# 个性化后台设计说明书

**变更记录**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **版本** | **修订时间** | **修订人** | **修订类型** | **修订章节** | **修订内容** |
| 1.0 | 2021-01-13 | 唐杰 | A | ALL | 建立 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

注1：修订类型分为A-ADDED，M-MODIFIED，D–DELETED

注2：对该文件内容增加、删除或修改均需填写此记录，详细记载变更信息，以保证其可追溯性

目录

[个性化后台设计说明书 1](#_Toc16162)

[一、 概述 3](#_Toc23178)

[1.1. 主题概述 3](#_Toc28627)

[1.1.1. 主题路径 3](#_Toc17063)

[1.1.2. META主题 3](#_Toc30352)

[1.1.3. GTK主题 3](#_Toc26711)

[1.1.4. 窗口主题 4](#_Toc29293)

[1.1.5. 图标和光标主题 4](#_Toc24087)

[二、 模块设计 4](#_Toc17609)

[2.1. 主题设置 4](#_Toc32086)

[2.1.1. 主题监控 4](#_Toc4276)

[2.1.2. 主题解析器 5](#_Toc22204)

[2.1.3. 主题管理 5](#_Toc7849)

[2.2. 字体设置 6](#_Toc27014)

[2.3. 背景设置 6](#_Toc3830)

[三、 接口设计 7](#_Toc703)

[3.1. DBUS接口 7](#_Toc22664)

[3.1.1. 函数 7](#_Toc27500)

[3.1.2. 属性 8](#_Toc16120)

[3.1.3. 信号 8](#_Toc1244)

# 概述

本文档对个性化设置后台程序的概要设计进行了阐述。个性化设置后台是作为一个插件由控制中心后台(kiran-session-daemon)加载，目前主要应用于桌面环境，为上层的桌面应用提供必要的接口。

个性化设置主要包含三个部分：主题设置，字体设置和背景设置。

## 主题概述

主题设置中可设置的主题类型包括：META主题、GTK主题、窗口主题、图标主题和光标主题。

### 主题路径

META/GTK/窗口主题搜索的路径相同，主题目录的父目录为themes或者.themes，例如/usr/share/themes/

图标和光标主题搜索的路径相同，主题目录的父目录为icons或者.icons，例如/usr/share/icons/

### META主题

META主题严格来说是属于一个主题包。这个主题包阐述了GTK/窗口/图标/光标应该使用什么样的主题，阐述的信息放在主题目录的index.theme文件中，具体格式如下：

[X-GNOME-Metatheme]

GtkTheme=Kiran

MetacityTheme=Kiran

IconTheme=Kiran

CursorTheme=Kiran

### GTK主题

GTK主题主要是使用在GTK开发的应用程序中，GTK是通过读取XSettings规范中定义的Net/ThemeName变量来得到主题名字。

GTK主版本分为GTK2和GTK3，在主题目录下分别对应gtk-2.x和gtk-3.x两个文件夹，GTK2中使用的主题样式配置为gtk-2.x/gtkrc，GTK3中使用的主题样式配置为gtk-3.x/gtk.css。

如果有多个路径有相同的主题名，那么搜索顺序靠前的主题路径会被加载，GTK3中读取主题(样式配置)的顺序如下：

$XDG\_CONFIG\_HOME/gtk-3.0/gtk.css

$XDG\_DATA\_HOME/themes/THEME/gtk-VERSION/gtk.css

$HOME/.themes/THEME/gtk-VERSION/gtk.css

$XDG\_DATA\_DIRS/themes/THEME/gtk-VERSION/gtk.css

DATADIR/share/themes/THEME/gtk-VERSION/gtk.css

DATADIR是定义在GTK3配置文件中或者来自环境变量GTK\_DATA\_PREFIX，目前GTK3没有提供API获取这个值。THEME为主题名，VERSION的的格式为major.minor，假设当前使用的GTK版本为GTK3-3.22，则搜索顺序为：gtk-3.22 -> gtk-3.20 -> gtk-3.18 -> gtk-3.16 -> gtk-3.14 -> gtk-3.0, 0到14中间不再搜索。

### 窗口主题

窗口主题是由窗口管理器(marco)负责加载。主题目录下会存放一个metacity-1文件夹，文件夹中的metacity-theme-1.xml、metacity-theme-2.xml和metacity-theme-3.xml为主题样式配置信息。

### 图标和光标主题

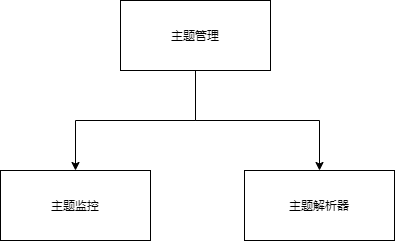
图标和光标主题名分别对应XSettings规范中定义的Net/IconThemeName和Gtk/CursorThemeName变量，GTK会根据这两个变量来加载图标和光标主题。

图标主题有对应的freedesktop图标主题规范，图标主题配置是主题目录中的index.theme文件。光标主题没有配置文件，对应的资源文件在主题目录中的cursors子目录中。

# 模块设计

## 主题设置

主题设置分为主题管理、主题监控和主题解析三个模块，具体的关系如下：



### 主题监控

主题监控模块主要负责监控主题目录/文件的变化，包括删除、添加和更改等，一旦出现变更，主题监控模块会发出主题变化的信号，并提供变化主题的位置、搜索优先级和主题类型等信息。

其中，META/GTK/窗口主题监控的路径为：1) $XDG\_DATA\_HOME/themes/; 2) $HOME/.themes/THEME/; 3) $XDG\_DATA\_DIRS/themes/THEME/。图标和光标主题监控的路径跟Gtk::IconTheme::get\_search\_path提供的路径列表相同。

### 主题解析器

主题解析器负责对主题目录的内容进行解析，并返回主题相关的信息。大致流程如下：

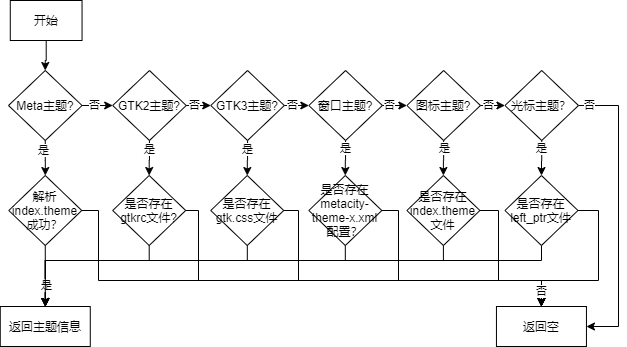


图 1 主题目录解析流程

Meta主题目录中的index.theme至少包含对GtkTheme/MetacityTheme/IconTheme字段的设置，否则解析失败。

### 主题管理

顾名思义，主题管理是用于管理所有的主题信息，可以通过主题管理模块获取不同主题类型的主题信息，也可以对不同主题类型进行设置。

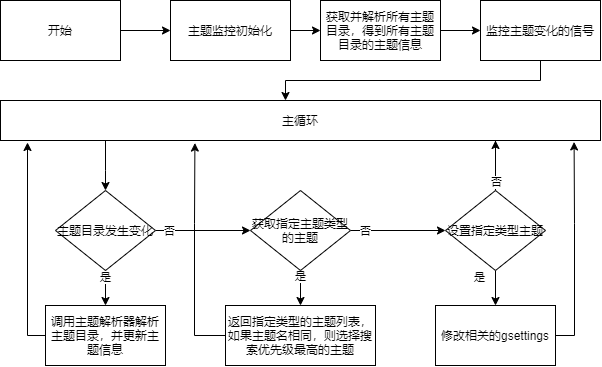


图 2 主题管理流程

## 字体设置

字体设置的类型包括：应用程序字体、文档字体、桌面字体、窗口标题字体、等宽字体。字体设置和获取是直接读写了相关的gsettings变量，使用相关字体类型的程序负责监听gsettings的变化，然后更新程序的字体，相关信息如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 字体类型 | 使用程序 | schema\_id | key |
| 应用程序字体 | GTK应用程序 | com.kylinsec.kiran.xsettings | gtk-font-name |
| 文档字体 | / | org.mate.interface | document-font-name |
| 桌面字体 | 文件管理器(caja) | org.mate.caja.desktop | font |
| 窗口标题字体 | 窗口管理器(marco) | org.mate.Marco.general | titlebar-font |
| 等宽字体 | 终端(mate-terminal) | org.mate.interface | monospace-font-name |

## 背景设置

背景设置只支持一种样式：将图片进行缩放后，然后（横向和纵向都）居中进行显示，缩放比例为显示器的分辨率/图片的大小，如果宽度和高度的缩放比不相同，则选择最大值。

由于用户可能会在多个背景图片之间来回进行设置，因此后端会对最近已经设置过的背景图片进行缓存，避免重复加载图片带来的IO开销。如果5分钟后未再调用背景设置接口，缓存数据会被清理，避免过多的暂用内存。

设置背景之前先要判断caja是否会去绘制桌面背景，如果caja程序正在运行且org.mate.background中的show-desktop-icons字段为true，则说明caja会进行桌面背景的绘制，此时背景设置插件不再重复去绘制桌面背景，绘制的操作交给caja去执行。

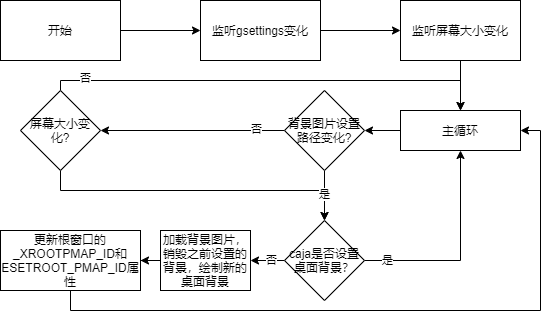


图 3 桌面背景设置流程

# 接口设计

## DBUS接口

### 函数

String GetThemes(Int32 type);

函数说明：获取指定主题类型的主题信息列表。

参数说明：

* type: 主题类型。包括META/GTK/窗口/图标/光标主题。

返回值：返回的主题信息列表为json字符串，需要对字符串进行反序列化解析，具体格式如下：[{“name”:”Kiran”, “path”: “/usr/share/themes/Kiran”}]，其中name为主题名，path为主题目录的文件路径。

SetTheme(Int32 type, String theme\_name);

函数说明：设置指定主题类型的主题。

参数说明：

* type: 主题类型。包括META/GTK/窗口/图标/光标主题。
* theme\_name: 主题名

返回值：如果主题类型非法或者主题名不存在则返回错误，否则返回成功。

String GetFont(Int32 type);

函数说明：获取字体类型为type的字体。

参数说明：

* type: 字体类型。包括应用程序/文档/桌面/窗口标题/等宽字体。

返回值：字体类型不存在则返回错误，否则返回对应的字体，字体包括了字体名和字体大小，例如’San 10’，表示字体名为San，字体大小为10。

SetFont(Int32 type, String font);

函数说明：设置字体类型为type的字体。

返回值：字体类型不存在则返回错误，否则返回成功。

SetDesktopBackground(String desktop\_background);

函数说明：设置桌面背景

参数说明：

* desktop\_background: 桌面背景图片位置。

返回值:函数不会失败

### 属性

|  |  |
| --- | --- |
| 属性名 | 含义 |
| gtk\_theme | GTK主题名 |
| metacity\_theme | 窗口主题名 |
| icon\_theme | 图标主题名 |
| cursor\_theme | 光标主题名 |
| desktop\_background | 桌面背景图片位置 |

### 信号

ThemeChanged(Int32 type)

信号说明：主题发生变化时会发送信号，可以调用GetThemes重新获取最新的主题信息列表。

参数说明：

* type: 发生变化的主题类型。包括META/GTK/窗口/图标/光标主题。