

## Update packages with datapack

Once you have updated the data objects and saved the metadata to a file, we can update the metadata and add the new pid to the resource map using `datapack::updateDataObject()`.

### Note

This part of the training assumes that you have an EML file created from earlier parts of the training

### Warning

You must use these R packages for data packages that have provenance or folder hierarchy (`arcticdatatutils` does not support those features)

## Get the package

Setting the node is done slightly differently:

```
d1c_test <- dataone::D1Client("STAGING", "urn:node:mnTestARCTIC")
```

Get a pre-existing package:

```
packageId <- "the resource map"

dp <- getDataPackage(d1c_test, identifier=packageId, lazyLoad=TRUE, quiet=FALSE)
```

## Update the metadata

Get the metadata identifier:

```
metadataId <- selectMember(dp, name="sysmeta@formatId", value="https://eml.ecoinformatics.org/eml-2.1.1")
```

### Note

Take note of the EML version. If it is EML 2.1.1 the value needs to be changed to `-"eml://ecoinformatics.org/eml-2.1.1"`

Read the EML doc:

```
doc <- read_eml(getObject(d1c@mn, metadataId))
```

Edit the EML as usual - see the documents in Edit EML for details

Once you are happy with your changes, you can update your data package to include the new metadata file using `replaceMember`:

```
eml_path <- "path/to/your/saved/eml.xml"
write_eml(doc, eml_path)

dp <- replaceMember(dp, metadataId, replacement=eml_path)
```

## Modify the data files

remove zip files:

```
zipId <- selectMember(dp, name="sysmeta@formatId", value="application/vnd.shp+zip")
removeMember(dp, zipId, removeRelationships = T)
```

add an existing data object:

```
dataObj <- getDataObject(d1c_test, id="urn:uuid: here", lazyLoad=T, limit="1TB")
dp <- addMember(dp, dataObj, mo=metadataId)
```

### Note

If you want to change the formatId please use `updateSystemMetadata`

## Publish update

### Using a DOI

If you want to publish with a DOI (Digital Object Identifier) instead of a UUID (Universally Unique Identifier), (a) you need to change the `public` argument to `TRUE` and generate a DOI identifier.

**This should only be done after the package is finalized and has been thoroughly reviewed!**

- (1) Updating a package with a new DOI:

A Digital Object Identifier (DOI) may be assigned to the metadata **DataObject**, using the `generateIdentifier`:

```
doi <- dataone::generateIdentifier(d1c_test@mn, "DOI")
dp <- replaceMember(dp, metadataId, replacement=eml_path, newId=doi)

newPackageId <- uploadDataPackage(d1c_test, dp, public=TRUE, quiet=FALSE)
```

(2) Updating a package with a pre-issued DOI:

```
dp <- replaceMember(dp, metadataId, replacement=eml_path, newId="your pre-issued doi previ

newPackageId <- uploadDataPackage(d1c_test, dp, public=TRUE, quiet=FALSE)
```

Refresh the landing page at [test.arcticdata.io/#view/...](http://test.arcticdata.io/#view/...) for this package and then follow the “newer version” link to view the latest.