

# Python X3D Implementation Progress

Don Brutzman

X3D Working Group, Web3D Consortium

[brutzman@nps.edu](mailto:brutzman@nps.edu)

26 August 2019

# Motivation

- Create fully functional support for X3D Graphics using the Python programming language.
- As with other X3D language bindings, capabilities include the requirements of X3D Scene Access Interface (SAI) ISO/IEC 19775-2.
- We openly design, implement, evaluate and specify all aspects of this work using Web3D Consortium [Standards Adoption Process](#).
- When initial implementation and complete draft specification exist, Web3D reviews suitability for New Work Item Proposal (NWIP).
- Package designers include Loren Peitso, John Carlson, Masaki Aono, Myeong Won Lee, and Don Brutzman.

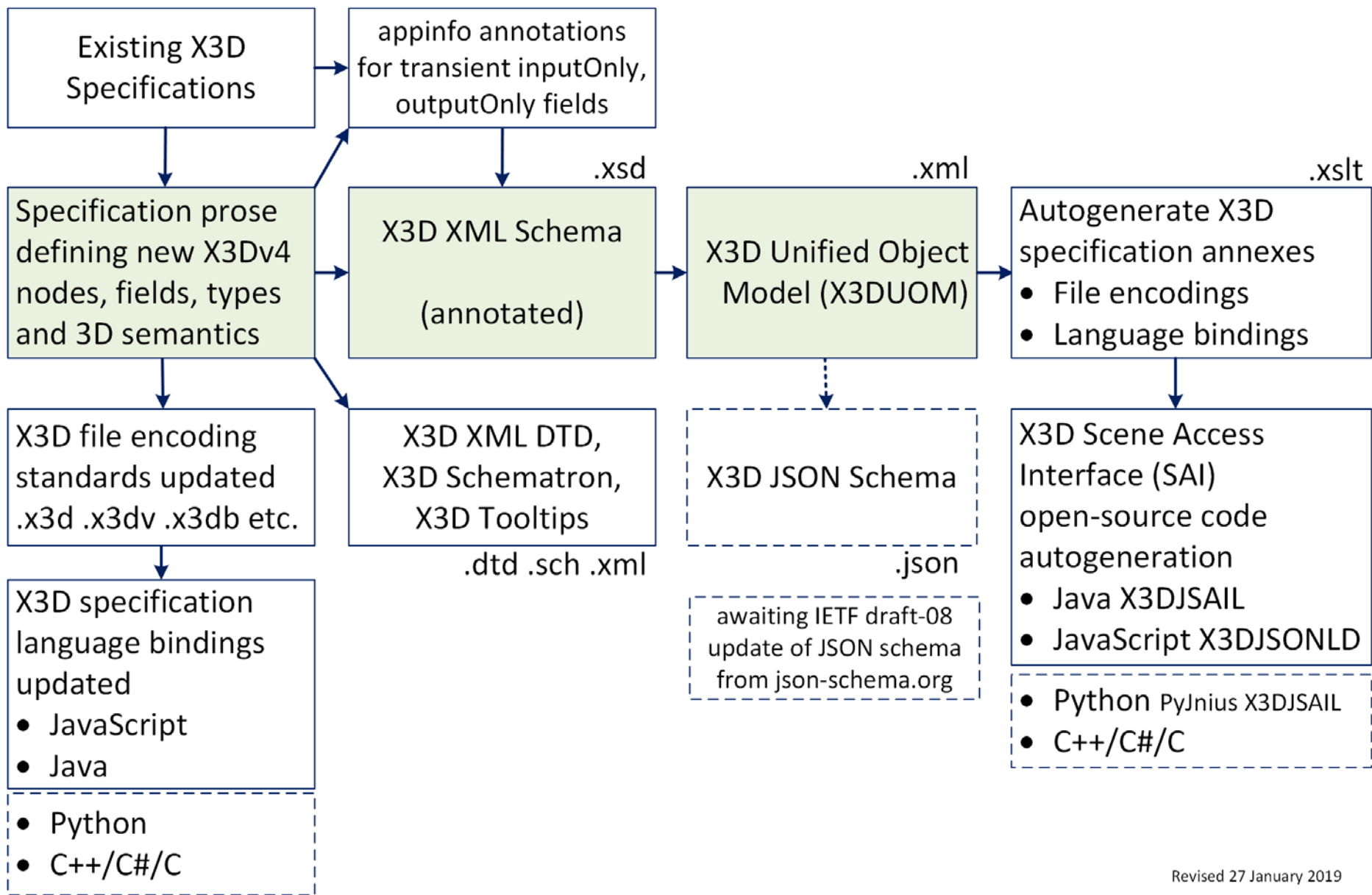
# Background: X3DJSAIL + Pyjnius efforts

- Given a number of unique features in Python, creating a 19777-6 draft requires a working implementation to resolve a number of design issues.
- Much effort has occurred over the past 18 months, with a long developmental period using X3DJSAIL and Pyjnius to implement X3D Python using Java. However we only achieved 80% success, primarily due to difficulties getting two codebases to align successfully.
- The X3DJSAIL-Pyjnius approach is no longer considered a viable potential path for implementation.

# Design

- [X3D Unified Object Model \(X3DUOM\)](#) is used to autogenerate the entire x3d.py package, ensuring that it is always up-to-date with the latest X3D Architecture.
- The current implementation strives to follow first principles of python package design, being as "pythonic" as possible.
- Python is exceedingly rigorous regarding design and implementation, demands crisp readability and often leads to a single best approach for each programming task.
- Python is deliberately different than imperative languages such as Java, C++ etc. Thus different approaches pertain (no getters/setters, etc. etc.)
- A single well-constructed open-source Python X3D implementation might serve all Python distributions and programmers well.

# X3D Unified Object Model (X3DUOM) Creation



# Renewed Progress

- The perhaps-obvious insight “let Python be Python” has led to a native Python X3D implementation that requires no external package dependencies.
- Best practices are followed throughout, hopefully leading to a widely used package across multiple Python distributions for many users.
- Because the X3D Unified Object Model (X3DUOM) is used as the basis for source-code autogeneration, all interfaces in the x3d.py package are rigorous and complete for X3Dv4.
- This open-source work is published on Python Package Index (PyPi) for simple deployment support by Python programmers.

# Demonstrated Examples

- Conversion routines have produced over 3500 self-validating programs that demonstrate models in the X3D Example Archives.
- Round-trip conversions and testing are formally demonstrating correctness and completeness.
- Smoke tests confirm basic functionality during development.
- Unit testing on a systematic basis is especially important for Python since errors are only reported at run time.



# Python X3D Package x3d.py



The x3d.py Python X3D Package supports programmers with Python interfaces and objects for standards-based X3D programming, all as open source. The [Python X3D Implementation Progress](#) presentation provides an overview and shows examples.

[Installation](#) | [Design Features and Development](#) | [Examples](#) | [TODO](#) | [Contact](#)

## Installation



The [Python X3D project](#) is published on the Python Packaging Index (PyPi). It was configured by following the [PyPi packaging-projects tutorial](#).

Python x3d.py package installation and update options:

- `pip install x3d`
- `python -m pip install x3d`

## Design Features and Development



Package designers: Loren Peitso, John Carlson, Masaki Aono, Myeong Won Lee, and Don Brutzman.

[X3D Unified Object Model \(X3DUOM\)](#) is a full set of object-oriented interfaces for all nodes, fields and statements in the X3D Architecture Specification. The X3DUOM is used to autogenerate the entire x3d.py package, ensuring that it is always up-to-date with the latest X3D Architecture.

The current implementation strives to follow first principles of python package design, being as "pythonic" as possible.



[Help](#)[Donate](#)[Log in](#)[Register](#)

# x3d 0.0.7

[Latest version](#)

```
pip install x3d
```



Last released: half a minute ago

Package support for Extensible 3D (X3D) Graphics International Standard (IS)

## Navigation

[Project description](#)[Release history](#)[Download files](#)

## Project links

[Homepage](#)

## Project description

### Python X3D Package x3d.py

This project creates the [Python X3D Package](#) which is available for import via [PyPi](#).  
[Web3D Consortium](#) maintains this package under a BSD-style [open-source license](#).

### Python /x3d.py/ package installation and installation options

- `pip install x3d`
- `python -m pip install x3d`

# X3D Resources, Examples: Scene Archives for X3D

- <https://www.web3d.org/x3d/content/examples/X3dResources.html#Examples>
- A simple example test scene is [HelloWorld.x3d](#) provided in a variety of X3D encodings: [.x3d XML](#), [.x3dv ClassicVRML](#), [.wrl VRML97](#), [.html listing](#), [.html X-ITE](#), [.xhtml X3DOM](#), [.java Java source](#), [.json JavaScript Object Notation](#), [.py Python source \(experimental\)](#), [.x3db compression](#), [C14N canonicalization](#), and [.png image](#)
  - <https://www.web3d.org/x3d/stylesheets/python/HelloWorld.py>
  - <https://www.web3d.org/x3d/content/examples/HelloWorld.html>

```
#####

# comment preceding root node
newModel=x3d.X3D(profile='Immersive',version='3.3',
  head=x3d.head(
    children=[
      x3d.meta(content='HelloWorld.x3d',name='title'),
      x3d.meta(content='Special test case: simple X3D scene example: Hello World!',name='description'),
      x3d.meta(content='30 October 2000',name='created'),
      x3d.meta(content='16 April 2018',name='modified'),
      x3d.meta(content='Don Brutzman',name='creator'),
      x3d.meta(content='images/HelloWorld.tall.png',name='Image'),
      x3d.meta(content='http://en.wikipedia.org/wiki/Hello_world',name='reference'),
      x3d.meta(content='http://en.wikipedia.org/wiki/Hello#"Hello,_World"_computer_program',name='reference'),
      x3d.meta(content='http://en.wikibooks.org/w/index.php?title=Computer_Programming/Hello_world',name='reference'),
      x3d.meta(content='http://www.HelloWorldExample.net',name='reference'),
      x3d.meta(content='http://www.web3D.org',name='reference'),
      x3d.meta(content='http://www.web3d.org/realtime-3d/news/internationalization-x3d',name='reference'),
      x3d.meta(content='http://www.web3d.org/x3d/content/examples/HelloWorld.x3d',name='reference'),
      x3d.meta(content='http://X3dGraphics.com/examples/X3dForAdvancedModeling/HelloWorldScenes',name='reference'),
      x3d.meta(content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter01-TechnicalOverview
/HelloWorld.x3d',name='identifier'),
      x3d.meta(content='http://www.web3d.org/x3d/content/examples/license.html',name='license'),
      x3d.meta(content='X3D-Edit 3.3, https://savage.nps.edu/X3D-Edit',name='generator'),
      # Alternate encodings: VRML97, X3D ClassicVRML Encoding, X3D Compressed Binary Encoding (CBE), X3DOM, JSON
      x3d.meta(content='HelloWorld.wrl',name='reference'),
      x3d.meta(content='HelloWorld.x3dv',name='reference'),
      x3d.meta(content='HelloWorld.x3db',name='reference'),
      x3d.meta(content='HelloWorld.xhtml',name='reference'),
      x3d.meta(content='HelloWorld.json',name='reference')])),
Scene=x3d.Scene(
```

Python source example, part 1

```

Scene=x3d.Scene(
    # Example scene to illustrate X3D nodes and fields (XML elements and attributes)
    children=[
        x3d.Group(
            children=[
                x3d.Viewpoint(DEF='ViewUpClose',centerOfRotation=(0,-1,0),description='Hello world!',position=(0,-1,7)),
                x3d.Transform(DEF='ScaleFeetToMeters',rotation=(0,1,0,3),scale=(0.3048,0.3048,0.3048),
                    children=[
                        x3d.Shape(
                            geometry=x3d.Sphere(),
                            appearance=x3d.Appearance(
                                material=x3d.Material(DEF='MaterialLightBlue',diffuseColor=(0.1,0.5,1)),
                                texture=x3d.ImageTexture(DEF='ImageCloudlessEarth',url=["earth-topo.png","earth-topo.jpg","earth-topo-
small.gif","http://www.web3d.org/x3d/content/examples/Basic/earth-topo.png","http://www.web3d.org/x3d/content/examples/Basic/earth-
topo.jpg","http://www.web3d.org/x3d/content/examples/Basic/earth-topo-small.gif"])))],
                        x3d.Transform(translation=(0,-2,0),
                            children=[
                                x3d.Shape(
                                    geometry=x3d.Text(DEF='TextMessage',string=["Hello","world!"],
                                        fontStyle=x3d.FontStyle(justify=["MIDDLE","MIDDLE"])),
                                    appearance=x3d.Appearance(
                                        material=x3d.Material(USE='MaterialLightBlue')))]))]
            ) # X3D model complete

# comment following root node

#####
# Self-test diagnostics
#####

if          x3d.metaDiagnostics(newModel):
    print (x3d.metaDiagnostics(newModel))

print ("python load successful HelloWorld.py")

```

Python source example, part 2



```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.3//EN" "http://www.web3d.org/specifications/x3d-3.3.dtd">
<X3D profile='Immersive' version='3.3' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance' xsd:noNamespaceSchemaLocation='http://www.web3d.org/specifications/x3d-3.3.xsd'>
  <head>
    <meta name='title' content='HelloWorld.x3d' />
    <meta name='description' content='Simple X3D scene example: Hello World!' />
    <meta name='created' content='30 October 2000' />
    <meta name='modified' content='23 September 2017' />
    <meta name='creator' content='Don Brutzman' />
    <meta name='Image' content='HelloWorld.tall.png' />
    <meta name='reference' content='http://en.wikipedia.org/wiki/Hello_world' />
    <meta name='reference' content='https://en.wikipedia.org/wiki/Hello#.22Hello.2C_World.22_computer_program' />
    <meta name='reference' content='https://en.wikipedia.org/wiki/Hello,_World!_program' />
    <meta name='reference' content='http://en.wikibooks.org/w/index.php?title=Computer_Programming/Hello_world' />
    <meta name='reference' content='http://www.HelloWorldExample.net' />
    <meta name='reference' content='http://www.web3D.org' />
    <meta name='reference' content='http://www.web3d.org/realtime-3d/news/internationalization-x3d' />
    <meta name='reference' content='http://www.web3d.org/x3d/content/examples/HelloWorld.x3d' />
    <meta name='reference' content='http://X3dGraphics.com/examples/X3dForAdvancedModeling/HelloWorldScenes' />
    <meta name='identifier' content='http://X3dGraphics.com/examples/X3dForWebAuthors/Chapter01TechnicalOverview/HelloWorld.x3d' />
    <meta name='license' content='http://www.web3d.org/x3d/content/examples/license.html' />
    <meta name='generator' content='X3D-Edit 3.3, https://savage.nps.edu/X3D-Edit' />
    <!-- Alternate encodings: VRML97, X3D ClassicVRML Encoding, X3D Compressed Binary Encoding (CBE), X3DOM, JSON -->
    <meta name='reference' content='HelloWorld.wrl' />
    <meta name='reference' content='HelloWorld.x3dv' />
    <meta name='reference' content='HelloWorld.x3db' />
    <meta name='reference' content='HelloWorld.xhtml' />
    <meta name='reference' content='HelloWorld.json' />
  </head>

```

X3D XML source example, part 1

```

<Scene>
  <!-- Example scene to illustrate X3D nodes and fields (XML elements and attributes) -->
  <WorldInfo title='Hello World!'/>
  <Group>
    <Viewpoint DEF='ViewUpClose' centerOfRotation='0 -1 0' description='Hello world!' position='0 -1 7'/>
    <Transform rotation='0 1 0 3'>
      <Shape>
        <Sphere/>
        <Appearance>
          <Material DEF='MaterialLightBlue' diffuseColor='0.1 0.5 1'/>
          <ImageTexture DEF='ImageCloudlessEarth'
            url=' "earth-topo.png" "earth-topo.jpg" "earth-topo-small.gif" "http://www.web3d.org/x3d/content/examples/Basic/earth-topo.png"
              "http://www.web3d.org/x3d/content/examples/Basic/earth-topo.jpg" "http://www.web3d.org/x3d/content/examples/Basic/earth-topo-small.gif" '/>
        </Appearance>
      </Shape>
    </Transform>
    <Transform translation='0 -2 0'>
      <Shape>
        <Text DEF='TextMessage' string='"Hello" "world!'">
          <FontStyle justify="MIDDLE" "MIDDLE"/>
        </Text>
        <Appearance>
          <Material USE='MaterialLightBlue'/>
        </Appearance>
      </Shape>
    </Transform>
  </Group>
</Scene>
</X3D>

```

X3D XML source example, part 2

# Next Steps

- [Python X3D Package x3d.py](#) is functional and undergoing active development.
- All work has been discussed and performed publicly on [x3d-public@web3d.org](mailto:x3d-public@web3d.org)
- Can our ISO colleagues who also work on a Python encoding comment on this work? Improvements in design patterns are easy to integrate to produce “best of breed” results.
- *Good discussions on merging designs are in progress!* Myeong Won Lee and Masaki Aono have draft NWIP, independent implementation and viewer.
- Once Python X3D is fully complete and stable, further deployment is expected for Jupyter Notebook, Apache Zeppelin, and Anaconda distributions of Python.
- Because of this thorough and rigorous approach to producing a native Python X3D implementation, it is likely that the general Web3D requirement for multiple independent implementations will be relaxed in this case.
- Availability of both Python x3d.py package and Java X3DJSAIL library are each expected to “open the door” for use of both X3D and HAnim models as part of Big Data workflows using Apache Hadoop, Apache Spark, and other Data Science approaches.

# Contact

**Don Brutzman, Ph.D.**

[brutzman@nps.edu](mailto:brutzman@nps.edu)

<http://faculty.nps.edu/brutzman>

Code USW/Br, Naval Postgraduate School

Monterey California 93943-5000 USA

1.831.656.2149 work

1.831.402.4809 cell