Schema Flattener

Structured XSD to Flat Monolithic XSD

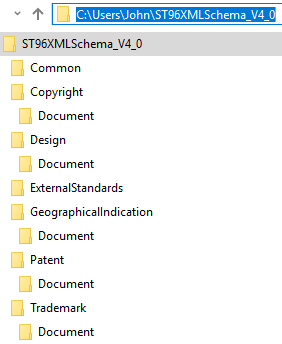
(For WIPO Compliant Design/Structured Schemas)

## Introduction

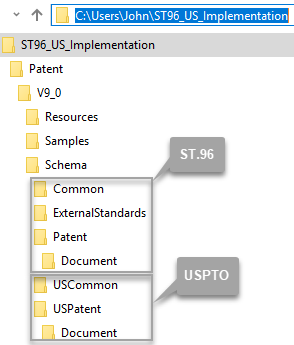
Schema Flattener utility is written in Java and intended to flatten World Intellectual Property Office (WIPO) complaint Design/Structured XML Schema Definition (XSD) files.

## Key Assumptions

* Input schemas (XSDs) are WIPO compliant Design/Structured format
  + Typical WIPO ST.96 Design/Structure Schema:



* + Typical WIPO Compliant IPO Design/Structure Schema:



* + Validated (free of errors)
* Output folder has privileges to write
* Please confirm validity of each output file by opening in any XSD editor for additional assurance
* Java runtime 1.8 +
  + Should be able to run java command from command-line

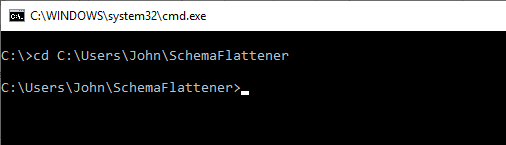
## Installation

No special installation is needed. Just download or copy “SchemaFlattener.jar” or “SchemaFlattener.zip” which contains a jar file and this instructions document in “SchemaFlattener” folder, to any desired location/folder.

From Windows Command-line terminal –

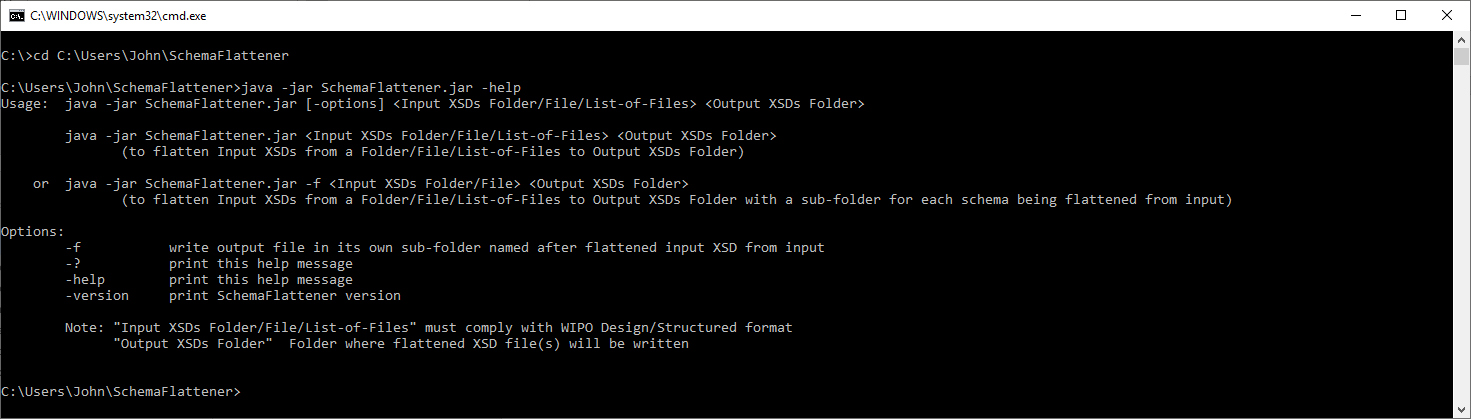
Navigate to installed (jar file) location. For example: C:\John\SchemaFlattener installation folder

Type cd C:\Users\John\SchemaFlattener



## Usage

Type “java -jar SchemaFlattener.jar -help” at command-line



### Supported Modes of Input from Command-line

There are 3 ways of passing input (from command-line) to this utility is supported.

1. Entire Set of Document Level Schemas Flattening (every Business Area)

*Note: Only this mode generates monolithic schemas for every business area*

1. Entire Set of Business Specific Document Level Schemas Flattening
2. Just one or multiple specific Component(s)/Document Level Schema(s) Flattening

### Supported Modes of Output from Command-line

There are 2 types of output generation is supported.

1. Write all flattening schema(s) to output folder with no sub-folder(s) – Default option
2. Write every flattening schema(s) to output folder with respective sub-folder(s) which contains complete set of dependent schemas and dependency report (Readme.txt) with details

### Entire Set of Document Level Schemas Flattening

#### To create Flattened Schemas for a Complete set of WIPO Release Design Schemas – output type-1 (no sub-folders)

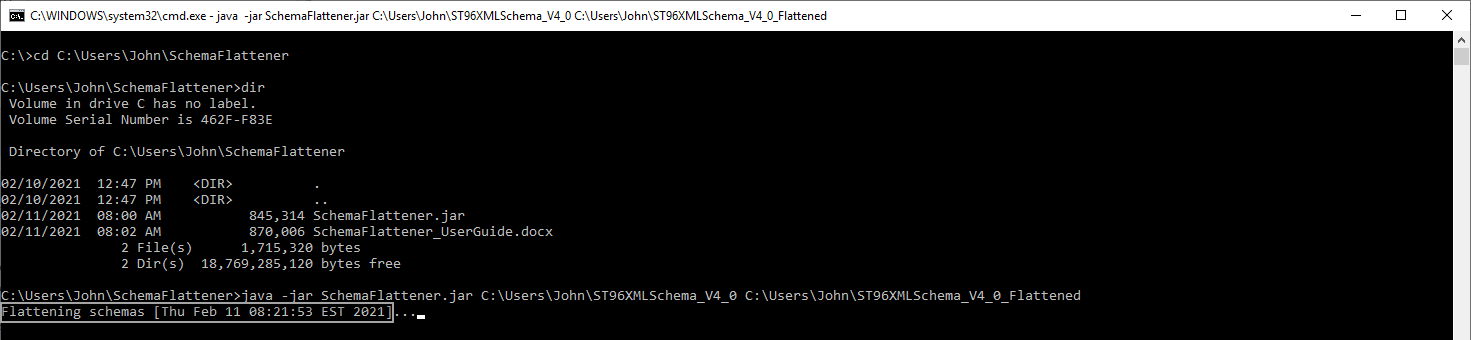
Example: For Release 4.0

Input folder of Design Schemas: C:\Users\John\ST96XMLSchema\_V4\_0

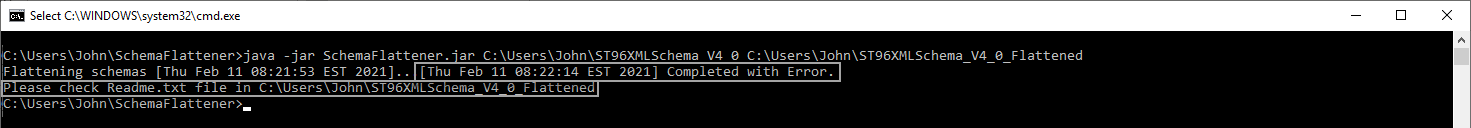
Output folder for Flattened Schemas: C:\Users\John\ST96XMLSchema\_V4\_0\_Flattened

*Note: Files in Output folder will always be overwritten. Please backup/clear before every run if reusing same output folder between runs.*

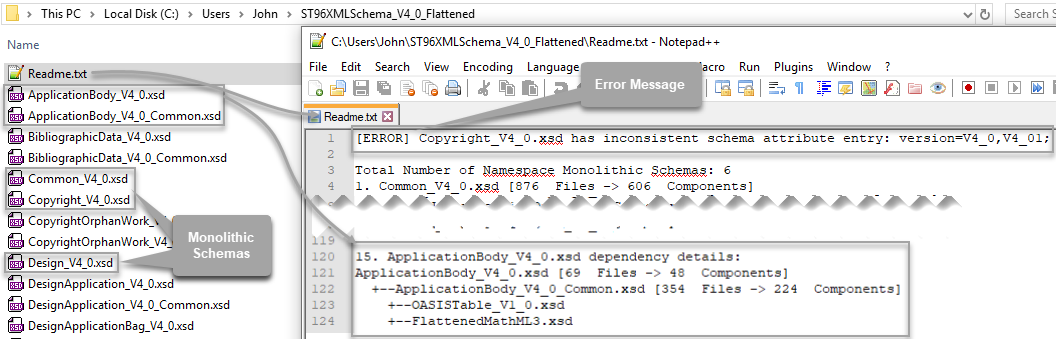
Type command “java -jar SchemaFlattener.jar C:\Users\John\ST96XMLSchema\_V4\_0 C:\Users\John\ST96XMLSchema\_V4\_0\_Flattened”



Upon successful completion you should see a “Completed” message with timestamp. Or “Completed with Error” message with timestamp similar to the image below: (Note: In ST.96 V4.0 one of the Copyright Namespace XSD has incorrect version attribute value “V4\_01”. This will result an invalid flattened XSDs)



Output folder (C:\Users\John\ST96XMLSchema\_V4\_0\_Flattened) should have all WIPO Document level schemas along with **Readme.txt** file which contains all generated document level schemas list and their relational dependant referring schemas including any errors during flattening process. All business area monolithic schemas will also be there in output folder.



#### To create Flattened Schemas for a Complete set of WIPO Release Design Schemas – output type-2 (respective sub-folders for every flattened schema)

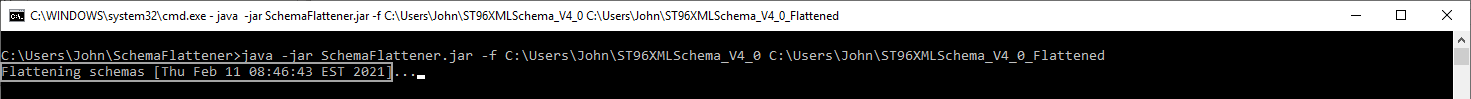
Example: For Release 4.0

Input folder of Design Schemas: C:\Users\John\ST96XMLSchema\_V4\_0

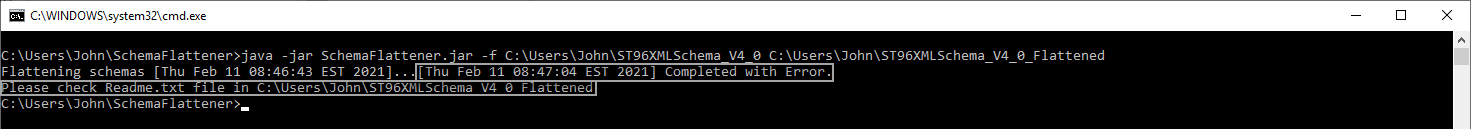
Output folder for Flattened Schemas: C:\Users\John\ST96XMLSchema\_V4\_0\_Flattened

*Note: Files in Output folder will always be overwritten. Please backup/clear before every run if reusing same output folder between runs.*

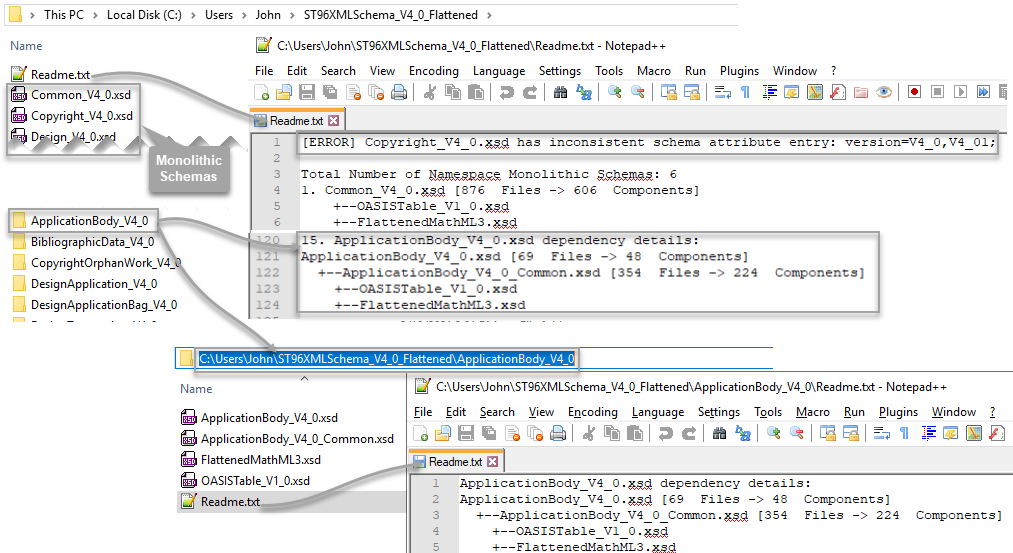
Type command “java -jar SchemaFlattener.jar -f C:\Users\John\ST96XMLSchema\_V4\_0 C:\Users\John\ST96XMLSchema\_V4\_0\_Flattened”



Upon successful completion you should see a “Completed” message with timestamp. Or “Completed with Errors” message with timestamp similar to the image below: (Note: In ST.96 V4.0 one of the Copyright Namespace XSD has incorrect version attribute value “V4\_01”. This will result an invalid flattened XSDs)



Output folder (C:\Users\John\ST96XMLSchema\_V4\_0\_Flattened) should have all WIPO Document level schemas (in Document folder for every business area) along with a **Readme.txt** file which contains all generated document level schemas list and their dependant referring schemas that are nested in their own sub-folder including any errors during flattening process.



### Entire Set of Business Specific Document Level Schemas Flattening

#### To crate Flattened Schema for an entire business area – output type-1 (no sub-folders)

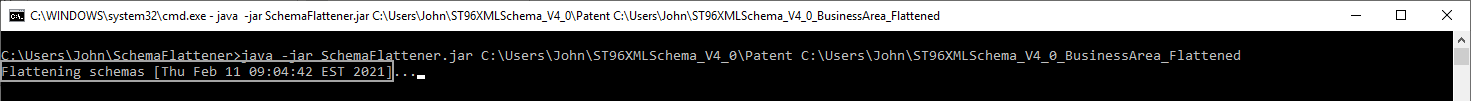
Example: Entire Patent namespace (Utility will automatically detect all Document level schemas)

Input Component file from Design Schemas: C:\Users\John\ST96XMLSchema\_V4\_0\Patent

Output folder for Flattened Schemas: C:\Users\John\ST96XMLSchema\_V4\_0\_BusinessArea\_Flattened

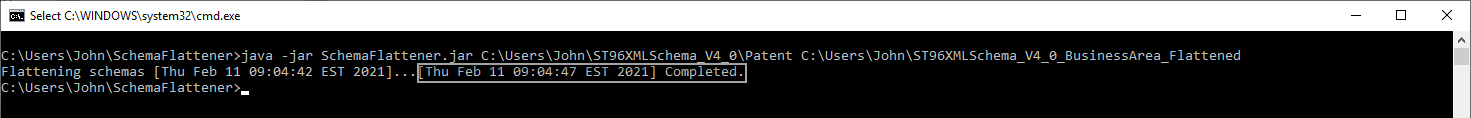
*Note: Files in Output folder will always be overwritten. Please backup/clear before every run if reusing same output folder between runs.*

Type command “java -jar SchemaFlattener.jar C:\Users\John\ST96XMLSchema\_V4\_0\Patent C:\Users\John\ST96XMLSchema\_V4\_0\_BusinessArea\_Flattened”

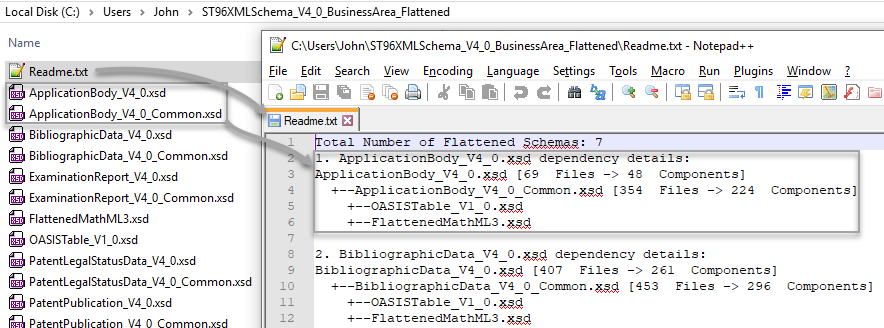


Upon successful completion you should see a “Completed” message with timestamp.

*Note: Depending on your case if source schemas has any errors, you may see “Completed with Error” or “Completed with Errors” message with timestamp. Allways check Readme.txt file for any errors during flattening process.*



Output folder (C:\Users\John\ST96XMLSchema\_V4\_0\_BusinessArea\_Flattened) should have all Patent Document level schemas (in Document folder for every business area) along with a **Readme.txt** file which contains all generated document level schemas list and their dependant referring schemas including any errors during flattening process.



#### To crate Flattened Schema for an entire business area – output type-2 (respective sub-folders for every flattened schema)

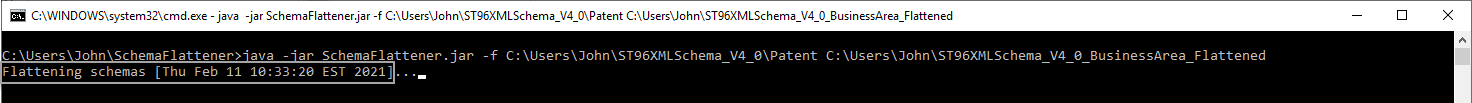
Example: Entire Patent namespace (Utility will automatically detect all Document level schemas)

Input Component file from Design Schemas: C:\Users\John\ST96XMLSchema\_V4\_0\Patent

Output folder for Flattened Schemas: C:\Users\John\ST96XMLSchema\_V4\_0\_BusinessArea\_Flattened

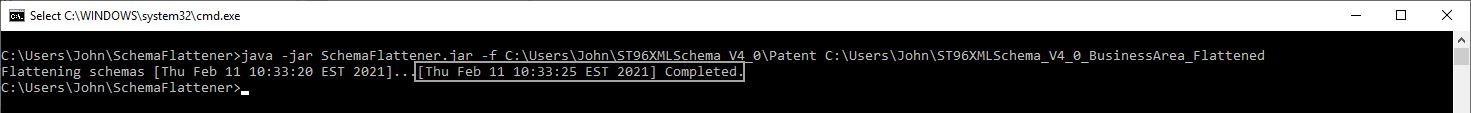
*Note: Files in Output folder will always be overwritten. Please backup/clear before every run if reusing same output folder between runs.*

Type command “java -jar SchemaFlattener.jar -f C:\Users\John\ST96XMLSchema\_V4\_0\Patent C:\Users\John\ST96XMLSchema\_V4\_0\_BusinessArea\_Flattened”

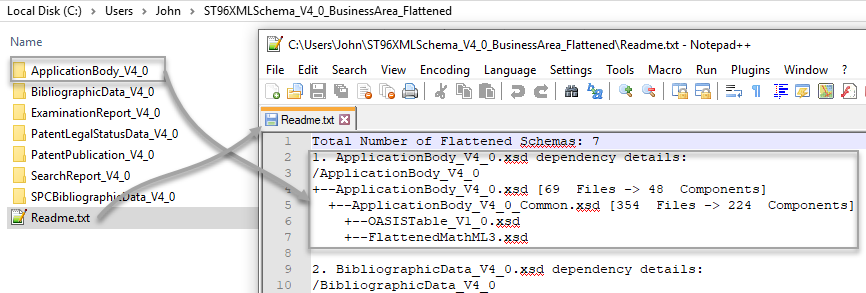


Upon successful completion you should see a “Completed” message with timestamp.

*Note: Depending on your case if source schemas has any errors, you may see “Completed with Error” or “Completed with Errors” message with timestamp. Allways check Readme.txt file for any errors during flattening process.*



Output folder (C:\Users\John\ST96XMLSchema\_V4\_0\_BusinessArea\_Flattened) should have all Patent Document level schemas (in Document folder for every business area) that are nested in their own sub-folder along with a **Readme.txt** file which contains all generated document level schemas list and their dependant referring schemas including any errors during flattening process.



### Just one or multiple specific Component(s)/Document Level Schema(s) Flattening

#### To crate Flattened Schema for one Individual Component – output type-1 (no sub-folders)

Example: Claims.xsd from Patent namespace

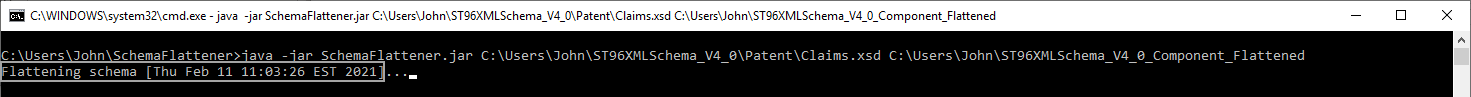
Input Component file from Design Schemas: C:\Users\John\ST96XMLSchema\_V4\_0\Patent\ Claims.xsd

*Note: Input can be one or more component XSDs, Incase of multiple please use comma separated list*

Output folder for Flattened Schemas: C:\Users\John\ST96XMLSchema\_V4\_0\_Component\_Flattened

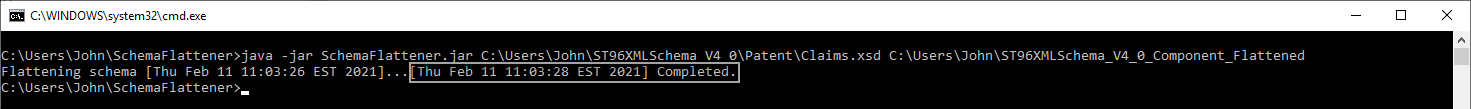
*Note: Files in Output folder will always be overwritten. Please backup/clear before every run if reusing same output folder between runs.*

Type command “java -jar SchemaFlattener.jar C:\Users\John\ST96XMLSchema\_V4\_0\Patent\Claims.xsd C:\Users\John\ST96XMLSchema\_V4\_0\_Component\_Flattened”

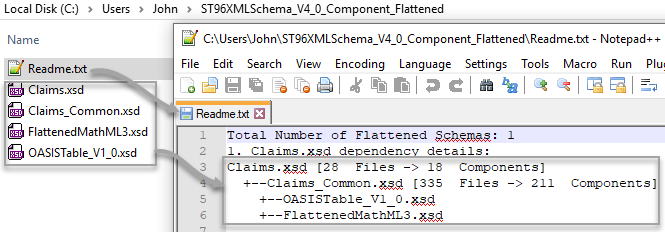


Upon successful completion you should see a “Completed” message with timestamp.

*Note: Depending on your case if source schemas has any errors, you may see “Completed with Error” or “Completed with Errors” message with timestamp. Allways check Readme.txt file for any errors during flattening process.*



Output folder (C:\Users\John\ST96XMLSchema\_V4\_0\_Component\_Flattened) should have flattened Claims schema and its dependent referring schemas along with **Readme.txt** file which contains dependant referring schemas information including any errors during flattening process.



#### To crate Flattened Schema for one Individual Component – output type-2 (respective sub-folders for every flattened schema)

Example: Claims.xsd from Patent namespace

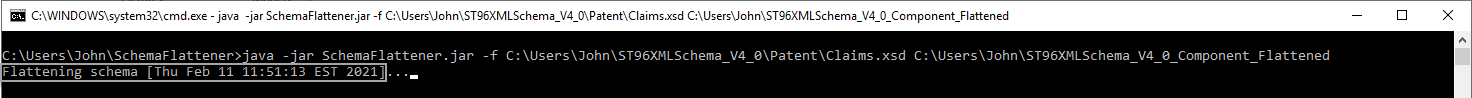
Input Component file from Design Schemas: C:\Users\John\ST96XMLSchema\_V4\_0\Patent\ Claims.xsd

*Note: Input can be one or more component XSDs, Incase of multiple please use comma separated list*

Output folder for Flattened Schemas: C:\Users\John\ST96XMLSchema\_V4\_0\_Component\_Flattened

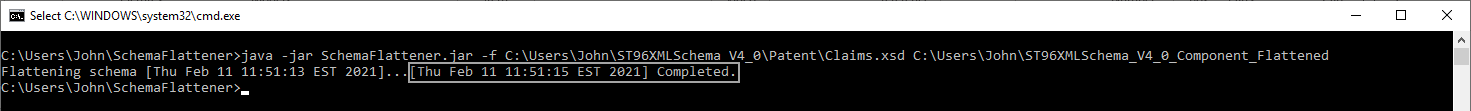
*Note: Files in Output folder will always be overwritten. Please backup/clear before every run if reusing same output folder between runs.*

Type command “java -jar SchemaFlattener.jar -f C:\Users\John\ST96XMLSchema\_V4\_0\Patent\Claims.xsd C:\Users\John\ST96XMLSchema\_V4\_0\_Component\_Flattened”

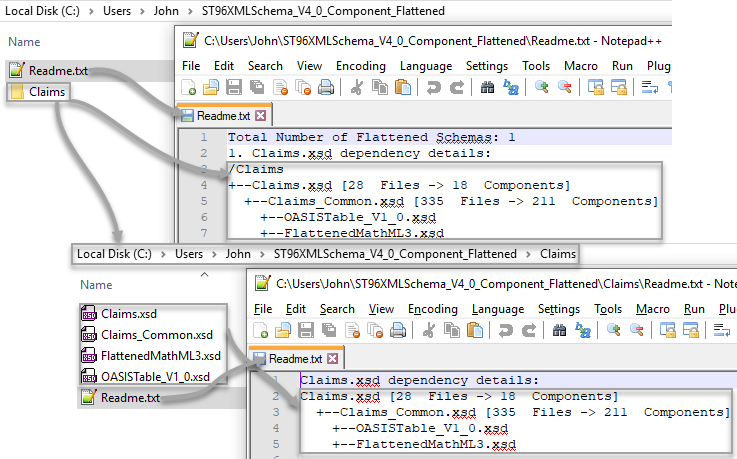


Upon successful completion you should see a “Completed” message with timestamp.

*Note: Depending on your case if source schemas has any errors, you may see “Completed with Error” or “Completed with Errors” message with timestamp. Allways check Readme.txt file for any errors during flattening process.*



Output folder (C:\Users\John\ST96XMLSchema\_V4\_0\_Component\_Flattened) should have flattened Claims schema and its dependent referring schemas that are nested in its own sub-folder along with **Readme.txt** file which contains dependant referring schemas information including any errors during flattening process.



#### To crate Flattened Schema for multiple Individual Components – output type-1 (no sub-folders)

Example: Claims.xsd from Patent namespace and AmendedBasis.xsd from Trademark namespace

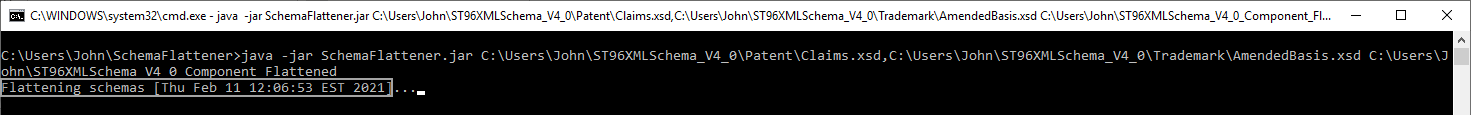
Input Component file from Design Schemas: C:\Users\John\ST96XMLSchema\_V4\_0\Patent\ Claims.xsd,C:\Users\John\ST96XMLSchema\_V4\_0\Trademark\AmendedBasis.xsd

*Note: Input can be one or more component XSDs, Incase of multiple please use comma separated list*

Output folder for Flattened Schemas: C:\Users\John\ST96XMLSchema\_V4\_0\_Component\_Flattened

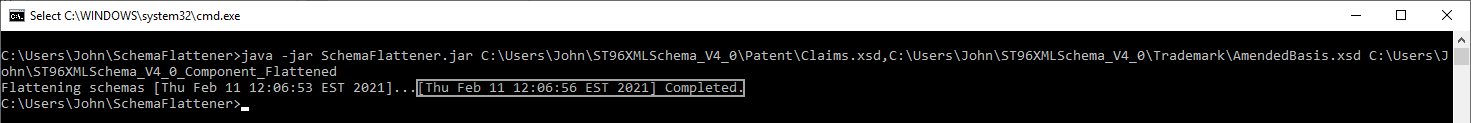
*Note: Files in Output folder will always be overwritten. Please backup/clear before every run if reusing same output folder between runs.*

Type command “java -jar SchemaFlattener.jar C:\Users\John\ST96XMLSchema\_V4\_0\Patent\Claims.xsd,C:\Users\John\ST96XMLSchema\_V4\_0\Trademark\AmendedBasis.xsd C:\Users\John\ST96XMLSchema\_V4\_0\_Component\_Flattened”

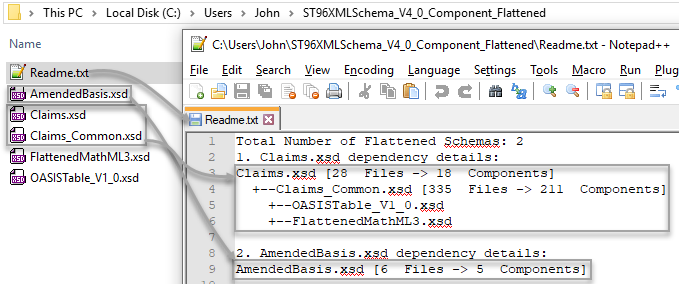


Upon successful completion you should see a “Completed” message with timestamp.

*Note: Depending on your case if source schemas has any errors, you may see “Completed with Error” or “Completed with Errors” message with timestamp. Allways check Readme.txt file for any errors during flattening process.*



Output folder (C:\Users\John\ST96XMLSchema\_V4\_0\_Component\_Flattened) should have flattened Claims, AmendedBasis schemas and their dependent referring schemas along with **Readme.txt** file which contains dependant referring schemas information including any errors during flattening process.



#### To crate Flattened Schema for multiple Individual Components – output type-2 (respective sub-folders for every flattened schema)

Example: Claims.xsd from Patent namespace and AmendedBasis.xsd from Trademark namespace

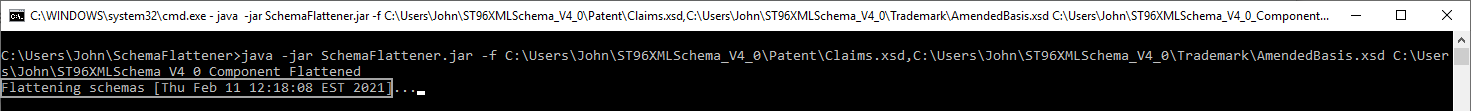
Input Component file from Design Schemas: C:\Users\John\ST96XMLSchema\_V4\_0\Patent\ Claims.xsd,C:\Users\John\ST96XMLSchema\_V4\_0\Trademark\AmendedBasis.xsd

*Note: Input can be one or more component XSDs, Incase of multiple please use comma separated list*

Output folder for Flattened Schemas: C:\Users\John\ST96XMLSchema\_V4\_0\_Component\_Flattened

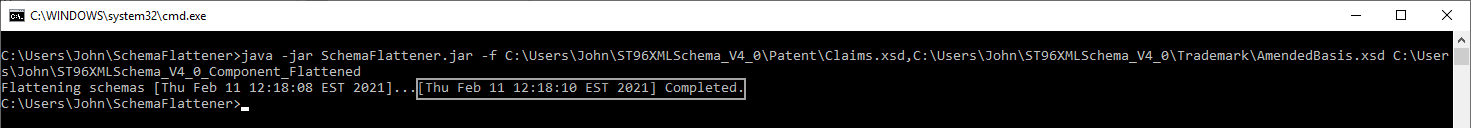
*Note: Files in Output folder will always be overwritten. Please backup/clear before every run if reusing same output folder between runs.*

Type command “java -jar SchemaFlattener.jar -f C:\Users\John\ST96XMLSchema\_V4\_0\Patent\Claims.xsd,C:\Users\John\ST96XMLSchema\_V4\_0\Trademark\AmendedBasis.xsd C:\Users\John\ST96XMLSchema\_V4\_0\_Component\_Flattened”

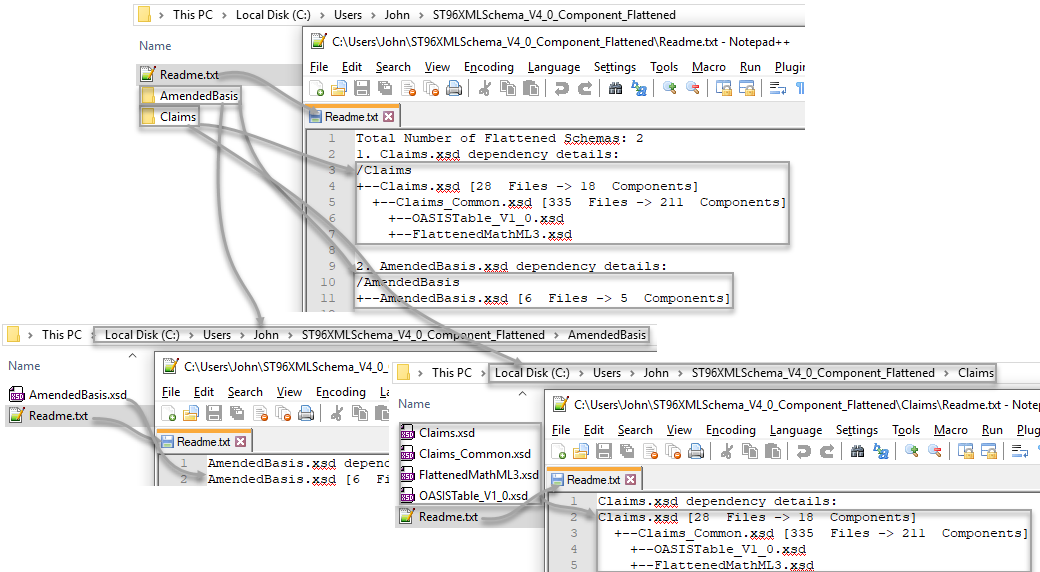


Upon successful completion you should see a “Completed” message with timestamp.

*Note: Depending on your case if source schemas has any errors, you may see “Completed with Error” or “Completed with Errors” message with timestamp. Allways check Readme.txt file for any errors during flattening process.*



Output folder (C:\Users\John\ST96XMLSchema\_V4\_0\_Component\_Flattened) should have flattened Claims, AmendedBasis schemas and their dependent referring schemas that are nested in their own sub-folders along with **Readme.txt** file which contains dependant referring schemas information including any errors during flattening process.



## Frequently Asked Questions (FAQ)

1. **What is Readme.txt file in my output file?**

Readme.txt file is a brief summary of each run of your flattening schema.

It lists –

* all the actions happened during flattening
* any errors during flattening (known exceptions)
  + Runtime exception might show up in terminal depending on your environment

Note: Readme.txt file will be overwritten for every run if you reuse output folder between different runs.

1. **Do Monolithic Common or Business Area schemas in my output folder are being referenced in any of my flattened schemas?**

No. All Business Area and Common monolithic schemas are generated for inventory purposes only. Except that these are inter related between them but not to any flattened schemas.

1. **Can I use this utility to flatten IPO Implemented XSDs?**

Yes. Any WIPO Compliant Design/Structured format IPO Implementation schemas can be flattened.

1. **Do I have to do anything extra to flatten IPO Implementation XSDs?**

No. Make sure your IPO Implementation schemas are in line with WIPO recommendation and compliant and its structure follows like “Typical WIPO Compliant IPO Design/Structure Schema” example in “Key Assumptions” section in this document.

1. **Can I use this utility to flatten any XSDs?**

Yes, if the XSDs are WIPO Compliant Design/Structured format and validated with no errors.

1. **Why do I see MathML and OASIS XSDs in my flattened XSD(s) output folder?**

MathML and OASIS schemas are external referencing schemas to many common/business specific components. If your schema being flattened refers these external schemas in their chain of dependency, they will be automatically copied to your output folder as dependent schemas. If your flattening schema has no dependency to any of the external schemas, they will not be in your output folder.