Hadoop

Configurations

ISEL
INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Configuration API

- Components in Hadoop can be configured using instances of the class `Configuration` contained in the package `org.apache.hadoop.conf`
- A configuration represents a set of properties and their corresponding values, and optionally a description

Configuration API

- Multiple configurations can be placed inside XML files

- Hadoop applications can read their configuration from configurations files

- Properties marked as final cannot be overridden

Configuration API

Configuration-1.xml

```
<?xml version="1.0" encoding="UTF-8">
<configuration>
  <property>
    <name>color</name>
    <value>yellow</value>
  </property>

  <property>
    <name>size</name>
    <value>10</value>
  </property>

  <property>
    <name>weight</name>
    <value>heavy</value>
    <final>true</final>
  </property>

  <property>
    <name>size-weight</name>
    <value>${size},${weight}</value>
  </property>
</configuration>
```

Configuration-2.xml

```
<?xml version="1.0" encoding="UTF-8">
<configuration>
  <property>
    <name>size</name>
    <value>12</value>
  </property>

  <property>
    <name>weight</name>
    <value>light</value>
  </property>
</configuration>
```

Configuration API

Ex22

```
import ...;

public class ReadConfigurationVer01 {
  public static void main(String[] args) throws Exception {
    String uri = args[0];
    Configuration conf = new Configuration();
    conf.addResource( new Path( uri, "configuration-1.xml" ) );

    System.out.println( "Asserting property \"color\"..." );
    assertThat( conf.get("color"), is("yellow") );
    System.out.println( "Asserting property \"size\"..." );
    assertThat( conf.getInt("size", 0), is(10) );
    System.out.println( "Asserting property \"breadth\"..." );
    assertThat( conf.get("breadth", "wide"), is("wide") );
  }
}
```

Configuration API

Ex23

```java
import ...;

public class ReadConfigurationVer02 {
  public static void main(String[] args) throws Exception {
    String uri = args[0];
    Configuration conf = new Configuration();

    conf.addResource( new Path( uri, "configuration-1.xml" ) );
    conf.addResource( new Path( uri, "configuration-2.xml" ) );

    System.out.println( "Property \"size\" is overriden." );
    assertThat( conf.getInt("size", 0), is(12) );

    System.out.println( "Property marked with final." );
    assertThat( conf.get("weight"), is("heavy") );
  }
}
```

Hadoop Jobs with configurations

- Almost any of the tools that comes with Hadoop support the -conf argument that allows to pass to those tools (and or MapReduce application) the name of a configuration file (in XML format)

```
hadoop <tool> -conf configurationFile.xml arguments
```

- Individual configurations can be made using the -D option

Configuration API

Ex25

```
import ...;

public class ConfigurationPrinter
        extends Configured
        implements Tool {

  static {
    Configuration.addDefaultResource("hdfs-default.xml");
    Configuration.addDefaultResource("hdfs-site.xml");
    Configuration.addDefaultResource("yarn-default.xml");
    Configuration.addDefaultResource("yarn-site.xml");
    Configuration.addDefaultResource("mapred-default.xml");
    Configuration.addDefaultResource("mapred-site.xml");
   }

   ...
```

Configuration API

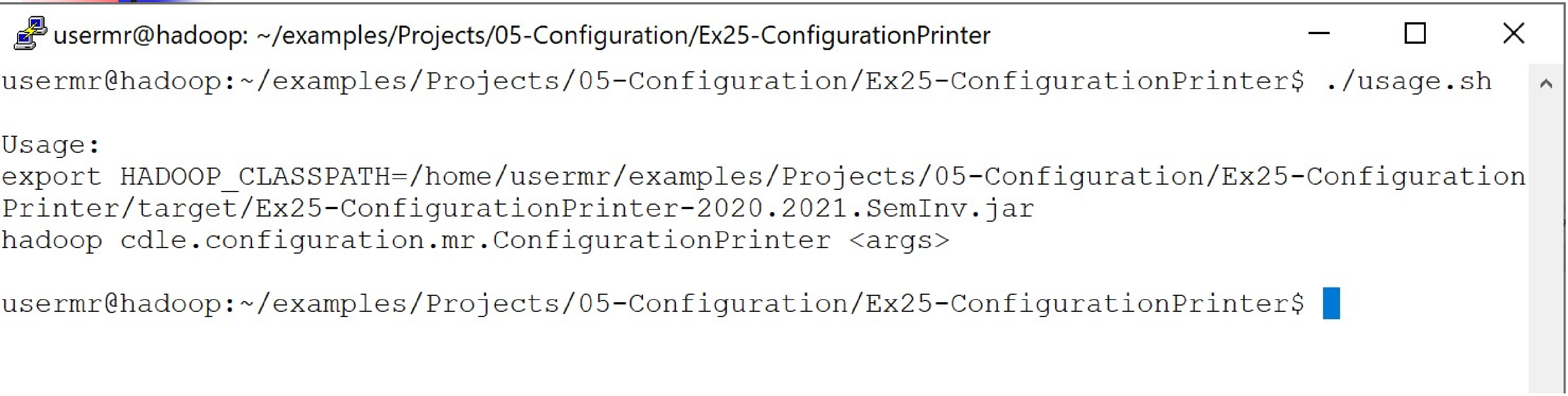
Ex25

```
  ...

  @Override
  public int run(String[] args) throws Exception {
    Configuration conf = getConf();
    for (Entry<String, String> entry: conf) {
      System.out.printf("%s=%s\n", entry.getKey(), entry.getValue());
    }
    return 0;
  }

  public static void main(String[] args) throws Exception {
    int exitCode = ToolRunner.run( new ConfigurationPrinter(), args );
    System.exit (exitCode );
  }
}
```

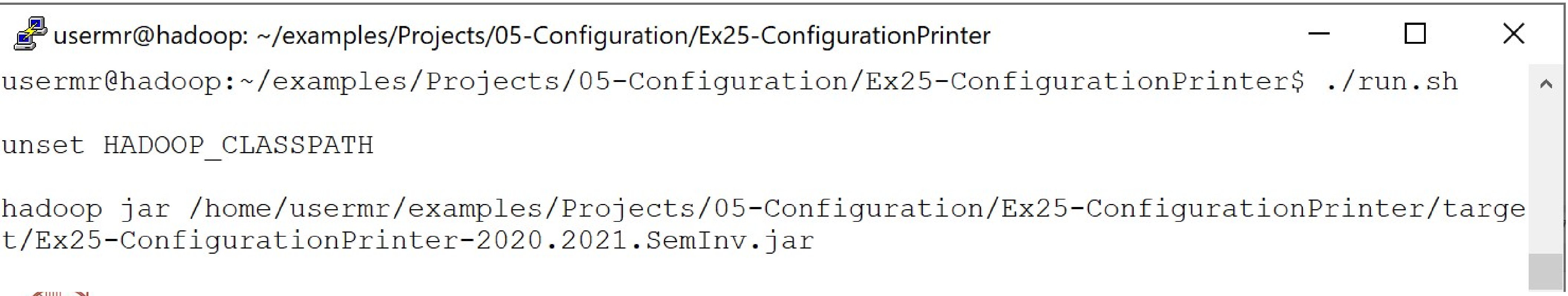
Configuration API


usermr@hadoop: ~/examples/Projects/05-Configuration/Ex25-ConfigurationPrinter

```
usermr@hadoop:~/examples/Projects/05-Configuration/Ex25-ConfigurationPrinter$ ./usage.sh

Usage:
export HADOOP_CLASSPATH=/home/usermr/examples/Projects/05-Configuration/Ex25-Configuration
Printer/target/Ex25-ConfigurationPrinter-2020.2021.SemInv.jar
hadoop cdle.configuration.mr.ConfigurationPrinter <args>

usermr@hadoop:~/examples/Projects/05-Configuration/Ex25-ConfigurationPrinter$
```

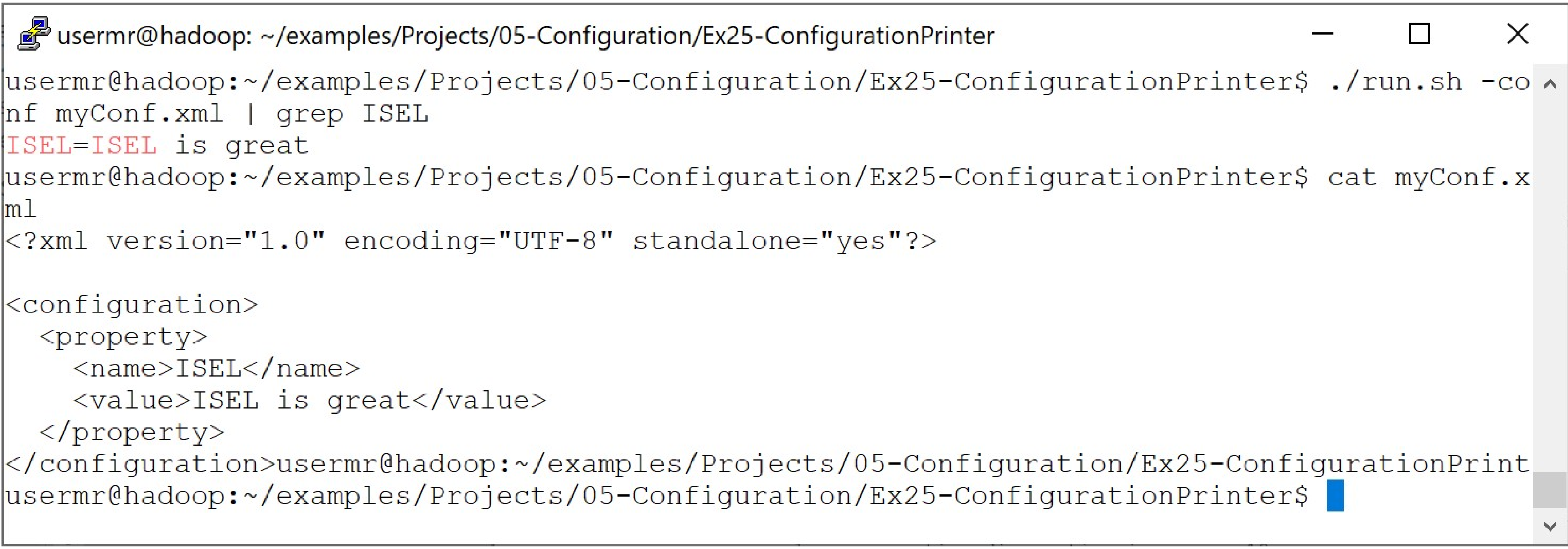

usermr@hadoop: ~/examples/Projects/05-Configuration/Ex25-ConfigurationPrinter

```
usermr@hadoop:~/examples/Projects/05-Configuration/Ex25-ConfigurationPrinter$ ./run.sh

unset HADOOP_CLASSPATH

hadoop jar /home/usermr/examples/Projects/05-Configuration/Ex25-ConfigurationPrinter/targe
t/Ex25-ConfigurationPrinter-2020.2021.SemInv.jar
```

Configuration API



usermr@hadoop: ~/examples/Projects/05-Configuration/Ex25-ConfigurationPrinter

```
usermr@hadoop:~/examples/Projects/05-Configuration/Ex25-ConfigurationPrinter$ ./run.sh -co
nf myConf.xml | grep ISEL
ISEL=ISEL is great
usermr@hadoop:~/examples/Projects/05-Configuration/Ex25-ConfigurationPrinter$ cat myConf.x
ml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<configuration>
  <property>
    <name>ISEL</name>
    <value>ISEL is great</value>
  </property>
</configuration>usermr@hadoop:~/examples/Projects/05-Configuration/Ex25-ConfigurationPrint
usermr@hadoop:~/examples/Projects/05-Configuration/Ex25-ConfigurationPrinter$
```