

Компонент Image

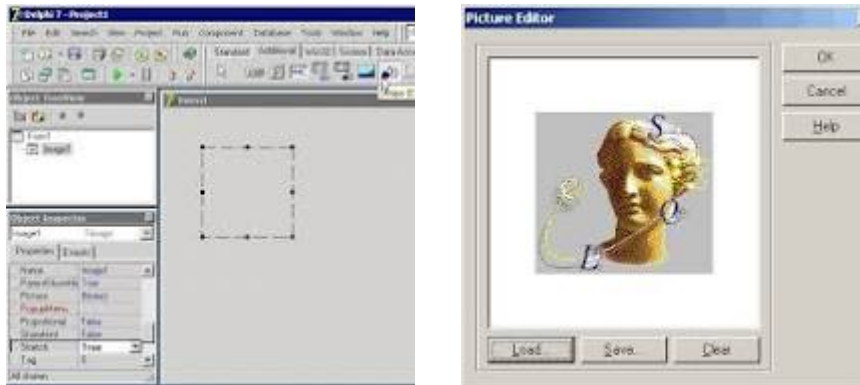
Дає відображення графічного зображення та забезпечує роботу із зображенням

Сторінка бібліотеки **Additional**

Клас **TImage**

Ієрархія **TObject** - **TPersistent** - **TComponent** - **TControl** - **TGraphicControl**

Модуль **extctrls**



Опис

Компонент **Image** відображає форму графічного зображення. Властивість **Picture** типу **TPicture** містить графічну складову, що відображається, у якій тип — бітова матриця, піктограми, метафайлу або певного користувачем типу. Властивість **Canvas** дозволяє створювати та редагувати зображення.

Під час проектування завантажити у властивість **Picture** графічний файл можна клацнувши на кнопці з трьома крапками біля властивості **Picture** у вікні Інспектора Об'єктів або при натисканні двічі на **Image**. Повинно відкритися вікно **Picture Editor**, яке дозволить завантажити у властивість **Picture** деякий графічний файл (через кнопку **Load**), зберегти відкритий файл, давши йому нове ім'я чи зберегти у новому каталозі.

Коли ви в процесі створення проекту зробили завантаження зображення з файлу в компонент **Image**, він не просто відобразить його, але й збереже завантажене зображення в програмі. Переваги полягає в тому, що ваш проект загалом буде поставлятися без окремого графічного файлу.

При встановленні якості **AutoSize** рівному **true**, то розмір компонента **Image** буде автоматично підігнаний під розмір поміщеної в зображення зображення. Якщо властивість **AutoSize** встановлено в значення **false**, то зображення може і не вміститися в компонент або ж, навпаки, площа компонента може виявитися набагато більшою за площу зображення.

Інша властивість - **Stretch** дозволяє реалізувати припасування не компонента під розмір малюнка, а сам малюнок під параметр компонента. Але оскільки навряд чи реально встановити розміри **Image** дуже точно пропорційними величині малюнка, то зображення буде спотворено. При установці **Stretch** рівне **true** може мати сенс тільки для якихось візерунків, але не для картинок. Властивість **Stretch** не діє зображення піктограм, які змінюють своїх розмірів.

Властивість — **Center** , яке встановлено в true, робить по центру зображення на площі Image, якщо параметри розміру компонента більші за параметр розміру малюнка.

Властивість – **Transparent** – прозорість. Якщо значення Transparent дорівнює true, зображення в Image буде прозорим. Це використовується для накладання зображень один на одного. Слід пам'ятати, що властивість Transparent поширюється лише бітові матриці. При цьому прозорим (тобто замінним на колір розташованого під ним зображення) робиться за замовчуванням колір нижнього лівого пікселя бітової матриці.

Властивість Picture дозволяє легко організувати обмін із графічними файлами будь-яких типів у процесі виконання програми. Ця властивість - об'єкт, який має в свою чергу підсвойства, які вказують на графічний об'єкт, що зберігається. Якщо в Picture зберігається бітова матриця, на неї вказує властивість Picture.Bitmap. Якщо піктограма зберігається, на неї вказує властивість Picture.Icon. На метафайл, що зберігається, вказує властивість Picture.Metafile. Зрештою, на графічний об'єкт довільного типу вказує властивість Picture.Graphic.

Об'єкт Picture та його властивості Bitmap, Icon, Metafile та Graphic мають методи файлового читання та запису LoadFromFile та SaveToFile. Для властивостей Picture.Bitmap, Picture.Icon та Picture.Metafile формат файлу повинен відповідати класу об'єкта: бітовій матриці, піктограмі, метафайлу. При читанні файлу як Picture.Graphic файл повинен мати формат метафайлу. А для самого об'єкта Picture методи читання та запису автоматично підлаштовуються під тип файлу.

Наприклад, якщо ви маєте в додатку компонент-діалог OpenPicture-Dialog, то завантаження в Image графічного фала, що вибирається користувачем, може бути організована оператором

if(OpenPictureDialogl.Execute) then

Image1.Picture.LoadFromFile(OpenPictureDialogl.FileName);

Завантажуваний цим оператором файл може бути будь-якого типу: бітова матриця, піктограма або метафайл. Якщо будуть відкриватися лише файли бітових матриць, то оператор завантаження файлу можна замінити на

Image1.Picture.Bitmap.LoadFromFile(OpenPictureDialogl.FileName);

Для піктограм можна було б використовувати оператор

Image1.Picture.Icon.LoadFromFile(OpenPictureDialogl.FileName);

а для метафайлів - оператор

Image1.Picture.Metafile.LoadFromFile(

OpenPictureDialogl.FileName);

АБО

Image1.Picture.Graphic.LoadFromFile(OpenPictureDialogl.FileName);

Але у всіх цих випадках, якщо формат файлу не збігатиметься з передбачуваним, виникне помилка.

Аналогічно працює і метод SaveToFile з тією відмінністю, що застосований до Picture або Picture.Graphic він зберігає у файлі зображення будь-якого формату. Наприклад, якщо ви

доповните свою програму діалогом SavePictureDialog, збереження зображення у файлі можна організувати оператором

if SavePictureDialog1.Execute then

Image1.Picture.SaveToFile(SavePictureDialog1.FileName);

У цьому випадку користувач матиме можливість зберегти зображення будь-якого формату у файлі з новим ім'ям. Тільки при цьому, щоб не виникало надалі плутанини, розширення файлу, що зберігається, все-таки повинно відповідати формату збереженого зображення.

Абсолютно ідентично для зображень будь-якого формату працюватиме програма, якщо оператор збереження ви замініте на

Image1.Picture.Graphic.SaveToFile(SavePictureDialog1.FileName);

використовує властивість Picture.Graphic. А якщо вам відомий формат зображення, що зберігається в компоненті Image, то ви можете застосувати метод SaveToFile до властивостей Picture.Bitmap, Picture.Icon, і Picture.Metafile.

Для всіх розглянутих об'єктів Picture, Picture.Bitmap, Picture.Icon та Picture.Metafile визначають методи присвоєння значень об'єктів Assign.

Однак для BitMap, Icon та Metafile присвоювати можна лише значення однорідних об'єктів: відповідно бітових матриць, піктограм, метафайлів.

При спробі надання значенням різноманітних об'єктів буде генеруватися виняток EConvertError. Об'єкт Picture є універсальним, йому можна присвоїти значення об'єктів будь-яких з інших трьох класів. А значення Picture можна присвоїти лише об'єкту, тип якого збігається з типом об'єкта, що зберігається в ньому.

Метод **Assign** можна використовувати для обміну зображеннями з буфером Clipboard. Наприклад, оператор

Clipboard.Assign(Image1.Picture);

занесе до буфера обміну зображення, що зберігається в Image1. Аналогічно оператор

Image1.Picture.Assign(Clipboard) ;

прочитає в Image1 зображення, що у буфері обміну. Причому це може бути будь-яке зображення і навіть текст.

Треба тільки не забути під час роботи з буфером обміну вставити в оператор uses вашого модуля посилання на модуль

Clipbrd . Автоматично Delphi це посилання не вставляє.

Завдяки наявності канви Canvas компонент Image широко використовується не тільки для зберігання готових зображень, але і для побудови різноманітних графічних редакторів.

Основні властивості

Св-во	Оголошення / Опис

AutoSize	property AutoSize: Boolean; Вказує, чи автоматично змінюється розмір компонента, підлаштовуючись під розмір зображення. За замовчуванням значення false не підлаштовується.
Canvas	property Canvas: TCanvas; Визначає поверхню (полотно, канву) для малювання пером Реп та пензлем Brush, для накладання один на одного кількох зображень. Доступний лише для читання. Можна отримати доступ, якщо властивість Picture містить бітову матрицю.
Center	property Center: Boolean; Вказує, чи має зображення центруватися в полі компонента, якщо його розміри менше розмірів поля. При значенні false зображення знаходиться у верхньому лівому куті поля. Властивість не діє, якщо AutoSize встановлено в true або Stretch встановлено в true і Picture містить не піктограму.
IncrementalDisplay	property IncrementalDisplay: Boolean; Вказує, чи має зображення частково малюватись під час повільних операцій з великими зображеннями. Замість такого малювання часто можна використовувати індикацію процесу обробки подій OnProgress.
Picture	property Picture: TPicture; Визначає графічний об'єкт типу TPicture, що відображається. Може завантажуватись програмно або під час проектування за допомогою Picture Editor.

Stretch	property Stretch: Boolean; Вказує, чи потрібно змінюватися розміри зображення, підганяючись під розміри компонента. Зверніть увагу, що при зміні розмірів зображення буде спотворення, коли співвідношення сторін графічного об'єкта і компонента Image не збігаються.
Transparent	property Transparent: Boolean; Вказує, чи колір фону повинен бути прозорим, щоб крізь нього було видно нижчележаче зображення.

Основні методи

Жодних спеціальних методів у компоненті не оголошено. Компонент успадковує безліч методів від базового класу **TControl**.

Основні події

Подія	Опис
OnProgress	Події відбуваються при повільних процесах зміни графічного зображення та дозволяють побудувати індикатор ходу процесу.

Інші події успадковуються від класу **TControl**

. image delphi image timage вудзрш image delphi image delphi