

GEORGIA SUPERIOR COURT CLERKS'
COOPERATIVE AUTHORITY

Real Estate eFiling Portal

County Implementation Guide and Transmission Standards

11/01/2018

Recent Changes

October 19, 2010:

- Initial Version

January 21, 2011:

- Corrected two typos. One of the typos occurred in the method declaration of the “RetrieveRejectedPackage” subsection on page 10. The functional prototype read “RetrieveRecordedPackage” and has been corrected to read “RetrieveRejectedPackage” per the section title. The other typo was a simple spelling error of no major consequence.

December 9, 2011:

- Changed the subtitle of this document from “County Implementation Guide” to “County Implementation Guide and Transmission Standards” in an effort to better convey the intent of the document.
- Added item #7 to the remarks section of the “GetPendingPackage()” service. This item discusses the placement of Participant IDs in the PRIA document, and verification practices.
- Added item #8 to the remarks section of the “GetPendingPackage()” service. This item discusses recording date and time requirements.
- Added item #9 to the remarks section of the “GetPendingPackage()” service. This item discusses non-recordable documents.
- Inserted a new item #16 to the remarks section of the “AcceptPackage()” service and renumbered other items appropriately. This item discusses the requirements for electronically stamping certain information on accepted documents.

March 13, 2012:

NOTE: Revisions in this change list relating to system changes are scheduled to take effect April 2, 2012

- Amended item #7 to the remarks section of the “GetPendingPackage()” service. This item discusses the placement of Participant IDs in the PRIA document, and verification practices.
- Fixed a typo in item #2 of the AcceptPackage() service. PRIA_REQUEST_GROUP has been changed to REQUEST_GROUP.

September 21, 2015:

- Changed http to https throughout document where appropriate.
- Removed all references to the existence of an invalid Certificate for the eFile test portal. This site now uses a valid SSL Certification.
- Fixed typos throughout the document.

January 26, 2017:

- Updated Minimum Data Requirements

July 26, 2017:

- Added new requirements for State Tax Liens

November 01, 2018:

- Fixed link to Implementer's Page
- Fixed link to Filer Guide
- Added information for State Tax Lien control number
- Added information for reading cross references
- Added information for payment method indicator
- Added intangible tax fee description requirements
- Included example for parsing and using non-recordable document
- Added general information on how to use GetPendingPackage() and workflow
- Added technical specifications for reading fee estimates
- Added web method ValidatePT61Status

Purpose

This technical document is intended for information technology personnel (programmers) who service those who wish to receive and process electronically submitted real property records from the Georgia Superior Court Clerks Cooperative Authority eFiling portal. The document describes the interface for automated retrieval and processing of such electronic filings to Clerks of Superior Court through the GSCCCA Real Estate eFiling Portal. This document does not address eFiling through the use of individual web browsers, nor the interface for filers to submit documents to the GSCCCA; such specifications are provided in other documents.

Assumptions

This document assumes the reader has a working understanding of the PRIA 2.4.1 standard for electronic recording of real estate filings. More information about PRIA and the standard can be found at <http://www.pria.us>. Furthermore, it is assumed that the reader has experience developing applications that can produce well-formed and schema compliant XML documents, parse XML documents, and consume web services based on a published WSDL.

Account Creation

All counties and vendor systems who will retrieve and process filings from the GSCCCA web portal must first create a filer account. Accounts can be created online by visiting <https://efile.gsccca.org> and selecting the “Register” link in the “Account Management Menu”. General information about the eFiling website and account creation is available in the GSCCCA Filer guide located at <https://efile.gsccca.org/flrguide.aspx>. The filer account must be granted special permissions to allow certain actions such as accepting and rejecting filings. To request elevated account permissions, please send a request to help@gsccca.org.

Web Service URL

The GSCCCA Real Estate web service portal can be accessed at <https://eFile.gsccca.org/eFileWebService/REService.aspx>

The formal service description (WSDL) is available at: <https://eFile.gsccca.org/eFileWebService/REService.aspx?WSDL>

Test Portal Web Service

The GSCCCA maintains a separate environment for processing test filings. The system is a mirror of the production system but uses a separate database. For this reason, users of the test portal must have a separate account created on the testing website. The test server for account creation is located at

<https://efiledev.gsccca.org>.

The web service for submissions is located at

<https://efiledev.gsccca.org/eFileWebService/REService.asmx>.

The formal service description (WSDL) is available at:

<https://efiledev.gsccca.org/eFileWebService/REService.asmx?WSDL>

Access to the test portal is IP restricted and access must be requested by sending email to

help@gsccca.org.

GSCCCA Real Estate eFiling API

The GSCCCA publishes the web services described in this document in order to provide open access across all systems and service platforms. Use of the web services alone requires a fairly comprehensive understanding of the PRIA standard. To ease the burden of implementation, the GSCCCA has developed and freely distributes a .NET DLL that provides simplified access to use and consume these web services. The API handles the details of conforming to the PRIA standard behind the scenes while exposing a familiar and simplified object model for implementers. More information about the GSCCCA Real Estate eFiling API is provided in the GSCCCA Real Estate eFiling API Implementation Guide, which is accessible at <https://efile.gsccca.org/implementers.htm>

Services Provided

The following web methods are provided for the submission, tracking, and retrieval of processed Real Estate filings.

- `GetPendingPackages()`
- `GetPendingPackage()`
- `AcceptPackage()`
- `RejectPackage()`
- `RetrieveRecordedPackage()`
- `RetrieveRejectedPackage()`
- `ValidatePT61Status()`

All methods return an XML string that represents a well-formed, PRIA 2.4.1 Response. Only the AcceptPackage() method expects to receive a well formatted PRIA Request envelope as a parameter. However, all method invocations are considered PRIA requests and will generate replies that conform to the PRIA response envelope. All responses will include a **RESPONSE_GROUP** root element, a child **RESPONSE** element, and a child **STATUS** element. Responses that do NOT contain error messages may also contain a single **RESPONSE_DATA** element with at least one **PRIA_RESPONSE** elements.

GetPendingPackagesMETHOD

Syntax

```
public string GetPendingPackages(string userID, string passwd)
```

Parameters

userID

Type: string

The previously established userID of the eFiling User

passwd

Type: string

The previously established password of the eFiling User

Return Value

Type: string

A XML string that represents a well-formed PRIA response including zero or more PRIA Packages.

Remarks

- 1) The filer is identified by the user id and password provided with the method invocation. The method invocation will retrieve the list of REAL ESTATE filings that are pending for the county associated with the logged in user.
- 2) The response will contain a **PACKAGE** for each filing that is pending. The PACKAGE element will be except for the **PACKAGE._Identifier** attribute which will contain the GSCCCA assigned ID for the package.
- 3) Subsequent actions on the package such as retrieving the package, accepting and or rejecting will require the GSCCCA ID supplied in the **PACKAGE._Identifier** attribute.

GetPendingPackageMETHOD

Syntax

```
public string GetPendingPackage(string userID, string passwd, string packageGUID)
```

Parameters

userID

Type: string

The previously established userID of the eFiling User

passwd

Type: string

The previously established password of the eFiling User

packageGUID

Type: string

The GSCCCA assigned package identifier for the document being requested

Return Value

Type: string

A XML string that represents a well-formed PRIA response including a single PRIA Packages that contains one or more filings

Remarks

- 1) The county user is identified by the user id and password provided with the method invocation. The package can only be obtained by an account associated with the county in which the package was filed.
- 2) The Package GUID is the GSCCCA system generated identifier that was returned in the **PACKAGE_Identifier** attribute of the original submission request
- 3) The Package GUID must be well formed. Supplying a nonconforming GUID will result in an error message being returned.
- 4) If the filing identified by the filing GUID cannot be found, an error message is generated.
- 5) The requested document must be in the "Pending" state. If the requested filing is not in the "Pending" state, an error message will be generated.
- 6) The response is guaranteed to contain a single **PACKAGE** element with one or more **PRIA_DOCUMENT** elements. Each **PRIA_DOCUMENT** element will always contain one or more **EMBEDDED_FILE** elements. The embedded file will convey the document as presented by the filer.
- 7) Each **PRIA_DOCUMENT** element is guaranteed to contain a **PARTIES** element with one or more child **_PREPARED_BY_PARTY** elements. Each **_PREPARED_BY_PARTY** element present corresponds to a Participant ID that is associated with the document, and the order in which these elements appear corresponds to the order in which the participant became associated with the document. Also, each **_PREPARED_BY_PARTY** will be guaranteed to have the following attributes set.
 - a. **_ID**: The eFile Registry Participant ID of the filing participant.
 - b. **_SequenceIdentifier**: A number that indicates the order in which the participant become associated with the filing. The **_SequenceIdentifier** ordinal will always match the order in which each **_PREPARED_BY_PARTY** element is presented (i.e., the first element will always be "1", the next "2", and so on.
 - c. **_TitleDescription**: The type of Participant ID specified in the **_ID** attribute. Valid values are "TRUSTER_USER", "SUBMITTER", and "SELF_FILER".
 - d. **_UnparsedName**: The name of the Participant for which the participant ID is associated with.

Each participant ID conveyed by a **_PREPARED_BY_PARTY** node has been previously validated by the GSCCCA eFiling registry system for validity. However, it is highly recommended that the county record management system revalidate each participant ID at the time of processing. This practice will ensure that there have been no adverse changes to the status of any participant ID during the time between receiving the filing and processing. Furthermore, certain participants are assigned a participant ID as a self-filer, which requires manual document inspection to verify the validity of the filing. In these cases, the record management system should alert the clerk staff to verify that the self-filer participant is actually a party to the document(s) being filed. The name of the participant is located in the **_UnparsedName** attribute and can be used to help the clerk verify the document. Information about verifying participant IDs can be found at:

<https://www.efileregistry.org/Resources/APlandCodeExamples.aspx>

- 8) Each **PRIA_DOCUMENT** element will contain a placeholder **RECORDING_ENDORSEMENT** element with the **RECORDING_ENDORSEMENT_RecordedDateTime** attribute set to the date and time that the document was received by the GSCCCA portal. Per Georgia standards, this date and time will be the official date and time recorded should the document be subsequently accepted by a county.
- 9) It is important to observe the **DocumentNonRecordableIndicator** attribute of each **DOCUMENT** element when processing the filing. This attribute will specify whether or not the document is intended to be recorded, or if it is an attachment sent to support the filing without becoming part of the recorded document
- 10) A package will not change once in a pending state, except by the county's vendor system. There is never a need to call to get the pending package more than once.
- 11) When the Authority is processing the filer's payment, the **PAYMENT** element will indicate how the filer is paying. When the Authority is not processing the payment, this element will be excluded from the XML.
- 12) As an opt in feature, counties may receive fee estimates along with the PRIA xml. Fee estimates are listed in the **CONSIDERATION** element of the **PRIA_DOCUMENT**. Each fee estimate will be of **_Type "Other"** with a description of Estimated Base Filing Fee, Estimated Assigned Fees, Estimated Documents Being Cancelled Fees, Estimated Cross Index Fees, Estimated PT-61 Tax, Estimated Intangible Tax or Estimated Page Fees.

Examples

Non-recordable documents

Many packages may include non-recordable documents alongside a deed, plat or lien. These non-recordable documents consist of required supporting documents, notes from the filer or other images that are helpful for processing but that will not be stamped or recorded in the docket alongside the instrument. When accepting the document and sending the stamped images back to the filer, non-

recordable documents should not be stamped and do not need to be returned. Below is an example of what a non-recordable document looks like.

```
<EMBEDDED_FILE ID="fa911f49-97d3-4441-9e75-57cde6b5d9b0" FileEncodingType="base64"
  MIMETYPE="image/tiff">
  <DOCUMENT DocumentNonRecordableIndicator="Y" RecordableDocumentSequenceIdentifier="1"
    RecordableDocumentType="Deed">[[PT-61 Base 64 Data Here]]</DOCUMENT>
</EMBEDDED_FILE>
<EMBEDDED_FILE ID="d35db555-34d1-412e-ba63-b62c08c5d407" FileEncodingType="base64"
  MIMETYPE="image/tiff">
  <DOCUMENT RecordableDocumentSequenceIdentifier="2" RecordableDocumentType="Deed">[[Deed Base
    64 Data Here]]</DOCUMENT>
</EMBEDDED_FILE>
```

AcceptPackage METHOD

Syntax

```
public string AcceptPackage(string userID, string passwd, string xml)
```

Parameters

userID

Type: string

The previously established userID of the eFiling User

passwd

Type: string

The previously established password of the eFiling Use

xml

Type: string

An xml string representing a well-formed PRIA request envelope with pria documents that contains endorsement and fee information

Return Value

Type: string

A XML string that represents a well-formed PRIA response including a single PRIA Status element

Remarks

- 1) The county user is identified by the user id and password provided with the method invocation. The package can only be Accepted by an account associated with the county in which the document was filed.
- 2) All calls to the AcceptPackage() method must contain a well-formed, PRIA 2.4.1 compliant **REQUEST_GROUP** root element.
- 3) The PRIA standard allows for each **PRIA_REQUEST_GROUP** document to include one or more **REQUEST** elements. The GSCCCA system allows ONLY a single **REQUEST** element per transaction. Transactions with multiple **REQUEST** elements will be rejected.

- 4) Each **REQUEST** element in the PRIA 2.4.1 standard allows for one or more **PRIA_REQUEST** elements. The GSCCCA system allows only a single **PRIA_REQUEST** element per **REQUEST**. Transactions with multiple **PRIA_REQUEST** elements will be rejected.
- 5) Each **PRIA_REQUEST** element in the PRIA 2.4.1 standard allows for a single **PACKAGE** element. Per rules 2 and 3 above, only a single **PACKAGE** element is allowed per transaction.
 - 6) Each **PACKAGE** element in the PRIA 2.4.1 standard allows for one or more **PRIA_DOCUMENT** elements. The package being accepted should contain as many **PRIA_DOCUMENT** elements as was submitted with the original request, and the **_UniqueIdentifier** attribute of each **PRIA_DOCUMENT** being accepted must match the **_UniqueIdentifier** attribute of the originally submitted filings. Acceptance is performed at the package level (meaning all documents in a package are accepted/rejected together).
 - 7) The **_ID** attribute **PRIA_DOCUMENT** element being accepted must match the **_ID** attribute that was originally submitted. This **_ID** is an identifier originating with the filer. In the case of a StateTaxLien or ReleaseOfStateTaxLien this is the control number (AKA REV number or execution number) of the document that has been submitted.
 - 8) In order to ensure proper server validation, it is important to correctly set the **DocumentNonRecordableIndicator** attribute of each **PRIA_DOCUMENT** element. Documents that are not marked non-recordable will be required to have endorsement information and fee information as they are assumed to have been recorded.
 - 9) The **PACKAGE._Identifier** must be set with a valid GSCCCA assigned GUID that is associated with the document to be accepted.
 - 10) The **PACKAGE._Identifier** must be well formed. Supplying a nonconforming GUID will result in an error message being returned.
 - 11) If the filing identified by the **PACKAGE._Identifier** cannot be found, an error message is generated.
 - 12) The requested document must be in the "Pending" state. If the requested filing is not in the "Pending" state, an error message will be generated.
 - 13) Each **PRIA_DOCUMENT** can contain one or more **EMBEDDED_FILE** elements. The GSCCCA system requires all AcceptPackage method invocations to include AT LEAST one **EMBEDDED_FILE** element. Transactions without at least one **EMBEDDED_FILE** element will be rejected.
 - 14) Each **EMBEDDED_FILE** element must include a valid value for the **MIMETYPE** attribute. The mime type must accurately describe the image represented by the **EMBEDDED_FILE** element, and must be one of the allowed MIME Types. Missing MIME Types, or mime types not allowed by the GSCCCA system will cause the transaction to be rejected. Allowed MIME Types are restricted to the following:
 - a. IMAGE/TIFF
 - b. IMAGE/TIF
 - 15) Each **EMBEDDED_FILE** element must include a valid value for the **FileEncodingType** attribute.
 - 16) Each **EMBEDDED_FILE** element must include a valid **DOCUMENT** element as defined by the PRIA 2.4.1 standard.
 - 17) The binary file embedded in the **DOCUMENT** should be electronically stamped with the

following information:

- a. The docket, book, and page of the accepted document. The style of this stamp should reflect existing county practices for marking accepted documents with this information
 - b. The date and time of recording (as originally specified in the **RECORDING_ENDORSEMENT._RecordedDateTime** attribute of the document when the “GetPendingPackage()” service was invoked).
 - c. Each Participant ID transmitted with the filing must be stamped on the document in the order in which it was presented in the filing XML.
- 18) Each **PRIA_DOCUMENT** can contain a single **RECORDING_ENDORSEMENT** element. The **RECORDING_ENDORSEMENT** element is required for all AcceptPackage invocations.
 - 19) The **RECORDING_ENDORSEMENT._Volume** attribute must contain the BOOK in which the filing is recorded.
 - 20) The **RECORDING_ENDORSEMENT._VolumeType** attribute must be set to a valid recording docket
 - 21) The **_VOLUME_PAGE** element must be present within the **RECORDING_ENDORSEMENT** element.
 - 22) The **_VOLUME_PAGE._NumberIdentifier** must be set with the first page of the recorded document.
 - 23) NOTE: If the document is not recorded using a volume/book/page docketing system, the **_InstrumentNumberIdentifier** attribute of the **RECORDING_ENDORSEMENT** must be set rather than the elements/attributes specified in rules 18-21.
 - 24) The **RECORDING_ENDORSEMENT._RecordedDateTime** attribute must be set with a valid date and time value.
 - 25) The **_FEES** element must be present, and the **_FEES._TotalAmount** attribute must contain a valid money amount.
 - 26) Specific recording fee information is required to be provided in the **_RECORDING_FEES** with a **RecordingEndorsementFeeDescription** of “Intangible Tax” (case sensitive) when the **RecordableDocumentType** is Mortgage and when the county collects intangible tax. This value may be listed at \$0.00 charged. If a value greater than \$25,000 is listed, the Authority system will return an error message.
 - 27) The **PRIA_DOCUMENT** element can contain a **RECORDABLE_DOCUMENT** element with one or more **_ASSOCIATED_DOCUMENT** elements included. Each **_ASSOCIATED_DOCUMENT** element represents a single cross-index that the filer believes is applicable to the filing. Each **_ASSOCIATED_DOCUMENT** element will have an **_ID**, **_BookNumber**, **_PageNumber**, **_CountyOfRecordingName**, **_RecordingDate**, and **_SequenceIdentifier** attribute. Each of these attributes reference a cross-indexed filing, not the current filing.
 - 28) The **_ASSOCIATED_DOCUMENT** elements may also have a **_Code** attribute if the **RecordableDocumentType** is a ReleaseOfStateLien. If the **_Code** attribute is “Release” then this is a cross index *and* a lien that is being released.

RejectPackage METHOD

Syntax

```
public string RejectPackage(string userID,  
    string passwd, string packageGUID,  
    string[] rejectionReasons)
```

Parameters

userID

Type: string

The previously established userID of the eFiling user

passwd

Type: string

The previously established password of the eFiling User

packageGUID

Type: string

The GSCCCA assigned package identifier for the package being rejected

rejectionReasons

Type: string[]

A list of reasons the package could not be recorded. At least one reason must be Specified

Return Value

Type: string

A XML string that represents a well-formed PRIA response envelope conveying the status of the rejection request.

Remarks

- 1) The filer is identified by the user id and password provided with the method invocation. The package can only be rejected by an account associated with the county in which the package was filed.
- 2) The Package GUID is the GSCCCA system generated identifier that was returned in the **PACKAGE_Identifier** attribute of the original submission request
- 3) The Package GUID must be well formed. Supplying a nonconforming GUID will result in an error message being returned.
- 4) If the package identified by the filing GUID cannot be found, an error message is generated.
- 5) The requested package must be in the "Pending" state. If the requested package is not in the "Pending" state, an error message will be generated.
- 6) At least one reason must be provided as to why the package could not be recorded.

RetrieveRecordedPackage METHOD

Syntax

```
public string RetrieveRecordedPackage(string userid, string password, string packageGUID)
```

Parameters

userid

Type: string

The previously established userID of the eFiling User

passwd

Type: string

The previously established password of the eFiling User

packageGUID

Type: string

The GSCCCA assigned package identifier for the recorded document being requested

Return Value

Type: string

A XML string that represents a well-formed PRIA response envelope conveying the recorded package if the filing exists and has in fact been recorded.

Remarks

- 1) The filer is identified by the user id and password provided with the method invocation. The package can only be obtained by the filer who originally submitted it or an account associated with the county in which the package was filed.
- 2) The package GUID is the GSCCCA system generated identifier that was returned in the **PACKAGE_Identifier** attribute of the original submission request
- 3) The package GUID must be well formed. Supplying a nonconforming GUID will result in an error message being returned.
- 4) If the package identified by the package GUID cannot be found, an error message is generated.
- 5) The requested package must be in the "Recorded" state. If the requested filing is not in the "Recorded" state, an error message will be generated.
- 6) The response is guaranteed to contain a single **PACKAGE** element with one or more **PRIA_DOCUMENT** element. Each **PRIA_DOCUMENT** element will always contain one or more **EMBEDDED_FILE** elements. The embedded file will convey the filed document with any recording information that was stamped at the time of acceptance.
- 7) When accepting a document, the county must provide a completed **RECORDING_ENDORSEMENT** element for each **PRIA_DOCUMENT** accepted. As such, the filer can be guaranteed that this element will be included in the response. At a minimum, the county must provide the following information within the **ENDORSEMENT** element:

- a. Attribute: **_Volume**
 - b. Attribute: **_VolumeType**
 - c. Attribute: **_RecordedDateTime**
 - d. Element: **_VOLUME_PAGE**
- 8) When accepting a document, the county must provide fee information in the **_FEES** element of the **RECORDING_ENDORSEMENT**. As such, the filer can be guaranteed that the **_FEES** element will be included in the response. At a minimum, the **_TotalAmount** attribute of the **_FEES** element must be provided by the county. Fee detail, if any, can be found in one or more **_RECORDING_FEE** elements within the **_FEES element**.

RetrieveRejectedPackage METHOD

Syntax

```
public string RetrieveRejectedPackage(string userid, string password, string packageGUID)
```

Parameters

userID

Type: string

The previously established userID of the eFiling User

passwd

Type: string

The previously established password of the eFiling User

pacakgeGUID

Type: string

The GSCCCA assigned package identifier for the rejected package being requested

Return Value

Type: string

A XML string that represents a well-formed PRIA response envelope conveying the rejected filing status and rejection reasons if the package exists and has in fact been rejected.

Remarks

- 1) The filer is identified by the user id and password provided with the method invocation. The package can only be obtained by the filer who originally submitted it, or an account associated with the county in which the package was filed.
- 2) The Package GUID is the GSCCCA system generated identifier that was returned in the **PACKAGE_Identifier** attribute of the original submission request
- 3) The package GUID must be well formed. Supplying a nonconforming GUID will result in an error message being returned.
- 4) If the package identified by the package cannot be found, an error message is generated.
- 5) The requested package must be in the "Rejected" state. If the requested package is not in the "Rejected" state, an error message will be generated.
- 6) The response generated will contain an empty **PACKAGE** element with only the **PACKAGE_Identifier** attribute set.
- 7) When rejecting a package, the county is required to provide at least one reason for the rejection. As such, the response is guaranteed to contain at least one **RECORDING_ERROR** element within then **STATUS** element of the response.

ValidatePT61Status METHOD

Syntax

```
public string ValidatePT61Status(string pt61Number)
```

Parameters

pt61Number
Type: string
The PT-61 number being queried

Return Value

Type: string
A XML string that represents an XML response envelope conveying the status of the request and information about the PT-61 document being validated.

Remarks

- 1) This web method is used to determine the validity of the PT-61 document passed in. The response XML consists of a **GSCCCA** element with an enclosing **CallStatus** element and optional Valid and NewPT61Number elements.
- 2) The **GSCCCA/CallStatus** element indicates whether the call itself had an error. If there are no errors, then this element will be "Success" and otherwise it will be "Error" to indicate that something went wrong with the request. For any error response, please try again.
- 3) If the call is a success, then the **GSCCCA/Valid** element will be present. If the PT-61 Number given is a valid number, then it will be "true." Otherwise, it will be "false."
- 4) In the case where the PT-61 number is not valid due to a change to the PT-61 since the original number was assigned, the PT-61 will have a new number. The new number is passed back in the **GSCCCA/NewPT61Number** element and is the current, valid number for the given PT-61.

Exception Handling and Service Interruptions

Care has been given in the implementation of all web methods to handle most predictable errors and system exceptions. Generally speaking, business rule violations are intercepted and reported back as errors within the response **STATUS** element. System errors on the GSCCCA server, such as temporary service interruption, are expected to be very rare. However, system errors due tend to occur from time to time and where such errors can be anticipated, the web methods will intercept such exceptions and report the issue in the **STATUS** element of the response. If the system error prevented the successful execution of an action, the status message may indicate that a call to GSCCCA technical support will be necessary. ALL ERRORS that can be anticipated are logged in the GSCCCA database when they occur and GSCCCA technical support staff is alerted.

It is unlikely that an exception will be raised from the web method to the consuming application unless there is a basic access with the method call itself. However, each method does not include a catch-all error handler and it is possible, though unlikely, that an unhandled exception will be raised from the web method itself. It is recommended that your application consider this in its design, and that the GSCCCA technical support department is notified whenever such errors occur.

Generally, if a system error occurs that prevents an action from being performed, it is safe to simply retry the transaction. If the system error occurred during the submission of a new filing, the internal duplicate prevention should prevent the same filing from being submitted twice. Therefore, it is safe to resubmit a document in the unlikely event that a submit() call fails.

Minimum Data Requirements

For its initial implementation, the GSCCCA has adopted a LEVEL II approach to real estate eFiling. Essentially, the document is presented as a digital representation and can stand alone without other meta data. The GSCCCA does require a minimum set of meta data to be included as previously discussed, such as the receiving party identifier and the document type.

Certain data fields, like **GRANTOR** and **GRANTEE** are REQUIRED by the PRIA 2.4.1 standard as part of the **PRIA_DOCUMENT** element when it is included in a **PACKAGE**. This requirement is enforced by GSCCCA as part of schema validation, though it is not anticipated that these data fields will be used DIRECTLY by the county clerk's office. Georgia has defined a statewide indexing standard that must be used when indexing such information, and such indexing must be done by individuals well versed in these standards. It is unlikely that data submitted through the GSCCCA portal will be compliant with Georgia Indexing standards.

Any data field required by the PRIA 2.4.1 standard in order to pass schema validation must be included in the request. However, since most of these fields are unused, it is permissible to either include actual data or to leave the data empty (""). Though not required, it may be good practice to include as much information as is available in the PRIA request to help the clerk verify and validate the document to be recorded.