

ModelOp Center v2.x: Model Metadata

SEPTEMBER 2020



DOCUMENT PURPOSE

The following document provides an overview of the core metadata stored within ModelOp Center.

TABLE OF CONTENTS

<u>DOCUMENT PURPOSE</u>	1
<u>BACKGROUND</u>	2
<u>DEFINITIONS</u>	2
<u>CORE METADATA ELEMENTS</u>	2
<u>EXAMPLE OF A STORED MODEL</u>	3
<u>CUSTOM METADATA</u>	9
OVERVIEW	9
ADDING/MODIFYING CUSTOM METADATA	9

Background

ModelOp Center enables consistent monitoring, governance, and automated orchestration of all models, regardless of the model factory in which it was developed, the language/framework used, or the platform in which the model executes. The foundation behind this capability is ModelOp Center's "standard model definition", which provides a common abstracted definition of a model across all languages, frameworks, factories, and execution environments. This document provides an overview of the core elements of the standard model definition. ModelOp Center captures and persists the "model metadata" that comprises the standard model definition with each version of the model.

Definitions

The following are several foundational elements within ModelOp Center:

- **Stored Model:**
 - Stored Models are works in progress that represent the latest information about the model: source code, metadata, assets, etc.
- **Deployable Model:**
 - Typically referred to as a "Version", Deployable Models are a snapshot of the StoredModel at a specific point in time
 - Captured for all batch jobs that are run, so we can see the exact revision of the model that was used
- **Deployed Model:**
 - These are deployable models that have actually been installed in the business
 - Contains the deployable model, but also full lineage of the lifecycle(s) followed to be installed

Core Metadata Elements

The following is the list of the primary elements of the core metadata for a StoredModel:

- [storedModel] "id"
- "group"
- "createdDate"
- "lastModifiedDate"
- "createdBy"
- "lastModifiedBy"
- "modelMetaData"
 - "custom": < **Customer-specific metadata for the storedModel** >
 - "tags": <list of tags>
 - "name"
 - "description"
 - "type"
 - "inputSchema": <with schema specific metadata>
 - "outputSchema": <with schema specific metadata>
 - "enableInputSchemaChecking"
 - "enableOutputSchemaChecking"
 - "attachments"
 - "platform": <list of required libraries/dependencies>
 - "repositoryInfo": <with git repository metadata>
 - "version":

- "modelAssets": <list of asset specific metadata>
- "gitDifferences"
- "associatedModels"

The deployableModel adds in the following elements in addition to a StoredModel's metadata:

- [deployableModel] "id"
- "group"
- "createdDate"
- "lastModifiedDate"
- "createdBy"
- "lastModifiedBy"
- "modelMLCs": <details of each process instance for each mlc>
- "metaData": <**Customer-specific metadata for the deployableModel** >
- "associatedModels": <list of associated models for the deployableModel>
- "modelDocuments": <list of model documents for the deployableModel >

The deployedModel adds in the following elements in addition to a DeployableModel's metadata:

- [deployedModel] "id"
- "group"
- "createdDate"
- "lastModifiedDate"
- "createdBy"
- "lastModifiedBy"
- "deployTime"
- "engine": <model execution/runtime-specific metadata >
- "state": <current state of the deployment>
- "metaData": <**Customer-specific metadata for the deployedModel** >

Example of a Stored Model

The following is an example of a StoredModel:

```
{
  "id": "f7bfb31f-15a4-4d0f-adba-3163396174f8",
  "group": "modelop",
  "createdDate": "2020-09-02T22:20:56.132Z",
  "lastModifiedDate": "2020-09-02T22:24:03.180Z",
  "createdBy": null,
  "lastModifiedBy": "SYSTEM",
  "modelMetaData": {
    "custom": null,
    "tags": [
      "ecomm-surveillance"
    ],
    "name": "enron_ecomm_lasso_demo",
    "description": "Imported from a GIT repository",
    "type": "PYTHON",
    "classicModel": false,
    "inputSchema": [
      {
        "schemaId": "5f5652b9-094d-451a-af52-040af42ecd9d",
        "name": "input_schema.avsc",
        "schemaDefinition": {
          "type": "record",
          "name": "email",
          "fields": [
```

```

    {
      "name": "id",
      "type": "int"
    },
    {
      "name": "content",
      "type": "string"
    }
  ]
}
],
"outputSchema": [
  {
    "schemaId": "b8133829-af53-490c-8203-95a48c86cf70",
    "name": "output_schema.avsc",
    "schemaDefinition": {
      "type": "record",
      "name": "email",
      "fields": [
        {
          "name": "id",
          "type": "int"
        },
        {
          "name": "content",
          "type": "string"
        },
        {
          "name": "prediction",
          "type": "int"
        }
      ]
    }
  }
],
"enableInputSchemaChecking": false,
"enableOutputSchemaChecking": false,
"attachments": [],
"platform": {
  "python": {
    "libraries": {
      "zipp": "3.1.0",
      "zict": "2.0.0",
      "xlrd": "1.2.0",
      "widgetsnbextension": "3.5.1",
      "wheel": "0.35.1",
      "webencodings": "0.5.1",
      "wewidth": "0.2.5",
      "vincent": "0.4.4",
      "urllib3": "1.25.10",
      "typing-extensions": "3.7.4.2",
      "traitlets": "4.3.3",
      "tqdm": "4.48.2",
      "tornado": "6.0.4",
      "toolz": "0.10.0",
      "tiffle": "2020.8.13",
      "threadpoolctl": "2.1.0",
      "testpath": "0.4.4",
      "terminado": "0.8.3",
      "tblib": "1.6.0",
      "tables": "3.6.1",
      "sympy": "1.6.2",
      "statsmodels": "0.11.1",
      "SQLAlchemy": "1.3.19",
      "soupsieve": "2.0.1",
      "sortedcontainers": "2.2.2",
      "six": "1.15.0",
      "setuptools": "49.6.0.post20200814",
      "Send2Trash": "1.5.0",
      "seaborn": "0.10.1",
      "scipy": "1.5.2",
      "scikit-learn": "0.23.2",
      "scikit-image": "0.17.2",
      "ruamel.yaml": "0.16.10",
      "ruamel.yaml.clib": "0.2.0",

```

```
"ruamel-yaml": "0.15.80",
"requests": "2.24.0",
"pyzmq": "19.0.2",
"PyYAML": "5.3.1",
"PyWavelets": "1.1.1",
"pytz": "2020.1",
"python-json-logger": "0.1.11",
"python-editor": "1.0.4",
"python-dateutil": "2.8.1",
"PySocks": "1.7.1",
"pysistent": "0.16.0",
"pyparsing": "2.4.7",
"pyOpenSSL": "19.1.0",
"PyJWT": "1.7.1",
"Pygments": "2.6.1",
"pycurl": "7.43.0.5",
"pyparser": "2.20",
"pycosat": "0.6.3",
"ptyprocess": "0.6.0",
"psutil": "5.7.2",
"protobuf": "3.12.4",
"prompt-toolkit": "3.0.6",
"prometheus-client": "0.8.0",
"pip": "20.2.2",
"Pillow": "7.2.0",
"pickleshare": "0.7.5",
"pexpect": "4.8.0",
"patsy": "0.5.1",
"partd": "1.1.0",
"parso": "0.7.1",
"pandocfilters": "1.4.2",
"pandas": "1.1.0",
"pamela": "1.0.0",
"packaging": "20.4",
"olefile": "0.46",
"oauthlib": "3.0.1",
"numpy": "1.19.1",
"numexpr": "2.7.1",
"numba": "0.48.0",
"notebook": "6.1.3",
"networkx": "2.4",
"nbformat": "5.0.7",
"nbconvert": "5.6.1",
"msgpack": "1.0.0",
"mpmath": "1.1.0",
"mock": "4.0.2",
"mistune": "0.8.4",
"matplotlib": "3.2.2",
"MarkupSafe": "1.1.1",
"Mako": "1.1.3",
"lxml": "4.5.2",
"locket": "0.2.0",
"llvmlite": "0.31.0",
"kiwisolver": "1.2.0",
"jupyterlab": "2.2.5",
"jupyterlab-server": "1.2.0",
"jupyterhub": "1.1.0",
"jupyter-telemetry": "0.0.5",
"jupyter-nbextensions-configurator": "0.4.1",
"jupyter-latex-envs": "1.4.6",
"jupyter-highlight-selected-word": "0.2.0",
"jupyter-core": "4.6.3",
"jupyter-contrib-nbextensions": "0.5.1",
"jupyter-contrib-core": "0.3.3",
"jupyter-client": "6.1.6",
"jsonschema": "3.2.0",
"json5": "0.9.4",
"joblib": "0.16.0",
"Jinja2": "2.11.2",
"jedi": "0.17.2",
"ipywidgets": "7.5.1",
"ipython": "7.17.0",
"ipython-genutils": "0.2.0",
"ipympl": "0.5.7",
"ipykernel": "5.3.4",
"importlib-metadata": "1.7.0",
```

```

"imageio": "2.9.0",
"imagecodecs": "2020.5.30",
"idna": "2.10",
"HeapDict": "1.0.1",
"h5py": "2.10.0",
"gmpy2": "2.1.0b1",
"fsspec": "0.8.0",
"fastcache": "1.1.0",
"entrypoints": "0.3",
"distributed": "2.23.0",
"dill": "0.3.2",
"defusedxml": "0.6.0",
"decorator": "4.4.2",
"dask": "2.20.0",
"cytoolz": "0.10.1",
"Cython": "0.29.21",
"cycler": "0.10.0",
"cryptography": "3.0",
"conda": "4.8.3",
"conda-package-handling": "1.6.1",
"cloudpickle": "1.5.0",
"click": "7.1.2",
"chardet": "3.0.4",
"cffib": "1.14.1",
"certipy": "0.1.3",
"certifi": "2020.6.20",
"brotlipy": "0.7.0",
"Bottleneck": "1.3.2",
"bokeh": "2.1.1",
"blinker": "1.4",
"bleach": "3.1.5",
"beautifulsoup4": "4.9.1",
"backports.functools-lru-cache": "1.6.1",
"backcall": "0.2.0",
"attrs": "19.3.0",
"async-generator": "1.10",
"argon2-cffi": "20.1.0",
"alembic": "1.4.2"
},
"system": {
"executable": "/opt/conda/bin/python",
"float_info": [
1.7976931348623157e+308,
1024,
308,
2.2250738585072014e-308,
-1021,
-307,
15,
53,
2.220446049250313e-16,
2,
1
],
"version": "3.8.5 | packaged by conda-forge | (default, Jul 31 2020, 02:39:48) \n[GCC 7.5.0]",
"api_version": 1013,
"version_info": [
3,
8,
5,
"final",
0
],
"platform": "linux",
"architecture": [
"64bit",
""
],
"processor": "x86_64",
"system": "Linux",
"num_processors": 4
}
},
"repositoryInfo": {
"classicRepository": false,

```

```

"repositoryType": "GIT_REPO",
"repositoryDirectory": "/tmp/model-manage-repos/master/f7bfb31f-15a4-4d0f-adba-3163396174f8",
"repositoryRemote": "https://github.com/modelop/enron_ecomm_lasso_demo",
"repositoryBranch": "master",
"lastCommitId": "921625129157310de26c272d77e20a5f56d7feac"
},
"version": null
},
"modelAssets": [
{
"assetId": "7cda7a5b-3494-3ae9-bca4-a12560a842f9",
"name": "enron_ecomm_lasso_metrics.py",
"assetType": "SOURCE_CODE",
"assetRole": "MODEL_SOURCE",
"metaData": null,
"repositoryInfo": {
"classicRepository": false,
"repositoryType": "GIT_REPO",
"repositoryDirectory": "/tmp/model-manage-repos/master/f7bfb31f-15a4-4d0f-adba-3163396174f8",
"repositoryRemote": "https://github.com/modelop/enron_ecomm_lasso_demo",
"repositoryBranch": "master",
"lastCommitId": "ed2450a13c317173303a40e4254e5f774619b8bf"
},
"importedRepository": {
"classicRepository": false,
"repositoryType": "GIT_REPO",
"repositoryDirectory": "/tmp/model-manage-repos/master/f7bfb31f-15a4-4d0f-adba-3163396174f8",
"repositoryRemote": "https://github.com/modelop/enron_ecomm_lasso_demo",
"repositoryBranch": "master",
"lastCommitId": "921625129157310de26c272d77e20a5f56d7feac"
},
"primaryModelSource": true,
"sourceCode": "",
"sourceCodeFilePath": "enron_ecomm_lasso_metrics.py",
"sourceCodeType": "MODEL",
"initFunction": "begin",
"scoreFunction": "action",
"metricsFunction": "metrics",
"trainingFunction": null
},
{
"assetId": "32d1e91e-8b31-36e1-8979-5ab21557f475",
"name": "example_comparator.dmn",
"assetType": "SOURCE_CODE",
"assetRole": "TEST_RESULT_COMPARATOR",
"metaData": null,
"repositoryInfo": {
"classicRepository": false,
"repositoryType": "GIT_REPO",
"repositoryDirectory": "/tmp/model-manage-repos/master/f7bfb31f-15a4-4d0f-adba-3163396174f8",
"repositoryRemote": "https://github.com/modelop/enron_ecomm_lasso_demo",
"repositoryBranch": "master",
"lastCommitId": "ed2450a13c317173303a40e4254e5f774619b8bf"
},
"importedRepository": {
"classicRepository": false,
"repositoryType": "GIT_REPO",
"repositoryDirectory": "/tmp/model-manage-repos/master/f7bfb31f-15a4-4d0f-adba-3163396174f8",
"repositoryRemote": "https://github.com/modelop/enron_ecomm_lasso_demo",
"repositoryBranch": "master",
"lastCommitId": "921625129157310de26c272d77e20a5f56d7feac"
},
"primaryModelSource": false,
"sourceCode": "",
"sourceCodeFilePath": "example_comparator.dmn",
"sourceCodeType": "MODEL",
"initFunction": null,
"scoreFunction": null,
"metricsFunction": null,
"trainingFunction": null
},
{
"assetId": "8c47705a-0f22-435b-8d7e-9c11eea1a99e",
"name": "f7bfb31f-15a4-4d0f-adba-3163396174f8/8c47705a-0f22-435b-8d7e-9c11eea1a99e.json",
"assetType": "EXTERNAL_FILE",
"assetRole": "TEST_DATA",

```



```

"metaData": null,
"repositoryInfo": {
  "secretKey": "",
  "accessKey": "",
  "secure": false,
  "host": "minio",
  "defaultBucket": "modelop",
  "port": 9000,
  "region": "default-region",
  "repositoryType": "S3_REPO"
},
"importedRepository": {
  "classicRepository": false,
  "repositoryType": "GIT_REPO",
  "repositoryDirectory": "/tmp/model-manage-repos/master/f7bfb31f-15a4-4d0f-adba-3163396174f8",
  "repositoryRemote": "https://github.com/modelop/enron_ecomm_lasso_demo",
  "repositoryBranch": "master",
  "lastCommitId": "921625129157310de26c272d77e20a5f56d7feac"
},
"fileUrl": "http://minio:9000/modelop/f7bfb31f-15a4-4d0f-adba-3163396174f8/8c47705a-0f22-435b-8d7e-9c11eea1a99e.json",
"filename": "enron_test_validated.json",
"fileFormat": "JSON",
"fileSize": 1865172
},
{
  "assetId": "66d08945-4d9b-4767-bdc4-1ec60ea8cf05",
  "name": "f7bfb31f-15a4-4d0f-adba-3163396174f8/66d08945-4d9b-4767-bdc4-1ec60ea8cf05.pkl",
  "assetType": "EXTERNAL_FILE",
  "assetRole": "WEIGHTS_FILE",
  "metaData": null,
  "repositoryInfo": {
    "secretKey": "",
    "accessKey": "",
    "secure": false,
    "host": "minio",
    "defaultBucket": "modelop",
    "port": 9000,
    "region": "default-region",
    "repositoryType": "S3_REPO"
  },
  "importedRepository": {
    "classicRepository": false,
    "repositoryType": "GIT_REPO",
    "repositoryDirectory": "/tmp/model-manage-repos/master/f7bfb31f-15a4-4d0f-adba-3163396174f8",
    "repositoryRemote": "https://github.com/modelop/enron_ecomm_lasso_demo",
    "repositoryBranch": "master",
    "lastCommitId": "921625129157310de26c272d77e20a5f56d7feac"
  },
  "fileUrl": "http://minio:9000/modelop/f7bfb31f-15a4-4d0f-adba-3163396174f8/66d08945-4d9b-4767-bdc4-1ec60ea8cf05.pkl",
  "filename": "lasso_model_artifacts.pkl",
  "fileFormat": "PKL",
  "fileSize": 150197
}
],
"gitDifferences": [
  {
    "oldPath": "enron_ecomm_lasso_metrics.py",
    "newPath": "enron_ecomm_lasso_metrics.py",
    "changeType": "MODIFY",
    "oldId": "AbbreviatedObjectId[bc437c88dc5aca477ddd501f37f331b24c278136]",
    "newId": "AbbreviatedObjectId[530e6c54577d214c16868b9567a5d5d7e297c358]",
    "diff": "diff --git a/enron_ecomm_lasso_metrics.py b/enron_ecomm_lasso_metrics.py\nindex bc437c8..530e6c5 100644\n---\na/enron_ecomm_lasso_metrics.py\n+++ b/enron_ecomm_lasso_metrics.py\n@@ -7,7 +7,7 @@\n import sklearn\n import sys\n \n\n-# An example model\nrepo\n+# Updates\n\n def remove_proper_nouns(string):\n list_of_words = string.split()\n",
    "author": "modelop",
    "authorEmail": "modelop@model-manager",
    "committer": "modelop",
    "committerEmail": "modelop@model-manager"
  }
],
"associatedModels": [],
"links": {
  "self": {
    "href": "http://a237f88b02a4441ceab40b6aac10cbbe-1017152738.us-east-2.elb.amazonaws.com:8090/api/storedModels/f7bfb31f-15a4-4d0f-adba-3163396174f8"
  }
}
}

```

```
"storedModel": {
  "href": "http://a237f88b02a4441ceab40b6aac10cbbe-1017152738.us-east-2.elb.amazonaws.com:8090/api/storedModels/f7bfb31f-15a4-4d0f-adba-3163396174f8"
}
}
```

Custom Metadata

Overview

ModelOp Center allows Customers to implement Customer-specific metadata for a model—for a StoredModel, DeployableModel, and DeployedModel. The “Custom” fields for each of these elements are flexible to allow the Customer to define simple or complex elements – the only requirement is that it implements standard json.

For example, a Customer may want to require that users add custom metadata upon registration. This metadata would be placed within the “custom” field of the StoredModel upon registration.

Another example is that a Customer’s model lifecycle might require that specific governance approvals be stored as custom metadata with a “version” of a model. This would be placed with the “metadata” field of the DeployableModel “snapshot” of the model for auditability.

Adding/Modifying Custom Metadata

Custom metadata can be added for StoredModels, DeployableModels, or DeployedModels through the ModelOp Center API or via an MLC.

Below are several mechanisms to add/update custom metadata via the ModelOp Center API:

- (1) **Register with a StoredModel:** add the custom metadata to a StoredModel upon initial registration via POST “/api/storedModels”
 - a. Simply add in metadata in the “modelMetaData”:”custom” field
- (2) **Add or update a StoredModel:** add or update the custom metadata to an existing StoredModel via “PATCH /api/storedModels/{id}”
 - a. An example input for the PATCH:

```
"modelMetaData": {
  "custom": {
    "CustomElement1": "Value1",
    "CustomElement2": "Value2",
    "CustomElement3": "Value3"
  }
}
```
- (3) **Add or update a DeployableModel:** add or update the custom metadata for a DeployableModel via PATCH “/api/deployableModels/{id}”
 - a. An example input for the PATCH:

```
"metaData": {
  "custom": {
    "DeployableModelCustomElement1": "Value1",
    "DeployableModelCustomElement2": "Value2",
    "DeployableModelCustomElement3": "Value3"
  }
}
```

```
}
```

(4) **Add or update a DeployedModel:** add or update the custom metadata for a DeployedModel via PATCH “/api/deployedModels/{id}”

a. An example input for the PATCH:

```
"metaData": {  
  "custom": {  
    "DeployedModelCustomElement1": "Value1",  
    "DeployedModelCustomElement2": "Value2",  
    "DeployedModelCustomElement3": "Value3"  
  }  
}
```