

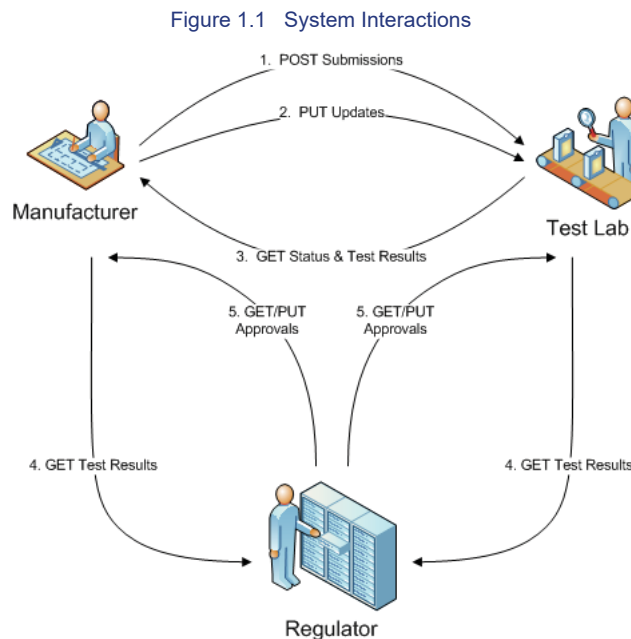
# Chapter 1

# Introduction

## 1.1 Overview

This document describes the Certification Database Interface (CDI). The Certification Database Interface can be utilized to access the certification databases of testing laboratories, vendors, and regulatory agencies. It is intended to be used by testing laboratories, vendors, and regulators to exchange information related to the testing and approval of gaming products including EGM hardware, EGM software, and iGaming software.

With the Certification Database Interface, a Client System uses the HTTP protocol and HTTP verbs — such as GET, POST, PUT, and DELETE — to access resources on a Host System. The data exchanged between systems is encoded using JSON. The resources available through the Certification Database Interface are described in subsequent chapters of this specification. The following diagram illustrates the intended interactions.



The process begins when a vendor submits a product to a test laboratory. The information about the product and the certification request is conveyed to the test laboratory using an HTTP POST operation. Subsequently, additional certification requests can be added to the submission using an HTTP PUT operation. The vendor can request status updates about the submission from the test laboratory using an HTTP GET operation.

Once a submission has been accepted by the test laboratory, the vendor can request information about the testing and certification of the product using an HTTP GET operation. The vendor can request information about a specific product or the vendor can request information about all products that have had status changes during a specific time period.

Similarly, once a submission has been accepted by the test laboratory, the regulator can also request information about the testing and certification of the product using an HTTP GET operation. Like the vendor, the regulator can request information about a specific product or the regulator can request information about all products that have had status changes during a specific time period.

After a product has been approved by the regulator, the vendor or test laboratory can request information about the approval from the regulator using an HTTP GET operation. The approval information can be requested for a specific product or for all products with status changes during a specific time period.

Additionally, the regulator can send updates to the approval information to the vendor or test laboratory using an HTTP PUT operation.

# Chapter 3

# Submission Resources

## 3.1 submissions Resource

The `submissions` resource can be used by a Client System to submit certification requests to a Host System, to add new jurisdictions to a previous certification request, and to retrieve status information about the submission. It is intended that vendors use this resource as Client Systems to submit certification requests to testing laboratories acting as Host Systems. After a submission has been accepted by a test laboratory, vendors should use the `certifications` resource to request status information about the testing and approval of the submitted product.

Table 3.1 submissions HTTP Verbs

Resource	HTTP Verbs			
	POST (create)	GET (read)	PUT (update)	DELETE (delete)
/cdi/[ver]/submissions	Yes	Yes	Yes	No

### 3.1.1 POST submissions Resource

The following table contains information about the `submissions` resource when the HTTP POST verb is used. It includes the pathname and content type used to access the resource. When accessing the `submissions` resource using the HTTP POST verb, the specified values **MUST** be used.

This resource is used to submit new certification requests to the Host System. The requests **MAY** not be accepted immediately. A manual review may be required before the requests are entered into the Host System.

Table 3.2 POST submissions Resource Information

HTTP Method	POST
Pathname	/cdi/[ver]/submissions
Request Content-Type	application/json; charset=utf-8
Request Content	submissions Object. See <a href="#">Section 3.1.1.1, submissions Object</a> for details.
Response Content-Type	application/json; charset=utf-8
Response Content	None.

#### 3.1.1.1 submissions Object

The following table identifies the properties of the `submissions` object. Additional properties **MAY** be included in the `submissions` object.

Table 3.3 submissions Properties

Property	Restrictions	Description
submissionArray	type: certificationSub use: required minItems: 1 maxItems: ∞	Array of certificationSub Objects. See <a href="#">Section 3.1.1.2, certificationSub Object</a> for details.

### 3.1.1.2 certificationSub Object

The following table identifies the properties of the `certificationSub` object. Additional properties MAY be included in the `certificationSub` object.

The `certificationId` MUST be a UUID generated in a manner compliant with the ISO/IEC 9834-8:2014 standard to guarantee uniqueness. It serves as the unique identifier for the submission record. Two submission records with the same `certificationId` are considered duplicates.

Table 3.4 certificationSub Properties

Property	Restrictions	Description
<code>certificationId</code>	type: <code>t_certificationId</code> use: required	UUID for the product certification request record.  <b>Example: 96CA8F38-A692-4748-847A-A1DA00D2B95C</b>
<code>certificationNumber</code>	type: <code>t_name</code> use: required	The human-readable identification number for the product certification request.  <b>Example: ABV_123456</b>
<code>vendorId</code>	type: <code>t_vendorId</code> use: required	UUID representing the vendor.  <b>Example: 456A3411-78FA-3234-B341-0078CA123489</b>
<code>testLabId</code>	type: <code>t_testLabId</code> use: required	UUID representing the test laboratory.  <b>Example: 1230987D-8976-4321-CC11-098123457634</b>
<code>certificationCode1</code>	type: <code>t_name</code> use: optional default: <empty>	Vendor-specific code for tracking the certification request.  <b>Example: My Code 1</b>
<code>certificationCode2</code>	type: <code>t_name</code> use: optional default: <empty>	Vendor-specific code for tracking the certification request.  <b>Example: My Code 2</b>
<code>noteArray</code>	type: <code>note</code> use: optional minItems: 1 maxItems: ∞	Array of <code>note</code> objects about the certification request. See <a href="#">Section 3.1.1.6, note Object</a> for details.
<code>componentSubArray</code>	type: <code>componentSub</code> use: required minItems: 1 maxItems: ∞	Array of <code>componentSub</code> objects. See <a href="#">Section 3.1.1.3, componentSub Object</a> for details.

### 3.1.1.3 componentSub Object

The following table identifies the properties of the `componentSub` object. Additional properties MAY be included in the `componentSub` object.

The `componentId` MUST be a UUID generated in a manner compliant with the ISO/IEC 9834-8:2014 standard to guarantee uniqueness. It serves as the unique identifier for the `componentSub` record. Two `componentSub` records with the same `componentId` are considered duplicates.

Table 3.5 componentSub Properties

Property	Restrictions	Description
componentId	type: <code>t_componentId</code> use: required	UUID for the product component record. <b>Example: 96CA8F38-A692-4748-847A-A1DA00D2B95C</b>
componentNumber	type: <code>t_name</code> use: required	The human-readable identification number for the product component. <b>Example: ABV_123456</b>
version	type: <code>t_version</code> Use: optional default: <code>&lt;empty&gt;</code>	Identifies the version of the component. <b>Example: 1.2.3</b>
function	type: <code>t_name</code> use: optional default: <code>&lt;empty&gt;</code>	Identifies the function of the component. <b>Example: Game Software</b>
mediaType	type: <code>t_name</code> use: optional default: <code>&lt;empty&gt;</code>	Identifies the type of media on which the component resides. <b>Example: Download</b>
mediaSize	type: <code>numeric</code> use: optional multipleOf: 1 default: 0	Identifies the size of media on which the component resides. <b>Example: 654321</b>
position	type: <code>t_name</code> use: optional default: <code>&lt;empty&gt;</code>	Identifies the position in which the component is installed; for example, the socket on an EGMs motherboard. <b>Example: Socket U2</b>
componentName	type: <code>t_name</code> use: optional default: <code>&lt;empty&gt;</code>	The human-readable name of the component, such as the name of a game. <b>Example: A Better Game</b>
componentCode1	type: <code>t_name</code> use: optional default: <code>&lt;empty&gt;</code>	Vendor-specific code for tracking the component. <b>Example: My Code 1</b>
componentCode2	type: <code>t_name</code> use: optional default: <code>&lt;empty&gt;</code>	Vendor-specific code for tracking the component. <b>Example: My Code 2</b>
noteArray	type: <code>note</code> use: optional minItems: 1 maxItems: $\infty$	Array of <code>note</code> objects about the component See <a href="#">Section 3.1.1.6, note Object</a> for details.
paytableArray	type: <code>paytable</code> use: optional minItems: 1 maxItems: $\infty$	Array of <code>paytable</code> objects associated with the component. See <a href="#">Section 3.1.1.7, paytable Object</a> for details.

Table 3.5 componentSub Properties

Property	Restrictions	Description
jurisdictionSubArray	type: jurisdictionSub use: required minItems: 1 maxItems: ∞	Array of jurisdictionSub objects. See Section 3.1.1.4, jurisdictionSub Object for details.
signatureArray	type: signature use: required minItems: 0 maxItems: ∞	Array of signature objects. See Section 3.1.1.5, signature Object for details.

### 3.1.1.4 jurisdictionSub Object

The following table identifies the properties of the jurisdictionSub object. Additional properties MAY be included in the jurisdictionSub object.

The submissionId MUST be UUID generated in a manner compliant with the ISO/IEC 9834-8:2014 standard to guarantee uniqueness. It serves as the unique identifier for the jurisdictionSub record. Two jurisdictionSub records with the same submissionId are considered duplicates.

Table 3.6 jurisdictionSub Properties

Property	Restrictions	Description
submissionId	type: t_submissionId use: required	UUID for the jurisdiction submission record. <b>Example: 96CA8F38-A692-4748-847AA1DA00D2B95C</b>
jurisdictionId	type: t_jurisdictionId use: required	UUID representing a specific jurisdiction. <b>Example: CFFFC5A7-38BE-4351-9A7C-D8A27D7C0BF2</b>
rush	type: boolean use: optional default: false	Indicates whether the submitter considers the submission a rush. <b>Example: true</b>
requestedDate	type: string format: date use: optional default: <empty>	Requested date for when the certification will be issued. <b>Example: 2015-12-31</b>
jurisdictionCode1	type: t_name use: optional default: <empty>	Vendor-specific code for tracking the jurisdictional testing and approval. <b>Example: My Code 1</b>
jurisdictionCode2	type: t_name use: optional default: <empty>	Vendor-specific code for tracking the jurisdictional testing and approval. <b>Example: My Code 2</b>
noteArray	type: note use: optional minItems: 1 maxItems: ∞	Array of note objects for the jurisdiction. See Section 3.1.1.6, note Object for details.



Table 3.6 jurisdictionSub Properties

Property	Restrictions	Description
documentArray	type: document use: optional minItems: 1 maxItems: ∞	Array of document objects associated with the submission for the jurisdiction. See <a href="#">Section 3.1.1.8, document Object</a> for details.

### 3.1.1.5 signature Object

The following table identifies the properties of the `signature` object. Additional properties MAY be included in the `signature` object.

The `signatureId` MUST be a UUID generated in a manner compliant with the ISO/IEC 9834-8: 2014 standard to guarantee uniqueness. It serves as the unique identifier for the `signature` record. Two `signature` records with the same `signatureId` are considered duplicates.

Table 3.7 signature Properties

Property	Restrictions	Description
signatureId	type: <code>t_signatureId</code> use: required	UUID for the signature record. <b>Example: 96CA8F38-A692-4748-847AA1DA00D2B95C</b>
algorithmId	type: <code>t_algorithmId</code> use: required	UUID representing the authentication algorithm. <b>Example: 754580E4-2B24-4DF9-A583-469688405F39</b>
seed	type: <code>t_seed</code> use: optional default: false	The seed for the algorithm. Certain algorithms, such as checksums (CRC), require a seed to define the starting value. <b>Example: 12345678</b>
salt	type: <code>t_salt</code> use: optional default: false	Arbitrary bytes that are prefixed to the component's byte buffer before the hash is generated (and after the offsets are applied). <b>Example: FEDCBA9876543210</b>
startOffset	type: <code>numeric</code> use: optional multipleOf: 1 default: 0 minIncl: 0	The starting offset for the verification component. <b>Example: 4321</b>
endOffset	type: <code>numeric</code> use: optional multipleOf: 1 default: -1 minIncl: -1	The ending offset for the verification component. <b>Example: -1</b>
verifyResult	type: <code>t_verifyResult</code> use: required	The software signature calculated by the Client System. <b>Example: 0F1E2D3C4B5A69788796A5B4C3D2E1F0</b>

Table 3.7 signature Properties

Property	Restrictions	Description
signatureSource	type: t_UUID use: required	Used to identify the source of the signature — that is, the vendor, test laboratory, or jurisdiction that originated the note.  <b>Example: 456A3411-78FA-3234-B3410078CA123489</b>

### 3.1.1.6 note Object

The following table identifies the properties of the `note` object. Additional properties MAY be included in the `note` object.

The `noteType` property should identify the type, category, or subject of the note. The `noteText` property should contain the text for the note.

The `noteId` MUST be a UUID generated in a manner compliant with the ISO/IEC 9834-8:2014 standard to guarantee uniqueness. It serves as the unique identifier for the `note` record. Two `note` records with the same `noteId` are considered duplicates.

Table 3.8 note Properties

Property	Restrictions	Description
noteId	type: t_noteId use: required	UUID for the note record.  <b>Example: 754580E4-2B24-4DF9-A583-469688405F39</b>
noteType	type: t_name use: required	Identifies the type or category of note – for example, Modification, Special Request, etc.  <b>Example: Special Request</b>
noteText	type: t_notes use: required	Contains the full text of the note.  <b>Example: Please handle this request with high priority.</b>
noteSource	type: t_UUID use: required	Used to identify the source of the note — that is, the vendor, test laboratory, or jurisdiction that originated the note.  <b>Example: 456A3411-78FA-3234-B3410078CA123489</b>

### 3.1.1.7 payable Object

The following table identifies the properties of the `paytable` object. Additional properties MAY be included in the `paytable` object.

The `paytableId` MUST be a UUID generated in a manner compliant with the ISO/IEC 9834-8:2014 standard to guarantee uniqueness. It serves as the unique identifier for the `paytable` record. Two `paytable` records with the same `paytableId` are considered duplicates.

Table 3.9 payable Properties

Property	Restrictions	Description
paytableId	type: <code>t_paytableId</code> use: required	UUID for the payable record. <b>Example: 754580E4-2B24-4DF9-A583-469688405F39</b>
paytableName	type: <code>t_name</code> use: required	The human-readable identifier for the payable. <b>Example: ABV_123456_9600</b>
paytableMax	type: <code>t_percent</code> use: required	Maximum payback percentage for the payable; expressed in hundredths of a percent; for example, 97.35% is expressed as 9735. <b>Example: 9735</b>
paytableMin	type: <code>t_percent</code> use: required	Minimum payback percentage for the payable; expressed in hundredths of a percent; for example, 96.65% is expressed as 9665. <b>Example: 9665</b>
paytableVI	type: <code>t_paytableVI</code> use: optional default: <empty>	Volatility index for the payable. <b>Example: 16.552</b>
paytableCI	type: <code>t_paytableCI</code> use: optional default: <empty>	Confidence interval for the payable. <b>Example: 90</b>

### 3.1.1.8 document Object

The following table identifies the properties of the document object. Additional properties MAY be included in the document object.

The `documentId` MUST be a UUID generated in a manner compliant with the ISO/IEC 9834-8:2014 standard to guarantee uniqueness. It serves as the unique identifier for the document record. Two document records with the same `documentId` are considered duplicates.

Table 3.10 document Properties

Property	Restrictions	Description
documentId	type: <code>t_documentId</code> use: required	UUID for the document record. <b>Example: 754580E4-2B24-4DF9-A583-469688405F39</b>
documentType	type: <code>t_name</code> use: required	Identifies the type or category of the document - for example, Product Documentation, Par Sheet, Certification Report, Approval Letter, etc. <b>Example: Product Documentation</b>
documentName	type: <code>t_name</code> use: required	The human-readable identifier for the document. <b>Example: ABV_Package_123456</b>

Table 3.10 document Properties

Property	Restrictions	Description
documentSource	type: <code>t_UUID</code> use: required	Used to identify the source of the document - that is, the vendor, test laboratory, or jurisdiction that originated the document.  <b>Example: 456A3411-78FA-3234-B3410078CA123489</b>
documentFormat	type: <code>t_code</code> use: required	Document format; typically, the file extension for the document; for example, pdf, doc, zip, etc.  <b>Example: pdf</b>
documentLanguage	type: <code>t_language</code> use: optional default: en	Language of the document.  <b>Example: en</b>
documentPath	type: <code>string</code> format: uri use: required	Network URI whence the document can be retrieved.  <b>Example: https://www.abv.com/subs/ABV_Package_123456.pdf</b>

### 3.1.1.9 POST submissions Example

The following example demonstrates the construction of a POST submissions request and a response indicating that the submission was accepted. In practice, additional HTTP headers may be included in the messages.

Request:

```
POST /cdi/1.0/submissions HTTP/1.1
Content-Length: 2271
Content-Type: application/json; charset=utf-8
```

```
{
  "submissionArray": [
    {
      "certificationId": "96CA8F38-A692-4748-847A-A1DA00D2B95C",
      "certificationNumber": "ABV_Cert_123",
      "vendorId": "456A3411-78FA-3234-B341-0078CA123489",
      "testLabId": "1230987D-8976-4321-CC11-098123457634",
      "certificationCode1": "My Certification Code 1",
      "certificationCode2": "My Certification Code 2",
      "noteArray": [
        {
          "noteId": "8F3896CA-A692-4748-847A-A1DA00D2B95C",
          "noteType": "Special Request",
          "noteText": "This submission is a rush.",
          "noteSource": "456A3411-78FA-3234-B341-0078CA123489"
        }
      ],
      "componentSubArray": [
        {
          "componentId": "BA8B450F-8C2F-43D3-8FEF-04B59366543D",
          "componentNumber": "ABV_Comp_1234",
          "version": "1.2a",
          "function": "Game Software",
          "mediaType": "Download",
          "mediaSize": 654321,
        }
      ]
    }
  ]
}
```

```

"position": "Socket U2",
"componentName": "Triple 7s",
"componentCode1": "My Component Code 1",
"componentCode2": "My Component Code 2",
"noteArray": [
  {
    "noteId": "8F3896CA-A692-4748-847A-A1DA00D2B95C",
    "noteType": "Note",
    "noteText": "This game requires skill to achieve max payback.",
    "noteSource": "456A3411-78FA-3234-B341-0078CA123489"
  }
],
"paytableArray": [
  {
    "paytableId": "3896CA8F-A692-4748-847A-A1DA00D2B95C",
    "paytableName": "ABV_123456_9600",
    "paytableMax": "9600",
    "paytableMin": "9450",
    "paytableVI": "16.552",
    "paytableCI": "90"
  }
],
"jurisdictionSubArray": [
  {
    "submissionId": "7FDE945E-24FE-414C-BF6E-E0F3A1DA00E3",
    "jursidictionId": "E9457FDE-24FE-414C-BF6E-A1DA00E0F3E3",
    "rush": "true",
    "requestedDate": "2015-12-31",
    "noteArray": [
      {
        "noteId": "968F38CA-A692-4748-847A-A1DA00D2B95C",
        "noteType": "Special Request",
        "noteText": "Field trial requested.",
        "noteSource": "456A3411-78FA-3234-B341-0078CA123489"
      }
    ]
  }
],
"documentArray": [
  {
    "documentId": "8F3896CA-A692-4748-847A-A1DA00D2B95C",
    "documentType": "Submission Package",
    "documentName": "ABV_Package_123456",
    "documentSource": "456A3411-78FA-3234-B341-0078CA123489",
    "documentFormat": "zip",
    "documentLanguage": "en",
    "documentPath": "https://www.abv.com/packages/ABV_Package_123456"
  }
]
},
"signatureArray": [
  {
    "signatureId": "80E47545-2B24-4DF9-A583-688404695F39",
    "algorithmId": "754580E4-2B24-4DF9-A583-469688405F39",
    "verifyResult": "1234567890123456789012345678901234567890"
  },
  {
    "signatureId": "E4758045-2B24-4DF9-A583-884046695F39",
    "algorithmId": "0BEEB53E-45AE-4F53-9ED5-456E1CE8FFB8",
    "seed": "0123",
    "verifyResult": "1234"
  }
]

```

```

    ]
  }
]
}
]
}

```

Response:

```

HTTP/1.1 200 OK
Content-Length: 0

```

### 3.1.2 GET submissions Resource

The following table contains information about the `submissions` resource when the HTTP GET verb is used. It includes the pathname and content type used to access the resource. When accessing the `submissions` resource using the HTTP GET verb, the specified values **MUST** be used.

This resource can be used to retrieve status information about previously submitted certification requests from the Host System.

Table 3.11 GET submissions Resource Information

<b>HTTP Method</b>	GET
<b>Pathname</b>	/cdi/[ver]/submissions
<b>Request Content-Type</b>	application/json; charset=utf-8
<b>Request Content</b>	None.
<b>Response Content-Type</b>	application/json; charset=utf-8
<b>Response Content</b>	submissionStatuses Object. See <a href="#">Section 3.1.2.2, submissionStatuses Object</a> for details.

#### 3.1.2.1 GET submissions Parameters

The following table identifies the parameters of the `submissions` resource when the HTTP GET verb is used. The parameters are appended to the resource URI in the query component of the HTTP request.

- If the `certificationId` parameter is included, only status information regarding certification requests with the specified `certificationId` are included in the response; otherwise, the `certificationId` parameter is ignored.
- If the `submissionStatus` parameter is included, only status information regarding certification requests with the specified `submissionStatus` are included in the response; otherwise, the `submissionStatus` parameter is ignored.
- If the `submissionStart` parameter is included, only information regarding certification requests with `submissionDateTime` values greater than or equal to the specified `submissionStart` value are included in the response; otherwise, the `submissionStart` parameter is ignored.
- If the `jurisdictionId` parameter is included, only status information regarding certification requests with the specified `jurisdictionId` are included in the response; otherwise, the `jurisdictionId` parameter is ignored.

- If no parameters are included, information regarding all certification requests to which the Client System has access are included in the response.

If the included parameters result in no certification requests being selected, the Host System MUST simply return an empty list of certification requests to the Client System.

Table 3.12 GET submissions Parameters

Parameter	Restrictions	Description
certificationId	type: <code>t_certificationId</code> use: optional	UUID representing a particular certification request record.  <b>Example: 96CA8F38-A692-4748-847A-A1DA00D2B95C</b>
jurisdictionId	type: <code>t_jurisdictionId</code> use: optional	UUID representing a specific jurisdiction.  <b>Example: CFFFC5A7-38BE-4351-9A7C-D8A27D7C0BF2</b>
submissionStatus	type: <code>t_subStatuses</code> use: optional	The status of a submission for a jurisdiction.  <b>Example: Approved</b>
submissionStart	type: <code>string</code> format: <code>dateTime</code> use: optional	The earliest date/time for which changes to the submission status are requested.  <b>Example: 2010-06-03T00:00:00</b>

### 3.1.2.2 submissionStatuses Object

The following table identifies the properties of the `submissionStatuses` object. Additional properties MAY be included in the `submissionStatuses` object.

Table 3.13 submissionStatuses Properties

Property	Restrictions	Description
<code>submissionStatusArray</code>	type: <code>submissionStatus</code> use: required minItems: 0 maxItems: $\infty$	Array of <code>submissionStatus</code> Objects. See <a href="#">Section 3.1.2.3, submissionStatus Object</a> for details.

### 3.1.2.3 submissionStatus Object

The following table identifies the properties of the `submissionStatus` object. Additional properties MAY be included in the `submissionStatus` object.

Table 3.14 submissionStatus Properties

Property	Restrictions	Description
certificationId	type: <code>t_certificationId</code> use: required	UUID for the product certification request record.  <b>Example: 96CA8F38-A692-4748-847A-A1DA00D2B95C</b>

Table 3.14 submissionStatus Properties

Property	Restrictions	Description
certificationNumber	type: <code>t_name</code> use: required	The human-readable identification number for the product certification request. <b>Example: ABV_123456</b>
vendorId	type: <code>t_vendorId</code> use: required	UUID representing the vendor. <b>Example: 456A3411-78FA-3234-B341-0078CA123489</b>
testLabId	type: <code>t_testLabId</code> use: required	UUID representing the test laboratory. <b>Example: 1230987D-8976-4321-CC11-098123457634</b>
certificationCode1	type: <code>t_name</code> use: optional default: <empty>	Vendor-specific code for tracking the certification request. <b>Example: My Code 1</b>
certificationCode2	type: <code>t_name</code> use: optional default: <empty>	Vendor-specific code for tracking the certification request. <b>Example: My Code 2</b>
noteArray	type: <code>note</code> use: optional minItems: 1 maxItems: $\infty$	Array of <code>note</code> objects about the certification request. See <a href="#">Section 3.1.1.6, note Object</a> for details.
componentStatusArray	type: <code>componentStatus</code> use: required minItems: 1 maxItems: $\infty$	Array of <code>componentStatus</code> objects. See <a href="#">Section 3.1.2.4, componentStatus Object</a> for details.

### 3.1.2.4 componentStatus Object

The following table identifies the properties of the `componentStatus` object. Additional properties MAY be included in the `componentStatus` object.

Table 3.15 componentStatus Properties

Property	Restrictions	Description
componentId	type: <code>t_componentId</code> use: required	UUID for the product component record. <b>Example: 96CA8F38-A692-4748-847A-A1DA00D2B95C</b>
componentNumber	type: <code>t_name</code> use: required	The human-readable identification number for the product component. <b>Example: ABV_123456</b>
version	type: <code>t_version</code> Use: optional default: <empty>	Identifies the version of the component. <b>Example: 1.2.3</b>



Table 3.15 componentStatus Properties

Property	Restrictions	Description
function	type: <a href="#">t_name</a> use: optional default: <empty>	Identifies the function of the component. <b>Example: Game Software</b>
mediaType	type: <a href="#">t_name</a> use: optional default: <empty>	Identifies the type of media on which the component resides. <b>Example: Download</b>
mediaSize	type: <a href="#">numeric</a> use: optional multipleOf: 1 default: 0	Identifies the size of media on which the component resides. <b>Example: 654321</b>
position	type: <a href="#">t_name</a> use: optional default: <empty>	Identifies the position in which the component is installed; for example, the socket on an EGMs motherboard. <b>Example: Socket U2</b>
componentName	type: <a href="#">t_name</a> use: optional default: <empty>	The human-readable name of the component, such as the name of a game. <b>Example: A Better Game</b>
componentCode1	type: <a href="#">t_name</a> use: optional default: <empty>	Vendor-specific code for tracking the component. <b>Example: My Code 1</b>
componentCode2	type: <a href="#">t_name</a> use: optional default: <empty>	Vendor-specific code for tracking the component. <b>Example: My Code 2</b>
noteArray	type: <a href="#">note</a> use: optional minItems: 1 maxItems: ∞	Array of <a href="#">note</a> objects about the certification request. See <a href="#">Section 3.1.1.6, note Object</a> for details.
paytableArray	type: <a href="#">paytable</a> use: optional minItems: 1 maxItems: ∞	Array of <a href="#">paytable</a> objects associated with the component. See <a href="#">Section 3.1.1.7, payable Object</a> for details.
jurisdictionStatusArray	type: <a href="#">jurisdictionStatus</a> use: required minItems: 1 maxItems: ∞	Array of <a href="#">jurisdictionStatus</a> objects. See <a href="#">Section 3.1.2.5, jurisdictionStatus Object</a> for details.
signatureArray	type: <a href="#">signature</a> use: required minItems: 0 maxItems: ∞	Array of <a href="#">signature</a> objects. See <a href="#">Section 3.1.1.5, signature Object</a> for details.

### 3.1.2.5 jurisdictionStatus Object

The following table identifies the properties of the `jurisdictionStatus` object. Additional properties MAY be included in the `jurisdictionStatus` object.

Table 3.16 jurisdictionStatus Properties

Property	Restrictions	Description
<code>submissionId</code>	type: <code>t_submissionId</code> use: required	UUID for the jurisdiction submission record. <b>Example: 96CA8F38-A692-4748-847AA1DA00D2B95C</b>
<code>jurisdictionId</code>	type: <code>t_jurisdictionId</code> use: required	UUID representing a specific jurisdiction. <b>Example: CFFFC5A7-38BE-4351-9A7C-D8A27D7C0BF2</b>
<code>rush</code>	type: <code>boolean</code> use: optional default: <code>false</code>	Indicates whether the submitter considers the submission a rush. <b>Example: true</b>
<code>requestedDate</code>	type: <code>string</code> format: <code>date</code> use: optional default: <code>&lt;empty&gt;</code>	Requested date for when the certification will be issued. <b>Example: 2015-12-31</b>
<code>jurisdictionCode1</code>	type: <code>t_name</code> use: optional default: <code>&lt;empty&gt;</code>	Vendor-specific code for tracking the jurisdictional testing and approval. <b>Example: My Code 1</b>
<code>jurisdictionCode2</code>	type: <code>t_name</code> use: optional default: <code>&lt;empty&gt;</code>	Vendor-specific code for tracking the jurisdictional testing and approval. <b>Example: My Code 2</b>
<code>submissionDateTime</code>	type: <code>string</code> format: <code>dateTime</code> use: required	Date/time that the <code>submissionStatus</code> property was last updated. <b>Example: 2015-12-10T14:42:00</b>
<code>submissionStatus</code>	type: <code>t_subStatuses</code> use: required	Status of the certification request for the jurisdiction. <b>Example: Approved</b>
<code>vendorSubmitted</code>	type: <code>string</code> format: <code>date</code> use: required	Date that the certification request for the jurisdiction was received by the test laboratory. <b>Example: 2015-12-10</b>
<code>noteArray</code>	type: <code>note</code> use: optional minItems: 1 maxItems: $\infty$	Array of <code>note</code> objects about the certification request. See <a href="#">Section 3.1.1.6, note Object</a> for details.

Table 3.16 jurisdictionStatus Properties

Property	Restrictions	Description
documentArray	type: document use: optional minItems: 1 maxItems: ∞	Array of document objects associated with the submission for the jurisdiction. See <a href="#">Section 3.1.1.8, document Object</a> for details.

### 3.1.2.6 GET submissions Example

The following example demonstrates the construction of a GET submissions request and a response containing a submissionStatus object. In practice, additional HTTP headers may be included in the messages.

Request:

```
GET /cdi/1.0/submissions?certificationId=96CA8F38-A692-4748-847A-A1DA00D2B95C HTTP/1.1
Accept: application/json
Accept-Charset: utf-8
```

Response:

```
HTTP/1.1 200 OK
Content-Length: 2390
Content-Type: application/json; charset=utf-8
```

```
{
  "submissionStatusArray": [
    {
      "certificationId": "96CA8F38-A692-4748-847A-A1DA00D2B95C",
      "certificationNumber": "ABV_Cert_123",
      "vendorId": "456A3411-78FA-3234-B341-0078CA123489",
      "testLabId": "1230987D-8976-4321-CC11-098123457634",
      "certificationCode1": "My Certification Code 1",
      "certificationCode2": "My Certification Code 2",
      "noteArray": [
        {
          "noteId": "8F3896CA-A692-4748-847A-A1DA00D2B95C",
          "noteType": "Special Request",
          "noteText": "This submission is a rush.",
          "noteSource": "456A3411-78FA-3234-B341-0078CA123489"
        }
      ],
      "componentStatusArray": [
        {
          "componentId": "BA8B450F-8C2F-43D3-8FEF-04B59366543D",
          "componentNumber": "ABV_Comp_1234",
          "version": "1.2a",
          "function": "Game Software",
          "mediaType": "Download",
          "mediaSize": 654321,
          "position": "Socket U2",
          "componentName": "Triple 7s",
          "componentCode1": "My Component Code 1",
          "componentCode2": "My Component Code 2",
          "noteArray": [
            {
              "noteId": "8F3896CA-A692-4748-847A-A1DA00D2B95C",
              "noteType": "Note",
              "noteText": "This game requires skill to achieve max payback."
            }
          ]
        }
      ]
    }
  ]
}
```



### 3.1.3 PUT submissions Resource

The following table contains information about the `submissions` resource when the HTTP PUT verb is used. It includes the pathname and content type used to access the resource. When accessing the `submissions` resource using the HTTP PUT verb, the specified values **MUST** be used.

This resource is used to add new certification requests for specific jurisdictions onto a previous submission. The requests **MAY** not be accepted immediately. A manual review may be required before the requests are entered into the Host System.

Table 3.17 PUT submissions Resource Information

<b>HTTP Method</b>	PUT
<b>Pathname</b>	/cdi/[ver]/submissions
<b>Request Content-Type</b>	application/json; charset=utf-8
<b>Request Content</b>	newJurisdictions Object. See <a href="#">Section 3.1.3.1, newJurisdictions Object</a> for details.
<b>Response Content-Type</b>	application/json; charset=utf-8
<b>Response Content</b>	None.

#### 3.1.3.1 newJurisdictions Object

The following table identifies the properties of the `newJurisdictions` object. Additional properties **MAY** be included in the `newJurisdictions` object.

Table 3.18 newJurisdictions Properties

Property	Restrictions	Description
<code>newJurisdictionArray</code>	type: <code>newJurisdiction</code> use: required mixItems: 1 maxItems: ∞	Array of <code>newJurisdiction</code> Objects. See <a href="#">Section 3.1.3.2, newJurisdiction Object</a> for details.

#### 3.1.3.2 newJurisdiction Object

The following table identifies the properties of the `newJurisdiction` object. Additional properties **MAY** be included in the `newJurisdiction` object.

- The `certificationId` and `componentId` **MUST** match the `certificationId` and `componentId` from a previously submitted certification request.

Table 3.19 newJurisdiction Properties

Property	Restrictions	Description
<code>certificationId</code>	type: <code>t_certificationId</code> use: required	UUID from a previous certification request record.  <b>Example:</b> 96CA8F38-A692-4748-847A-A1DA00D2B95C

Table 3.19 newJurisdiction Properties

Property	Restrictions	Description
componentId	type: t_componentId use: required	UUID for the product component.  <b>Example: 96CA8F38-A692-4748-847A-A1DA00D2B95C</b>
jurisdictionSubArray	type: jurisdictionSub use: required minItems: 0 maxItems: ∞	Array of jurisdictionSub objects. See <a href="#">Section 3.1.1.4, jurisdictionSub Object</a> for details.
signatureArray	type: signature use: required minItems: 0 maxItems: ∞	Array of signature objects. See <a href="#">Section 3.1.1.5, signature Object</a> for details.

### 3.1.3.3 PUT submissions Example

The following example demonstrates the construction of a PUT submissions request and a response indicating that the request was accepted. In practice, additional HTTP headers may be included in the messages.

Request:

```
PUT /cdi/1.0/submissions HTTP/1.1
Content-Length: 1177
Content-Type: application/json; charset=utf-8
```

```
{
  "newJurisdictionArray": [
    {
      "certificationId": "96CA8F38-A692-4748-847A-A1DA00D2B95C",
      "componentId": "BA8B450F-8C2F-43D3-8FEF-04B59366543D",
      "jurisdictionSubArray": [
        {
          "submissionId": "7FDE945E-24FE-414C-BF6E-E0F3A1DA00E3",
          "jursidictionId": "E9457FDE-24FE-414C-BF6E-A1DA00E0F3E3",
          "rush": "true",
          "requestedDate": "2015-12-31",
          "noteArray": [
            {
              "noteId": "968F38CA-A692-4748-847A-A1DA00D2B95C",
              "noteType": "Special Request",
              "noteText": "Field trial requested.",
              "noteSource": "456A3411-78FA-3234-B341-0078CA123489"
            }
          ],
          "documentArray": [
            {
              "documentId": "8F3896CA-A692-4748-847A-A1DA00D2B95C",
              "documentType": "Submission Package",
              "documentName": "ABV_Package_123456",
              "documentSource": "456A3411-78FA-3234-B341-0078CA123489",
              "documentFormat": "zip",
              "documentLanguage": "en",
              "documentPath": "https://www.abv.com/packages/ABV_Package_123456"
            }
          ]
        }
      ]
    }
  ]
}
```

```
    }  
  ],  
  "signatureArray": [  
    {  
      "signatureId": "80E47545-2B24-4DF9-A583-688404695F39",  
      "algorithmId": "754580E4-2B24-4DF9-A583-469688405F39",  
      "verifyResult": "1234567890123456789012345678901234567890"  
    },  
    {  
      "signatureId": "E4758045-2B24-4DF9-A583-884046695F39",  
      "algorithmId": "0BEEB53E-45AE-4F53-9ED5-456E1CE8FFB8",  
      "seed": "0123",  
      "verifyResult": "1234"  
    }  
  ]  
]  
}
```

Response:

```
HTTP/1.1 200 OK  
Content-Length: 0
```