Reference number
ISO/WD 44001

ISO/TC 184/SC 4
Date: 2026-02-03
Stage: 20.00 (Working Draft)

Industrial automation systems and integration — OpenCAD Interchange Standard (OCIS)

Part 1: Core Architecture and File Formats

Contents

1 Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees.

This document was prepared by Technical Committee ISO/TC 184, **Automation systems and integration**, Subcommittee SC 4, **Industrial data**.

2 Introduction

The exchange of 3D engineering data has historically been hindered by the "Silo Effect" of proprietary file formats. Existing neutral formats (e.g., ISO 10303 STEP) successfully transfer boundary representation (B-Rep) geometry but fail to preserve the "design intent" or parametric history (the "Change Tree").

The OpenCAD Interchange Standard (OCIS) defines a set of cloud-native file formats designed to preserve this parametric history and facilitate interoperability between disparate Computer-Aided Design (CAD) and Electronic Computer-Aided Design (ECAD) software suites. It leverages the WebDAV protocol to ensure data integrity in cloud storage environments.

3 1 Scope

This document specifies the logical structure, syntax, and semantic definitions for the OpenCAD Interchange Standard (OCIS).

It covers:

1. The definition of the three core file types: Part (`.ocp`), Electrical (`.oce`), and Assembly (`.oca`).
2. The schema for the Unified Change Tree (UCT), enabling lossless transfer of parametric modeling operations.
3. The mapping of OCIS metadata to WebDAV properties for cloud integration.
4. The definition of logical constraints and relationships between file types.

It does not cover:

1. The specific algorithms used by geometric modeling kernels to solve the features defined in the UCT.
2. Visual rendering protocols (e.g., shaders, lighting).

4 2 Normative References

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document.

ISO/IEC 21778:2017, Information technology — The JSON data interchange syntax RFC 4918, **HTTP Extensions for Web Distributed Authoring and Versioning (WebDAV) ISO 10303-21 , Industrial automation systems and integration — Product data representation and exchange — Part 21: Implementation methods: Clear text encoding of the exchange structure**

5 3 Terms and Definitions

Change Tree: Ordered sequence of parametric operations (features) used to construct a 3D geometric model.

Feature: A single parametric operation, such as an extrude, revolve, fillet, or chamfer.

Unified Change Tree (UCT): The normative JSON schema defined in this standard for representing a Change Tree in a software-agnostic format.

WebDAV: Set of extensions to the HTTP protocol which allows users to collaboratively edit and manage files on remote web servers.

6 4 General Architecture

6.1 4.1 The Tri-File Structure

The OCIS architecture decouples engineering data into three distinct semantic domains. This separation ensures that specialised tools (e.g., PCB design software) can interact with the data model without needing to parse unrelated data (e.g., mechanical geometry).

The three file types are:
1. OpenCAD Part (`.ocp`): Contains geometric definitions, material properties, and the UCT.
2. OpenCAD Electrical (`.oce`): Contains schematic logic, netlists, and pin definitions.
3. OpenCAD Assembly (`.oca`): Contains hierarchical references, constraints, and bill of materials (BOM) data.

6.2 4.2 Encoding

All OCIS files shall be encoded using UTF-8. The core data structure for all file types shall be strictly compliant with ISO/IEC 21778 (JSON).

7 5 File Specifications

7.1 5.1 The Part File (.ocp)

5.1.1 Overview The `.ocp` file represents a single rigid body or a multi-body part defined by a unified history of operations.

5.1.2 Root Structure The root JSON object of an `.ocp` file shall contain the following keys:

```
{
  "header": { "version": "1.0", "generator": "Software_Name_vX" },
  "metadata": { "uuid": "...", "mass_units": "kg", "length_units": "mm" },
  "uct": { ... }, // Unified Change Tree
  "cache": { ... } // Optional B-Rep Cache (Base64 encoded STEP)
}
```

5.1.3 The Unified Change Tree (UCT) The `uct` object contains an ordered array of operations. Compliant software parsers must execute these operations in order to reconstruct the geometry.

Normative Operation Types: **`EXTRUDE`: Linear projection of a 2D sketch profile.** `REVOLVE`: Rotational projection of a 2D sketch profile around an axis. **`FILLET`: Rounding of a specific edge reference.** `BOOLEAN`: Union, Difference, or Intersection of bodies.

Example UCT Node:

```
{
  "op_id": "feat_001",
  "type": "EXTRUDE",
  "input": {
    "sketch_ref": "sk_001",
```

```
    "distance": 15.0,
    "direction": [0, 0, 1],
    "taper_angle": 0.0
  }
}
```

7.2 5.2 The Electrical File (.oce)

5.2.1 Overview The `.oce` file contains logical electrical data. It acts as the bridge between 2D schematics and 3D routing.

5.2.2 Connectivity Model The `.oce` file shall define connectivity using a `netlist` array.

```
{
  "components": [
    { "ref_des": "R1", "part_number": "RES-10K", "footprint": "0603" }
  ],
  "nets": [
    { "name": "VCC", "nodes": ["R1:1", "U1:8"] }
  ]
}
```

7.3 5.3 The Assembly File (.oca)

5.3.1 Overview The `.oca` file contains no geometry. It links `.ocp` and `.oce` files via relative or absolute paths (URI).

5.3.2 Constraint Definitions Constraints define the kinematic relationship between components.

Supported Constraints: `MATE`: Coincident planes or faces. `ALIGN`: Coaxial axes. **`OFFSET`: Distance between entities.**

```
{
  "instances": [
    { "id": "inst_01", "source": "./bracket.ocp" },
    { "id": "inst_02", "source": "./sensor.oce" }
  ],
  "constraints": [
    {
      "type": "MATE",
      "target_a": "inst_01:face_top",
      "target_b": "inst_02:face_btm"
    }
  ]
}
```

8 6 Cloud Integration (WebDAV Binding)

8.1 6.1 Objective

OCIS files are designed to reside on WebDAV-compliant servers. This section defines the mapping between OCIS internal metadata and WebDAV dead properties.

8.2 6.2 Property Mapping

Compliant servers and clients shall map internal JSON metadata to WebDAV XML properties in the `ocis:` namespace.

OCIS JSON Field	WebDAV Property	Description
metadata.uuid	ocis:uuid	Unique Identifier
metadata.material	ocis:material	Material Name
metadata.mass	ocis:mass	Calculated Mass
header.generator	ocis:generator	Software used to save

8.3 6.3 Concurrency Control

To prevent "last-write-wins" data loss:

1. Open: Clients MUST issue a WebDAV `LOCK` request on the target URI before beginning an edit session.
2. Save: Clients MUST include the `If` header containing the Lock-Token during `PUT` operations.
3. Close: Clients MUST issue an `UNLOCK` request upon closing the file.

9 Annex A (Normative): JSON Schemas

(Note: In the full standard, the complete JSON schemas would appear here. For this draft, a placeholder is provided.)

9.1 A.1 Common Header Schema

```
{
  "$schema": "[http://json-schema.org/draft-07/schema](http://json-schema.org/draft-07/schema)",
  "type": "object",
  "properties": {
    "version": { "type": "string", "pattern": "^\\d+\\.\\d+$" },
    "generator": { "type": "string" }
  },
  "required": ["version", "generator"]
}
```