
pyIPXACT

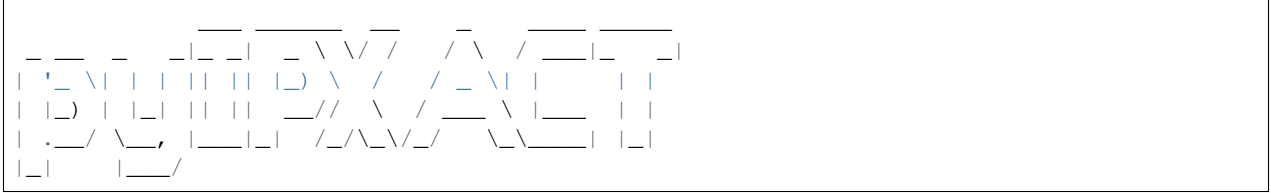
Release v0.1

Patrick Lehmann

Jan 02, 2020

OVERVIEW

1	Implemented Features	3
2	Examples	5
3	IP-XACT Resources	7
4	Contributors	9
5	License	11
5.1	Installation/Updates	11
5.2	Dependencies	11
5.3	pyIPXACT	11
5.4	pyIPXACT.Catalog	12
5.5	pyIPXACT.Component	13
5.6	pyIPXACT.Design	16
5.7	pyIPXACT.DesignConfiguration	17
5.8	pyIPXACT.GeneratorChain	18
5.9	Apache License 2.0	19
5.10	Index	22
5.11	Python Module Index	22
	Python Module Index	23
	Index	25



pyIPXACT - An IP-XACT DOM (Document Object Model) for [IEEE 1685-2014](#) in Python.

IMPLEMENTED FEATURES

- Generate IP-XACT files for
 - IP-XACT catalogs

EXAMPLES

IP-XACT RESOURCES

- Standards:
 - [IEEE 1685-2009](#)
 - [IEEE 1685-2014](#)
- Schema files:
 - [IPXACT-Schema at GitHub](#)
 - [IP-XACT at Accellera](#)

CONTRIBUTORS

- [Patrick Lehmann](#) (Maintainer)

This library is licensed under **Apache License 2.0**.

5.1 Installation/Updates

5.1.1 Using PIP

Installation using PIP

```
pip3 install pyIPXACT
```

Updating using PIP

```
pip3 install -U pyIPXACT
```

5.2 Dependencies

- lxml

5.3 pyIPXACT

```
class pyIPXACT.IpxactSchemaStruct(version: str, namespacePrefix: str, schemaUri: str,  
schemaUrl: str, localPath: pathlib.Path)
```

Bases: `object`

Schema descriptor made of version, namespace prefix, URI, URL and local path.

```
__init__(version: str, namespacePrefix: str, schemaUri: str, schemaUrl: str, localPath: pathlib.Path)  
Constructor
```

```
Version = None  
Schema version
```

```
NamespacePrefix = None  
XML namespace prefix
```

SchemaUri = None
schema URI

SchemaUrl = None
schema URL

LocalPath = None
local path

class `pyIPXACT.Vlnv` (*vendor, library, name, version*)

Bases: `object`

VLNV data structure (Vendor, Library, Name, Version) as a unique identifier in IP-XACT.

__init__ (*vendor, library, name, version*)
Constructor

Vendor = None
Vendor name in a VLNv unique identifier

Library = None
Library name in a VLNv unique identifier

Name = None
Component name in a VLNv unique identifier

Version = None
Version in a VLNv unique identifier

ToXml (*indent=1, isVersionedIdentifier=False*)
Converts the object's data into XML format.

class `pyIPXACT.RootElement` (*vlnv*)

Bases: `object`

Base-class for all IP-XACT data classes.

__init__ (*vlnv*)
Base-constructor to set a VLNv field for all derives classes.

_vlnv = None
VLNV unique identifier.

exception `pyIPXACT.PyIpxactException`

Bases: `Exception`

Base-exception for all exceptions in this package.

5.4 pyIPXACT.Catalog

class `pyIPXACT.Catalog.IpxactFile` (*vlnv: pyIPXACT.Vlnv, name: str, description: str*)

Bases: `object`

Represents a IP-XACT file.

__init__ (*vlnv: pyIPXACT.Vlnv, name: str, description: str*)
Constructor

_vlnv = None
VLNV unique identifier


```

_name = None
    Name

_description = None
    Description

classmethod FromXml (element)
    Constructs an instance of IpxactFile from an lxml element.

ToXml (indent)
    Converts the object's data into XML format.

```

```

class pyIPXACT.Catalog.Catalog (vlnv: pyIPXACT.Vlnv, description: str)
    Bases: pyIPXACT.RootElement

```

Represents an IP-XACT catalog.

```

__init__ (vlnv: pyIPXACT.Vlnv, description: str)
    Base-constructor to set a VLNV field for all derives classes.

```

```

classmethod FromFile (filePath: pathlib.Path)
    Constructs an instance of Catalog from a file.

```

```

ToXml ()
    Converts the object's data into XML format.

```

5.5 pyIPXACT.Component

```

class pyIPXACT.Component.Component (vlnv, description)
    Bases: pyIPXACT.RootElement

```

Represents an IP-XACT components.

```

__init__ (vlnv, description)
    Base-constructor to set a VLNV field for all derives classes.

```

```

ToXml ()
    Converts the object's data into XML format.

```

```

class pyIPXACT.Component.BusInterface
    Bases: object

```

Represents an IP-XACT bus interface.

```

__init__ ()
    Initialize self. See help(type(self)) for accurate signature.

```

```

ToXml (indent=0)
    Converts the object's data into XML format.

```

```

class pyIPXACT.Component.IndirectInterface
    Bases: object

```

Represents an IP-XACT indirect interface.

```

__init__ ()
    Initialize self. See help(type(self)) for accurate signature.

```

```

ToXml (indent=0)
    Converts the object's data into XML format.

```

class pyIPXACT.Component.**Channel**
Bases: `object`

Represents an IP-XACT channel.

`__init__()`
Initialize self. See `help(type(self))` for accurate signature.

`ToXml(indent=0)`
Converts the object's data into XML format.

class pyIPXACT.Component.**RemapState**
Bases: `object`

Represents an IP-XACT remap state.

`__init__()`
Initialize self. See `help(type(self))` for accurate signature.

`ToXml(indent=0)`
Converts the object's data into XML format.

class pyIPXACT.Component.**AddressSpace**
Bases: `object`

Represents an IP-XACT address space.

`__init__()`
Initialize self. See `help(type(self))` for accurate signature.

`ToXml(indent=0)`
Converts the object's data into XML format.

class pyIPXACT.Component.**MemoryMap**
Bases: `object`

Represents an IP-XACT memory map.

`__init__()`
Initialize self. See `help(type(self))` for accurate signature.

`ToXml(indent=0)`
Converts the object's data into XML format.

class pyIPXACT.Component.**Model**
Bases: `object`

Represents an IP-XACT model.

`__init__()`
Initialize self. See `help(type(self))` for accurate signature.

`ToXml(indent=0)`
Converts the object's data into XML format.

class pyIPXACT.Component.**ComponentGenerator**
Bases: `object`

Represents an IP-XACT component generator.

`__init__()`
Initialize self. See `help(type(self))` for accurate signature.

`ToXml(indent=0)`
Converts the object's data into XML format.

```

class pyIPXACT.Component.Choice
    Bases: object

    Represents an IP-XACT choice.

    __init__()
        Initialize self. See help(type(self)) for accurate signature.

    ToXml (indent=0)
        Converts the object's data into XML format.

class pyIPXACT.Component.FileSet
    Bases: object

    Represents an IP-XACT fileset.

    __init__()
        Initialize self. See help(type(self)) for accurate signature.

    ToXml (indent=0)
        Converts the object's data into XML format.

class pyIPXACT.Component.WhiteboxElement
    Bases: object

    Represents an IP-XACT whitebos element.

    __init__()
        Initialize self. See help(type(self)) for accurate signature.

    ToXml (indent=0)
        Converts the object's data into XML format.

class pyIPXACT.Component.Cpu
    Bases: object

    Represents an IP-XACT cpu.

    __init__()
        Initialize self. See help(type(self)) for accurate signature.

    ToXml (indent=0)
        Converts the object's data into XML format.

class pyIPXACT.Component.OtherClockDriver
    Bases: object

    Represents an IP-XACT other clock driver.

    __init__()
        Initialize self. See help(type(self)) for accurate signature.

    ToXml (indent=0)
        Converts the object's data into XML format.

class pyIPXACT.Component.ResetType
    Bases: object

    Represents an IP-XACT reset type.

    __init__()
        Initialize self. See help(type(self)) for accurate signature.

    ToXml (indent=0)
        Converts the object's data into XML format.

```

class pyIPXACT.Component.**Parameter**

Bases: `object`

Represents an IP-XACT parameter.

`__init__()`

Initialize self. See `help(type(self))` for accurate signature.

`ToXml(indent=0)`

Converts the object's data into XML format.

class pyIPXACT.Component.**Assertion**

Bases: `object`

Represents an IP-XACT assertion.

`__init__()`

Initialize self. See `help(type(self))` for accurate signature.

`ToXml(indent=0)`

Converts the object's data into XML format.

5.6 pyIPXACT.Design

class pyIPXACT.Design.**Design** (*vlmv: pyIPXACT.Vlmv, description: str*)

Bases: `pyIPXACT.RootElement`

Represents an IP-XACT design.

`__init__(vlmv: pyIPXACT.Vlmv, description: str)`

Base-constructor to set a VLNV field for all derives classes.

`ToXml()`

Converts the object's data into XML format.

class pyIPXACT.Design.**ComponentInstance**

Bases: `object`

Represents an IP-XACT component instance.

`__init__()`

Initialize self. See `help(type(self))` for accurate signature.

`ToXml(indent=0)`

Converts the object's data into XML format.

class pyIPXACT.Design.**Interconnection**

Bases: `object`

Represents an IP-XACT interconnection.

`__init__()`

Initialize self. See `help(type(self))` for accurate signature.

`ToXml(indent=0)`

Converts the object's data into XML format.

class pyIPXACT.Design.**AdHocConnection**

Bases: `object`

Represents an IP-XACT ad-hoc connection.

`__init__()`
Initialize self. See help(type(self)) for accurate signature.

`ToXml(indent=0)`
Converts the object's data into XML format.

5.7 pyIPXACT.DesignConfiguration

class pyIPXACT.DesignConfiguration.**DesignConfiguration** (*vlmv: pyIPXACT.Vlmv, description: str*)

Bases: *pyIPXACT.RootElement*

Represents an IP-XACT design configuration.

`__init__ (vlmv: pyIPXACT.Vlmv, description: str)`
Base-constructor to set a VLNV field for all derives classes.

`ToXml ()`
Converts the object's data into XML format.

class pyIPXACT.DesignConfiguration.**GeneratorChainConfiguration**

Bases: *object*

Represents an IP-XACT generator chain configuration.

`__init__ ()`
Initialize self. See help(type(self)) for accurate signature.

`ToXml(indent=0)`
Converts the object's data into XML format.

class pyIPXACT.DesignConfiguration.**InterconnectionConfiguration**

Bases: *object*

Represents an IP-XACT interconnection configuration.

`__init__ ()`
Initialize self. See help(type(self)) for accurate signature.

`ToXml(indent=0)`
Converts the object's data into XML format.

class pyIPXACT.DesignConfiguration.**ViewConfiguration**

Bases: *object*

Represents an IP-XACT view configuration.

`__init__ ()`
Initialize self. See help(type(self)) for accurate signature.

`ToXml(indent=0)`
Converts the object's data into XML format.

5.8 pyIPXACT.GeneratorChain

class pyIPXACT.GeneratorChain.**GeneratorChain** (*vlnv: pyIPXACT.Vlnv, displayName: str, description: str, chainGroup*)

Bases: *pyIPXACT.RootElement*

Represents an IP-XACT generator chain.

__init__ (*vlnv: pyIPXACT.Vlnv, displayName: str, description: str, chainGroup*)
Constructor

ToXml ()
Converts the object's data into XML format.

class pyIPXACT.GeneratorChain.**GeneratorChainSelector**

Bases: *object*

Represents an IP-XACT generator chain selector.

__init__ ()
Initialize self. See help(type(self)) for accurate signature.

ToXml (*indent=0*)
Converts the object's data into XML format.

class pyIPXACT.GeneratorChain.**ComponentGeneratorSelector**

Bases: *object*

Represents an IP-XACT component generator selector.

__init__ ()
Initialize self. See help(type(self)) for accurate signature.

ToXml (*indent=0*)
Converts the object's data into XML format.

class pyIPXACT.GeneratorChain.**Generator**

Bases: *object*

Represents an IP-XACT generator.

__init__ ()
Initialize self. See help(type(self)) for accurate signature.

ToXml (*indent=0*)
Converts the object's data into XML format.

Note: This is a local copy of the [Apache License Version 2.0](#).

5.9 Apache License 2.0

Version 2.0, January 2004

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

5.9.1 1. Definitions.

“**License**” shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

“**Licensor**” shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

“**Legal Entity**” shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, “**control**” means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

“**You**” (or “**Your**”) shall mean an individual or Legal Entity exercising permissions granted by this License.

“**Source**” form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

“**Object**” form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

“**Work**” shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

“**Derivative Works**” shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

“**Contribution**” shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, “**submitted**” means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as “**Not a Contribution.**”

“**Contributor**” shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

5.9.2 2. Grant of Copyright License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

5.9.3 3. Grant of Patent License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

5.9.4 4. Redistribution.

You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- You must give any other recipients of the Work or Derivative Works a copy of this License; and
- You must cause any modified files to carry prominent notices stating that You changed the files; and
- You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- If the Work includes a “NOTICE” text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5.9.5 5. Submission of Contributions.

Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

5.9.6 6. Trademarks.

This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

5.9.7 7. Disclaimer of Warranty.

Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

5.9.8 8. Limitation of Liability.

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

5.9.9 9. Accepting Warranty or Additional Liability.

While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

Appendix: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets “[]” replaced with your own identifying information. (Don’t include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same “printed page” as the copyright notice for easier identification within third-party archives.

```
Copyright [yyyy] [name of copyright owner]
```

```
Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at
```

```
http://www.apache.org/licenses/LICENSE-2.0
```

```
Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.
```

5.10 Index

5.11 Python Module Index

PYTHON MODULE INDEX

p

pyIPXACT, [11](#)
pyIPXACT.Catalog, [12](#)
pyIPXACT.Component, [13](#)
pyIPXACT.Design, [16](#)
pyIPXACT.DesignConfiguration, [17](#)
pyIPXACT.GeneratorChain, [18](#)

Symbols

- `__init__()` (*pyIPXACT.Catalog.Catalog* method), 13
- `__init__()` (*pyIPXACT.Catalog.IpxactFile* method), 12
- `__init__()` (*pyIPXACT.Component.AddressSpace* method), 14
- `__init__()` (*pyIPXACT.Component.Assertion* method), 16
- `__init__()` (*pyIPXACT.Component.BusInterface* method), 13
- `__init__()` (*pyIPXACT.Component.Channel* method), 14
- `__init__()` (*pyIPXACT.Component.Choice* method), 15
- `__init__()` (*pyIPXACT.Component.Component* method), 13
- `__init__()` (*pyIPXACT.Component.ComponentGenerator* method), 14
- `__init__()` (*pyIPXACT.Component.Cpu* method), 15
- `__init__()` (*pyIPXACT.Component.FileSet* method), 15
- `__init__()` (*pyIPXACT.Component.IndirectInterface* method), 13
- `__init__()` (*pyIPXACT.Component.MemoryMap* method), 14
- `__init__()` (*pyIPXACT.Component.Model* method), 14
- `__init__()` (*pyIPXACT.Component.OtherClockDriver* method), 15
- `__init__()` (*pyIPXACT.Component.Parameter* method), 16
- `__init__()` (*pyIPXACT.Component.RemapState* method), 14
- `__init__()` (*pyIPXACT.Component.ResetType* method), 15
- `__init__()` (*pyIPXACT.Component.WhiteboxElement* method), 15
- `__init__()` (*pyIPXACT.Design.AdHocConnection* method), 16
- `__init__()` (*pyIPXACT.Design.ComponentInstance* method), 16
- `__init__()` (*pyIPXACT.Design.Design* method), 16
- `__init__()` (*pyIPXACT.Design.Interconnection* method), 16
- `__init__()` (*pyIPXACT.DesignConfiguration.DesignConfiguration* method), 17
- `__init__()` (*pyIPXACT.DesignConfiguration.GeneratorChainConfiguration* method), 17
- `__init__()` (*pyIPXACT.DesignConfiguration.InterconnectionConfiguration* method), 17
- `__init__()` (*pyIPXACT.DesignConfiguration.ViewConfiguration* method), 17
- `__init__()` (*pyIPXACT.GeneratorChain.ComponentGeneratorSelector* method), 18
- `__init__()` (*pyIPXACT.GeneratorChain.Generator* method), 18
- `__init__()` (*pyIPXACT.GeneratorChain.GeneratorChain* method), 18
- `__init__()` (*pyIPXACT.GeneratorChain.GeneratorChainSelector* method), 18
- `__init__()` (*pyIPXACT.IpxactSchemaStruct* method), 11
- `__init__()` (*pyIPXACT.RootElement* method), 12
- `__init__()` (*pyIPXACT.Vlnv* method), 12
- `_description` (*pyIPXACT.Catalog.IpxactFile* attribute), 13
- `_name` (*pyIPXACT.Catalog.IpxactFile* attribute), 12
- `_vlnv` (*pyIPXACT.Catalog.IpxactFile* attribute), 12
- `_vlnv` (*pyIPXACT.RootElement* attribute), 12

A

- AddressSpace (*class in pyIPXACT.Component*), 14
- AdHocConnection (*class in pyIPXACT.Design*), 16
- Assertion (*class in pyIPXACT.Component*), 16

B

BusInterface (class in pyIPXACT.Component), 13

C

Catalog (class in pyIPXACT.Catalog), 13

Channel (class in pyIPXACT.Component), 13

Choice (class in pyIPXACT.Component), 14

Component (class in pyIPXACT.Component), 13

ComponentGenerator (class in pyIPXACT.Component), 14

ComponentGeneratorSelector (class in pyIPXACT.GeneratorChain), 18

ComponentInstance (class in pyIPXACT.Design), 16

Cpu (class in pyIPXACT.Component), 15

D

Design (class in pyIPXACT.Design), 16

DesignConfiguration (class in pyIPXACT.DesignConfiguration), 17

F

FileSet (class in pyIPXACT.Component), 15

FromFile() (pyIPXACT.Catalog.Catalog class method), 13

FromXml() (pyIPXACT.Catalog.IpxactFile class method), 13

G

Generator (class in pyIPXACT.GeneratorChain), 18

GeneratorChain (class in pyIPXACT.GeneratorChain), 18

GeneratorChainConfiguration (class in pyIPXACT.DesignConfiguration), 17

GeneratorChainSelector (class in pyIPXACT.GeneratorChain), 18

I

IndirectInterface (class in pyIPXACT.Component), 13

Interconnection (class in pyIPXACT.Design), 16

InterconnectionConfiguration (class in pyIPXACT.DesignConfiguration), 17

IpxactFile (class in pyIPXACT.Catalog), 12

IpxactSchemaStruct (class in pyIPXACT), 11

L

Library (pyIPXACT.Vlnv attribute), 12

LocalPath (pyIPXACT.IpxactSchemaStruct attribute), 12

M

MemoryMap (class in pyIPXACT.Component), 14

Model (class in pyIPXACT.Component), 14

N

Name (pyIPXACT.Vlnv attribute), 12

NamespacePrefix (pyIPXACT.IpxactSchemaStruct attribute), 11

O

OtherClockDriver (class in pyIPXACT.Component), 15

P

Parameter (class in pyIPXACT.Component), 15

pyIPXACT (module), 11

pyIPXACT.Catalog (module), 12

pyIPXACT.Component (module), 13

pyIPXACT.Design (module), 16

pyIPXACT.DesignConfiguration (module), 17

pyIPXACT.GeneratorChain (module), 18

PyIpxactException, 12

R

RemapState (class in pyIPXACT.Component), 14

ResetType (class in pyIPXACT.Component), 15

RootElement (class in pyIPXACT), 12

S

SchemaUri (pyIPXACT.IpxactSchemaStruct attribute), 11

SchemaUrl (pyIPXACT.IpxactSchemaStruct attribute), 12

T

ToXml() (pyIPXACT.Catalog.Catalog method), 13

ToXml() (pyIPXACT.Catalog.IpxactFile method), 13

ToXml() (pyIPXACT.Component.AddressSpace method), 14

ToXml() (pyIPXACT.Component.Assertion method), 16

ToXml() (pyIPXACT.Component.BusInterface method), 13

ToXml() (pyIPXACT.Component.Channel method), 14

ToXml() (pyIPXACT.Component.Choice method), 15

ToXml() (pyIPXACT.Component.Component method), 13

ToXml() (pyIPXACT.Component.ComponentGenerator method), 14

ToXml() (pyIPXACT.Component.Cpu method), 15

ToXml() (pyIPXACT.Component.FileSet method), 15

ToXml() (pyIPXACT.Component.IndirectInterface method), 13

ToXml() (pyIPXACT.Component.MemoryMap method), 14

ToXml() (pyIPXACT.Component.Model method), 14

ToXml() (pyIPXACT.Component.OtherClockDriver method), 15

ToXml () (*pyIPXACT.Component.Parameter method*),
 16
 ToXml () (*pyIPXACT.Component.RemapState method*),
 14
 ToXml () (*pyIPXACT.Component.ResetType method*),
 15
 ToXml () (*pyIPXACT.Component.WhiteboxElement
 method*), 15
 ToXml () (*pyIPXACT.Design.AdHocConnection
 method*), 17
 ToXml () (*pyIPXACT.Design.ComponentInstance
 method*), 16
 ToXml () (*pyIPXACT.Design.Design method*), 16
 ToXml () (*pyIPXACT.Design.Interconnection method*),
 16
 ToXml () (*pyIPXACT.DesignConfiguration.DesignConfiguration
 method*), 17
 ToXml () (*pyIPXACT.DesignConfiguration.GeneratorChainConfiguration
 method*), 17
 ToXml () (*pyIPXACT.DesignConfiguration.InterconnectionConfiguration
 method*), 17
 ToXml () (*pyIPXACT.DesignConfiguration.ViewConfiguration
 method*), 17
 ToXml () (*pyIPXACT.GeneratorChain.ComponentGeneratorSelector
 method*), 18
 ToXml () (*pyIPXACT.GeneratorChain.Generator
 method*), 18
 ToXml () (*pyIPXACT.GeneratorChain.GeneratorChain
 method*), 18
 ToXml () (*pyIPXACT.GeneratorChain.GeneratorChainSelector
 method*), 18
 ToXml () (*pyIPXACT.Vlnv method*), 12

V

Vendor (*pyIPXACT.Vlnv attribute*), 12
 Version (*pyIPXACT.IpxactSchemaStruct attribute*), 11
 Version (*pyIPXACT.Vlnv attribute*), 12
 ViewConfiguration (*class in pyIPX-
 ACT.DesignConfiguration*), 17
 Vlnv (*class in pyIPXACT*), 12

W

WhiteboxElement (*class in pyIPXACT.Component*),
 15