acuant

AssureID™

Connect Web Services

Integration Guide

Version 2.0.1
May 2018

May 2018

Acuant Inc.
6080 Center Drive
Suite 850
Los Angeles, CA 90045

Contents

AssureID Connect Web Services 1

Using AssureID Connect 2

Web Services authentication 8

Programmability 9

Language support 9

Integration tools 10

Changes in this release 11

Changes in earlier releases 12

Image requirements and recommendations 19

Image cropping 19

Image resolution 20

Image compression 20

HTTP Status codes 21

AssureID Connect Web Services

This document outlines and describes the AssureID Connect Web Services document authentication and provides information that will enable client applications to consume AssureID document authentication technology using a document-centric enterprise-level service.

AssureID has been historically a desktop-oriented system, coupling the AssureID authentication engine with one or more *i-D*entify document readers. This system captures images from and performs a deep analysis of a variety of identification and travel documents. This analysis generates a rather large set of data describing the contents and forensics of presented documents. By default, this data is transient in nature and is generally viewed in real-time with a variety of third-party software products that leverage the AssureID SDK.

The AssureID Connect Web Services seeks to take the power of AssureID beyond the desktop and move it into the enterprise, dramatically expanding the benefit of deploying AssureID by enabling enterprise applications to leverage the information captured by document readers and capture devices present in the system.

The current version of AssureID Connect Web Services enables the following core features:

- Identification document classification, data capture, and authentication against a database of over 3,400 supported document types
- Storage and retrieval of individual documents
- Secure communication over SSL/TLS with user-based authentication and authorization
- Support for a wide range of document scanners and imaging devices from purpose-built ID scanners to flatbed scanners and mobile phones
- Upload individual document resources including image files, binary data files and results of contactless chip authentications
- Automatic detection and cropping of the document from within the uploaded image
- Data transfer via XML or JavaScript Object Notation (JSON)
- Easy integration into existing or new applications using either the included .NET SDK or the REST API
- Includes the AssureID Connect Service Workbench application, which allows integrators to easily test the service using previously captured data

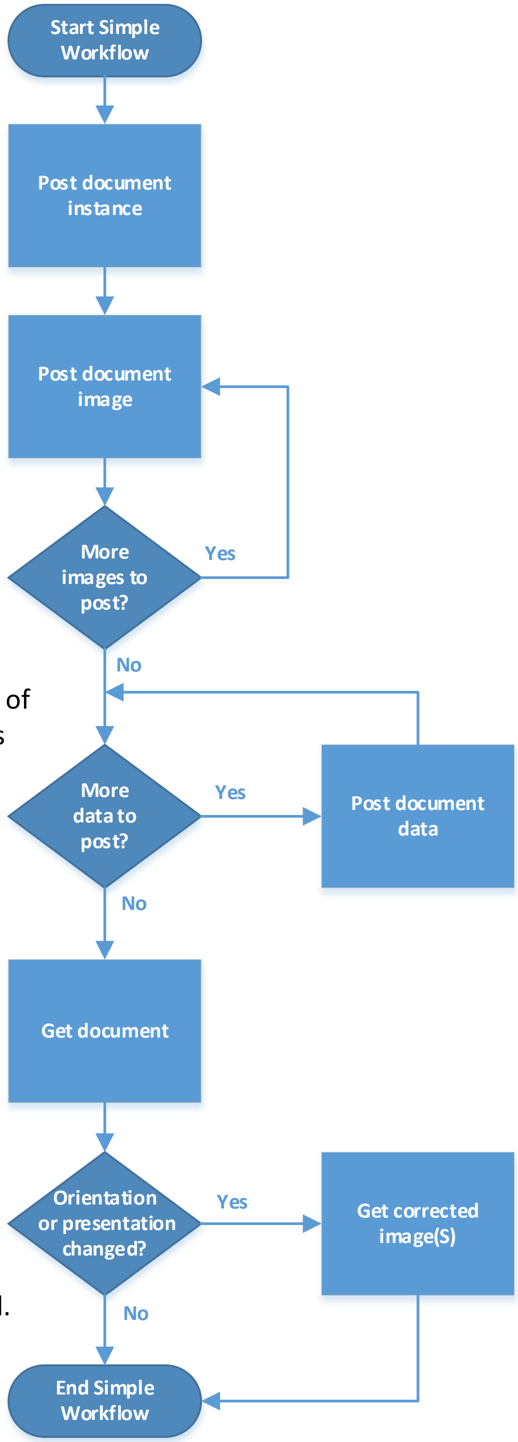
Using AssureID Connect

This section provides details and examples of how to use the AssureID Connect API to process individual document data capture and authentication transactions. For specifics on the C# wrapper and REST API interfaces, see the .NET Web SDK and REST API sections in this document.

Simple workflow

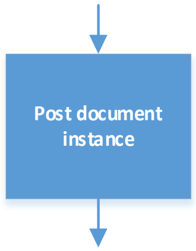
Processing a document with AssureID Connect can be accomplished using a simple workflow, which includes the following steps:

- Create a new document instance
- Post data to the document instance
- Get the document instance results
- Get corrected images if the orientation or presentation has been updated (optional)



Post document instance

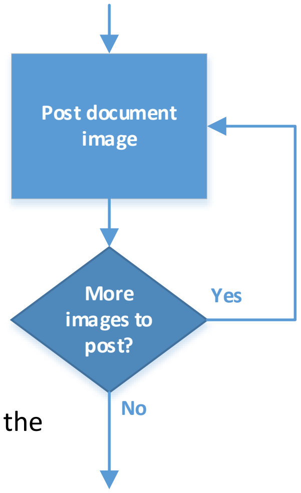
In order to initiate a document authentication transaction, a document instance must first be created. This is accomplished by posting a new document instance. This operation requires settings on how the transaction will be processed as well as details on the input device used. The result of this operation is a document instance identifier. This identifier is required for all subsequent operations to associate the operations with the document instance.



The document settings specify the source that was used to capture the document images such as device model and manufacturer. It is important that the sensor type also be specified as Scanner, Camera, or Mobile as this parameter is used to determine how the authentication process is performed. Failure to specify this property or set it incorrectly may result in degraded authentication performance.

The settings also specify image processing settings such as whether the document images posted will need to be cropped by the service or whether they are already cropped. See the SDK documentation for more details.

Post document image

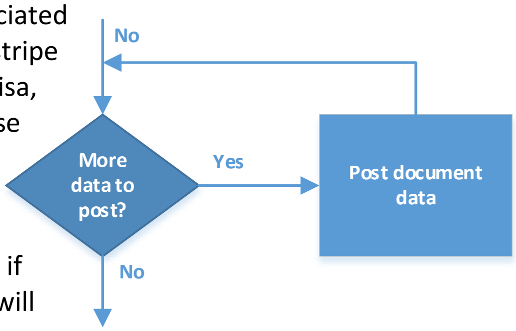


After the document instance has been created, each image that has been captured of the document must then be posted to the service. A minimum of one image is required to perform a transaction, and this image must be using the white light source of the front side of the document. Other images of different light sources may also be included as well as images of the back side of the document, some of which will be necessary to fully capture data or authenticate the document.

The images tend to be very large in size, therefore it is important that this be considered when posting the data. Posting uncompressed images would result in moving a large amount of data across the network, resulting in not only a longer overall transaction time, but also in higher network bandwidth requirements. Acuant recommends that you compress the image data compressed prior to transmission.

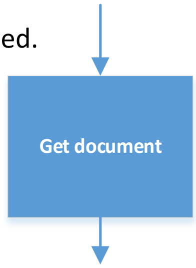
Note Failure to follow the documented guidance on image capture and compression requirements may result in degraded performance and accuracy. See the Image requirements and recommendations section for more information. See also the PostDocumentImage operation in the .NET Web SDK or REST API reference guides for specific requirements for image capture, file format, and compression.

Post more document data



AssureID Connect also accepts non-imaging data that is associated with the document. This information may include magnetic stripe data, the machine-readable zone (MRZ) from a passport or visa, or information extracted from a contactless chip such as those embedded in e-Passport and electronic identification cards. It is not required that this information be included, but if present, it will be used to augment the accuracy of the data capture and authentication processes. For instance, if the chip data from an e-Passport is posted, this information will be used as the most accurate source for data field extraction.

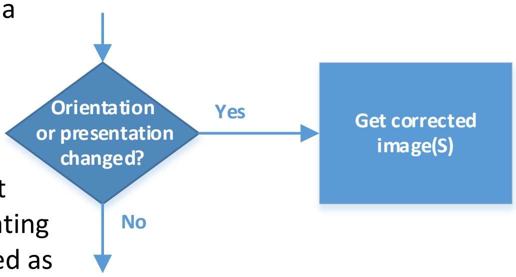
Get document



After all data has been posted to the document instance, the results can then be retrieved. This operation should be performed immediately after all data has been posted to the document instance as it is this operation that causes the actual document classification, data capture, and authentication processing to be performed. The operation may take several seconds to return the document instance results due to this processing.

You can also perform this operation later to retrieve the document instance results, assuming that the document instance has not been deleted from the system. Future requests to retrieve the document instance will not cause the document to be processed again, but will retrieve the same results that were returned when the operation was first invoked.

Get corrected image



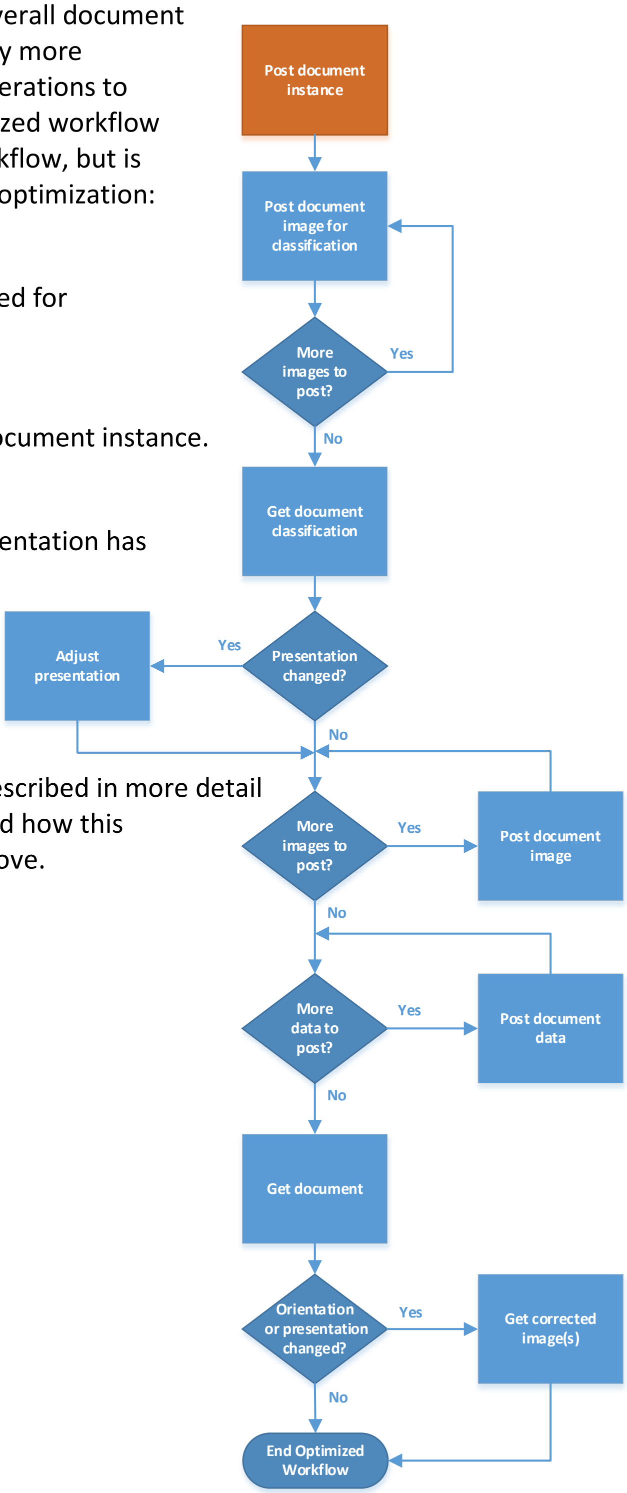
During classification, AssureID Connect identifies the type of a document as well as its presentation and orientation. When posting images to AssureID Connect, one side of the document must be specified as the front, but this may not actually be the front of the document. If the **PresentationChanged** property has been set, this means that the presentation of the document has been corrected, indicating that the document was presented with the back-side specified as the front-side. Orientation refers to whether the top edge of the document was at the top of the captured image. If the document was rotated to correct the orientation, the **OrientationChanged** property will be set. If either of these two properties are set, additional action may be required by the application to ensure that displayed images or images stored external to AssureID Connect reflect the corrected orientation and/or presentation.

To obtain images with corrected presentation and orientation, an application can retrieve each image from AssureID Connect, using the URI provided in the document result. It is also possible to retrieve images using the Web SDK.

Optimized workflow

It is possible to optimize the workflow to reduce the overall document transaction time significantly, but this requires a slightly more complicated workflow and the overlapping of some operations to parallelize some operations in the workflow. An optimized workflow requires the same data to be posted as the simple workflow, but is broken down into several additional steps to allow for optimization:

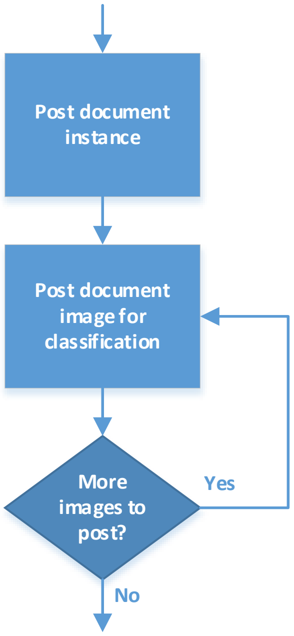
1. Create a new document instance.
2. Post image(s) to the document instance required for classification of the document.
3. Get the classification of the document.
4. Post the remaining image(s) and data to the document instance.
5. Get the document instance results.
6. Get corrected images if the orientation or presentation has been updated (*optional*).



For optimal performance, this workflow should also be overlapped with the image capture workflow, such that the document instance is created and images are posted as soon as available.

The optimized workflow has been broken down and described in more detail below, providing details on each step in the process and how this process differs from the simple workflow described above.

Post document image



The document instance is created in the same manner as the simple workflow, by posting the document instance and obtaining a document instance identifier. However, after the document instance has been created, the optimized workflow differs from the simple workflow. Rather than posting all document images, only the images that are required to classify the document (for example, identify the type of document) are posted. At a minimum, this is the white light source image of one side of the document, but may also include the white light source image of the reverse side of the document when images of both sides of the document have been captured.

Further optimization of this process is possible by parallelizing some of the operations. Posting images typically requires three distinct operations:

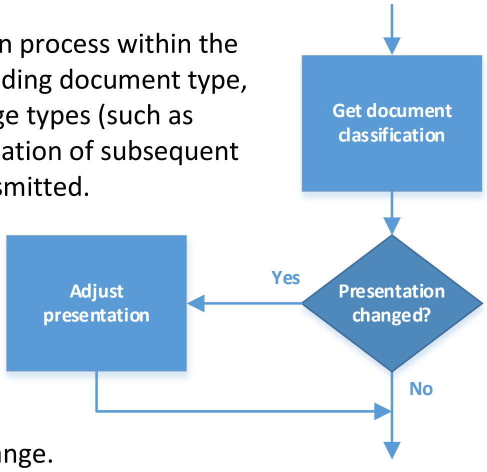
- Capture the image
- Compress the image
- Post (upload) the compressed image

These operations are largely independent and are constrained by different resource types, therefore they can be overlapped for significant performance gains. Image capture tends to be constrained by device I/O, image compression is CPU-driven, and the posting of the image is limited by network bandwidth. See Image Requirements and Recommendations for more information.

Note Non-imaging data (such as magnetic stripe, machine readable zone) is very small compared to imaging data; therefore, it can be beneficial to post this information prior to initiating document classification provided that the data is already available. In some cases, this information can be used to speed up the classification process. If the data is not immediately available, it should be posted after classification has been completed.

Get document classification

Getting the document classification will initiate the classification process within the service and return the results of the classification process, including document type, issuer, class, and series. It will also return which additional image types (such as nearinfrared and ultraviolet) are supported to allow for optimization of subsequent captures and transmission so no images are unnecessarily transmitted.

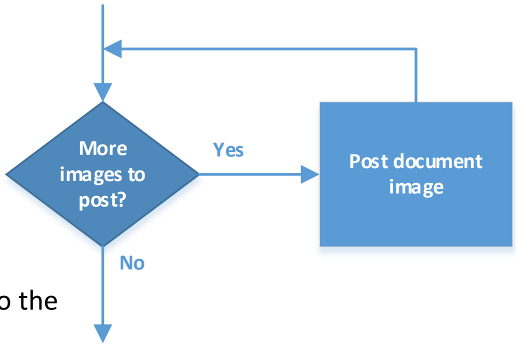


During classification, the presentation of the document may also be determined. That is, if an image was posted as the front side of the document but is actually the back side of the document, the **PresentationChanged** property will indicate that the document presentation (for example, sides) were reversed during classification. If this occurs, all subsequent image posts must account for this presentation change.

For example, if a white/front image was posted for classification, but it classified as white/back, the remaining white image should be posted as white/front to properly account for the presentation change.

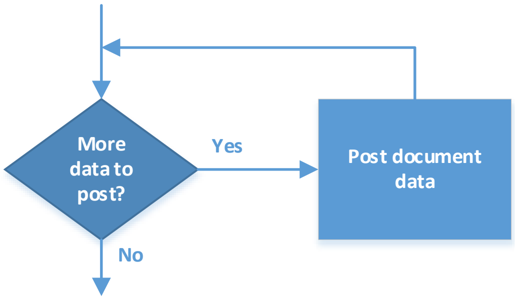
Post document image

All remaining images are then posted to the document instance.



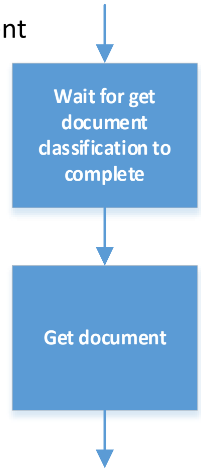
Post document data

Any non-imaging data that is to be included is posted to the document instance.



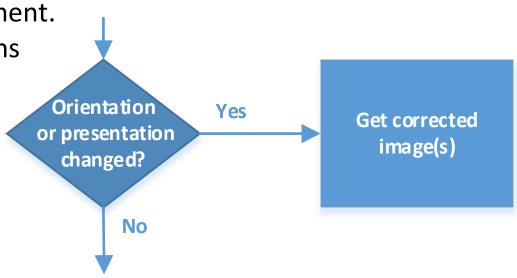
Post any non-imaging data

After all data is posted, the client shall then check and wait for the get document classification operation to complete (if it has not yet completed). Once the operation has completed, the results can be retrieved. This operation will perform any remaining processing, including data capture and authentication that had not yet been performed on the server.



Get corrected image

During classification, AssureID Connect identifies the type of a document as well as its presentation and orientation. When posting images to AssureID Connect, one side of the document must be specified as the front, but this may not actually be the front of the document. If the **PresentationChanged** property has been set, this means that the presentation of the document has been corrected, indicating that the document was presented with the back-side specified as the front-side and vice versa. Orientation refers to whether the top edge of the document was at the top of the captured image. If the document was rotated to correct the orientation, the **OrientationChanged** property will be set. If either of these two properties are set, additional action may be required by the application to ensure that displayed images or images stored external to AssureID Connect reflect the corrected orientation and/or presentation.



To obtain images with corrected presentation and orientation, an application can retrieve each image from AssureID Connect, using the image URIs provided in the document result. It is also possible to retrieve images using the Web SDK.

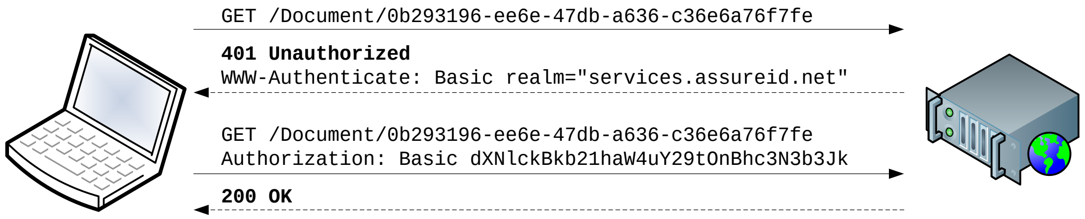
Web Services authentication

The AssureID Connect Web Services utilizes the HTTP Basic authentication scheme as defined in RFC 2617, HTTP Authentication: Basic and Digest Access Authentication.

Basic authentication works as follows:

- For requests to Web Services that require authentication, the server returns 401 (Unauthorized). The response will include a WWW-Authenticate header, indicating the server supports Basic authentication.
- The client sends another request, with the client credentials in the Authorization header. The credentials are formatted as the string "*name:password,*" base64-encoded. The credentials are not encrypted.

Because the credentials are sent unencrypted to the server, the resources contained within the AssureID Connect Web Services are only accessible using the HTTPS protocol.



```
GET /Document/0b293196-ee6e-47db-a636-c36e6a76f7fe

401 Unauthorized
WWW-Authenticate: Basic realm="services.assureid.net"

GET /Document/0b293196-ee6e-47db-a636-c36e6a76f7fe
Authorization: Basic dXNlckBkb21haW4uY29tOnBhc3N3b3Jk

200 OK
```

Figure 1. HTTP Basic authentication

Basic authentication is performed within the context of a "realm". The server will include "services.assureid.net" as the realm in the WWW-Authenticate header.

Programmability

The following example demonstrates one approach on how a C# client can easily access a secured AssureID Connect Web Services Basic Authentication resource.

Example

```
using System;
using System.Net;

using AssureTec.AssureID.Web.SDK;

try
{
    //--- Create web service client.
    Uri serverAddress = new Uri("https://services.assureid.net");
    NetworkCredential credential = new NetworkCredential("user@domain.com", 
"password");
    AssureIDServiceClient client = new AssureIDServiceClient(serverAddress, 
credential);

    //--- Get document from web service.
    Guid instanceId = new Guid("0b293196-ee6e-47db-a636-c36e6a76f7fe");
    Document document = client.GetDocument(instanceId);

    //--- Do additional work.
}
catch (WebException e)
{
    Console.WriteLine(e.Message);
}
```

Language support

All API calls now support the Accept-Language header, which lets you display authentication details (responses) in a specified language. AssureID supports the following languages:

▪ Arabic (ar)	▪ French (fr)	▪ Portuguese (pt)
▪ Czech (cs)	▪ German (de)	▪ Russian (ru)
▪ Dutch (nl)	▪ Greek (el)	▪ Spanish (es)
▪ English (en)	▪ Italian (it)	▪ Turkish (tr)

The following example shows the Accept-Language header for the languages English (en) and German (de).

Example

```
GET http://connect-service-host/AssureIDService/Document/3701e605-b28f-480b-
8354-3f0b308646e1 HTTP/1.1
Authorization: Basic XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Accept-Language: en
Accept: application/json
Host: connect-service-host
```

```
GET http://connect-service-host/AssureIDService/Document/3701e605-b28f-480b-
8354-3f0b308646e1 HTTP/1.1
Authorization: Basic XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Accept-Language: de
Accept: application/json
Host: connect-service-host
```

Integration tools

Two sets of programming tools are currently available that enable integration with AssureID Connect:

- The .NET Web SDK is a .NET-based assembly that encapsulates the REST API and provides an easy-to-integrate set of classes for interacting with the AssureID Connect web service.
- The REST API is the actual AssureID Connect Web Services interface. Developers may choose to integrate directly with this interface and not utilize the .NET Web SDK.

.NET Web SDK

The AssureID Connect Web Services Integration Kit includes a set of .NET classes that encapsulate the AssureID Connect Web Services and provide an easy-to-use interface. For more information, see the *AssureID Connect SDK (.NET) Reference Guide (AssureTec.AssureID.Web.SDK.NET)* included in the kit.

REST API

Developers also have the option of interfacing directly to the AssureID Connect REST web service. If using this interface, it is the responsibility of the developer to properly set up request structure, content, and authentication as well as parse the returned responses. For more information, see the *AssureID Connect SDK (REST) Reference Guide* (AssureTec.AssureID.Web.SDK.REST) included in the kit.

Changes in this release

The recommended image compression for an image using a visible (white) light source has been increased in order to perform additional authentication tests:

Light Source	Minimum	Recommended
Visible	500 KB	600-1200 KB

See the Image compression section for more information.

The AssureID Connect SDK was modified to add the optional calculation of metrics for sharpness and glare. This feature allows users to determine the quality of a captured image, on a scale of 0–100 (with 100 being the best quality), after an image is posted but before completing the transaction. You can use these metrics to evaluate whether the captured images are of sufficient quality. Acuant recommends that you accept images that yield a quality factor of at *least* 50, however, you may choose to modify this threshold to better suit your application's specific requirements. A setting of 50 means that you would want to recapture images that are determined to be of insufficient quality (below 50).

Note The use of these metrics do *not* have any bearing on the document result. It provides the user with a set of metrics to evaluate the image quality *before* completing a transaction.

The following changes and additions are included this version of the AssureID Connect .NET SDK:

.NET SDK member	Parameter	Description of changes
DocumentImage class	**GlareMetric** property	New. If specified, indicates the relative amount of glare detected on an image, on a scale of **0** (obscured by glare) to **100** (no glare detected).
	SharpnessMetric property	New. If specified, indicates the relative sharpness of this image, on a scale of **0** (blurry) to **100** (sharp).
AssureIDSessionClient	**GetDocument method**	Modified. Returns metrics via the **DocumentImage** class.
AssureIDSessionClient	**GetDocumentImageMetrics** method	New. Returns metrics via the **DocumentImage** class.
AssureIDSessionClient	**PostDocumentImage** method	New parameter. If specified, calculates the quality metrics of sharpness and glare on an image. **True |** False (default)

The following changes and additions are included this version of the AssureID Connect REST API:

REST API member	Parameter	Description of changes
POST DocumentImage	**metrics**	New **metrics** parameter. Calculates the amount of glare or sharpness present on an image. **True |** False (default)
GET DocumentImageMetrics		New endpoint. Returns any available metadata, including the the quality metrics of sharpness and glare, if requested for a previously posted image.
GET Document		Modified to include returning any available metadata, including the quality metrics of sharpness and glare, if requested for a previously posted image.

Changes in earlier releases

As changes are made to the AssureID Connect Web Services API, they are documented here to ensure that consumers of the interface fully understand what changes have been made, their impact on compatibility, and guidance regarding the change.

API changes introduced in versions 1.9.1 — 1.9.9

There were no API changes from version **1.9.1** to **1.9.9**.

API changes introduced in version 1.9.0

The following interface-impacting enhancements and changes were made in version 1.9.0 of the AssureID Connect .NET SDK:

SDK class	SDK member	Description of changes
DocumentSettings	**ProcessMode.Barcode**	New enumeration. Only the document's barcode is processed.
DocumentClass	**WeaponLicense** **TribalIdentification** **VoterIdentification**	New document types added

The following interface-impacting enhancements and changes were made in version 1.9.0 of the AssureID Connect REST API:

REST method	REST element	Description of changes
POST Document/Instance	**ProcessMode.Barcode**	New enumeration. Only the document's barcode is processed.
DocumentClass	**WeaponLicense** **TribalIdentification** **VoterIdentification**	New document types added

All API calls now support the Accept-Language header, which enables you to display authentication details (responses) in a specified language.

API changes introduced in version 1.8.8

The **Barcode** enumeration was added to support 2D barcode-only read mode. It decodes and parses the barcode from an image of the back of a North American (U.S./Canada) ID. If a 2D barcode string is specified, any images are ignored. If 2D barcode string is *not* specified, AssureID will look first for a Front/White image, then a Back/White image. The first image from which it is able to extract a barcode will be the data processed.

Note The AssureTec.AssureID.Web.SDK.Document returned by AssureID will include all data fields available in the 2D barcode. AssureID will not classify the document; therefore, the returned classification will be Unknown. No authentication tests will be run.

To process a barcode, use **AssureTec.AssureID.Web.SDK.DocumentProcessMode.Barcode**. To use this mode, callers must post either a parsed 2D barcode data string, or an image that contains a 2D barcode, for example:

```
POST /Document/Instance, DocumentSettings.ProcessMode.Barcode
POST /Document/{instanceId}/Data, DocumentDataType.Barcode2D, string
GET /Document/{instanceId}
(optional) DELETE /Document/{instanceId}
```

or

```
POST /Document/Instance, DocumentSettings.ProcessMode.Barcode
POST /Document/{instanceId}/Image, DocumentSide.Front|Back,
   LightSource.White, image
GET /Document/{instanceId}
(optional) DELETE /Document/{instanceId}
```

API changes introduced in version 1.8.7

There were no API changes from version 1.8.6 to 1.8.7.

API changes introduced in version 1.8.6

API changes in version 1.8.6 were made to detect document size if cropping is enabled.

SDK Class	SDK Member	Description of Changes
CroppingMode	**ImageCroppingMode**	New enumeration
CroppingExpectedSize	**ImageCroppingExpectedSize**	New enumeration

API changes introduced in version 1.8.5

API changes were made in version 1.8.5 to improve automatic document image cropping support. These changes are backward-compatible such that existing applications written using the .NET or REST APIs will continue to function without modification. Modification will be required to take advantage of new functionality.

SDK Class	SDK Member	Description of Changes
CroppingMode		Added new enumeration value Always

API changes introduced in version 1.8.4

There were no API changes from version 1.8.3 to 1.8.4.

API changes introduced in version 1.7.0

API changes from version 1.6.0 to 1.7.0 were made to support automatic document image cropping. These changes are backward-compatible such that existing applications written using the .NET or REST APIs will continue to function without modification. Modification will be required to take advantage of new functionality.

The following interface-impacting enhancements and changes were made in release 1.7.0 of the AssureID Connect .NET SDK:

SDK Class	SDK Member	Description of Changes
CroppingMode	**ImageCroppingMode**	New enumeration
CroppingExpectedSize	**ImageCroppingExpectedSize**	New enumeration
DocumentSettings	**ImageCroppingMode**	New property
	ImageCroppingExpectedSize	New property
AssureIDSessionClient	**PostDocumentImage**	Image may now be uncropped without accurate resolution specified if Automatic cropping was enabled when the document instance was created.

The following interface-impacting enhancements and changes were made in release 1.7.0 of the AssureID Connect REST API:

REST Method	REST Element	Description of Changes
POST Document/Instance	**ImageCroppingMode**	New enumeration
	ImageCroppedExpectedSize	New enumeration
POST DocumentImage	**Image**	Image may now be uncropped without accurate resolution specified if Automatic cropping was enabled when the document instance was created.

API changes introduced in version 1.6.0

There were no API changes from version 1.5.0 to version 1.6.0.

API changes introduced in version 1.5.0

The AssureID Connect Web Services API 1.5.0 is generally backward-compatible with the AssureID Connect Web Services API 1.4.0. However, the AssureID Connect Web Services client SDK is not backward-compatible due to some of the below-listed enhancements and changes that break object serialization and deserialization.

Applications that were compiled against the AssureID Connect Web Services API 1.4.0 may not need to be recompiled; those that were compiled against the AssureID Connect client SDK 1.4.0 must be recompiled. In all cases, some minor source code changes may be necessary because of the improvements made to the software.

The following interface-impacting enhancements and changes were made in release 1.5.0 of the AssureID Connect Web Services API:

Web Services Operation	What changed?	Comments
GET DocumentTypes	Number of query string arguments changed	Added a **SubscriptionId** query string argument. If supplied, the value of this argument must be set to a valid subscription identifier that is obtained using the new GET Subscriptions operation.
GET Subscriptions	New operation	New

The following interface-impacting enhancements and changes were made in release 1.5 of the AssureID Connect Web Services API:

Client SDK Member	What changed?	Comments
Document class	Enhanced	Added a **ProcessMode** read-only property that yields a **DocumentProcessMode** enumeration value Added a **Subscription** read-only property that yields a **Subscription** object
DocumentClassification class	Enhanced	Added a **PresentationChanged** read-only property
DocumentDataSource enumeration	Enhanced	Added an **Other** member
DocumentField class	Enhanced	Added a **DataSource** read-only property that yields a **DocumentDataSource** enumeration value
DocumentProcessMode enumeration	Added	

Client SDK Member	What changed?	Comments
DocumentSettings class	Changed/ enhanced	Renamed the **ManualDocumentTypeId** read/write property to **ManualDocumentType**. Note The value of this property must be set to a valid DocumentType object that is obtained from the service using the GET DocumentType operation.
		Added a **ProcessMode** read/write property containing a **DocumentProcessMode** enumeration value for specifying how a document shall be processed. Added a **SubscriptionId** read/write property for specifying the subscription that shall be used for document processing. Note The value of this property must be set to a valid Subscription identifier Guid that is obtained from the service using the GET Subscriptions operation.
Subscription class	Added	New

API changes introduced in version 1.4.0

The AssureID Connect Web Services API and client SDK 1.4.0 are not binary compatible with the AssureID Connect Web Services API and client SDK 1.3.0. Applications that were compiled against the AssureID Connect Web Services API or client SDK 1.3.0 should be recompiled. Some source code changes may be necessary because of the improvements made to the software.

Note The serialization namespace of all data returned from the AssureID Connect Web Services and the client SDK 1.4.0 data contracts has been changed to http://services.assureid.net/2014/09.

The following interface-impacting enhancements and changes were made in release 1.4.0 of the AssureID Connect Web Services API:

Web Services Operation	What changed?	Comments
GET DocumentChipData	Added	New operation
POST DocumentChipData	Added	New operation
GET DocumentData	Number of query string arguments changed	Removed the **DocumentDataSubType** query string argument. It was used to specify contactless chip data items. The new **GET DocumentChipData** operation can be used for this purpose.
POST DocumentData	Number of query string arguments changed	Removed the **DocumentDataSubType** query string argument. It was used to specify contactless chip data items. The new **POST DocumentChipData** operation can be used for this purpose.
GET DocumentImage	Query string argument changed	Renamed the **DeviceLight** query string argument to **LightSource**.
POST DocumentImage	Query string argument changed	Renamed the **DeviceLight** query string argument to **LightSource**.
POST DocumentInstance	New operation	New operation
GET DocumentLog	New operation	New operation
POST DocumentSettings	Removed operation	Use the **POST DocumentInstance** operation instead.

Client SDK Member	What changed?	Comments
AssureIDServiceClient class	**Enhanced**	Added a **ServiceAddress** read-only property
ChipAuthentication class	Changed	Modified the return type of the Result read-only property from **AuthenticationResult** to **ChipAuthenticationResult**
DeviceInfo class	Enhanced	Added a constructor overload accepting **SensorType** argument and added new **SensorType** read-only property
DeviceType class	Enhanced	Added constructor overload accepting **SensorType** argument and added new **SensorType** read-only property
Document class	Changed/enhanced	Renamed the **Id** read-only property to **InstanceId**. Added an **AuthenticationSensitivity** read-only property

Client SDK Member	What changed?	Comments
DocumentClassification class	Changed	Removed the GenericIssuerCode and **GenericIssuerName** read-only properties **Note** Use the read-only **Type.Issuer** and **Type.IssuerName** properties instead.
DocumentDataField class	Enhanced	Added an **IsImage** read-only property.
DocumentField class	Enhanced	Added an **IsImage** read-only property.
DocumentImage class	Changed	Modified the return type of the **Light** read-only property from the **DeviceLight** enumeration to the **LightSource** enumeration.
DocumentImageType class	Changed	Modified the return type of the **Light** read-only property from the **DeviceLight** enumeration to the **LightSource** enumeration.
ChipAuthenticationResult enumeration	Added	New enumeration
ChipAuthenticationType enumeration	Changed/enhanced	Removed the following members: ▪ **CountrySignerRevocation** ▪ **DocumentSignerRevocation** ▪ **DocumentSignerValidation** ▪ **HashValueValidation** ▪ **SignatureValidation** Added the following members: ▪ **PassiveAuthentication** ▪ **SupplementalAuthentication**
ChipDataGroup enumeration	Added	New enumeration
DeviceLight enumeration	Removed	This enumeration is obsolete. Use the **LightSource** enumeration instead.
DocumentDataSubType enumeration	Removed	This enumeration is obsolete. Use the **ChipDataGroup** enumeration instead.
DocumentDataType enumeration	Changed	Removed the **ContactlessChip** member
LightSource enumeration	Added	New enumeration

Image requirements and recommendations

The minimum requirement for processing a document with AssureID Connect is posting the white (light source) front (side) image of the document. There are specific imaging requirements that must be met to ensure that a document is correctly and accurately processed. These include cropping, resolution, and compression.

Note All images should be 24-bit RGB, except Nearinfrared images, which should be 8-bit grayscale.

The following table describes the supported image file formats and their respective file extensions.

Format	File extension
Windows Bitmap	bmp
PNG	png
TIFF	tiff
JPEG	jpeg
JPEG-XR	vnd.ms-photo
JPEG 2000	jp2

Image cropping

The following table describes the settings for image cropping.

Cropping mode	Description
Automatic	Automatically determines whether image cropping is required on each image and performs the cropping and calculates image resolution when necessary. **Note** Acuant recommends using the Automatic setting because it intelligently determines whether cropping is required for each image, avoiding unnecessary processing.
Always	Cropping will always be performed on posted images. Automatic cropping is the preferred mode, as it avoids unnecessary processing, but this mode will ensure that cropping is always attempted, resulting in additional processing time when cropped images are submitted.
None	No cropping will be performed on posted images. This mode should only be used when the images are already tightly cropped to the document when posted. This is the default setting.

Use the **Automatic** setting if the image is not tightly cropped to the edge of the document to ensure that the images are automatically cropped when posted. If the cropping mode is set to **None** and the expected image size is not specified as ID1 or ID3, then the physical resolution of image, as specified within the image header, *must* be specified and accurate to within 1% of the actual image resolution. In some rare cases, using the **Automatic** setting may skip cropping when it is actually required; therefore, in these cases, Acuant recommends setting the cropping mode to **Always**.

Image resolution

Although the minimum supported ***physical*** resolution is 300 DPI (dots per inch), higher resolutions can improve document classification and OCR accuracy. For document authentication, Acuant *strongly* recommends using a minimum resolution of 600 DPI.

Note The original images from an imaging device should not be resized or scaled to a smaller size, unless the resolution is very high (greater than 800 DPI).

Images with a resolution greater than 800 DPI should be scaled down in size and resolution *before* compression to optimize processing performance. The aspect ratio of the image should not be changed, and the horizontal and vertical resolution of the image should ideally be within 1% of each other.

Note The AssureID Connect photo printing and substrate printing tests require a high-quality image to be effective. Submitting a low-resolution document image may result in these tests being skipped.

Image compression

Do *not* excessively compress an image. Overcompression may degrade data capture and authentication performance. Use the following recommended minimum compressed image file sizes that correspond to the applicable light source.

Light Source	Minimum	Recommended
Visible	500 KB	600-1200 KB
Near-Infrared	200 KB	250-500 KB
Ultraviolet	125 KB	150-300 KB
Coaxial	125 KB	150-300 KB

Note The AssureID Connect photo printing and substrate printing tests may be affected by compression, therefore you should avoid overcompressing an image. Submitting an overcompressed image may result in these tests being skipped.

HTTP Status codes

The following table lists the HTTP status codes and which API calls use these codes.

Code	Summary	Description	API Calls
401	Unauthorized	The caller could not be authenticated (typically due to invalid credentials).	All
403	Forbidden	The referenced or default subscription may be inactive.	All
404	Not Found	The resource, document, or document data could not be found.	All
409	Conflict	An attempt was made to submit the same document image, data, or chip data multiple times (e.g. the Front/White image may only be submitted once per transaction).	PostDocumentImage, PostDocumentChipData, PostDocumentChipAuthentication
430	Required service operation argument not specified	One or more of the required arguments to the service operation were not specified.	GetDocumentData, GetDocumentFieldData, GetDocumentFieldImage, PostDocumentChipAuthentication, PostDocumentChipData, PostDocumentData, PostDocumentImage, PostDocumentInstance
431	Invalid or unsupported service operation argument	One or more of the arguments supplied to the service operation are invalid or unsupported.	PostDocumentImage
432	Document settings incomplete	The supplied document settings do not contain all of the required information necessary for processing. At a minimum, a subscription and a document capture device that includes manufacturer and model names are required. If manual classification is requested, a valid document type is also required.	PostDocumentInstance
433	Resource identifier not in expected format	The identifier used to reference a specific resource was not supplied in the expected format.	All
434	Subscription not found	The referenced subscription was not found or is temporarily unavailable.	PostDocumentInstance
435	Subscription not specified	Multiple subscriptions were found for the authenticated user and one must be specified to invoke this service operation.	PostDocumentInstance
436	Manual classification document type not found.	The referenced document type used for manual classification was not found or is temporarily unavailable.	PostDocumentInstance
437	Document complete or in error state.	The status of the referenced document is complete or in an error state; additional images and/or data cannot be posted to the document and no further processing may be performed.	All, except PostDocumentInstance

<table>
<tr><th>Code</th><th>Summary</th><th>Description</th><th>API Calls</th></tr>
<tr><td>438</td><td>Document image load failure.</td><td>The document image could not be loaded from the supplied stream due to a format, compatibility or integrity issue.</td><td>PostDocumentImage</td></tr>
<tr><td>439</td><td>Invalid or unsupported document image pixel depth.</td><td>The pixel depth of the document image is invalid or unsupported. The following image pixel depths are supported:
<table>
<tr><th>Lighting</th><th>Pixel Depth</th></tr>
<tr><td>Visible</td><td>24-bit RGB</td></tr>
<tr><td>Ultraviolet</td><td>24-bit RGB</td></tr>
<tr><td>Near-Infrared</td><td>8-bit Grayscale</td></tr>
</table></td><td>PostDocumentImage</td></tr>
<tr><td>440</td><td>Document image size is outside of the acceptable range.</td><td>The width and/or height of the document image must not be less than 100 pixels.</td><td>PostDocumentImage</td></tr>
<tr><td>441</td><td>Document image resolution is outside of the acceptable range.</td><td>The resolution of the document image must be greater than 96 DPI and must not exceed 1,500 DPI.</td><td>PostDocumentImage</td></tr>
<tr><td>442</td><td>Document image resolution difference between each axis is outside of the acceptable range.</td><td>The resolution difference between the horizontal and vertical axis of the document image must not exceed 5%.</td><td>PostDocumentImage</td></tr>
<tr><td>500</td><td>Internal Server Error.</td><td>An internal error occurred. See system log.</td><td>All</td></tr>
<tr><td>503</td><td>Service Unavailable.</td><td>The website may be down, it may not be configured properly, or it may have insufficient credentials to run the website.</td><td>All</td></tr>
</table>