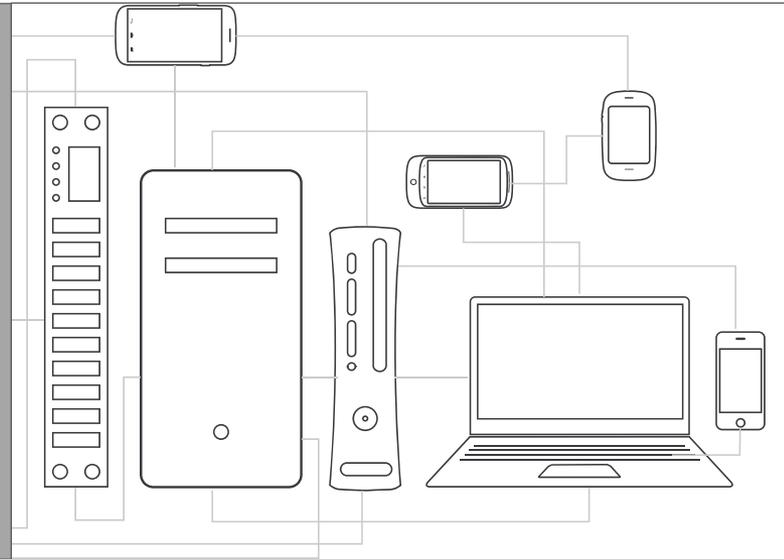


# Python Autopsy: Easier Forensics Scripting (not dead snakes)



Richard Cordovano



**BASIS**  
TECHNOLOGY

# Have you heard about Autopsy?

---



**Autopsy**<sup>®</sup>  
OPEN | EXTENSIBLE | FAST

- An open source desktop digital forensics tool, built on top of the SleuthKit

# Step 1: Add a data source

**Add Data Source**

**Steps**

- 1. Enter Data Source Information**
2. Configure Ingest Modules
3. Add Data Source

**Enter Data Source Information wizard (Step 1 of 3)**

Select source type to add: Image File

Browse for an image file:

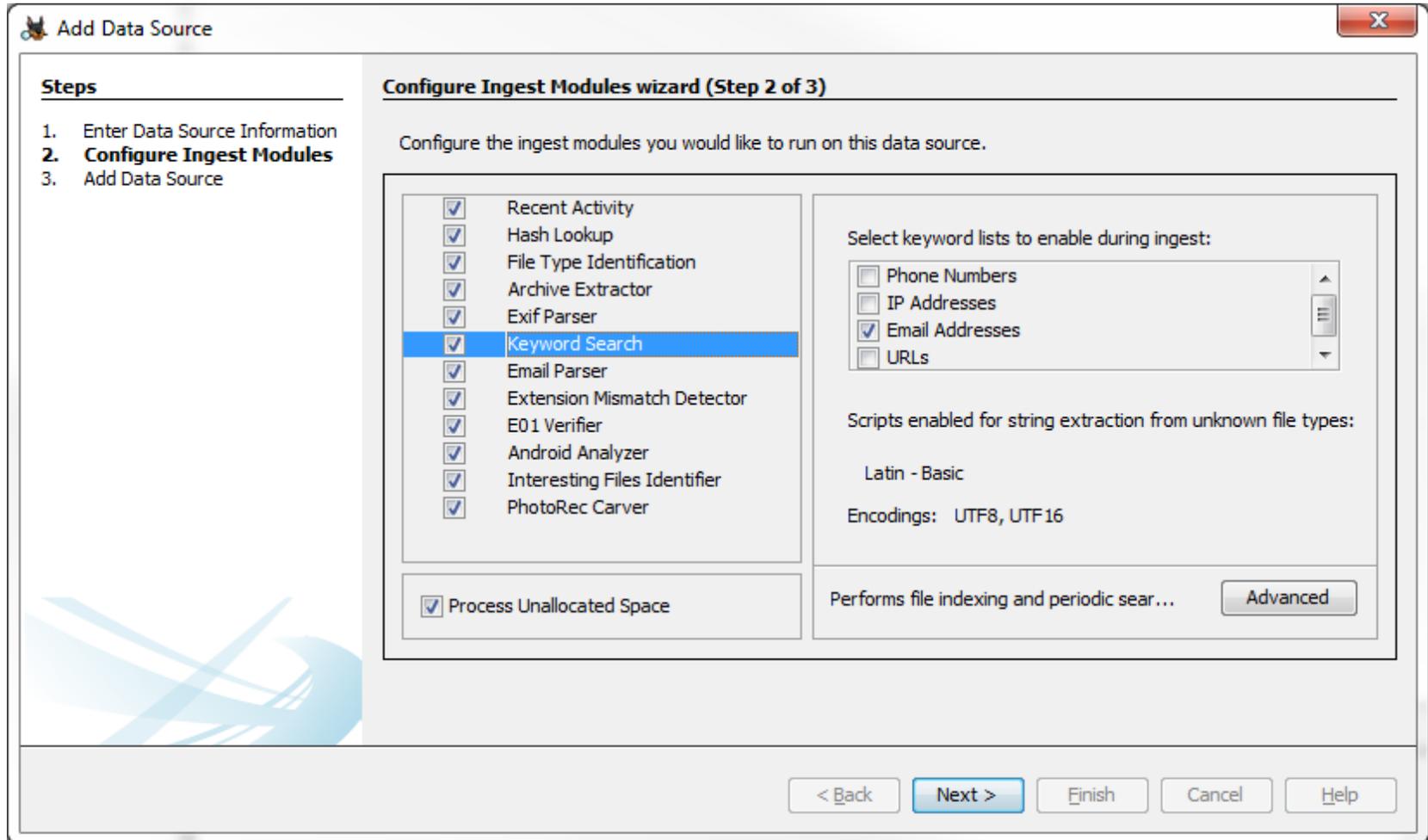
Please select the input timezone: (GMT-5:00) America/New\_York

Ignore orphan files in FAT file systems  
(faster results, although some data will not be searched)

Press 'Next' to analyze the input data, extract volume and file system data, and populate a local database.

< Back   Next >   Finish   Cancel   Help

# Step 2: Analyze it with ingest modules



# Step 3: Review the analysis results

XP - Autopsy 3.1.0

File View Tools Window Help

Close Case Add Data Source Generate Report

Keyword Lists Keyword Search

Directory Listing  
img\_xp-sp3-v4.001/vol2/Documents and Settings/All Users/Documents/My Pictures/Sample Pictures  
9 Results

Name	Modified Time	Change Time	Access Time	Created Time	Size	Flags(Dir)	Flags(Meta)	Mode	UserID	GroupID	Meta Addr.	Att
[current folder]	2012-01-20 18:19:25 EST	2012-01-20 18:19:25 EST	2012-03-10 14:44:59 EST	2012-01-20 17:20:10 EST	56	Allocated	Allocated	d-wx-wx-wx	0	0	6575	144-
[parent folder]	2012-01-20 17:20:10 EST	2012-01-20 17:20:10 EST	2012-03-10 14:44:59 EST	2012-01-20 17:19:58 EST	368	Allocated	Allocated	d-wx-wx-wx	0	0	5929	144-
Blue hills.jpg	2001-08-23 08:00:00 EDT	2012-01-20 17:20:10 EST	2012-03-10 19:23:39 EST	2012-01-20 17:20:10 EST	28521	Allocated	Allocated	rwxrwxrwx	0	0	6576	128-
desktop.ini	2012-01-20 17:20:18 EST	2012-01-20 17:20:18 EST	2012-03-19 14:06:50 EDT	2012-01-20 17:20:18 EST	42	Allocated	Allocated	r-xr-xr-x	0	0	6674	128-
Sunset.jpg	2001-08-23 08:00:00 EDT	2012-01-20 17:20:10 EST	2012-03-10 19:23:39 EST	2012-01-20 17:20:10 EST	71189	Allocated	Allocated	rwxrwxrwx	0	0	6577	128-
Thumbs.db	2012-01-20 18:19:25 EST	2012-01-20 18:19:25 EST	2012-01-20 18:19:25 EST	2012-01-20 18:19:25 EST	7680	Allocated	Allocated	r-xr-xr-x	0	0	1451	128-
Thumbs.db:encryptable	2012-01-20 18:19:25 EST	2012-01-20 18:19:25 EST	2012-01-20 18:19:25 EST	2012-01-20 18:19:25 EST	0	Allocated	Allocated	r-xr-xr-x	0	0	1451	128-
Water lilies.jpg	2001-08-23 08:00:00 EDT	2012-01-20 17:20:10 EST	2012-03-10 19:23:39 EST	2012-01-20 17:20:10 EST	83794	Allocated	Allocated	rwxrwxrwx	0	0	6578	128-
Winter.jpg	2001-08-23 08:00:00 EDT	2012-01-20 17:20:10 EST	2012-03-10 19:23:39 EST	2012-01-20 17:20:10 EST	105542	Allocated	Allocated	rwxrwxrwx	0	0	6579	128-

Hex Strings Metadata Results Text Media

# Ingest modules can...

---

- Access every byte of the data source
  - Data source file
  - Files in the data source courtesy of SleuthKit and other modules (archive extractors, carvers)
- Read and write the case database
- Use the blackboard to examine results of other modules and post results for other modules to see
- Submit files they discover (i.e., extracted, carved) for analysis
- So how do I write one?

# With...Java?

The screenshot displays the NetBeans IDE 8.0.1 interface. The main editor window shows the source code for `FilesIdentifierIngestModule.java`. The code includes a `process` method that iterates through `filesSets` and posts artifacts to a blackboard. The search results window at the bottom shows the following findings:

```
Search Results | Usages | Output | Breakpoints
-----
Usages of getIngestJobSnapshot | Usages of getJobSnapshots
-----
Usages of IngestJob.getJobSnapshots [1 occurrence]
-----
Autopsy-Core
├── IngestProgressSnapshotPanel.java
│   └── 170: jobSnapshots = IngestJob.getJobSnapshots();
```

# Works for me...does it work for you?

---

- Are you a professional software developer?
- Do you know Java or have time to learn it?
- Are you prepared to package and distribute your Autopsy plugins as NetBeans modules?

# The people want Python!

---

- Python is already familiar to many working in the digital forensics domain and lots of Python scripts are available for reuse
- Jython could be used as a code bridge between Java and Python to support:
  - A simple development environment, all you would need is a text editor
  - Faster development: change code and rerun without shutting down Autopsy
  - Easier module installation
- You got it!

# Getting started: one simple decision

---

- What kind of ingest module do you want to make?
  - Data source level module if you want to analyze the data source file or a subset of files in the data source
  - File level module if you want to analyze many or all files in the data source

# Finishing up: two things to do

---

- Write a few lines of script for an ingest module factory to make instances of your module for Autopsy
- Write as much script as you want inside your module to do your custom analysis

# Ingest module factory skeleton

```
class SampleJythonIngestModuleFactory(IngestModuleFactoryAdapter):  
  
    def getModuleDisplayName(self):  
        return "Sample Jython Ingest Module"  
  
    def getModuleDescription(self):  
        return "A sample Jython ingest module"  
  
    def getModuleVersionNumber(self):  
        return "1.0"  
  
    def isFileIngestModuleFactory(self):  
        return True  
  
    def createFileIngestModule(self, ingestOptions):  
        return SampleJythonFileIngestModule()
```

# Data Source ingest module skeleton

---

```
] class DataSourceIngestModuleSkeleton(DataSourceIngestModule) :  
|  
|     def startUp(self, context):  
|         pass  
|  
|     def process(self, dataSource, progressBar):  
|  
|         return IngestModule.ProcessResult.OK;
```

# File Ingest Module Skeleton

---

```
class FileIngestModuleSkeleton(FileIngestModule):  
  
    def startUp(self, context):  
        pass  
  
    def process(self, file):  
  
        return IngestModule.ProcessResult.OK;  
  
def shutDown(self):  
    pass
```

# Let's make an ingest module!

---

- We'll make it simple, let's find all files with "ebola" in the name and post them to the blackboard
- We only want some of the files, so we want to make a data source ingest module (or do we?)

# Ebola Finder module factory

```
]class EbolaFileFinderFactory(IngestModuleFactoryAdapter):  
  
]     def getModuleDisplayName(self):  
-         return "Ebola File Finder"  
  
]     def getModuleDescription(self):  
-         return "Finds files that have 'ebola' in the file name."  
  
]     def getModuleVersionNumber(self):  
-         return "1.0"  
  
]     def isDataSourceIngestModuleFactory(self):  
-         return True  
  
]     def createDataSourceIngestModule(self, ingestOptions):  
-         return EbolaFileFinder()
```

# How about this?

```
def process(self, dataSource, progressBar):

    # Get the file manager service
    autopsyCase = Case.getCurrentCase()
    sleuthkitCase = autopsyCase.getSleuthkitCase()
    services = Services(sleuthkitCase)
    fileManager = services.getFileManager()

    # Get files with "ebola" in name and post them to the blackboard.
    files = fileManager.findFiles(dataSource, "%malaria%")
    for file in files:
        art = file.newArtifact(
            BlackboardArtifact.ARTIFACT_TYPE.TSK_INTERESTING_FILE_HIT)
        att = BlackboardAttribute(
            BlackboardAttribute.ATTRIBUTE_TYPE.TSK_SET_NAME.getTypeID(),
            "Ebola File Finder", "Ebola Files")
        art.addAttribute(att)

    return IngestModule.ProcessResult.OK;
```

# Better!

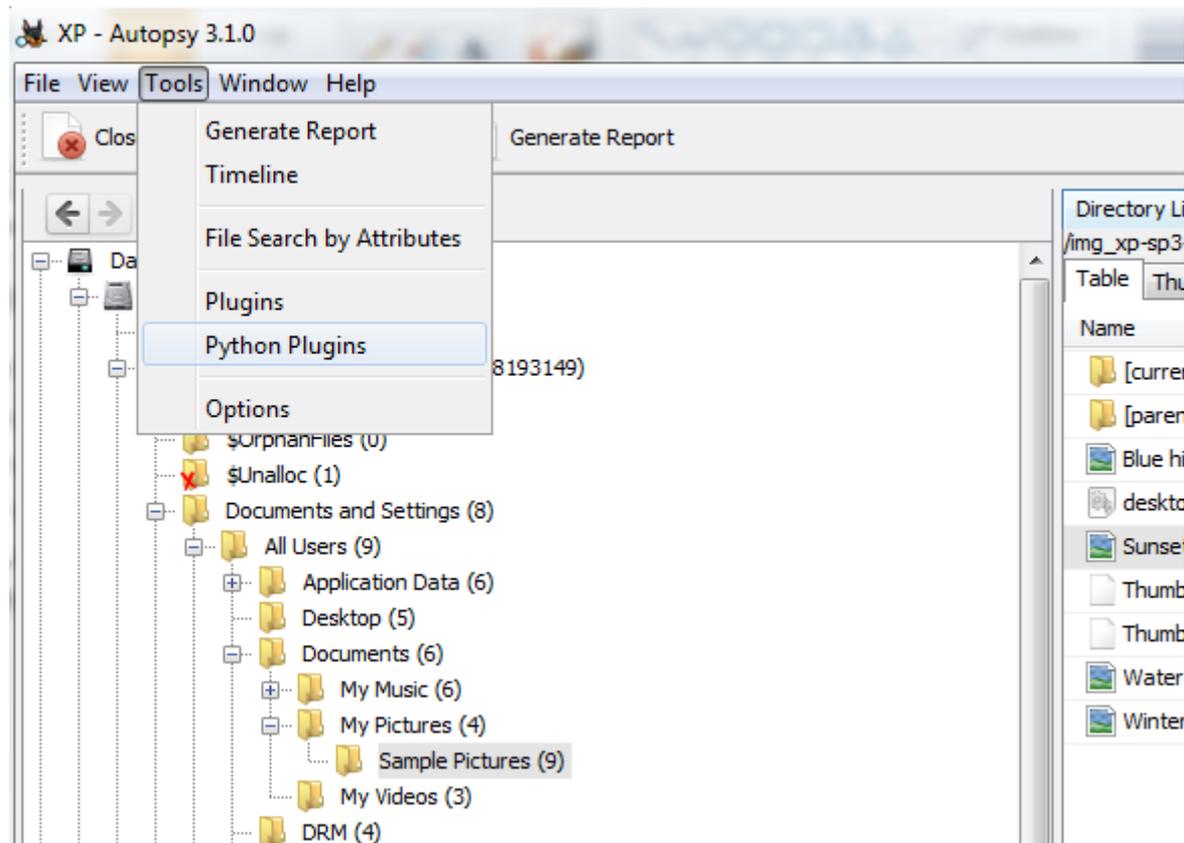
```
def process(self, file):
    # If the file name has "ebola" in it, post it to the blackboard.
    if file.getName().find("ebola") != -1:
        art = file.newArtifact(
            BlackboardArtifact.ARTIFACT_TYPE.TSK_INTERESTING_FILE_HIT)
        att = BlackboardAttribute(
            BlackboardAttribute.ATTRIBUTE_TYPE.TSK_SET_NAME.getTypeID(),
            "Ebola File Finder", "Text Files")
        art.addAttribute(att)

    return IngestModule.ProcessResult.OK
```

# Ebola Finder file module factory

```
]class EbolaFileFinderFactory(IngestModuleFactoryAdapter):  
  
]     def getModuleDisplayName(self):  
]         return "Ebola File Finder"  
  
]     def getModuleDescription(self):  
]         return "Finds files that have 'ebola' in the file name."  
  
]     def getModuleVersionNumber(self):  
]         return "1.0"  
  
]     def isFileIngestModuleFactory(self):  
]         return True  
  
]     def createFileIngestModule(self, ingestOptions):  
]         return EbolaFileFinder()
```

# Tools -> Python Plugins



# Drop!

The screenshot displays the XP - Autopsy 3.1.0 application interface. The main window shows a directory listing for the path `/img_xp-sp3-v4.001/vol_vol2/Documents and Settings/All Users/Documents/My Pictures/Sample Pictures`. The listing includes a table with columns for Name, Modified Time, Change Time, and Access. The first entry is `[current folder]` with a modified time of `2012-01-20 18:19:25 EST`.

Overlaid on the Autopsy window is a Windows Explorer window showing the path `Computer > OS (C:) > autopsy > build > testuserdir > python_modules`. The Explorer window displays a table with columns for Name, Date modified, Type, and Size. The first entry is `MySnazzyModule` with a date modified of `10/29/2014 12:07 ...` and a type of `File folder`.

Name	Modified Time	Change Time	Access
[current folder]	2012-01-20 18:19:25 EST	2012-01-20 18:19:25 EST	2012-

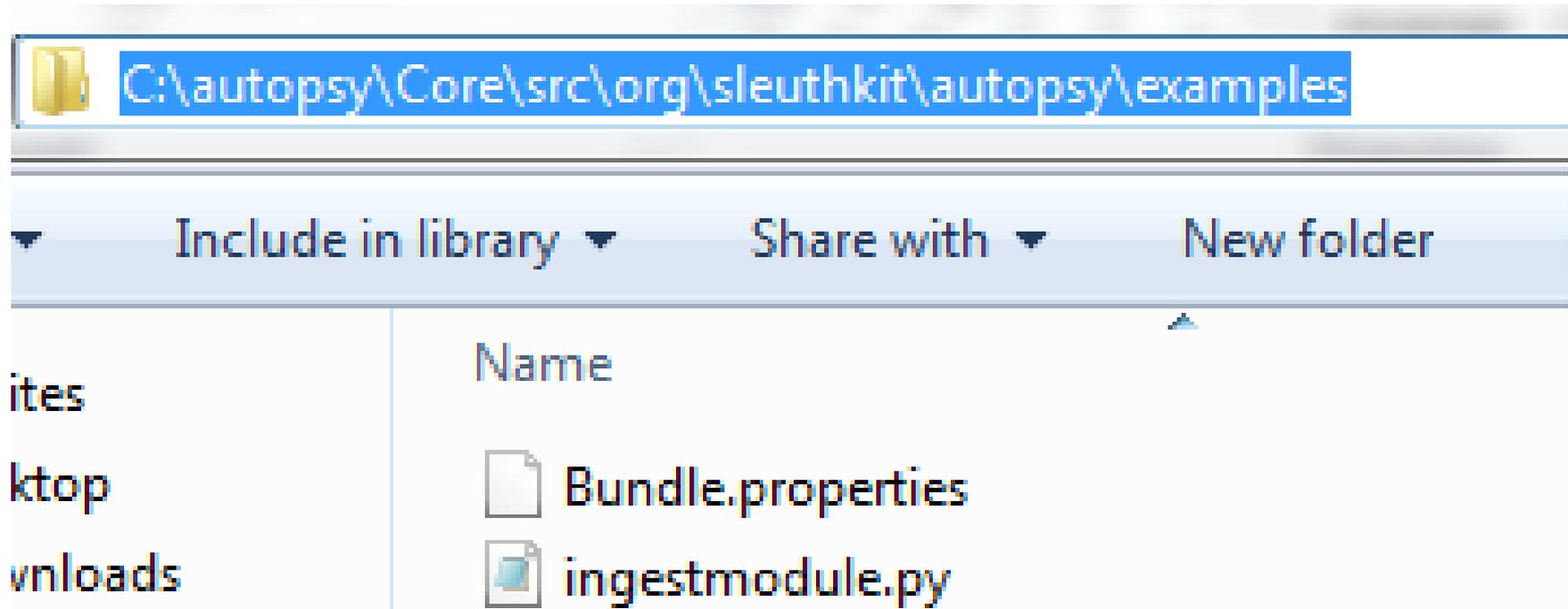
Name	Date modified	Type	Size
MySnazzyModule	10/29/2014 12:07 ...	File folder	

# Resources: SleuthKit Wiki

---

- [http://www.sleuthkit.org/autopsy/docs/api-docs/3.1/mod\\_dev\\_py\\_page.html](http://www.sleuthkit.org/autopsy/docs/api-docs/3.1/mod_dev_py_page.html)
- [http://www.sleuthkit.org/autopsy/docs/api-docs/3.1/platform\\_page.html](http://www.sleuthkit.org/autopsy/docs/api-docs/3.1/platform_page.html)
- [http://www.sleuthkit.org/autopsy/docs/api-docs/3.1/mod\\_ingest\\_page.html](http://www.sleuthkit.org/autopsy/docs/api-docs/3.1/mod_ingest_page.html)
- [http://www.sleuthkit.org/autopsy/docs/api-docs/3.1/mod\\_report\\_page.html](http://www.sleuthkit.org/autopsy/docs/api-docs/3.1/mod_report_page.html)

# Resources: Sample in source



# Getting file bytes

---

```
# Read the contents of the file.  
inputStream = ReadContentInputStream(file)  
buffer = jarray.zeros(1024, "b")  
len = inputStream.read(buffer)  
while (len != -1):  
    len = inputStream.read(buffer)
```

# The End (Questions?)

