

# Zotero to MODS mapping

## Zotero to MODS mapping

- This is the doExport function for the MODS translator in [scrapers.sql](#)
  - Use [Scaffold](#) to test changes

```
function doExport() {
    Zotero.setCharacterSet("utf-8");
    var modsCollection = <modsCollection xmlns="http://www.loc.gov/mods/v3" xmlns:xsi="http://www.w3.org
/2001/XMLSchema-instance" xsi:schemaLocation="http://www.loc.gov/mods/v3 http://www.loc.gov/standards/mods/v3
/mods-3-2.xsd" />;

    var item;
    while(item = Zotero.nextItem()) {
        var isPartialItem = Zotero.Utilities.inArray(item.itemType, partialItemTypes);

        var mods = <mods />;

        /** CORE FIELDS **/

        // XML tag titleInfo; object field title
        if(item.title) {
            mods.titleInfo.title = item.title;
        }

        // XML tag typeOfResource/genre; object field type
        var modsType, marcGenre;
        if(item.itemType == "book" || item.itemType == "bookSection") {
            modsType = "text";
            marcGenre = "book";
        } else if(item.itemType == "journalArticle" || item.itemType == "magazineArticle") {
            modsType = "text";
            marcGenre = "periodical";
        } else if(item.itemType == "newspaperArticle") {
            modsType = "text";
            marcGenre = "newspaper";
        } else if(item.itemType == "thesis") {
            modsType = "text";
            marcGenre = "theses";
        } else if(item.itemType == "letter") {
            modsType = "text";
            marcGenre = "letter";
        } else if(item.itemType == "manuscript") {
            modsType = "text";
            modsType.@manuscript = "yes";
        } else if(item.itemType == "interview") {
            modsType = "text";
            marcGenre = "interview";
        } else if(item.itemType == "film") {
            modsType = "moving image";
            marcGenre = "motion picture";
        } else if(item.itemType == "artwork") {
            modsType = "still image";
            marcGenre = "art original";
        } else if(item.itemType == "webpage") {
            modsType = "multimedia";
            marcGenre = "web site";
        } else if(item.itemType == "note" || item.itemType == "attachment") {
            continue;
        }
        mods.typeOfResource = modsType;
        mods.genre += <genre authority="local">{item.itemType}</genre>;
        if(marcGenre) {
            mods.genre += <genre authority="marcgt">{marcGenre}</genre>;
        }

        // XML tag genre; object field thesisType, type
```

```

if(item.thesisType) {
    mods.genre += <genre>{item.thesisType}</genre>;
}
if(item.type) {
    mods.genre += <genre>{item.type}</genre>;
}

// XML tag name; object field creators
for(var j in item.creators) {
    var roleTerm = "";
    if(item.creators[j].creatorType == "author") {
        roleTerm = "aut";
    } else if(item.creators[j].creatorType == "editor") {
        roleTerm = "edt";
    } else if(item.creators[j].creatorType == "creator") {
        roleTerm = "ctb";
    }

    // FIXME - currently all names are personal
    mods.name += <name type="personal">
        <namePart type="family">{item.creators[j].lastName}</namePart>
        <namePart type="given">{item.creators[j].firstName}</namePart>
        <role><roleTerm type="code" authority="marcrelator">{roleTerm}</roleTerm></role>
    </name>;
}

// XML tag recordInfo.recordOrigin; used to store our generator note
//mods.recordInfo.recordOrigin = "Zotero for Firefox "+Zotero.Utilities.getVersion();

/** FIELDS ON NEARLY EVERYTHING BUT NOT A PART OF THE CORE **/

// XML tag recordInfo.recordContentSource; object field source
if(item.source) {
    mods.recordInfo.recordContentSource = item.source;
}

// XML tag recordInfo.recordIdentifier; object field accessionNumber
if(item.accessionNumber) {
    mods.recordInfo.recordIdentifier = item.accessionNumber;
}

// XML tag accessCondition; object field rights
if(item.rights) {
    mods.accessCondition = item.rights;
}

/** SUPPLEMENTAL FIELDS **/

// Make part its own tag so we can figure out where it goes later
var part = new XML();

// XML tag detail; object field volume
if(item.volume) {
    if(Zotero.Utilities.isInt(item.volume)) {
        part += <detail type="volume"><number>{item.volume}</number></detail>;
    } else {
        part += <detail type="volume"><text>{item.volume}</text></detail>;
    }
}

// XML tag detail; object field number
if(item.issue) {
    if(Zotero.Utilities.isInt(item.issue)) {
        part += <detail type="issue"><number>{item.issue}</number></detail>;
    } else {
        part += <detail type="issue"><text>{item.issue}</text></detail>;
    }
}

// XML tag detail; object field section
if(item.section) {
    if(Zotero.Utilities.isInt(item.section)) {

```

```

        part += <detail type="section"><number>{item.section}</number></detail>;
    } else {
        part += <detail type="section"><text>{item.section}</text></detail>;
    }
}

// XML tag detail; object field pages
if(item.pages) {
    var range = Zotero.Utilities.getPageRange(item.pages);
    part += <extent unit="pages"><start>{range[0]}</start><end>{range[1]}</end></extent>;
}

// Assign part if something was assigned
if(part.length() != 1) {
    if(isPartialItem) {
        // For a journal article, bookSection, etc., the part is the host
        mods.relatedItem.part += <part>{part}</part>;
    } else {
        mods.part += <part>{part}</part>;
    }
}

// XML tag originInfo; object fields edition, place, publisher, year, date
var originInfo = new XML();
if(item.edition) {
    originInfo += <edition>{item.edition}</edition>;
}
if(item.place) {
    originInfo += <place><placeTerm type="text">{item.place}</placeTerm></place>;
}
if(item.publisher) {
    originInfo += <publisher>{item.publisher}</publisher>;
} else if(item.distributor) {
    originInfo += <publisher>{item.distributor}</publisher>;
}
if(item.date) {
    if(Zotero.Utilities.inArray(item.itemType, ["book", "bookSection"])) {
        // Assume year is copyright date
        var dateType = "copyrightDate";
    } else if(Zotero.Utilities.inArray(item.itemType, ["journalArticle", "magazineArticle",
"newspaperArticle"])) {
        // Assume date is date issued
        var dateType = "dateIssued";
    } else {
        // Assume date is date created
        var dateType = "dateCreated";
    }
    var tag = <{dateType}>{item.date}</{dateType}>;
    originInfo += tag;
}
if(item.accessDate) {
    originInfo += <dateCaptured>{item.accessDate}</dateCaptured>;
}
if(originInfo.length() != 1) {
    if(isPartialItem) {
        // For a journal article, bookSection, etc., this goes under the host
        mods.relatedItem.originInfo += <originInfo>{originInfo}</originInfo>;
    } else {
        mods.originInfo += <originInfo>{originInfo}</originInfo>;
    }
}

// XML tag identifier; object fields ISBN, ISSN
if(isPartialItem) {
    var identifier = mods.relatedItem;
} else {
    var identifier = mods;
}
if(item.ISBN) {
    identifier.identifier += <identifier type="isbn">{item.ISBN}</identifier>;
}

```

```

if(item.ISSN) {
    identifier.identifier += <identifier type="issn">{item.ISSN}</identifier>;
}
if(item.DOI) {
    mods.identifier += <identifier type="doi">{item.DOI}</identifier>;
}

// XML tag relatedItem.titleInfo; object field publication
if(item.publicationTitle) {
    mods.relatedItem.titleInfo += <titleInfo><title>{item.publicationTitle}</title><
/titleInfo>;
}

// XML tag classification; object field callNumber
if(item.callNumber) {
    mods.classification = item.callNumber;
}

// XML tag location.physicalLocation; object field archiveLocation
if(item.archiveLocation) {
    mods.location.physicalLocation = item.archiveLocation;
}

// XML tag location.url; object field archiveLocation
if(item.url) {
    mods.location.url = item.url;
}

// XML tag title.titleInfo; object field journalAbbreviation
if(item.journalAbbreviation) {
    mods.relatedItem.titleInfo += <titleInfo type="abbreviated"><title>{item.
journalAbbreviation}</title></titleInfo>;
}

// XML tag abstract; object field abstractNote
if(item.abstractNote) {
    mods.abstract = item.abstractNote;
}

if(mods.relatedItem.length() == 1 && isPartialItem) {
    mods.relatedItem.@type = "host";
}

/** NOTES */

if(Zotero.getOption("exportNotes")) {
    for(var j in item.notes) {
        // Add note tag
        var note = <note type="content">{item.notes[j].note}</note>;
        mods.note += note;
    }
}

/** TAGS */

for(var j in item.tags) {
    mods.subject += <subject><topic>{item.tags[j].tag}</topic></subject>;
}

// XML tag relatedItem.titleInfo; object field series
if(item.seriesTitle || item.series || item.seriesNumber || item.seriesText) {
    var series = <relatedItem type="series"/>;

    if(item.series) {
        series.titleInfo.title = item.series;
    }

    if(item.seriesTitle) {
        series.titleInfo.partTitle = item.seriesTitle;
    }
}

```

```
        if(item.seriesText) {
            series.titleInfo.subTitle = item.seriesText;
        }

        if(item.seriesNumber) {
            series.titleInfo.partNumber = item.seriesNumber;
        }

        // TODO: make this work in import
        /*if(item.itemType == "bookSection") {
            // For a book section, series info must go inside host tag
            mods.relatedItem.relatedItem = series;
        } else {*/
            mods.relatedItem += series;
        //}
    }

    modsCollection.mods += mods;
}

Zotero.write('<?xml version="1.0"?>'+'\n');
Zotero.write(modsCollection.toXMLString());
}
```