OCRStudioSDK Library Reference

version 1.0

Generated on Thu Mar 21 2024 16:25:53 for OCRStudioSDK Library Reference by Doxygen 1.10.0

Thu Mar 21 2024 16:25:53

1 Class Documentation

1.1 ocrstudio::OCRStudioSDKDelegate Class Reference

Public Member Functions

- virtual ~**OCRStudioSDKDelegate** ()=default

  Virtual destructor.
- virtual void Callback (const char *json_message)=0

  Callback for receiving messages from processing session.

1.1.1 Detailed Description

Definition at line 19 of file ocr_studio_delegate.h.

1.1.2 Member Function Documentation

Callback()

```
virtual void ocrstudio::OCRStudioSDKDelegate::Callback (
            const char * json_message )  [pure virtual]
```

Callback for receiving messages from processing session.

Parameters

json_message	- callback message encoded in JSON string

1.2 ocrstudio::OCRStudioSDKException Class Reference

Public Member Functions

- virtual ~**OCRStudioSDKException** ()
  Non-trivial destructor.
- **OCRStudioSDKException** (const char *type, const char *msg)
  Main constructor.
- **OCRStudioSDKException** (const OCRStudioSDKException &copy)
  Copy constructor.
- const char * **Type** () const
  Returns exception type.
- const char * **Message** () const
  Returns exception message.

Private Attributes

- char * type_
  stored exception type
- char * msg_
  stored exception message

1.2.1 Detailed Description

Definition at line 19 of file ocr_studio_exception.h.

1.2.2 Member Data Documentation

type_

```
char* ocrstudio::OCRStudioSDKException::type_  [private]
```

stored exception type

Definition at line 39 of file ocr_studio_exception.h.

msg_

```
char* ocrstudio::OCRStudioSDKException::msg_  [private]
```

stored exception message

Definition at line 40 of file ocr_studio_exception.h.

1.3 ocrstudio::OCRStudioSDKImage Class Reference

Bitmap image class.

```
#include <ocr_studio_image.h>
```

Public Member Functions

- virtual ~**OCRStudioSDKImage** ()=default

  Default destructor.
- virtual OCRStudioSDKImage * DeepCopy () const =0

  Copies an image with copying of all pixels.
- virtual OCRStudioSDKImage * ShallowCopy () const =0

  Copies an image without copying the pixels, retaining internal memory reference. The operations with the copied image will be invalid after the source is deleted.
- virtual void **Clear** ()=0

  Clears the internal structure of the image.
- virtual int ExportPixelBufferLength () const =0

  Returns the required size of the export pixel buffer.
- virtual int ExportPixelBuffer (unsigned char *export_buffer, int export_buffer_length) const =0

  Copies the pixels into an external buffer. For any image the exported buffer pixels will have 8-bit channels (0 means lowest intensity, 255 means highest intensity). 1-channel images are exported as grayscale, 3-channel images are exported as RGB, other images are copied as-is.
- virtual OCRStudioSDKString ExportBase64JPEG () const =0

  Exports image as a JPEG buffer encoded in base64.
- virtual void Scale (int width, int height)=0

  Scales the internal image to a new size.
- virtual OCRStudioSDKImage * DeepCopyScaled (int width, int height) const =0

  Copies the image with scaling to a new size.
- virtual void CropByQuad (const char *quad_json, int width, int height)=0

  Crops an image quadrilateral to a new image, with a new provided size. If width or height is less or equal to zero, the size will be calculated approximately based on an input quadrilateral.
- virtual OCRStudioSDKImage * DeepCopyCroppedByQuad (const char *quad_json, int width, int height) const =0

  Copies an image cropped by a quadrilateral, with a new provided size. If width or height is less or equal to zero, the size will be calculated approximately based on an input quadrilateral.
- virtual void CropByRect (int x, int y, int width, int height)=0

  Crops an image to a rectangular region.
- virtual OCRStudioSDKImage * DeepCopyCroppedByRect (int x, int y, int width, int height) const =0

  Copies an image cropped to a rectangular region.
- virtual OCRStudioSDKImage * ShallowCopyCroppedByRect (int x, int y, int width, int height) const =0

  Shallow-copies an image cropped to a rectangular region. Operations on the resulting image are invalid after the source image is deleted.

- virtual void RotateByNinety (int num_rotations)=0

  Rotates the image clockwise by 90 degrees.
- virtual OCRStudioSDKImage * DeepCopyRotatedByNinety (int num_rotations) const =0

  Copies the image rotated clockwise by 90 degrees.
- virtual int **Width** () const =0

  Image width in pixels.
- virtual int **Height** () const =0

  Image height in pixels.
- virtual int **BytesPerLine** () const =0

  Size of the image row in bytes, including alignment.
- virtual int **Channels** () const =0

  The number of channels per pixel.
- virtual void * **UnsafeBufferPtr** () const =0

  Gets the pointer to the pixels buffer.
- virtual bool **OwnsPixelData** () const =0

  Whether this instance owns and will release pixel data.
- virtual void **ForcePixelDataOwnership** ()=0

  Forces pixel data ownership - for shallow images, copies all pixels.

Static Public Member Functions

- static int PagesCount (const char *filename)

  For multi-page images, returns the number of pages in an image file.
- static OCRStudioSDKString PageName (const char *filename, int page_number)

  For multi-page images, returns the filename of a particular page.
- static OCRStudioSDKImage * CreateEmpty ()

  Creates an empty image.
- static OCRStudioSDKImage * CreateFromFile (const char *filename, int page_number=0, int max_width=25000, int max_height=25000)

  Creates an image from file.
- static OCRStudioSDKImage * CreateFromFileBuffer (unsigned char *data, int data_size, int page_number=0, int max_width=25000, int max_height=25000)

  Creates an image from file loaded in a buffer.
- static OCRStudioSDKImage * CreateFromBase64FileBuffer (const char *base64_data, int page_number=0, int max_width=25000, int max_height=25000)

  Creates an image from file loaded in a buffer encoded in base64.
- static OCRStudioSDKImage * CreateFromPixelBuffer (unsigned char *data, int data_size, int width, int height, int bytes_per_line, int bytes_per_channel, OCRStudioSDKPixelFormat pixel_format)

  Creates an image from a pixel buffer, the content is copied.
- static OCRStudioSDKImage * CreateFromBuffer (unsigned char *data, int data_size, int width, int height, int bytes_per_line, int channels)

  Creates an image from a buffer, the content is copied.
- static OCRStudioSDKImage * CreateFromYUVSimple (unsigned char *yuv_data, int yuv_data_size, int width, int height)

  Creates an image from a simple YUV NV21 buffer.
- static OCRStudioSDKImage * CreateFromYUV (unsigned char *y_plane, int y_plane_size, int y_plane_row_stride, int y_plane_pixel_stride, unsigned char *u_plane, int u_plane_size, int u_plane_row_stride, int u_plane_pixel_stride, unsigned char *v_plane, int v_plane_size, int v_plane_row_stride, int v_plane_pixel_stride, int width, int height, OCRStudioSDKYUVFormat yuv_format)

  Creates an image from a universal YUV buffer.

1.3.1 Detailed Description

Bitmap image class.

Definition at line 50 of file ocr_studio_image.h.

1.3.2 Member Function Documentation

PagesCount()

```
static int ocrstudio::OCRStudioSDKImage::PagesCount (
            const char * filename )  [static]
```

For multi-page images, returns the number of pages in an image file.

Parameters

filename	- path to an image file

Returns

The number of pages in an image

PageName()

```
static OCRStudioSDKString ocrstudio::OCRStudioSDKImage::PageName (
            const char * filename,
            int page_number )  [static]
```

For multi-page images, returns the filename of a particular page.

Parameters

filename	- Filename of a particular image page
page_number	- page number, starting with 0

Returns

The string representation of a page filename

CreateEmpty()

```
static OCRStudioSDKImage * ocrstudio::OCRStudioSDKImage::CreateEmpty ( )  [static]
```

Creates an empty image.

Returns

Pointer to a new image, the ownership is relinquished.

CreateFromFile()

```
static OCRStudioSDKImage * ocrstudio::OCRStudioSDKImage::CreateFromFile (
            const char * filename,
            int page_number = 0,
            int max_width = 25000,
            int max_height = 25000 )  [static]
```

Creates an image from file.

Parameters

filename	- path to an image file (png, jpg, tif)
page_number	- page number, starting with 0
max_width	- maximum image width in pixels (0 for unrestricted)
max_height	- maximum image height in pixels (0 for unrestricted)

Returns

Pointer to a new image, the ownership is relinquished.

CreateFromFileBuffer()

```
static OCRStudioSDKImage * ocrstudio::OCRStudioSDKImage::CreateFromFileBuffer (
            unsigned char * data,
            int data_size,
            int page_number = 0,
            int max_width = 25000,
            int max_height = 25000 )  [static]
```

Creates an image from file loaded in a buffer.

Parameters

data	- pointer to a loaded file buffer
data_size	- size of the loaded file buffer
page_number	- page number, starting with 0
max_width	- maximum image width in pixels (0 for unrestricted)
max_height	- maximum image height in pixels (0 for unrestricted)

Returns

Pointer to a new image, the ownership is relinquished.

CreateFromBase64FileBuffer()

```
static OCRStudioSDKImage * ocrstudio::OCRStudioSDKImage::CreateFromBase64FileBuffer (
            const char * base64_data,
```

```
            int page_number = 0,
            int max_width = 25000,
            int max_height = 25000 )  [static]
```

Creates an image from file loaded in a buffer encoded in base64.

Parameters

base64_data	- file buffer encoded as a base64 string
page_number	- page number, starting with 0
max_width	- maximum image width in pixels (0 for unrestricted)
max_height	- maximum image height in pixels (0 for unrestricted)

Returns

Pointer to a new image, the ownership is relinquished.

CreateFromPixelBuffer()

```
static OCRStudioSDKImage * ocrstudio::OCRStudioSDKImage::CreateFromPixelBuffer (
            unsigned char * data,
            int data_size,
            int width,
            int height,
            int bytes_per_line,
            int bytes_per_channel,
            OCRStudioSDKPixelFormat pixel_format )  [static]
```

Creates an image from a pixel buffer, the content is copied.

Parameters

data	- pointer to a pixels buffer
data_size	- size of the pixels buffer
width	- width of the image in pixels
height	- height of the image in pixels
bytes_per_line	- size of an image row in bytes (including alignment)
bytes_per_channel	- size of a pixel component in bytes
pixel_format	- pixel format

Returns

Pointer to a new image, the ownership is relinquished.

CreateFromBuffer()

```
static OCRStudioSDKImage * ocrstudio::OCRStudioSDKImage::CreateFromBuffer (
            unsigned char * data,
```

```
            int data_size,
            int width,
            int height,
            int bytes_per_line,
            int channels )  [static]
```

Creates an image from a buffer, the content is copied.

Parameters

data	- pointer to a pixels buffer
data_size	- size of the pixels buffer
width	- width of the image in pixels
height	- height of the image in pixels
bytes_per_line	- size of an image row in bytes (including alignment)
channels	number of channels per-pixel

Returns

Pointer to a new image, the ownership is relinquished.

CreateFromYUVSimple()

```
static OCRStudioSDKImage * ocrstudio::OCRStudioSDKImage::CreateFromYUVSimple (
            unsigned char * yuv_data,
            int yuv_data_size,
            int width,
            int height )  [static]
```

Creates an image from a simple YUV NV21 buffer.

Parameters

yuv_data	- pointer to YUV NV21 buffer
yuv_data_size	- size of the YUV NV21 buffer
width	- width of the image in pixels
height	- height of the image in pixels

Returns

Pointer to a new image, the ownership is relinquished.

CreateFromYUV()

```
static OCRStudioSDKImage * ocrstudio::OCRStudioSDKImage::CreateFromYUV (
            unsigned char * y_plane,
            int y_plane_size,
            int y_plane_row_stride,
```

```
int y_plane_pixel_stride,
unsigned char * u_plane,
int u_plane_size,
int u_plane_row_stride,
int u_plane_pixel_stride,
unsigned char * v_plane,
int v_plane_size,
int v_plane_row_stride,
int v_plane_pixel_stride,
int width,
int height,
OCRStudioSDKYUVFormat yuv_format )  [static]
```

Creates an image from a universal YUV buffer.

Parameters

y_plane	- pointer to Y plane buffer
y_plane_size	- Y plane buffer size
y_plane_row_stride	- Y plane row stride
y_plane_pixel_stride	- Y plane pixel stride
u_plane	- pointer to U plane buffer
u_plane_size	- U plane buffer size
u_plane_row_stride	- U plane row stride
u_plane_pixel_stride	- U plane pixel stride
v_plane	- pointer to V plane buffer
v_plane_size	- V plane buffer size
v_plane_row_stride	- V plane row stride
v_plane_pixel_stride	- V plane pixel stride
width	- image width in pixels
height	- image height in pixels
yuv_format	- YUV format specification

Returns

Pointer to a new image, the ownership is relinquished.

DeepCopy()

```
virtual OCRStudioSDKImage * ocrstudio::OCRStudioSDKImage::DeepCopy ( ) const  [pure virtual]
```

Copies an image with copying of all pixels.

Returns

Pointer to a new copied image, the ownership is relinquished.

ShallowCopy()

```
virtual OCRStudioSDKImage * ocrstudio::OCRStudioSDKImage::ShallowCopy ( ) const  [pure virtual]
```

Copies an image without copying the pixels, retaining internal memory reference. The operations with the copied image will be invalid after the source is deleted.

Returns

Pointer to a new copied image, the ownership is relinquished.

ExportPixelBufferLength()

```
virtual int ocrstudio::OCRStudioSDKImage::ExportPixelBufferLength ( ) const  [pure virtual]
```

Returns the required size of the export pixel buffer.

Returns

Number of required bytes

ExportPixelBuffer()

```
virtual int ocrstudio::OCRStudioSDKImage::ExportPixelBuffer (
            unsigned char * export_buffer,
            int export_buffer_length ) const  [pure virtual]
```

Copies the pixels into an external buffer. For any image the exported buffer pixels will have 8-bit channels (0 means lowest intensity, 255 means highest intensity). 1-channel images are exported as grayscale, 3-channel images are exported as RGB, other images are copied as-is.

Parameters

export_buffer	- pointer to an output pixels buffer
export_buffer_length	- available buffer size. Must be at least the size returned by the ExportPixelBufferLength() method.

Returns

The number of written bytes

ExportBase64JPEG()

```
virtual OCRStudioSDKString ocrstudio::OCRStudioSDKImage::ExportBase64JPEG ( ) const  [pure
virtual]
```

Exports image as a JPEG buffer encoded in base64.

Returns

Base64 JPEG encoding of an image in a OCRStudioSDKString form

Scale()

```
virtual void ocrstudio::OCRStudioSDKImage::Scale (
            int width,
            int height )  [pure virtual]
```

Scales the internal image to a new size.

Parameters

width	- new width of the image in pixels
height	- new height of the image in pixels

DeepCopyScaled()

```
virtual OCRStudioSDKImage * ocrstudio::OCRStudioSDKImage::DeepCopyScaled (
            int width,
            int height ) const  [pure virtual]
```

Copies the image with scaling to a new size.

Parameters

width	- new width of the image in pixels
height	- new height of the image in pixels

Returns

Pointer to a new scaled image, the ownership is relinquished.

CropByQuad()

```
virtual void ocrstudio::OCRStudioSDKImage::CropByQuad (
            const char * quad_json,
            int width,
            int height )  [pure virtual]
```

Crops an image quadrilateral to a new image, with a new provided size. If width or height is less or equal to zero, the size will be calculated approximately based on an input quadrilateral.

Parameters

quad_json	- JSON representation of a quadrangle coordinates, in form [[x1, y1], [x2, y2], [x3, y3], [x4, y4]]
width	- new width of the image in pixels (or <= 0 for size autoselection)
height	- new height of the image in pixels (or <= 0 for size autoselection)

DeepCopyCroppedByQuad()

```
virtual OCRStudioSDKImage * ocrstudio::OCRStudioSDKImage::DeepCopyCroppedByQuad (
            const char * quad_json,
            int width,
            int height ) const  [pure virtual]
```

Copies an image cropped by a quadrilateral, with a new provided size. If width or height is less or equal to zero, the size will be calculated approximately based on an input quadrilateral.

Parameters

quad_json	- JSON representation of a quadrangle coordinates, in form [[x1, y1], [x2, y2], [x3, y3], [x4, y4]]
width	- new width of the image in pixels (or <= 0 for size autoselection)
height	- new height of the image in pixels (or <= 0 for size autoselection)

Returns

Pointer to a new cropped image, the ownership is relinquished.

CropByRect()

```
virtual void ocrstudio::OCRStudioSDKImage::CropByRect (
            int x,
            int y,
            int width,
            int height )  [pure virtual]
```

Crops an image to a rectangular region.

Parameters

x	- horizontal coordinate of the top-left corner
y	- vertical coordinate of the top-left corner
width	- width of the rectangle
height	- height of the rectangle

DeepCopyCroppedByRect()

```
virtual OCRStudioSDKImage * ocrstudio::OCRStudioSDKImage::DeepCopyCroppedByRect (
            int x,
            int y,
            int width,
            int height ) const  [pure virtual]
```

Copies an image cropped to a rectangular region.

Parameters

x	- horizontal coordinate of the top-left corner

Parameters

y	- vertical coordinate of the top-left corner
width	- width of the rectangle
height	- height of the rectangle

Returns

Pointer to a new cropped image, the ownership is relinquished.

ShallowCopyCroppedByRect()

```
virtual OCRStudioSDKImage * ocrstudio::OCRStudioSDKImage::ShallowCopyCroppedByRect (
            int x,
            int y,
            int width,
            int height ) const  [pure virtual]
```

Shallow-copies an image cropped to a rectangular region. Operations on the resulting image are invalid after the source image is deleted.

Parameters

x	- horizontal coordinate of the top-left corner
y	- vertical coordinate of the top-left corner
width	- width of the rectangle
height	- height of the rectangle

Returns

Pointer to a new cropped image, the ownership is relinquished.

RotateByNinety()

```
virtual void ocrstudio::OCRStudioSDKImage::RotateByNinety (
            int num_rotations )  [pure virtual]
```

Rotates the image clockwise by 90 degrees.

Parameters

num_rotations	- the number of times the rotation is performed

DeepCopyRotatedByNinety()

```
virtual OCRStudioSDKImage * ocrstudio::OCRStudioSDKImage::DeepCopyRotatedByNinety (
            int num_rotations ) const  [pure virtual]
```

Copies the image rotated clockwise by 90 degrees.

Parameters

num_rotations	- the number of times the rotation is performed

Returns

Pointer to a new rotated image, the ownership is relinquished.

1.4 ocrstudio::OCRStudioSDKInstance Class Reference

Main recognition engine class containing configuration for creating recognition sessions.

`#include <ocr_studio_instance.h>`

Public Member Functions

- virtual ~**OCRStudioSDKInstance** ()=default

  Default destructor.
- virtual const char * Description () const =0

  Returns a description of a configured engine in JSON format.
- virtual OCRStudioSDKSession * CreateSession (const char *authorization_signature, const char *json_↩ session_params, OCRStudioSDKDelegate *callback_delegate=nullptr) const =0

  Creates a processing session with the provided parameters.

Static Public Member Functions

- static OCRStudioSDKInstance * CreateStandalone (const char *json_instance_init_params=nullptr)

  Creates a new recognition engine instance from an internal configuration, embedded inside the library, if one is available. If no configuration is embedded inside, the method will throw an exception.
- static OCRStudioSDKInstance * CreateFromPath (const char *configuration_filename, const char *json_↩ instance_init_params=nullptr)

  Creates a new recognition engine instance from a configuration file (a binary file with an extension '.ocr').
- static OCRStudioSDKInstance * CreateFromBuffer (unsigned char *configuration_buffer, int configuration↩ _buffer_size, const char *json_instance_init_params=nullptr)

  Creates a new recognition engine instance from a configuration buffer (a binary buffer where the configuration file is loaded).
- static const char * **LibraryVersion** ()

  Returns a string representation of the OCRStudioSDK library version.

1.4.1 Detailed Description

Main recognition engine class containing configuration for creating recognition sessions.

Definition at line 25 of file ocr_studio_instance.h.

1.4.2 Member Function Documentation

CreateStandalone()

```
static OCRStudioSDKInstance * ocrstudio::OCRStudioSDKInstance::CreateStandalone (
            const char * json_instance_init_params = nullptr )  [static]
```

Creates a new recognition engine instance from an internal configuration, embedded inside the library, if one is available. If no configuration is embedded inside, the method will throw an exception.

Parameters

json_instance_init_params	- optional JSON with initialization parameters, in the following format (all keys are optional): { "enable_lazy_initialization": (bool), "enable_delayed_initialization": (bool), "initialization_num_threads": (int >= 0) }

Returns

Pointer to a new instance object, the ownership is relinquished.

CreateFromPath()

```
static OCRStudioSDKInstance * ocrstudio::OCRStudioSDKInstance::CreateFromPath (
            const char * configuration_filename,
            const char * json_instance_init_params = nullptr )  [static]
```

Creates a new recognition engine instance from a configuration file (a binary file with an extension '.ocr').

Parameters

configuration_filename	- path to a configuration file *.ocr
json_instance_init_params	- optional JSON with initialization parameters, in the following format (all keys are optional): { "enable_lazy_initialization": (bool), "enable_delayed_initialization": (bool), "initialization_num_threads": (int >= 0) }

Returns

Pointer to a new instance object, the ownership is relinquished.

CreateFromBuffer()

```
static OCRStudioSDKInstance * ocrstudio::OCRStudioSDKInstance::CreateFromBuffer (
            unsigned char * configuration_buffer,
            int configuration_buffer_size,
            const char * json_instance_init_params = nullptr )  [static]
```

Creates a new recognition engine instance from a configuration buffer (a binary buffer where the configuration file is loaded).

Parameters

configuration_buffer	- pointer to a binary configuration buffer
configuration_buffer_size	- size of the configuration buffer in bytes
json_instance_init_params	- optional JSON with initialization parameters, in the following format (all keys are optional): { "enable_lazy_initialization": (bool), "enable_delayed_initialization": (bool), "initialization_num_threads": (int >= 0) }

Returns

Pointer to a new instance object, the ownership is relinquished.

Description()

```
virtual const char * ocrstudio::OCRStudioSDKInstance::Description ( ) const  [pure virtual]
```

Returns a description of a configured engine in JSON format.

Returns

a JSON description in the following format: { "session_types": [(list of available session types)], "target_groups": [// present if there are target-oriented sessions { "target_group_type": "(group_type_name)", "targets": ["(target_name)", ...], "target_masks": ["(target_mask)", ...] }, ...] }

CreateSession()

```
virtual OCRStudioSDKSession * ocrstudio::OCRStudioSDKInstance::CreateSession (
            const char * authorization_signature,
            const char * json_session_params,
            OCRStudioSDKDelegate * callback_delegate = nullptr ) const  [pure virtual]
```

Creates a processing session with the provided parameters.

Parameters

authorization_signature	- signature of an authorized SDK user
json_session_params	- parameters of the created session, encoded in JSON in the following format: { "session_type": "(session_type)", "target_group_type": "(group_type_name)", "target_masks": ["(target_name_or_mask", ...], // optional, single string permitted "options": { // optional "(option_name)": "(option_value)", ... }, "output_modes": ["(output_mode)", ...] }
callback_delegate	- optional pointer to an implemented instance of a delegate for receiving runtime messages.

Returns

Pointer to a new session, the ownership is relinquished.

1.5 ocrstudio::OCRStudioSDKItem Class Reference

A constituent object of a recognized or analyzed target.

```
#include <ocr_studio_result.h>
```

Public Member Functions

- virtual ~**OCRStudioSDKItem** ()=default

  Default destructor.
- virtual OCRStudioSDKItem * DeepCopy () const =0

  Copies an item with copying of all internal information.
- virtual const char * **Type** () const =0

  Returns the type of the item.
- virtual const char * **Name** () const =0

  Returns the name of the item.
- virtual const char * **Value** () const =0

  Returns the string representatio of the value of the item.
- virtual double **Confidence** () const =0

  Returns the item confidence value (doubole in range [0.0, 1.0])
- virtual bool **Accepted** () const =0

  Returns the item accept flag.
- virtual const char * Attributes () const =0

  Returns the attributes of the item in JSON format.
- virtual bool **HasImage** () const =0

  Returns true iff the item has an associated image.
- virtual const OCRStudioSDKImage & **Image** () const =0

  Returns the associated image.
- virtual const char * **Description** () const =0

  Returns a detailed JSON description (format depends on the type)

1.5.1 Detailed Description

A constituent object of a recognized or analyzed target.

Definition at line 24 of file ocr_studio_result.h.

1.5.2 Member Function Documentation

DeepCopy()

```
virtual OCRStudioSDKItem * ocrstudio::OCRStudioSDKItem::DeepCopy ( ) const  [pure virtual]
```

Copies an item with copying of all internal information.

Returns

Pointer to a new item structure, the ownership is relinquished.

Attributes()

```
virtual const char * ocrstudio::OCRStudioSDKItem::Attributes ( ) const  [pure virtual]
```

Returns the attributes of the item in JSON format.

Returns

a JSON attributes map in the following format: { "(attribute_name)": "(attribute_value)" }

1.6 ocrstudio::OCRStudioSDKItemIterator Class Reference

Map-like iterator for a collection of OCRStudioSDKItem objects.

`#include <ocr_studio_result.h>`

Public Member Functions

- ~**OCRStudioSDKItemIterator** ()

  Non-trivial destructor.
- **OCRStudioSDKItemIterator** (const OCRStudioSDKItemIterator &copy)

  Copy constructor.
- OCRStudioSDKItemIterator & **operator=** (const OCRStudioSDKItemIterator &other)

  Assignment operator.
- bool **IsEqualTo** (const OCRStudioSDKItemIterator &other) const

  Returns true iff the instances point to the same item.
- bool **operator==** (const OCRStudioSDKItemIterator &other) const

  Equality operator.
- bool **operator!=** (const OCRStudioSDKItemIterator &other) const

  Inequality operator.
- OCRStudioSDKItemIterator **Next** () const

  Returns the iterator to the next item in the collection.
- void **Step** ()

  Moves the iterator to the next item in the collection.
- void **operator++** ()

  Moves the iterator to the next item in the collection.
- const char * **Key** () const

  Returns the key of the item in the collection.
- const OCRStudioSDKItem & **Item** () const

  Returns the item to which the iterator points (const ref)

Static Public Member Functions

- static OCRStudioSDKItemIterator **CreateFromImplementation** (const OCRStudioSDKItemIterator↩ Implementation &rimpl)

  Creates an OCRStudioSDKItemIterator object from its internal implementation.

Private Member Functions

- **OCRStudioSDKItemIterator** (const OCRStudioSDKItemIteratorImplementation &rimpl)

  Private constructor from an internal implementation.

Private Attributes

- OCRStudioSDKItemIteratorImplementation * pimpl_

  Internal implementation.

1.6.1 Detailed Description

Map-like iterator for a collection of OCRStudioSDKItem objects.

Definition at line 79 of file ocr_studio_result.h.

1.6.2 Member Data Documentation

pimpl_

```
OCRStudioSDKItemIteratorImplementation* ocrstudio::OCRStudioSDKItemIterator::pimpl_  [private]
```

Internal implementation.

Definition at line 124 of file ocr_studio_result.h.

1.7 ocrstudio::OCRStudioSDKResult Class Reference

Main session result class - container with full session result.

```
#include <ocr_studio_result.h>
```

Public Member Functions

- virtual ~**OCRStudioSDKResult** ()=default
  Default destructor.
- virtual OCRStudioSDKResult * DeepCopy () const =0
  Copies a result with copying of all internal information.
- virtual int TargetsCount () const =0
  Returns the number of stored targets.
- virtual const OCRStudioSDKTarget & TargetByIndex (int target_index) const =0
  Returns a specific stored target by its index.
- virtual bool AllTargetsFinal () const =0
  Returns true if all targets can be considered final.

1.7.1 Detailed Description

Main session result class - container with full session result.

Definition at line 214 of file ocr_studio_result.h.

1.7.2 Member Function Documentation

DeepCopy()

```
virtual OCRStudioSDKResult * ocrstudio::OCRStudioSDKResult::DeepCopy ( ) const  [pure virtual]
```

Copies a result with copying of all internal information.

Returns

Pointer to a new result structure, the ownership is relinquished.

TargetsCount()

```
virtual int ocrstudio::OCRStudioSDKResult::TargetsCount ( ) const  [pure virtual]
```

Returns the number of stored targets.

Returns

The number of stored targets

TargetByIndex()

```
virtual const OCRStudioSDKTarget & ocrstudio::OCRStudioSDKResult::TargetByIndex (
            int target_index ) const  [pure virtual]
```

Returns a specific stored target by its index.

Parameters

target_index	- 0-based index of a stored target

Returns

Specific stored target (constant reference)

AllTargetsFinal()

```
virtual bool ocrstudio::OCRStudioSDKResult::AllTargetsFinal ( ) const  [pure virtual]
```

Returns true if all targets can be considered final.

Returns

All targets can be considered final

1.8 ocrstudio::OCRStudioSDKSession Class Reference

Main processing session class - agent for performing image analysis.

`#include <ocr_studio_session.h>`

Public Member Functions

- virtual ~**OCRStudioSDKSession** ()=default

  Default destructor.
- virtual const char * Description () const =0

  Returns a description of a created session in JSON format.
- virtual void ProcessImage (const OCRStudioSDKImage &image)=0

  Processes an input image or video frame, updates the internal session state.
- virtual const OCRStudioSDKResult & CurrentResult () const =0

  Returns the current accumulated result.
- virtual void **Reset** ()=0

  Resets the state of the session to the initial one.

1.8.1 Detailed Description

Main processing session class - agent for performing image analysis.

Definition at line 24 of file ocr_studio_session.h.

1.8.2 Member Function Documentation

Description()

```
virtual const char * ocrstudio::OCRStudioSDKSession::Description ( ) const  [pure virtual]
```

Returns a description of a created session in JSON format.

Returns

a JSON description in the following format: { "session_type": "(session_type)", "target_group_type": "(group↩ _type_name)", "targets": ["(target_name)", ...], "options": { "(option_name)": "(option_value)", ... }, "output_↩ modes": ["(output_mode)", ...] }

ProcessImage()

```
virtual void ocrstudio::OCRStudioSDKSession::ProcessImage (
            const OCRStudioSDKImage & image )  [pure virtual]
```

Processes an input image or video frame, updates the internal session state.

Parameters

image	- input image to be processed

CurrentResult()

```
virtual const OCRStudioSDKResult & ocrstudio::OCRStudioSDKSession::CurrentResult ( ) const
[pure virtual]
```

Returns the current accumulated result.

Returns

Current accumulated session result (constant reference to an internal structure, the memory is owned by the session)

1.9 ocrstudio::OCRStudioSDKString Class Reference

Public Member Functions

- ~**OCRStudioSDKString** ()

Non-trivial destructor.

- **OCRStudioSDKString** ()

  Default constructor.

- **OCRStudioSDKString** (const char *c_str)

  Constructor from a C-string.

- **OCRStudioSDKString** (const OCRStudioSDKString &copy)

  Copy constructor.

- OCRStudioSDKString & **operator=** (const OCRStudioSDKString &other)

  Assignment operator.

- OCRStudioSDKString & **operator+=** (const OCRStudioSDKString &other)

  Inplace concatenation.

- OCRStudioSDKString **operator+** (const OCRStudioSDKString &other) const

  General concatenation.

- const char * **CStr** () const

  Returns internal c-string.

- int **Size** () const

  Returns number of bytes stored.

Private Attributes

- int size_

  length of the internal string in bytes

- char * str_

  internal c-string

1.9.1 Detailed Description

Definition at line 19 of file ocr_studio_string.h.

1.9.2 Member Data Documentation

size_

```
int ocrstudio::OCRStudioSDKString::size_  [private]
```

length of the internal string in bytes

Definition at line 53 of file ocr_studio_string.h.

str_

```
char* ocrstudio::OCRStudioSDKString::str_  [private]
```

internal c-string

Definition at line 54 of file ocr_studio_string.h.

1.10 ocrstudio::OCRStudioSDKTarget Class Reference

Recognition or analysis target (document or other object)

`#include <ocr_studio_result.h>`

Public Member Functions

- virtual ~**OCRStudioSDKTarget** ()=default

  Default destructor.
- virtual OCRStudioSDKTarget * DeepCopy () const =0

  Copies a target with copying of all internal information.
- virtual const char * Description () const =0

  Returns a description of a target in JSON format.
- virtual int ItemsCountByType (const char *item_type) const =0

  Returns the number of items with a provided item type.
- virtual bool HasItem (const char *item_type, const char *item_name) const =0

  Checks whether ther is an item of a specified type with a specified item name.
- virtual const OCRStudioSDKItem & Item (const char *item_type, const char *item_name) const =0

  Returns a specific item.
- virtual OCRStudioSDKItemIterator ItemsBegin (const char *item_type) const =0

  Returns a map-like iterator to the start of the collection of items with the specified type.
- virtual OCRStudioSDKItemIterator ItemsEnd (const char *item_type) const =0

  Returns a map-like iterator to the end of the collection of items with the specified type.
- virtual bool IsFinal () const =0

  Returns true if the target can be considered final.

1.10.1 Detailed Description

Recognition or analysis target (document or other object)

Definition at line 132 of file ocr_studio_result.h.

1.10.2 Member Function Documentation

DeepCopy()

```
virtual OCRStudioSDKTarget * ocrstudio::OCRStudioSDKTarget::DeepCopy ( ) const  [pure virtual]
```

Copies a target with copying of all internal information.

Returns

Pointer to a new target structure, the ownership is relinquished.

Description()

```
virtual const char * ocrstudio::OCRStudioSDKTarget::Description ( ) const  [pure virtual]
```

Returns a description of a target in JSON format.

Returns

a JSON description in the following format: { "target_type": "(target_type_name)", "specific_type": "(specific↩ _type_name)", "item_types": ["(item_type_name)", ...], "attributes": { "(attribute_name)": "(attribute_value)", ... } }

ItemsCountByType()

```
virtual int ocrstudio::OCRStudioSDKTarget::ItemsCountByType (
            const char * item_type ) const  [pure virtual]
```

Returns the number of items with a provided item type.

Parameters

item_type	- name of the item type

Returns

The number of items of the specified type. The number of items is zero if the stored collection has zero size or if the specified item type is not supported for the returned target type

HasItem()

```
virtual bool ocrstudio::OCRStudioSDKTarget::HasItem (
            const char * item_type,
            const char * item_name ) const  [pure virtual]
```

Checks whether ther is an item of a specified type with a specified item name.

Parameters

item_type	- name of the item type
item_name	- name of the specific item

Returns

true iff there exists an item with a provided name in the collection of items of the provided type

Item()

```
virtual const OCRStudioSDKItem & ocrstudio::OCRStudioSDKTarget::Item (
```

```
            const char * item_type,
            const char * item_name ) const  [pure virtual]
```

Returns a specific item.

Parameters

item_type	- name of the item type
item_name	- name of the specific item

Returns

Specific item object (constant reference)

ItemsBegin()

```
virtual OCRStudioSDKItemIterator ocrstudio::OCRStudioSDKTarget::ItemsBegin (
            const char * item_type ) const  [pure virtual]
```

Returns a map-like iterator to the start of the collection of items with the specified type.

Parameters

item_type	- name of the item type

Returns

A map-like 'begin' iterator to the collection of items

ItemsEnd()

```
virtual OCRStudioSDKItemIterator ocrstudio::OCRStudioSDKTarget::ItemsEnd (
            const char * item_type ) const  [pure virtual]
```

Returns a map-like iterator to the end of the collection of items with the specified type.

Parameters

item_type	- name of the item type

Returns

A map-like 'end' iterator to the collection of items

IsFinal()

```
virtual bool ocrstudio::OCRStudioSDKTarget::IsFinal ( ) const  [pure virtual]
```

Returns true if the target can be considered final.

Returns

Can the target be considered final

2 File Documentation

2.1 ocr_studio_delegate.h File Reference

Feedback base class, allows to receive runtime messages from OCRStudioSDKSession.

Classes

- class ocrstudio::OCRStudioSDKDelegate

2.1.1 Detailed Description

Feedback base class, allows to receive runtime messages from OCRStudioSDKSession.

Definition in file ocr_studio_delegate.h.

2.1.2 Macro Definition Documentation

OCRSTUDIOSDK_OCR_STUDIO_DELEGATE_H_INCLUDED

```
#define OCRSTUDIOSDK_OCR_STUDIO_DELEGATE_H_INCLUDED
```

Definition at line 13 of file ocr_studio_delegate.h.

2.2 ocr_studio_delegate.h

Go to the documentation of this file.

```
00001
00011 #pragma once
00012 #ifndef OCRSTUDIOSDK_OCR_STUDIO_DELEGATE_H_INCLUDED
00013 #define OCRSTUDIOSDK_OCR_STUDIO_DELEGATE_H_INCLUDED
00014
00015 #include <ocrstudiosdk/ocr_studio_export.h>
00016
00017 namespace ocrstudio {
00018
00019 class OCR_STUDIO_SDK_DLL_EXPORT OCRStudioSDKDelegate {
00020 public:
00022   virtual ~OCRStudioSDKDelegate() = default;
00023
00028   virtual void Callback(const char* json_message) = 0;
00029 };
00030
00031 } // namespace ocrstudio
00032
00033 #endif // OCRSTUDIOSDK_OCR_STUDIO_DELEGATE_H_INCLUDED
```

2.3 ocr_studio_exception.h File Reference

Main C++ exception class.

Classes

- class ocrstudio::OCRStudioSDKException

2.3.1 Detailed Description

Main C++ exception class.

Definition in file ocr_studio_exception.h.

2.3.2 Macro Definition Documentation

OCRSTUDIOSDK_OCR_STUDIO_EXCEPTION_H_INCLUDED

```
#define OCRSTUDIOSDK_OCR_STUDIO_EXCEPTION_H_INCLUDED
```

Definition at line 13 of file ocr_studio_exception.h.

2.4 ocr_studio_exception.h

Go to the documentation of this file.

```
00001
00011 #pragma once
00012 #ifndef OCRSTUDIOSDK_OCR_STUDIO_EXCEPTION_H_INCLUDED
00013 #define OCRSTUDIOSDK_OCR_STUDIO_EXCEPTION_H_INCLUDED
00014
00015 #include <ocrstudiosdk/ocr_studio_export.h>
00016
00017 namespace ocrstudio {
00018
00019 class OCR_STUDIO_SDK_DLL_EXPORT OCRStudioSDKException {
00020 public:
00022   virtual ~OCRStudioSDKException();
00023
00025   OCRStudioSDKException(
00026       const char* type, const char* msg);
00027
00029   OCRStudioSDKException(
00030       const OCRStudioSDKException& copy);
00031
00033   const char* Type() const;
00034
00036   const char* Message() const;
00037
00038 private:
00039   char* type_;
00040   char* msg_;
00041 };
00042
00043 } // namespace ocrstudio
00044
00045 #endif // OCRSTUDIOSDK_OCR_STUDIO_EXCEPTION_H_INCLUDED
```

2.5 ocr_studio_export.h File Reference

Common definitions for library exports.

2.5.1 Detailed Description

Common definitions for library exports.

Definition in file ocr_studio_export.h.

2.5.2 Macro Definition Documentation

OCRSTUDIOSDK_OCR_STUDIO_EXPORT_H_INCLUDED

```
#define OCRSTUDIOSDK_OCR_STUDIO_EXPORT_H_INCLUDED
```

Definition at line 13 of file ocr_studio_export.h.

OCR_STUDIO_SDK_DLL_EXPORT

```
#define OCR_STUDIO_SDK_DLL_EXPORT
```

Definition at line 21 of file ocr_studio_export.h.

2.6 ocr_studio_export.h

Go to the documentation of this file.

```
00001
00011 #pragma once
00012 #ifndef OCRSTUDIOSDK_OCR_STUDIO_EXPORT_H_INCLUDED
00013 #define OCRSTUDIOSDK_OCR_STUDIO_EXPORT_H_INCLUDED
00014
00015 #if defined _WIN32
00016 # define OCR_STUDIO_SDK_DLL_EXPORT __declspec(dllexport)
00017 #else // defined _WIN32
00018 # if defined(__clang__) || defined(__GNUC__)
00019 #  define OCR_STUDIO_SDK_DLL_EXPORT __attribute__ ((visibility ("default")))
00020 # else // clang of gnuc
00021 #  define OCR_STUDIO_SDK_DLL_EXPORT
00022 # endif // clang of gnuc
00023 #endif // defined _WIN32
00024
00025 #endif // OCRSTUDIOSDK_OCR_STUDIO_EXPORT_H_INCLUDED
```

2.7 ocr_studio_image.h File Reference

Common image manipulation facilities.

Classes

- class ocrstudio::OCRStudioSDKImage
  Bitmap image class.

Variables

- OCRSTUDIOSDK_PIXEL_FORMAT_G = 0
  Greyscale.
- OCRSTUDIOSDK_PIXEL_FORMAT_GA
  Greyscale + Alpha.
- OCRSTUDIOSDK_PIXEL_FORMAT_AG
  Alpha + Greyscale.
- OCRSTUDIOSDK_PIXEL_FORMAT_RGB
  RGB.
- OCRSTUDIOSDK_PIXEL_FORMAT_BGR
  BGR.
- OCRSTUDIOSDK_PIXEL_FORMAT_BGRA
  BGR + Alpha.
- OCRSTUDIOSDK_PIXEL_FORMAT_ARGB
  Alpha + RGB.
- OCRSTUDIOSDK_YUV_FORMAT_NOT_SET = 0
  Not set.
- OCRSTUDIOSDK_YUV_FORMAT_NV21
  NV 21.

2.7.1 Detailed Description

Common image manipulation facilities.

Definition in file ocr_studio_image.h.

2.7.2 Macro Definition Documentation

OCRSTUDIOSDK_OCR_STUDIO_IMAGE_H_INCLUDED

```
#define OCRSTUDIOSDK_OCR_STUDIO_IMAGE_H_INCLUDED
```

Definition at line 13 of file ocr_studio_image.h.

2.7.3 Variable Documentation

OCRSTUDIOSDK_PIXEL_FORMAT_G

```
OCRSTUDIOSDK_PIXEL_FORMAT_G = 0
```

Greyscale.

Definition at line 24 of file ocr_studio_image.h.

OCRSTUDIOSDK_PIXEL_FORMAT_GA

```
OCRSTUDIOSDK_PIXEL_FORMAT_GA
```

Greyscale + Alpha.

Definition at line 25 of file ocr_studio_image.h.

OCRSTUDIOSDK_PIXEL_FORMAT_AG

```
OCRSTUDIOSDK_PIXEL_FORMAT_AG
```

Alpha + Greyscale.

Definition at line 26 of file ocr_studio_image.h.

OCRSTUDIOSDK_PIXEL_FORMAT_RGB

```
OCRSTUDIOSDK_PIXEL_FORMAT_RGB
```

RGB.

Definition at line 27 of file ocr_studio_image.h.

OCRSTUDIOSDK_PIXEL_FORMAT_BGR

```
OCRSTUDIOSDK_PIXEL_FORMAT_BGR
```

BGR.

Definition at line 28 of file ocr_studio_image.h.

OCRSTUDIOSDK_PIXEL_FORMAT_BGRA

```
OCRSTUDIOSDK_PIXEL_FORMAT_BGRA
```

BGR + Alpha.

Definition at line 29 of file ocr_studio_image.h.

OCRSTUDIOSDK_PIXEL_FORMAT_ARGB

```
OCRSTUDIOSDK_PIXEL_FORMAT_ARGB
```

Alpha + RGB.

Definition at line 30 of file ocr_studio_image.h.

OCRSTUDIOSDK_YUV_FORMAT_NOT_SET

```
OCRSTUDIOSDK_YUV_FORMAT_NOT_SET = 0
```

Not set.

Definition at line 40 of file ocr_studio_image.h.

OCRSTUDIOSDK_YUV_FORMAT_NV21

```
OCRSTUDIOSDK_YUV_FORMAT_NV21
```

NV 21.

Definition at line 41 of file ocr_studio_image.h.

2.8 ocr_studio_image.h

Go to the documentation of this file.

```
00001
00011 #pragma once
00012 #ifndef OCRSTUDIOSDK_OCR_STUDIO_IMAGE_H_INCLUDED
00013 #define OCRSTUDIOSDK_OCR_STUDIO_IMAGE_H_INCLUDED
00014
00015 #include <ocrstudiosdk/ocr_studio_export.h>
00016 #include <ocrstudiosdk/ocr_studio_string.h>
00017
00018 namespace ocrstudio {
00019
00023 enum OCR_STUDIO_SDK_DLL_EXPORT OCRStudioSDKPixelFormat {
00024   OCRSTUDIOSDK_PIXEL_FORMAT_G = 0,
00025   OCRSTUDIOSDK_PIXEL_FORMAT_GA,
00026   OCRSTUDIOSDK_PIXEL_FORMAT_AG,
00027   OCRSTUDIOSDK_PIXEL_FORMAT_RGB,
00028   OCRSTUDIOSDK_PIXEL_FORMAT_BGR,
00029   OCRSTUDIOSDK_PIXEL_FORMAT_BGRA,
00030   OCRSTUDIOSDK_PIXEL_FORMAT_ARGB,
00031   OCRSTUDIOSDK_PIXEL_FORMAT_RGBA
00032 };
00033
00034
00035
00039 enum OCR_STUDIO_SDK_DLL_EXPORT OCRStudioSDKYUVFormat {
00040   OCRSTUDIOSDK_YUV_FORMAT_NOT_SET = 0,
00041   OCRSTUDIOSDK_YUV_FORMAT_NV21,
00042   OCRSTUDIOSDK_YUV_FORMAT_420_888
00043 };
00044
00045
00046
00050 class OCR_STUDIO_SDK_DLL_EXPORT OCRStudioSDKImage {
00051 public:
00057   static int PagesCount(const char* filename);
00058
00065   static OCRStudioSDKString PageName(
00066       const char *filename, int page_number);
00067
00068 public:
00073   static OCRStudioSDKImage* CreateEmpty();
00074
00083   static OCRStudioSDKImage* CreateFromFile(
00084       const char* filename,
00085       int         page_number = 0,
00086       int         max_width = 25000,
00087       int         max_height = 25000);
00088
00098   static OCRStudioSDKImage* CreateFromFileBuffer(
00099       unsigned char* data,
00100       int            data_size,
00101       int            page_number = 0,
00102       int            max_width = 25000,
00103       int            max_height = 25000);
```

```
00104
00113   static OCRStudioSDKImage* CreateFromBase64FileBuffer(
00114       const char* base64_data,
00115       int         page_number = 0,
00116       int         max_width = 25000,
00117       int         max_height = 25000);
00118
00130   static OCRStudioSDKImage* CreateFromPixelBuffer(
00131       unsigned char* data,
00132       int            data_size,
00133       int            width,
00134       int            height,
00135       int            bytes_per_line,
00136       int            bytes_per_channel,
00137       OCRStudioSDKPixelFormat pixel_format);
00138
00149   static OCRStudioSDKImage* CreateFromBuffer(
00150       unsigned char* data,
00151       int            data_size,
00152       int            width,
00153       int            height,
00154       int            bytes_per_line,
00155       int            channels);
00156
00165   static OCRStudioSDKImage* CreateFromYUVSimple(
00166       unsigned char* yuv_data,
00167       int            yuv_data_size,
00168       int            width,
00169       int            height);
00170
00190   static OCRStudioSDKImage* CreateFromYUV(
00191       unsigned char* y_plane,
00192       int            y_plane_size,
00193       int            y_plane_row_stride,
00194       int            y_plane_pixel_stride,
00195       unsigned char* u_plane,
00196       int            u_plane_size,
00197       int            u_plane_row_stride,
00198       int            u_plane_pixel_stride,
00199       unsigned char* v_plane,
00200       int            v_plane_size,
00201       int            v_plane_row_stride,
00202       int            v_plane_pixel_stride,
00203       int            width,
00204       int            height,
00205       OCRStudioSDKYUVFormat yuv_format);
00206
00207 public:
00209   virtual ~OCRStudioSDKImage() = default;
00210
00215   virtual OCRStudioSDKImage* DeepCopy() const = 0;
00216
00223   virtual OCRStudioSDKImage* ShallowCopy() const = 0;
00224
00228   virtual void Clear() = 0;
00229
00234   virtual int ExportPixelBufferLength() const = 0;
00235
00246   virtual int ExportPixelBuffer(unsigned char* export_buffer, int export_buffer_length) const = 0;
00247
00252   virtual OCRStudioSDKString ExportBase64JPEG() const = 0;
00253
00259   virtual void Scale(int width, int height) = 0;
00260
00267   virtual OCRStudioSDKImage* DeepCopyScaled(int width, int height) const = 0;
00268
00279   virtual void CropByQuad(const char* quad_json, int width, int height) = 0;
00280
00291   virtual OCRStudioSDKImage* DeepCopyCroppedByQuad(
00292       const char* quad_json, int width, int height) const = 0;
00293
00301   virtual void CropByRect(int x, int y, int width, int height) = 0;
00302
00311   virtual OCRStudioSDKImage* DeepCopyCroppedByRect(
00312       int x, int y, int width, int height) const = 0;
00313
00324   virtual OCRStudioSDKImage* ShallowCopyCroppedByRect(
00325       int x, int y, int width, int height) const = 0;
00326
00331   virtual void RotateByNinety(int num_rotations) = 0;
00332
00338   virtual OCRStudioSDKImage* DeepCopyRotatedByNinety(int num_rotations) const = 0;
00339
00341   virtual int Width() const = 0;
00342
00344   virtual int Height() const = 0;
00345
```

```
00347   virtual int BytesPerLine() const = 0;
00348
00350   virtual int Channels() const = 0;
00351
00353   virtual void* UnsafeBufferPtr() const = 0;
00354
00356   virtual bool OwnsPixelData() const = 0;
00357
00359   virtual void ForcePixelDataOwnership() = 0;
00360 };
00361
00362
00363 } // namespace ocrstudio
00364
00365 #endif // OCRSTUDIOSDK_OCR_STUDIO_IMAGE_H_INCLUDED
```

2.9 ocr_studio_instance.h File Reference

Main recognition engine instance class declaration.

Classes

- class ocrstudio::OCRStudioSDKInstance

  Main recognition engine class containing configuration for creating recognition sessions.

2.9.1 Detailed Description

Main recognition engine instance class declaration.

Copyright (c) 2024-2024, OCR Studio All rights reserved.

Definition in file ocr_studio_instance.h.

2.9.2 Macro Definition Documentation

OCRSTUDIOSDK_OCR_STUDIO_INSTANCE_H_INCLUDED

```
#define OCRSTUDIOSDK_OCR_STUDIO_INSTANCE_H_INCLUDED
```

Definition at line 13 of file ocr_studio_instance.h.

2.10 ocr_studio_instance.h

Go to the documentation of this file.

```
00001
00011 #pragma once
00012 #ifndef OCRSTUDIOSDK_OCR_STUDIO_INSTANCE_H_INCLUDED
00013 #define OCRSTUDIOSDK_OCR_STUDIO_INSTANCE_H_INCLUDED
00014
00015 #include <ocrstudiosdk/ocr_studio_export.h>
00016 #include <ocrstudiosdk/ocr_studio_session.h>
00017 #include <ocrstudiosdk/ocr_studio_delegate.h>
00018
00019 namespace ocrstudio {
00020
00025 class OCR_STUDIO_SDK_DLL_EXPORT OCRStudioSDKInstance {
00026 public:
00040   static OCRStudioSDKInstance* CreateStandalone(
00041       const char* json_instance_init_params = nullptr);
00042
00056   static OCRStudioSDKInstance* CreateFromPath(
00057       const char* configuration_filename,
00058       const char* json_instance_init_params = nullptr);
00059
00074   static OCRStudioSDKInstance* CreateFromBuffer(
00075       unsigned char* configuration_buffer,
00076       int            configuration_buffer_size,
00077       const char*    json_instance_init_params = nullptr);
00078
00079 public:
00080
00084   static const char* LibraryVersion();
00085
00086 public:
00088   virtual ~OCRStudioSDKInstance() = default;
00089
00107   virtual const char* Description() const = 0;
00108
00128   virtual OCRStudioSDKSession* CreateSession(
00129       const char* authorization_signature,
00130       const char* json_session_params,
00131       OCRStudioSDKDelegate* callback_delegate = nullptr) const = 0;
00132 };
00133
00134 } // namespace ocrstudio
00135
00136 #endif // OCRSTUDIOSDK_OCR_STUDIO_INSTANCE_H_INCLUDED
```

2.11 ocr_studio_result.h File Reference

Result containers.

Classes

- class ocrstudio::OCRStudioSDKItem

  A constituent object of a recognized or analyzed target.
- class ocrstudio::OCRStudioSDKItemIterator

  Map-like iterator for a collection of OCRStudioSDKItem objects.
- class ocrstudio::OCRStudioSDKTarget

  Recognition or analysis target (document or other object)
- class ocrstudio::OCRStudioSDKResult

  Main session result class - container with full session result.

2.11.1 Detailed Description

Result containers.

Definition in file ocr_studio_result.h.

2.11.2 Macro Definition Documentation

OCRSTUDIOSDK_OCR_STUDIO_RESULT_H_INCLUDED

```
#define OCRSTUDIOSDK_OCR_STUDIO_RESULT_H_INCLUDED
```

Definition at line 13 of file ocr_studio_result.h.

2.12 ocr_studio_result.h

Go to the documentation of this file.

```
00001
00011 #pragma once
00012 #ifndef OCRSTUDIOSDK_OCR_STUDIO_RESULT_H_INCLUDED
00013 #define OCRSTUDIOSDK_OCR_STUDIO_RESULT_H_INCLUDED
00014
00015 #include <ocrstudiosdk/ocr_studio_export.h>
00016 #include <ocrstudiosdk/ocr_studio_string.h>
00017 #include <ocrstudiosdk/ocr_studio_image.h>
00018
00019 namespace ocrstudio {
00020
00024 class OCR_STUDIO_SDK_DLL_EXPORT OCRStudioSDKItem {
00025 public:
00027   virtual ~OCRStudioSDKItem() = default;
00028
00033   virtual OCRStudioSDKItem* DeepCopy() const = 0;
00034
00036   virtual const char* Type() const = 0;
00037
00039   virtual const char* Name() const = 0;
00040
00042   virtual const char* Value() const = 0;
00043
00045   virtual double Confidence() const = 0;
00046
00048   virtual bool Accepted() const = 0;
00049
00057   virtual const char* Attributes() const = 0;
00058
00060   virtual bool HasImage() const = 0;
00061
00063   virtual const OCRStudioSDKImage& Image() const = 0;
00064
00066   virtual const char* Description() const = 0;
00067 };
00068
00069
00070
00074 class OCRStudioSDKItemIteratorImplementation;
00075
00079 class OCR_STUDIO_SDK_DLL_EXPORT OCRStudioSDKItemIterator {
00080 public:
00082   ~OCRStudioSDKItemIterator();
00083
00085   OCRStudioSDKItemIterator(const OCRStudioSDKItemIterator& copy);
00086
00088   OCRStudioSDKItemIterator& operator =(const OCRStudioSDKItemIterator& other);
00089
00091   bool IsEqualTo(const OCRStudioSDKItemIterator& other) const;
00092
00094   bool operator ==(const OCRStudioSDKItemIterator& other) const;
00095
00097   bool operator !=(const OCRStudioSDKItemIterator& other) const;
00098
00100   OCRStudioSDKItemIterator Next() const;
00101
00103   void Step();
00104
00106   void operator ++();
00107
00109   const char* Key() const;
00110
00112   const OCRStudioSDKItem& Item() const;
00113
00114 public:
00116   static OCRStudioSDKItemIterator CreateFromImplementation(
00117       const OCRStudioSDKItemIteratorImplementation& rimpl);
```

```
00118
00119 private:
00121   OCRStudioSDKItemIterator(const OCRStudioSDKItemIteratorImplementation& rimpl);
00122
00124   OCRStudioSDKItemIteratorImplementation* pimpl_;
00125 };
00126
00127
00128
00132 class OCR_STUDIO_SDK_DLL_EXPORT OCRStudioSDKTarget {
00133 public:
00135   virtual ~OCRStudioSDKTarget() = default;
00136
00141   virtual OCRStudioSDKTarget* DeepCopy() const = 0;
00142
00156   virtual const char* Description() const = 0;
00157
00165   virtual int ItemsCountByType(const char* item_type) const = 0;
00166
00175   virtual bool HasItem(const char* item_type, const char* item_name) const = 0;
00176
00183   virtual const OCRStudioSDKItem& Item(
00184       const char* item_type, const char* item_name) const = 0;
00185
00192   virtual OCRStudioSDKItemIterator ItemsBegin(const char* item_type) const = 0;
00193
00200   virtual OCRStudioSDKItemIterator ItemsEnd(const char* item_type) const = 0;
00201
00206   virtual bool IsFinal() const = 0;
00207 };
00208
00209
00210
00214 class OCR_STUDIO_SDK_DLL_EXPORT OCRStudioSDKResult {
00215 public:
00217   virtual ~OCRStudioSDKResult() = default;
00218
00223   virtual OCRStudioSDKResult* DeepCopy() const = 0;
00224
00229   virtual int TargetsCount() const = 0;
00230
00236   virtual const OCRStudioSDKTarget& TargetByIndex(int target_index) const = 0;
00237
00242   virtual bool AllTargetsFinal() const = 0;
00243 };
00244
00245 } // namespace ocrstudio
00246
00247 #endif // OCRSTUDIOSDK_OCR_STUDIO_RESULT_H_INCLUDED
```

2.13 ocr_studio_session.h File Reference

Main processing session class declaration.

Classes

- class ocrstudio::OCRStudioSDKSession

  Main processing session class - agent for performing image analysis.

2.13.1 Detailed Description

Main processing session class declaration.

Copyright (c) 2024-2024, OCR Studio All rights reserved.

Definition in file ocr_studio_session.h.

2.13.2 Macro Definition Documentation

OCRSTUDIOSDK_OCR_STUDIO_SESSION_H_INCLUDED

```
#define OCRSTUDIOSDK_OCR_STUDIO_SESSION_H_INCLUDED
```

Definition at line 13 of file ocr_studio_session.h.

2.14 ocr_studio_session.h

Go to the documentation of this file.

```
00001
00011 #pragma once
00012 #ifndef OCRSTUDIOSDK_OCR_STUDIO_SESSION_H_INCLUDED
00013 #define OCRSTUDIOSDK_OCR_STUDIO_SESSION_H_INCLUDED
00014
00015 #include <ocrstudiosdk/ocr_studio_export.h>
00016 #include <ocrstudiosdk/ocr_studio_image.h>
00017 #include <ocrstudiosdk/ocr_studio_result.h>
00018
00019 namespace ocrstudio {
00020
00024 class OCR_STUDIO_SDK_DLL_EXPORT OCRStudioSDKSession {
00025 public:
00027   virtual ~OCRStudioSDKSession() = default;
00028
00029 public:
00044   virtual const char* Description() const = 0;
00045
00051   virtual void ProcessImage(const OCRStudioSDKImage& image) = 0;
00052
00058   virtual const OCRStudioSDKResult& CurrentResult() const = 0;
00059
00063   virtual void Reset() = 0;
00064 };
00065
00066 } // namespace ocrstudio
00067
00068 #endif // OCRSTUDIOSDK_OCR_STUDIO_SESSION_H_INCLUDED
```

2.15 ocr_studio_string.h File Reference

String manipulation facilities.

Classes

- class ocrstudio::OCRStudioSDKString

2.15.1 Detailed Description

String manipulation facilities.

Definition in file ocr_studio_string.h.

2.15.2 Macro Definition Documentation

OCRSTUDIOSDK_OCR_STUDIO_STRING_H_INCLUDED

```
#define OCRSTUDIOSDK_OCR_STUDIO_STRING_H_INCLUDED
```

Definition at line 13 of file ocr_studio_string.h.

2.16 ocr_studio_string.h

Go to the documentation of this file.

```
00001
00011 #pragma once
00012 #ifndef OCRSTUDIOSDK_OCR_STUDIO_STRING_H_INCLUDED
00013 #define OCRSTUDIOSDK_OCR_STUDIO_STRING_H_INCLUDED
00014
00015 #include <ocrstudiosdk/ocr_studio_export.h>
00016
00017 namespace ocrstudio {
00018
00019 class OCR_STUDIO_SDK_DLL_EXPORT OCRStudioSDKString {
00020 public:
00022   ~OCRStudioSDKString();
00023
00025   OCRStudioSDKString();
00026
00028   explicit OCRStudioSDKString(const char* c_str);
00029
00031   OCRStudioSDKString(
00032       const OCRStudioSDKString& copy);
00033
00035   OCRStudioSDKString& operator =(
00036       const OCRStudioSDKString& other);
00037
00039   OCRStudioSDKString& operator +=(
00040       const OCRStudioSDKString& other);
00041
00043   OCRStudioSDKString operator +(
00044       const OCRStudioSDKString& other) const;
00045
00047   const char* CStr() const;
00048
00050   int Size() const;
00051
00052 private:
00053   int size_;
00054   char* str_;
00055 };
00056
00057 } // namespace ocrstudio
00058
00059 #endif // OCRSTUDIOSDK_OCR_STUDIO_STRING_H_INCLUDED
```

Index