

# Grafické prostředí Red/View

## Table of Contents

1. Záměry a cíle .....	2
2. Piškoty .....	2
3. Strom piškotů .....	2
4. Typy piškotů .....	3
4.1. Base .....	3
4.2. Text .....	3
4.3. Button .....	4
4.4. Toggle .....	4
4.5. Check .....	5
4.6. Radio .....	5
4.7. Field .....	6
4.8. Area .....	6
4.9. Text-list .....	7
4.10. Drop-list .....	8
4.11. Drop-down .....	8
4.12. Calendar .....	9
4.13. Progress .....	9
4.14. Slider .....	9
4.15. Camera .....	10
4.16. Panel .....	10
4.17. Tab-panel .....	10
4.18. Window .....	11
4.19. Screen .....	12
4.20. Group-box .....	12
5. Životní cyklus piškotu .....	12
6. Aspekty .....	13
6.1. Objekt font .....	14
6.2. Objekt para .....	15
6.3. Aspekt options .....	16
6.4. Aspekt state .....	16
7. Funkce SHOW .....	17
8. Aktualizace zobrazení .....	17
9. Dvojsměrné propojení .....	17
10. Události .....	18
10.1. Názvy událostí .....	18
10.2. Datový typ Event! .....	19

10.3. Aktéři .....	21
10.4. Tok událostí .....	22
10.5. Globální ovladače událostí .....	23
10.5.1. insert-event-func .....	23
10.5.2. remove-event-func .....	23
11. Objekt system/view .....	24
12. Připojení komponenty View .....	24
13. Extra funkce .....	25

# 1. Záměry a cíle

Systém View (View engine) je grafický systém jazyka Red. Jeho cíle lze shrnout do těchto bodů:

- Minimální, datově orientované API
- Virtuální strom objektů (strom piškotů) jako programovací rozhraní
- Okamžitá (realtime) nebo odložená synchronizace mezi virtuálním stromem a zobrazovacím systémem
- Vytvoření jednoduše podporované dvoucestné vazby (two-way binding)
- Schopnost mít různé backendy pro různé platformy
- Podpora OS a uživatelských sad widgetů
- Malé provozní nároky

Virtuální strom se skládá z piškotů. Každý piškot koresponduje s příslušnou zobrazenou grafickou komponentou dvojcestným propojením.

# 2. Piškoty

Piškoty (tvář neboli face - od slova interface), jsou základní stavební kameny grafického rozhraní. Interní součástí každého piškotu je sada 23 aspektů (facets), jimiž jsou určovány jednotlivé parametry piškotu. Těmito parametry je určován nejen vzhled piškotů, ale i jejich odezva na různé události a souvislost s jinými piškoty.

Nový piškot (face) se vytvoří klonováním prototypu **face!** a zadáním alespoň platného názvu **typu**.

```
button: make face! [type: 'button']
```

Dalšími deklarovanými prototypy v systému View jsou *font!*, *para!*, *reactor!* a *deep-reactor!*. Datovým typem prototypů je *object!*.

# 3. Strom piškotů

Piškoty jsou organizovány do stromu, který vyjadřuje hierarchii jejich vztahů v zobrazení. Vztahy u

stromu jsou odvozeny z:

- facetu **pane**: výpis jednoho či více dětí piškotu v bloku.
- facetu **parent**: odkaz na rodičovský piškot.

Pořadí piškotů v bloku **pane** je důležité - odvozuje se z něj zetové uspořádání grafických objektů (piškot v čele seznamu **pane** je zobrazen jako poslední, piškot na konci seznamu je zobrazen nad všemi ostatními).

Kořenem stromu piškotů je piškot **screen**. Piškot **screen** může zobrazit pouze piškoty window z bloku ve svém poli **pane**.

Aby mohl být kterýkoli piškot zobrazen na obrazovce, musí být spojen s piškotem screen přímo (pro okna) nebo nepřímo (přes piškot **window** pro jiné typy piškotů).

[Face tree] | *face-tree.png*

## 4. Typy piškotů

### 4.1. Base

Piškot typu **base** je nejzákladnějším a nejvšestrannějším typem piškotů. Implicitně zobrazí pouze pozadí barvy **128.128.128**.

Facet	Popis
<b>type</b>	<b>base</b>
<b>image</b>	Lze určit hodnotu typu <b>image!</b> , kanál alfa je podporován.
<b>color</b>	Lze určit barvu pozadí, kanál alfa je podporován.
<b>text</b>	Uvnitř piškotu lze zobrazit libovolný text.
<b>draw</b>	Transparentnost je plně podporována pro primitiva systému Draw.

#### NOTE

- Úplná skladba následujících facetů je podporována a zobrazována v uvedeném pořadí: **color**, **image**, **text**, **draw**.
- Průhlednost lze zařídit u facetů **color**, **image**, **text** a **draw** určením hodnoty kanálu alfa v entici **R.G.B.A**, kde **A = 0** indikuje úplnou neprůhlednost a **A = 255** indikuje úplnou průhlednost.

*Tento typ piškotu může být použit pro zavedení libovolné grafické komponenty.*

### 4.2. Text

Typ **text** tvoří statický popisek.

Facet	Popis
<b>type</b>	'text
<b>text</b>	Text popisku.
<b>data</b>	Hodnota, zobrazená jako text.
<b>options</b>	Podporovaná pole: default.

Facet **data** je synchronizován v reálném čase s facetem **text** s použitím následujících pravidel:

- při změně facetu **text** je facet **data** nastaven na načítanou (**load**) hodnotu **text** nebo **none** nebo na **options/default**, je-li určeno
- při změně facetu **data** je facet **text** nastaven na utvářenou (**form**) hodnotu facetu **data**

Facet **options** přijímá následující vlastnosti:

- **default**: může být nastaven na libovolnou hodnotu, bude použit facetem **data**, vrátí-li konverze facetu **text** hodnotu **none**, stejně jako u nenačítatelných řetězců.

## 4.3. Button

Tento typ piškotu představuje jednoduché tlačítko.

Facet	Popis
<b>type</b>	'button
<b>text</b>	Popisek tlačítka.
<b>image</b>	Uvnitř tlačítka se zobrazí obrázek. Lze kombinovat s textem..

Typ události	Ovladač	Popis
<b>click</b>	on-click	Spustí se při poklepu na tlačítko.

## 4.4. Toggle

Tento typ reprezentuje tlačítko, které si zachovává svůj stav i po stisknutí. This type represents a button that retains its state after being pushed.

Facet	Description
<b>type</b>	'toggle
<b>text</b>	Text popisku pro toggle.
<b>para</b>	Řídí svislé a vodorovné přiřazení textu.
<b>data</b>	true: toggled; false: untoggled (default).
<b>image</b>	Uvnitř přepínacího tlačítka se zobrazí obrázek. Lze kombinovat s textem.

Typ události	Ovladač	Popis
change	on-change	Spustí se při změně stavu přepínače aktivitou uživatele.

## 4.5. Check

Tento typ představuje zatržítko s volitelným popiskem na levé či pravé straně.

Facet	Popis
type	'check
text	Text popisku.
para	Pole <b>align</b> určuje, zda je text zobrazen na levé nebo pravé straně.
data	<b>true</b> : zatrženo, <b>false</b> : nezatrženo; <b>none</b> : nezatrženo pro dvou-stavové zatržítko, neurčeno pro 3-stavové zatržítko (default).
flags	Zapnout tří-stavové zatržítko (word!).

Podporované flagy:

- **tri-state**: umožňuje třetí, neurčený stav, který je reprezentován hodnotou **none** aspektu **data**.

Event type	Ovladač	Popis
change	on-change	Spustí se změnou stavu zatržení uživatelem.

## 4.6. Radio

Tento typ představuje výběrové tlačítko s popiskem na levé či pravé straně. Ve skupině tlačítek na jednom panelu může být zvoleno pouze jedno tlačítko.

Facet	Popis
type	'radio
text	Text popisku.
para	Pole <b>align</b> určuje, zda je text zobrazen na levé ( <b>left</b> ) nebo pravé ( <b>right</b> ) straně.
data	<b>true</b> : vybráno, <b>false</b> : nevybráno (implicitní).

Event type	Ovladač	Popis
change	on-change	Spustí se změnou stavu po akci uživatele.

## 4.7. Field

Tento typ představuje jednořádkové vstupní pole.

Facet	Popis
<b>type</b>	'field'
<b>text</b>	Vkládaný text, hodnota read/write.
<b>data</b>	Hodnota, zobrazená jako text.
<b>selected</b>	Vybraný text (pair! none!).
<b>options</b>	Podporovaná pole: default.
<b>flags</b>	Zapnout či vypnout některé speciální vlastnosti polí (block!).

Vybraný (**selected**) piškot řídí zvýraznění textu (read/write). Hodnota typu **pair!** označuje první a poslední vybraný znak. Hodnota **none** indikuje, že v poli není vybrán žádný text.

### Podporované flagy:

- **no-border**: odstraní dekoraci okraje, vytvořenou výchozím frameworkem GUI.
- **password**: místo zadávaných znaků se zobrazují (\*).

Facet **data** je synchronizován v reálném čase s facetem **text** s použitím následujících pravidel:

- při změně facetu **text** je facet **data** nastaven na načítanou hodnotu **text** nebo **none** nebo na **options/default**, je-li určeno
- při změně facetu **data** je facet **text** nastaven na utvářenou hodnotu **data**

Facet **options** přijímá následující vlastnosti:

- **default**: může být nastaven na libovolnou hodnotu, bude použit facetem **data**, vrátí-li konverze facetu **text** hodnotu **none**, stejně jako u nenačítatelných řetězců.

Typ události	Ovladač	Popis
<b>enter</b>	<b>on-enter</b>	Vyskytne se při každém stisku klávesnice Enter v poli.
<b>change</b>	<b>on-change</b>	Vyskytne se při každém vložení hodnoty do pole.
<b>select</b>	<b>on-select</b>	Vyskytne se při každém výběru textu myší nebo klávesnicí.
<b>key</b>	<b>on-key</b>	Vyskytne se při každém stisku libovolné klávesnice v poli.

## 4.8. Area

Tento typ představuje víceřádkové vstupní pole.

Facet	Popis
<b>type</b>	' <b>area</b>
<b>text</b>	Zadávaný text, hodnota read/write.
<b>selected</b>	Vybraný text (pair! none!)
<b>flags</b>	Zapnout či vypnout některé speciální vlastnosti (block!).

Aspekt **selected** řídí zvýraznění textu (read/write). Hodnota typu **pair!** označuje první a poslední vybraný znak. Hodnota **none** indikuje, že v poli není vybrán žádný text.

#### Podporované flagy:

- **no-border**: odstraní dekoraci okraje, vytvořenou výchozím frameworkem GUI.

#### NOTE

- Může se objevit svislá posuvná lišta, jestliže oblast (area) nepojme všechny řádky textu (v budoucnu bude možné kontrolovat volbou **flags**).

Event type	Ovladač	Popis
<b>change</b>	<b>on-change</b>	Vyskytne se při každém vložení hodnoty do oblasti.
<b>select</b>	<b>on-select</b>	Vyskytne se po každém výběru textu myší nebo klávesnicí.
<b>key</b>	<b>on-key</b>	Vyskytne se v dané oblasti při každém stisku klávesy.

## 4.9. Text-list

Tento typ představuje svislý seznam textových řetězců, zobrazený v pevném rámečku. Automaticky se objeví vertikální posuvník, jestliže se všechny řádky nevejdou do rámečku.

Facet	Popis
<b>type</b>	' <b>text-list</b>
<b>data</b>	Seznam zobrazovaných řetězců (block! hash!).
<b>selected</b>	Index vybraného řetězce nebo hodnota none, není-li žádný vybrán (read/write).

Event type	Ovladač	Popis
<b>select</b>	<b>on-select</b>	Vyskytne se při výběru položky seznamu. Facet <b>selected</b> odkazuje na index starší vybrané položky.
<b>change</b>	<b>on-change</b>	Vyskytne se po události <b>select</b> . Facet <b>selected</b> odkazuje na index nově vybrané položky.

#### NOTE

- počet viditelných položek zatím nemůže být určen uživatelem.

## 4.10. Drop-list

Tento typ představuje vertikální seznam textových řetězců, zobrazený ve svinutelném rámečku. Automaticky se objeví svislý posuvník, přesahuje-li obsah velikost rámečku.

Facet	Popis
<b>type</b>	'drop-list
<b>data</b>	Seznam zobrazovaných řetězců (block! hash!).
<b>selected</b>	Index vybraného řetězce nebo hodnota none, není-li žádný vybrán (read/write).

Facet **data** přijímá libovolné hodnoty, avšak pouze hodnoty typu string jsou zařazeny do seznamu a zobrazeny. Neřetězcové hodnoty lze použít pro vytvoření asociativní řady (array) v níž řetězce slouží jako klíče. Facet **selected** je celočíselný index, počínající jedničkou, indikující pozici vybraného řetězce v seznamu, nikoliv ve facetu **data**.

### Podporované flagy:

#### DOSUD

- **scrollable**: Ručně povolit vertikální scroll-bar.

Event type	Ovladač	Popis
<b>select</b>	on-select	Vyskytne se při výběru položky v seznamu. Facet <b>selected</b> odkazuje na index starší vybrané položky.
<b>change</b>	on-change	Vyskytne se po události <b>select</b> . Facet <b>selected</b> odkazuje na index nově vybrané položky.

#### NOTE

- počet viditelných položek zatím nemůže být určen uživatelem.

## 4.11. Drop-down

Tento typ představuje editovatelné pole s vertikálním seznamem zobrazeným ve svinovacím rámečku. Automaticky se objeví svislý posuvník, překračuje-li obsah velikost rámečku.

Facet	Popis
<b>type</b>	'drop-down
<b>data</b>	Seznam zobrazovaných řetězců (block! hash!).
<b>selected</b>	Index vybraného řetězce nebo hodnota none, není-li žádný vybrán (read/write).

Facet **data** přijímá libovolné hodnoty avšak pouze hodnoty typu string jsou zařazeny do seznamu a zobrazeny. Neřetězcové hodnoty lze použít pro vytvoření asociativní řady, v níž řetězce slouží jako klíče. Facet **selected** je celočíselný index, počínající jedničkou, indikující pozici vybraného řetězce v



seznamu.

#### Podporované flagy:

- **scrollable**: Ručně povolit vertikální scroll-bar.

Event type	Ovladač	Popis
<b>select</b>	<b>on-select</b>	Vyskytne se při výběru položky v seznamu. Facet <b>selected</b> odkazuje na index starší vybrané položky.
<b>change</b>	<b>on-change</b>	Vyskytne se po události <b>select</b> . Facet <b>selected</b> odkazuje na index nově vybrané položky.

#### NOTE

- počet viditelných položek zatím nemůže být určen uživatelem

## 4.12. Calendar

Tento typ představuje měsíční Gregoriánský kalendář pro období od 1-Jan-1601 do 31-Dec-9999.

Facet	Description
<b>type</b>	'calendar'
<b>data</b>	hodnota <b>date!</b> , která představuje vybraný den.

Event type	Handler	Description
<b>change</b>	<b>on-change</b>	Vyskytne se při výběru data v kalendáři.

#### NOTE

- Implicitně je aspekt **data** nastaven na "dnes".
- hodnota **date!** nad nebo pod zadanými mezemi v kalendáři vybere minimální nebo maximální podporované datum.

## 4.13. Progress

Tento typ představuje horizontální nebo vertikální průběhovou lištu.

Facet	Popis
<b>type</b>	'progress'
<b>data</b>	Hodnota, představující průběh postupu (hodnota percent! nebo float!).

#### NOTE

- Je-li pro zobrazení průběhu ve facetu **data** použita hodnota typu float, musí být v rozsahu od 0.0 do 1.0.

## 4.14. Slider

Tento typ představuje kurzor, posuvný podél horizontální nebo vertikální osy.

Facet	Popis
type	'slider
data	Hodnota, představující pozici kurzoru (percent! nebo float!).

#### NOTE

- Je-li pro zobrazení průběhu ve facetu **data** použita hodnota typu float, musí být v rozsahu od 0.0 do 1.0.

## 4.15. Camera

Tento typ se používá k zobrazení "video camera feed".

Facet	Popis
type	'camera
data	Seznam názvů kamer jako blok řetězců.
selected	Vybraný název kamery ze seznamu <b>data</b> , s použitím celočíselného indexu. Nastaveno-li na <b>none</b> , není "camera feed" povolen.

#### NOTE

- Facet **data** je implicitně nastaven na **none**. Během prvního volání je připraven seznam kamer k zobrazení v piškotu kamery.
- Je možné zachytit obsah piškotu kamery s použitím **to-image** na piškotu.

## 4.16. Panel

Panel je kontejner pro další piškoty.

Facet	Popis
type	'panel
pane	Blok dětských piškotů. Pořadí v bloku určuje jejich zetové pořadí při zobrazení.

#### NOTE

- Souřadnice pro **offset** (odsazení) dětí jsou vztaženy k levému hornímu rohu rodiče.
- Piškoty dětí jsou vloženy do rámečku (frame) panelu.

## 4.17. Tab-panel

Tab-panel je seznam karet (panelů), z nichž pouze jedna může být v daném okamžiku viditelná. Seznam názvů těchto karet je zobrazen jako "tabs" a používá se k přepínání mezi kartami.

Facet	Popis
type	'tab-panel

Facet	Popis
<b>data</b>	Blok názvů karet (hodnoty string).
<b>pane</b>	Seznam panelů, odpovídající seznamu karet (block!).
<b>selected</b>	Index vybraného panelu nebo hodnota none (integer!, read/write).

Event type	Ovladač	Popis
<b>change</b>	on-change	Vyskytne se při výběru nové karty. Facet <b>event/picked</b> obsahuje index nově vybrané karty. Vlastnost <b>selected</b> je aktualizována ihned po této události.

#### NOTE

- K řádnému zobrazení panelu s kartami je zapotřebí aby facety **data** i **pane** byly řádně vyplněny.
- Obsahuje-li facet **pane** víc panelů než zadaných karet, budou ignorovány.
- Při přidávání či odebrání karty musí být odpovídající panel rovněž přidán či odebrán v/ze seznamu **pane**.

## 4.18. Window

Reprezentuje okno, zobrazené na monitoru počítače.

Facet	Popis
<b>type</b>	' <b>window</b> '
<b>text</b>	Název okna (string!).
<b>offset</b>	Odsazení okna od levého horního rohu obrazovky, nezahrnující v to dekoraci rámečku okna (pair!).
<b>size</b>	Velikost okna bez dekorací rámečku (pair!).
<b>flags</b>	Zapíná či vypíná některé speciální vlastnosti okna (block!).
<b>menu</b>	Zobrazí nabídkovou lištu v okně (block!).
<b>pane</b>	Seznam piškotů, zobrazených uvnitř okna (block!).
<b>selected</b>	Výběr piškotu, jež se stane nositelem zaměření (object!).

#### Podporované flagy:

- **modal**: učiní okno modálním, zneplatňující všechna předtím otevřená okna
- **resize**: umožní změnu velikosti okna (implicitní je pevná velikost, nikoliv měnitelná).
- **no-title**: nezobrazovat název okna
- **no-border**: odebrat dekorace rámečku okna
- **no-min**: odebrat tlačítko 'minimalizovat' z přesuvné lišty okna
- **no-max**: odebrat tlačítko 'maximalizovat' z přesuvné lišty okna

- **no-buttons**: odebrat všechna tlačítka z přesuvné lišty okna
- **popup**: alternativní menší dekorace rámečku (jen u Windows).

#### NOTE

- Použití klíčového slova **popup** na počátku bloku se specifikacemi menu vynutí implicitně kontextuální menu v okně namísto nabídkové lišty.

## 4.19. Screen

Představuje grafickou zobrazovací jednotku, připojenou k počítači (obvykle monitor).

Facet	Popis
<b>type</b>	'screen'
<b>size</b>	Velikost zobrazení monitoru v pixelech. Nastaveno při startu prostředím View engine (pair!).
<b>pane</b>	Seznam zobrazovaných oken na monitoru (block!).

Všechny zobrazené piškoty oken musí být dětmi piškotu screen.

## 4.20. Group-box

Group-box je kontejner pro jiné piškoty s viditelným rámečkem. *Tento dočasný formát bude odebrán až bude k dispozici podpora pro facet `edge`.*

Facet	Popis
<b>type</b>	'group-box'
<b>pane</b>	Blok s dětskými piškoty. Pořadí v bloku určuje zetové pořadí (z-order) při zobrazení.

#### NOTE

- Souřadnice odsazení (offset) dětí jsou relativní k levému hornímu rohu group-boxu.
- Dětské piškoty jsou připojeny do rámečku group-boxu.

## 5. Životní cyklus piškotu

1. Piškot (face) je vytvořen pomocí prototypu **face!**.
2. Je vložen do stromu piškotů spojeného s piškotem screen.
3. Je zobrazen na monitoru příkazem **show**.
  - a. v tomto okamžiku jsou přiděleny systémové zdroje
  - b. blok **face/state** je nastaven.
4. Odebráním piškotu z aspektu pane je odebrán ze zobrazení.
5. Vymizí-li odkazy na piškot, postará se garbage collector o uvolnění obsazených zdrojů.

## NOTE

- U aplikací, hladových na zdroje, lze použít funkci **free** pro manuální uvolnění systémových zdrojů.

## 6. Aspekty

Pole piškotu zvané **facet** (aspekt) určuje jeho různé vlastnosti.

Seznam vytvořených aspektů:

Facet	Datatype	Povinné?	Použití	Popis
<b>type</b>	word!	yes	all	Interně používané pro ověření, že je objekt strukturován jako piškot.
<b>offset</b>	pair!	yes	all	Odsazení od piškotu rodiče vlevo nahoře.
<b>size</b>	pair!	yes	all	Velikost piškotu.
<b>text</b>	string!	no	all	Text, zobrazený v piškotu. Atributy textu jsou určeny facetu <b>font</b> a <b>para</b> .
<b>image</b>	image!	no	some	Obraz na pozadí piškotu.
<b>color</b>	tuple!	no	some	Barva pozadí piškotu ve formátu R.G.B nebo R.G.B.A.
<b>menu</b>	block!	no	all	Lišta nabídky (menu) nebo kontextuálního menu.
<b>data</b>	any-type!	no	all	Data o stavu piškotu a další relevantní informace.
<b>enabled?</b>	logic!	yes	all	Umožnit nebo znemožnit události vstupu v piškotu.
<b>visible?</b>	logic!	yes	all	Zobrazit či skrýt piškot.
<b>selected?</b>	integer!, pair!, object!	no	some	Index aktuálně vybraného elementu seznamu.
<b>flags</b>	block!, word!	no	some	Seznam speciálních klíčových slov, měnících zobrazení nebo chování piškotu.
<b>options</b>	block!	no	some	Další vlastnosti piškotu ve formátu [name: value].
<b>parent</b>	object!	no	all	Odkaz na rodičovský piškot (pokud existuje).
<b>pane</b>	block!	no	some	Seznam dětí (child faces), zobrazených uvnitř piškotu.
<b>state</b>	block!	no	all	Informace o interním stavu piškotu (používané pouze prostředím View engine).

Facet	Datatype	Povinné?	Použití	Popis
<b>rate</b>	integer! time!	no	all	Časovač piškotu. Periodicky spouští událost <b>time</b> . Celé číslo udává frekvenci v Hz, <b>time!</b> nastavuje interval a <b>none</b> činnost končí.
<b>edge</b>	object!	no	all	<i>(vyhrazeno pro budoucí použití)</i>
<b>para</b>	object!	no	all	Nastavení úpravy textu v odstavci - počátek, okraje, odsazení, tabulátor, rolování textu a další.
<b>font</b>	object!	no	all	Nastavení vlastností fontu (style, velikost, barva, zarovnání, a jiné) pro piškot.
<b>actors</b>	object!	no	all	Uživatelsky vytvořené ovladače událostí.
<b>extra</b>	any-type!	no	all	Volitelná uživatelská data připojená k piškotu.
<b>draw</b>	block!	no	all	Seznam příkazů Draw, jež mají být provedeny v piškotu.

Seznam globálně použitelných flagů pro facet **flags**:

Flag	Popis
<b>all-over</b>	Poslat všechny události <b>over</b> do piškotu.

Ostatní specifické flagy piškotů jsou dokumentovány v příslušných sekcích.

#### NOTE

- Nepovinné faceti mohou být nastaveny na **none**.
- Velikosti **offset** a **size** se určují v pixelech;
- Hodnoty pro **offset** a **size** lze někdy před zobrazením nastavit na **none**. Systém View se postará o jejich nastavení (jako u panelů v typu **tab-panel**).
- Pořadí zobrazení (odzadu dopředu): color, image, text, draw.

## 6.1. Objekt font

Objekt **font** je instance prototypu **font!**. Instance prototypu se vytvoří příkazem:

```
font: make font! [field: value ...]
```

Na jeden objekt font mohou odkazovat aspekty **font** více piškotů, což umožňuje ovládat vlastnosti fontu z jednoho místa pro celou skupinu piškotů.

Při deklaraci se název atributu uvádí za pomlčkou, například **font-name**.

Field	Datatype	Mandatory?	Popis
name	string!	no	Platný název fontu, instalovaného v OS.

Field	Datatype	Mandatory?	Popis
size	integer!	no	Velikost fontu v bodech (points).
style	word!, block!	no	Styl nebo blok stylů.
angle	integer!	yes	Sklon textu ve stupních (default is 0).
color	tuple!	yes	Barva fontu ve formátu R.G.B nebo R.G.B.A.
anti-alias?	logic!, word!	no	Režim vyhlazení (active/inactive or special mode).
shadow	(reserved)	no	(reserved for future use)
state	block!	no	Informace o interním stavu piškotu (používané pouze prostředím View).
parent	block!	no	Interní odkaz na rodičovský piškot (používané pouze prostředím View).

#### NOTE

- Nepovinné facety mohou být nastaveny na hodnotu **none**.
- Pole **angle** zatím nepracuje jak má.
- V budoucnu se hodnoty všech polí mají stát nepovinné.

Nabízené styly fontů:

- **bold**
- **italic**
- **underline**
- **strike**

Nabízené režimy vyhlazení:

- active/inactive (**anti-alias?:** yes/no)
- ClearType mode (**anti-alias?:** 'ClearType')

## 6.2. Objekt para

Objekt **para** je instance prototypu **para!**. Instance prototypu se vytvoří příkazem:

```
para: make para! [field: value ...]
```

Na jeden objekt para mohou odkazovat aspekty **para** více piškotů, což umožňuje ovládat vlastnosti odstavce z jednoho místa pro celou skupinu piškotů.

Field	Datatype	Description
origin	(reserved)	(reserved for future use)

Field	Datatype	Description
padding	(reserved)	(reserved for future use)
scroll	(reserved)	(reserved for future use)
align	word!	Nastavení horizontálního přiřazení textu: <i>left, center, right</i> .
v-align	(reserved)	Nastavení vertikálního přiřazení textu: <i>top, middle, bottom</i> .
wrap?	logic!	Povolit zarovnání (wrapping) textu.
parent	block!	Interní odkaz na rodičovský piškot.

#### NOTE

- Všechna pole objektu para mohou být nastavena na hodnotu **none**.

## 6.3. Aspekt options

Aspekt options obsahuje volitelné parametry, kterými se zadává specifické chování piškotu:

Option	Popis
<b>drag-on</b>	Možné volby jsou: 'down', 'mid-down', 'alt-down', 'aux-down'. Používá se pro operace drag'n drop.
<b>offset</b>	Odsazení okna od levého horního rohu obrazovky v pixelech ve formátu <b>pair!</b> .
<b>size</b>	Velikost okna v pixelech ve formátu <b>pair!</b> .

## 6.4. Aspekt state

Následující informace je uváděna jen pro úplnost, při normálním používání by aspekt **state** neměl být používán. Lze jej ovšem použít, když uživatel volá OS API přímo nebo když je zapotřebí změnit chování View engine.

Position/Field	Popis
<b>1 (handle)</b>	Na operačním systému závislý ovladač pro grafický objekt ( <b>integer!</b> ).
<b>2 (changes)</b>	'Bit flags array' označení, který facet byl změněn od posledního volání fce <b>show</b> ( <b>integer!</b> ).
<b>3 (deferred)</b>	Seznam odložených změn od posledního volání funkce <b>show</b> , kdy aktualizace v reálném čase jsou vypnuty ( <b>block!</b> , <b>none!</b> ).
<b>4 (drag-offset)</b>	Uchovává výchozí pozici kurzoru myši při vstupu do režimu vlečení piškotu ( <b>pair!</b> , <b>none!</b> ).

#### NOTE

- Po zavolání **show** je pole **changes** nastaveno na 0 a pole bloku **deferred** je smazáno.
- Datový typ **handle!** bude použit pro nejasné ovladače OS.



## 7. Funkce SHOW

### Syntaxe

```
show <face>
```

```
<face>: klon prototypu face!
```

### Popis

Tato funkce se používá pro aktualizaci piškotu nebo seznamu piškotů na monitoru. Řádně zobrazen může být ten piškot, na který odkazuje strom piškotů, spojený s piškotem screen. Při prvním volání jsou přiřazeny systémové zdroje, je nastaven facet **state** a grafický komponent je zobrazen na monitoru. Následná volání budou na monitoru reflektovat jakoukoliv změnu piškotu. Je-li definován facet **pane**, bude funkce **show** rekurzivně aplikována i na piškoty dětí.

## 8. Aktualizace zobrazení

**View engine** má dva různé režimy pro aktualizaci zobrazení po úpravě stromu piškotů:

- Aktualizace v reálném čase: jakákoli změna piškotu je okamžitě zobrazena v monitoru.
- Odložená aktualizace: žádné změny nejsou přeneseny na monitor, dokud se pro piškot nebo pro jeho rodiče nezavolá funkce **show**.

Přechod mezi těmito dvěma režimy je řízen slovem **system/view/auto-sync?** : je-li nastaveno na **yes**, realizuje se aktualizace v reálném čase (implicitní nastavení), je-li nastaveno na **no**, View engine odloží všechny aktualizace.

Motivací pro implicitně nastavenou aktualizaci v reálném čase jsou:

- Jednodušší a kratší zdrojový kód, neboť není zapotřebí volat **show** po každé změně piškotu.
- Menší penzum k pamatování pro začátečníky.
- Postačující pro jednoduché či prototypové aplikace.
- Zjednodušuje experimentování z konzoly.

Odložená aktualizace realizuje mnoho změn na monitoru zároveň, což vede ke snížení zádrhelů a je rychlejší.

#### NOTE

- Prostředí Rebol/View podporuje pouze režim odložené aktualizace.

## 9. Dvojsměrné propojení

Piškoty se při svém propojení s řadami (series), použitými ve facetech, opírají o vlastnický systém Redu. Jakákoli změna v některém z facetu je piškotem detekována a vyřízena ve shodě s aktuálně nastaveným režimem aktualizace ( realtime or deferred).

Na druhé straně změny, provedené v zobrazených grafických objektech, jsou okamžitě reflektovány v odpovídajících facetech. Například, zápis do piškotu `field` reflektuje vstup do facetu `text` v živém režimu.

Toto dvojsměrné propojení zjednodušuje programování interakce s grafickými objekty bez potřeby jakéhokoli specifického API. Zcela postačí úprava facetů s použitím akcí pro řady (series).

Příklad:

```
view [  
  list: text-list data ["John" "Bob" "Alice"]  
  button "Add" [append list/data "Sue"]  
  button "Change" [lowercase list/data/1]  
]
```

## 10. Události

### 10.1. Názvy událostí

Jméno	Typ vstupu	Příčina
<b>down</b>	mouse	Left mouse button pressed.
<b>up</b>	mouse	Left mouse button released.
<b>mid-down</b>	mouse	Middle mouse button pressed.
<b>mid-up</b>	mouse	Middle mouse button released.
<b>alt-down</b>	mouse	Right mouse button pressed.
<b>alt-up</b>	mouse	Right mouse button released.
<b>aux-down</b>	mouse	Auxiliary mouse button pressed.
<b>aux-up</b>	mouse	Auxiliary mouse button released.
<b>drag-start</b>	mouse	A face dragging starts.
<b>drag</b>	mouse	A face is being dragged.
<b>drop</b>	mouse	A dragged face has been dropped.
<b>click</b>	mouse	Left mouse click (button widgets only).
<b>dbl-click</b>	mouse	Left mouse double-click.
<b>over</b>	mouse	Kurzor myši přechází přes piškot. Tato událost vzniká jednou, když kurzor vstupuje na piškot a jednou, když jej opouští. Pokud facet <code>flags</code> obsahuje flag <code>all-over</code> , jsou rovněž produkovány všechny mezilehlé události
<b>move</b>	mouse	A window has moved.
<b>resize</b>	mouse	A window has been resized.

Jméno	Typ vstupu	Příčina
<b>moving</b>	mouse	A window is being moved.
<b>resizing</b>	mouse	A window is being resized.
<b>wheel</b>	mouse	The mouse wheel is being moved.
<b>zoom</b>	touch	A zooming gesture (pinching) has been recognized.
<b>pan</b>	touch	A panning gesture (sweeping) has been recognized.
<b>rotate</b>	touch	A rotating gesture (sweeping) has been recognized.
<b>two-tap</b>	touch	A double tapping gesture has been recognized.
<b>press-tap</b>	touch	A press-and-tap gesture has been recognized.
<b>key-down</b>	keyboard	A key is pressed down.
<b>key</b>	keyboard	Byl zadán znak nebo stisknuta speciální klávesa (kromě control, shift a nabídkové klávesy).
<b>key-up</b>	keyboard	A pressed key is released.
<b>enter</b>	keyboard	Enter key is pressed down.
<b>focus</b>	any	A face just got the focus.
<b>unfocus</b>	any	A face just lost the focus.
<b>select</b>	any	A selection is made in a face with multiple choices.
<b>change</b>	any	Vyskytla se změna v piškotu, přijímající vstup uživatele (text nebo výběr ze seznamu).
<b>menu</b>	any	A menu entry is picked.
<b>close</b>	any	A window is closing.
<b>time</b>	timer	Proběhla prodleva, nastavená facetem <b>rate</b> piškotu.

#### NOTE

- Dotekové (touch) události nejsou realizovatelné ve Windows XP.
- Jedna či více *pohybových* událostí vždy předchází událost **move**.
- Jedna či více událostí *měnicích rozměr* vždy předchází událost **resize**.

## 10.2. Datový typ Event!

Hodnota události je opaktní objekt, obsahující všechny informace o dané události. K polím události se přistupuje pomocí cesty (path notation).

Field	Returned value
<b>type</b>	Typ události ( <b>word!</b> ).
<b>face</b>	Piškot, ve kterém došlo k události ( <b>object!</b> ).
<b>window</b>	Piškot okna, ve kterém došlo k události ( <b>object!</b> ).

Field	Returned value
<b>offset</b>	Odsazení kurzoru myši od piškotu při vzniku události ( <b>pair!</b> ). Pro události gest se vrací souřadnice středu.
<b>key</b>	Stisknutá klávesa ( <b>char!</b> , <b>word!</b> ).
<b>picked</b>	Nová položka, vybraná v piškotu ( <b>integer!</b> , <b>percent!</b> ). Pro událost <b>wheel</b> vrací počet kroků rotace. Pozitivní hodnota indikuje pootočení kolečka vpřed, od uživatele; negativní hodnota indikuje pootočení kolečka zpět, směrem k uživateli. Pro událost <b>menu</b> se vrací odpovídající ID nabídky. ( <b>word!</b> ) U zoomovacího gesta se vrací procentní hodnota představující relativní zvětšení/zmenšení. U jiných gest je tato hodnota prozatím závislá na systému (Windows: <b>allArguments</b> , pole z <b>GESTUREINFO</b> )
<b>flags</b>	Vrací seznam s jedním či více flagy (viz list níže) ( <b>block!</b> ).
<b>away?</b>	Vrací <b>true</b> , opouští-li kurzor hranice piškotu ( <b>logic!</b> ). Platí pouze při aktivní události <b>over</b> .
<b>down?</b>	Vrací true při stisknutí levého tlačítka myši ( <b>logic!</b> ).
<b>mid-down?</b>	Vrací true při stisknutí prostředního tlačítka myši ( <b>logic!</b> ).
<b>alt-down?</b>	Vrací true při stisknutí pravého tlačítka myši ( <b>logic!</b> ).
<b>ctrl?</b>	Vrací true při stisku klávesy CTRL ( <b>logic!</b> ).
<b>shift?</b>	Vrací true při stisku klávesy SHIFT ( <b>logic!</b> ).

Seznam možných flagů z **event/flags**:

- **away**
- **down**
- **mid-down**
- **alt-down**
- **aux-down**
- **control**
- **shift**

[NOTE] Všechna pole (kromě **type**) jsou pouze pro čtení. Nastavení pole **type** je používáno pouze interně prostředím View.

Zde je seznam speciálních kláves, vracených jako slova polem **event/key**:

- **page-up**
- **page-down**
- **end**
- **home**
- **left**

- `up`
- `right`
- `down`
- `insert`
- `delete`
- `F1`
- `F2`
- `F3`
- `F4`
- `F5`
- `F6`
- `F7`
- `F8`
- `F9`
- `F10`
- `F11`
- `F12`

Následující názvy kláves mohou být vráceny polem `event/key` pouze pro sdělení `key-down` a `key-up`:

- `left-control`
- `right-control`
- `left-shift`
- `right-shift`
- `left-menu`
- `right-menu`

## 10.3. Aktéři

Aktéři (actors) jsou ovládací funkce pro události `View engine`. Jsou definovány v objektu, vytvořeném volnou formou (nemají prototyp) a odkazuje na ně facet `actors`. Všichni aktéři mají stejný blok specifikací.

### Syntaxe

```
on-<event>: func [face [object!] event [event!]]
```

<event> : jakýkoliv platný název události (viz tabulka výše)

face : piškot, který je příjemcem události

event : hodnota události.

Kromě událostí GUI je také možné definovat aktéra **on-create**, který je volán při prvním zobrazení piškotu, ještě před tím, než jsou pro něj alokovány systémové zdroje. Na rozdíl od jiných aktérů má **on-create** pouze jeden argument a to **face**.

### Vratná hodnota

'stop : opustí smyčku událostí.

'done : zabrání události přetéci do jiného piškotu.

Ostatní vratné hodnoty nemají žádný účinek.

## 10.4. Tok událostí

Události jsou obvykle generovány pro určitou pozici na obrazovce a jsou přiřazeny k nejbližšímu čelnímu (front) piškotu. Událost ovšem cestuje od jednoho piškotu ke druhému v hierarchii předchůdců ve dvou směrech, obecně známých jako:

- **odchycení (capturing) události:** událost postupuje od piškotu **window** k čelnímu (front) piškotu, kde vznikla. Pro každý piškot je generována událost **detect** a volán odpovídající ovladač (handler), pokud existuje.
- **probublávání (bubbling) události:** událost postupuje od piškotu k rodičovskému oknu. Pro každý piškot je volán lokální ovladač události.

[Event flow] | *event-flow.png*

Typická cesta toku událostí:

1. Na tlačítku je generována událost **click**, zpracovávají se globální ovladače (viz další odstavec).
2. Začíná etapa odchycení události:
  - a. Nejprve se událost dostane k piškotu **window**, je volán jeho ovladač **on-detect**.
  - b. Poté se událost dostane k piškotu **panel**, je volán jeho ovladač **on-detect**.
  - c. Nakonec se událost dostane k piškotu **button**, je volán jeho ovladač **on-detect**.
3. Začíná etapa probublávání události:
  - a. Nejprve se událost dostane k piškotu **button**, je volán jeho ovladač **on-click**.
  - b. Poté se událost dostane k piškotu **panel**, je volán jeho ovladač **on-click**.
  - c. Nakonec se událost dostane k piškotu **window** a je volán jeho ovladač **on-click**.

#### NOTE

- Zrušení události se docílí vrácením slova **'done'** z jakéhokoli ovladače.
- Odchycení události není implicitně povoleno z kapacitních důvodů. Odchycení povolíte nastavením **system/view/capturing?: yes**.

## 10.5. Globální ovladače událostí

Před vstupem do cesty toku událostí je možné zajistit specifické předzpracování použitím takzvaných "globálních ovladačů událostí". Jsou k dispozici následující API pro jejich přidání a odebrání.

### 10.5.1. insert-event-func

#### Syntaxe

```
insert-event-func <handler>
```

<handler> : funkce ovladače nebo blok kódu pro předzpracování události

specifikace funkce ovladače:: func [face [object!] event [event!]]

#### Vratná hodnota

Nově přidaná funkce ovladače (function!).

#### Popis

Instaluje funkci globálního ovladače, která umí předzpracovat události před tím, než se dostanou k ovladačům piškotu. Pro každou událost jsou volány všechny globální ovladače, takže kód těla ovladače potřebuje být optimalizován na rychlost a využití paměti. Je-li jako argument poskytnut blok, je konvertován na funkci s použitím konstruktoru **function**.

Vratná hodnota funkce ovladače:

- **none** : událost může být zpracována jinými ovladači (**none!**).
- **'done'** : jiné globální ovladače jsou přeskočeny ale událost je šířena k dětským piškotům (**word!**).
- **'stop'** : opustit smyčku událostí (**word!**).

Vrací se odkaz na funkci ovladače a ten by měl být uložen, má-li být ovladač později odebrán.

### 10.5.2. remove-event-func

#### Syntaxe

```
remove-event-func <handler>
```

<handler> : předchozí instalovaná funkce ovladače události

## Popis

Zneplatní předchozí instalovaný globální ovladač události jeho odebráním z interního seznamu.

# 11. Objekt system/view

Word	Popis
<b>screens</b>	Seznam piškotů <b>screen</b> , reprezentujících připojená zobrazení.
<b>event-port</b>	<i>reserved for future use</i>
<b>metrics</b>	<i>reserved for future use</i>
<b>platform</b>	Nízko úroňový platformní kód prostředí View (includes backend code).
<b>VID</b>	Procesní kód VIDu.
<b>handlers</b>	Seznam globálních ovladačů událostí
<b>reactors</b>	Interní asociativní tabulka pro reaktivní piškoty a jejich akční bloky.
<b>evt-names</b>	Interní tabulka pro konverzi události na název aktéra.
<b>init</b>	Inicializační funkce z View engine; případně může být volána i uživatelem.
<b>awake</b>	Funkce vstupního bodu pro hlavní 'high-level' události.
<b>capturing?</b>	<b>yes</b> = umožňuje etapu odchycení události a generování událostí <b>detect</b> (implicitně <b>no</b> ).
<b>auto-sync?</b>	<b>yes</b> = aktualizace piškotů v reálném čase (default), <b>no</b> = odložená aktualizace piškotů.
<b>debug?</b>	<b>yes</b> = výstup verbózních záznamů interních událostí prostředí View (implicitně <b>no</b> ).
<b>silent?</b>	<b>yes</b> = nehlásit procesní chyby dialektů VID nebo Draw (implicitně <b>no</b> ).

# 12. Připojení komponenty View

Komponenta View není implicitně připojena při kompilaci. Za účelem jejího připojení musí hlavní skript Redu deklarovat svou závislost v záhlaví použitím pole **Needs** :

```
Red [  
  Needs: 'View'  
]
```

[NOTE] Při použití konzol, automaticky generovaných binárnkou **red**, je komponenta View implicitně



připojena na podporujících platformách; pole **Needs** v záhlaví tedy není požadováno ve skriptech, spouštěných z těchto konzol.

## 13. Extra funkce

Function	Popis
<b>view</b>	Zobrazit na monitoru okno ze stromu poškotů nebo z bloku kódu VID. Zavede smyčku událostí, pokud není použito upřesnění (refinement) <b>/no-wait</b> .
<b>unview</b>	Zruší jedno či více oken.
<b>layout</b>	Přemění blok kódu VID na strom piškotů.
<b>center-face</b>	Vystředí piškot relativně ke svému rodiči.
<b>dump-face</b>	Poskytne kompaktní popis stromové struktury piškotů (pro účely ladění).
<b>do-actor</b>	Vyhodnotí ručně aktéra piškotu.
<b>do-events</b>	Spustí smyčku událostí (optionally just process pending events and return).
<b>draw</b>	Vykreslí blok dialektu Draw do obrázku.
<b>to-image</b>	Přemění jakýkoliv vykreslovaný piškot na obrázek.
<b>set-focus</b>	Nastaví focus na určený piškot (face).
<b>size-text</b>	Změří v pixelech velikost textu v piškotu (berouce v úvahu vybraný font).