5.6 上传 multipart 对象

若对象大小超过了 100M，建议使用 Multipart Upload 方式上传，示例七给出了创建一个 multipart 的代码示例，示例中上传了一个名为 Test 的 multipart 对象到 Bucket_multipart 桶中，其中 Test 对象是由三个 file 文件组成。

具体流程包括：case 1：根据 bucket（桶）和 key 值（对象），初始化 Multipart 上传，获取一个 uploadId；case 2：根据 uploadId 及相关参数，按 partNumber 分块上传 part 数据，并获取每个 part 返回的 ETag 信息；case 3：查看已经上传的 part 列表；case 4：上传 partETags 列表及相关参数，完成 Multipart 上传。

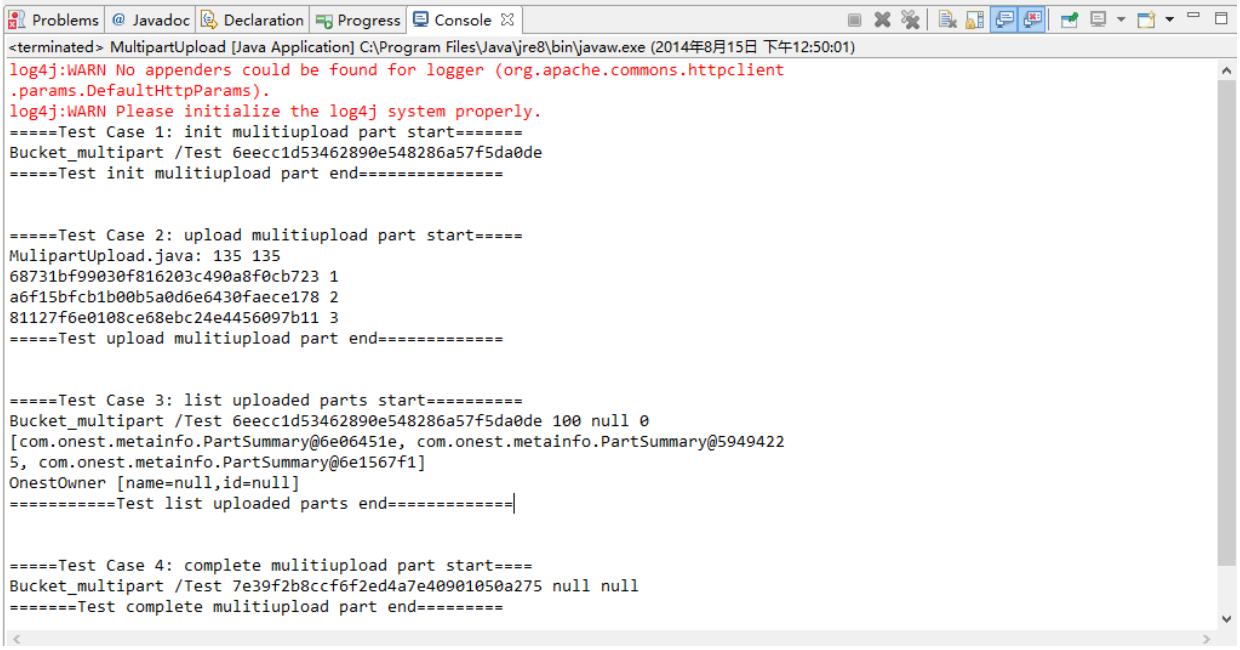
```
package onest.sdk.test;

import java.io.File;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

import com.onest.auth.PropertiesCredentials;
import com.onest.client.Onest;
import com.onest.client.OnestClientException;
import com.onest.client.OnestClientFactory;
import com.onest.metainfo.*;
import com.onest.request.*;

public class MultipartUpload {
    public static void main(String[] args) throws IOException {
        Onest onest = null;
        /*获取oNest客户端工厂实例*/
        OnestClientFactory ocf = OnestClientFactory.getInstance();
        /*读取配置文件OnestCredentials.properties*/
        PropertiesCredentials onestProperites = new PropertiesCredentials
(CreateObject.class.getResourceAsStream("/OnestCredentials.properties"));
        String bucketName = onestProperites.getOnestBucketName();
        String key = onestProperites.getOnestObjectKey();
        ocf.setFactoryInfo(onestProperites, onestProperites
.getOnestHostAddr());
        /*读取配置文件中参数isIntraAddr和isIntraSrvLocation的参数值*/
        boolean isIntraAddr = false;
        boolean isIntraSrvLocation = false;
        if(onestProperites.getIsIntraAddr() != null) {
            isIntraAddr = onestProperites.getIsIntraAddr().equals("true") ? true : false;
        }
        if(onestProperites.getIsIntraSvrAddr() != null) {
```

```
            isIntraSrvLocation = onestProperites.getIsIntraSvrAddr().equals("true") ?
true : false;
        }
        try {
            /*创建oNest客户端*/
            onest = ocf.getLocation(isIntraAddr, isIntraSrvLocation);
            System.out.println("=====Test Case 1: init mulitiupload part start=======\n");
            //构建一个上传multipart的初始化请求
            InitiateMultipartUploadRequest initiateMultipartUploadRequest =
new InitiateMultipartUploadRequest(bucketName,key);
            //发送请求，并获取一个uploadID
            InitiateMultipartUploadResult result =
onest.initiateMultipartUpload(initiateMultipartUploadRequest);
            System.out.println(result.getBucketName() + " " + result.getKey() + " "
+ result.getUploadId());
            System.out.println("=====Test init mulitiupload part end==============\n\n");

            System.out.println("=====Test Case 2: upload mulitiupload part start=====\n");
            final long contentLength = 135;//file.length();
            long partSize = contentLength;
            File file = new File("D:/下载/云计算新员工BC-EC技术培训计划.xlsx");
            System.out.println("MulipartUpload.java: " + contentLength + " " + partSize);
            //上传3个part
            for(int i=1;i<4;i++){
                //构建一个uploadRequest的参数,包含的参数如下:
                UploadPartRequest uploadPartRequest = new UploadPartRequest()
.withBucketName(bucketName).withKey(key).withFile(file).withUploadId(result.getUploadId())
.withPartNumber(i).withPartSize(partSize).withFileOffset(0);
                //上传一个part,返回一个ETag
                UploadPartResult uploadPartResult = onest.uploadPart(uploadPartRequest);
                System.out.println(uploadPartResult.getETag() + " " + uploadPartResult
.getPartNumber());
            }
            System.out.println("=====Test upload mulitiupload part end============\n\n");

            //查看上传的part
            System.out.println("=====Test Case 3: list uploaded parts start==========\n");
            //构建一个listPartRequest
            ListPartsRequest listPartsRequest =new ListPartsRequest(bucketName, key,
result.getUploadId());
            //获取上传part的所有信息
            PartListing partListing = onest.listParts(listPartsRequest);
            System.out.println(partListing.getBucketName() + " " + partListing.getKey()
+ " " + partListing.getUploadId() + " "+partListing.getMaxParts() + " " + partListing
```

```
.getNextPartNumberMarker()+ " "+ partListing.getPartNumberMarker()+ " " + partListing
.getParts()+ " " + partListing.getOwner());
            System.out.println("===========Test list uploaded parts end============\n\n");

            System.out.println("=====Test Case 4: complete mulitiupload part start====\n");
            //获取上传part的概况信息，包括partNumber,etag等
            List<PartSummary> partSummariesList = partListing.getParts();
            //创建partEtags列表
            List<PartETag> partETags = new ArrayList<PartETag>();
            for(int i=1;i<4;i++){
                //将partSummariesList的Etag信息添加到PartEtags列表中
                PartETag partETag = new PartETag(i, partSummariesList.get(i-1).getETag());
                partETags.add(partETag);
            }
            //构建一个completeMultipartUpload的请求
            CompleteMultipartUploadRequest completeMultipartUploadRequest =
new CompleteMultipartUploadRequest(bucketName, key,result.getUploadId(), partETags);
            //获取返回参数
            CompleteMultipartUploadResult completeMultipartUploadResult = onest
.completeMultipartUpload(completeMultipartUploadRequest);
            System.out.println(completeMultipartUploadResult.getBucketName() + " " +
completeMultipartUploadResult.getKey()+ " " + completeMultipartUploadResult.getETag() +
" "+completeMultipartUploadResult.getLocation() + " "
+ completeMultipartUploadResult.getVersionId());
            System.out.println("=======Test complete mulitiupload part end=========\n\n");
        }catch (OnestClientException ce) {
            ce.printStackTrace();
        }
    }
}
```

示例七 MultipartUpload

在 Eclipse 中执行 multipart 上传的程序，成功的上传了 Test 对象，其中 Test 是由三个 File 文件组成，如图十五是程序执行结果。其中 case 1 是初始化 multipart，case 2 是上传 part 的结果，case 3 是获取上传 part 的列表，case 4 是完成 multipart 上传。



图十五 示例七的执行结果

5.7 设置 Bucket 的权限

用户可以通过设置权限，来指定其他用户访问自己的容器，设置 Bucket 首先要定义权限者，其次要给权限者赋予指定的权限，包括读和写两种方式。示例八给出了设置桶 Bucket_multipart 权限的代码示例，桶的拥有者 CIDC-U-0000002527，授予权限者 admin 包括 read 和 write 的访问权限，之后通过获取桶的权限，验证设置 Bucket_multipart 是否成功。

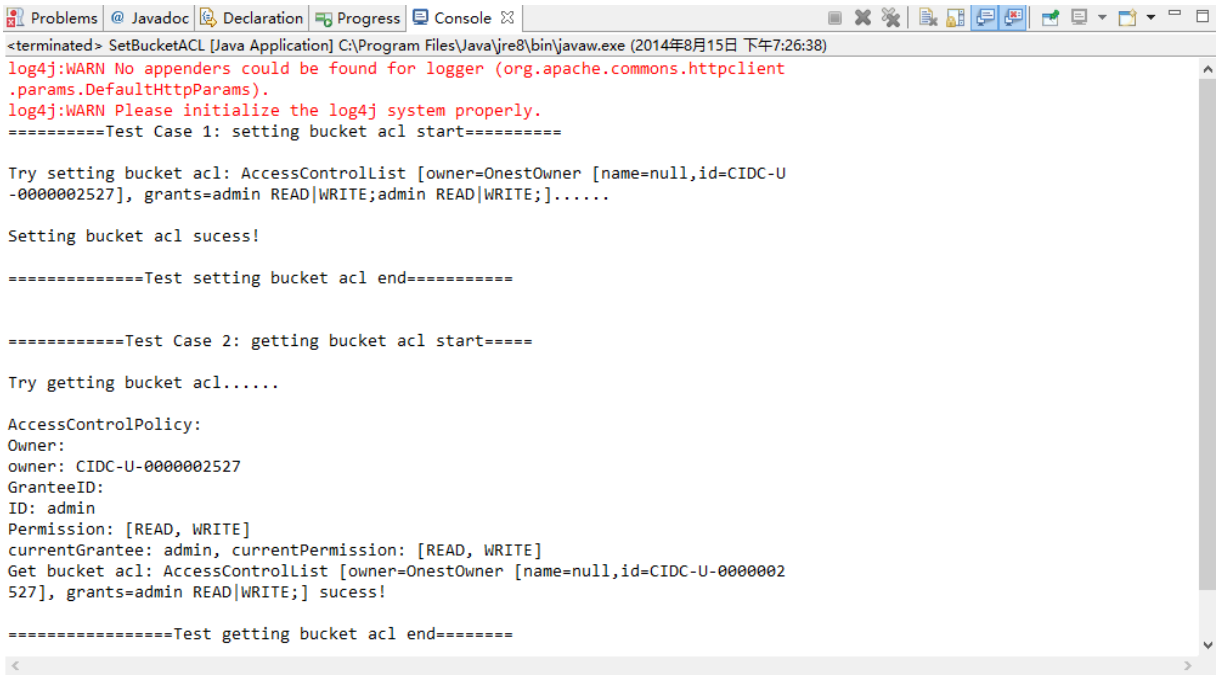
```
package onest.sdk.test;

import java.io.IOException;
import java.util.LinkedList;
import java.util.List;

import com.onest.auth.PropertiesCredentials;
import com.onest.client.Onest;
import com.onest.client.OnestClientException;
import com.onest.client.OnestClientFactory;
import com.onest.metainfo.AccessControlList;
import com.onest.metainfo.CanonicalGrantee;
import com.onest.metainfo.Grantee;
import com.onest.metainfo.Owner;
import com.onest.metainfo.Permission;

public class SetBucketACL {
    /**
```

```
     * @param args
     * @throws IOException
     */
    public static void main(String[] args) throws IOException {
        Onest onest = null;
        /*获取oNest客户端工厂实例*/
        OnestClientFactory ocf = OnestClientFactory.getInstance();
        /*读取配置文件OnestCredentials.properties*/
        PropertiesCredentials onestProperites = new PropertiesCredentials(SetBucketACL.class
.getResourceAsStream("/OnestCredentials.properties"));
        ocf.setFactoryInfo(onestProperites, onestProperites.getOnestHostAddr());
        /*获取原用户，以及权限用户*/
        String userName = onestProperites.getUserName();
        String bucketAclUser = onestProperites.getOnestBucketAclUser1();
        /*读取配置文件中参数isIntraAddr和isIntraSrvLocation的参数值*/
        boolean isIntraAddr = false;
        boolean isIntraSrvLocation = false;
        if(onestProperites.getIsIntraAddr() != null) {
            isIntraAddr = onestProperites.getIsIntraAddr().equals("true") ? true : false;
        }
        if(onestProperites.getIsIntraSvrAddr() != null) {
            isIntraSrvLocation = onestProperites.getIsIntraSvrAddr().equals("true")
? true : false;
            }
        try {
            /*创建oNest客户端*/
            onest = ocf.getLocation(isIntraAddr, isIntraSrvLocation);
            System.out.println("==========Test Case 1: setting bucket acl start==========\n");
            //创建一个新的acl
            AccessControlList acl = new AccessControlList();
            Owner own = new Owner();
            //桶拥有者的ID
            own.setId(userName);
            acl.setOwner(own);
            /*定义权限者grantee*/
            Grantee grantee = new CanonicalGrantee(bucketAclUser);
            //设置perimission，包括read和write两种权限
            List<Permission> listPermission = new LinkedList<Permission>();
            listPermission.add(Permission.Read);
            listPermission.add(Permission.Write);
            acl.grantPermission(grantee, listPermission);
            System.out.println("Try setting bucket acl: " + acl.toString() + "......\n");

            //注意，acl的grantee 必须是存在的用户
```

```
            onest.setBucketAcl("Bucket_multipart", acl);
            System.out.println("Setting bucket acl sucess!\n");
            System.out.println("==============Test setting bucket acl end==========\n\n");

            System.out.println("============Test Case 2: getting bucket acl start=====\n");
            System.out.println("Try getting bucket acl......\n");
            acl = onest.getBucketAcl("Bucket_multipart");
            System.out.println("Get bucket acl: "+ acl.toString() + " sucess!\n");
            System.out.println("=================Test getting bucket acl end========\n\n");
        }catch (OnestClientException ce) {
            ce.printStackTrace();
        }
    }
}
```

示例八 SetBucketAcl

在 Eclipse 执行以上程序，图十六给出了程序执行结果。Case 1 成功设置 Bucket_multipart 的权限，授予 admin 用户拥有 read 和 write 的访问权限；Case 2 通过获取桶的权限列表，显示 Bucktet_multipart 的拥有者 Owner 和权限者 grants，以及 grants 拥有的权限。



图十六 示例八的执行结果

5.7 设置 Object 的权限

设置 Object 的权限与设置 Bucket 的权限的逻辑相同，用户可以通过设置权限，来指定其他用户访问自己的对象，设置 Object 首先要定义权限者，其次要给权限者赋予指定的权限。示例九给出了设置桶 Bucket_multipart 里面对象 Test 访问权限的代码示例，对象的拥有

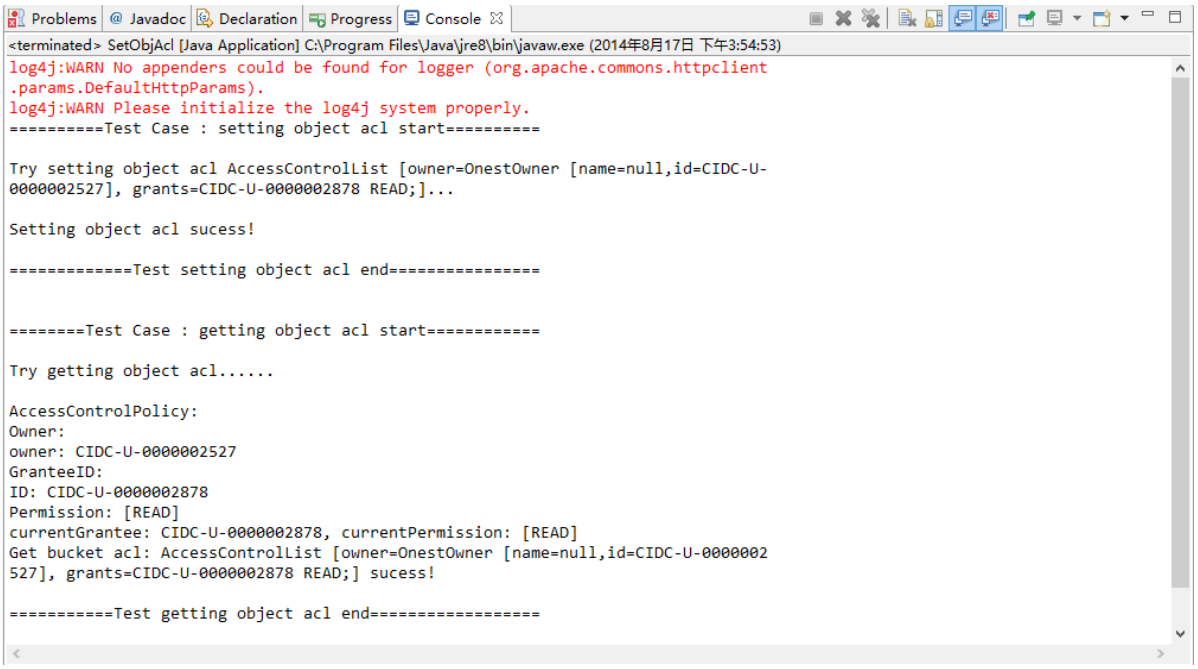
者 CIDC-U-0000002527，授予权限者 CIDC-U-0000002878 拥有 read 的访问权限，之后通过获取对象的权限列表，验证设置 Test 的权限是否成功。

```
package onest.sdk.test;

import java.io.IOException;
import java.util.LinkedList;
import java.util.List;

import com.onest.auth.PropertiesCredentials;
import com.onest.client.Onest;
import com.onest.client.OnestClientException;
import com.onest.client.OnestClientFactory;
import com.onest.metainfo.AccessControlList;
import com.onest.metainfo.CanonicalGrantee;
import com.onest.metainfo.Grantee;
import com.onest.metainfo.Owner;
import com.onest.metainfo.Permission;

public class SetObjAcl {
    public static void main(String[] args) throws IOException {
        Onest onest = null;
        /*获取oNest客户端工厂实例*/
        OnestClientFactory ocf = OnestClientFactory.getInstance();
        /*读取配置文件OnestCredentials.properties*/
        PropertiesCredentials onestProperites = new PropertiesCredentials(SetObjAcl.class
.getResourceAsStream("/OnestCredentials.properties"));
        ocf.setFactoryInfo(onestProperites, onestProperites.getOnestHostAddr());
        String bucketName = onestProperites.getOnestBucketName();
        String key = onestProperites.getOnestObjectKey();
         /*获取对象原用户及权限用户*/
        String userName = onestProperites.getUserName();
        String ObjAclUser = onestProperites.getOnestBucketAclUser2();
        /*读取配置文件中参数isIntraAddr和isIntraSrvLocation的参数值*/
        boolean isIntraAddr = false;
        boolean isIntraSrvLocation = false;
        if(onestProperites.getIsIntraAddr() != null) {
            isIntraAddr = onestProperites.getIsIntraAddr().equals("true") ? true : false;
       }
        if(onestProperites.getIsIntraSvrAddr() != null) {
            isIntraSrvLocation = onestProperites.getIsIntraSvrAddr().equals("true")
? true : false;
        }
        try {
            /*创建oNest客户端*/
```

```
            onest = ocf.getLocation(isIntraAddr, isIntraSrvLocation);
            System.out.println("==========Test Case 1: setting object acl start=========\n");
            AccessControlList object_acl = new AccessControlList();
              //对象拥有者的ID
            Owner own = new Owner();
            own.setId(userName);
            object_acl.setOwner(own);
            //*定义权限者grantee1*/
            Grantee grantee1 = new CanonicalGrantee(ObjAclUser);
            //设置perimission，grantee1拥有read权限
            object_acl.grantPermission(grantee1, Permission.Read);
            System.out.println("Try setting object acl " + object_acl.toString() + "...\n");
            //grantee 必须是存在的用户，否则设置不成功
            onest.setObjectAcl(bucketName, key, object_acl);
            System.out.println("Setting object acl sucess!\n");
            System.out.println("===========Test setting object acl end==============\n\n");

            System.out.println("========Test Case : getting object acl start==========\n");
            System.out.println("Try getting object acl......\n");
            //获取对象的权限列表
            AccessControlList obj_acl = onest.getObjectAcl(bucketName, key);
            System.out.println("Get bucket acl: "+ obj_acl.toString() + " sucess!\n");
            System.out.println("==========Test getting object acl end===============\n\n");
        }catch (OnestClientException ce) {
            ce.printStackTrace();
        }
    }
}
```

示例九 SetObjAcl

在 Eclipse 执行以上程序，图十七给出了程序执行结果。Case 1 成功设置 Test 的访问权限，授予 CIDC-U-0000002878 用户拥有 read 的访问权限；Case 2 通过获取对象的权限列表，显示 Test 的拥有者 Owner 和权限者 grants，以及 grants 拥有的权限。



图十七 示例九执行的结果

5.8 设置 Object 的属性

用户可以设置 Object 的属性，包括用户自定义属性 usermeta、版本号 version 等，用户设置的数据通过 HTTP HEAD 的请求方式，添加到请求的消息头中，示例十给出了用户自定义对象”Test“属性的代码示例，测试代码中 case 1 添加了”Test“的 usermeta 属性，key 值分别为：level、maker；value 值为：important、david。case 2 通过获取 object 的属性参数，检查设置是否成功。

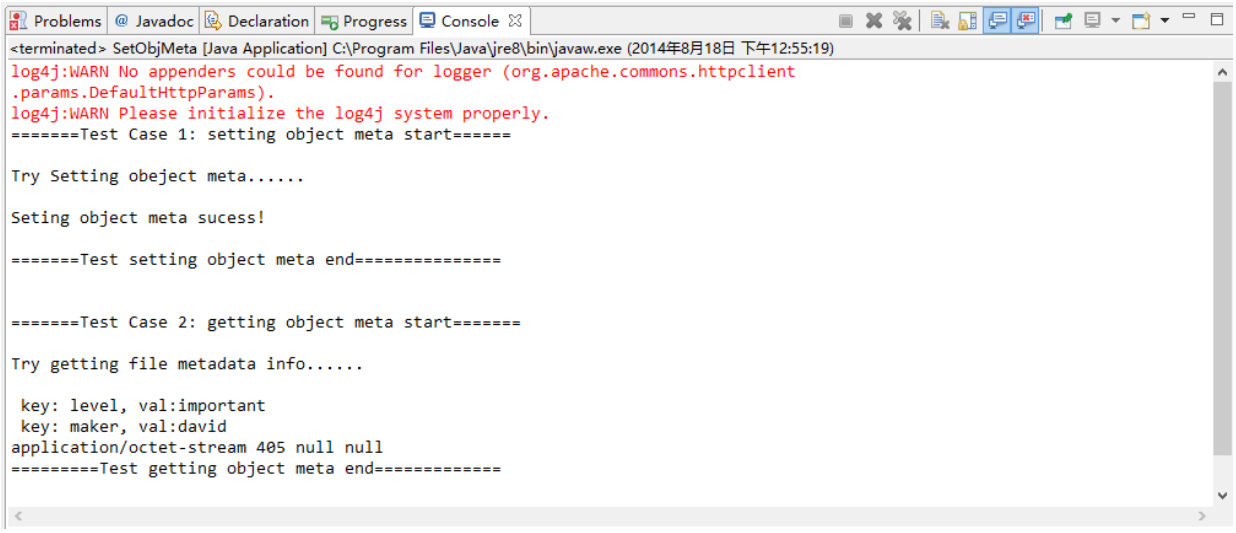
```
package onest.sdk.test;

import java.io.IOException;
import java.util.Iterator;
import java.util.Map;

import com.onest.auth.PropertiesCredentials;
import com.onest.client.Onest;
import com.onest.client.OnestClientException;
import com.onest.client.OnestClientFactory;
import com.onest.metainfo.AccessControlList;
import com.onest.metainfo.ObjectMetadata;
import com.onest.metainfo.Owner;
```

```
import com.onest.request.GetObjectMetadataRequest;

public class SetObjMeta {
    public static void main(String[] args) throws IOException {
        Onest onest = null;
        /*获取oNest客户端工厂实例*/
        OnestClientFactory ocf = OnestClientFactory.getInstance();
        /*读取配置文件OnestCredentials.properties*/
        PropertiesCredentials onestProperites = new PropertiesCredentials(SetObjAcl.class
.getResourceAsStream("/OnestCredentials.properties"));
        ocf.setFactoryInfo(onestProperites, onestProperites.getOnestHostAddr());
        /*读取配置文件中参数isIntraAddr和isIntraSrvLocation的参数值*/
        boolean isIntraAddr = false;
        boolean isIntraSrvLocation = false;
        String bucketName = onestProperites.getOnestBucketName();
        String key = onestProperites.getOnestObjectKey();
        String userName = onestProperites.getUserName();
        Owner own = new Owner();
        own.setId(userName);
        if(onestProperites.getIsIntraAddr() != null) {
            isIntraAddr = onestProperites.getIsIntraAddr ().equals("true") ? true : false;
       }
        if(onestProperites.getIsIntraSvrAddr() != null) {
            isIntraSrvLocation = onestProperites.getIsIntraSvrAddr().equals("true")
? true : false;
        }
        try {
            /*创建oNest客户端*/
            onest = ocf.getLocation(isIntraAddr, isIntraSrvLocation);
           System.out.println("=======Test Case 1: setting object meta start======\n");
           //定义一个objectMetadata的类
           ObjectMetadata objMetadata = new ObjectMetadata();
           //获取一个objmetadata里面的uesrmeta
           Map<String, String> userMeta = objMetadata.getUserMetadata();
           //设置两个属性 Key-value
           userMeta.put("level", "important");
           userMeta.put("maker","david");
           System.out.println("Try Setting obeject meta......\n");
           //通过传递参数objmetadata，设置obj的metadata
           onest.setObjectMetadata(bucketName, key, objMetadata);
           System.out.println("Seting object meta sucess!\n");
           System.out.println("=======Test setting object meta end==============\n\n");

           System.out.println("=======Test Case 2: getting object meta start=======\n");
```

```
            System.out.println("Try getting file metadata info......\n");
            //发起getobjectMetaData的请求
            GetObjectMetadataRequest getObjectMetadataRequest = new
GetObjectMetadataRequest(bucketName, key);
            //获取对象的metadata
            objMetadata = onest.getObjectMetadata(getObjectMetadataRequest);
            Map<String, String> userMetadata = objMetadata.getUserMetadata();
            //使用迭代方法遍历userMetadata（map表）的数据
            Iterator iter = userMetadata.entrySet().iterator();
            while (iter.hasNext()) {
                Map.Entry entry = (Map.Entry) iter.next();
                Object key1 = entry.getKey();
                Object val = entry.getValue();
                System.out.println(" key: " + key1 +", val:" + val);
            }
            System.out.println(objMetadata.getContentType() + " " +
objMetadata.getContentLength() + " " + objMetadata.getETag() + " " +
objMetadata.getLastModified());
            System.out.println("=========Test getting object meta end=============\n\n");
        }catch (OnestClientException ce) {
             ce.printStackTrace();
         }
    }
}
```

示例十 SetObjMeta

在 Eclipse 执行以上程序，图十八给出了程序执行结果。Case 1 成功设置了 Test 的 usermeta 参数；Case 2 通过获取 Test 的属性参数，显示 key :level , val:important; key:maker, val:david 验证了属性设置成功。



图十八 示例十执行的结果