

ABACUS REST API

Overview

The ABACUS REST API exposes endpoints for querying and modifying an ABACUS Object Model.

The API supports **OData V4** query syntax. OData query parameters are optional; however, they offer complex operations including filtering and aggregate queries. See the official OData website for information: <https://www.odata.org/>

Using the ABACUS REST API

For information about the basic operation of the ABACUS REST API, please refer to the *ABACUS Integration Guide* which is located in the *Manuals* directory of your ABACUS installation.

EDM Model

Please refer to the OData Entity Data Model (EDM) metadata for the complete OData v4 data model reference for this API.

The EDM Metadata endpoint can be found at `/$metadata` relative to the root of the API; for example:

```
https://example.com/api/$metadata
```

OData Client Code Generation

It is possible to use a tool to generate a data model from the ABACUS REST API endpoints automatically. Please refer to the following web page to learn more about consuming an OData service using a library: <https://www.odata.org/libraries/>

OData Expand Option

When querying datasets, it is possible to *expand* one or more **navigation properties**.

A **navigation property** is a reference to a related object. For example, a **Component** has a **ComponentType** navigation property.

Navigation properties are not *expanded* by default. To expand a navigation property, use the `$expand={property}` query parameter. Example:

```
https://example.com/api/Components?$expand=ComponentType
```

To expand navigation properties inside an expanded navigation property, use the nested syntax:

```
https://example.com/api/Components?$expand=ComponentType($expand=SuperType)
```

It is also possible to expand all navigation properties by using a wildcard:

```
https://example.com/api/Components?$expand=*
```

OData Filtering

It is possible to apply a filter to a query to return a subset of the data at an endpoint. For example, it is possible to list all Components whose name starts with 'Example':

```
https://example.com/api/Components?$filter=startsWith(name, 'Example')
```

It is possible to filter based on properties within navigation properties without expanding. The following example lists Components whose Component Type's *name* starts with 'Example':

```
https://example.com/api/Components?$filter=startsWith(ComponentType/name, 'Example')
```

OData Response Length Limits

When requesting a large collection of entities (that have more than 512 entities), the response will only contain a subset of the requested collection. In this case, the response will have the `@odata.nextLink` property that contains a URL for the retrieval of the next subset of the requested selection. The following example shows a possible value of the `@odata.nextLink` property:

```
https://example.com/api/Components?$skip=512
```

More Information on OData

To learn more about writing OData queries, please refer to the following web page:

<https://www.odata.org/getting-started/basic-tutorial/>

Copyright

Copyright © 2001 - 2022 Avolution Pty Ltd, related entities and/or licensors. All rights reserved.

Getting Started

To get started with using this API, please consider using an API development tool when learning the endpoints and their responses. We recommended [Postman](#).

Throughout this document, cURL examples are provided for key operations. cURL is a command-line utility, available on Windows and Unix, for making HTTP requests programatically. We provide cURL examples to clarify how requests are to be made to the API.

Example Requests

This section will guide you through a typical procedure of surveying the ABACUS project contents and creating and updating elements of the project.

These examples are designed to work with an ABACUS Console instance (listening at <http://localhost:8080/>) and with the ABACUS Example Project (Example.abacus). Authentication is not required for this project.

1) Update the project to the latest version

This updates the ABACUS Collaboration Server project to the latest edition.

Note that this will only work when the project is a Collaboration Server project.

```
curl -X POST \
  http://localhost:8080/Update \
  -H 'Accept: application/json' \
  -d ''
```

2) List all architectures in the project

This will return all architectures and their sub-architectures recursively.

```
curl -X GET \
  http://localhost:8080/AllArchitectures \
  -H 'Accept: application/json'
```

3) List details of the architecture with the EEID "4929"

This will return the details of an architecture with the EEID "4929" (shown in parentheses).

```
curl -X GET \
  http://localhost:8080/AllArchitectures\ (4929) \
  -H 'Accept: application/json'
```

4) Create a new component in the architecture with the EEID "4929"

A new component will be created in the Architecture with the EEID "4929", and its type will be the component type with the EEID "13".

```
curl -X POST \
  http://localhost:8080/Components \
  -H 'Accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "Name": "Example Application Component",
    "Description": "Submitted from the API",
    "ArchitectureEEID": 4929,
    "ComponentTypeEEID": 13
  }'
```

5) Modify the name of the component with the EEID "5507" (the "ABACUS" Application component)

The component's name and description will be changed

```
curl -X PATCH \
  http://localhost:8080/Components\ (5507) \
  -H 'Accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "Name": "ABACUS (modified)",
    "Description": "Modified from the API"
  }'
```

6) Create a new Interfaces connection from Application 123 to Application 321 in the Production (As-Is) architecture

A new connection will be created. Note that only names are used in this request (no EEIDs).

```
curl -X POST \
  http://localhost:8080/Connections \
  -H 'Accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "Name": "Example Interfaces Connection",
    "ArchitectureName": "Production (As-Is)",
    "ConnectionTypeName": "Interfaces",
    "SourceComponentName": "Application 123",
    "SinkComponentName": "Application 321"
  }'
```

7) Save changes to the file / commit to Collaboration Server

Once you have finished making changes using the API, you can save your changes (and commit them if the project is a Collaboration Server project).

Note that this operation can take a long time to complete in some projects as simulators and scripts may be required to run.

```
curl -X POST \
  http://localhost:8080/Commit \
  -H 'Accept: application/json'
```

(Advanced) List all Applications in the Production architecture whose "Lifecycle|Recommendation" property value is "Retain / Invest", max 10 results

Note that the request path is URL-encoded so that it is compatible with the CLI.

```
curl -X GET \
  'http://localhost:8080/Components?$filter=ArchitectureName%20eq%20%27Production%20%28As-Is%29%27%20and%20ComponentTypeName%20eq%20%27Application%27%20and%20Properties/any%28prop:%20prop/Type%20eq%20%27Lifecycle%27%20and%20prop/Name%20eq%20%27Recommendation%27%20and%20prop/Value%20eq%20%27Retain%20/%20Invest%27%29&$top=10' \
  -H 'Accept: application/json'
```

Version information

Version : 10.0.0

Contact information

Contact : Avolution Pty Ltd

Contact Email : support@avolutionsoftware.com

Tags

- Administrative : Endpoints relating to management of the ABACUS host (server)
- Architectures : Architectures in the model

- Components : Components in the model
- Connections : Connections in the model
- Constraints : Constraints defined in the model
- Object Model : The root of the ABACUS project
- Properties : Properties on objects in the model
- Types : Component Types and Connection Types in the metamodel
- User Management : Endpoints relating to the management of users and user groups

Paths

Get information about the server

GET /About

Description

Get informatoin regarding the API server and the hosted file

Responses

HTTP Code	Description	Schema
200	OK	About

Tags

- Administrative

Get all Architectures in the model, including children

GET /AllArchitectures

Responses

HTTP Code	Description	Schema
200	OK	< CArchitecture > array
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Architectures

Get the architecture with the specified EEID

```
GET /AllArchitectures({EEID})
```

Parameters

Type	Name	Schema
Path	EEID <i>required</i>	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	CArchitecture
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Architectures

Get the child architectures of an architecture

```
GET /AllArchitectures({EEID})/Architectures
```

Parameters

Type	Name	Schema
Path	EEID <i>required</i>	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	< CArchitecture > array
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Architectures

Get the child catalogues of an architecture

```
GET /AllArchitectures({EEID})/Catalogues
```

Parameters

Type	Name	Schema
Path	EEID <i>required</i>	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	< CCatalog > array
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Architectures
- Catalogues

Get the child charts of an architecture

```
GET /AllArchitectures({EEID})/Charts
```

Parameters

Type	Name	Schema
Path	EEID <i>required</i>	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	< CChart > array
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Architectures
- Charts

Get the child components of an architecture

```
GET /AllArchitectures({EEID})/Components
```

Parameters

Type	Name	Schema
Path	EEID <i>required</i>	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	< CComponent > array
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Architectures
- Components

Get the child connections of an architecture

```
GET /AllArchitectures({EEID})/Connections
```

Parameters

Type	Name	Schema
Path	EEID <i>required</i>	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	< CConnection > array
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Architectures
- Connections

Get the child diagrams of an architecture

```
GET /AllArchitectures({EEID})/Diagrams
```

Parameters

Type	Name	Schema
Path	EEID <i>required</i>	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	< CTwoDVisualisation > array
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Architectures
- Diagrams

Get the child matrices of an architecture

```
GET /AllArchitectures({EEID})/Matrices
```

Parameters

Type	Name	Schema
Path	EEID <i>required</i>	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	< CMatrixView > array
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Architectures
- Matrices

Get the properties on an architecture

```
GET /AllArchitectures({EEID})/Properties
```

Parameters

Type	Name	Schema
Path	EEID <i>required</i>	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	< CProperty > array
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Architectures
- Properties

Get the child 3D visualisations of an architecture

```
GET /AllArchitectures({EEID})/ThreeDVisualisations
```

Parameters

Type	Name	Schema
Path	EEID <i>required</i>	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	< CAbacus3D > array
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- 3D Visualisations
- Architectures

Get all constraints

GET /AllConstraints

Responses

HTTP Code	Description	Schema
200	OK	< CConstraint > array
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Constraints

Get the constraint with the specified EEID

```
GET /AllConstraints({EEID})
```

Parameters

Type	Name	Schema
Path	EEID <i>required</i>	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	CConstraint
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Constraints

Get the top-level Architectures in the model

GET /Architectures

Responses

HTTP Code	Description	Schema
200	OK	< CArchitecture > array
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Architectures

Get Catalogues from all Architectures

GET /Catalogues

Responses

HTTP Code	Description	Schema
200	OK	< CCatalog > array
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Catalogues

Get the Catalogue with the specified EEID

```
GET /Catalogues({EEID})
```

Parameters

Type	Name	Schema
Path	EEID <i>required</i>	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	CCatalog
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Catalogues

Get Charts from all Architectures

GET /Charts

Responses

HTTP Code	Description	Schema
200	OK	< CChart > array
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Charts

Get the Chart with the specified EEID

```
GET /Charts({EEID})
```

Parameters

Type	Name	Schema
Path	EEID <i>required</i>	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	CChart
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Charts

Get child constraints

GET /ChildConstraints

Responses

HTTP Code	Description	Schema
200	OK	< CChildConstraint > array
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Constraints

Get the child constraint with the specified EEID

```
GET /ChildConstraints({EEID})
```

Parameters

Type	Name	Schema
Path	EEID <i>required</i>	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	CChildConstraint
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Constraints

Commit changes to Collaboration Server

POST /Commit

Description

Push changes made by the API (or other processes running on the same ABACUS project host) to the Collaboration Server. Only supported when the ABACUS project is a Collaboration Server project. It is recommended that this is called once all necessary changes have been made to the project, but not necessarily between changes.

Responses

HTTP Code	Description	Schema
200	Committed, Saved (when file is a Standalone File) or NoChanges (there are no changes to commit)	enum (Committed, Saved, NoChanges)
401	Bearer token is missing or expired (unauthorized)	No Content
500	Error saving, committing or preparing to commit	enum (UnknownError, SaveFailed, FileInvalid, UpdateFailed, CommitFailed, EECLSaveFailed)

Tags

- Administrative

Example HTTP response

Response 200

"Committed"

Response 500

"UpdateFailed"

Get "component must be attached" constraints

```
GET /ComponentAttachedConstraints
```

Responses

HTTP Code	Description	Schema
200	OK	< CComponentAttachedConstraint > array
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Constraints

Get the component-attached constraint with the specified EEID

```
GET /ComponentAttachedConstraints({EEID})
```

Parameters

Type	Name	Schema
Path	EEID <i>required</i>	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	CComponentAttachedConstraint
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Constraints

Get component constraints

GET /ComponentConstraints

Responses

HTTP Code	Description	Schema
200	OK	< CConstraint > array
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Components
- Constraints

Get the component constraint with the specified EEID

```
GET /ComponentConstraints({EEID})
```

Parameters

Type	Name	Schema
Path	EEID <i>required</i>	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	CConstraint
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Components
- Constraints

Get all component types

GET /ComponentTypes

Responses

HTTP Code	Description	Schema
200	OK	< CComponentType > array
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Types

Get the Component Type with the specified EEID

```
GET /ComponentTypes({EEID})
```

Parameters

Type	Name	Schema
Path	EEID <i>required</i>	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	CComponentType
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Types

Get the supertype of a component type

```
GET /ComponentTypes({EEID})/SuperType
```

Parameters

Type	Name	Description	Schema
Path	EEID <i>required</i>	EEID of the component type whose supertype to return	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	CComponentType
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Types

Create a new component

POST /Components

Description

The following pairs of properties are **mutually exclusive**, meaning that if one is supplied, the other is not required.: - *ComponentTypeEEID* or *ComponentTypeName* - *ArchitectureEEID* or *ArchitectureName* - *ParentEEID* or *ParentName*

It is possible to specify the element type, Architecture or parent element by EEID or name (when the name is unique).

cURL Examples

Create a component in an architecture using EEIDs:

```
curl -X POST \
  http://localhost:8080/Components \
  -H 'Accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "Name": "Example Application Component",
    "Description": "Submitted from the API",
    "ArchitectureEEID": 4929,
    "ComponentTypeEEID": 13
  }'
```

Create a component under an existing component using names:

```
curl -X POST \
  http://localhost:8080/Components \
  -H 'Accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "Name": "Brazil",
    "Description": "A country",
    "ParentEEID": 5992,
    "ComponentTypeName": "Location"
  }'
```

Body parameter

Describes the component to be created

Name : body

Flags : required

Name	Description	Schema
ArchitectureEEID <i>optional</i>	The EEID of the Architecture in which to create the Component. This parameter does not need to be supplied if ParentEEID / ParentName is supplied Minimum value : 1 Example : 1	integer (int64)

Name	Description	Schema
ArchitectureName <i>optional</i>	The Name of the Architecture in which to create the Component. This parameter is optional and can be supplied instead of ArchitectureEEID Example : "OPTIONAL (overrides EEID) "	string
ComponentTypeEEID <i>optional</i>	The EEID of the type of the Component to create Minimum value : 1 Example : 1	integer (int64)
ComponentTypepeName <i>optional</i>	The Name of the type of the Component to create. This parameter is optional and can be supplied instead of ComponentTypeEEID Example : "OPTIONAL (overrides EEID) "	string
Description <i>optional</i>		string
Name <i>optional</i>		string
ParentEEID <i>optional</i>	Minimum value : -1 Example : -1	integer (int64)
ParentName <i>optional</i>	Optional : use this when updating a Component to specify the parent element by name instead of EEID . Example : "OPTIONAL (overrides EEID) "	string

Responses

HTTP Code	Description	Schema
200	Created	CComponent
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Components

Get all components in the Object Model

GET /Components

Responses

HTTP Code	Description	Schema
200	OK	< CComponent > array
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Components

Get the component with the specified EEID

```
GET /Components({EEID})
```

Description

Returns a single component

Parameters

Type	Name	Description	Schema
Path	EEID <i>required</i>	EEID of the component to return	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	CComponent
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Components

Delete the component with the specified EEID from the model

```
DELETE /Components({EEID})
```

Parameters

Type	Name	Description	Schema
Path	EEID <i>required</i>	EEID of the component to delete	integer (int64)
Query	ignoreDependencies <i>optional</i>	By default, requests to delete a component with incoming or outgoing connections will be refused. Set this parameter "true" to force-delete the component and detach the connections.	boolean

Responses

HTTP Code	Description	Schema
200	OK	No Content
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Components

Update the component with the specified EEID

```
PATCH /Components({EEID})
```

Description

Update any or all of the available properties on the component with the specified EEID.

The following pairs of properties are **mutually exclusive**, meaning that if one is supplied, the other is not required.: - *ParentEEID* or *ParentName*

cURL Example

Modify the name of the component with the EEID "5507" (the "ABACUS" Application component)

```
curl -X PATCH \
  http://localhost:8080/Components\ (5507\ ) \
  -H 'Accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "Name": "ABACUS (modified)",
    "Description": "Modified from the API"
  }'
```

Change the parent component of "ABACUS" (EEID 5507) to "Japan" (EEID 1741835)

```
curl -X PATCH \
  http://localhost:8080/Components\ (5507\ ) \
  -H 'Accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "ParentEEID": 1741835
  }'
```

Parameters

Type	Name	Description	Schema
Path	EEID <i>required</i>	EEID of the component to update	integer (int64)

Body parameter

Which properties to update and their values

Name : body

Flags : required

Type : [CComponentPatch](#)

Responses

HTTP Code	Description	Schema
200	Updated	No Content

HTTP Code	Description	Schema
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Components

Get the incoming and outgoing connections of a component

```
GET /Components({EEID})/AllConnections
```

Parameters

Type	Name	Description	Schema
Path	EEID <i>required</i>	EEID of the component whose connections to return	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	< CConnection > array
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Components
- Connections

Get the parent Architecture of the specified component

```
GET /Components({EEID})/Architecture
```

Parameters

Type	Name	Description	Schema
Path	EEID <i>required</i>	EEID of the component whose property to return	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	CArchitecture
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Architectures
- Components

Get the type of a component

```
GET /Components({EEID})/ComponentType
```

Parameters

Type	Name	Description	Schema
Path	EEID <i>required</i>	EEID of the component whose type to return	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	CComponentType
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Components
- Types

Get the children of a component

```
GET /Components({EEID})/Components
```

Parameters

Type	Name	Description	Schema
Path	EEID <i>required</i>	EEID of the component whose children to return	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	< CComponent > array
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Components

Get the incoming connections of a component

```
GET /Components({EEID})/InConnections
```

Parameters

Type	Name	Description	Schema
Path	EEID <i>required</i>	EEID of the component whose incoming connections to return	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	< CConnection > array
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Components
- Connections

Get the outgoing connections of a component

```
GET /Components({EEID})/OutConnections
```

Parameters

Type	Name	Description	Schema
Path	EEID <i>required</i>	EEID of the component whose outgoing connections to return	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	< CConnection > array
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Components
- Connections

Lists the properties of a component

```
GET /Components({EEID})/Properties
```

Parameters

Type	Name	Description	Schema
Path	EEID <i>required</i>	EEID of the component whose properties to return	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	< CElementProperty > array
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Components
- Properties

Get a specific property of a component

```
GET /Components({EEID})/Properties('{type}|{name}')
```

Parameters

Type	Name	Description	Schema
Path	EEID <i>required</i>	EEID of the component whose property to return	integer (int64)
Path	name <i>required</i>	The property name of the property	string
Path	type <i>required</i>	The property type of the property	string

Responses

HTTP Code	Description	Schema
200	OK	CElementProperty
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Components
- Properties

Restores the value of a specific property of a component to its default value

```
DELETE /Components({EEID})/Properties('{type}|{name}')
```

Parameters

Type	Name	Description	Schema
Path	EEID <i>required</i>	EEID of the component whose property to reset	integer (int64)
Path	name <i>required</i>	The property name of the property	string
Path	type <i>required</i>	The property type of the property	string

Responses

HTTP Code	Description	Schema
200	OK	No Content
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Components
- Properties

Update the value of a specific property of a component

```
PATCH /Components({EEID})/Properties('{type}|{name}')
```

Description

cURL Example

Change the "Lifecycle|Stage" property of the ABACUS component (EEID 5507) to "Test". Note that the request path is URL-encoded.

```
curl -X PATCH \
  'http://localhost:8080/Components%285507%29/Properties%28%27Lifecycle|Stage%27%29' \
  -H 'Accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "Value": "Test"
  }'
```

Parameters

Type	Name	Description	Schema
Path	EEID <i>required</i>	EEID of the component whose property to update	integer (int64)
Path	name <i>required</i>	The property name of the property	string
Path	type <i>required</i>	The property type of the property	string

Body parameter

The new property value

Name : body

Flags : optional

Type : [CElementPropertyPatch](#)

Responses

HTTP Code	Description	Schema
200	Updated	No Content
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Components
- Properties

Get the standards on a component

```
GET /Components({EEID})/Standards
```

Parameters

Type	Name	Description	Schema
Path	EEID <i>required</i>	EEID of the component whose standards to return	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	< CStandard > array
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Components
- Standards

Get the components attached to a component

```
GET /ComponentsAttachedTo({EEID})
```

Description

Get the components which are attached to a specified component via a connection. It is possible to filter the results by the connection type by specifying the **connType** query parameter.

Parameters

Type	Name	Description	Schema
Path	EEID <i>required</i>	The EEID of the component on the whose attached components to get.	integer (int64)
Query	connType <i>optional</i>	To select components attached to this component which are attached via a specific connection type , specify the EEID of a connection type in this parameter.	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	< CComponent > array
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Components
- Connections

Get "connection must be attached" constraints

GET /ConnectionAttachedConstraints

Responses

HTTP Code	Description	Schema
200	OK	< CConnectionAttac hedConstraint > array
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Constraints

Get the connection-attached constraint with the specified EEID

```
GET /ConnectionAttachedConstraints({EEID})
```

Parameters

Type	Name	Schema
Path	EEID <i>required</i>	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	CConnectionAttachedConstraint
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Constraints

Get connection constraints

GET /ConnectionConstraints

Responses

HTTP Code	Description	Schema
200	OK	< CConstraint > array
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Connections
- Constraints

Get the connection constraint with the specified EEID

```
GET /ConnectionConstraints({EEID})
```

Parameters

Type	Name	Schema
Path	EEID <i>required</i>	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	CConstraint
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Connections
- Constraints

Get all connection types

GET /ConnectionTypes

Responses

HTTP Code	Description	Schema
200	OK	< CConnectionType > array
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Types

Get the Connection Type with the specified EEID

```
GET /ConnectionTypes({EEID})
```

Parameters

Type	Name	Schema
Path	EEID <i>required</i>	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	CConnectionType
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Types

Get the supertype of a connection type

```
GET /ConnectionTypes({EEID})/SuperType
```

Parameters

Type	Name	Description	Schema
Path	EEID <i>required</i>	EEID of the connection type whose supertype to return	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	CConnectionType
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Types

Create a new connection

POST /Connections

Description

The following pairs of properties are **mutually exclusive**, meaning that if one is supplied, the other is not required.: - *ConnectionTypeEEID* or *ConnectionTypeName* - *ArchitectureEEID* or *ArchitectureName* - *ParentEEID* or *ParentName* - *SourceComponentEEID* or *SourceComponentName* - *SinkComponentEEID* or *SinkComponentName*

It is possible to specify the element type, Architecture, parent element, source or sink by EEID or name (when the name is unique).

cURL Example

Create a new Interfaces connection from Application 123 to Application 321 in the Production (As-Is) architecture

```
curl -X POST \
  http://localhost:8080/Connections \
  -H 'Accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "Name": "Example Interfaces Connection",
    "ArchitectureName": "Production (As-Is)",
    "ConnectionTypeName": "Interfaces",
    "SourceComponentName": "Application 123",
    "SinkComponentName": "Application 321"
  }'
```

Body parameter

Describes the connection to be created

Name : body

Flags : required

Name	Description	Schema
ArchitectureEEID <i>optional</i>	The EEID of the Architecture in which to create the Connection. This parameter is not required if ParentEEID / ParentName is supplied Minimum value : 1 Example : 1.0	number (int64)
ArchitectureName <i>optional</i>	The Name of the Architecture in which to create the Connection. This parameter is optional and can be used instead of ArchitectureEEID Example : "OPTIONAL (overrides EEID)"	string
ConnectionTypeEEID <i>optional</i>	The EEID of the type of Connection to create Minimum value : 1 Example : 1.0	number (int64)

Name	Description	Schema
ConnectionType peName <i>optional</i>	The Name of the type of Connection to create. This parameter is optional and can be supplied instead of ConnectionTypeEEID Example : "OPTIONAL (overrides EEID) "	string
Description <i>optional</i>		string
Name <i>optional</i>		string
ParentEEID <i>optional</i>	Minimum value : -1 Example : -1	integer (int64)
ParentName <i>optional</i>	Optional: use this when updating a Connection to specify the parent element by name instead of EEID . Example : "OPTIONAL (overrides EEID) "	string
SinkComponentEEID <i>optional</i>	Minimum value : -1 Example : -1	integer (int64)
SinkComponentName <i>optional</i>	Optional: use this when updating a Connection to specify the sink element by name instead of EEID . Example : "OPTIONAL (overrides EEID) "	string
SourceComponentEEID <i>optional</i>	Minimum value : -1 Example : -1	integer (int64)
SourceComponentName <i>optional</i>	Optional: use this when updating a Connection to specify the source element by name instead of EEID . Example : "OPTIONAL (overrides EEID) "	string

Responses

HTTP Code	Description	Schema
200	Created	CConnection
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Connections

Lists all connections in the model

GET /Connections

Responses

HTTP Code	Description	Schema
200	OK	< CConnection > array
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Connections

Get the connection with the specified EEID

```
GET /Connections({EEID})
```

Description

Returns a single connection

Parameters

Type	Name	Description	Schema
Path	EEID <i>required</i>	EEID of the connection to return	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	CConnection
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Connections

Delete the connection with the specified EEID from the model

```
DELETE /Connections({EEID})
```

Parameters

Type	Name	Description	Schema
Path	EEID <i>required</i>	EEID of the connection to delete	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	No Content
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Connections

Update the connection with the specified EEID

```
PATCH /Connections({EEID})
```

Description

Update any or all of the available properties on the connection with the specified EEID

The following pairs of properties are **mutually exclusive**, meaning that if one is supplied, the other is not required.: - *ParentEEID* or *ParentName* - *SourceComponentEEID* or *SourceComponentName* - *SinkComponentEEID* or *SinkComponentName*

Parameters

Type	Name	Description	Schema
Path	EEID <i>required</i>	EEID of the connection to update	integer (int64)

Body parameter

Which properties to update and their values

Name : body

Flags : required

Type : [CConnectionPatch](#)

Responses

HTTP Code	Description	Schema
200	Updated	No Content
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Connections

Lists the properties of a connection

```
GET /Connections({EEID})/Properties
```

Parameters

Type	Name	Description	Schema
Path	EEID <i>required</i>	EEID of the connection whose properties to return	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	< CElementProperty > array
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Connections
- Properties

Get a specific property of a connection

```
GET /Connections({EEID})/Properties('{type}|{name}')
```

Parameters

Type	Name	Description	Schema
Path	EEID <i>required</i>	EEID of the connection whose property to return	integer (int64)
Path	name <i>required</i>	The property name of the property	string
Path	type <i>required</i>	The property type of the property	string

Responses

HTTP Code	Description	Schema
200	OK	CElementProperty
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Connections
- Properties

Restores the value of a specific property of a connection to its default value

```
DELETE /Connections({EEID})/Properties('{type}|{name}')
```

Parameters

Type	Name	Description	Schema
Path	EEID <i>required</i>	EEID of the connection whose property to reset	integer (int64)
Path	name <i>required</i>	The property name of the property	string
Path	type <i>required</i>	The property type of the property	string

Responses

HTTP Code	Description	Schema
200	OK	No Content
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Connections
- Properties

Update the value of a specific property of a connection

```
PATCH /Connections({EEID})/Properties('{type}|{name}')
```

Parameters

Type	Name	Description	Schema
Path	EEID <i>required</i>	EEID of the connection whose property to update	integer (int64)
Path	name <i>required</i>	The property name of the property	string
Path	type <i>required</i>	The property type of the property	string

Body parameter

The new property value

Name : body

Flags : optional

Type : [CElementPropertyPatch](#)

Responses

HTTP Code	Description	Schema
200	Updated	No Content
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Connections
- Properties

Get all diagrams

GET /Diagrams

Responses

HTTP Code	Description	Schema
200	OK	< CTwoDVisualisatio n > array
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Diagrams

Get the diagram with the specified EEID

```
GET /Diagrams({EEID})
```

Parameters

Type	Name	Schema
Path	EEID <i>required</i>	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	CTwoDVisualisation
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Diagrams

Get a property of an element

```
GET /ElementProperties('{ElementEEID}|{Type}|{Name}')
```

Description

Use this endpoint as a shorthand method to retrieve a property of a component or connection

Parameters

Type	Name	Description	Schema
Path	ElementEEID <i>required</i>	The EEID of the component or connection which holds the property	integer (int64)
Path	Name <i>required</i>	The Name of the property	string
Path	Type <i>required</i>	The Type of the property	string

Responses

HTTP Code	Description	Schema
200	OK	CElementProperty
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Properties

Reset a property of an element to its default

```
DELETE /ElementProperties('{ElementEEID}|{Type}|{Name}')
```

Description

Use this endpoint as a shorthand method to reset a property of a component or connection to its default value

Parameters

Type	Name	Description	Schema
Path	ElementEEID <i>required</i>	The EEID of the component or connection which holds the property	integer (int64)
Path	Name <i>required</i>	The Name of the property	string
Path	Type <i>required</i>	The Type of the property	string

Responses

HTTP Code	Description	Schema
200	OK	No Content
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Properties

Update the value of a property of an element

```
PATCH /ElementProperties('{ElementEEID}|{Type}|{Name}')
```

Description

Use this endpoint as a shorthand method to update the value of a property of a component or connection

Parameters

Type	Name	Description	Schema
Path	ElementEEID <i>required</i>	The EEID of the component or connection which holds the property	integer (int64)
Path	Name <i>required</i>	The Name of the property	string
Path	Type <i>required</i>	The Type of the property	string

Body parameter

Name : body

Flags : required

Type : [CElementPropertyPatch](#)

Responses

HTTP Code	Description	Schema
200	Updated	No Content
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Properties

Get Matrices from all Architectures

GET /Matrices

Responses

HTTP Code	Description	Schema
200	OK	< CMatrixView > array
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Matrices

Get the Matrix with the specified EEID

```
GET /Matrices({EEID})
```

Parameters

Type	Name	Schema
Path	EEID <i>required</i>	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	CMatrixView
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Matrices

Get details about the Object Model (ABACUS project)

GET /ObjectModel

Description

cURL Example

```
curl -X GET \  
  http://localhost:8080/ObjectModel \  
  -H 'Accept: application/json' \  
  -H 'Content-Type: application/json'
```

Responses

HTTP Code	Description	Schema
200	OK	CObjectModel
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Object Model

Get parent constraints

GET /ParentConstraints

Responses

HTTP Code	Description	Schema
200	OK	< CParentConstraint > array
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Constraints

Get the parent constraint with the specified EEID

```
GET /ParentConstraints({EEID})
```

Parameters

Type	Name	Schema
Path	EEID <i>required</i>	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	CParentConstraint
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Constraints

Get all standards

GET /Standards

Responses

HTTP Code	Description	Schema
200	OK	< CStandard > array
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Standards

Get the standard with the specified EEID

```
GET /Standards({EEID})
```

Parameters

Type	Name	Schema
Path	EEID <i>required</i>	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	CStandard
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- Standards

Get all 3D visualisations

GET /ThreeDViews

Responses

HTTP Code	Description	Schema
200	OK	< CAbacus3D > array
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- 3D Visualisations

Get the 3D visualisation with the specified EEID

```
GET /ThreeDViews({EEID})
```

Parameters

Type	Name	Schema
Path	EEID <i>required</i>	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	CAbacus3D
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- 3D Visualisations

Pull updates from Collaboration Server

POST /Update

Description

Pulls the latest revision of the ABACUS project from the Collaboration Server. Only supported when the ABACUS project is a Collaboration Server project. It is recommended that this is called before any subsequent calls which might make changes to the project in order to mitigate change conflicts.

Responses

HTTP Code	Description	Schema
200	Updated or NoUpdatesAvailable	enum (Updated, NoUpdatesAvailable)
401	Bearer token is missing or expired (unauthorized)	No Content
500	NotEEFile (project is not a Collaboration Server file), FileInvalid, ErrorUpdating or UnknownError	enum (UnknownError, ErrorUpdating, FileInvalid, NotEEFile)

Tags

- Administrative

Example HTTP response

Response 200

"Updated"

Response 500

"NotEEFile"

Upload an ABACUS project

POST /UploadProject

Description

Use this endpoint to upload a standalone ABACUS project file (*.abacus or *.xml). Once the file is uploaded onto the server, the hosted file is substituted with the new file and subsequent calls to the API will refer to the new file. Uploading an ABACUS Collaboration Server project is not supported.

Parameters

Type	Name	Description	Schema
FormData	file <i>required</i>	The ABACUS file to upload	file
FormData	password <i>optional</i>	The password of the ABACUS user with which to open the file	string
FormData	username <i>optional</i>	The username of the ABACUS user with which to open the file	string

Responses

HTTP Code	Description	Schema
200	OK	No Content
400	The uploaded file is not a *.abacus or ABACUS XML file	No Content
401	Bearer token is missing or expired (unauthorized)	No Content
406	Failed to substitute new project file	No Content
415	The request content is not multipart/form-data	No Content
500	Internal error	No Content

Consumes

- `multipart/form-data`

Tags

- Administrative

Create a new user group

POST /UserGroups

Body parameter

Describes the user group to be created

Name : body

Flags : required

Type : [CUserGroupPatch](#)

Responses

HTTP Code	Description	Schema
201	Created	CUserGroup
400	Bad request	No Content
401	Bearer token is missing or expired (unauthorized)	No Content
403	Forbidden (unauthorised)	No Content

Tags

- User Management

Get all user groups

GET /UserGroups

Responses

HTTP Code	Description	Schema
200	OK	< CUserGroup > array
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- User Management

Get a user group with the specified EEID

```
GET /UserGroups({EEID})
```

Parameters

Type	Name	Schema
Path	EEID <i>required</i>	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	CUserGroup
401	Bearer token is missing or expired (unauthorized)	No Content
404	Not found	No Content

Tags

- User Management

Delete a user group with the specified EEID

```
DELETE /UserGroups({EEID})
```

Parameters

Type	Name	Schema
Path	EEID <i>required</i>	integer (int64)

Responses

HTTP Code	Description	Schema
204	No Content	No Content
401	Bearer token is missing or expired (unauthorized)	No Content
403	Forbidden (unauthorised)	No Content
404	Not found	No Content

Tags

- User Management

Update the properties of a user group

```
PATCH /UserGroups({EEID})
```

Parameters

Type	Name	Schema
Path	EEID <i>required</i>	integer (int64)

Body parameter

Describes the updated user group properties

Name : body

Flags : required

Type : [CUserGroupPatch](#)

Responses

HTTP Code	Description	Schema
204	No content	No Content
400	Bad request	No Content
401	Bearer token is missing or expired (unauthorized)	No Content
403	Forbidden (unauthorised)	No Content
404	Not found	No Content

Tags

- User Management

Add a user to a user group

```
POST /UserGroups({usergroupEEID})/Users
```

Description

Note that if adding a user to a user group would exceed the AEModeler / AEEEditor limit, a Bad Request response is returned

Parameters

Type	Name	Schema
Path	usergroupEEID <i>required</i>	integer (int64)

Body parameter

Describes the user to be added to the user group by their EEID

Name : body

Flags : required

Name	Description	Schema
EEID <i>optional</i>	Minimum value : 1 Example : 1	integer (int64)

Responses

HTTP Code	Description	Schema
204	No content	No Content
400	Bad request	No Content
401	Bearer token is missing or expired (unauthorized)	No Content
403	Forbidden (unauthorised)	No Content
404	Not found	No Content

Tags

- User Management

Remove a user from a user group

```
DELETE /UserGroups({usergroupEEID})/Users({userEEID})
```

Parameters

Type	Name	Schema
Path	userEEID <i>required</i>	integer (int64)
Path	usergroupEEID <i>required</i>	integer (int64)

Responses

HTTP Code	Description	Schema
204	No content	No Content
401	Bearer token is missing or expired (unauthorized)	No Content
403	Forbidden (unauthorised)	No Content
404	Not found	No Content

Tags

- User Management

Create a new user

POST /Users

Body parameter

Describes the user to be created

Name : body

Flags : required

Type : [CUserPatch](#)

Responses

HTTP Code	Description	Schema
201	Created	CUser
400	Bad request	No Content
401	Bearer token is missing or expired (unauthorized)	No Content
403	Forbidden (unauthorised)	No Content

Tags

- User Management

Get all users

GET /Users

Responses

HTTP Code	Description	Schema
200	OK	< CUser > array
401	Bearer token is missing or expired (unauthorized)	No Content

Tags

- User Management

Get a user with the specified EEID

```
GET /Users({EEID})
```

Parameters

Type	Name	Schema
Path	EEID <i>required</i>	integer (int64)

Responses

HTTP Code	Description	Schema
200	OK	CUser
401	Bearer token is missing or expired (unauthorized)	No Content
404	Not found	No Content

Tags

- User Management

Delete a user with the specified EEID

```
DELETE /Users({EEID})
```

Parameters

Type	Name	Schema
Path	EEID <i>required</i>	integer (int64)

Responses

HTTP Code	Description	Schema
204	No Content	No Content
401	Bearer token is missing or expired (unauthorized)	No Content
403	Forbidden (unauthorised)	No Content
404	Not found	No Content

Tags

- User Management

Update the properties of a user

```
PATCH /Users({EEID})
```

Parameters

Type	Name	Schema
Path	EEID <i>required</i>	integer (int64)

Body parameter

Describes the updated user properties

Name : body

Flags : required

Type : [CUserPatch](#)

Responses

HTTP Code	Description	Schema
204	No content	No Content
400	Bad request	No Content
401	Bearer token is missing or expired (unauthorized)	No Content
403	Forbidden (unauthorised)	No Content
404	Not found	No Content

Tags

- User Management

Definitions

About

Name	Description	Schema
ABACUSVersion <i>optional</i>	The version of ABACUS which is hosting the API	ABACUSVersion
Copyright <i>optional</i>	Copyright notice	string
Help <i>optional</i>	Provides help information for the API	string
ID <i>optional</i>	Always "0"	integer (int32)
ODataAPIVersion <i>optional</i>		ODataAPIVersion
OpenFile <i>optional</i>	The name of the currently open ABACUS file	string
Opened <i>optional</i>	The date-time the file was opened	string (date-time)

ABACUSVersion

Name	Description	Schema
Build <i>optional</i>	Example : 0	integer (int32)
Major <i>optional</i>	Example : 10	integer (int32)
Minor <i>optional</i>	Example : 0	integer (int32)
Revision <i>optional</i>	Example : 0	integer (int32)

ODataAPIVersion

Name	Description	Schema
Build <i>optional</i>	Example : 0	integer (int32)

Name	Description	Schema
Major <i>optional</i>	Example : 4	integer (int32)
Minor <i>optional</i>	Example : 0	integer (int32)
Revision <i>optional</i>	Example : 0	integer (int32)

C3DComponent

Polymorphism : Composition

Name	Description	Schema
B <i>optional</i>		number (double)
Component <i>optional</i>		CComponent
EEID <i>optional</i> <i>read-only</i>	Minimum value : 1 Example : 1	integer (int64)
G <i>optional</i>		number (double)
R <i>optional</i>		number (double)
Radius <i>optional</i>		number (double)
Shape <i>optional</i>		enum (Sphere, Cube)
SubComponents <i>optional</i>		< C3DComponent > array
X <i>optional</i>		number (double)
Y <i>optional</i>		number (double)
Z <i>optional</i>		number (double)

C3DConnection

Polymorphism : Composition

Name	Description	Schema
B <i>optional</i>		number (double)
Component1 <i>optional</i>		C3DComponent
Component2 <i>optional</i>		C3DComponent
Connection <i>optional</i>		CConnection
EEID <i>optional</i> <i>read-only</i>	Minimum value : 1 Example : 1	integer (int64)
G <i>optional</i>		number (double)
R <i>optional</i>		number (double)
Radius <i>optional</i>		number (double)
Visible <i>optional</i>	Example : true	boolean

CAbacus3D

Polymorphism : Composition

Name	Description	Schema
Architecture <i>optional</i>		CArchitecture
Components <i>optional</i>		< C3DComponent > array
Connections <i>optional</i>		< C3DConnection > array
Created <i>optional</i>		string (date-time)

Name	Description	Schema
CreatedBy <i>optional</i>		string
Description <i>optional</i>		string
EEID <i>optional</i> <i>read-only</i>	Minimum value : 1 Example : 1	integer (int64)
Name <i>optional</i>		string
Properties <i>optional</i>		< CProperty > array
SyncWithServiceNow <i>optional</i>		boolean
SyncWithTechnopedia <i>optional</i>		boolean

CArchitecture

Polymorphism : Composition

Name	Description	Schema
Architectures <i>optional</i>		< CArchitecture > array
Catalogues <i>optional</i>		< CCatalog > array
Charts <i>optional</i>		< CChart > array
Components <i>optional</i>		< CComponent > array
Connections <i>optional</i>		< CConnection > array
Description <i>optional</i>		string
Diagrams <i>optional</i>		< CTwoDVisualisation > array

Name	Description	Schema
EEID <i>optional</i> <i>read-only</i>	Minimum value : 1 Example : 1	integer (int64)
Matrices <i>optional</i>		< CMatrixView > array
Name <i>optional</i>		string
Properties <i>optional</i>		< CProperty > array
Standards <i>optional</i>		< CStandard > array
SyncWithServiceNow <i>optional</i>		boolean
SyncWithTechnopedia <i>optional</i>		boolean
ThreeDVisualizations <i>optional</i>		< CAbacus3D > array

CAttachmentConstraint

Polymorphism : Composition

Name	Description	Schema
ComponentType <i>optional</i>		CComponentType
ConnRole <i>optional</i>		ConstraintConnectionRole
ConnectionType <i>optional</i>		CConnectionType
ConstraintKind <i>optional</i>		ConstraintKind

Name	Description	Schema
EEID <i>optional</i> <i>read-only</i>	Minimum value : 1 Example : 1	integer (int64)
Enforcement Mode <i>optional</i>		ConstraintEnforceMode
Name <i>optional</i>		string

CCatalog

Polymorphism : Composition

Name	Description	Schema
Architecture <i>optional</i>		CArchitecture
Created <i>optional</i>		string (date-time)
CreatedBy <i>optional</i>		string
Description <i>optional</i>		string
EEID <i>optional</i> <i>read-only</i>	Minimum value : 1 Example : 1	integer (int64)
ElementTypes <i>optional</i>		< CElementType > array
Name <i>optional</i>		string
Properties <i>optional</i>		< CProperty > array
SyncWithServiceNow <i>optional</i>		boolean
SyncWithTechnopedia <i>optional</i>		boolean

CChart

Polymorphism : Composition

Name	Description	Schema
Architecture <i>optional</i>		CArchitecture
ChartType <i>optional</i>		enum (Standard, Bubble)
Components <i>optional</i>		< CComponent > array
Connections <i>optional</i>		< CConnection > array
Created <i>optional</i>		string (date-time)
CreatedBy <i>optional</i>		string
Description <i>optional</i>		string
EEID <i>optional</i> <i>read-only</i>	Minimum value : 1 Example : 1	integer (int64)
Name <i>optional</i>		string
Properties <i>optional</i>		< CProperty > array
SyncWithServiceNow <i>optional</i>		boolean
SyncWithTechnopedia <i>optional</i>		boolean

CChildConstraint

Polymorphism : Composition

Name	Description	Schema
ChildType <i>optional</i>		CElementType
ConstraintKind <i>optional</i>		ConstraintKind
EEID <i>optional</i> <i>read-only</i>	Minimum value : 1 Example : 1	integer (int64)
EnforcementMode <i>optional</i>		ConstraintEnforceMode
LowerBound <i>optional</i>		integer
Name <i>optional</i>		string
ParentType <i>optional</i>		CElementType
UpperBound <i>optional</i>		integer

CComponent

A component in the object model

Polymorphism : Composition

Name	Description	Schema
AllConnections <i>optional</i> <i>read-only</i>		< CConnection > array
Architecture <i>optional</i>		CArchitecture
ArchitectureEEID <i>optional</i>	Minimum value : 1	integer (int64)
ArchitectureName <i>optional</i>	Optional: use this when creating a Component to specify the architecture by name instead of EEID . Example : "OPTIONAL (overrides EEID) "	string

Name	Description	Schema
BaselineElementEEID <i>optional</i> <i>read-only</i>		integer (int64)
ComponentType <i>optional</i>		CComponentType
ComponentTypeEEID <i>optional</i>	Minimum value : 1	integer (int64)
ComponentTypepeName <i>optional</i>	Optional: use this when creating a Component to specify the element type by name instead of EEID . Example : "OPTIONAL (overrides EEID) "	string
Components <i>optional</i> <i>read-only</i>		< CComponent > array
Created <i>optional</i>		ReadOnlyDate
CreatedBy <i>optional</i> <i>read-only</i>		string
Description <i>optional</i>		string
EEID <i>optional</i> <i>read-only</i>	Minimum value : 1 Example : 1	integer (int64)
InConnections <i>optional</i> <i>read-only</i>		< CConnection > array
Modified <i>optional</i>		ReadOnlyDate
ModifiedBy <i>optional</i> <i>read-only</i>		string
Name <i>optional</i>		string

Name	Description	Schema
OutConnections <i>optional</i> <i>read-only</i>		< CConnection > array
ParentComponent <i>optional</i>		CComponent
ParentEEID <i>optional</i>	Minimum value : -1 Example : -1	integer (int64)
ParentName <i>optional</i>	Optional: use this when creating a Component to specify the parent element by name instead of EEID . Example : "OPTIONAL (overrides EEID) "	string
Properties <i>optional</i> <i>read-only</i>		< CElementProperty > array
Standards <i>optional</i> <i>read-only</i>		< CStandard > array

CComponentAttachedConstraint

Polymorphism : Composition

Name	Description	Schema
ComponentType <i>optional</i>		CComponentType
ConnRole <i>optional</i>		ConstraintConnectionRole
ConnectionType <i>optional</i>		CConnectionType
ConstraintKind <i>optional</i>		ConstraintKind
EEID <i>optional</i> <i>read-only</i>	Minimum value : 1 Example : 1	integer (int64)

Name	Description	Schema
Enforcement Mode <i>optional</i>		ConstraintEnforceMode
Name <i>optional</i>		string

CComponentPatch

A simplified component used only when requesting to update a component through a PATCH endpoint

Name	Description	Schema
Description <i>optional</i>		string
Name <i>optional</i>		string
ParentEEID <i>optional</i>	Minimum value : -1 Example : -1	integer (int64)
ParentName <i>optional</i>	Optional: use this when updating a Component to specify the parent element by name instead of EEID . Example : "OPTIONAL (overrides EEID) "	string

CComponentType

Polymorphism : Composition

Name	Description	Schema
DefaultHierarchyPath <i>optional</i>		< string > array
Description <i>optional</i>		string
EEID <i>optional</i> <i>read-only</i>	Minimum value : 1 Example : 1	integer (int64)
Name <i>optional</i>		string
Properties <i>optional</i>		< CProperty > array

Name	Description	Schema
Standards <i>optional</i>		< CStandard > array
SyncWithServiceNow <i>optional</i>		boolean
SyncWithTechnopedia <i>optional</i>		boolean

CConnection

A connection in the object model

Polymorphism : Composition

Name	Description	Schema
Architecture <i>optional</i>		CArchitecture
ArchitectureEEID <i>optional</i>	Minimum value : 1	integer (int64)
ArchitectureName <i>optional</i>	Optional : use this when creating a Connection to specify the architecture by name instead of EEID . Example : "OPTIONAL (overrides EEID) "	string
BaselineElementEEID <i>optional</i> <i>read-only</i>		integer (int64)
ConnectionType <i>optional</i>		CConnectionType
ConnectionTypeEEID <i>optional</i>	Minimum value : 1	integer (int64)
ConnectionTypeName <i>optional</i>	Optional : use this when creating a Connection to specify the element type by name instead of EEID . Example : "OPTIONAL (overrides EEID) "	string
Connections <i>optional</i> <i>read-only</i>		< CConnection > array

Name	Description	Schema
Created <i>optional</i>		ReadOnlyDate
CreatedBy <i>optional</i> <i>read-only</i>		string
Description <i>optional</i>		string
EEID <i>optional</i> <i>read-only</i>	Minimum value : 1 Example : 1	integer (int64)
Modified <i>optional</i>		ReadOnlyDate
ModifiedBy <i>optional</i> <i>read-only</i>		string
Name <i>optional</i>		string
ParentConnection <i>optional</i>		CConnection
ParentEEID <i>optional</i>	Minimum value : -1 Example : -1	integer (int64)
ParentName <i>optional</i>	Optional: use this when creating a Connection to specify the parent element by name instead of EEID . Example : "OPTIONAL (overrides EEID) "	string
Properties <i>optional</i> <i>read-only</i>		< CElementProperty > array
SinkComponent <i>optional</i>		CComponent
SinkComponentEEID <i>optional</i>	Minimum value : -1 Example : -1	integer (int64)
SinkComponentName <i>optional</i>	Optional: use this when creating a Connection to specify the sink element by name instead of EEID . Example : "OPTIONAL (overrides EEID) "	string

Name	Description	Schema
SourceComponent <i>optional</i>		CComponent
SourceComponentEEID <i>optional</i>	Minimum value : -1 Example : -1	integer (int64)
SourceComponentName <i>optional</i>	Optional: use this when creating a Connection to specify the source element by name instead of EEID . Example : "OPTIONAL (overrides EEID) "	string
Standards <i>optional</i> <i>read-only</i>		< CStandard > array

CConnectionAttachedConstraint

Polymorphism : Composition

Name	Description	Schema
ComponentType <i>optional</i>		CComponentType
ConnRole <i>optional</i>		ConstraintConnectionRole
ConnectionType <i>optional</i>		CConnectionType
ConstraintKind <i>optional</i>		ConstraintKind
EEID <i>optional</i> <i>read-only</i>	Minimum value : 1 Example : 1	integer (int64)
EnforcementMode <i>optional</i>		ConstraintEnforcementMode
IsFullConnectionConstraint <i>optional</i>		boolean
Name <i>optional</i>		string

Name	Description	Schema
SinkComponentType <i>optional</i>		CComponentType
SourceComponentType <i>optional</i>		CComponentType

CConnectionPatch

A simplified connection used only when requesting to update a connection through a PATCH endpoint

Name	Description	Schema
Description <i>optional</i>		string
Name <i>optional</i>		string
ParentEEID <i>optional</i>	Minimum value : -1 Example : -1	integer (int64)
ParentName <i>optional</i>	Optional: use this when updating a Connection to specify the parent element by name instead of EEID . Example : "OPTIONAL (overrides EEID) "	string
SinkComponentEEID <i>optional</i>	Minimum value : -1 Example : -1	integer (int64)
SinkComponentName <i>optional</i>	Optional: use this when updating a Connection to specify the sink element by name instead of EEID . Example : "OPTIONAL (overrides EEID) "	string
SourceComponentEEID <i>optional</i>	Minimum value : -1 Example : -1	integer (int64)
SourceComponentName <i>optional</i>	Optional: use this when updating a Connection to specify the source element by name instead of EEID . Example : "OPTIONAL (overrides EEID) "	string

CConnectionType

Polymorphism : Composition

Name	Description	Schema
DefaultHierarchyPath <i>optional</i>		< string > array
Description <i>optional</i>		string
EEID <i>optional</i> <i>read-only</i>	Minimum value : 1 Example : 1	integer (int64)
Name <i>optional</i>		string
Properties <i>optional</i>		< CProperty > array
Standards <i>optional</i>		< CStandard > array
SyncWithServiceNow <i>optional</i>		boolean
SyncWithTechnopedia <i>optional</i>		boolean

CConstraint

Polymorphism : Composition

Name	Description	Schema
ConstraintKind <i>optional</i>		ConstraintKind
EEID <i>optional</i> <i>read-only</i>	Minimum value : 1 Example : 1	integer (int64)
EnforcementMode <i>optional</i>		ConstraintEnforceMode
Name <i>optional</i>		string

CElement

A component or connection

Polymorphism : Composition

Name	Description	Schema
BaselineElementEEID <i>optional</i> <i>read-only</i>		integer (int64)
Created <i>optional</i>		ReadOnlyDate
CreatedBy <i>optional</i> <i>read-only</i>		string
Description <i>optional</i>		string
EEID <i>optional</i> <i>read-only</i>	Minimum value : 1 Example : 1	integer (int64)
Modified <i>optional</i>		ReadOnlyDate
ModifiedBy <i>optional</i> <i>read-only</i>		string
Name <i>optional</i>		string
Properties <i>optional</i> <i>read-only</i>		< CElementProperty > array
Standards <i>optional</i> <i>read-only</i>		< CStandard > array

CElementProperty

Name	Description	Schema
DataType <i>optional</i> <i>read-only</i>		enum (Text, Integer, Decimal, DateTime, Boolean, List)

Name	Description	Schema
DataTypeEnforcement <i>optional</i> <i>read-only</i>		enum (Disabled, Warned, Enforced)
Description <i>optional</i> <i>read-only</i>	The description of the property. Read-only because its value is determined by the description on the parent CProperty.	string
Element <i>optional</i>		CElement
Identifier <i>optional</i> <i>read-only</i>	The EEID of the element, the property Type and the property Name. In the format "EEID Type Name".	string
Name <i>optional</i> <i>read-only</i>		string
Type <i>optional</i> <i>read-only</i>		string
TypeProperty <i>optional</i>		CTypeProperty
Unit <i>optional</i> <i>read-only</i>		string
Value <i>optional</i>		string

CElementPropertyPatch

Name	Schema
Value <i>optional</i>	string

CElementType

Polymorphism : Composition

Name	Description	Schema
DefaultHierarchyPath <i>optional</i>		< string > array

Name	Description	Schema
Description <i>optional</i>		string
EEID <i>optional</i> <i>read-only</i>	Minimum value : 1 Example : 1	integer (int64)
Name <i>optional</i>		string
Properties <i>optional</i>		< CProperty > array
Standards <i>optional</i>		< CStandard > array
SyncWithServiceNow <i>optional</i>		boolean
SyncWithTechnopedia <i>optional</i>		boolean

CEntityWithDescription

Polymorphism : Composition

Name	Description	Schema
Description <i>optional</i>		string
EEID <i>optional</i> <i>read-only</i>	Minimum value : 1 Example : 1	integer (int64)

CEntityWithProperties

Polymorphism : Composition

Name	Description	Schema
Description <i>optional</i>		string
EEID <i>optional</i> <i>read-only</i>	Minimum value : 1 Example : 1	integer (int64)

Name	Description	Schema
Name <i>optional</i>		string
Properties <i>optional</i>		< CProperty > array
SyncWithServiceNow <i>optional</i>		boolean
SyncWithTechnopedia <i>optional</i>		boolean

CEntityWithStandards

Polymorphism : Composition

Name	Description	Schema
Description <i>optional</i>		string
EEID <i>optional</i> <i>read-only</i>	Minimum value : 1 Example : 1	integer (int64)
Name <i>optional</i>		string
Properties <i>optional</i>		< CProperty > array
Standards <i>optional</i>		< CStandard > array
SyncWithServiceNow <i>optional</i>		boolean
SyncWithTechnopedia <i>optional</i>		boolean

CMatrixView

Polymorphism : Composition

Name	Description	Schema
Architecture <i>optional</i>		CArchitecture
ClassicMode <i>optional</i>		boolean
Created <i>optional</i>		string (date-time)
CreatedBy <i>optional</i>		string
Description <i>optional</i>		string
EEID <i>optional</i> <i>read-only</i>	Minimum value : 1 Example : 1	integer (int64)
Name <i>optional</i>		string
Properties <i>optional</i>		< CProperty > array
SinkComponentTypes <i>optional</i>		< CComponentType > array
SourceComponentTypes <i>optional</i>		< CComponentType > array
SyncWithServiceNow <i>optional</i>		boolean
SyncWithTechnopedia <i>optional</i>		boolean

CObjectModel

Polymorphism : Composition

Name	Description	Schema
AnonymousAccess <i>optional</i>		boolean

Name	Description	Schema
Architectures <i>optional</i>	The architectures at the root level of the model	< CArchitecture > array
ChangesMade <i>optional</i>		boolean
ComplexityCalculatePropertiesPerType <i>optional</i>		boolean
ComplexityRunOnServer <i>optional</i>		boolean
ComponentTypes <i>optional</i>		< CComponentType > array
ConnectionTypes <i>optional</i>		< CConnectionType > array
DefaultGroupingName <i>optional</i>		string
EEID <i>optional</i> <i>read-only</i>	Minimum value : 1 Example : 1	integer (int64)
FileName <i>optional</i>	The file name of the ABACUS project	string
FileVersion <i>optional</i>		Version
FinancialCalculatePerYear <i>optional</i>		boolean
FinancialRunOnServer <i>optional</i>		boolean
FinancialYears <i>optional</i>		integer
MaxID <i>optional</i>		integer (int64)

Name	Description	Schema
OpenTime <i>optional</i>		string (date-time)
ReadOnly <i>optional</i>		boolean
ReliabilityDuration <i>optional</i>		integer
ReliabilityRecoveryTimeUnit <i>optional</i>		string
ReliabilityRunOnServer <i>optional</i>		boolean
ReliabilityUnit <i>optional</i>		string
Standards <i>optional</i>		< CStandard > array
TechnopediaImportStrategy <i>optional</i>		string
TechnopediaImportTypeEED <i>optional</i>		integer (int64)
TechnopediaServerSyncFrequency <i>optional</i>	The frequency, per day, that Technopedia Sync runs automatically Minimum value : 0	integer
TechnopediaServerSyncTime <i>optional</i>	The base time (starting time) at which Technopedia Sync runs	string (time)
UseWindowsUsernameLogin <i>optional</i>		boolean
UserGroups <i>optional</i>		< CUserGroup > array

Name	Description	Schema
Users <i>optional</i>		< CUser > array

CParentConstraint

Polymorphism : Composition

Name	Description	Schema
ChildType <i>optional</i>		CElementType
ConstraintKind <i>optional</i>		ConstraintKind
EEID <i>optional</i> <i>read-only</i>	Minimum value : 1 Example : 1	integer (int64)
Enforcement Mode <i>optional</i>		ConstraintEnforceMode
Name <i>optional</i>		string
ParentType <i>optional</i>		CElementType

CProperty

Name	Schema
DataType <i>optional</i>	enum (Text, Integer, Decimal, DateTime, Boolean, List)
DataTypeEnforcement <i>optional</i>	enum (Disabled, Warned, Enforced)
Description <i>optional</i>	string
EEID <i>optional</i> <i>read-only</i>	integer (int64)
Name <i>optional</i>	string

Name	Schema
Type <i>optional</i>	string
Unit <i>optional</i>	string
Value <i>optional</i>	string

CStandard

Polymorphism : Composition

Name	Description	Schema
Description <i>optional</i>		string
EEID <i>optional</i> <i>read-only</i>	Minimum value : 1 Example : 1	integer (int64)
Name <i>optional</i>		string
Properties <i>optional</i>		< CProperty > array
SyncWithServiceNow <i>optional</i>		boolean
SyncWithTechnopedia <i>optional</i>		boolean

CTwoDComponent

Polymorphism : Composition

Name	Description	Schema
Angle <i>optional</i>		number (float)
Component <i>optional</i>		CComponent

Name	Description	Schema
EEID <i>optional</i> <i>read-only</i>	Minimum value : 1 Example : 1	integer (int64)
Height <i>optional</i>		integer
Width <i>optional</i>		integer
X <i>optional</i>		integer
Y <i>optional</i>		integer
ZOrder <i>optional</i>		integer

CTwoDConnection

Polymorphism : Composition

Name	Description	Schema
Angle <i>optional</i>		number (float)
Connection <i>optional</i>		CConnection
EEID <i>optional</i> <i>read-only</i>	Minimum value : 1 Example : 1	integer (int64)
Height <i>optional</i>		integer
Width <i>optional</i>		integer
X <i>optional</i>		integer
Y <i>optional</i>		integer
ZOrder <i>optional</i>		integer

CTwoDObject

Polymorphism : Composition

Name	Description	Schema
EEID <i>optional</i> <i>read-only</i>	Minimum value : 1 Example : 1	integer (int64)

CTwoDShape

Polymorphism : Composition

Name	Description	Schema
Angle <i>optional</i>		number (float)
EEID <i>optional</i> <i>read-only</i>	Minimum value : 1 Example : 1	integer (int64)
Height <i>optional</i>		integer
Width <i>optional</i>		integer
X <i>optional</i>		integer
Y <i>optional</i>		integer
ZOrder <i>optional</i>		integer

CTwoDTemplate

Polymorphism : Composition

Name	Description	Schema
ComponentStencilTypeMappings <i>optional</i>		< object > array

Name	Description	Schema
ConnectionStencilTypeMappings <i>optional</i>		< object > array
Description <i>optional</i>		string
EEID <i>optional</i> <i>read-only</i>	Minimum value : 1 Example : 1	integer (int64)
Name <i>optional</i>		string
Properties <i>optional</i>		< CProperty > array
SyncWithServiceNow <i>optional</i>		boolean
SyncWithTechnopedia <i>optional</i>		boolean

CTwoDVisualisation

Polymorphism : Composition

Name	Description	Schema
Architecture <i>optional</i>		CArchitecture
ComponentStencilTypeMappings <i>optional</i>		< object > array
ConnectionStencilTypeMappings <i>optional</i>		< object > array
Created <i>optional</i>		string (date-time)
CreatedBy <i>optional</i>		string

Name	Description	Schema
Description <i>optional</i>		string
EEID <i>optional</i> <i>read-only</i>	Minimum value : 1 Example : 1	integer (int64)
Name <i>optional</i>		string
Properties <i>optional</i>		< CProperty > array
ScrollX <i>optional</i>		integer
ScrollY <i>optional</i>		integer
SyncWithServiceNow <i>optional</i>		boolean
SyncWithTechnopedia <i>optional</i>		boolean
TwoDComponents <i>optional</i>		< CTwoDComponent > array
TwoDConnections <i>optional</i>		< CTwoDConnection > array
Zoom <i>optional</i>		integer

CTypeProperty

Polymorphism : Composition

Name	Schema
DataType <i>optional</i>	enum (Text, Integer, Decimal, DateTime, Boolean, List)
DataTypeEnforcement <i>optional</i>	enum (Disabled, Warned, Enforced)

Name	Schema
Description <i>optional</i>	string
EEID <i>optional</i> <i>read-only</i>	integer (int64)
Name <i>optional</i>	string
Required <i>optional</i>	boolean
RequiredEnforcement <i>optional</i>	enum (Disabled, Warned, Enforced)
Type <i>optional</i>	string
Unit <i>optional</i>	string
Value <i>optional</i>	string

CUndoObject

Name	Description	Schema
EEID <i>optional</i> <i>read-only</i>	Minimum value : 1 Example : 1	integer (int64)

CUser

Polymorphism : Composition

Name	Description	Schema
AEEditor <i>optional</i> <i>read-only</i>	Whether the user has AEEditor privileges if they are assigned to any user group that contains AEEditor turned on	boolean
AEModeler <i>optional</i> <i>read-only</i>	Whether the user has AEModeler privileges if they are assigned to any user group that contains AEModeler turned on	boolean

Name	Description	Schema
Admin <i>optional</i> <i>read-only</i>	Whether the user has Admin privileges if they are assigned to any user group that contains Admin turned on	boolean
ArchitectureSync <i>optional</i> <i>read-only</i>	Whether the user has Architecture Sync privileges if they are assigned to any user group that contains Architecture Sync turned on	boolean
Configurer <i>optional</i> <i>read-only</i>	Whether the user has Configurer privileges if they are assigned to any user group that contains Configurer turned on	boolean
EEID <i>optional</i> <i>read-only</i>	Minimum value : 1 Example : 1	integer (int64)
Email <i>optional</i>	The user's email address if applicable/known	string
IsDeleted <i>optional</i> <i>read-only</i>		boolean
IsSSOServiceAccount <i>optional</i>		boolean
LastLogin <i>optional</i>	The user's last recorded login date ('0001-01-01T00:00:00Z' if never logged in)	ReadOnlyDate
Location <i>optional</i>	The user's location if applicable/known	string
Name <i>optional</i>	The user's username (note that a username is unique within the object model) - only Admins can modify this	string
Password <i>optional</i>		string
PhoneNumber <i>optional</i>	The user's phone number if applicable/known	string
SSOIdentifier <i>optional</i>		string
Scripts <i>optional</i> <i>read-only</i>	Whether the user has Scripts privileges if they are assigned to any user group that contains Scripts turned on	boolean

Name	Description	Schema
ServiceNow <i>optional</i> <i>read-only</i>	Whether the user has ServiceNow privileges if they are assigned to any user group that contains ServiceNow turned on	boolean
Sharepoint <i>optional</i> <i>read-only</i>	Whether the user has Sharepoint privileges if they are assigned to any user group that contains Sharepoint turned on	boolean
Simulator <i>optional</i> <i>read-only</i>	Whether the user has Simulator privileges if they are assigned to any user group that contains Simulator turned on	boolean
Technopedia <i>optional</i> <i>read-only</i>	Whether the user has Technopedia privileges if they are assigned to any user group that contains Technopedia turned on	boolean

CUserGroup

Polymorphism : Composition

Name	Description	Schema
AEEditor <i>optional</i>	The user group's access to ABACUS Enterprise as Editor user - only Admins can modify this, also throws an exception if setting property would cause the maximum number of Editors to be exceeded	boolean
AEModeler <i>optional</i>	The user group's access to ABACUS Enterprise as Modeler user - only Admins can modify this, also throws an exception if setting property would cause the maximum number of Modelers to be exceeded	boolean
Admin <i>optional</i>		boolean
ArchitectureSync <i>optional</i>		boolean
AutoCommit <i>optional</i>	The user group's autocommit status - whether for Enterprise purposes, users of this group must autocommit their files - only Admins can modify this	boolean
Configurer <i>optional</i>		boolean
EEID <i>optional</i> <i>read-only</i>	Minimum value : 1 Example : 1	integer (int64)

Name	Description	Schema
IsDeleted <i>optional</i> <i>read-only</i>		boolean
Name <i>optional</i>	The user group's name (note that a username is unique within the object model) - only Admins can modify this	string
Scripts <i>optional</i>		boolean
ServiceNow <i>optional</i>		boolean
Sharepoint <i>optional</i>		boolean
Simulator <i>optional</i>		boolean
Technopedia <i>optional</i>		boolean
Users <i>optional</i> <i>read-only</i>	The collection of the users that are part of this user group	< CUser > array

CUserGroupPatch

Polymorphism : Composition

Name	Description	Schema
AEEditor <i>optional</i>	The user group's access to ABACUS Enterprise as Editor user - only Admins can modify this, also throws an exception if setting property would cause the maximum number of Editors to be exceeded	boolean
AEModeler <i>optional</i>	The user group's access to ABACUS Enterprise as Modeler user - only Admins can modify this, also throws an exception if setting property would cause the maximum number of Modelers to be exceeded	boolean
Admin <i>optional</i>		boolean
ArchitecturesSync <i>optional</i>		boolean

Name	Description	Schema
AutoCommit <i>optional</i>	The user group's autocommit status - whether for Enterprise purposes, users of this group must autocommit their files - only Admins can modify this	boolean
Configurer <i>optional</i>		boolean
Name <i>optional</i>	The user group's name (note that a username is unique within the object model) - only Admins can modify this	string
Scripts <i>optional</i>		boolean
ServiceNow <i>optional</i>		boolean
Sharepoint <i>optional</i>		boolean
Simulator <i>optional</i>		boolean
Technopedia <i>optional</i>		boolean

CUserPatch

Polymorphism : Composition

Name	Description	Schema
Email <i>optional</i>	The user's email address if applicable/known	string
IsSSOServiceAccount <i>optional</i>		boolean
Location <i>optional</i>	The user's location if applicable/known	string
Name <i>optional</i>	The user's username (note that a username is unique within the object model) - only Admins can modify this	string
Password <i>optional</i>		string
PhoneNumber <i>optional</i>	The user's phone number if applicable/known	string

Name	Description	Schema
SSOIdentifier <i>optional</i>		string

ConstraintConnectionRole

Type : enum (Source, Sink, Any)

ConstraintEnforceMode

Type : enum (Disabled, Warned, Enforced)

ConstraintKind

Type : enum (ParentConstraint, ChildConstraint, ComponentAttachedConstraint, ConnectionAttachedConstraint)

PropertyDataType

Type : enum (Text, Integer, Decimal, DateTime, Boolean, List)

ReadOnlyDate

Type : string (date-time)

Version

Name	Schema
Build <i>optional</i>	integer (int32)
Major <i>optional</i>	integer (int32)
Minor <i>optional</i>	integer (int32)
Revision <i>optional</i>	integer (int32)