ZyLAB Service Utility Tool Manual

Contact Us

For full contact details, visit the ZyLAB website - http://www.zylab.com

For support, visit our ZyLAB support website - http://support.zylab.com

Contents

ZyLAB Service Utility Tool

The ZyLAB Service Utility command line tool (ZyLAB.Platform.ServiceUtil.exe) is a part of the TBIEService solution and enables you to communicate with the following services (with default end point addresses) :

- Indexing (TBIE service)
  net.tcp://localhost:4112/IndexingService
- PlatformSDK
  net.tcp://localhost:4115/ZyIMAGE
- Content Access (CAS)
  net.tcp://localhost:4112/ContentAccess
- Cluster
  net.tcp://localhost:4115/Cluster
- Cluster Management
  net.tcp://localhost:4115/ClusterManagement

Use the tool to create/delete/rebuild/optimize indexes, search, retrieve text of the document from the index, and diagnose problems with platform services.

It can be found at \\Program Files\ZyLAB\Information Management Platform\Services\Tools

Commands

- The tool has several predefined commands that allow you to communicate with the listed services. To see the list of available commands run:

  ```
  ZyLAB.Platform.ServiceUtil.exe --help
  ```

- Use the following format to use this tool:

  ```
  ZyLAB.Platform.ServiceUtil.exe command [--option1 value1 [--option2 value2 ...]]
  ```

- Use the following syntax to get information about options for a specific command:

  ```
  ZyLAB.Platform.ServiceUtil.exe command --help
  ```

Command	Options	Description
action	**-a, --api** - (Default: tbie) API to execute command (tbie or sdk) **-i, --index** - Name of index (required when action name is specified) **-n, --name** - Name of action. Omit this parameter to get the list actions supported by TBIE service API. PlatformSDK service API supports following actions: validate, rebuild, delete **-u, --uri** - Service uri (if not set - default will be used)	Perform action on index.
build	**-a, --api** - (Default: tbie) API to execute command (tbie or sdk) **-i, --index** - Required. Name of index **-f, --force** - Delete existing index (this will prevent indexing and adding the same files twice) **-t, --template** - Template for index **-p, --path** - Required. Path to documents **-m, --mask** - Pattern to select files *.*, *.pdf, etc **-r, --repeat** - (Default: 1) Number of times documents should be indexed **-b, --batch** - (Default: 0) Build index in batches (0 means build all at once) **-u, --uri** - Service uri (if not set - default will be used)	Create or add documents to index.
cachesearch	**-i, --index** - Required. Name of index **-q, --query** - Required. Fulltext query **-u, --uri** - Service uri (if not set - default will be used)	Perform search and cache results using TBIE service API.

Command	Options	Description
content	**-i, --index** - Required. Name of index **-f, --file** - Required. File name of the document **-u, --uri** - Service uri (if not set - default will be used)	Generate the content of a document using TBIE service API.
deletedocuments	**-a, --api** - (Default: tbie) API to execute command (tbie or sdk) **-i, --index** - Required. Name of index **-d, --documents** - Required. Semicolon separated list of document IDs or a name of the file with the list of document IDs **-u, --uri** - Service uri (if not set - default will be used)	Delete documents from index.
hits	**-a, --api** - (Default: tbie) API to execute command (tbie or sdk) **-i, --index** - Required. Name of index **-q, --query** - Required. Fulltext query **-d, --document** - Required. Id of the document **-t, --timeout** - (Default: 2147483647) Operation timeout in ms (supported only when using TBIE service API) **-o, --output** - Name of the output file (outputs to stdout if not set) **-u, --uri** - Service uri (if not set - default will be used)	Get hits from index for specified document id. If timeout is specified - hits operation will be cancelled after the specified time (supported only when using TBIE service API).

Command	Options	Description

invoke	**-s, --service** - Required. Use service name to execute method with (TBIE, SDK, CAS, Cluster, ClusterManagement) or an interface name.
	
	-u, --uri - Service uri (if not set - default for the corresponding service will be used)
	
	-m, --method - Name of the method (if not specified - list of all available methods will be printed)
	
	-p, --parameters - Method parameters (file name or JSON directly)
	
	-o, --output - Name of the output file (outputs to stdout if not set)
		Invoke some method of service client. Parameters can be passed as JSON string directly or as a file name that contains it. If JSON string is passed directly - all single quotes (') are replaced with double quotes (") before parsing the string. JSON object passed as parameters is an array of key-value pairs where key is a name of the parameter required by the invoked method and value is a value of that parameter serialized to JSON.
		
		Simple parameters
		
		`{`
		`"param1": 123,`
		`"param2": "text"`
		`}`
		
		It is possible to specify a type hint for the parameter (for example, if invoked method expects an interface). In this case, the value should have additional parameter $type. For example, TBIE service client provides a method Search that accepts two parameters of types string and SearchInfo. SearchInfo is an abstract class and so a type hint is required to create an object of some specific type.
		
		Type hint
		
		`{`
		`"indexName": "3gb_text",`
		`"searchInfo": {`
		`"$type": "FulltextSearchInfo",`
		`"Query": "test1"`
		`}`
		`}`

Command	Options	Description
listindexes	**-a, --api** - (Default: tbie) API to execute command (tbie or sdk) **-u, --uri** - Service uri (if not set - default will be used)	List all available indexes.

processworkflow	**-l, --list** - List all supported workflow types (annotationanalyzer, burnfields, converttomultipage, converttotiff, htmlexport, textextraction, tifftopdf, tifftopng, transformimage)
	
	-p, --path - Path to file (required when not using -l or --list option)
	
	-w, --workflow - Workflow name (listed in workflow.xml file, default location c:\zylab data\ContentAccessService\Workflows)
	
	-t, --type - Workflow type (required when not using -l or --list option)
	
	-o, --output - Output filename (required when not using -l or --list option)
	
	-a, --arguments - Process workflow arguments (file name or JSON directly, required when not using -l or --list option)
	
	-u, --uri - Service uri (if not set - default will be used)
		Process document using CAS. Arguments can be passed as JSON string directly or as a file name that contains it. If JSON string is passed directly - all single quotes (') are replaced with double quotes (") before parsing the string. JSON object passed as argument should have a type hint. Example of arguments object passed for workflow type ConvertToMultiPage:
		
		Type hint
		
		`{`
		`"$type": "ConvertToMultiPageArguments",`
		`"ImageOutputFormat": "SearchablePdf",`
		`"Tifffiles": [`
		`"D:\\Temp\\2\\000001L2.TIF",`
		`"D:\\Temp\\2\\000001L3.TIF"`
		`]`
		`}`
		
		Output will always be a zip file. If output path does not end with .zip - it will be added.

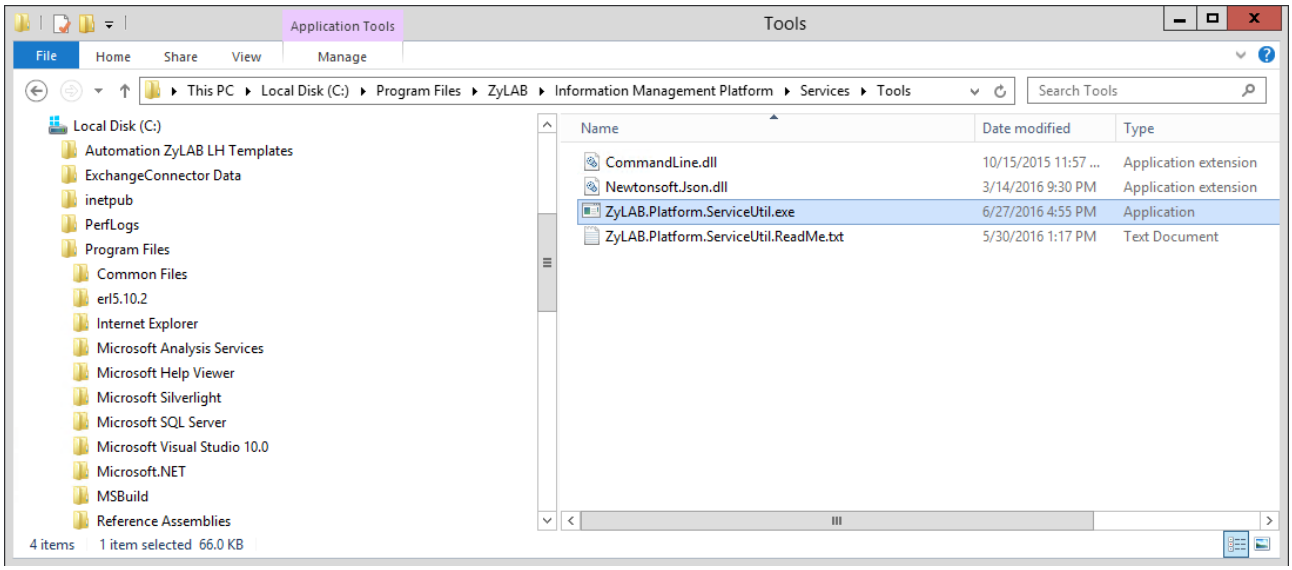
Command	Options	Description
read	**-t, --type** - Required. Stream type (search, hit, term, text) **-i, --input** - Required. Name of the input file (with raw stream data) **-o, --output** - Name of the output file (outputs to stdout if not set)	Read saved stream of the specified type from file and outputs the processed result. Supported stream types: • search (using ZyLAB.Platform.Services.Indexing.SearchStreamReader) • hit (using ZyLAB.Platform.Services.Indexing.IClientHitStreamReader) • term (using ZyLAB.Platform.Services.Indexing.TermStreamReader) • text (using ZyLAB.TBIE.Service.ContentAccess.TextStreamReader)
scan	**-i, --index** - Required. Name of index **-p, --pattern** - Required. Pattern **-t, --timeout** - (Default: 2147483647) Operation timeout in ms **-u, --uri** - Service uri (if not set - default will be used)	Scan index dictionary for terms matching to specified pattern using TBIE service API. If timeout parameter is set - scan operation will be cancelled after the specified time.

Command	Options	Description
search	**-a, --api** - (Default: tbie) API to execute command (tbie or sdk) **-i, --index** - Required. Name of index **-q, --query** - Required. Fulltext query **-d, --document** - Do not output internal document ID, will disable printing of document id in the results. Use together with -f option. **-h, --hits** - Do not output number of hits in the document, will disable printing of hits in the results. Used together with -f option. **-f, --fields** - Coma separated list of fields to retrieve (retrieves all fields if not set) **-s, --skip** - (Default: 0) Skip a specified number of search results. For example, when there are 10 results and you skip 2, you will only get the last 8. **-c, --count** - (Default: 18446744073709551615) Count returns not more than the specified number of results. **-t, --timeout** - (Default: 2147483647) Operation timeout in ms (supported only when using TBIE service API) **-o, --output** - Name of the output file (outputs to stdout if not set), where results will be printed. **-u, --uri** - Service uri (if not set - default will be used), define if the service is run on a different machine or port. **-b, --benchmark** - Benchmark mode (no results will be outputted), results will not be printed. It is used to measure speed and for debugging.	Search index. This command allows to skip a number of matches and get some specific number of matches. If timeout is specified - search operation will be cancelled after the specified time (supported only when using TBIE service API). It is possible to request system fields in '-f' option using @FieldName syntax (e.g. @ErrorCode, @HitDensity, etc.)

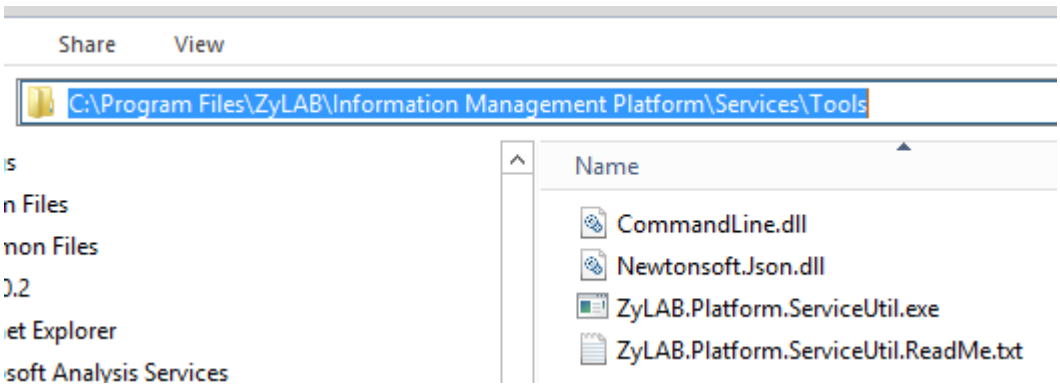
Using the Service Utility Tool

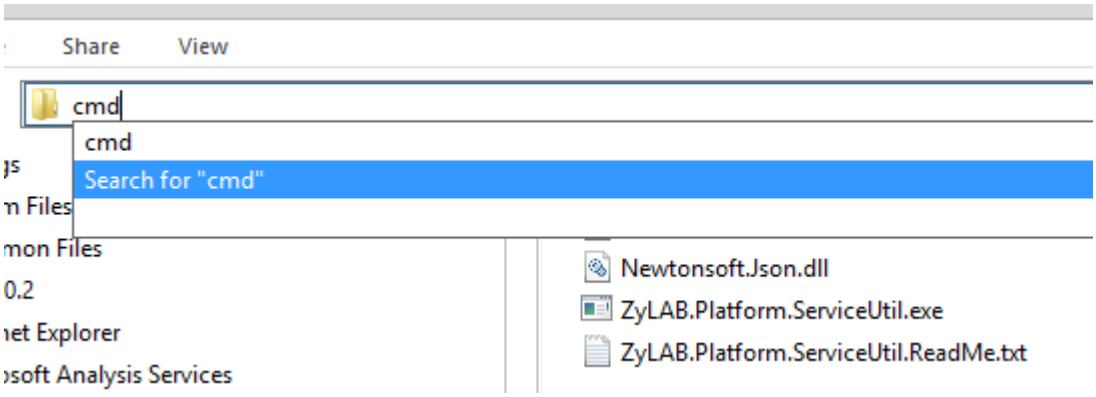
Get started

1. Go to \\Program Files\ZyLAB\Information Management Platform\Services\Tools and select ZyLAB.Platform.ServiceUtil.exe.

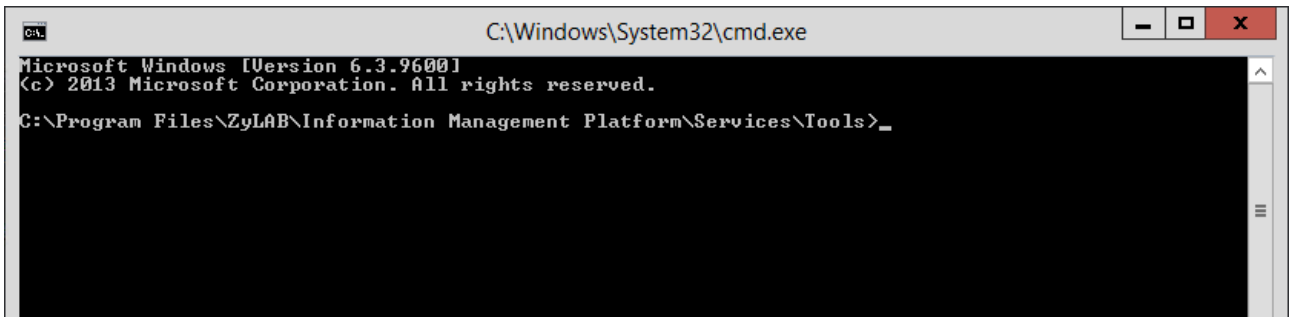


2. Click in the address bar, type cmd and press Enter.



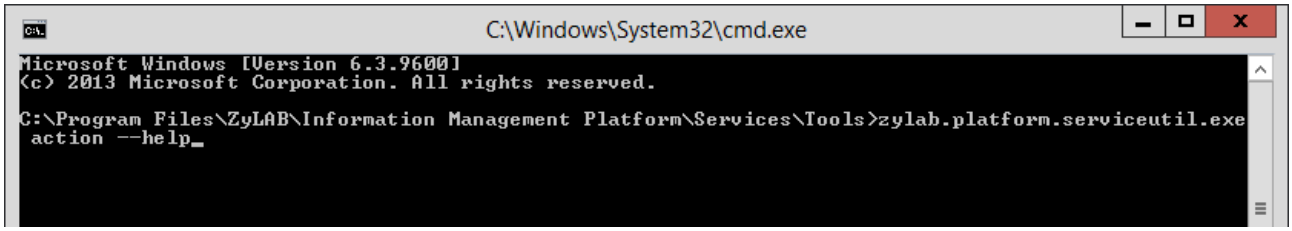


3. The command line tool is opened.

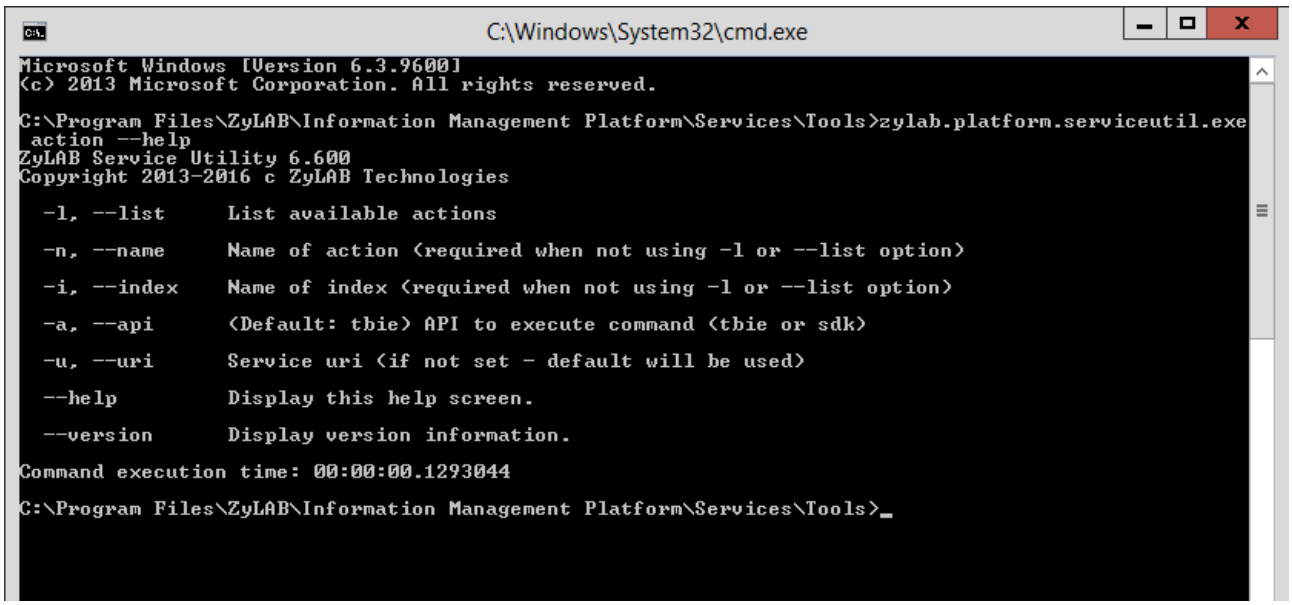


4. You can use 12 commands:
   - action
   - build
   - cachesearch
   - content
   - deletedocuments
   - hits
   - invoke
   - listindexes
   - processworkflow
   - read
   - scan
   - search
5. To view the options you can use with a command, run the following:

```
zylab.platform.serviceutil.exe command --help
```



6. A list with all the options for that command (in this example 'action') will be shown:



Tips
- Use the Up Arrow Key to insert a previously used command line
- Use double quotes when adding, for example, a file path with spaces: "c:\temp\test data"

Create an Index and Index Files

1. To create an index, use the build command.

   For example, to create an index with the name 'testindex' using files located on c:\temp, run the following:

   ```
   zylab.platform.serviceutil.exe build --index testindex --path c:\temp
   ```

   Result: You created a default TBIE index, located at C:\ZyLAB Data\TBIE Indexes\testindex.

2. You can add more options, for example:

   - --force
     To make sure a new index is created when using the same index name. This to prevent that the same files are indexed and added again to the existing index.
   - --mask
     To add only specfied files (*.*, *.pdf, *.txt, etc.).
   - --repeat
     To define how many times documents should be indexed (default 1).
   - --batch
     To create several batches (default 0, which means build all at once).

     You can, for example, run the following:

     ```
     zylab.platform.serviceutil.exe build --index testindex --force --
     path c:\temp --mask *.pdf --repeat 2 --batch 2
     ```

     This would create an index called 'testindex', containing only .pdf files from the c:\temp location (indexed twice) and saving those indexed files in batches.

The Repeat option will index the files two times: file1, file2, file3, file1, file2, file3
The Batch option will index the files in parts (each part consisting of two files): file1, file2 / file3, file1 / file2, file3
The result will be presented as one index.

Notes

- The Default template is used when creating an index. Other templates that can be used are located at C:\ZyLAB Data\SearchEngine\Templates (or C:\ZyLAB Data\TBIE Templates). HAPI index templates can be found at C:\ZyLAB Data\ZyINDEX Templates.

  For example, use the audio template:

  ```
  zylab.platform.serviceutil.exe build --index testindex --template
  audio --path c:\temp
  ```

  Or, create a HAPI index using the Audit Trail Database template:

  ```
  zylab.platform.serviceutil.exe build --api sdk --index testindex --
  template "c:\zylab data\zyindex templates\audit trail database" --
  path c:\temp
  ```

- You can only open HAPI indexes (created with the -a sdk option) in ZyINDEX.

Delete an Index

To delete an index, use the action command.

For example, to delete an index with the name 'testindex', run the following:

```
zylab.platform.serviceutil.exe action -i testindex -n delete
```

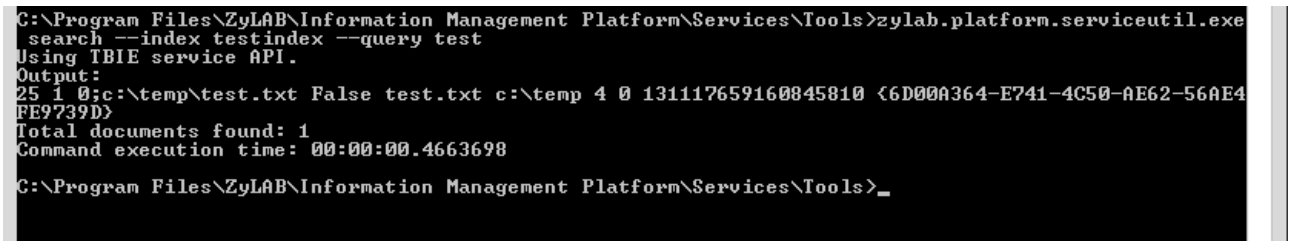
Or, if the index was created with the `-a sdk` parameter:

```
zylab.platform.serviceutil.exe action -a sdk -i testindex -n delete
```

Search files

- To search files, use the search command. For example, to search for 'test', run the following:

```
zylab.platform.serviceutil.exe search --index testindex --query
test
```



- When you search for a field value, run the following:

```
zylab.platform.serviceutil.exe search --index testindex --query
test -f file_name
```

The field results are preceded by the document id and the number of hits. If you do not want to see them, run the following:

```
zylab.platform.serviceutil.exe search --index testindex --query
test -f file_name --document --hits
```

- When you want to search and cache the results, run the following:

```
zylab.platform.serviceutil.exe cachesearch --index testindex --
query test
```

Next time when the search command is used, results will be taken from cache.

Run Multiple Searches in Parallel

- Use the following command (to simulate high load on the search engine):

```
FOR /L %G IN (1,1,10) DO start ZyLAB.Platform.ServiceUtil.exe
search -a sdk -i "INDEXNAME" -q "QUERY"
```

- Use the %G parameter to randomize your queries. For example like this:

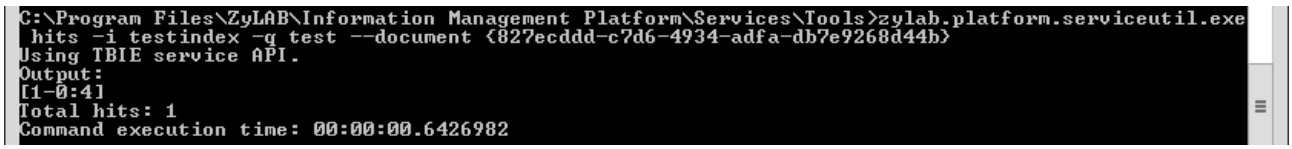
```
FOR /L %G IN (1,1,10) DO start ZyLAB.Platform.ServiceUtil.exe
search -a sdk -i "INDEXNAME" -q "[a-z]{1,%G}"
```

Get Hits

To get the number of hits in a file, use the hits command.

For example, to get the number of hits for the query 'test' in a specific file in index 'testindex', run the following:

```
zylab.platform.serviceutil.exe hits -i testindex -q test --document
{827ecddd-c7d6-4934-adfa-db7e9268d44b}
```



One hit was found.

View Content

To view the content of a file, use the content command.

For example, run the following:

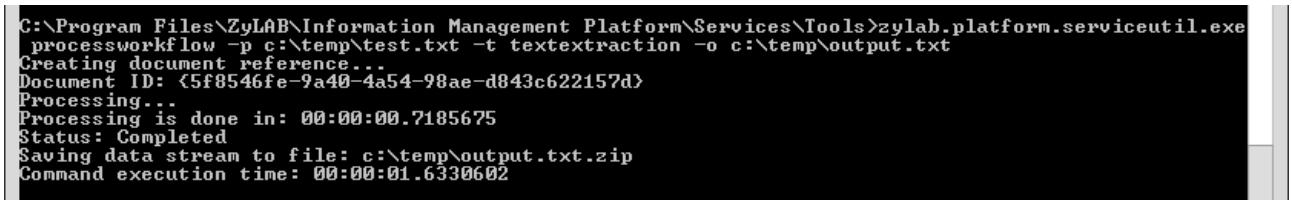
```
zylab.platform.serviceutil.exe content -i testindex -file
c:\temp\test.txt
```

Extract Text

To extract text, use the processworkflow command.

For example, run the following:

```
zylab.platform.serviceutil.exe processworkflow -p c:\temp\test.txt -t
textextraction -o c:\temp\output.txt
```



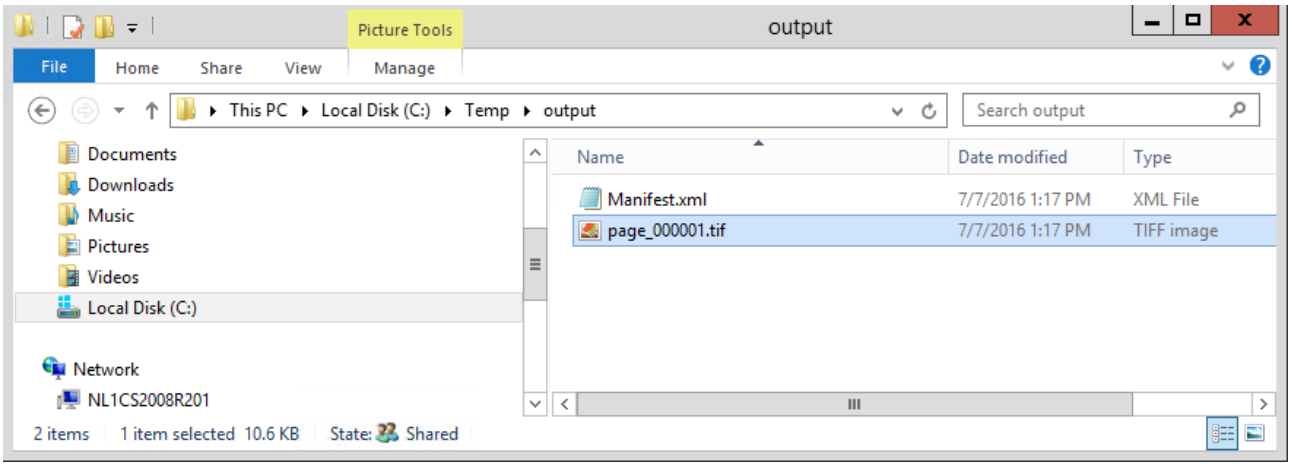
The output file will always be saved as a zipfile.

Generate Tiff (or png or jpg) of document

- To generate a tiff of a document, use the processworkflow command.

  For example, run the following:

  ```
  zylab.platform.serviceutil.exe processworkflow -p c:\temp\test.txt
  -t converttotiff -o c:\temp\output
  ```



- Another example for an inputfile made accessible to ContentAccess service:

  ```
  zylab.platform.serviceutil.exe processworkflow -t converttotiff -p
  "INPUTFILE" -o result.zip
  ```

  You can use Content Access service located on another machine by specifying url with -u argument.

- To provide specific tiff conversion arguments (for example, convert specific pages of the document), run the following:

  ```
  ZyLAB.Platform.ServiceUtil.exe processworkflow  -t ConvertToTiff -p
  "INPUTFILE" -o result.zip -a
  "{\"$type\":\"ZyLAB.TBIE.Service.ContentAccess.ConvertToTiffArgumen
  ts\",UsePageRange:true,StartPage:2,EndPage:3}"
  ```

- To convert to tiff all files in a given folder, run the following:

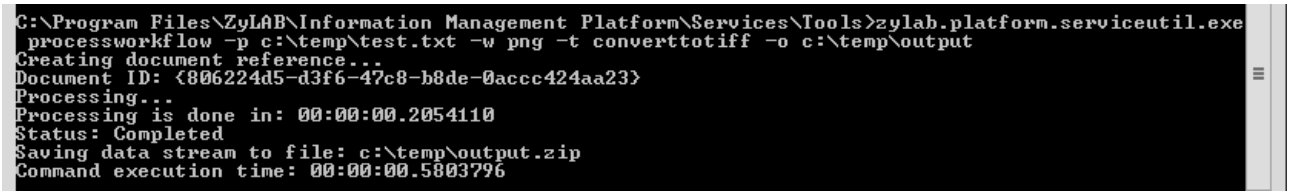
  ```
  FOR /R "<FOLDER WITH FILES>" %I IN (*.*) DO
  ```

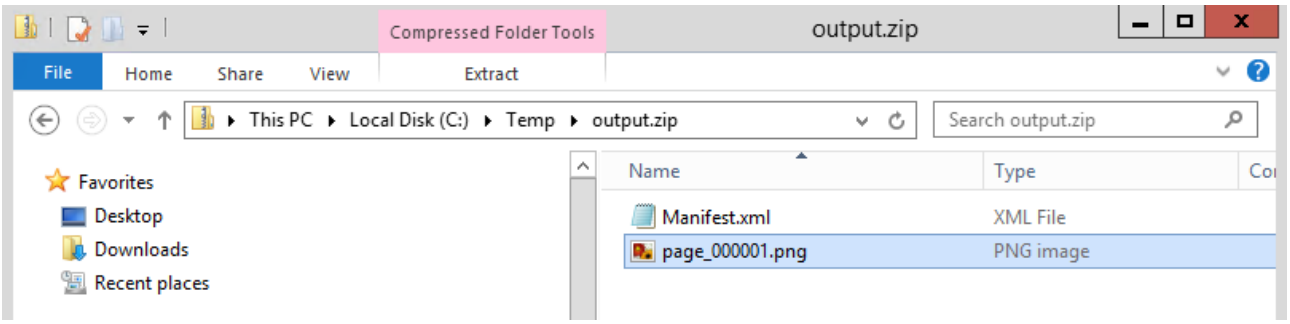
```
ZyLAB.Platform.ServiceUtil.exe processworkflow -t ConvertToTiff -p
"%I" -o "%~nxI.zip"
```

Generate PNG or JPG file

In the workflow.xml file (default location c:\zylab data\ContentAccessService\Workflows) you can see if there are workflow names. For the option converttotiff, we have the following workflow names: png and jpg. Use them to convert a file to png or jpg. For example, run the following:

```
zylab.platform.serviceutil.exe processworkflow -p c:\temp\test.txt -w png
-t converttotiff -o c:\temp\output
```





View a list of all indexes

To view a list of all indexes, use the listindexes command.

```
ZyLAB.Platform.ServiceUtil.exe listindexes
```