目录

[1 Camera能用Camera从摄像头上捕捉图片和电影。可以操作捕捉和设置参数到图像上。我们能用VideoOutput显示捕捉的画面，需要设置Camera为VideoOutput的源 1](#_Toc397438846)

[2 CameraCapture允许我们捕捉图片，当图片准备好时给出通知或保存图片到磁盘。我们可以调节捕捉图片的分辨率和设置图片的保存位置。这个组件是Camera的子模块，不能在QML中直接创建使用imageCapture访问。 10](#_Toc397438847)

[3 CameraExposure允许我们调节曝光设置。像光圈、快门速度、测光、感光速度。它用于Camera的exposure属性，不能被单独构建。 16](#_Toc397438848)

[4 CameraFlash允许我们操作相机的闪光灯硬件和设置闪光模式。不是所有的相机都有闪光灯硬件。在一些情况下，闪光灯和照明共享硬件。它是Camera的一部分，不能直接建立，使用Camera的flash属性访问它。 20](#_Toc397438849)

[5 CameraFocus允许我们设置人工还是自动对焦。包括选择自动对焦时相机帧的任何部分的信息。它是Camera的一部分，使用Camera的focus属性访问，它不能直接被使用。 23](#_Toc397438850)

[6 CameraRecorder允许记录视频流到文件，可以为视频调节录像设置和metadata。它是Camera的一个属性，不能被单独构建 videoRecorder。 27](#_Toc397438851)

[7 CameraImageProcessing作为相机系统的一部份，提供相机图片的后期处理。包括白平衡调节、对比度调节、饱和度调节、锐化和去噪。它不能单独被使用，而是作为Camera的属性来访问。 33](#_Toc397438852)

1 Camera能用Camera从摄像头上捕捉图片和电影。可以操作捕捉和设置参数到图像上。我们能用VideoOutput显示捕捉的画面，需要设置Camera为VideoOutput的源

ImportStatement:   import QtMultimedia 5.0

Inherits:      Item

Properties

availability: enumeration

cameraState: enumeration

cameraStatus: enumeration

captureMode: enumeration

digitalZoom: real

errorCode :enumeration

errorString: string

lockStatus :enumeration

maximumDigitalZoom: real

maximumOpticalZoom: real

mediaObject: variant

opticalZoom: real

Signals

digitalZoomChanged(zoom)

error(errorCode,errorString)

lockStatusChanged()

manualWhiteBalanceChanged(qreal)

maximumDigitalZoomChanged(zoom)

maximumOpticalZoomChanged(zoom)

opticalZoomChanged(zoom)

stateChanged(state)

whiteBalanceModeChanged(Camera::WhiteBalanceMode)

Methods

searchAndLock()

start()

stop()

unlock()

DetailedDescription

Camera是QtMultimedia 5.0模块的一部分。

我们能用Camera从摄像头上捕捉图片和电影。可以操作捕捉和设置参数到图像上。我们能用VideoOutput显示捕捉的画面，需要设置Camera为VideoOutput的源。

示例如下：

importQtQuick 2.0

importQtMultimedia 5.0

Item {

    width: 640

    height: 360

    Camera {

        id: camera

        imageProcessing.whiteBalanceMode:CameraImageProcessing.WhiteBalanceFlash

        exposure {

            exposureCompensation: -1.0

            exposureMode:Camera.ExposurePortrait

        }

        flash.mode: Camera.FlashRedEyeReduction

        imageCapture {

            onImageCaptured: {

                photoPreview.source =preview  // Show the preview in an Image

            }

        }

    }

    VideoOutput {

        source: camera

        anchors.fill: parent

        focus : visible // to receive focus andcapture key events when visible

    }

    Image {

        id: photoPreview

    }

}

Camera的不同设置和功能栈被分布在Camera的不同子属性中。

| 属性 | 描述 |
| --- | --- |
| imageCapture | 捕捉图片的方法和属性。 |
| videoRecording | 捕捉电影的方法和属性。 |
| exposure | 调整曝光的方法和属性。（光圈、快门速度等） |
| focus | 调整对焦的方法和属性；提供自动对焦的反馈。 |
| flash | 控制闪光灯的方法和属性。 |
| imageProcessing | 调整图像处理参数的方法和属性。 |

基本的相机状态管理，错误报告和简单的缩放控制，Camera自身的属性足以支持。另外，我们可以使用mediaObject属性扩展标准的Qt Multimedia相机控制接口。

相机的许多设置值需要一些时间才能有效应用，有些值可能还存在限制，这些都取决于相机的硬件。一些相机可以设置成手动或自动。这些设置属性包含当前设置值。例如，当自动对焦被打开，焦点区域就暴露在焦点属性中。

PropertyDocumentation

availability: enumeration

相机的可用性。它是下面值中的一个。

| 值 | 描述 |
| --- | --- |
| Available | 相机可用 |
| Busy | 相机属于另一个进程，处于忙状态。 |
| Unavailable | 相机不可用（可能没有相机的硬件） |
| ResourceMissing | 相机因为缺少资源，不可用，可稍后在试。 |

cameraState: enumeration

相机的当前状态，可能是下面的属性之一：

| 值 | 描述 |
| --- | --- |
| UnloadedState | 相机的初始状态，相机没有加载。除了捕捉功能，相机的其他功能未知。基本上是上电后的状态，准备开始捕捉需要花费最长的时间。  在这个状态，支持的设置未知。但我们还能为捕捉设置编解码器，分辨率和帧率。 |
| LoadedState | 相机被加载并准备配置。在空闲时，我们可以查询相机的功能，设置捕捉的分辨率，编解码器等等。在这个状态，取景器没有被激活。 |
| ActiveState | 在这个状态，取景器已经被激活，相机已经为捕捉做好准备。 |

默认相机状态是ActiveState。

cameraStatus: enumeration

相机当前的状况，它是下面的其中一个值：

| 值 | 描述 |
| --- | --- |
| ActiveStatus | 相机已经启动，可以产生数据，取景器显示视频帧。依赖于后端，在ActiveState状态改变相机的设置（诸如捕捉模式，编解码器，分辨率），可能改变cameraStatus 到LoadedStatus和StartingStatus。当设置被应用，相机准备好时，cameraStatus回到ActiveStatus。 |
| StartingStatus | 相机开始过渡到Camera.ActiveState。此时相机还没有准备捕捉数据。 |
| StoppingStatus | 相机正处于从Camera.ActiveState过渡到Camera.LoadedState或Camera.UnloadedState。 |
| StandbyStatus | 相机在一段时间不活动后，进入Camera.LoadedState的节能待机模式的状态。 |
| LoadedStatus | 相机被加载，准备被设置。这个状况指明了相机设备被打开，能够查询支持的图片和影片的捕捉设置。（诸如分辨率、帧率、编解码器） |
| LoadingStatus | 相机正处于Camera.UnloadedState到Camera.LoadedState或Camera.ActiveState的过渡中。 |
| UnloadingStatus | 相机正处于从Camera.LoadedState或Camera.ActiveState过渡到Camera.UnloadedState中。 |
| UnloadedStatus | 相机一开始的状况，相机没有加载。相机支持的功能，包括捕捉设置为未知。 |
| UnavailableStatus | 相机或相机后端不可用。 |

captureMode: enumeration

相机的捕捉模式，它是下面的其中一个值：

| 值 | 描述 |
| --- | --- |
| CaptureViewfinder | 相机仅仅设置在取景器显示。 |
| CaptureStillImage | 相机捕捉图片。 |
| CaptureVideo | 相机捕捉视频。 |

digitalZoom: real

当前的数字变焦。

errorCode :enumeration

最后的错误码。

errorString: string

最后的错误的详细信息。

lockStatus :enumeration

相机针对所有请求的状态。属性是下面的一个值。

| 值 | 描述 |
| --- | --- |
| Unlocked | 应用对相机设置没有兴趣。相机保持同样的参数不变。大家使用共同的焦距，曝光调节，白平衡设置，保证取景器的图像是优质的。 |
| Searching | 应用请求设置相机聚焦，曝光和白平衡。searchAndLock()启动，这个状态指明相机正在聚焦或计算曝光和白平衡。 |
| Locked | 相机的焦距，曝光和白平衡被锁定。相机准备捕捉数据。应用可以检查曝光参数。锁状态意味着请求的参数将保持不变，除非再次请求更新参数。例如在连续聚焦状态，焦距被锁定。即使实际焦距变化，参数也不改变。 |

maximumDigitalZoom: real

支持的最大数码变焦。如果不支持数码变焦，此属性为1.0。

maximumOpticalZoom: real

支持的最大光学变焦，如果不支持光学变焦，此属性为1.0。

mediaObject: variant

相机的媒体对象。

opticalZoom: real

当前的光学变焦。

SignalDocumentation

digitalZoomChanged(zoom)

当数码变焦设置值变化时，发送这个信号。

error(errorCode,errorString)

当错误发生，这个信号被发射。errorCode是下面的一个值。错误可利用的描述在errorString中。

| 值 | 描述 |
| --- | --- |
| NoError | 没有错误发生 |
| CameraError | 有错误 |
| InvalidRequestError | 系统不支持请求的功能 |
| ServiceMissingError | 没有相机服务可用 |
| NotSupportedFeatureError | 不支持的特征 |

lockStatusChanged()

当锁状态变化时，发送这个信号。这发生在锁（例如自动变焦）完成或失败。

manualWhiteBalanceChanged(qreal)

当manualWhiteBalance属性变化时，发送这个信号。

maximumDigitalZoomChanged(zoom)

最大数字对焦改变，将发射这个信号。这一般发生在我们在图片和影片之间改变捕捉模式或改变捕捉设置时。

maximumOpticalZoomChanged(zoom)

当最大光学对焦发生变化时，发送这个信号。这一般发生在我们在图片和影片之间改变捕捉模式或改变捕捉设置时。

opticalZoomChanged(zoom)

当光学对焦发生变化时，发送这个信号。

stateChanged(state)

当相机状态发生变化时，发送这个信号。状态变化需要花一些时间，所以这个信号可能在请求发生后的某个时刻抵达。

whiteBalanceModeChanged(Camera::WhiteBalanceMode)

当whiteBalanceMode属性发生变化时，发送这个信号。

MethodDocumentation

searchAndLock()

开始焦距、曝光和白平衡的计算。这用于相机的焦距按钮被按下或相机的捕捉按钮半按。如果相机支持自动对焦，焦点区域可以通过focus属性获得。

start()

启动相机，取景帧可用，图片或电影可以被捕捉。

stop()

停止相机，但不处理相机栈

unlock()

解锁焦距，曝光和白平衡锁。

2 CameraCapture允许我们捕捉图片，当图片准备好时给出通知或保存图片到磁盘。我们可以调节捕捉图片的分辨率和设置图片的保存位置。这个组件是Camera的子模块，不能在QML中直接创建使用imageCapture访问。

ImportStatement:   import QtMultimedia 5.0

Properties

capturedImagePath: string

errorString: string

ready : bool

resolution :size

Signals

captureFailed(requestId,message)

imageCaptured(requestId,preview)

imageMetadataAvailable(requestId,key, value)

imageSaved(requestId,path)

Methods

cancelCapture()

capture()

captureToLocation(location)

setMetadata(key,value)

DetailedDescription

这个组件允许我们捕捉图片，当图片准备好时给出通知或保存图片到磁盘。我们可以调节捕捉图片的分辨率和设置图片的保存位置。这个组件是Camera的子模块，不能在QML中直接创建。

importQtQuick 2.0

importQtMultimedia 5.0

Item {

    width: 640

    height: 360

    Camera {

        id: camera

        imageCapture {

            onImageCaptured: {

                // Show the preview in an Image

                photoPreview.source = preview

            }

        }

    }

    VideoOutput {

        source: camera

        focus : visible // to receive focus andcapture key events when visible

        anchors.fill: parent

        MouseArea {

            anchors.fill: parent;

            onClicked:camera.imageCapture.capture();

        }

    }

    Image {

        id: photoPreview

    }

}

PropertyDocumentation

capturedImagePath: string

最后捕捉的图片的位置。

errorString: string

最后捕捉的错误信息，如果有错误的化。

ready : bool

相机是否准备好捕捉。当相机处于准备好的状态，就能调用capture()去捕捉图片。如果相机没有准备好，请求和相关联的所有设置就会被放入队列，等待尽可能快地被执行。

resolution :size

这个属性保存捕捉图片的分辨率/尺寸。如果为空，那么系统选择适当的分辨率。

SignalDocumentation

captureFailed(requestId,message)

当具有requestId这个ID号的捕捉请求出现错误时，发送这个信号。错误的详细信息保存在message中。

imageCaptured(requestId,preview)

当具有requestId这个ID号的请求捕捉了图片，又没有保存到磁盘时，这个信号被发射。preview是可以利用的图片的URL。

imageMetadataAvailable(requestId,key, value)

当requestId的请求有新的metadata可利用时，这个信号被发射。key和value为metadata的key和value。

imageSaved(requestId,path)

当requestId捕捉的图片被写入磁盘时，这个信号被发射。path是文件的路径，不是URL。

MethodDocumentation

cancelCapture()

取消图片捕捉请求。

capture()

开始图片捕捉。当捕捉完成，imageCaptured和imageSaved将会按各自的条件发射。

捕获的图片通常在默认的系统位置：

QStandardPaths::writableLocation(QStandardPaths::PicturesLocation)用于保存图片。

QStandardPaths::writableLocation(QStandardPaths::MoviesLocation)用于保存视频。

相机会保存所有捕捉参数（如曝光设置或图像处理参数），所以在capture()调用之后改变相机参数，不会影响之前的捕捉请求。

CameraCapture::capture返回requestId参数，用于imageExposed()、imageCaptured()、imageMetadataAvailable()和imageSaved()信号。

captureToLocation(location)

捕捉图片到指定位置。当捕捉完成，imageCaptured和imageSaved信号将根据条件发射。CameraCapture::captureToLocation返回requestId参数，用于和imageExposed()、imageCaptured()、imageMetadataAvailable()和imageSaved()信号匹配。如果应用不能写数据到指定位置，则发射错误信号。可能的原因为要写的目标路径是错误的，location不存在或没有写权限。

setMetadata(key,value)

为随后的捕捉设置一个特定的metadata，将key设置到value。同时参阅QMediaMetaData.

3 CameraExposure允许我们调节曝光设置。像光圈、快门速度、测光、感光速度。它用于Camera的exposure属性，不能被单独构建。

ImportStatement:   import QtMultimedia 5.0

Properties

aperture :real

exposureCompensation: real

exposureMode: enumeration

iso : int

manualAperture: real

manualIso :real

manualShutterSpeed: real

meteringMode: enumeration

shutterSpeed: real

spotMeteringPoint: QPointF

Methods

setAutoAperture()

setAutoIsoSensitivity()

setAutoShutterSpeed()

DetailedDescription

这个组件是QtMultimedia 5.0模块的一部分。CameraExposure允许我们调节曝光设置。像光圈、快门速度、测光、感光速度。它用于Camera的exposure属性，不能被单独构建。

importQtQuick 2.0

importQtMultimedia 5.0

Camera {

    id: camera

    exposure.exposureCompensation: -1.0

    exposure.exposureMode:Camera.ExposurePortrait

}

几个设置有自动和手动两种工作模式，在自动模式中，相机自己决定什么是合适的设置。在大多数情况下，这些设置能被指定的手动设置覆盖。

例如选择自动快门速度：

    camera.exposure.setAutoShutterSpeed()

或者指定快门速度：

    camera.exposure.manualShutterSpeed = 0.01// 10ms

我们仅能选择两种模式中的一个。

PropertyDocumentation

aperture :real

这个属性保存当前镜头光圈的F数量（焦距到有效孔径直径的比值）同时参阅manualAperture和setAutoAperture()。

exposureCompensation: real

自动计算曝光的调整值。单位是EV。

exposureMode: enumeration

相机的曝光模式，它是下面的一个值：

| 值 | 描述 |
| --- | --- |
| Camera.ExposureManual | 手动模式 |
| Camera.ExposureAuto | 自动模式 |
| Camera.ExposureNight | 夜模式 |
| Camera.ExposureBacklight | 背光曝光模式。 |
| Camera.ExposureSpotlight | 焦点曝光模式 |
| Camera.ExposureSports | 点曝光模式。 |
| Camera.ExposureSnow | 雪曝光模式 |
| Camera.ExposureBeach | 海滩曝光模式 |
| Camera.ExposureLargeAperture | 使用大光圈小景深模式 |
| Camera.ExposureSmallAperture | 用小光圈 |
| Camera.ExposurePortrait | 肖像曝光模式 |
| Camera.ExposureModeVendor | 基于设备特定值得曝光模式 |

iso : int

摄像头的感光灵敏度值。

manualAperture: real

人工设置捕捉照片的光圈值（F数目）如果小于0，相机自动决定光圈值。

同时参阅aperture和setAutoAperture()

manualIso :real

人工设置捕捉照片的感光度。如果指定负值，则相机自动决定感光度值。

同时参阅iso和setAutoIsoSensitivity()。

manualShutterSpeed: real

人工设置快门速度（以秒为单位）如果值小于0，则相机自动决定快门速度。

同时参阅shutterSpeed和setAutoShutterSpeed()。

meteringMode: enumeration

这个属性是相机的测光模式（曝光如何平衡）。它是下面的值之一。

| 值 | 描述 |
| --- | --- |
| Camera.MeteringMatrix | 样本点的矩阵用于测量曝光 |
| Camera.MeteringAverage | 平均值用于测量曝光。 |
| Camera.MeteringSpot | 一个指定的点用于测量曝光 |

shutterSpeed: real

几秒内的快门速度值。可用manualShutterSpeed属性和setAutoShutterSpeed()来影响这个值。

spotMeteringPoint: QPointF

在点测光模式中，用来测量曝光的点的坐标。典型的值默认为中心点(0.5，0.5)

MethodDocumentation

setAutoAperture()

切换成自动光圈选择。人工光圈被设置成-1.0。

setAutoIsoSensitivity()

切换自动感光度选择。人工ISO值被设置成-1。

setAutoShutterSpeed()

切换至自动快门速度。人工快门速度值被设置成-1.0

4 CameraFlash允许我们操作相机的闪光灯硬件和设置闪光模式。不是所有的相机都有闪光灯硬件。在一些情况下，闪光灯和照明共享硬件。它是Camera的一部分，不能直接建立，使用Camera的flash属性访问它。

ImportStatement:   import QtMultimedia 5.0

Properties

mode :enumeration

ready : bool

Signals

flashModeChanged(int)

flashReady(bool)

DetailedDescription

CameraFlash是QtMultimedia 5.0模块的一部分。这个组件允许我们操作相机的闪光灯硬件和设置闪光模式。不是所有的相机都有闪光灯硬件。在一些情况下，闪光灯和照明共享硬件。它是Camera的一部分，不能直接建立，使用Camera的flash属性访问它。

importQtQuick 2.0

importQtMultimedia 5.0

Camera {

    id: camera

    exposure.exposureCompensation: -1.0

    flash.mode: Camera.FlashRedEyeReduction

}

PropertyDocumentation

mode :enumeration

闪光灯的工作模式，它是下面的值的一种：

| 值 | 描述 |
| --- | --- |
| Camera.FlashOff | 关闭闪光灯 |
| Camera.FlashOn | 打开闪光灯 |
| Camera.FlashAuto | 自动闪光灯 |
| Camera.FlashRedEyeReduction | 红眼消除闪光灯。 |
| Camera.FlashFill | 闪光填阴影. |
| Camera.FlashTorch | 恒广源，用于聚焦和视频捕捉。 |
| Camera.FlashSlowSyncFrontCurtain | 结合较慢快门速度的闪光，这个模式用于更好的曝光远处的物体或运动模糊的影响。 |
| Camera.FlashSlowSyncRearCurtain | 和上面那条类似，但在曝光结束，闪光会取消。 |
| Camera.FlashManual | 手动设置闪光。 |

ready : bool

这个属性指明了闪光灯是否已充电。

SignalDocumentation

flashModeChanged(int)

当flashMode属性发生变化时，这个信号被发射。

flashReady(bool)

当QCameraExposure指明闪光准备好被使用时，这个信号发射。

5 CameraFocus允许我们设置人工还是自动对焦。包括选择自动对焦时相机帧的任何部分的信息。它是Camera的一部分，使用Camera的focus属性访问，它不能直接被使用。

ImportStatement:   import QtMultimedia 5.0

Properties

customFocusPoint: point

focusMode :enumeration

focusPointMode: enumeration

focusZones :list<focusZone>

Methods

boolisFocusModeSupported(mode)

boolisFocusPointModeSupported(mode)

DetailedDescription

CameraFocus是QtMultimedia 5.0模块的一部分。

这个组件允许我们设置人工还是自动对焦。包括选择自动对焦时相机帧的任何部分的信息。它是Camera的一部分，使用Camera的focus属性访问，它不能直接被使用。

importQtQuick 2.0

importQtMultimedia 5.0

Item {

    width: 640

    height: 360

    Camera {

        id: camera

        focus {

            focusMode: Camera.FocusMacro

            focusPointMode:Camera.FocusPointCustom

            customFocusPoint: Qt.point(0.2,0.2) // Focus relative to top-left corner

        }

    }

    VideoOutput {

        source: camera

       anchors.fill: parent

    }

}

PropertyDocumentation

customFocusPoint: point

指定焦点的中心，以帧的左上角顶点为(0,0)，帧的中心为(0.5,0.5)的焦点的坐标点。这个点只用在FocusPointCustom的集中模式。

focusMode :enumeration

当前相机的焦点模式，它是下面的值之一：

| 值 | 描述 |
| --- | --- |
| FocusManual | 手动或定焦模式。. |
| FocusHyperfocal | 关注超过焦距的距离，最大景深被激活，所有目标的距离在这个距离的一半到无限远之间将被设置。 |
| FocusInfinity | 焦点严格到无穷大。 |
| FocusAuto | 一次性自动对焦模式。 |
| FocusContinuous | 连续自动对焦模式。 |
| FocusMacro | 一次性和接近相机的物体对焦的模式。. |

联合多个模式是可能的，例如Camera.FocusMacro + Camera.FocusContinuous。在自动对焦模式，focusPointMode和focusZones属性控制自动对焦怎样执行。

focusPointMode: enumeration

当前相机的焦点模式。它被用于自动对焦模式，决定相机聚焦在什么上。如果当前的焦点模式是Camera.FocusPointCustom，customFocusPoint属性允许我们指定对焦到当前帧的那一部分。这个属性的值是下列之一：

| 值 | 描述 |
| --- | --- |
| FocusPointAuto | 自动选择一个或多个焦点。 |
| FocusPointCenter | 帧中心为焦点 |
| FocusPointFaceDetection | 聚焦在帧上. |
| FocusPointCustom | 聚焦在customFocusPoint  指定的点。 |

focusZones :list<focusZone>

当前相机焦点区域的列表。每个包含的区域指定同一坐标作为customFocusPoint。区域状态是下面的一个值之一。

| 值 | 描述 |
| --- | --- |
| Camera.FocusAreaUnused | 焦点区域未用在自动对焦上. |
| Camera.FocusAreaSelected | 焦点区域用于自动对焦，但不是焦点 |
| Camera.FocusAreaFocused | 焦点区域用于自动对焦，是焦点 |

VideoOutput{

    id: viewfinder

    source: camera

    //display focus areas on camera viewfinder:

    Repeater {

          model: camera.focus.focusZones

          Rectangle {

              border {

                  width: 2

                  color: status ==Camera.FocusAreaFocused ? "green" : "white"

              }

              color: "transparent"

              // Map from the relative,normalized frame coordinates

              property variant mappedRect:viewfinder.mapNormalizedRectToItem(area);

              x: mappedRect.x

              y: mappedRect.y

              width: mappedRect.width

              height: mappedRect.height

          }

    }

}

MethodDocumentation

boolisFocusModeSupported(mode) const

如果提供的模式支持集中模式，则返回真，否则返回假。

boolisFocusPointModeSupported(mode) const

如果提供的模式支持焦点模式，则返回真，否则返回假。

6 CameraRecorder允许记录视频流到文件，可以为视频调节录像设置和metadata。它是Camera的一个属性，不能被单独构建 videoRecorder。

ImportStatement:   import QtMultimedia 5.0

Properties

actualLocation: string

audioBitRate: int

audioChannels: int

audioCodec :string

audioEncodingMode: enumeration

audioSampleRate: int

duration :int

errorCode :enumeration

errorString: string

frameRate :qreal

mediaContainer: string

muted : bool

outputLocation: string

recorderState: enumeration

recorderStatus: enumeration

resolution :size

videoBitRate: int

videoCodec :string

videoEncodingMode: enumeration

Methods

record()

setMetadata(key,value)

stop()

DetailedDescription

CameraRecorder允许记录视频流到文件，可以为视频调节录像设置和metadata。

它是Camera的一个属性，不能被单独构建。

Camera {

    videoRecorder.audioEncodingMode:CameraRecorder.ConstantBitrateEncoding;

    videoRecorder.audioBitRate: 128000

    videoRecorder.mediaContainer:"mp4"

    // ...

}

这个组件为录像的每一部分都提供了许多不同设置（声音、视频和输出格式），它和控制静音和文件的保存位置。

PropertyDocumentation

actualLocation: string

最后保存的媒体的路径。它常常处于可利用的状态。当新的路径设置或新录像开始，它复位。

audioBitRate: int

用于录像的音频比特率。（每秒的比特数）

audioChannels: int

录像时编码的音频通道数（1是mono，2是立体声）

audioCodec :string

用于录像的音频编码器。典型的编码器是AAC或AMR-Wb

audioEncodingMode: enumeration

录制音频时的编码方法，它是下面的值之一：

| 值 | 描述 |
| --- | --- |
| ConstantQualityEncoding | 恒定质量编码，比特率会自动调整。这是默认值，audioBitRate将被忽略。 |
| ConstantBitRateEncoding | 恒定比特率编码，质量将会调整。它常常用于优化存储空间。 |
| AverageBitRateEncoding | 编码尽量保持在设置的比特率。实际可能比需要的大或者小。 |

audioSampleRate: int

录像时用于编码音频的声音采样率。

duration :int

以毫秒计算的最后的录像的时间。

errorCode :enumeration

最后的错误码。它是下面的一个值：

| 值 | 描述 |
| --- | --- |
| NoError | 没有错误。 |
| ResourceError | 设备没有准备好或不可用。 |
| FormatError | 当前格式不支持。 |
| OutOfSpaceError | 设备没有更多的空间。 |

errorString: string

最后的错误的详细信息。

frameRate :qreal

录像时每秒的帧数。

mediaContainer: string

录像时媒体的容器，典型值是mp4。

muted : bool

录像是否静音。

outputLocation: string

媒体的保存路径。如果路径为空，录像将自动命名录像，并保存在系统指定位置。

recorderState: enumeration

当前录像的状态，它是下面的一个值。

StoppedState：相机没有录像

RecordingState：相机正在录像

recorderStatus: enumeration

录像的状况，它是下面的值之一：

| 值 | 描述 |
| --- | --- |
| UnavailableStatus | 录像不被相机支持。 |
| UnloadedStatus | 录像机可用，但没有加载。 |
| LoadingStatus | 录像机正在初始化。 |
| LoadedStatus | 录像机完成初始化，准备录像 |
| StartingStatus | 录像被请求，但还没有激活。 |
| RecordingStatus | 录像被激活，录像中 |
| PausedStatus | 暂停录像 |
| FinalizingStatus | 录像停止，媒体文件被结束。 |

resolution :size

视频捕捉时的帧尺寸。

videoBitRate: int

录像时的视频比特率（每秒的比特数）。

videoCodec :string

录像时视频的编码器，典型值是H264

videoEncodingMode: enumeration

录像时的视频编码模式，它是下面的值之一：

| 值 | 描述 |
| --- | --- |
| ConstantQualityEncoding | 恒定质量编码，码率会变化。 |
| ConstantBitRateEncoding | 恒定码率编码，质量会变化。 |
| AverageBitRateEncoding | 平均编码。尝试保持比特流在设定值，实际可能大或小 |

MethodDocumentation

record()

开始录像

setMetadata(key,value)

为下一个录像设置metadata，为给定的键配上值。

stop()

停止录像。

7 CameraImageProcessing作为相机系统的一部份，提供相机图片的后期处理。包括白平衡调节、对比度调节、饱和度调节、锐化和去噪。它不能单独被使用，而是作为Camera的属性来访问。

ImportStatement:   import QtMultimedia 5.0

Properties

contrast :qreal

denoisingLevel: qreal

manualWhiteBalance: qreal

saturation :qreal

sharpeningLevel: qreal

whiteBalanceMode: enumeration

DetailedDescription

CameraImageProcessing作为相机系统的一部份，提供相机图片的后期处理。包括白平衡调节、对比度调节、饱和度调节、锐化和去噪。它不能单独被使用，而是作为Camera的属性来访问  imageProcessing。

importQtQuick 2.0

importQtMultimedia 5.0

Camera {

    id: camera

    imageProcessing {

        whiteBalanceMode:Camera.WhiteBalanceTungsten

        contrast: 0.66

        saturation: -0.5

    }

}

PropertyDocumentation

contrast :qreal

图片对比度调节。有效的调节值在-1.0到1.0之间，默认为0。

denoisingLevel: qreal

去噪调整值。有效值在-1.0到1.0之间。-1.0为去噪关闭。0为默认去噪级别，1.0为最大去噪级别。

manualWhiteBalance: qreal

手动白平衡模式时的颜色温度。单位是Kelvin。

saturation :qreal

饱和度调节，有效值在-1.0和1.0之间，默认值为0。

sharpeningLevel: qreal

锐化调节，有效值在-1.0到1.0之间，-1.0为锐化关闭，0为默认锐化级别，1.0为最大锐化级别。

whiteBalanceMode: enumeration

| 值 | 描述 |
| --- | --- |
| WhiteBalanceManual | 手动白平衡。 |
| WhiteBalanceAuto | 自动白平衡模式。 |
| WhiteBalanceSunlight | 日光白平衡模式。 |
| WhiteBalanceCloudy | 阴天白平衡模式。 |
| WhiteBalanceShade | 阴影下白平衡模式。 |
| WhiteBalanceTungsten | 钨光灯白平衡模式。 |
| WhiteBalanceFluorescent | 荧光灯白平衡模式。 |
| WhiteBalanceFlash | 闪光灯白平衡模式。 |
| WhiteBalanceSunset | 日落白平衡模式。 |
| WhiteBalanceVendor | 供应商指定白平衡模式。 |