

Podręcznik tworzenia skórek do Nowego Gadu-Gadu

Spis treści

Podręcznik tworzenia skórek do Nowego Gadu-Gadu.....	1
Spis treści.....	2
Wstęp.....	3
Podstawy.....	3
Elementy graficzne.....	4
Pliki podstawowe.....	4
Gadżety.....	5
Zaawansowane.....	5
Sekcja MainFontColors.....	6
Sekcja MainColors.....	6
Sekcja ExtraColors.....	6
Lista kontaktów.....	7
Obramowanie.....	7
Czcionki.....	8
Hover.....	9
Panel zmiany statusu.....	9
Pozostałe.....	9
Udostępnianie.....	10
Dodatek A: Wykaz plików graficznych skórki.....	11
Dodatek B: Specyfikacja pliku config.xml.....	12

Wstęp

Najnowsza odsłona Gadu-Gadu daje duże możliwości modyfikacji interfejsu poprzez wbudowany mechanizm skórek. Całość opiera się na jednym pliku .xml oraz zestawie plików graficznych w określonym formacie. Do stworzenia skórki dla Nowego GG niezbędne będą następujące narzędzia:

- edytor tekstowy (przykładowo Notatnik, Notepad++)
- edytor plików graficznych (GIMP, Inkscape)
- archiwizator plików (opcjonalny)

Podstawy

Najważniejszym elementem składowym każdej skórki jest plik config.xml umieszczony bezpośrednio w katalogu ze skórką. Jego zawartość podzielona jest na dwie części zawarte pomiędzy znacznikami Info oraz Global. Pierwsza z nich zawiera informacje o nazwie skórki i jej autorze (odpowiednio Name oraz Author). Zasadniczo interesuje nas tylko część Global, w której definiowane będą wszystkie dodatkowe parametry. Podzielona jest ona na kilka podsekcji określających zakres definiowanych zmian:

- *MainColors* – najważniejsze kolory w aplikacji
- *FontColors* – kolory bazowych czcionek
- *ExtraColors* – pozwala na zdefiniowanie dużej liczby dodatkowych atrybutów

Poniżej przedstawiona została przykładowa zawartość pliku config.xml. Jest ona wymaganym minimum dla każdej tworzonej skórki:

```
<?xml version="1.0" encoding="utf-8"?>
<Skin>
  <Info>
    <!-- Nazwa skórki -->
    <Name>Winter Time</Name>

    <!-- Autor skórki -->
    <Author> Krzysztof Anusiewicz</Author>
  </Info>
  <Global>
    <MainTransparency>0</MainTransparency>
    <ContactListTransparency>0</ContactListTransparency>

  </Global>
</Skin>
```

Elementy zawarte pomiędzy znacznikami <!-- oraz --> to komentarze i nie są one brane pod uwagę podczas interpretacji pliku przez komunikator.

Parametry *ContactListTransparency* oraz *MainTransparency* decydują o przezroczystości okna listy kontaktów oraz wszystkich pozostałych okien programu. Prawidłowe wartości dla tych parametrów mieszczą się w zakresie od 0 (brak przezroczystości) do 255 (maksymalna przezroczystość). Jest to jeden z dwóch typów danych wykorzystywanych przy tworzeniu skórki.

Drugim bardzo ważnym typem jest kolor przedstawiony w postaci uporządkowanej czwórki parametrów RGBA odpowiedzialnych za barwy: czerwoną, zieloną, niebieską oraz przezroczystość. Przykładowa definicja koloru fioletowego wygląda następująco: r="189" g="24" b="231" a="255".

W dalszej części niniejszej instrukcji znajduje się szczegółowe omówienie wszystkich elementów komunikatora modyfikowalnych z poziomu pliku config.xml.

Elementy graficzne

Oprócz wspomnianego pliku do skórki można dołączyć pliki graficzne, które mają dwa podstawowe zastosowania:

- są zamiennikami do wybranych elementów konfigurowalnych z poziomu pliku config.xml
- stanowią dodatki w postaci gadżetów – elementów wykraczających poza obszar okna listy kontaktów

Wszystkie pliki graficzne muszą być zapisane w jednym z formatów: BMP, GIF, JPG, JPEG, MNG, PNG, PBM, PGM, PPM lub TIFF. W celu większego ujednoczenia efektów swojej pracy zalecane jest stosowanie jednego formatu, jednakże nie jest to wymogiem koniecznym do prawidłowego wyświetlania skórki.

Pliki podstawowe

Do najważniejszych plików graficznych wykorzystywanych w mechanizmie skórek Nowego GG możemy zaliczyć dwa pliki związane z tłem, oraz kolejne dwa definiujące podstawowe kolory programu. Plik *defaultBackground* odpowiedzialny jest za tło wszystkich okien aplikacji, *contactListBackground* natomiast stanowi wypełnienie listy kontaktów. W zasadzie pliki te mogą mieć dowolną wielkość, jednakże w celu uzyskania lepszego efektu zalecamy stosowanie plików o minimalnym rozmiarze 500x500 pikseli i rozdzielczości 96dpi. Ogólna zasada jest taka, że im mniejszy plik tła, tym częściej program musi powielać jego zawartość co może spowodować niechciane spowolnienie w działaniu. Przykładowo jako tło okien można ustawić grafikę, jak na poniższym obrazku:



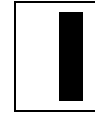
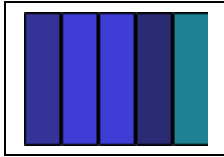
Kolejne dwa ważne pliki graficzne, które można dołączyć do skórki to:

- *fontColors* – odpowiadający za bazowe kolory czcionek, podzielony na dwie równe części.
- *mainColors* – zawierający pięć ułożonych jeden po drugim pasków kolorów odpowiedzialnych za główne barwy aplikacji.

Najprościej jest skopiować te dwa pliki z istniejącej już skórki i w edytorze graficznym zmodyfikować poszczególne barwy narzędziem wypełniającym. Informacje na temat znaczenia poszczególnych kolorów w tych dwóch plikach znajdują się w dalszej części instrukcji. Dla potrzeb tej instrukcji zastosowaliśmy takie oto pliki:

mainColors

fontColors

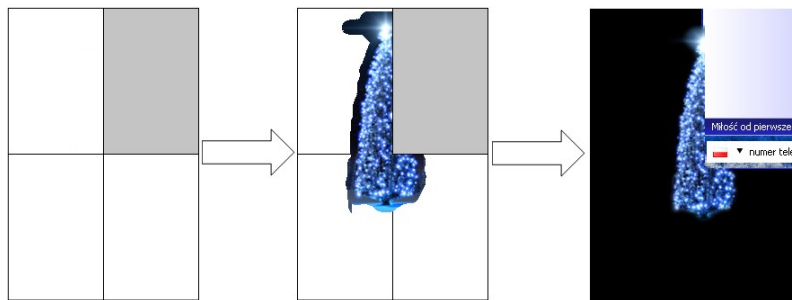


Gadżety

Gadżety można dodać we wszystkich czterech rogach okna komunikatora, jak i po jego czterech bokach. Pliki mogą mieć dowolny rozmiar i kształt, jednak należy je stosować z rozmysłem w celu zachowania odrobiny estetyki :-). Poniższa tabela zawiera zestawienie wszystkich gadżetów oraz miejsce, w którym konkretny element będzie widoczny.

Nazwa pliku	Umieszczenie w oknie programu
gadgetTopLeft	Lewy górny róg
gadgetTop	Górna krawędź
gadgetTopRight	Prawy górny róg
gadgetLeft	Lewa krawędź
gadgetRight	Prawa krawędź
gadgetBottomLeft	Lewy dolny róg
gadgetBottom	Dolna krawędź
gadgetBottomRight	Prawy dolny róg

Dla celów przykładowej skórki stworzymy dwa gadżety – w dolnym lewym oraz górnym prawym rogu okna komunikatora. W ulubionym programie graficznym tworzymy pusty plik o wymiarach 240x320 pikseli i przezroczystym tłem (jeżeli tło będzie białe, to wokół stworzonego obrazka będzie widoczny biały prostokąt). Ponieważ gadżet ma znajdować się w dolnym lewym rogu, to przeciwległa do niego ćwiartka obrazu musi pozostać pusta – na poniższym obrazku szary prostokąt. Dzięki zastosowanemu w mechanizmie skórek wygładzaniu oraz kilku drobnym poprawkom udało się uzyskać końcowy efekt:



Drugi gadżet zawierający napis „It's Winter Time!” został stworzony w analogiczny sposób, dając efekt jak poniżej:



Zaawansowane

W tym rozdziale zostaną omówione najważniejsze parametry, które można definiować w pliku config.xml. Kompletną specyfikację można znaleźć na końcu tego podręcznika.

Sekcja MainFontColors

Zawiera tylko dwa definiowalne znaczniki, które mogą być stosowane zamiennie z plikiem *fontColors*. Parametr *MainFontColor* (pierwszy kolor w przypadku pliku) odpowiada za czcionki w menu kontekstowym, paskach tytułowych, w panelu zmiany statusu oraz kolor elementów w hoverze. *ContactListFontColor* definiuje natomiast kolor czcionki kontaktów na liście.

W przypadku gdy zdecydujemy się na wprowadzenie tej sekcji wymagane jest zdefiniowanie obydwu kolorów. Odpowiednikiem przedstawionego trochę wcześniej pliku czcionek jest następująca definicja:

```
<MainFontColors>
  <MainFontColor r="255" g="255" b="255" a="255" />
  <ContactListFontColor r="0" g="0" b="0" a="255" />
</MainFontColors>
```

Sekcja MainColors

Jest to druga bardzo ważna sekcja, której elementy odpowiadają za podstawowe kolory skóry. Wśród nich możemy wyróżnić:

- *TitleBar* - kolor paska tytułowego oraz niektórych przycisków, który definiujemy w postaci dwóch parametrów odpowiadających za kolor początkowy (Top) oraz końcowy (Bottom) gradientu generowanego przez komunikator. Ponadto parametr *TitleBarBottom* jest kolorem podświetlenia elementów menu po najechaniu na nich kursorem myszy.
- *NotificationAreaBackground* – kolor tła elementów powiadomień
- *WindowBackground* – kolor tła prawie wszystkich okien aplikacji. Jeżeli w skórcie znajduje się plik *defaultBackground*, to parametr ten jest aplikowany tylko dla pasków przewijania, koloru tła wszystkich menu rozwijanych i niektórych elementów dodatkowych.
- *ContactListHighlight* – kolor zaznaczonego (poprzez kliknięcie) elementu listy kontaktów, listy skórek oraz listy profili.

Sekcję tą można stosować zamiennie z plikiem *mainColors*, znaczenie poszczególnych barw jest takie samo zgodnie z kolejnością podanych parametrów. Podobnie jak w sekcji *MainFontColors* decydując się na wprowadzenie tej sekcji wymagane jest zadeklarowanie wszystkich pięciu parametrów. Odpowiednikiem przedstawionego wcześniej pliku kolorów jest następująca definicja:

```
<MainColors>
  <TitleBarTop r="52" g="52" b="152" a="255" />
  <TitleBarBottom r="62" g="62" b="215" a="255" />
  <NotificationAreaBackground r="62" g="62" b="215" a="255" />
  <WindowBackground r="43" g="43" b="116" a="255" />
  <ContactListHighlight r="31" g="131" b="148" a="255" />
</MainColors>
```

Sekcja ExtraColors

Jest to najbardziej rozbudowana sekcja, i w odróżnieniu od dwóch poprzednich nie jest wymagane definiowanie wszystkich parametrów. Wewnątrz nie ma uporządkowania na kolejne podsekcje, jednakże można wyróżnić kilka zastawów parametrów związanych z

konkretnymi elementami komunikatora takimi jak lista kontaktów, obramowanie, czcionki, hover, panel zmiany statusu oraz dodatkowe elementy okien. Przejdźmy zatem do ich omówienia.

Listy kontaktów

Do dyspozycji mamy 5 parametrów, odpowiedzialnych za: tło kontaktów na pozycjach parzystych i nieparzystych (odpowiednio *ContactListOddFieldColor* i *ContactListEvenFieldColor*), tło każdego elementu po najechaniu kursorem myszki – *ContactListHoveredFieldColor* oraz dwa kolory określające gradient tła wybranego kontaktu (poprzez kliknięcie na nim) – *ContactlistSelectedFieldTopColor* oraz *ContactlistSelectedFieldBottomColor*.

Obramowanie

Twórcy skórek tutaj również znajdują bardzo duże możliwości modyfikacji zarówno obramowania okien komunikatora – *WindowBorderColor* oraz *WindowInactiveBorderColor*, jak i wszelkiego rodzaju przycisków:

- okna rozmowy – *ChatWindowButtonsBorderColor*
- panelu szybkiego wyszukiwania kontaktów – *SearchButtonsBorderColor*
- oraz tych znajdujących się w hoverze – *HoverButtonsBorderColor*

Dodatkowo mamy możliwość zdefiniowania obramowania pól tekstowych do wpisywania danych (ale nie dotyczy to pola nowej wiadomości tekstowej) – *InputInnerBorderColor* oraz pola do wyszukiwania kontaktów po wpisaniu tekstu – *FilterEditSignaledBorderColor*.

Dla naszej przykładowej skórki wszystkie obramowania mają kolor biały, z wyjątkiem pól do wpisywania tekstu, które mają ramkę niebieską:

```
<ExtraColors>
  <WindowBorderColor r="255" g="255" b="255" a="255" />
  <WindowInactiveBorderColor r="255" g="255" b="255" a="255" />
  <NormalButtonsBorderColor r="255" g="255" b="255" a="255" />
  <SpecialButtonsBorderColor r="255" g="255" b="255" a="255" />
  <ChatWindowButtonsBorderColor r="255" g="255" b="255" a="255" />
  <SearchButtonsBorderColor r="255" g="255" b="255" a="255" />
  <HoverButtonsBorderColor r="255" g="255" b="255" a="255" />
  <InputInnerBorderColor r="52" g="52" b="152" a="255" />
  <FilterEditSignaledBorderColor r="52" g="52" b="152" a="255" />
</ExtraColors>
```

Oczekiwanym efektem jest między innymi białe obramowanie elementów hovera. Jak widać udało się to osiągnąć (zaznaczony przycisk wysyłania wiadomości SMS):



Czcionki

Do tej grupy zaliczamy następujące modyfikowalne kolory czcionek:

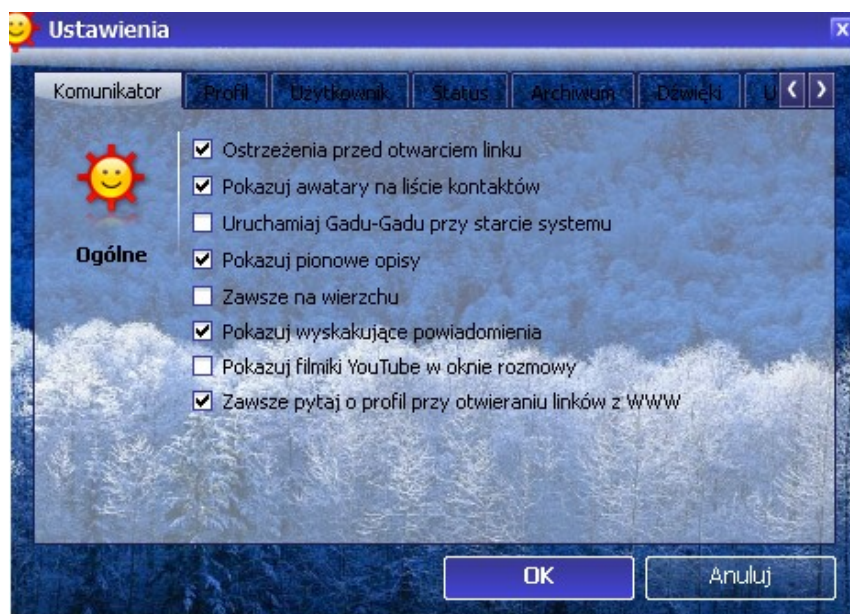
- Na pasku tytułowym okna listy kontaktów – *TitleBarFontColor*
- Na pasku tytułowym okna logowania – *LoginBarFontColor*
- Opisu pod kontaktami – *ContactListDescFontColor*
- Odnośników w opisie pod kontaktem – *ContactListLinkFontColor*
- Okna ustawień - *ConfigWindowFontColor*
- Bazowych przycisków aplikacji (Zadzwoń, SMS, panel zmiany statusu...) - *NormalButtonFontColor*
- Dodatkowych przycisków – *SpecialButtonFontColor*
- Przycisków w powiadomieniach *EventWidgetAcceptLinkFontColor* – odnośnik odpowiadający za „akceptuj” oraz *EventWidgetRejectLinkFontColor* odpowiadający za „odrzuć”. Stosowany na przykład podczas przesyłania plików.

Do naszej przykładowej skórki dodamy pięć atrybutów. Pierwsze dwa związane z opisami w oknie listy kontaktów:

```
<ContactListDescFontColor r="52" g="52" b="152" a="255" />  
<ContactListLinkFontColor r="255" g="0" b="0" a="255" />
```

Kolejnym będzie kolor czcionki w oknie ustawień – jak pewnie zauważyliście obraz tła jest dość jasny i biała czcionka nie będzie się dobrze wyróżniać. Zmieńmy ją zatem na ciemniejszą:

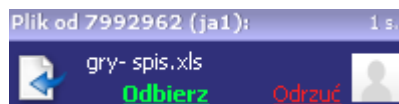
```
<ConfigWindowFontColor r="0" g="0" b="0" a="255" />
```



Ostatecznie dokonamy zmian w kolorze przycisków powiadomień – niech akceptacja będzie oznaczona na zielono, a odrzucenie na czerwono. Poniższa definicja:

```
<EventWidgetAcceptLinkFontColor r="0" g="255" b="64" a="255" />
<EventWidgetRejectLinkFontColor r="255" g="28" b="28" a="255" />
```

Da następujący efekt:



Hover

Hover czyli niewielki element pojawiający się po najechaniu lub kliknięciu na kontakt z listy również został wyposażony w komplet modyfikowalnych atrybutów. Za domyślny kolor tła tego elementu odpowiadają dwa parametry definiujące wypełnienie gradientem - *HoverGradientTopColor* oraz *HoverGradientBottomColor*. W analogiczny sposób zdefiniować można gradient tła zaznaczonego hovera (poprzez kliknięcie na danym kontakcie) – *HoverSelectedGradientTopColor* oraz *HoverSelectedGradientBottomColor*. Ponadto mamy możliwość ustalenia koloru ikonki w hoverze niezależnie od bazowego koloru ustalonego przez plik *fontColor*. Do tego celu należy posłużyć się parametrem *HoverButtonColor*.

Panel zmiany statusu

Do modyfikacji tego elementu komunikatora udostępnione zostały cztery atrybuty:

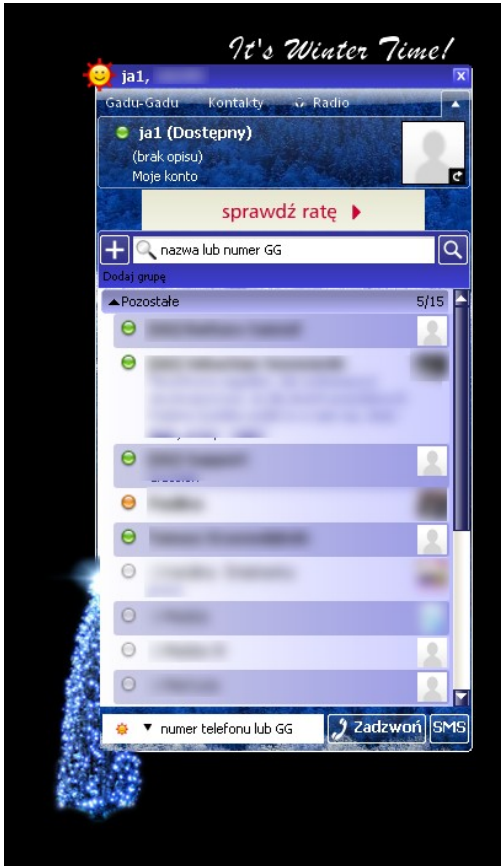
- *FoldingButtonTopColor* – określający kolor przycisku do zwinięcia lub rozwinięcia panelu
- *LoginWidgetTopColor* oraz *LoginWidgetBottomColor* – definiują barwę początkową i końcową gradientu wypełnienia panelu
- *LoginWidgetBorder* – kolor obramowania panelu

Pozostałe

Podobnie do listy kontaktów możemy niezależnie zmodyfikować wygląd elementów na liście profili, liście skórek oraz w oknie edycji danych kontaktu:

- *WindowOddFieldColor* oraz *WindowEvenFieldColor* odpowiadają za kolory elementów parzystych oraz nieparzystych
- *WindowOddSignaledFieldColor* i *WindowEvenSignaledFieldColor* – w oknie edycji kontaktu zakładki, które będą „mrużyć” po pobraniu danych z katalogu publicznego
- Kolor klikniętego elementu definiowany jest poprzez gradient – jego barwy określamy przy pomocy parametrów *WindowSelectedFieldTopColor* oraz *WindowSelectedFieldBottomColor*
- Ostatecznie, kolor elementu po najechaniu myszką można zmienić poprzez wartość określoną w *WindowHoveredFieldColor*

Po tych zabiegach skórka jest już gotowa do udostępnienia znajomym:



```
<?xml version="1.0" encoding="utf-8"?>
<Skin>
<Info>
<!-- Nazwa skórki -->
<Name>Winter Time</Name>

<!-- Autor skórki -->
<Author>Krzysztof Anusiewicz</Author>
</Info>
<Global>
<MainTransparency>0</MainTransparency>
<ContactListTransparency>128</ContactListTransparency>
<MainFontColors>
<MainFontColor r="255" g="255" b="255" a="255" />
<ContactListFontColor r="0" g="0" b="0" a="255" />
</MainFontColors>

<ExtraColors>
<WindowBorderColor r="255" g="255" b="255" a="255" />
<WindowInactiveBorderColor r="255" g="255" b="255" a="255" />
<NormalButtonsBorderColor r="255" g="255" b="255" a="255" />
<SpecialButtonsBorderColor r="255" g="255" b="255" a="255" />
<ChatWindowButtonsBorderColor r="255" g="255" b="255" a="255" />
<SearchButtonsBorderColor r="255" g="255" b="255" a="255" />
<HoverButtonsBorderColor r="255" g="255" b="255" a="255" />
<InputInnerBorderColor r="52" g="52" b="152" a="255" />
<FilterEditSignaledBorderColor r="52" g="52" b="152" a="255" />

<ContactListDescFontColor r="52" g="52" b="152" a="255" />
<ContactListLinkFontColor r="255" g="0" b="0" a="255" />
<ConfigWindowFontColor r="0" g="0" b="0" a="255" />

<EventWidgetAcceptLinkFontColor r="0" g="255" b="64" a="255" />
<EventWidgetRejectLinkFontColor r="255" g="28" b="28" a="255" />
</ExtraColors>

</Global>

</Skin>
```

Udostępnianie

Jeżeli chcemy udostępnić stworzoną skórkę innym, należy zawartość folderu skompresować dowolnym archiwizatorem do formatu ZIP. Jeżeli natomiast skórka została stworzona tylko do użytku własnego to nie ma takiej potrzeby, wystarczy umieścić folder ze skórką w katalogu *skins*, gdzie został zainstalowany program.

Dodatek A: Wykaz plików graficznych skórki

Nazwa pliku	Preferowany rozmiar (w pikselach)
defaultBackground	500x500 lub zbliżony
contactListBackground	500x500 lub zbliżony
fontColors	64x64
mainColors	99x65
gadgetTopLeft	dowolny
gadget top	dowolny
gadgetTopRight	dowolny
gadgetLeft	dowolny
gadgetRight	dowolny
gadgetBottomLeft	dowolny
gadgetBottom	dowolny
gadgetBottomRight	dowolny

Dodatek B: Specyfikacja pliku config.xml

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="Skin" type="SkinType" />
  <xsd:complexType name="SkinType">
    <xsd:all minOccurs="1" maxOccurs="1">
      <xsd:element name="Info" type="InfoType" />
      <xsd:element name="Global" type="GlobalType" />
    </xsd:all>
  </xsd:complexType>
  <xsd:complexType name="InfoType">
    <xsd:all>
      <xsd:element name="Name" type="xsd:string" />
      <xsd:element name="Author" type="xsd:string" default="" />
    </xsd:all>
  </xsd:complexType>
  <xsd:complexType name="GlobalType">
    <xsd:all>
      <xsd:element name="MainColors" type="MainColorsType" minOccurs="0" maxOccurs="1" />
      <xsd:element name="MainFontColors" type="MainFontColorsType" minOccurs="0" maxOccurs="1" />
      <xsd:element name="ExtraColors" type="ExtraColorsType" minOccurs="0" maxOccurs="1" />
      <xsd:element name="MainTransparency" type="TransparencyType" />
      <xsd:element name="ContactListTransparency" type="TransparencyType" />
    </xsd:all>
  </xsd:complexType>
  <xsd:complexType name="MainColorsType" >
    <xsd:all>
      <xsd:element name="TitleBarTop" type="ColorType" />
      <xsd:element name="TitleBarBottom" type="ColorType" />
      <xsd:element name="NotificationAreaBackground" type="ColorType" />
      <xsd:element name="WindowBackground" type="ColorType" />
      <xsd:element name="ContactListHighlight" type="ColorType" />
    </xsd:all>
  </xsd:complexType>
  <xsd:complexType name="MainFontColorsType">
    <xsd:all>
      <xsd:element name="MainFontColor" type="ColorType" />
      <xsd:element name="ContactListFontColor" type="ColorType" />
    </xsd:all>
  </xsd:complexType>
  <xsd:complexType name="ExtraColorsType">
    <xsd:all>
      <xsd:element name="ContactListOddFieldColor" type="ColorType" minOccurs="0" />
      <xsd:element name="ContactListEvenFieldColor" type="ColorType" minOccurs="0" />
      <xsd:element name="ContactListSelectedFieldTopColor" type="ColorType" minOccurs="0" />
      <xsd:element name="ContactListSelectedFieldBottomColor" type="ColorType" minOccurs="0" />
      <xsd:element name="ContactListHoveredFieldColor" type="ColorType" minOccurs="0" />

      <xsd:element name="WindowOddFieldColor" type="ColorType" minOccurs="0" />
      <xsd:element name="WindowEvenFieldColor" type="ColorType" minOccurs="0" />
      <xsd:element name="WindowOddSignaledFieldColor" type="ColorType" minOccurs="0" />
      <xsd:element name="WindowEvenSignaledFieldColor" type="ColorType" minOccurs="0" />
      <xsd:element name="WindowSelectedFieldTopColor" type="ColorType" minOccurs="0" />
      <xsd:element name="WindowSelectedFieldBottomColor" type="ColorType" minOccurs="0" />
      <xsd:element name="WindowHoveredFieldColor" type="ColorType" minOccurs="0" />

      <xsd:element name="WindowBorderColor" type="ColorType" minOccurs="0" />
      <xsd:element name="WindowInactiveBorderColor" type="ColorType" minOccurs="0" />

      <xsd:element name="NormalButtonsBorderColor" type="ColorType" minOccurs="0" />
      <xsd:element name="SpecialButtonsBorderColor" type="ColorType" minOccurs="0" />
      <xsd:element name="ChatWindowButtonsBorderColor" type="ColorType" minOccurs="0" />
      <xsd:element name="SearchButtonsBorderColor" type="ColorType" minOccurs="0" />
      <xsd:element name="HoverButtonsBorderColor" type="ColorType" minOccurs="0" />

      <xsd:element name="ContactWindowBackgroundColor" type="ColorType" minOccurs="0"/>
      <xsd:element name="ScrollBarColor" type="ColorType" minOccurs="0" />

      <xsd:element name="CallWidgetColor" type="ColorType" minOccurs="0" />
      <xsd:element name="InputInnerBorderColor" type="ColorType" minOccurs="0" />

      <xsd:element name="TitleBarFontColor" type="ColorType" minOccurs="0"/>
      <xsd:element name="LoginBarFontColor" type="ColorType" minOccurs="0" />
    </xsd:all>
  </xsd:complexType>

```

```

<xsd:element name="ContactListDescFontColor" type="ColorType" minOccurs="0" />
<xsd:element name="ContactListLinkFontColor" type="ColorType" minOccurs="0" />
<xsd:element name="ConfigWindowFontColor" type="ColorType" minOccurs="0" />
<xsd:element name="NormalButtonFontColor" type="ColorType" minOccurs="0"/>
<xsd:element name="SpecialButtonFontColor" type="ColorType" minOccurs="0" />
<xsd:element name="FilterEditSignaledBorderColor" type="ColorType" minOccurs="0" />

    <xsd:element name="HoverGradientTopColor" type="ColorType" minOccurs="0" />
    <xsd:element name="HoverGradientBottomColor" type="ColorType" minOccurs="0" />
    <xsd:element name="HoverSelectedGradientTopColor" type="ColorType" minOccurs="0" />
    <xsd:element name="HoverSelectedGradientBottomColor" type="ColorType" minOccurs="0" />
<xsd:element name="HoverButtonColor" type="ColorType" minOccurs="0" />

<xsd:element name="FoldingButtonTopColor" type="ColorType" minOccurs="0"/>
<xsd:element name="LoginWidgetTopColor" type="ColorType" minOccurs="0" />
<xsd:element name="LoginWidgetBottomColor" type="ColorType" minOccurs="0" />
    <xsd:element name="LoginWidgetBorder" type="ColorType" minOccurs="0" />

<xsd:element name="EventWidgetAcceptLinkFontColor" type="ColorType" minOccurs="0" />
<xsd:element name="EventWidgetRejectLinkFontColor" type="ColorType" minOccurs="0" />

</xsd:all>
</xsd:complexType>

<xsd:complexType name="ColorType">
    <xsd:attribute name="r" type="ColorNumberType" />
    <xsd:attribute name="g" type="ColorNumberType" />
    <xsd:attribute name="b" type="ColorNumberType" />
    <xsd:attribute name="a" type="ColorNumberType" default="255" />
</xsd:complexType>

<xsd:simpleType name="ColorNumberType">
    <xsd:restriction base="xsd:int">
        <xsd:minInclusive value="0" />
        <xsd:maxInclusive value="255" />
    </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="TransparencyType">
    <xsd:restriction base="xsd:int">
        <xsd:minInclusive value="0" />
        <xsd:maxInclusive value="255" />
    </xsd:restriction>
</xsd:simpleType>
</xsd:schema>

```