APRIL 10, 2024

GSCCCA UCC EFILING COUNTY IMPLEMENTATION GUIDE

WILLIAMS, CHAS

GEORGIA SUPERIOR COURT CLERKS' COOPERATIVE AUTHORITY

https://gsccca.org

REVISION HISTORY

Revision	Description	Date
1	Initial Version	08/01/2019
1.1	Improved clarity of text	10/20/2020
1.2	Test site reference changed to https://efilestg.gsccca.org/	11/09/2020
1.3	API Endpoints for retrieving attachments and rotating PDFs added	04/10/2024

CONTENTS

INFORMATION FOR IMPLEMENTERS

The Georgia Superior Court Clerks' Cooperative Authority offers a public Simple Object Access Protocol (SOAP) API that allows the submission, payment, and processing of UCC Filings. This API allows integration with filer and county systems to facilitate an automated filing process. API users can seamlessly move between their own system and the https://efile.gsccca.org eFiling dashboard to view, accept, and reject filings in their queue.

ASSUMPTIONS

This is a technical document intended for developers and service providers who wish to utilize the UCC eFile API programmatically. Anyone reading this document should be well-versed in XML and XSDs as well as have knowledge on how to connect to a SOAP web service to make calls and process responses. Expectations are that readers will be familiar with the basics of the IACA File XML and IACA Status and Receipt XML formats. Lastly, familiarity with UCCs and Georgia's laws will be beneficial.

DOCUMENTATION

Visit our eFile API Integration page to access various guides and downloadable files.

MODIFIED IACA FILE/STATUS DOCUMENTATION

The GSCCCA has made various modifications to IACA's XML schemas for filing UCCs. On our eFile API Integration page, you can find schema definitions for download as well as documentation that explains in detail XML requirements for those modified versions of the IACA File XML and IACA Receipt and Status XML.

INTEGRATION API

The GSCCCA offers a downloadable DLL that can be referenced by other applications to help ease integration with the GSCCCA UCC eFile API. This library contains functionality to make calls to all public web methods and to validate and serialize to and from XML and objects. Detailed documentation is located on the Implementer's Page.

The GSCCCA makes available the DLLs as well as source code and a demo application that uses examples to show how to use the Integration API. A help website as well as downloadable file have been made available as well.

SUPPORT/CONTACT

For all https://efile.gsccca.org and eFiling related questions, please contact our customer support at (800) 304-5174 or email at help@gsccca.org. Customer service will be able to help you or escalate any issues or questions to the correct Authority personnel.

WHERE DO I CONNECT?

The UCC eFile API is a SOAP service that provides various web methods reproducing the functionality found on the eFile website. This allows users to programmatically accept and reject submitted UCCs via third party applications.

TEST SERVICE ENDPOINTS

For testing, the GSCCCA makes available a test environment. To begin testing, connect to the following endpoints. The Account Service endpoint is used to query and manipulate a user's account. The UCC Service allows users to interface with the filing process.

The public Account API endpoint is here:

➢ https://efilestg.gsccca.org/efilewebservice/accountservice.asmx

The public UCC eFile API endpoint is here:

➢ https://efilestg.gsccca.org/efilewebservice/uccservice.asmx

PRODUCTION SERVICE ENDPOINTS

Once you have sufficiently tested and are ready to begin filing UCCs through the GSCCCA UCC eFile API, use the below production URLs.

The public Account API endpoint is here:

➢ https://efile.gsccca.org/efilewebservice/accountservice.asmx

The public UCC eFile API endpoint is here:

➢ https://efile.gsccca.org/efilewebservice/uccservice.asmx

ACCOUNT CREATION

In order to call our web services, users must have an account on our eFile website. This account must be enabled as a county user by our administrative staff. Please contact us in order to activate access for your account so that you can begin testing.

When testing, be sure to create an account on the test website and not the production website. This username and password must be used for most web method calls found on the above endpoints.

Our test website can be located at
https://efilestg.gsccca.org

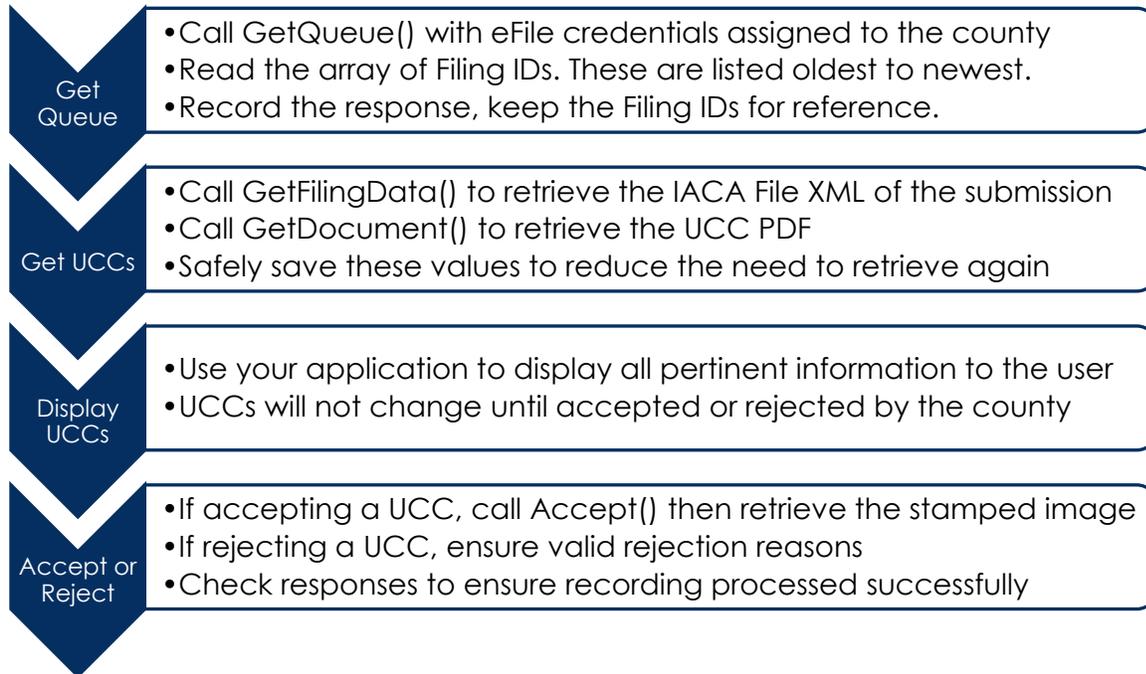Our production website can be located at
https://efile.gsccca.org

GSCCCA EFILE WORKFLOW

The workflow for the GSCCCA UCC eFile API mirrors that of filing through the eFile website.

PROCESS UCC QUEUE

Once a filer submits a UCC to the GSCCCA, this submission is available for the counties to view and process. Query the API to obtain a queue of UCCs by filing ID, and then process each one individually accepting or rejecting the submitted UCCs.

Get Queue
- Call GetQueue() with eFile credentials assigned to the county
- Read the array of Filing IDs. These are listed oldest to newest.
- Record the response, keep the Filing IDs for reference.

Get UCCs
- Call GetFilingData() to retrieve the IACA File XML of the submission
- Call GetDocument() to retrieve the UCC PDF
- Safely save these values to reduce the need to retrieve again

Display UCCs
- Use your application to display all pertinent information to the user
- UCCs will not change until accepted or rejected by the county

Accept or Reject
- If accepting a UCC, call Accept() then retrieve the stamped image
- If rejecting a UCC, ensure valid rejection reasons
- Check responses to ensure recording processed successfully

UCC STAMPING

All effort is made to ensure that UCCs are properly stamped by the eFile system. In the case where a stamp fails, the IACA File XML returned by the Accept() call will list the status of the UCC as NoStamp. This indicates that the UCC was successfully accepted and payment collected, but it will need a manual stamp by the county office. The unstamped image can be retrieved by calling GetDocument() on the Filing ID.

FILE NUMBER ASSIGNMENT

As an optional county opt-in feature, the Authority offers to enable eFile Only for counties. When opted in, the GSCCCA eFile system will assign file numbers to accepted UCCs with incrementing sequence numbers.

If opted in, callers to Accept() must pass in an empty string or null string as the filenumber parameter. The eFile system will then assign the next file number in sequence. This file number will be included in the response IACA File XML to the Accept() call. Callers may then record this value and display it to the county user.

WHAT APIS ARE PROVIDED?

Below are listed all web methods that have been made available for filers to facilitate the submission and processing of UCCs. Each definition includes a description of what the web method is used for, the signature including the return type, the web method name, and the parameters. A brief description of the response is included along with any remarks that may be pertinent to how to use the given web method.

ACCOUNTSERVICE.CHECKCREDENTIALS

Call as a test to determine connectivity and if the username and password are able to make calls to the public web APIs.

SIGNATURE

`bool CheckCredentials(string user, string pass)`

> user: eFile website username
> pass: eFile website password

The response is true if validation succeeds.

UCCSERVICE.ACCEPT

Call to accept a UCC.

SIGNATURE A

`string Accept(string user, string pass, int id, string filenumber)`

> user: eFile website username
> pass: eFile website password
> id: Filing ID to accept
> filenumber: File number

SIGNATURE B

`string AcceptSetDate(string user, string pass, int id, string filenumber, string datetime)`

> user: eFile website username
> pass: eFile website password
> id: Filing ID to accept
> filenumber; File number
> datetime: Datetime as a string to set as the accepted date and time (backdate)

The return is the final IACA File XML string that is in the format in which it will be sent to the filer. This XML includes the assigned number and collected fees.

The UCC will be marked as Accepted and the system will process payment. The system will stamp the PDF.

REMARKS

This web method returns the modified IACA File XML. The county can then retrieve the stamped image by using the UCCService.GetDocument() web method call.

It is possible to backdate UCCs in the eFile system. If a UCC needs to be backdated, call the second signature and pass in the date to stamp on the document. As usual, all images can be viewed and downloaded on the eFile Website dashboard by any user and notifications are created for each UCC acceptance.

If a county is eFile only for UCCs and does not take paper filings, then the option is available to have the GSCCCA increment the sequence for the file number. If the county opts into this feature, then the `string filenumber` parameter must be an empty string or null. The assigned file number can then be read from the returned XML.

UCCSERVICE.ARCHIVE

Call to archive a UCC on an eFile website dashboard.

SIGNATURE

`bool Archive(string user, string pass, int id)`

> user: eFile website username
> pass: eFile website password
> id: The Filing ID of the UCC to archive

The response is a boolean value that is true if the archival was a success.

UCCSERVICE.GETFILINGDATA

Call to get the XML data for a UCC.

SIGNATURE

`string GetFilingData(string user, string pass, int id)`

> user: eFile website username
> pass: eFile website password
> id: The Filing ID of the UCC to retrieve

The response is the IACA File XML with all current data included of the UCC.

REMARKS

As a UCC goes through workflow, the XML data returned will change from the original submission XML to the accepted or rejected XML from the county.

UCCSERVICE.GETFILINGDOCUMENT

Call to get the PDF document for a UCC.

SIGNATURE

```
byte[] GetFilingDocument(string user, string pass, int id)
```

> user: eFile website username
> pass: eFile website password
> id: The Filing ID of the UCC to retrieve

The response is the current PDF image of the UCC document.

REMARKS

Prior to acceptance, the UCC will only have an initial stamp. After acceptance by the county, all UCCs will be stamped with recording information. This web method will always return the most recent and up to date document in our system.

UCCSERVICE.GETFILINGATTACHMENT

Call to get the PDF attachment for a UCC.

SIGNATURE

```
byte[] GetFilingAttachment(string user, string pass, int id)
```

> user: eFile website username
> pass: eFile website password
> id: The Filing ID of the UCC to retrieve

The response is the current PDF image of the documents attached to the UCC.

REMARKS

When submitting a UCC, filers can attach one or more additional files. This call retrieves all attachment files as a single PDF and returns them to the caller as a byte array. If the UCC has been accepted, the returned PDF will be stamped.

UCCSERVICE.GETREJECTIONREASONS

Allows the caller to retrieve a list of valid rejection reasons.

SIGNATURE

```
string[] GetRejectionReasons()
```

The return is an array of valid UCC rejection reasons as defined by statute.

REMARKS

When calling reject, the rejection reasons given, with the exception below, must exactly match the rejection reasons listed by the return value from this web method. The county is allowed a single, free form rejection reason that can be given during rejection that is not on this list.

UCCSERVICE.GETQUEUE

Call to get an array of all UCC Filing IDs that are in the Awaiting Clerk Review status.

SIGNATURE

```
string[] GetQueue(string user, string pass)
```

user: eFile website username
pass: eFile website password

The return value is an array of strings, where each value is a Filing ID

UCCSERVICE.REJECT

Call to reject a UCC.

SIGNATURE

```
bool Reject(string user, string pass, int id, string[] reasons)
```

user: eFile website username
pass: eFile website password
id: Filing ID to accept
reasons: Array of rejection reasons

The return value is a boolean that is true on successful rejection.

REMARKS

Only valid rejection reasons and a single free form reason may be included in this call. County users can retrieve those values by calling the UCCService.GetRejectionReasons() web method.

UCCSERVICE.ROTATEFILINGDOCUMENT

Used to rotate an upside down UCC Filing Document or page180 degrees to be right side up.

SIGNATURE

```
byte[] RotateFilingDocument(string user, string pass, int id, int page)
```

user: eFile website username
pass: eFile website password
id: The Filing ID of the UCC to rotate
page: The page number of the UCC to rotate

The response is the modified UCC PDF.

REMARKS

When a filer uploads a UCC image with one or more pages which have an incorrect orientation, this call can be used to correct the orientation prior to acceptance. This call rotates the UCC Document page indicated by 180 degrees. In order to rotate all pages of the UCC at once, the value "0" can be passed in as the page parameter to the call.

UCCSERVICE.ROTATEFILINGATTACHMENT

Used to rotate an upside down UCC Filing Attachment or page 180 degrees to be right side up.

SIGNATURE

```
byte[] RotateFilingDocument(string user, string pass, int id, int page)
```

> user: eFile website username
> pass: eFile website password
> id: The Filing ID of the UCC to rotate
> page: The page number of the attachment to rotate

The response is the modified PDF attachment PDF.

REMARKS

When a filer uploads a UCC attachment with one or more pages which have an incorrect orientation, this call can be used to correct the orientation prior to acceptance. This call rotates the UCC Attachment page indicated by 180 degrees. In order to rotate all pages of the UCC at once, the value "0" can be passed in as the page parameter to the call.

UCCSERVICE.UNREJECT

Call to unreject a UCC and put it back in awaiting clerk review.

SIGNATURE

```
bool Unreject(string user, string pass, int id)
```

> user: eFile website username
> pass: eFile website password
> id: Filing ID to accept

The return value is a boolean that is true on successful unrejection.

REMARKS

Once a filer cancels a UCC this can no longer be called on the filing.

ERRORS AND SERVICE INTERRUPTIONS

All errors will cause a SOAP fault envelope in the response to the client. Please check your framework documentation to determine how your application will interpret the fault envelope. Errors can be caused by a wide range of issues, but they are generally broken down into client errors and server errors.

The fault envelope consists of a fault code, an actor, and a detail element.

The fault code will indicate whether the error was processed as a client or server fault. Client faults indicate something that was wrong with the submission in some way. This can be invalid credentials or authorization, a problem validating or parsing input parameters, or similar. If a client fault is created, then resubmission without changing the problem value will almost certainly lead to the same fault being created.

Server fault codes include things like database timeouts, IO errors, and other server faults that the client has no control over. In most cases, waiting and then attempting the call again will produce different results in the case of server faults.

The fault actor is the piece of code that caused the error. In almost all cases, this will be the calling web method.

The fault detail element is an XML string that contains basic information about the error that the client can parse for error information. For example, the error "Invalid username or password" would be found in the detail element. The XML string will also contain an error code specific to the GSCCCA UCC eFiling system which if provided can help us track the problem. Sometimes different errors will return the same error messages but different codes. In those cases, the code becomes useful for tracking down exactly what the issue may be.

It is highly recommended that the caller log all pertinent SOAP fault data in order to help troubleshoot any issues that users may experience.

DATA AND IMAGE REQUIREMENTS

For initial implementation of the UCC eFile API, Level 2 transfer has been implemented. While UCC data is transferred via XML, the official document is the attached PDF. This PDF is considered the submitted filing by the county, and all acceptances and rejections are based on the PDF and not the XML data. The stamped PDF is then the official UCC recorded document, regardless of what data may be found in the XML even if it is contradictory.