potřebná záhlaví těchto objektů, která budou viditelná v masce.

Vlastnost Proměnná

Všechny objekty, které mají ve vlastnostech **Jméno**, mají zde také pole **Proměnná**. Klepnutím na šipku na konci pole se zobrazí seznam proměnných ze Slovníku dat



Nejsou zobrazeny všechny proměnné, ale vždy jen ty, které jsou pro daný typ objektu povoleny (viz popis objektů).

Upozornění:

V objektu **Skupina přepínačů** musíte před výběrem proměnné nastavit vlastnost **Výsledek**



Při výběru **Index** dostanete k dispozici pouze proměnné typu *integer*, při výběru **Text** pouze proměnné typu *string*. (viz Objekt Skupina přepínačů)

Jaký objekt lze použít pro daný typ proměnné:

string	Vstupní pole, Memo(text), Rozbalovací seznam, Skupina přepínačů
integer	Vstupní pole, Rozbalovací seznam, Skupina přepínačů
date	Datumové pole, Rozbalovací seznam
boolean	Zaškrťovací pole


všechny typy list Grid


Vlastnost Kontrola

Většina objektů s proměnnou má ve vlastnostech i pole **Kontrola** (Vstupní pole, Memo(text), Datumové pole, Rozbalovací pole, Grid).

Kontrola probíhá v okamžiku, kdy uživatel provede s maskou akci s očekávaným výsledkem True (stisk standardního tlačítka OK nebo jinou akci). Nastavené kontroly proměnných se vyhodnotí a při nesouhlasu je vydáno

příslušné upozornění v samostatném okně a na stavové liště, pokud je zapnuta. Pro lepší přehled, kterého objektu se kontrola týká, bliká tento objekt

Uživatel potvrdí upozornění tlačítkem . Akce není dokončena, hodnoty nejsou uloženy do proměnných a maska není uzavřena. Uživatel může nyní provést opravu proměnné, která nevyhověla kontrole a akci zopakovat.

Při vytváření masky si můžete tyto kontrolní funkce vyzkoušet pomocí ikony  v panelu nástrojů Standard (viz Test masky). Kontroly zadávaných hodnot je též možno provádět ve skriptu s opakovaným voláním masky při nesprávných hodnotách.

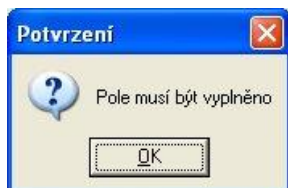
V kontrolních dialogích je zobrazeno standardní nastavení, které můžete změnit:

Kontrola pro typ Řetězec a Grid

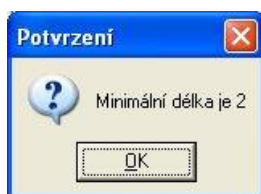


Řetězec	
Povinné pole	Ne
Max. délka	45
Min. délka	0
Kontrola	(Řetězec)
Znak pro prefix	None
Varovat při doplnění	Ne

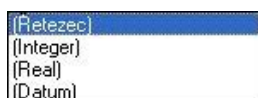
- **Povinné pole** nastaveno na Ano - pokud uživatel pole nevyplní, objeví se upozornění:



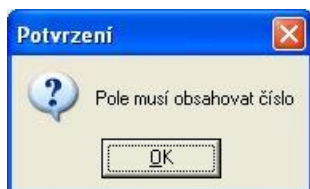
- **Max. délka:** Omezení délky se projeví již při zápisu hodnoty, více než zadaný počet znaků nelze zapsat.
- **Min. délka:** pokud uživatel zadá menší počet znaků, objeví se upozornění



- **Kontrola:** Zde lze určit, jaké znaky mají být v řetězci, tj. libovolné znaky, číslice, číslice s desetinnou čárkou (měna) nebo tvar datumu.



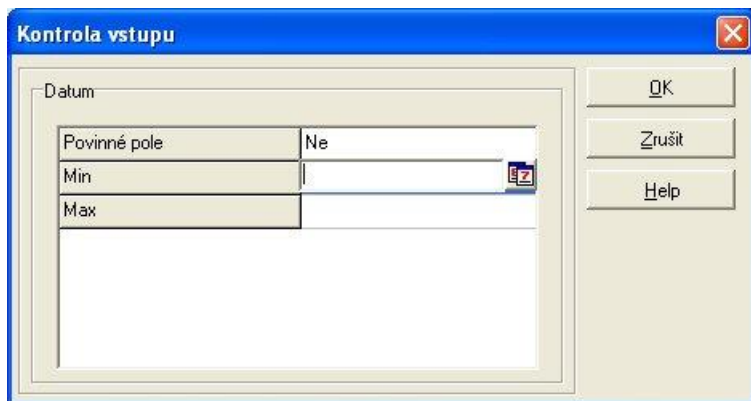
- Např. při nastavení (Integer) a zadání písmen se objeví upozornění:



Znak pro prefix: Doporučení – nechat nastaveno na None.


Varovat při doplnění: Doporučení – nechat nastaveno na Ne.

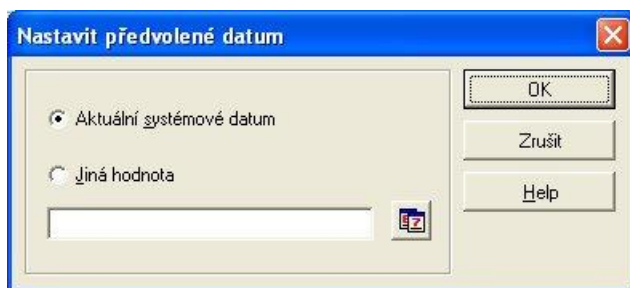
Kontrola pro typ Datum




- **Povinné pole** nastaveno na Ano - pokud uživatel pole nevyplní, objeví se upozornění:



- **Min, Max:** Klepnutím do pole a pak na ikonu  na konci pole vyvoláte dialog



- **Aktuální systémové datum:** V poli se objeví text *Today*, Min/Max datum bude odpovídat datu použití kontroly.

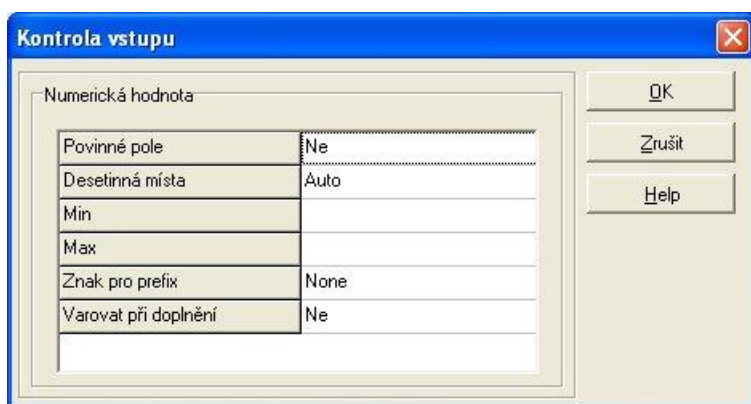
- **Jiná hodnota** .Stiskem ikony  zobrazíte kalendář, vyberete datum Min/Max a potvrdíte



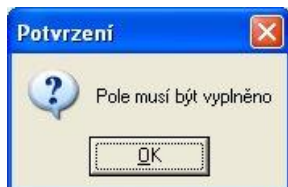


Upozornění: Datum Min musí být menší nebo rovno Max, jinak vznikne situace, kdy žádné datum nevyhovuje podmínce.

Kontrola pro typ Číslo (numerická hodnota) .



- **Povinné pole** nastaveno na Ano - pokud uživatel pole nevyplní, objeví se upozornění:



- **Pozn.** U typu proměnné **integer** při použití masky ve workflow je předem vyplněna hodnota 0, takže tato situace nenastane.
- **Desetinná místa:** **Doporučení:** ponechat nastavení **Auto.**, desetinná místa u typu proměnné integer nepoužívat, jinak se hodnota proměnné nastaví na 0).
- **Min/Max:** Ručně zadat celá čísla. **Min** musí být menší nebo rovno **Max**, jinak nastane situace, že žádné číslo nevyhoví kontrole.
- **Znak pro prefix:** **Doporučení** – nechat nastaveno na None.
- **Varovat při doplnění:** **Doporučení** – nechat nastaveno na Ne.

Vlastnost Editace

Uživatelům lze určit způsob práce s některými objekty (Vstupní pole, Memo(text), Datumové pole) v poli Edit ve vlastnostech. Výběr je z těchto možností:



Možnost **Normální** znamená, že uživatel může objekt editovat, tj. měnit hodnotu proměnné. **Jen čist** znamená, že uživatel bude informován o hodnotě proměnné, ale nemůže ji změnit (nepsaným zvykem je použít šedou barvu pozadí editačního pole pro odlišení needitovatelných položek od editovatelných, kterým se ponechá standardní bílé pozadí). **Neviditelné** objekty uživatel nevidí, mohou být využity např. na nastavení standardních hodnot (viz dále).

Vlastnost Standardní hodnota

Toto pole je použito ve vlastnostech objektů Vstupní pole, Datumové pole, Rozbalovací seznam, Skupina přepínačů a Grid. Zde uvedená hodnota je použita při otevření masky jako hodnota proměnné.

Stejného výsledku lze docílit nastavením hodnoty proměnné ve skriptu před otevřením masky s výhodou kontroly, za jakých okolností se maska otevírá a zda má být hodnota proměnné skutečně změněna. Proto se nastavení standardní hodnoty proměnné přímo v masce doporučuje jen v odůvodněných případech.

Vlastnosti Sloupec a řádek

Umístění všech objektů lze stanovit v rámci okna masky jako hodnotu řádku (řádek v mřížce) a sloupce (bod v mřížce). Hodnoty se vztahují k levému hornímu rohu objektu.

Vlastnosti Šířka a výška

U většiny objektů můžete stanovit výšku (počet řádků mřížky) i šířku (v bodech mřížky).

U **Vstupního** a **Zaškrťovacího** pole lze stanovit pouze šířku, výška je vždy jeden řádek.

Datumové pole a **Nápis** nemají ani jeden z rozměrů uveden v editoru vlastností, velikost je dána rozměry zapsaných hodnot.

Vlastnost Barva

Barvu nelze nastavit u Tlačítka a Rámečku.

Nastavení barvy u Nápisu, Zaškrťovacího pole a Skupiny přepínačů se projeví změnou barvy písma.

Změna barvy u ostatních objektů se týká barvy pozadí, proto pozor na dodržení čitelnosti písma, volit spíše světlejší barvy.

7.5.3 Popis objektů

Některé vlastnosti objektů již byly popsány v předcházející kapitole. Nyní následuje popis objektů v pořadí, jak je naleznete na panelu nástrojů nebo v menu.

Nápis



Používá se k popiskům a vysvětlivkám, zápis do pole Text.

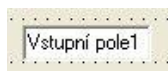


Vstupní pole



Nejpoužívanější objekt, možno vybrat proměnnou typu string, integer a currency (na tu ale není zabudována kontrola). Výška vždy 1 řádek.

Vlastnosti	
Jméno	Vstupní pole1
Proměnná	
Kontrola	Žádný
Edit	Normální
Std. hodnota	
Barva	Standard
Sloupec	16
Řádka	1
Šířka	21

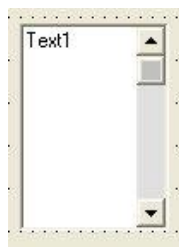


Memo (Text)



Používá se pokud je nutné vymežit na text oblast vyšší než jeden řádek. Lze použít pouze proměnnou typu string.

Vlastnosti	
Jméno	Text1
Proměnná	
Kontrola	Žádný
Edit	Normální
Barva	Standard
Posuvník	Žádný
Sloupec	10
Řádka	3
Šířka	9
Výška	1



Editor vlastností obsahuje pole Posuvník, kde určíte použití posuvníků v oblasti pro text.

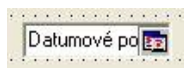



Datumové pole



Používá se pro proměnné typu date. Nelze určit vlastní rozměry, pouze umístění v masce.

Vlastnosti	
Jméno	Datumové pole1
Proměnná	
Kontrola	Žádný
Edit	Normální
Std. hodnota	
Formát	DD.MM.YYYY
Barva	Standard
Sloupec	6
Řádka	5




V poli formát lze stanovit použití formátu data DD.MM.YYYY nebo DD.MM.YY. Uživatel má k dispozici výběr z kalendáře po klepnutí na ikonu . Pokud uživatel zadává datum ručně, má možnost jej zadat bez dělicích znaků (teček) jak o číslo DDMMYYYY nebo DDMMYY.

Poznámka:

Další formáty (DD/MM/YYYY, DD,MM,YY atd.) je možno vepsat ručně. Doporučuje se pak vytvořit ve skriptu pro uživatele nápovědu, jaký je požadovaný tvar datumu.


Zaškrťovací pole

 Jediný objekt, kde lze použít proměnnou typu **boolean**.



Popis zadejte do pole **Text**, nikoli do pole **Jméno**!

Rozbalovací seznam

 Lze použít proměnnou typu date, integer a string.




Combo style

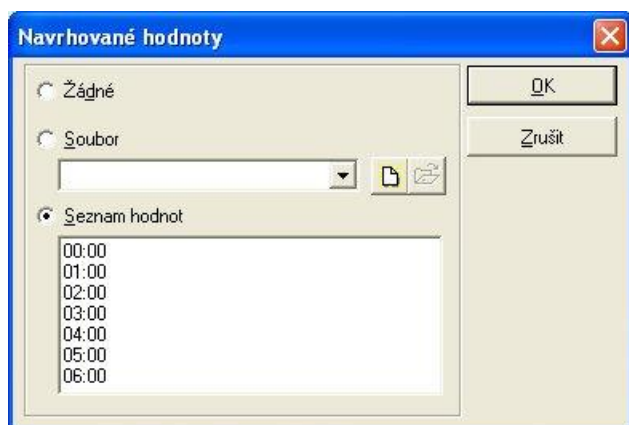
Zde jsou dvě možnosti .

První možnost znamená, že uživatel může vybranou hodnotu měnit nebo zapsat vlastní.

Druhá možnost znamená, že lze pouze vybírat ze seznamu nastaveném v poli **Hodnoty**.

Hodnoty

Do pole **Hodnoty** se zadávají položky pro rozbalení seznamu, ze kterých je pak uživatelem proveden výběr jedné položky. Na tuto hodnotu je nastavena proměnná. Klepnutím do pole **Hodnoty** a stiskem ikony  na konci pole se otevře dialog:



Pro navrhované hodnoty jsou tyto možnosti:

- **Žádné.** Pak může uživatel v případě nastavení pole **Combo style** na Editovatelné pouze vyplnit hodnotu, která se stane obsahem proměnné (ekvivalent objektu **Vstupní pole**).
- **Soubor:** V seznamu vyberte textový soubor, ve kterém jsou položky vyplněny na řádcích. Pokud soubor neexistuje, je možné jej vytvořit po stisku . Otevře se standardní okno pro uložení souboru a pak okno pro zápis hodnot (textový editor). Vybraný soubor lze otevřít ikonou a provádět změny.
- **Upozornění:** Pokud jste nevybrali složku, kde je uložena Serie M/...(obvykle m\bin), musíte ručně zadat cestu k .txt souboru do řádku pro soubor před jeho jméno: a
- Je také možné textový soubor vytvořit až v průběhu zpracování kroku ve skriptu pomocí funkce StringToFile z knihovny ActualDocument.dll
- **Seznam hodnot:** Hodnoty lze přímo vyplnit, pro odřádkování stisknete klávesu <Enter>.
- **Doporučení:** Ponechat jeden řádek prázdný, aby uživatel mohl vyprázdnit pole.

Seřadit

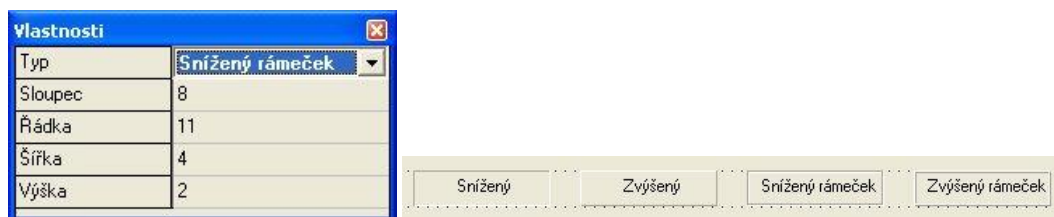
Volba Ano/Ne. Někdy je vhodné pro lepší hledání v seznamu zadané hodnoty seřadit, např. texty podle abecedy, čísla vzestupně. Jindy je seřazení nevhodné, např. názvy měsíců, dnů v týdnu atd.



Rámeček

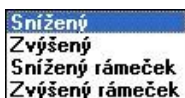


Grafický objekt bez proměnné. Používá se v maskách ke sdružování logicky souvisejících objektů ap.



Typ

K dispozici jsou čtyři podoby rámečku. První dvě se týkají celé plochy rámečku, druhé dvě jen ohraničení.



Skupina přepínačů



Používá se v případě, kdy je třeba vybrat jednu z více možností. Lze použít proměnnou typu integer nebo string, podle nastavení vlastnosti **Výsledek**. Záleží na dalším použití proměnné např. ve skriptu.


Upozornění:

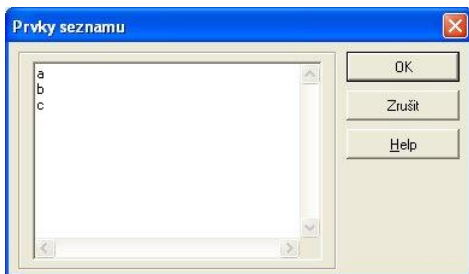
Vlastnost **Výsledek** je třeba nastavit před výběrem proměnné (viz dále u vlastnosti **Výsledek**)

Vlastnosti	
Jméno	Skupina přepínačů1
Proměnná	
Text	Skupina přepínačů1
Std. hodnota	
Položky	(Seznam)
Sloupce	1
Barva	Standard
Výsledek	Text
Sloupec	22
Řádka	10
Šířka	11
Výška	3




Položky

Klepnutím do pole a pak na ikonu  na konci pole se zobrazí dialog pro zadání položek:



Položky udávají popis jednotlivých výběrových bodů. Jejich počet udává počet těchto bodů. Lze použít i prázdnou hodnotu stiskem <Enter> (kolikrát stisknete Enter, tolik bude výběrových bodů bez popisu)

Sloupce

Standardně je zadán 1 sloupec. Po klepnutí do pole lze počet měnit pomocí ikony . Hodnoty menší nebo rovné nule se neberou v úvahu. Počet sloupců by se měl rovnat počtu výběrových bodů (v našem případě 3).



Výsledek

Upozornění:


Toto pole musíte nastavit ještě před výběrem proměnné!

V tomto poli stanovíte, jakého typu bude proměnná (integer pro Index, string pro Text) a co bude jejím obsahem až uživatel provede výběr z daných možností.

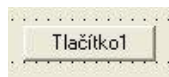
Index
Text

- **Index:** Obsahem proměnné typu integer bude celé číslo vzestupně od nuly v pořadí seznamu položek (tj. 0 u 1.položky, 1 u 2.položky atd.)
- **Text:** Obsahem proměnné typu string bude text popisu u vybraného bodu.

Tlačítko


 Slouží uživateli masky pro vykonání nějaké akce. Jedná se obvykle o spuštění externího programu, funkce z knihovny dll, nastavení globální proměnné pro další běh workflow, atd. Je možné jím nahradit i standardní tlačítka, protože lze nastavit i logickou hodnotu výsledku zpracování masky.

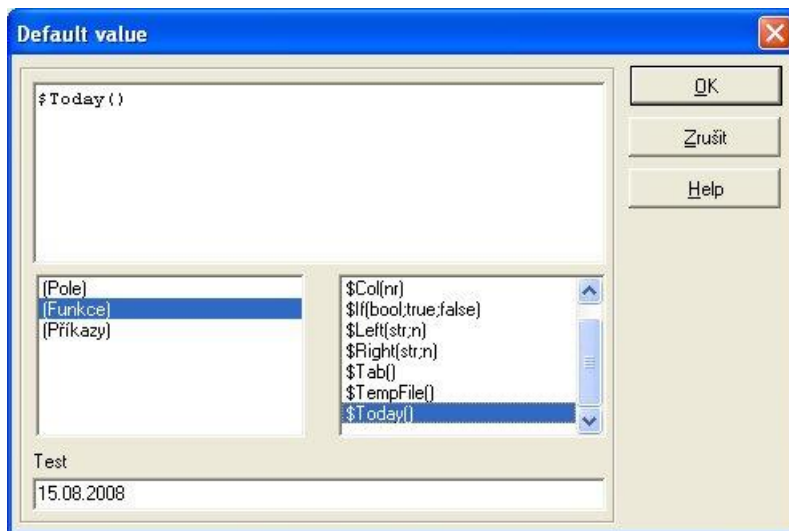
Vlastnosti	
Text	Tlačítko1
Při stisknutí	
Hot Key	Žádný
Sloupec	16
Řádka	10
Šířka	2
Výška	1
Zavřít dialog	Ne
Výsledek dialogu	Ne



Při stisknutí

Tato vlastnost bude určovat akci vykonávanou při stisku tlačítka.

Klepněte do pole **Při stisknutí** a pak klepněte na ikonu  na konci pole. Otevře se dialog



K dispozici máte objekty definované v masce, tyto funkce a příkazy:

- **\$AbortMessage(messagetext; type; abort):** Při stisknutí tlačítka vyvolá další okno s textem "messagetext" a po potvrzení **Ano** lze ve skriptu volat další funkci.
- **\$Col(nr):** Údaje o sloupci pro umístění polí
- **\$If(bool;true;false):** Funkce KDYŽ ve vztahu k jednotlivým vstupním polím
- **\$Left(str;n):** Udává, kolik znaků (n) vstupního pole (str) zleva se má zadat
- **\$Right(str;n):** Udává, kolik znaků (n) vstupního pole (str) zprava se má zadat

- **\$Tab()**: Vloží středník k oddělení jednotlivých polí
- **\$TempFile()**: Platné jen při volání programu pomocí EXEC
- **\$Today()**: Vrací aktuální datum

Příkazy:

- **\$clear()**: Vymaže hodnoty proměnných ve všech objektech masky
- **\$dllcall("dll"; "function"; param)**: Volá funkci z knihovny dll
- **\$exec(pgm; param)**: Volá externí program
- **\$setVar("variableName"; "value")**: Nastaví globální proměnnou na zadanou hodnotu

Poklepáním vyberte v pravém dolním okně komponentu, kterou potřebujete pro volání funkce nebo příkazu. Tím ji umístíte do horního okna. Zde upravte syntaxi na skutečné hodnoty.

V dolním poli TEST se zobrazí okamžitá hodnota funkce nebo příkazu.

Příklady:

- Mazání všech polí v masce

```
• $clear()
```

- Nastavení hodnoty proměnné

```
• $setVar("JakDal"; "zavrit")
```

Potvrzením **OK** nastavení vlastnosti uložíte.

V praxi se nejčastěji používají příkazy pro nastavení hodnoty globální proměnné, volání externího programu a volání funkce z knihovny dll.

- **HotKey**

Tato vlastnost umožňuje definovat klávesu, která bude provádět stejnou činnost jako tlačítko. K dispozici jsou funkční klávesy, klávesa <Enter> a <Esc>.

- **Zavřít dialog**

Nastavením této vlastnosti určíte, zda se maska po stisku tlačítka a vykonání nastavených činností uzavře nebo zůstane otevřená.

Doporučení:

Z důvodu uložení hodnot proměnných a dalších činností ve skriptu je vhodné nastavit tuto vlastnost na **Ano** a pokud má maska zůstat otevřená, znovu ji ze skriptu zavolat.

- **Výsledek dialogu**

Zde nastavujete výslednou logickou hodnotu masky při nastavení vlastnosti **Zavřít dialog** na **Ano**. Lze využít pro náhradu standardních tlačítek vlastními tlačítky. Výsledná hodnota masky může být použita jako výsledná logická hodnota celého skriptu, čímž se pak řídí další průběh workflow.

Např.

```
result:=idx("maska_udaje")
```

Grid

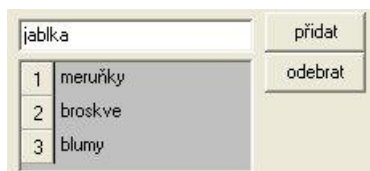
Slouží pro práci se všemi typy proměnných typu **list** (výraz z angličtiny znamená mřížka, rastr).

Vlastnosti	
Jméno	Grid1
Proměnná	
Kontrola	Žádný
Min. řádek	0
Max. řádek	0
Std. hodnota	
Hodnoty	Žádné
Sloupec	38
Řádka	3
Šířka	11
Výška	3
Jen pro čtení	Ne
Číslovat řádky	Ano
Barva	Standard



Objekt zobrazí položky proměnné list pod sebou na rádcích. Změny se dají provádět v případě, že vlastnost **Jen pro čtení** je nastavena na **Ne**. Nelze ale přímo v masce přidat položku (s výjimkou, kdy je proměnná před otevřením masky prázdná, pak lze v masce přidat pouze jednu položku, která se uloží do proměnné).

Obvyklé je použití tohoto objektu pouze pro zobrazení obsahu proměnné (pak se doporučuje použít šedou barvu objektu) a editaci a přidávání položek se řeší ve skriptu např. použitím tlačítek:

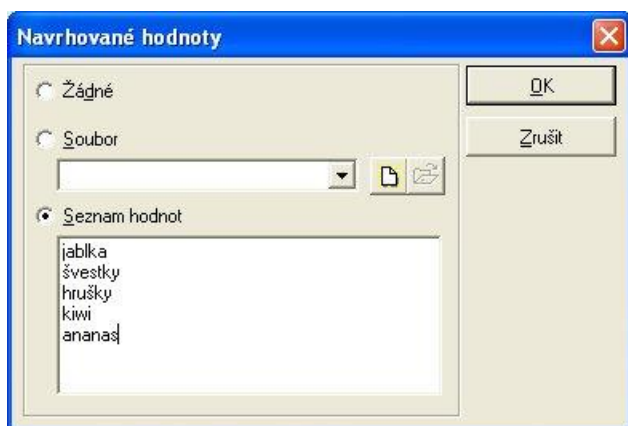


- **Min.řádek, Max. řádek**


Tyto vlastnosti se netýkají rozměrů objektu, ale počtu položek v proměnné typu list. Proto pokud neznáte předem tento údaj, ponechte hodnoty na nule. Použijete-li tyto vlastnosti, pak hodnota Min. musí být menší nebo rovna a Max. větší nebo rovna počtu položek, jinak kontrola objektu hlásí chybu a nelze pokračovat v práci s maskou, pouze ji opustit bez uložení údajů.


- **Hodnoty**

Grid obsahuje vlastnost **Hodnoty** stejně jako objekt **Rozbalovací seznam**. Klepnutím do pole **Hodnoty** a stiskem ikony  na konci pole se otevře dialog:



Pro navrhované hodnoty jsou tyto možnosti:

- **Žádné.** Pak není rozbalovací seznam hodnot použit.
- **Soubor:** V seznamu vyberte textový soubor, ve kterém jsou položky vyplněny na rádcích. Pokud soubor neexistuje, je možné jej vytvořit po stisku . Otevře se standardní okno pro uložení souboru

a pak okno pro zápis hodnot (textový editor). Vybraný soubor lze otevřít ikonou  a provádět změny.

- **Upozornění:** Pokud jste nevybrali složku, kde je uložena Serie M/...(obvykle m\bin), musíte ručně

zadat cestu k .txt souboru do řádku pro soubor před jeho jméno:



- **Seznam hodnot:** Hodnoty lze přímo vyplnit, pro odřádkování stisknete klávesu <Enter>.

Rozbalovací seznam, který jste vytvořili pomocí druhé nebo třetí možnosti, bude použit pro výběr hodnoty pro každou položku proměnné. Tzn. klepnutím na kteroukoliv položku proměnné se zobrazí pole se šipkou. Klepnutím na šipku se rozbalí seznam hodnot, ze kterého uživatel provede výběr.

Je nutné počítat s tím, že už jen klepnutím uživatele na položku proměnné nabude tato položka hodnoty první hodnoty v seznamu. Tento stav nelze již nijak vrátit na původní hodnotu. Navíc první položka proměnné je přemazána vždy, protože na její pozici se objeví rozbalovací seznam již při otevření masky.



Doporučení:

Použití rozbalovacího seznamu pro editaci proměnných typu list není příliš vhodné a doporučuje se pouze v odůvodněných případech.

Jen pro čtení

Pro jakoukoli editaci objektu ať už přímým zápisem nebo přes rozbalovací seznam hodnot je třeba nastavit tuto vlastnost na **Ne**.

Číslovat řádky

Pro lepší přehlednost je možné zapnout grafické znázornění čísel řádků s položkami proměnné.

Upozornění:

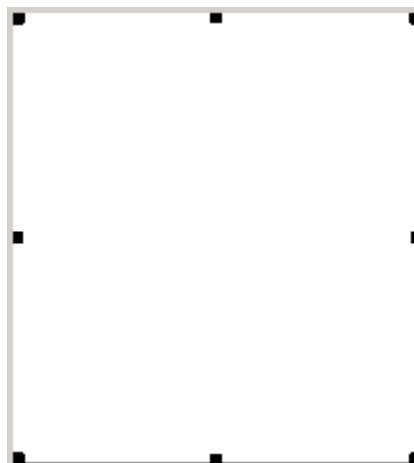
Tato čísla jsou o 1 zvýšena oproti indexu jednotlivých položek proměnné typu list, který se používá ve skriptu (začíná od 0)

Webový prohlížeč



Lze použít k zobrazení obrázků, dokumentů PDF a lokálních HTML souborů

Vlastnosti	
Jméno	WebBrowser1
Proměnná	
Sloupec	21
Řádka	3
Šířka	29
Výška	9
Typ rámečku	Zvýšený
Šířka rámečku	1



Primární určení tohoto objektu je zobrazování PDF dokumentů a obrázků (např. z příloh procesu). Vedlejší funkcí je i zobrazování obrázků (POZOR! V některých případech je nutné doinstalovat příslušný plugin do Vašeho Internet Exploreru.

- **Typ rámečku**

Tato volba nastavuje různé podoby rámečku podle jeho efektu.

- **Šířka rámečku**

Určuje šířku rámečku (maximální hodnota je omezena)