

# AIX Frontend Compiler Reference Manual

ver. 1.0

Written by Youngsun Han (youngsun@pknu.ac.kr)

High Performance Computing Lab.  
Department of Computer Engineering  
Pukyong National University



# Contents

<b>1</b>	<b>SKT AIX Frontend Compiler</b>	<b>1</b>
<b>2</b>	<b>aix_tf</b>	<b>3</b>
<b>3</b>	<b>Namespace Index</b>	<b>5</b>
3.1	Packages . . . . .	5
<b>4</b>	<b>Hierarchical Index</b>	<b>7</b>
4.1	Class Hierarchy . . . . .	7
<b>5</b>	<b>Class Index</b>	<b>9</b>
5.1	Class List . . . . .	9
<b>6</b>	<b>File Index</b>	<b>11</b>
6.1	File List . . . . .	11
<b>7</b>	<b>Namespace Documentation</b>	<b>13</b>
7.1	aixh_pb2 Namespace Reference . . . . .	13
7.1.1	Variable Documentation . . . . .	13
7.1.1.1	_AIXGRAPH . . . . .	13
7.1.1.2	_AIXLAYER . . . . .	14
7.1.1.3	_AIXLAYER_AIXACTIVATIONMODE . . . . .	14
7.1.1.4	_AIXLAYER_AIXCONVOLUTIONDESC . . . . .	14
7.1.1.5	_AIXLAYER_AIXDATATYPE . . . . .	14
7.1.1.6	_AIXLAYER_AIXEWADDDDESC . . . . .	14
7.1.1.7	_AIXLAYER_AIXLAYERTYPE . . . . .	15

7.1.1.8	<a href="#">_AIXLAYER_AIXSAMPLINGDESC</a>	15
7.1.1.9	<a href="#">_AIXLAYER_AIXSAMPLINGMODE</a>	15
7.1.1.10	<a href="#">_AIXLAYER_AIXTENSOR</a>	15
7.1.1.11	<a href="#">_AIXLAYER_AIXTENSORFORMAT</a>	15
7.1.1.12	<a href="#">_b</a>	16
7.1.1.13	<a href="#">_options</a>	16
7.1.1.14	<a href="#">_sym_db</a>	16
7.1.1.15	<a href="#">AIXGraph</a>	16
7.1.1.16	<a href="#">AIXLayer</a>	16
7.1.1.17	<a href="#">containing_type</a>	17
7.1.1.18	<a href="#">DESCRIPTOR</a>	17
7.1.1.19	<a href="#">enum_type</a>	17
7.1.1.20	<a href="#">has_options</a>	17
7.1.1.21	<a href="#">message_type</a>	17
7.2	<a href="#">AxfcError Namespace Reference</a>	17
7.3	<a href="#">AxfcFrontendCompiler Namespace Reference</a>	18
7.4	<a href="#">AxfcGraphWriter Namespace Reference</a>	18
7.5	<a href="#">AxfcIRBlock Namespace Reference</a>	18
7.6	<a href="#">AxfcIRBuilder Namespace Reference</a>	18
7.7	<a href="#">AxfcIRGraph Namespace Reference</a>	18
7.8	<a href="#">AxfcIRNode Namespace Reference</a>	18
7.9	<a href="#">AxfcIRTranslator Namespace Reference</a>	19
7.10	<a href="#">AxfcLauncherWriter Namespace Reference</a>	19
7.11	<a href="#">AxfcMachineDesc Namespace Reference</a>	19
7.12	<a href="#">AxfcMain Namespace Reference</a>	19
7.12.1	<a href="#">Function Documentation</a>	19
7.12.1.1	<a href="#">__main()</a>	20
7.12.2	<a href="#">Variable Documentation</a>	20
7.12.2.1	<a href="#">args</a>	20
7.12.2.2	<a href="#">help</a>	20
7.12.2.3	<a href="#">metavar</a>	20
7.12.2.4	<a href="#">parser</a>	20
7.12.2.5	<a href="#">required</a>	21
7.12.2.6	<a href="#">str</a>	21
7.12.2.7	<a href="#">type</a>	21
7.13	<a href="#">AxfcTFIRBuilder Namespace Reference</a>	21
7.14	<a href="#">AxfcTFIRTranslator Namespace Reference</a>	21
7.14.1	<a href="#">Variable Documentation</a>	22
7.14.1.1	<a href="#">aix_data_type_tbl</a>	22
7.14.1.2	<a href="#">aix_tensor_format_tbl</a>	22

<b>8 Class Documentation</b>	<b>23</b>
8.1 AxfcMachineDesc.AxfcMachineDesc.AIXLayerInfo Class Reference	23
8.1.1 Detailed Description	23
8.1.2 Constructor & Destructor Documentation	24
8.1.2.1 __init__()	24
8.1.3 Member Function Documentation	24
8.1.3.1 __str__()	24
8.1.4 Member Data Documentation	24
8.1.4.1 activation	24
8.1.4.2 is_conv	24
8.1.4.3 is_group	25
8.1.4.4 layer	25
8.1.4.5 op	25
8.1.4.6 profit	25
8.2 AxfcIRTranslator.AIXTensorType Class Reference	26
8.2.1 Detailed Description	26
8.2.2 Member Data Documentation	26
8.2.2.1 AIX_TENSOR_BIAS	26
8.2.2.2 AIX_TENSOR_FILTER	27
8.2.2.3 AIX_TENSOR_INPUT	27
8.2.2.4 AIX_TENSOR_MEAN	27
8.2.2.5 AIX_TENSOR_OUTPUT	27
8.2.2.6 AIX_TENSOR_SCALE	27
8.2.2.7 AIX_TENSOR_UNKNOWN	27
8.2.2.8 AIX_TENSOR_VARIANCE	28
8.3 AxfcError.AxfcError Class Reference	28
8.3.1 Detailed Description	29
8.3.2 Member Data Documentation	29
8.3.2.1 DUMP_IR_GRAPH_ERROR	29
8.3.2.2 EMPTY_IR_BLOCK	29

8.3.2.3	INVALID_ACTIVATION_LAYER . . . . .	29
8.3.2.4	INVALID_AIX_GRAPH . . . . .	30
8.3.2.5	INVALID_AIX_LAYER_TYPE . . . . .	30
8.3.2.6	INVALID_AIX_TENSOR_FORMAT . . . . .	30
8.3.2.7	INVALID_AIX_TENSOR_INPUT . . . . .	30
8.3.2.8	INVALID_BATCHNORM_LAYER . . . . .	30
8.3.2.9	INVALID_CONVOLUTION_LAYER . . . . .	30
8.3.2.10	INVALID_EWADD_LAYER . . . . .	31
8.3.2.11	INVALID_FILE_PATH . . . . .	31
8.3.2.12	INVALID_GROUP_CONV_LAYER . . . . .	31
8.3.2.13	INVALID_IDENTITY_LAYER . . . . .	31
8.3.2.14	INVALID_INPUT_TYPE . . . . .	31
8.3.2.15	INVALID_IR_GRAPH . . . . .	31
8.3.2.16	INVALID_MAXPOOL_LAYER . . . . .	32
8.3.2.17	INVALID_MD_FORMAT . . . . .	32
8.3.2.18	INVALID_PAD_LAYER . . . . .	32
8.3.2.19	INVALID_PARAMETER . . . . .	32
8.3.2.20	INVALID_TF_GRAPH . . . . .	32
8.3.2.21	INVALID_BIASADD_LAYER . . . . .	32
8.3.2.22	NOT_AIXH_SUPPORT . . . . .	33
8.3.2.23	NOT_IMPLEMENTED . . . . .	33
8.3.2.24	PRED_NODE_NOT_FOUND . . . . .	33
8.3.2.25	SUCCESS . . . . .	33
8.3.2.26	UNKNOWN_TENSOR_TYPE . . . . .	33
8.3.2.27	UNREMOVED_IDENTITY . . . . .	33
8.3.2.28	UNSUPPORTED_AIX_LAYER_EMIT . . . . .	34
8.4	AxhcFrontendCompiler.AxhcFrontendCompiler Class Reference . . . . .	34
8.4.1	Detailed Description . . . . .	34
8.4.2	Constructor & Destructor Documentation . . . . .	35
8.4.2.1	__init__() . . . . .	35

8.4.3	Member Function Documentation . . . . .	35
8.4.3.1	__str__() . . . . .	35
8.4.3.2	compile() . . . . .	35
8.4.3.3	dump_aix_graphs() . . . . .	36
8.4.3.4	dump_launcher() . . . . .	36
8.4.3.5	get_ir_graph() . . . . .	36
8.4.3.6	read_md_file() . . . . .	37
8.4.4	Member Data Documentation . . . . .	37
8.4.4.1	__ir_builder . . . . .	37
8.4.4.2	__ir_translator . . . . .	37
8.4.4.3	__md . . . . .	37
8.5	AxgcGraphWriter.AxgcGraphWriter Class Reference . . . . .	38
8.5.1	Detailed Description . . . . .	38
8.5.2	Constructor & Destructor Documentation . . . . .	38
8.5.2.1	__init__() . . . . .	38
8.5.3	Member Function Documentation . . . . .	39
8.5.3.1	add_edge() . . . . .	39
8.5.3.2	add_node() . . . . .	39
8.5.3.3	write_file() . . . . .	39
8.5.4	Member Data Documentation . . . . .	40
8.5.4.1	__edge_id . . . . .	40
8.5.4.2	__graph . . . . .	40
8.5.4.3	__nodes . . . . .	40
8.5.4.4	__x_axis . . . . .	41
8.5.4.5	__y_axis . . . . .	41
8.6	AxgcIRBlock.AxgcIRBlock Class Reference . . . . .	41
8.6.1	Detailed Description . . . . .	42
8.6.2	Constructor & Destructor Documentation . . . . .	42
8.6.2.1	__init__() . . . . .	42
8.6.3	Member Function Documentation . . . . .	42

8.6.3.1	<code>__analyse_inout()</code>	42
8.6.3.2	<code>__str__()</code>	43
8.6.3.3	<code>analyse_liveness()</code>	43
8.6.3.4	<code>analyze_profit()</code>	43
8.6.4	Member Data Documentation	44
8.6.4.1	<code>aix_graph</code>	44
8.6.4.2	<code>aixh_profit</code>	44
8.6.4.3	<code>id</code>	44
8.6.4.4	<code>is_aixh_support</code>	44
8.6.4.5	<code>live_in</code>	44
8.6.4.6	<code>live_out</code>	45
8.6.4.7	<code>nodes</code>	45
8.7	<code>AxfclRBuilder.AxfclRBuilder</code> Class Reference	45
8.7.1	Detailed Description	46
8.7.2	Constructor & Destructor Documentation	46
8.7.2.1	<code>__init__()</code>	46
8.7.3	Member Function Documentation	46
8.7.3.1	<code>__find_aixh_blocks()</code>	46
8.7.3.2	<code>__perform_maximal_munch()</code>	47
8.7.3.3	<code>__str__()</code>	47
8.7.3.4	<code>_build_naive_ir()</code>	47
8.7.3.5	<code>_read_model_graph()</code>	48
8.7.3.6	<code>build_ir()</code>	48
8.7.4	Member Data Documentation	49
8.7.4.1	<code>_ir_graph</code>	49
8.7.4.2	<code>_ir_symtab</code>	49
8.7.4.3	<code>_md</code>	49
8.7.4.4	<code>_tf_graph</code>	50
8.8	<code>AxfclRGraph.AxfclRGraph</code> Class Reference	50
8.8.1	Detailed Description	50



8.8.2	Constructor & Destructor Documentation . . . . .	51
8.8.2.1	__init__() . . . . .	51
8.8.3	Member Function Documentation . . . . .	51
8.8.3.1	__str__() . . . . .	51
8.8.3.2	analyse_liveness() . . . . .	51
8.8.3.3	append_block() . . . . .	52
8.8.3.4	append_node() . . . . .	52
8.8.3.5	dump_to_file() . . . . .	52
8.8.4	Member Data Documentation . . . . .	53
8.8.4.1	blocks . . . . .	53
8.8.4.2	nodes . . . . .	53
8.8.4.3	root_node . . . . .	53
8.8.4.4	syntab . . . . .	54
8.9	AxfclRNode.AxfclRNode Class Reference . . . . .	54
8.9.1	Detailed Description . . . . .	55
8.9.2	Constructor & Destructor Documentation . . . . .	55
8.9.2.1	__init__() . . . . .	55
8.9.2.2	__del__() . . . . .	55
8.9.3	Member Function Documentation . . . . .	56
8.9.3.1	__eq__() . . . . .	56
8.9.3.2	__hash__() . . . . .	56
8.9.3.3	__str__() . . . . .	56
8.9.3.4	analyze_profit() . . . . .	57
8.9.4	Member Data Documentation . . . . .	57
8.9.4.1	aix_layer . . . . .	57
8.9.4.2	aixh_profit . . . . .	57
8.9.4.3	block_ref . . . . .	57
8.9.4.4	eval_flag . . . . .	58
8.9.4.5	id . . . . .	58
8.9.4.6	is_aixh_support . . . . .	58

8.9.4.7	<code>is_input</code>	58
8.9.4.8	<code>is_output</code>	58
8.9.4.9	<code>layer_id</code>	59
8.9.4.10	<code>name</code>	59
8.9.4.11	<code>node_def</code>	59
8.9.4.12	<code>op</code>	59
8.9.4.13	<code>preds</code>	59
8.9.4.14	<code>succs</code>	60
8.10	<code>AxfclRTranslator.AxfclRTranslator Class Reference</code>	60
8.10.1	Detailed Description	61
8.10.2	Constructor & Destructor Documentation	61
8.10.2.1	<code>__init__()</code>	62
8.10.3	Member Function Documentation	62
8.10.3.1	<code>__emit_aixh_block()</code>	62
8.10.3.2	<code>__emit_aixh_node()</code>	62
8.10.3.3	<code>__str__()</code>	63
8.10.3.4	<code>_emit_aix_convolution_desc()</code>	63
8.10.3.5	<code>_emit_aix_layer_activation()</code>	63
8.10.3.6	<code>_emit_aix_layer_avgpool()</code>	63
8.10.3.7	<code>_emit_aix_layer_batchnorm()</code>	64
8.10.3.8	<code>_emit_aix_layer_biasadd()</code>	64
8.10.3.9	<code>_emit_aix_layer_convolution()</code>	64
8.10.3.10	<code>_emit_aix_layer_ewadd()</code>	64
8.10.3.11	<code>_emit_aix_layer_group_conv()</code>	64
8.10.3.12	<code>_emit_aix_layer_maxpool()</code>	65
8.10.3.13	<code>_emit_aix_layer_softmax()</code>	65
8.10.3.14	<code>_emit_aix_sampling_desc()</code>	65
8.10.3.15	<code>_emit_aix_tensor_bias()</code>	65
8.10.3.16	<code>_emit_aix_tensor_filter()</code>	65
8.10.3.17	<code>_emit_aix_tensor_input()</code>	66

8.10.3.18	<code>_emit_aix_tensor_mean()</code>	66
8.10.3.19	<code>_emit_aix_tensor_output()</code>	66
8.10.3.20	<code>_emit_aix_tensor_scale()</code>	66
8.10.3.21	<code>_emit_aix_tensor_variance()</code>	66
8.10.3.22	<code>_get_emitted_input_nodes()</code>	66
8.10.3.23	<code>emit_aixh_graphs()</code>	67
8.10.4	Member Data Documentation	67
8.10.4.1	<code>__emit_aix_layer_tbl</code>	67
8.10.4.2	<code>_aix_graph</code>	68
8.10.4.3	<code>_ir_symtab</code>	68
8.10.4.4	<code>_md</code>	68
8.10.4.5	<code>aix_graphs</code>	68
8.11	<code>AxhcLauncherWriter.AxhcLauncherWriter</code> Class Reference	68
8.11.1	Detailed Description	69
8.11.2	Constructor & Destructor Documentation	69
8.11.2.1	<code>__init__()</code>	69
8.11.3	Member Function Documentation	69
8.11.3.1	<code>__str__()</code>	69
8.11.3.2	<code>emit_aixh_launcher()</code>	69
8.11.4	Member Data Documentation	70
8.11.4.1	<code>__ir_graph</code>	70
8.11.4.2	<code>__md</code>	70
8.12	<code>AxhcMachineDesc.AxhcMachineDesc</code> Class Reference	70
8.12.1	Detailed Description	71
8.12.2	Constructor & Destructor Documentation	71
8.12.2.1	<code>__init__()</code>	71
8.12.3	Member Function Documentation	72
8.12.3.1	<code>__str__()</code>	72
8.12.3.2	<code>get_aixh_support()</code>	72
8.12.3.3	<code>get_in_type()</code>	72

8.12.3.4	<a href="#">get_layer_info()</a>	73
8.12.3.5	<a href="#">get_profit_threshold()</a>	73
8.12.3.6	<a href="#">read_file()</a>	74
8.12.4	<a href="#">Member Data Documentation</a>	74
8.12.4.1	<a href="#">__aix_layer_info_tbl</a>	74
8.12.4.2	<a href="#">__aix_model_info_tbl</a>	74
8.12.4.3	<a href="#">DEFAULT_PROFIT_THRESHOLD</a>	75
8.12.4.4	<a href="#">TYPE_MXNET</a>	75
8.12.4.5	<a href="#">TYPE_PYTORCH</a>	75
8.12.4.6	<a href="#">TYPE_TENSORFLOW</a>	75
8.12.4.7	<a href="#">TYPE_UNKNOWN</a>	75
8.13	<a href="#">AxcTFIRBuilder.AxcTFIRBuilder Class Reference</a>	76
8.13.1	<a href="#">Detailed Description</a>	76
8.13.2	<a href="#">Constructor &amp; Destructor Documentation</a>	77
8.13.2.1	<a href="#">__init__()</a>	77
8.13.3	<a href="#">Member Function Documentation</a>	77
8.13.3.1	<a href="#">__append_node_def()</a>	77
8.13.3.2	<a href="#">__prune_ir_nodes()</a>	77
8.13.3.3	<a href="#">__str__()</a>	78
8.13.3.4	<a href="#">_build_naive_ir()</a>	78
8.13.3.5	<a href="#">_read_model_graph()</a>	78
8.13.4	<a href="#">Member Data Documentation</a>	79
8.13.4.1	<a href="#">_tf_graph</a>	79
8.14	<a href="#">AxcTFIRTranslator.AxcTFIRTranslator Class Reference</a>	79
8.14.1	<a href="#">Detailed Description</a>	81
8.14.2	<a href="#">Constructor &amp; Destructor Documentation</a>	81
8.14.2.1	<a href="#">__init__()</a>	81
8.14.3	<a href="#">Member Function Documentation</a>	81
8.14.3.1	<a href="#">__get_aix_data_type()</a>	81
8.14.3.2	<a href="#">__get_aix_tensor_dims()</a>	82

8.14.3.3	<a href="#">__get_aix_tensor_format()</a>	82
8.14.3.4	<a href="#">__get_values_of_format()</a>	82
8.14.3.5	<a href="#">_emit_aix_convolution_desc()</a>	83
8.14.3.6	<a href="#">_emit_aix_layer_activation()</a>	83
8.14.3.7	<a href="#">_emit_aix_layer_avgpool()</a>	84
8.14.3.8	<a href="#">_emit_aix_layer_batchnorm()</a>	84
8.14.3.9	<a href="#">_emit_aix_layer_biasadd()</a>	85
8.14.3.10	<a href="#">_emit_aix_layer_convolution()</a>	85
8.14.3.11	<a href="#">_emit_aix_layer_ewadd()</a>	86
8.14.3.12	<a href="#">_emit_aix_layer_group_conv()</a>	86
8.14.3.13	<a href="#">_emit_aix_layer_maxpool()</a>	87
8.14.3.14	<a href="#">_emit_aix_layer_softmax()</a>	87
8.14.3.15	<a href="#">_emit_aix_sampling_desc()</a>	87
8.14.3.16	<a href="#">_emit_aix_tensor_bias()</a>	89
8.14.3.17	<a href="#">_emit_aix_tensor_filter()</a>	89
8.14.3.18	<a href="#">_emit_aix_tensor_input()</a>	90
8.14.3.19	<a href="#">_emit_aix_tensor_mean()</a>	90
8.14.3.20	<a href="#">_emit_aix_tensor_output()</a>	91
8.14.3.21	<a href="#">_emit_aix_tensor_scale()</a>	91
8.14.3.22	<a href="#">_emit_aix_tensor_variance()</a>	91
<b>9</b>	<b>File Documentation</b>	<b>93</b>
9.1	<a href="#">/home/youngsun/Project/SKT-AIX/Development/aixc/README.md File Reference</a>	93
9.2	<a href="#">/home/youngsun/Project/SKT-AIX/Development/aixc/src/aixh_pb2.py File Reference</a>	93
9.3	<a href="#">/home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcError.py File Reference</a>	94
9.4	<a href="#">/home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcFrontendCompiler.py File Reference</a>	94
9.5	<a href="#">/home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcGraphWriter.py File Reference</a>	94
9.6	<a href="#">/home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcIRBlock.py File Reference</a>	94
9.7	<a href="#">/home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcIRBuilder.py File Reference</a>	95
9.8	<a href="#">/home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcIRGraph.py File Reference</a>	95
9.9	<a href="#">/home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcIRNode.py File Reference</a>	95
9.10	<a href="#">/home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcIRTranslator.py File Reference</a>	96
9.11	<a href="#">/home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcLauncherWriter.py File Reference</a>	96
9.12	<a href="#">/home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcMachineDesc.py File Reference</a>	96
9.13	<a href="#">/home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcMain.py File Reference</a>	96
9.14	<a href="#">/home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcTFIRBuilder.py File Reference</a>	97
9.15	<a href="#">/home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcTFIRTranslator.py File Reference</a>	97
9.16	<a href="#">/home/youngsun/Project/SKT-AIX/Development/aixc/tst/aix_tf.md File Reference</a>	97
	<b>Index</b>	<b>99</b>



# Chapter 1

## SKT AIX Frontend Compiler

This README describes the organization and usage of the SKT AIX Frontend Compiler.

### 1. Source Organization

#### Common

- [AxfcFrontendCompiler](#)
- [AxfcIRBuilder](#)
- [AxfcIRTranslator](#)
- [AxfcMachineDesc](#)
- [AxfcGraphWriter](#)
- [AxfcIRGraph](#)
- [AxfcIRBlock](#)
- [AxfcIRNode](#)
- [AxfcError](#)
- [AxfcMain](#)

#### Tensorflow

- [AxfcTFIRBuilder](#)
- [AxfcTFIRTranslator](#)

#### SKT-AIX

- [aixh\\_pb2](#)

## 2. Usage

```
$$ python3 AxfcMain.py [-m] [-i] [-o] [-l] [-g]

-m: Path to a machine description file
-i: Path to the protocol buffer of a frozen model
-o: Path to output the generated AIXGraph
-l: Path to log out (optional)
-g: Path to dump out an IR graph (optional)
```

## 3. Contact

- Youngsun Han ([youngsun@pknu.ac.kr](mailto:youngsun@pknu.ac.kr))
- Associate Professor
- Department of Computer Engineering, Pukyong National University



## Chapter 2

### aix\_tf

```
{ "AIX_MODEL_TYPE": "TENSORFLOW", "AIX_PROFIT_THRESHOLD": 500, "AIX_LAYER": { "Conv2D": {  
  "layer": "AIX_LAYER_CONVOLUTION", "activation": "AIX_ACTIVATION_IDENTITY", "is_group": false, "is_conv": true, "profit": 100 }, "DepthwiseConv2dNative": { "layer": "AIX_LAYER_GROUP_CONV", "activation":  
  "AIX_ACTIVATION_IDENTITY", "is_group": true, "is_conv": true, "profit": 100 }, "FusedBatchNorm": { "layer":  
  "AIX_LAYER_BATCHNORM", "activation": null, "is_group": false, "is_conv": false, "profit": 100 }, "BatchNorm": {  
  "layer": "AIX_LAYER_BATCHNORM", "activation": null, "is_group": false, "is_conv": false, "profit": 100 }, "AvgPool": { "layer": "AIX_LAYER_AVGPOOL", "activation": null, "is_group": false, "is_conv": false, "profit": 100 },  
  "MaxPool": { "layer": "AIX_LAYER_MAXPOOL", "activation": null, "is_group": false, "is_conv": false, "profit": 100 },  
  "Softmax": { "layer": "AIX_LAYER_SOFTMAX", "activation": null, "is_group": false, "is_conv": false, "profit": 100 },  
  "Add": { "layer": "AIX_LAYER_EWADD", "activation": null, "is_group": false, "is_conv": false, "profit": 100 }, "Relu":  
  { "layer": "AIX_LAYER_ACTIVATION", "activation": "AIX_ACTIVATION_RELU", "is_group": false, "is_conv": false,  
    "profit": 100 }, "Relu6": { "layer": "AIX_LAYER_ACTIVATION", "activation": "AIX_ACTIVATION_LEAKY_RELU",  
    "is_group": false, "is_conv": false, "profit": 100 }, "BiasAdd": { "layer": "AIX_LAYER_BIASADD", "activation": null,  
    "is_group": false, "is_conv": false, "profit": 100 }, "Sigmoid": { "layer": "AIX_LAYER_ACTIVATION", "activation":  
    "AIX_ACTIVATION_SIGMOID", "is_group": false, "is_conv": false, "profit": 100 }, "Prelu": { "layer": "AIX_LAYER_ACTIVATION", "activation": "AIX_ACTIVATION_PRELU", "is_group": false, "is_conv": false, "profit": 100 },  
  "Tanh": { "layer": "AIX_LAYER_ACTIVATION", "activation": "AIX_ACTIVATION_TANH", "is_group": false, "is_conv": false, "profit": 100 } }
```



## Chapter 3

# Namespace Index

### 3.1 Packages

Here are the packages with brief descriptions (if available):

<a href="#">aixh_pb2</a>	13
<a href="#">AxfcError</a>	17
<a href="#">AxfcFrontendCompiler</a>	18
<a href="#">AxfcGraphWriter</a>	18
<a href="#">AxfcIRBlock</a>	18
<a href="#">AxfcIRBuilder</a>	18
<a href="#">AxfcIRGraph</a>	18
<a href="#">AxfcIRNode</a>	18
<a href="#">AxfcIRTranslator</a>	19
<a href="#">AxfcLauncherWriter</a>	19
<a href="#">AxfcMachineDesc</a>	19
<a href="#">AxfcMain</a>	19
<a href="#">AxfcTFIRBuilder</a>	21
<a href="#">AxfcTFIRTranslator</a>	21



## Chapter 4

# Hierarchical Index

### 4.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

AxfcMachineDesc.AxfcMachineDesc.AIXLayerInfo . . . . .	23
AxfcFrontendCompiler.AxfcFrontendCompiler . . . . .	34
AxfcGraphWriter.AxfcGraphWriter . . . . .	38
AxfcIRBlock.AxfcIRBlock . . . . .	41
AxfcIRBuilder.AxfcIRBuilder . . . . .	45
AxfcTFIRBuilder.AxfcTFIRBuilder . . . . .	76
AxfcIRGraph.AxfcIRGraph . . . . .	50
AxfcIRNode.AxfcIRNode . . . . .	54
AxfcIRTranslator.AxfcIRTranslator . . . . .	60
AxfcTFIRTranslator.AxfcTFIRTranslator . . . . .	79
AxfcLauncherWriter.AxfcLauncherWriter . . . . .	68
AxfcMachineDesc.AxfcMachineDesc . . . . .	70
Enum	
AxfcError.AxfcError . . . . .	28
AxfcIRTranslator.AIXTensorType . . . . .	26



## Chapter 5

# Class Index

### 5.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">AxfcMachineDesc.AxfcMachineDesc.AIXLayerInfo</a>	
<a href="#">AIXLayerInfo</a> inner class . . . . .	23
<a href="#">AxfcIRTranslator.AIXTensorType</a>	
<a href="#">AIXInputType</a> enum class . . . . .	26
<a href="#">AxfcError.AxfcError</a>	
<a href="#">AxfcError</a> enum class . . . . .	28
<a href="#">AxfcFrontendCompiler.AxfcFrontendCompiler</a>	
<a href="#">AxfcFrontendCompiler</a> . . . . .	34
<a href="#">AxfcGraphWriter.AxfcGraphWriter</a>	
<a href="#">AxfcGraphWriter</a> class . . . . .	38
<a href="#">AxfcIRBlock.AxfcIRBlock</a>	
<a href="#">AxfcIRBlock</a> class . . . . .	41
<a href="#">AxfcIRBuilder.AxfcIRBuilder</a>	
<a href="#">AxfcIRBuilder</a> class . . . . .	45
<a href="#">AxfcIRGraph.AxfcIRGraph</a>	
<a href="#">AxfcIRGraph</a> class . . . . .	50
<a href="#">AxfcIRNode.AxfcIRNode</a>	
<a href="#">AxfcIRNode</a> . . . . .	54
<a href="#">AxfcIRTranslator.AxfcIRTranslator</a> . . . . .	60
<a href="#">AxfcLauncherWriter.AxfcLauncherWriter</a>	
<a href="#">AxfcLauncherWriter</a> class . . . . .	68
<a href="#">AxfcMachineDesc.AxfcMachineDesc</a>	
<a href="#">AxfcMachineDesc</a> class . . . . .	70
<a href="#">AxfcTFIRBuilder.AxfcTFIRBuilder</a>	
<a href="#">AxfcTFIRBuilder</a> class . . . . .	76
<a href="#">AxfcTFIRTranslator.AxfcTFIRTranslator</a>	
<a href="#">AxfcTFIRTranslator</a> class . . . . .	79





## Chapter 6

# File Index

### 6.1 File List

Here is a list of all files with brief descriptions:

/home/youngsun/Project/SKT-AIX/Development/aixc/src/aixh_pb2.py . . . . .	93
/home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcError.py . . . . .	94
/home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcFrontendCompiler.py . . . . .	94
/home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcGraphWriter.py . . . . .	94
/home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfclRBlock.py . . . . .	94
/home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfclRBuilder.py . . . . .	95
/home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfclRGraph.py . . . . .	95
/home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfclRNode.py . . . . .	95
/home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfclRTranslator.py . . . . .	96
/home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcLauncherWriter.py . . . . .	96
/home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcMachineDesc.py . . . . .	96
/home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcMain.py . . . . .	96
/home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcTFIRBuilder.py . . . . .	97
/home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcTFIRTranslator.py . . . . .	97



## Chapter 7

# Namespace Documentation

### 7.1 aixh\_pb2 Namespace Reference

#### Variables

- `int _b = sys.version_info[0] < 3 and (lambda x: x) or (lambda x: x.encode('latin1'))`
- `_sym_db = _symbol_database.Default()`
- `DESCRIPTOR`
- `_AIXLAYER_AIXLAYERTYPE`
- `_AIXLAYER_AIXACTIVATIONMODE`
- `_AIXLAYER_AIXSAMPLINGMODE`
- `_AIXLAYER_AIXDATATYPE`
- `_AIXLAYER_AIXTENSORFORMAT`
- `_AIXLAYER_AIXCONVOLUTIONDESC`
- `_AIXLAYER_AIXSAMPLINGDESC`
- `_AIXLAYER_AIXEWADDDDESC`
- `_AIXLAYER_AIXTENSOR`
- `_AIXLAYER`
- `_AIXGRAPH`
- `enum_type`
- `containing_type`
- `message_type`
- `AIXLayer`
- `AIXGraph`
- `has_options`
- `_options`

#### 7.1.1 Variable Documentation

##### 7.1.1.1 `_AIXGRAPH`

`aixh_pb2._AIXGRAPH` [private]

Definition at line 678 of file `aixh_pb2.py`.

### 7.1.1.2 `_AIXLAYER`

`aixh_pb2._AIXLAYER` [private]

Definition at line 509 of file `aixh_pb2.py`.

### 7.1.1.3 `_AIXLAYER_AIXACTIVATIONMODE`

`aixh_pb2._AIXLAYER_AIXACTIVATIONMODE` [private]

Definition at line 158 of file `aixh_pb2.py`.

### 7.1.1.4 `_AIXLAYER_AIXCONVOLUTIONDESC`

`aixh_pb2._AIXLAYER_AIXCONVOLUTIONDESC` [private]

Definition at line 298 of file `aixh_pb2.py`.

### 7.1.1.5 `_AIXLAYER_AIXDATATYPE`

`aixh_pb2._AIXLAYER_AIXDATATYPE` [private]

Definition at line 230 of file `aixh_pb2.py`.

### 7.1.1.6 `_AIXLAYER_AIXEWADDDDESC`

`aixh_pb2._AIXLAYER_AIXEWADDDDESC` [private]

**Initial value:**

```
1 = _descriptor.Descriptor(
2     name='AIXEWAddDesc',
3     full_name='aixh.AIXLayer.AIXEWAddDesc',
4     filename=None,
5     file=DESCRIPTOR,
6     containing_type=None,
7     fields=[
8         _descriptor.FieldDescriptor(
9             name='scale', full_name='aixh.AIXLayer.AIXEWAddDesc.scale', index=0,
10            number=1, type=2, cpp_type=6, label=3,
11            has_default_value=False, default_value=[],
12            message_type=None, enum_type=None, containing_type=None,
13            is_extension=False, extension_scope=None,
14            options=None),
15    ],
16    extensions=[
17    ],
18    nested_types=[],
19    enum_types=[
20    ],
21    options=None,
22    is_extendable=False,
23    syntax='proto2',
24    extension_ranges=[],
25    oneofs=[
26    ],
27    serialized_start=972,
28    serialized_end=1001,
29 )
```

Definition at line 407 of file `aixh_pb2.py`.

## 7.1.1.7 \_AIXLAYER\_AIXLAYERTYPE

aixh\_pb2.\_AIXLAYER\_AIXLAYERTYPE [private]

Definition at line 72 of file aixh\_pb2.py.

## 7.1.1.8 \_AIXLAYER\_AIXSAMPLINGDESC

aixh\_pb2.\_AIXLAYER\_AIXSAMPLINGDESC [private]

Definition at line 356 of file aixh\_pb2.py.

## 7.1.1.9 \_AIXLAYER\_AIXSAMPLINGMODE

aixh\_pb2.\_AIXLAYER\_AIXSAMPLINGMODE [private]

Definition at line 196 of file aixh\_pb2.py.

## 7.1.1.10 \_AIXLAYER\_AIXTENSOR

aixh\_pb2.\_AIXLAYER\_AIXTENSOR [private]

Definition at line 437 of file aixh\_pb2.py.

## 7.1.1.11 \_AIXLAYER\_AIXTENSORFORMAT

aixh\_pb2.\_AIXLAYER\_AIXTENSORFORMAT [private]

**Initial value:**

```

1 = _descriptor.EnumDescriptor(
2     name='AIXTensorFormat',
3     full_name='aixh.AIXLayer.AIXTensorFormat',
4     filename=None,
5     file=DESCRIPTOR,
6     values=[
7         _descriptor.EnumValueDescriptor(
8             name='AIX_FORMAT_NCHW', index=0, number=0,
9             options=None,
10            type=None),
11        _descriptor.EnumValueDescriptor(
12            name='AIX_FORMAT_NHWC', index=1, number=1,
13            options=None,
14            type=None),
15        _descriptor.EnumValueDescriptor(
16            name='AIX_FORMAT_NWHC', index=2, number=2,
17            options=None,
18            type=None),
19        _descriptor.EnumValueDescriptor(
20            name='AIX_FORMAT_VECTOR', index=3, number=3,
21            options=None,
22            type=None),
23    ],
24    containing_type=None,
25    options=None,
26    serialized_start=2094,
27    serialized_end=2197,
28 )

```

Definition at line 268 of file aixh\_pb2.py.

#### 7.1.1.12 `_b`

```
int aixh_pb2._b = sys.version_info[0] < 3 and (lambda x: x) or (lambda x: x.encode('latin1'))  
[private]
```

Definition at line 6 of file `aixh_pb2.py`.

#### 7.1.1.13 `_options`

```
aixh_pb2._options [private]
```

Definition at line 798 of file `aixh_pb2.py`.

#### 7.1.1.14 `_sym_db`

```
aixh_pb2._sym_db = _symbol_database.Default() [private]
```

Definition at line 15 of file `aixh_pb2.py`.

#### 7.1.1.15 `AIXGraph`

```
aixh_pb2.AIXGraph
```

**Initial value:**

```
1 = _reflection.GeneratedProtocolMessageType('AIXGraph', (_message.Message,), dict(  
2     DESCRIPTOR=_AIXGRAPH,  
3     __module__='aixh_pb2'  
4     # @@protoc_insertion_point(class_scope:aixh.AIXGraph)  
5 ))
```

Definition at line 790 of file `aixh_pb2.py`.

#### 7.1.1.16 `AIXLayer`

```
aixh_pb2.AIXLayer
```

Definition at line 751 of file `aixh_pb2.py`.

#### 7.1.1.17 containing\_type

`aixh_pb2.containing_type`

Definition at line 723 of file `aixh_pb2.py`.

#### 7.1.1.18 DESCRIPTOR

`aixh_pb2.DESRIPTOR`

Definition at line 17 of file `aixh_pb2.py`.

#### 7.1.1.19 enum\_type

`aixh_pb2.enum_type`

Definition at line 722 of file `aixh_pb2.py`.

#### 7.1.1.20 has\_options

`aixh_pb2.has_options`

Definition at line 797 of file `aixh_pb2.py`.

#### 7.1.1.21 message\_type

`aixh_pb2.message_type`

Definition at line 731 of file `aixh_pb2.py`.

## 7.2 AxfcError Namespace Reference

### Classes

- class [AxfcError](#)  
*[AxfcError](#) enum class.*

## 7.3 AxfcFrontendCompiler Namespace Reference

### Classes

- class [AxfcFrontendCompiler](#)  
*[AxfcFrontendCompiler](#).*

## 7.4 AxfcGraphWriter Namespace Reference

### Classes

- class [AxfcGraphWriter](#)  
*[AxfcGraphWriter](#) class.*

## 7.5 AxfcIRBlock Namespace Reference

### Classes

- class [AxfcIRBlock](#)  
*[AxfcIRBlock](#) class.*

## 7.6 AxfcIRBuilder Namespace Reference

### Classes

- class [AxfcIRBuilder](#)  
*[AxfcIRBuilder](#) class.*

## 7.7 AxfcIRGraph Namespace Reference

### Classes

- class [AxfcIRGraph](#)  
*[AxfcIRGraph](#) class.*

## 7.8 AxfcIRNode Namespace Reference

### Classes

- class [AxfcIRNode](#)  
*[AxfcIRNode](#).*



## 7.9 AxfclRTranslator Namespace Reference

### Classes

- class [AIXTensorType](#)  
*AIXInputType enum class.*
- class [AxfclRTranslator](#)

## 7.10 AxfclLauncherWriter Namespace Reference

### Classes

- class [AxfclLauncherWriter](#)  
*AxfclLauncherWriter class.*

## 7.11 AxfclMachineDesc Namespace Reference

### Classes

- class [AxfclMachineDesc](#)  
*AxfclMachineDesc class.*

## 7.12 AxfclMain Namespace Reference

### Functions

- def [\\_\\_main](#) (vargs)  
*main function*

### Variables

- [parser](#)
- [metavar](#)
- [type](#)
- [str](#)
- [required](#)
- [help](#)
- [args](#) = parser.parse\_args()

### 7.12.1 Function Documentation

#### 7.12.1.1 `__main()`

```
def AxfcMain.__main (
    vargs ) [private]
```

main function

Definition at line 20 of file AxfcMain.py.

### 7.12.2 Variable Documentation

#### 7.12.2.1 `args`

```
AxfcMain.args = parser.parse_args()
```

Definition at line 100 of file AxfcMain.py.

#### 7.12.2.2 `help`

```
AxfcMain.help
```

Definition at line 90 of file AxfcMain.py.

#### 7.12.2.3 `metavar`

```
AxfcMain.metavar
```

Definition at line 89 of file AxfcMain.py.

#### 7.12.2.4 `parser`

```
AxfcMain.parser
```

**Initial value:**

```
1 = argparse.ArgumentParser(
2     description='SKT AIX Frontend Compiler',
3     usage='use "%(prog)s -h/--help" for more information')
```

Definition at line 85 of file AxfcMain.py.

#### 7.12.2.5 required

`AxfcMain.required`

Definition at line 89 of file `AxfcMain.py`.

#### 7.12.2.6 str

`AxfcMain.str`

Definition at line 89 of file `AxfcMain.py`.

#### 7.12.2.7 type

`AxfcMain.type`

Definition at line 89 of file `AxfcMain.py`.

## 7.13 AxfcTFIRBuilder Namespace Reference

### Classes

- class [AxfcTFIRBuilder](#)  
*AxfcTFIRBuilder class.*

## 7.14 AxfcTFIRTranslator Namespace Reference

### Classes

- class [AxfcTFIRTranslator](#)  
*AxfcTFIRTranslator class.*

### Variables

- dictionary [aix\\_data\\_type\\_tbl](#)  
*Global tables for AIXDataType and AIXTensorFormat.*
- dictionary [aix\\_tensor\\_format\\_tbl](#)  
*AIXTensorFormat table.*

### 7.14.1 Variable Documentation

#### 7.14.1.1 aix\_data\_type\_tbl

dictionary `AxfcTFIRTranslator.aix_data_type_tbl`

##### Initial value:

```
1 = {
2     tf.float16: AIXLayer.AIXDataType.AIX_DATA_HALF,
3     tf.float32: AIXLayer.AIXDataType.AIX_DATA_FLOAT,
4     tf.float64: AIXLayer.AIXDataType.AIX_DATA_DOUBLE,
5     tf.uint8: AIXLayer.AIXDataType.AIX_DATA_UINT8,
6     tf.int8: AIXLayer.AIXDataType.AIX_DATA_SINT8,
7     tf.int16: AIXLayer.AIXDataType.AIX_DATA_SINT16
8 }
```

Global tables for `AIXDataType` and `AIXTensorFormat`.

`AIXDataType` table

Definition at line 23 of file `AxfcTFIRTranslator.py`.

#### 7.14.1.2 aix\_tensor\_format\_tbl

dictionary `AxfcTFIRTranslator.aix_tensor_format_tbl`

##### Initial value:

```
1 = {
2     b"NCHW": AIXLayer.AIXTensorFormat.AIX_FORMAT_NCHW,
3     b"NHWC": AIXLayer.AIXTensorFormat.AIX_FORMAT_NHWC,
4     b"NWHC": AIXLayer.AIXTensorFormat.AIX_FORMAT_NWHC,
5     b"VECTOR": AIXLayer.AIXTensorFormat.AIX_FORMAT_VECTOR
6 }
```

`AIXTensorFormat` table.

Definition at line 33 of file `AxfcTFIRTranslator.py`.

## Chapter 8

# Class Documentation

### 8.1 AxfcMachineDesc.AxfcMachineDesc.AIXLayerInfo Class Reference

[AIXLayerInfo](#) inner class.

#### Public Member Functions

- `def __init__ (self, op)`  
*The constructor.*
- `def __str__ (self)`  
*For debugging.*

#### Public Attributes

- [op](#)  
*layer operation name of the layer info*
- [layer](#)  
*AIX layer ID of the layer info.*
- [activation](#)  
*AIX activation ID of the layer info.*
- [is\\_group](#)  
*indicate whether this layer is group layer or not*
- [is\\_conv](#)  
*indicate whether this layer is convolution layer or not*
- [profit](#)  
*the profit that can be obtained by accelerating this layer using AIXH*

#### 8.1.1 Detailed Description

[AIXLayerInfo](#) inner class.

Definition at line 148 of file `AxfcMachineDesc.py`.

## 8.1.2 Constructor & Destructor Documentation

### 8.1.2.1 `__init__()`

```
def AxfcMachineDesc.AxfcMachineDesc.AIXLayerInfo.__init__ (
    self,
    op )
```

The constructor.

Definition at line 168 of file AxfcMachineDesc.py.

## 8.1.3 Member Function Documentation

### 8.1.3.1 `__str__()`

```
def AxfcMachineDesc.AxfcMachineDesc.AIXLayerInfo.__str__ (
    self )
```

For debugging.

Definition at line 177 of file AxfcMachineDesc.py.

## 8.1.4 Member Data Documentation

### 8.1.4.1 `activation`

```
AxfcMachineDesc.AxfcMachineDesc.AIXLayerInfo.activation
```

AIX activation ID of the layer info.

Definition at line 171 of file AxfcMachineDesc.py.

### 8.1.4.2 `is_conv`

```
AxfcMachineDesc.AxfcMachineDesc.AIXLayerInfo.is_conv
```

indicate whether this layer is convolution layer or not

Definition at line 173 of file AxfcMachineDesc.py.

#### 8.1.4.3 is\_group

`AxfcMachineDesc.AxfcMachineDesc.AIXLayerInfo.is_group`

indicate whether this layer is group layer or not

Definition at line 172 of file `AxfcMachineDesc.py`.

#### 8.1.4.4 layer

`AxfcMachineDesc.AxfcMachineDesc.AIXLayerInfo.layer`

AIX layer ID of the layer info.

Definition at line 170 of file `AxfcMachineDesc.py`.

#### 8.1.4.5 op

`AxfcMachineDesc.AxfcMachineDesc.AIXLayerInfo.op`

layer operation name of the layer info

Definition at line 169 of file `AxfcMachineDesc.py`.

#### 8.1.4.6 profit

`AxfcMachineDesc.AxfcMachineDesc.AIXLayerInfo.profit`

the profit that can be obtained by accelerating this layer using AIXH

Definition at line 174 of file `AxfcMachineDesc.py`.

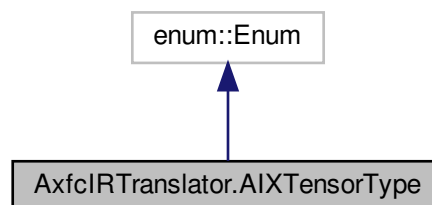
The documentation for this class was generated from the following file:

- `/home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcMachineDesc.py`

## 8.2 AxfclRTranslator.AIXTensorType Class Reference

AIXInputType enum class.

Inheritance diagram for AxfclRTranslator.AIXTensorType:



### Static Public Attributes

- int `AIX_TENSOR_INPUT` = 0
- int `AIX_TENSOR_FILTER` = 1
- int `AIX_TENSOR_BIAS` = 2
- int `AIX_TENSOR_SCALE` = 3
- int `AIX_TENSOR_MEAN` = 4
- int `AIX_TENSOR_VARIANCE` = 5
- int `AIX_TENSOR_OUTPUT` = 6
- int `AIX_TENSOR_UNKNOWN` = 7

### 8.2.1 Detailed Description

AIXInputType enum class.

Definition at line 22 of file `AxfclRTranslator.py`.

### 8.2.2 Member Data Documentation

#### 8.2.2.1 AIX\_TENSOR\_BIAS

```
int AxfclRTranslator.AIXTensorType.AIX_TENSOR_BIAS = 2 [static]
```

Definition at line 25 of file `AxfclRTranslator.py`.



### 8.2.2.2 AIX\_TENSOR\_FILTER

```
int AxfclIRTranslator.AIXTensorType.AIX_TENSOR_FILTER = 1 [static]
```

Definition at line 24 of file AxfclIRTranslator.py.

### 8.2.2.3 AIX\_TENSOR\_INPUT

```
int AxfclIRTranslator.AIXTensorType.AIX_TENSOR_INPUT = 0 [static]
```

Definition at line 23 of file AxfclIRTranslator.py.

### 8.2.2.4 AIX\_TENSOR\_MEAN

```
int AxfclIRTranslator.AIXTensorType.AIX_TENSOR_MEAN = 4 [static]
```

Definition at line 27 of file AxfclIRTranslator.py.

### 8.2.2.5 AIX\_TENSOR\_OUTPUT

```
int AxfclIRTranslator.AIXTensorType.AIX_TENSOR_OUTPUT = 6 [static]
```

Definition at line 29 of file AxfclIRTranslator.py.

### 8.2.2.6 AIX\_TENSOR\_SCALE

```
int AxfclIRTranslator.AIXTensorType.AIX_TENSOR_SCALE = 3 [static]
```

Definition at line 26 of file AxfclIRTranslator.py.

### 8.2.2.7 AIX\_TENSOR\_UNKNOWN

```
int AxfclIRTranslator.AIXTensorType.AIX_TENSOR_UNKNOWN = 7 [static]
```

Definition at line 30 of file AxfclIRTranslator.py.

### 8.2.2.8 AIX\_TENSOR\_VARIANCE

```
int AxfcIRTranslator.AIXTensorType.AIX_TENSOR_VARIANCE = 5 [static]
```

Definition at line 28 of file AxfcIRTranslator.py.

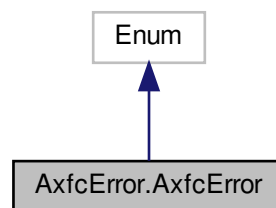
The documentation for this class was generated from the following file:

- /home/youngsun/Project/SKT-AIX/Development/aixc/src/[AxfcIRTranslator.py](#)

## 8.3 AxfcError.AxfcError Class Reference

[AxfcError](#) enum class.

Inheritance diagram for AxfcError.AxfcError:



### Static Public Attributes

- int [SUCCESS](#) = 0
- int [INVALID\\_PARAMETER](#) = 1
- int [INVALID\\_FILE\\_PATH](#) = 2
- int [INVALID\\_INPUT\\_TYPE](#) = 3
- int [INVALID\\_TF\\_GRAPH](#) = 4
- int [INVALID\\_IR\\_GRAPH](#) = 5
- int [EMPTY\\_IR\\_BLOCK](#) = 6
- int [PRED\\_NODE\\_NOT\\_FOUND](#) = 7
- int [INVALID\\_MD\\_FORMAT](#) = 8
- int [NOT\\_AIXH\\_SUPPORT](#) = 9
- int [NOT\\_IMPLEMENTED](#) = 10
- int [UNKNOWN\\_TENSOR\\_TYPE](#) = 11
- int [UNSUPPORTED\\_AIX\\_LAYER\\_EMIT](#) = 12
- int [INVALID\\_AIX\\_LAYER\\_TYPE](#) = 13
- int [INVALID\\_AIX\\_TENSOR\\_FORMAT](#) = 14
- int [INVALID\\_CONVOLUTION\\_LAYER](#) = 15
- int [INVALID\\_GROUP\\_CONV\\_LAYER](#) = 16
- int [INVALID\\_BATCHNORM\\_LAYER](#) = 17
- int [INVALID\\_ACTIVATION\\_LAYER](#) = 18

- int `INVALID_IDENTITY_LAYER` = 19
- int `INVALID_PAD_LAYER` = 20
- int `INVALID_AIX_TENSOR_INPUT` = 21
- int `DUMP_IR_GRAPH_ERROR` = 22
- int `INVALID_AIX_GRAPH` = 23
- int `INVALID_MAXPOOL_LAYER` = 24
- int `INVALID_EWADD_LAYER` = 25
- int `INVALID_BIASADD_LAYER` = 26
- int `UNREMOVED_IDENTITY` = 27

### 8.3.1 Detailed Description

`AxfcError` enum class.

Definition at line 20 of file `AxfcError.py`.

### 8.3.2 Member Data Documentation

#### 8.3.2.1 DUMP\_IR\_GRAPH\_ERROR

```
int AxfcError.AxfcError.DUMP_IR_GRAPH_ERROR = 22 [static]
```

Definition at line 49 of file `AxfcError.py`.

#### 8.3.2.2 EMPTY\_IR\_BLOCK

```
int AxfcError.AxfcError.EMPTY_IR_BLOCK = 6 [static]
```

Definition at line 29 of file `AxfcError.py`.

#### 8.3.2.3 INVALID\_ACTIVATION\_LAYER

```
int AxfcError.AxfcError.INVALID_ACTIVATION_LAYER = 18 [static]
```

Definition at line 45 of file `AxfcError.py`.

#### 8.3.2.4 INVALID\_AIX\_GRAPH

```
int AxfcError.AxfcError.INVALID_AIX_GRAPH = 23 [static]
```

Definition at line 50 of file AxfcError.py.

#### 8.3.2.5 INVALID\_AIX\_LAYER\_TYPE

```
int AxfcError.AxfcError.INVALID_AIX_LAYER_TYPE = 13 [static]
```

Definition at line 40 of file AxfcError.py.

#### 8.3.2.6 INVALID\_AIX\_TENSOR\_FORMAT

```
int AxfcError.AxfcError.INVALID_AIX_TENSOR_FORMAT = 14 [static]
```

Definition at line 41 of file AxfcError.py.

#### 8.3.2.7 INVALID\_AIX\_TENSOR\_INPUT

```
int AxfcError.AxfcError.INVALID_AIX_TENSOR_INPUT = 21 [static]
```

Definition at line 48 of file AxfcError.py.

#### 8.3.2.8 INVALID\_BATCHNORM\_LAYER

```
int AxfcError.AxfcError.INVALID_BATCHNORM_LAYER = 17 [static]
```

Definition at line 44 of file AxfcError.py.

#### 8.3.2.9 INVALID\_CONVOLUTION\_LAYER

```
int AxfcError.AxfcError.INVALID_CONVOLUTION_LAYER = 15 [static]
```

Definition at line 42 of file AxfcError.py.

#### 8.3.2.10 INVALID\_EWADD\_LAYER

```
int AxfcError.AxfcError.INVALID_EWADD_LAYER = 25 [static]
```

Definition at line 52 of file AxfcError.py.

#### 8.3.2.11 INVALID\_FILE\_PATH

```
int AxfcError.AxfcError.INVALID_FILE_PATH = 2 [static]
```

Definition at line 23 of file AxfcError.py.

#### 8.3.2.12 INVALID\_GROUP\_CONV\_LAYER

```
int AxfcError.AxfcError.INVALID_GROUP_CONV_LAYER = 16 [static]
```

Definition at line 43 of file AxfcError.py.

#### 8.3.2.13 INVALID\_IDENTITY\_LAYER

```
int AxfcError.AxfcError.INVALID_IDENTITY_LAYER = 19 [static]
```

Definition at line 46 of file AxfcError.py.

#### 8.3.2.14 INVALID\_INPUT\_TYPE

```
int AxfcError.AxfcError.INVALID_INPUT_TYPE = 3 [static]
```

Definition at line 26 of file AxfcError.py.

#### 8.3.2.15 INVALID\_IR\_GRAPH

```
int AxfcError.AxfcError.INVALID_IR_GRAPH = 5 [static]
```

Definition at line 28 of file AxfcError.py.

#### 8.3.2.16 INVALID\_MAXPOOL\_LAYER

```
int AxfcError.AxfcError.INVALID_MAXPOOL_LAYER = 24 [static]
```

Definition at line 51 of file AxfcError.py.

#### 8.3.2.17 INVALID\_MD\_FORMAT

```
int AxfcError.AxfcError.INVALID_MD_FORMAT = 8 [static]
```

Definition at line 33 of file AxfcError.py.

#### 8.3.2.18 INVALID\_PAD\_LAYER

```
int AxfcError.AxfcError.INVALID_PAD_LAYER = 20 [static]
```

Definition at line 47 of file AxfcError.py.

#### 8.3.2.19 INVALID\_PARAMETER

```
int AxfcError.AxfcError.INVALID_PARAMETER = 1 [static]
```

Definition at line 22 of file AxfcError.py.

#### 8.3.2.20 INVALID\_TF\_GRAPH

```
int AxfcError.AxfcError.INVALID_TF_GRAPH = 4 [static]
```

Definition at line 27 of file AxfcError.py.

#### 8.3.2.21 INVALID\_BIASADD\_LAYER

```
int AxfcError.AxfcError.INVALID_BIASADD_LAYER = 26 [static]
```

Definition at line 53 of file AxfcError.py.

#### 8.3.2.22 NOT\_AIXH\_SUPPORT

```
int AxfcError.AxfcError.NOT_AIXH_SUPPORT = 9 [static]
```

Definition at line 34 of file AxfcError.py.

#### 8.3.2.23 NOT\_IMPLEMENTED

```
int AxfcError.AxfcError.NOT_IMPLEMENTED = 10 [static]
```

Definition at line 35 of file AxfcError.py.

#### 8.3.2.24 PRED\_NODE\_NOT\_FOUND

```
int AxfcError.AxfcError.PRED_NODE_NOT_FOUND = 7 [static]
```

Definition at line 30 of file AxfcError.py.

#### 8.3.2.25 SUCCESS

```
int AxfcError.AxfcError.SUCCESS = 0 [static]
```

Definition at line 21 of file AxfcError.py.

#### 8.3.2.26 UNKNOWN\_TENSOR\_TYPE

```
int AxfcError.AxfcError.UNKNOWN_TENSOR_TYPE = 11 [static]
```

Definition at line 36 of file AxfcError.py.

#### 8.3.2.27 UNREMOVED\_IDENTITY

```
int AxfcError.AxfcError.UNREMOVED_IDENTITY = 27 [static]
```

Definition at line 54 of file AxfcError.py.

### 8.3.2.28 UNSUPPORTED\_AIX\_LAYER\_EMIT

```
int AxfcError.AxfcError.UNSUPPORTED_AIX_LAYER_EMIT = 12 [static]
```

Definition at line 39 of file AxfcError.py.

The documentation for this class was generated from the following file:

- /home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcError.py

## 8.4 AxfcFrontendCompiler.AxfcFrontendCompiler Class Reference

[AxfcFrontendCompiler](#).

### Public Member Functions

- def [\\_\\_init\\_\\_](#) (self)  
*The constructor.*
- def [get\\_ir\\_graph](#) (self)  
*This method returns the IR graph.*
- def [read\\_md\\_file](#)  
*This method is used to read a machine description in the given path.*
- def [compile](#)  
*This method is used to compile an input AI network model into an AIXGraph object.*
- def [dump\\_aix\\_graphs](#)  
*This method is used to dump out the generated AIXGraphs.*
- def [dump\\_launcher](#)
- def [\\_\\_str\\_\\_](#) (self)  
*For debugging.*

### Private Attributes

- [\\_\\_md](#)  
*machine description object*
- [\\_\\_ir\\_builder](#)  
*AIXIR builder.*
- [\\_\\_ir\\_translator](#)  
*AIXIR-to-AIXGraph translator.*

### 8.4.1 Detailed Description

[AxfcFrontendCompiler](#).

Definition at line 21 of file AxfcFrontendCompiler.py.



## 8.4.2 Constructor & Destructor Documentation

### 8.4.2.1 `__init__()`

```
def AxfcFrontendCompiler.AxfcFrontendCompiler.__init__ (
    self )
```

The constructor.

Definition at line 33 of file AxfcFrontendCompiler.py.

## 8.4.3 Member Function Documentation

### 8.4.3.1 `__str__()`

```
def AxfcFrontendCompiler.AxfcFrontendCompiler.__str__ (
    self )
```

For debugging.

Definition at line 120 of file AxfcFrontendCompiler.py.

### 8.4.3.2 `compile()`

```
def AxfcFrontendCompiler.AxfcFrontendCompiler.compile (
    self,
    path )
```

This method is used to compile an input AI network model into an AIXGraph object.

#### Parameters

<i>self</i>	this object
<i>path</i>	file path of an input AI network model

#### Returns

error info and an AIXGraph objects

Definition at line 61 of file AxfcFrontendCompiler.py.

#### 8.4.3.3 dump\_aix\_graphs()

```
def AxfcFrontendCompiler.AxfcFrontendCompiler.dump_aix_graphs (
    self,
    out_path )
```

This method is used to dump out the generated AIXGraphs.

##### Parameters

<i>self</i>	this object
<i>out_path</i>	a file path to output the AIXGraphs
<i>aix_graphs</i>	a list of AIXGraphs to be dumped out

##### Returns

error info

Definition at line 96 of file AxfcFrontendCompiler.py.

#### 8.4.3.4 dump\_launcher()

```
def AxfcFrontendCompiler.AxfcFrontendCompiler.dump_launcher (
    self,
    path )
```

Definition at line 116 of file AxfcFrontendCompiler.py.

#### 8.4.3.5 get\_ir\_graph()

```
def AxfcFrontendCompiler.AxfcFrontendCompiler.get_ir_graph (
    self )
```

This method returns the IR graph.

##### Parameters

<i>self</i>	this object
-------------	-------------

##### Returns

the IR graph

Definition at line 42 of file AxfcFrontendCompiler.py.

#### 8.4.3.6 read\_md\_file()

```
def AxfcFrontendCompiler.AxfcFrontendCompiler.read_md_file (
    self,
    path )
```

This method is used to read a machine description in the given path.

##### Parameters

<i>self</i>	this object
<i>path</i>	file path of AIXH machine description

##### Returns

an AXIGraph object

Definition at line 50 of file AxfcFrontendCompiler.py.

### 8.4.4 Member Data Documentation

#### 8.4.4.1 \_\_ir\_builder

```
AxfcFrontendCompiler.AxfcFrontendCompiler.__ir_builder [private]
```

AIXIR builder.

Definition at line 35 of file AxfcFrontendCompiler.py.

#### 8.4.4.2 \_\_ir\_translator

```
AxfcFrontendCompiler.AxfcFrontendCompiler.__ir_translator [private]
```

AIXIR-to-AIXGraph translator.

Definition at line 36 of file AxfcFrontendCompiler.py.

#### 8.4.4.3 \_\_md

```
AxfcFrontendCompiler.AxfcFrontendCompiler.__md [private]
```

machine description object

Definition at line 34 of file AxfcFrontendCompiler.py.

The documentation for this class was generated from the following file:

- /home/youngsun/Project/SKT-AIX/Development/aixc/src/[AxfcFrontendCompiler.py](#)

## 8.5 AxfcGraphWriter.AxfcGraphWriter Class Reference

[AxfcGraphWriter](#) class.

### Public Member Functions

- `def __init__ (self)`  
*The constructor.*
- `def add_edge (self, source_node_id, target_node_id)`  
*This method is used insert the edges of node.*
- `def add_node`  
*This method is used insert the node.*
- `def write_file`  
*This method is used to write the edges and nodes to Sigma js json format.*

### Private Attributes

- `__edge_id`  
*Edge's ID (auto increase)*
- `__graph`  
*dictionary of edges and nodes*
- `__nodes`  
*set of nodes*
- `__x_axis`  
*x axis of edges*
- `__y_axis`  
*y axis of edges*

### 8.5.1 Detailed Description

[AxfcGraphWriter](#) class.

Definition at line 23 of file `AxfcGraphWriter.py`.

### 8.5.2 Constructor & Destructor Documentation

#### 8.5.2.1 \_\_init\_\_()

```
def AxfcGraphWriter.AxfcGraphWriter.__init__ (
    self )
```

The constructor.

Definition at line 41 of file `AxfcGraphWriter.py`.

### 8.5.3 Member Function Documentation

#### 8.5.3.1 add\_edge()

```
def AxfcGraphWriter.AxfcGraphWriter.add_edge (
    self,
    source_node_id,
    target_node_id )
```

This method is used insert the edges of node.

##### Parameters

<i>self</i>	this object
<i>source_node_id</i>	node's id for source
<i>target_node_id</i>	node's id for target

Definition at line 53 of file AxfcGraphWriter.py.

#### 8.5.3.2 add\_node()

```
def AxfcGraphWriter.AxfcGraphWriter.add_node (
    self,
    ir_node )
```

This method is used insert the node.

##### Parameters

<i>self</i>	this object
<i>ir_node</i>	<a href="#">AxfcIRNode</a> node

Definition at line 68 of file AxfcGraphWriter.py.

#### 8.5.3.3 write\_file()

```
def AxfcGraphWriter.AxfcGraphWriter.write_file (
    self,
    file_path )
```

This method is used to write the edges and nodes to Sigma js json format.

**Parameters**

<i>self</i>	this object
<i>file_path</i>	file path for dumping the IR graph

**Returns**

error info

Definition at line 95 of file AxfcGraphWriter.py.

## 8.5.4 Member Data Documentation

### 8.5.4.1 `__edge_id`

`AxfcGraphWriter.AxfcGraphWriter.__edge_id` [private]

Edge's ID (auto increase)

Definition at line 42 of file AxfcGraphWriter.py.

### 8.5.4.2 `__graph`

`AxfcGraphWriter.AxfcGraphWriter.__graph` [private]

dictionary of edges and nodes

Definition at line 43 of file AxfcGraphWriter.py.

### 8.5.4.3 `__nodes`

`AxfcGraphWriter.AxfcGraphWriter.__nodes` [private]

set of nodes

Definition at line 44 of file AxfcGraphWriter.py.

#### 8.5.4.4 `__x_axis`

`AxfcGraphWriter.AxfcGraphWriter.__x_axis` [private]

x axis of edges

Definition at line 45 of file `AxfcGraphWriter.py`.

#### 8.5.4.5 `__y_axis`

`AxfcGraphWriter.AxfcGraphWriter.__y_axis` [private]

y axis of edges

Definition at line 46 of file `AxfcGraphWriter.py`.

The documentation for this class was generated from the following file:

- `/home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcGraphWriter.py`

## 8.6 AxfclRBlock.AxfclRBlock Class Reference

[AxfclRBlock](#) class.

### Public Member Functions

- `def __init__ (self)`  
*The constructor.*
- `def analyse_liveness (self)`  
*This method is used to perform the local liveness analysis in the scope of an IR block.*
- `def analyze_profit (self)`  
*This method is used to calculate the profit that we can achieve by accelerating this block in hardware-manner.*
- `def __str__ (self)`  
*For debugging.*

### Public Attributes

- `id`  
*block ID*
- `nodes`  
*a list of nodes that make up this block*
- `live_in`  
*a list of live-in node IDs*
- `live_out`  
*a list of live-out node IDs*
- `is_aixh_support`  
*indicate whether this node can be executed in hardware-manner*
- `aixh_profit`  
*specify the profit to be obtained by using AIXH*
- `aix_graph`  
*an AIX graph emitted from this IR block*

## Private Member Functions

- def `__analyse_inout` (self)

*This method is used to find the input and output nodes of this block.*

### 8.6.1 Detailed Description

`AxfcIRBlock` class.

Definition at line 20 of file `AxfcIRBlock.py`.

### 8.6.2 Constructor & Destructor Documentation

#### 8.6.2.1 `__init__()`

```
def AxfcIRBlock.AxfcIRBlock.__init__ (
    self )
```

The constructor.

Definition at line 44 of file `AxfcIRBlock.py`.

### 8.6.3 Member Function Documentation

#### 8.6.3.1 `__analyse_inout()`

```
def AxfcIRBlock.AxfcIRBlock.__analyse_inout (
    self,
    AxfcError ) [private]
```

This method is used to find the input and output nodes of this block.

#### Parameters

<code>self</code>	this object
-------------------	-------------

#### Returns

error info.

Definition at line 96 of file `AxfcIRBlock.py`.



### 8.6.3.2 \_\_str\_\_()

```
def AxfcIRBlock.AxfcIRBlock.__str__ (
    self )
```

For debugging.

Definition at line 162 of file AxfcIRBlock.py.

### 8.6.3.3 analyse\_liveness()

```
def AxfcIRBlock.AxfcIRBlock.analyse_liveness (
    self,
    AxfcError )
```

This method is used to perform the local liveness analysis in the scope of an IR block.

We employ a simple heuristic scheme to find live-ins and live-outs of a block without global liveness analysis on the entire IR graph.

#### Parameters

<i>self</i>	this object
-------------	-------------

#### Returns

error info

Definition at line 59 of file AxfcIRBlock.py.

### 8.6.3.4 analyze\_profit()

```
def AxfcIRBlock.AxfcIRBlock.analyze_profit (
    self,
    AxfcError )
```

This method is used to calculate the profit that we can achieve by accelerating this block in hardware-manner.

#### Parameters

<i>self</i>	this object
-------------	-------------

#### Returns

error info

Definition at line 140 of file AxfcIRBlock.py.

## 8.6.4 Member Data Documentation

### 8.6.4.1 aix\_graph

`AxfcIRBlock.AxfcIRBlock.aix_graph`

an AIX graph emitted from this IR block

Definition at line 51 of file `AxfcIRBlock.py`.

### 8.6.4.2 aixh\_profit

`AxfcIRBlock.AxfcIRBlock.aixh_profit`

specify the profit to be obtained by using AIXH

Definition at line 50 of file `AxfcIRBlock.py`.

### 8.6.4.3 id

`AxfcIRBlock.AxfcIRBlock.id`

block ID

Definition at line 45 of file `AxfcIRBlock.py`.

### 8.6.4.4 is\_aixh\_support

`AxfcIRBlock.AxfcIRBlock.is_aixh_support`

indicate whether this node can be executed in hardware-manner

Definition at line 49 of file `AxfcIRBlock.py`.

### 8.6.4.5 live\_in

`AxfcIRBlock.AxfcIRBlock.live_in`

a list of live-in node IDs

Definition at line 47 of file `AxfcIRBlock.py`.

#### 8.6.4.6 live\_out

`AxfcIRBlock.AxfcIRBlock.live_out`

a list of live-out node IDs

Definition at line 48 of file `AxfcIRBlock.py`.

#### 8.6.4.7 nodes

`AxfcIRBlock.AxfcIRBlock.nodes`

a list of nodes that make up this block

Definition at line 46 of file `AxfcIRBlock.py`.

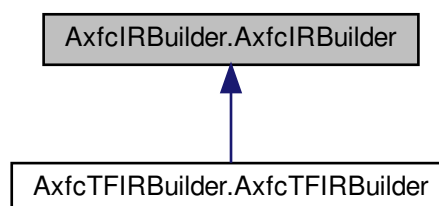
The documentation for this class was generated from the following file:

- `/home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcIRBlock.py`

## 8.7 AxfcIRBuilder.AxfcIRBuilder Class Reference

[AxfcIRBuilder](#) class.

Inheritance diagram for `AxfcIRBuilder.AxfcIRBuilder`:



### Public Member Functions

- `def \_\_init\_\_ (self, md)`  
*The constructor.*
- `def build\_ir`  
*This method is used to build AXI IR.*
- `def \_\_str\_\_ (self)`  
*For debugging.*

## Private Member Functions

- `def __find_aixh_blocks (self)`  
*This method is used to find AIXH blocks comprised of AIXH-supported nodes.*
- `def __perform_maximal_munch`  
*This method performs maximal munch algorithm to recursively find the longest successive AIXH-supported nodes.*
- `def __read_model_graph`  
*Abstract methods.*
- `def __build_naive_ir`  
*This method is used to construct a naive AIXIR using a tensorflow graph.*

## Private Attributes

- `_md`  
*AIX machine description.*
- `_tf_graph`  
*input Tensorflow graph*
- `_ir_graph`  
*output AIXIR graph*
- `_ir_symtab`  
*symbol table for IR graph*

### 8.7.1 Detailed Description

`AxfcIRBuilder` class.

Definition at line 21 of file `AxfcIRBuilder.py`.

### 8.7.2 Constructor & Destructor Documentation

#### 8.7.2.1 `__init__()`

```
def AxfcIRBuilder.AxfcIRBuilder.__init__ (
    self,
    md )
```

The constructor.

Definition at line 36 of file `AxfcIRBuilder.py`.

### 8.7.3 Member Function Documentation

#### 8.7.3.1 `__find_aixh_blocks()`

```
def AxfcIRBuilder.AxfcIRBuilder.__find_aixh_blocks (
    self,
    AxfcError ) [private]
```

This method is used to find AIXH blocks comprised of AIXH-supported nodes.

We employ a maximal munching scheme to find the longest successive AIXH-supported nodes and build up a block with the nodes.

## Parameters

<i>self</i>	this object
-------------	-------------

## Returns

error info

Definition at line 95 of file AxfclRBuilder.py.

## 8.7.3.2 \_\_perform\_maximal\_munch()

```
def AxfclRBuilder.AxfclRBuilder.__perform_maximal_munch (
    self,
    ir_node ) [private]
```

This method performs maximal munch algorithm to recursively find the longest successive AIXH-supported nodes.

## Parameters

<i>self</i>	this object
<i>ir_node</i>	a start node to perform maximal munching
<i>an</i>	IR block of the successive IR nodes supported by the AIX hardware

## Returns

error info

Definition at line 141 of file AxfclRBuilder.py.

## 8.7.3.3 \_\_str\_\_()

```
def AxfclRBuilder.AxfclRBuilder.__str__ (
    self )
```

For debugging.

Definition at line 172 of file AxfclRBuilder.py.

## 8.7.3.4 \_build\_naive\_ir()

```
def AxfclRBuilder.AxfclRBuilder._build_naive_ir (
    self,
    path ) [private]
```

This method is used to construct a naive AIXIR using a tensorflow graph.

**Parameters**

<i>self</i>	this object
<i>path</i>	file path of input network model

**Returns**

error info

Definition at line 192 of file AxfcIRBuilder.py.

**8.7.3.5 \_read\_model\_graph()**

```
def AxfcIRBuilder.AxfcIRBuilder._read_model_graph (
    self,
    path ) [private]
```

Abstract methods.

This method is used to read a tensorflow graph from an input file in the given path.

**Parameters**

<i>self</i>	this object
<i>path</i>	file path of input network model

**Returns**

error info

Definition at line 184 of file AxfcIRBuilder.py.

**8.7.3.6 build\_ir()**

```
def AxfcIRBuilder.AxfcIRBuilder.build_ir (
    self,
    path )
```

This method is used to build AXI IR.

1) it builds a naive IR using the given input model. 2) it checks the IR nodes to be executed in hardware-manner. 3) it finds AIXH IR blocks. each block consist of several AIXH IR nodes. 4) it performs the liveness analysis for resolving the input and output of the blocks.

**Parameters**

<i>self</i>	this object
<i>path</i>	input path of a frozen model

**Returns**

error info and an [AxfcIRGraph](#) object

Definition at line 51 of file AxfcIRBuilder.py.

**8.7.4 Member Data Documentation****8.7.4.1 `_ir_graph`**

`AxfcIRBuilder.AxfcIRBuilder._ir_graph` [private]

output AIXIR graph

Definition at line 39 of file AxfcIRBuilder.py.

**8.7.4.2 `_ir_symtab`**

`AxfcIRBuilder.AxfcIRBuilder._ir_symtab` [private]

symbol table for IR graph

Definition at line 40 of file AxfcIRBuilder.py.

**8.7.4.3 `_md`**

`AxfcIRBuilder.AxfcIRBuilder._md` [private]

AIX machine description.

Definition at line 37 of file AxfcIRBuilder.py.

#### 8.7.4.4 `_tf_graph`

`AxflIRBuilder.AxflIRBuilder._tf_graph` [private]

input Tensorflow graph

Definition at line 38 of file `AxflIRBuilder.py`.

The documentation for this class was generated from the following file:

- `/home/youngsun/Project/SKT-AIX/Development/aixc/src/AxflIRBuilder.py`

## 8.8 `AxflIRGraph`.`AxflIRGraph` Class Reference

`AxflIRGraph` class.

### Public Member Functions

- `def __init__`  
*The constructor.*
- `def append_node`  
*This method is used to append the given IR node into the graph.*
- `def append_block`  
*This method is used to append the given IR block into the graph.*
- `def analyse_liveness` (self)  
*This method is used to perform the liveness analysis of this graph.*
- `def dump_to_file`  
*This method is used to visualize the IR graph using Sigma.js.*
- `def __str__` (self)  
*For debugging.*

### Public Attributes

- `root_node`  
*output root node of this graph*
- `nodes`  
*a list of nodes consisting this graph*
- `blocks`  
*a list of blocks that are contained this graph*
- `symtab`  
*a reference to an IR symbol table*

#### 8.8.1 Detailed Description

`AxflIRGraph` class.

Definition at line 21 of file `AxflIRGraph.py`.



## 8.8.2 Constructor & Destructor Documentation

### 8.8.2.1 \_\_init\_\_()

```
def AxfcIRGraph.AxfcIRGraph.__init__ (
    self,
    symtab )
```

The constructor.

#### Parameters

<i>self</i>	this object
<i>symtab</i>	a symbol table for referring to an IR node using its name

Definition at line 38 of file AxfcIRGraph.py.

## 8.8.3 Member Function Documentation

### 8.8.3.1 \_\_str\_\_()

```
def AxfcIRGraph.AxfcIRGraph.__str__ (
    self )
```

For debugging.

Definition at line 120 of file AxfcIRGraph.py.

### 8.8.3.2 analyse\_liveness()

```
def AxfcIRGraph.AxfcIRGraph.analyse_liveness (
    self,
    AxfcError )
```

This method is used to perform the liveness analysis of this graph.

#### Parameters

<i>self</i>	this object
-------------	-------------

**Returns**

error info

Definition at line 75 of file AxfcIRGraph.py.

**8.8.3.3 append\_block()**

```
def AxfcIRGraph.AxfcIRGraph.append_block (
    self,
    ir_block )
```

This method is used to append the given IR block into the graph.

**Parameters**

<i>self</i>	this object
<i>ir_block</i>	an IR block to be appended

Definition at line 65 of file AxfcIRGraph.py.

**8.8.3.4 append\_node()**

```
def AxfcIRGraph.AxfcIRGraph.append_node (
    self,
    ir_node )
```

This method is used to append the given IR node into the graph.

**Parameters**

<i>self</i>	this object
<i>ir_node</i>	an IR node to be appended

Definition at line 50 of file AxfcIRGraph.py.

**8.8.3.5 dump\_to\_file()**

```
def AxfcIRGraph.AxfcIRGraph.dump_to_file (
    self,
    file_path )
```

This method is used to visualize the IR graph using Sigma js.

**Parameters**

<i>self</i>	this object
<i>file_path</i>	a file path to dump out the IR graph
<i>ignore_ops</i>	a list of operations to be ignored

**Returns**

error info

Definition at line 90 of file AxfcIRGraph.py.

## 8.8.4 Member Data Documentation

### 8.8.4.1 blocks

`AxfcIRGraph.AxfcIRGraph.blocks`

a list of blocks that are contained this graph

Definition at line 43 of file AxfcIRGraph.py.

### 8.8.4.2 nodes

`AxfcIRGraph.AxfcIRGraph.nodes`

a list of nodes consisting this graph

Definition at line 42 of file AxfcIRGraph.py.

### 8.8.4.3 root\_node

`AxfcIRGraph.AxfcIRGraph.root_node`

output root node of this graph

Definition at line 39 of file AxfcIRGraph.py.

#### 8.8.4.4 symtab

`AxflIRGraph.AxflIRGraph.symtab`

a reference to an IR symbol table

Definition at line 44 of file `AxflIRGraph.py`.

The documentation for this class was generated from the following file:

- `/home/youngsun/Project/SKT-AIX/Development/aixc/src/AxflIRGraph.py`

## 8.9 AxflIRNode.AxflIRNode Class Reference

`AxflIRNode`.

### Public Member Functions

- `def __init__(self, node_def)`  
*The constructor.*
- `def analyze_profit(self)`  
*This method is used to calculate and return the profit that we can get by accelerating the operation of this node in hardware-manner.*
- `def __eq__(self, other)`  
*This methods is used to compare id with equal (==) for using Set.*
- `def __hash__(self)`  
*This methods make this object become hasable by id.*
- `def __del__(self)`  
*Destructor.*
- `def __str__(self)`  
*For debugging.*

### Public Attributes

- `id`  
*node ID*
- `name`  
*node name*
- `layer_id`  
*node layer ID*
- `succs`  
*node a list of successor nodes*
- `preds`  
*node a list of predecessor nodes*
- `node_def`  
*node a reference to an input node object*
- `block_ref`  
*reference to the IR block that contains this node*

- [aixh\\_profit](#)  
*specify the profit to be obtained by using AIXH*
- [is\\_aixh\\_support](#)  
*indicate whether this node can be executed in hardware-manner*
- [eval\\_flag](#)  
*indicate whether this node has been already evaluated or not for maximal munching*
- [is\\_input](#)  
*indicate whether this node is an input node or not*
- [is\\_output](#)  
*indicate whether this node is an output node or not*
- [aix\\_layer](#)  
*reference to the AIX layer derived from this node*
- [op](#)  
*operation of this node*

### 8.9.1 Detailed Description

[AxfcIRNode](#).

Created: 2020. 08. 03

Authors: Youngsun Han ([youngsun@pknu.ac.kr](mailto:youngsun@pknu.ac.kr)) Heng Sengthai ([sengthai37@gmail.com](mailto:sengthai37@gmail.com))

High Performance Computing Laboratory ([hpcl.pknu.ac.kr](http://hpcl.pknu.ac.kr)) [AxfcIRNode](#) class

Definition at line 17 of file [AxfcIRNode.py](#).

### 8.9.2 Constructor & Destructor Documentation

#### 8.9.2.1 `__init__()`

```
def AxfcIRNode.AxfcIRNode.__init__ (
    self,
    node_def )
```

The constructor.

Definition at line 62 of file [AxfcIRNode.py](#).

#### 8.9.2.2 `__del__()`

```
def AxfcIRNode.AxfcIRNode.__del__ (
    self )
```

Destructor.

Definition at line 115 of file [AxfcIRNode.py](#).

### 8.9.3 Member Function Documentation

#### 8.9.3.1 `__eq__()`

```
def AxfcIRNode.AxfcIRNode.__eq__ (
    self,
    other )
```

This methods is used to compare id with equal (==) for using Set.

##### Parameters

<i>self</i>	this object
<i>other</i>	another <a href="#">AxfcIRNode</a> object

Definition at line 100 of file AxfcIRNode.py.

#### 8.9.3.2 `__hash__()`

```
def AxfcIRNode.AxfcIRNode.__hash__ (
    self )
```

This methods make this object become hasable by id.

##### Parameters

<i>self</i>	this object
-------------	-------------

Definition at line 111 of file AxfcIRNode.py.

#### 8.9.3.3 `__str__()`

```
def AxfcIRNode.AxfcIRNode.__str__ (
    self )
```

For debugging.

Definition at line 124 of file AxfcIRNode.py.

#### 8.9.3.4 analyze\_profit()

```
def AxfclRNode.AxfclRNode.analyze_profit (
    self,
    int )
```

This method is used to calculate and return the profit that we can get by accelerating the operation of this node in hardware-manner.

##### Parameters

<i>self</i>	this object
-------------	-------------

##### Returns

the calculated profit

Definition at line 85 of file AxfclRNode.py.

### 8.9.4 Member Data Documentation

#### 8.9.4.1 aix\_layer

```
AxfclRNode.AxfclRNode.aix_layer
```

reference to the AIX layer derived from this node

Definition at line 78 of file AxfclRNode.py.

#### 8.9.4.2 aixh\_profit

```
AxfclRNode.AxfclRNode.aixh_profit
```

specify the profit to be obtained by using AIXH

Definition at line 71 of file AxfclRNode.py.

#### 8.9.4.3 block\_ref

```
AxfclRNode.AxfclRNode.block_ref
```

reference to the IR block that contains this node

Definition at line 69 of file AxfclRNode.py.

#### 8.9.4.4 eval\_flag

`AxfcIRNode.AxfcIRNode.eval_flag`

indicate whether this node has been already evaluated or not for maximal munching

Definition at line 73 of file `AxfcIRNode.py`.

#### 8.9.4.5 id

`AxfcIRNode.AxfcIRNode.id`

node ID

Definition at line 63 of file `AxfcIRNode.py`.

#### 8.9.4.6 is\_aixh\_support

`AxfcIRNode.AxfcIRNode.is_aixh_support`

indicate whether this node can be executed in hardware-manner

Definition at line 72 of file `AxfcIRNode.py`.

#### 8.9.4.7 is\_input

`AxfcIRNode.AxfcIRNode.is_input`

indicate whether this node is an input node or not

Definition at line 75 of file `AxfcIRNode.py`.

#### 8.9.4.8 is\_output

`AxfcIRNode.AxfcIRNode.is_output`

indicate whether this node is an output node or not

Definition at line 76 of file `AxfcIRNode.py`.



#### 8.9.4.9 layer\_id

`AxfcIRNode.AxfcIRNode.layer_id`

node layer ID

Definition at line 65 of file AxfcIRNode.py.

#### 8.9.4.10 name

`AxfcIRNode.AxfcIRNode.name`

node name

Definition at line 64 of file AxfcIRNode.py.

#### 8.9.4.11 node\_def

`AxfcIRNode.AxfcIRNode.node_def`

node a reference to an input node object

Definition at line 68 of file AxfcIRNode.py.

#### 8.9.4.12 op

`AxfcIRNode.AxfcIRNode.op`

operation of this node

Definition at line 117 of file AxfcIRNode.py.

#### 8.9.4.13 preds

`AxfcIRNode.AxfcIRNode.preds`

node a list of predecessor nodes

Definition at line 67 of file AxfcIRNode.py.

#### 8.9.4.14 succs

`AxfcIRNode.AxfcIRNode.succs`

node a list of successor nodes

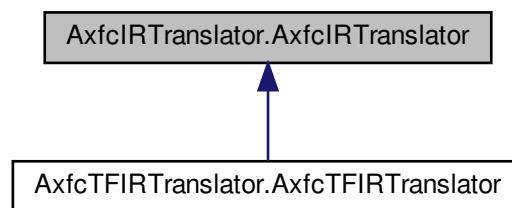
Definition at line 66 of file `AxfcIRNode.py`.

The documentation for this class was generated from the following file:

- `/home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcIRNode.py`

## 8.10 AxfcIRTranslator.AxfcIRTranslator Class Reference

Inheritance diagram for `AxfcIRTranslator.AxfcIRTranslator`:



### Public Member Functions

- `def __init__(self, md)`  
*The constructor.*
- `def emit_aixh_graphs`  
*This method translates IR blocks of the given IR graph into AIXGraphs and return them.*
- `def __str__(self)`  
*For debugging.*

### Public Attributes

- `aix_graphs`  
*a list of AIXGraphs translated from an input model*

## Private Member Functions

- def [\\_\\_emit\\_aixh\\_block](#)  
*This method is used to translate an IR block into an AIXGraph.*
- def [\\_\\_emit\\_aixh\\_node](#)  
*This method is used to translate an IR node into an AIXLayer object.*
- def [\\_\\_get\\_emitted\\_input\\_nodes](#)  
*This method is used to return a list of already emitted input nodes.*
- def [\\_\\_emit\\_aix\\_layer\\_convolution](#)  
*Abstract methods.*
- def [\\_\\_emit\\_aix\\_layer\\_group\\_conv](#)
- def [\\_\\_emit\\_aix\\_layer\\_batchnorm](#)
- def [\\_\\_emit\\_aix\\_layer\\_maxpool](#)
- def [\\_\\_emit\\_aix\\_layer\\_ewadd](#)
- def [\\_\\_emit\\_aix\\_layer\\_avgpool](#)
- def [\\_\\_emit\\_aix\\_layer\\_biasadd](#)
- def [\\_\\_emit\\_aix\\_layer\\_softmax](#)
- def [\\_\\_emit\\_aix\\_layer\\_activation](#)
- def [\\_\\_emit\\_aix\\_tensor\\_input](#)  
*emission methods for AIX tensors*
- def [\\_\\_emit\\_aix\\_tensor\\_filter](#)
- def [\\_\\_emit\\_aix\\_tensor\\_bias](#)
- def [\\_\\_emit\\_aix\\_tensor\\_scale](#)
- def [\\_\\_emit\\_aix\\_tensor\\_mean](#)
- def [\\_\\_emit\\_aix\\_tensor\\_variance](#)
- def [\\_\\_emit\\_aix\\_tensor\\_output](#)
- def [\\_\\_emit\\_aix\\_convolution\\_desc](#)  
*emission methods for AIX convolution dec*
- def [\\_\\_emit\\_aix\\_sampling\\_desc](#)  
*emission methods for AIX sampling dec*

## Private Attributes

- [\\_md](#)  
*AIX machine description.*
- [\\_ir\\_syntab](#)  
*a symbol table of pairs of an IR node's name and itself*
- [\\_\\_emit\\_aix\\_layer\\_tbl](#)  
*a dictionary of pairs of AIXLayerType and its AIX layer emission method*
- [\\_aix\\_graph](#)  
*the current AIX graph being translated*

### 8.10.1 Detailed Description

Definition at line 37 of file AxfclIRTranslator.py.

### 8.10.2 Constructor & Destructor Documentation

### 8.10.2.1 `__init__()`

```
def AxfcIRTranslator.AxfcIRTranslator.__init__ (
    self,
    md )
```

The constructor.

Definition at line 55 of file AxfcIRTranslator.py.

## 8.10.3 Member Function Documentation

### 8.10.3.1 `__emit_aixh_block()`

```
def AxfcIRTranslator.AxfcIRTranslator.__emit_aixh_block (
    self,
    ir_block ) [private]
```

This method is used to translate an IR block into an AIXGraph.

#### Parameters

<i>self</i>	this object
<i>ir_block</i>	input IR block

#### Returns

error info and an output AIXGraph

Definition at line 107 of file AxfcIRTranslator.py.

### 8.10.3.2 `__emit_aixh_node()`

```
def AxfcIRTranslator.AxfcIRTranslator.__emit_aixh_node (
    self,
    ir_node ) [private]
```

This method is used to translate an IR node into an AIXLayer object.

#### Parameters

<i>self</i>	this object
<i>ir_node</i>	input IR node to be translated

**Returns**

error info and an output AIXLayer object

Definition at line 142 of file AxfclIRTranslator.py.

**8.10.3.3 \_\_str\_\_()**

```
def AxfclIRTranslator.AxfclIRTranslator.__str__ (
    self )
```

For debugging.

Definition at line 234 of file AxfclIRTranslator.py.

**8.10.3.4 \_emit\_aix\_convolution\_desc()**

```
def AxfclIRTranslator.AxfclIRTranslator._emit_aix_convolution_desc (
    self,
    ir_node ) [private]
```

emission methods for AIX convolution dec

Definition at line 292 of file AxfclIRTranslator.py.

**8.10.3.5 \_emit\_aix\_layer\_activation()**

```
def AxfclIRTranslator.AxfclIRTranslator._emit_aix_layer_activation (
    self,
    ir_node ) [private]
```

Definition at line 266 of file AxfclIRTranslator.py.

**8.10.3.6 \_emit\_aix\_layer\_avgpool()**

```
def AxfclIRTranslator.AxfclIRTranslator._emit_aix_layer_avgpool (
    self,
    ir_node ) [private]
```

Definition at line 257 of file AxfclIRTranslator.py.

#### 8.10.3.7 `_emit_aix_layer_batchnorm()`

```
def AxfcIRTranslator.AxfcIRTranslator._emit_aix_layer_batchnorm (
    self,
    ir_node ) [private]
```

Definition at line 248 of file AxfcIRTranslator.py.

#### 8.10.3.8 `_emit_aix_layer_biasadd()`

```
def AxfcIRTranslator.AxfcIRTranslator._emit_aix_layer_biasadd (
    self,
    ir_node ) [private]
```

Definition at line 260 of file AxfcIRTranslator.py.

#### 8.10.3.9 `_emit_aix_layer_convolution()`

```
def AxfcIRTranslator.AxfcIRTranslator._emit_aix_layer_convolution (
    self,
    ir_node ) [private]
```

Abstract methods.

emission methods for AIX layers

Definition at line 242 of file AxfcIRTranslator.py.

#### 8.10.3.10 `_emit_aix_layer_ewadd()`

```
def AxfcIRTranslator.AxfcIRTranslator._emit_aix_layer_ewadd (
    self,
    ir_node ) [private]
```

Definition at line 254 of file AxfcIRTranslator.py.

#### 8.10.3.11 `_emit_aix_layer_group_conv()`

```
def AxfcIRTranslator.AxfcIRTranslator._emit_aix_layer_group_conv (
    self,
    ir_node ) [private]
```

Definition at line 245 of file AxfcIRTranslator.py.

**8.10.3.12 \_emit\_aix\_layer\_maxpool()**

```
def AxfcIRTranslator.AxfcIRTranslator._emit_aix_layer_maxpool (
    self,
    ir_node ) [private]
```

Definition at line 251 of file AxfcIRTranslator.py.

**8.10.3.13 \_emit\_aix\_layer\_softmax()**

```
def AxfcIRTranslator.AxfcIRTranslator._emit_aix_layer_softmax (
    self,
    ir_node ) [private]
```

Definition at line 263 of file AxfcIRTranslator.py.

**8.10.3.14 \_emit\_aix\_sampling\_desc()**

```
def AxfcIRTranslator.AxfcIRTranslator._emit_aix_sampling_desc (
    self,
    ir_node ) [private]
```

emission methods for AIX sampling dec

Definition at line 296 of file AxfcIRTranslator.py.

**8.10.3.15 \_emit\_aix\_tensor\_bias()**

```
def AxfcIRTranslator.AxfcIRTranslator._emit_aix_tensor_bias (
    self,
    ir_node ) [private]
```

Definition at line 276 of file AxfcIRTranslator.py.

**8.10.3.16 \_emit\_aix\_tensor\_filter()**

```
def AxfcIRTranslator.AxfcIRTranslator._emit_aix_tensor_filter (
    self,
    ir_node ) [private]
```

Definition at line 273 of file AxfcIRTranslator.py.

#### 8.10.3.17 \_emit\_aix\_tensor\_input()

```
def AxfcIRTranslator.AxfcIRTranslator._emit_aix_tensor_input (
    self,
    ir_node ) [private]
```

emission methods for AIX tensors

Definition at line 270 of file AxfcIRTranslator.py.

#### 8.10.3.18 \_emit\_aix\_tensor\_mean()

```
def AxfcIRTranslator.AxfcIRTranslator._emit_aix_tensor_mean (
    self,
    ir_node ) [private]
```

Definition at line 282 of file AxfcIRTranslator.py.

#### 8.10.3.19 \_emit\_aix\_tensor\_output()

```
def AxfcIRTranslator.AxfcIRTranslator._emit_aix_tensor_output (
    self,
    ir_node ) [private]
```

Definition at line 288 of file AxfcIRTranslator.py.

#### 8.10.3.20 \_emit\_aix\_tensor\_scale()

```
def AxfcIRTranslator.AxfcIRTranslator._emit_aix_tensor_scale (
    self,
    ir_node ) [private]
```

Definition at line 279 of file AxfcIRTranslator.py.

#### 8.10.3.21 \_emit\_aix\_tensor\_variance()

```
def AxfcIRTranslator.AxfcIRTranslator._emit_aix_tensor_variance (
    self,
    ir_node ) [private]
```

Definition at line 285 of file AxfcIRTranslator.py.

#### 8.10.3.22 \_get\_emitted\_input\_nodes()

```
def AxfcIRTranslator.AxfcIRTranslator._get_emitted_input_nodes (
    self,
    ir_node ) [private]
```

This method is used to return a list of already emitted input nodes.

If there are input nodes that have not translated yet, we perform \_\_emit\_aixh\_node method repeatedly to emit them all.



**Parameters**

<i>self</i>	this object
<i>ir_node</i>	current node to emit its input nodes

**Returns**

a list of emitted input nodes

Definition at line 206 of file AxfclIRTranslator.py.

**8.10.3.23 emit\_aixh\_graphs()**

```
def AxfclIRTranslator.AxfclIRTranslator.emit_aixh_graphs (
    self,
    ir_graph )
```

This method translates IR blocks of the given IR graph into AIXGraphs and return them.

**Parameters**

<i>self</i>	this object
<i>ir_graph</i>	input IR graph

**Returns**

error info and a list of AIXGraphs

Definition at line 78 of file AxfclIRTranslator.py.

**8.10.4 Member Data Documentation****8.10.4.1 \_\_emit\_aix\_layer\_tbl**

```
AxfclIRTranslator.AxfclIRTranslator.__emit_aix_layer_tbl [private]
```

a dictionary of pairs of AIXLayerType and its AIX layer emission method

Definition at line 60 of file AxfclIRTranslator.py.

#### 8.10.4.2 `_aix_graph`

`AxfcIRTranslator.AxfcIRTranslator._aix_graph` [private]

the current AIX graph being translated

Definition at line 111 of file `AxfcIRTranslator.py`.

#### 8.10.4.3 `_ir_symtab`

`AxfcIRTranslator.AxfcIRTranslator._ir_symtab` [private]

a symbol table of pairs of an IR node's name and itself

Definition at line 58 of file `AxfcIRTranslator.py`.

#### 8.10.4.4 `_md`

`AxfcIRTranslator.AxfcIRTranslator._md` [private]

AIX machine description.

Definition at line 56 of file `AxfcIRTranslator.py`.

#### 8.10.4.5 `aix_graphs`

`AxfcIRTranslator.AxfcIRTranslator.aix_graphs`

a list of AIXGraphs translated from an input model

Definition at line 57 of file `AxfcIRTranslator.py`.

The documentation for this class was generated from the following file:

- `/home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcIRTranslator.py`

## 8.11 `AxfcLauncherWriter.AxfcLauncherWriter` Class Reference

[AxfcLauncherWriter](#) class.

## Public Member Functions

- `def __init__`  
*The constructor.*
- `def emit_aixh_launcher` (self)  
*This method is used to emit a launcher for the generated AIXGraph.*
- `def __str__` (self)  
*For debugging.*

## Private Attributes

- `__md`  
*AIX machine description.*
- `__ir_graph`  
*an AIXIR graph that will be used for writing the launcher*

### 8.11.1 Detailed Description

`AxfcLauncherWriter` class.

Definition at line 20 of file `AxfcLauncherWriter.py`.

### 8.11.2 Constructor & Destructor Documentation

#### 8.11.2.1 `__init__()`

```
def AxfcLauncherWriter.AxfcLauncherWriter.__init__ (
    self,
    md )
```

The constructor.

Definition at line 29 of file `AxfcLauncherWriter.py`.

### 8.11.3 Member Function Documentation

#### 8.11.3.1 `__str__()`

```
def AxfcLauncherWriter.AxfcLauncherWriter.__str__ (
    self )
```

For debugging.

Definition at line 40 of file `AxfcLauncherWriter.py`.

#### 8.11.3.2 `emit_aixh_launcher()`

```
def AxfcLauncherWriter.AxfcLauncherWriter.emit_aixh_launcher (
    self )
```

This method is used to emit a launcher for the generated AIXGraph.

## Parameters

<i>self</i>	this object
-------------	-------------

Definition at line 35 of file AxfcLauncherWriter.py.

## 8.11.4 Member Data Documentation

### 8.11.4.1 `__ir_graph`

`AxfcLauncherWriter.AxfcLauncherWriter.__ir_graph` [private]

an AIXIR graph that will be used for writing the launcher

Definition at line 31 of file AxfcLauncherWriter.py.

### 8.11.4.2 `__md`

`AxfcLauncherWriter.AxfcLauncherWriter.__md` [private]

AIX machine description.

Definition at line 30 of file AxfcLauncherWriter.py.

The documentation for this class was generated from the following file:

- [/home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcLauncherWriter.py](#)

## 8.12 AxfcMachineDesc.AxfcMachineDesc Class Reference

[AxfcMachineDesc](#) class.

### Classes

- class [AIXLayerInfo](#)  
*AIXLayerInfo* inner class.

## Public Member Functions

- def `__init__` (self)  
*The constructor.*
- def `read_file`  
*This method is used to read a machine description from the given input path.*
- def `get_layer_info`  
*This method returns the information of a specific AIX layer.*
- def `get_aixh_support`  
*This method indicates whether the given operation is supported by the AIX hardware or not.*
- def `get_in_type` (self)  
*This method returns the type of AI framework.*
- def `get_profit_threshold` (self)  
*This method returns the input type of the frontend compilation.*
- def `__str__` (self)  
*For debugging.*

## Static Public Attributes

- int `TYPE_TENSORFLOW` = 0
- int `TYPE_PYTORCH` = 1
- int `TYPE_MXNET` = 2
- int `TYPE_UNKNOWN` = 3
- int `DEFAULT_PROFIT_THRESHOLD` = 1000

## Private Attributes

- `__aix_model_info_tbl`  
*general info.*
- `__aix_layer_info_tbl`  
*general machine description info.*

### 8.12.1 Detailed Description

`AxfcMachineDesc` class.

Definition at line 23 of file `AxfcMachineDesc.py`.

### 8.12.2 Constructor & Destructor Documentation

#### 8.12.2.1 `__init__`()

```
def AxfcMachineDesc.AxfcMachineDesc.__init__ (
    self )
```

The constructor.

Definition at line 40 of file `AxfcMachineDesc.py`.

### 8.12.3 Member Function Documentation

#### 8.12.3.1 `__str__()`

```
def AxfcMachineDesc.AxfcMachineDesc.__str__ (
    self )
```

For debugging.

Definition at line 138 of file AxfcMachineDesc.py.

#### 8.12.3.2 `get_aixh_support()`

```
def AxfcMachineDesc.AxfcMachineDesc.get_aixh_support (
    self,
    layer_type )
```

This method indicates whether the given operation is supported by the AIX hardware or not.

##### Parameters

<i>self</i>	this object
<i>layer_type</i>	the name of an AIX layer type to be checked

##### Returns

the input type of the frontend compilation

Definition at line 100 of file AxfcMachineDesc.py.

#### 8.12.3.3 `get_in_type()`

```
def AxfcMachineDesc.AxfcMachineDesc.get_in_type (
    self )
```

This method returns the type of AI framework.

##### Parameters

<i>self</i>	this object
-------------	-------------

**Returns**

the input type of the frontend compilation

Definition at line 111 of file AxfcMachineDesc.py.

**8.12.3.4 get\_layer\_info()**

```
def AxfcMachineDesc.AxfcMachineDesc.get_layer_info (
    self,
    layer_type )
```

This method returns the information of a specific AIX layer.

**Parameters**

<i>self</i>	this object
<i>layer_type</i>	the name of an AIX layer type to be returned

**Returns**

an operation information

Definition at line 88 of file AxfcMachineDesc.py.

**8.12.3.5 get\_profit\_threshold()**

```
def AxfcMachineDesc.AxfcMachineDesc.get_profit_threshold (
    self )
```

This method returns the input type of the frontend compilation.

**Parameters**

<i>self</i>	this object
-------------	-------------

**Returns**

the profit threshold to determine whether to use hardware acceleration

Definition at line 128 of file AxfcMachineDesc.py.

#### 8.12.3.6 read\_file()

```
def AxfcMachineDesc.AxfcMachineDesc.read_file (
    self,
    path )
```

This method is used to read a machine description from the given input path.

##### Parameters

<i>self</i>	this object
<i>path</i>	file path of the machine description

##### Returns

error info

Definition at line 49 of file AxfcMachineDesc.py.

### 8.12.4 Member Data Documentation

#### 8.12.4.1 \_\_aix\_layer\_info\_tbl

```
AxfcMachineDesc.AxfcMachineDesc.__aix_layer_info_tbl [private]
```

general machine description info.

of dictionary type

Definition at line 42 of file AxfcMachineDesc.py.

#### 8.12.4.2 \_\_aix\_model\_info\_tbl

```
AxfcMachineDesc.AxfcMachineDesc.__aix_model_info_tbl [private]
```

general info.

of dictionary type for AIX compiler

Definition at line 41 of file AxfcMachineDesc.py.



#### 8.12.4.3 DEFAULT\_PROFIT\_THRESHOLD

```
int AxfcMachineDesc.AxfcMachineDesc.DEFAULT_PROFIT_THRESHOLD = 1000 [static]
```

Definition at line 31 of file AxfcMachineDesc.py.

#### 8.12.4.4 TYPE\_MXNET

```
int AxfcMachineDesc.AxfcMachineDesc.TYPE_MXNET = 2 [static]
```

Definition at line 27 of file AxfcMachineDesc.py.

#### 8.12.4.5 TYPE\_PYTORCH

```
int AxfcMachineDesc.AxfcMachineDesc.TYPE_PYTORCH = 1 [static]
```

Definition at line 26 of file AxfcMachineDesc.py.

#### 8.12.4.6 TYPE\_TENSORFLOW

```
int AxfcMachineDesc.AxfcMachineDesc.TYPE_TENSORFLOW = 0 [static]
```

Definition at line 25 of file AxfcMachineDesc.py.

#### 8.12.4.7 TYPE\_UNKNOWN

```
int AxfcMachineDesc.AxfcMachineDesc.TYPE_UNKNOWN = 3 [static]
```

Definition at line 28 of file AxfcMachineDesc.py.

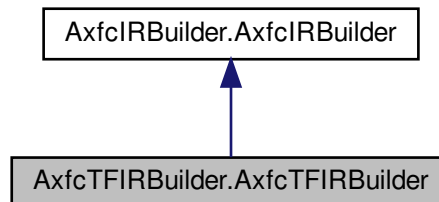
The documentation for this class was generated from the following file:

- /home/youngsun/Project/SKT-AIX/Development/aixc/src/[AxfcMachineDesc.py](#)

## 8.13 AxfcTFIRBuilder.AxfcTFIRBuilder Class Reference

[AxfcTFIRBuilder](#) class.

Inheritance diagram for AxfcTFIRBuilder.AxfcTFIRBuilder:



### Public Member Functions

- `def __init__(self, md)`  
*The constructor.*
- `def __str__(self)`  
*For debugging.*

### Private Member Functions

- `def _read_model_graph`  
*This method is used to read a tensorflow graph from an input file in the given path.*
- `def _build_naive_ir`  
*This method is used to construct a naive AIXIR using a tensorflow graph.*
- `def __prune_ir_nodes(self)`  
*This method is used to prune unnecessary nodes from the IR graph.*
- `def __append_node_def`

### Private Attributes

- `_tf_graph`

### 8.13.1 Detailed Description

[AxfcTFIRBuilder](#) class.

Definition at line 23 of file AxfcTFIRBuilder.py.

### 8.13.2 Constructor & Destructor Documentation

#### 8.13.2.1 \_\_init\_\_()

```
def AxfcTFIRBuilder.AxfcTFIRBuilder.__init__ (
    self,
    md )
```

The constructor.

Definition at line 26 of file AxfcTFIRBuilder.py.

### 8.13.3 Member Function Documentation

#### 8.13.3.1 \_\_append\_node\_def()

```
def AxfcTFIRBuilder.AxfcTFIRBuilder.__append_node_def (
    self,
    tf_node_def ) [private]
```

Definition at line 141 of file AxfcTFIRBuilder.py.

#### 8.13.3.2 \_\_prune\_ir\_nodes()

```
def AxfcTFIRBuilder.AxfcTFIRBuilder.__prune_ir_nodes (
    self,
    AxfcError ) [private]
```

This method is used to prune unnecessary nodes from the IR graph.

Currently, we will remove "identity" and "pad" nodes for the IR translation.

#### Parameters

<i>self</i>	this object
-------------	-------------

#### Returns

error info

Definition at line 83 of file AxfcTFIRBuilder.py.

#### 8.13.3.3 `__str__()`

```
def AxfcTFIRBuilder.AxfcTFIRBuilder.__str__ (
    self )
```

For debugging.

Definition at line 182 of file AxfcTFIRBuilder.py.

#### 8.13.3.4 `_build_naive_ir()`

```
def AxfcTFIRBuilder.AxfcTFIRBuilder._build_naive_ir (
    self,
    path ) [private]
```

This method is used to construct a naive AIXIR using a tensorflow graph.

##### Parameters

<i>self</i>	this object
<i>path</i>	file path of input network model

##### Returns

error info

Definition at line 55 of file AxfcTFIRBuilder.py.

#### 8.13.3.5 `_read_model_graph()`

```
def AxfcTFIRBuilder.AxfcTFIRBuilder._read_model_graph (
    self,
    path ) [private]
```

This method is used to read a tensorflow graph from an input file in the given path.

##### Parameters

<i>self</i>	this object
<i>path</i>	file path of input network model

##### Returns

error info

Definition at line 34 of file AxfcTFIRBuilder.py.

## 8.13.4 Member Data Documentation

### 8.13.4.1 \_tf\_graph

AxfcTFIRBuilder.AxfcTFIRBuilder.\_tf\_graph [private]

Definition at line 46 of file AxfcTFIRBuilder.py.

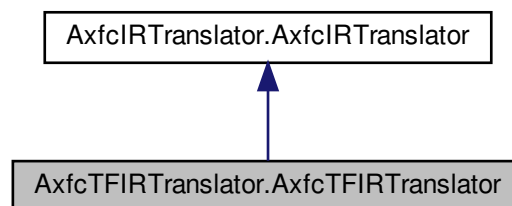
The documentation for this class was generated from the following file:

- /home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcTFIRBuilder.py

## 8.14 AxfcTFIRTranslator.AxfcTFIRTranslator Class Reference

[AxfcTFIRTranslator](#) class.

Inheritance diagram for AxfcTFIRTranslator.AxfcTFIRTranslator:



## Public Member Functions

- def [\\_\\_init\\_\\_](#) (self, md)  
*The constructor.*

## Private Member Functions

- `def __get_aix_data_type`  
*private methods*
- `def __get_aix_tensor_format`  
*This method returns the tensor format of the given node\_def.*
- `def __get_aix_tensor_dims`  
*This method get aixtensor dims from format as dictionary.*
- `def __get_values_of_format`  
*This method get data from aix\_tensor\_format format as dictionary.*
- `def _emit_aix_layer_convolution`  
*protected methods*
- `def _emit_aix_layer_group_conv`  
*This method emits some tensorflow-specific information of the given IR node into the given AIX group convolution layer object.*
- `def _emit_aix_layer_batchnorm`  
*This method emits some tensorflow-specific information of the given IR node into the given AIX batchnorm layer object.*
- `def _emit_aix_layer_avgpool`  
*This method emits some tensorflow-specific information of the given IR node into the given AIX avgpool layer object.*
- `def _emit_aix_layer_maxpool`  
*This method emits some tensorflow-specific information of the given IR node into the given AIX maxpool layer object.*
- `def _emit_aix_layer_ewadd`  
*This method emits some tensorflow-specific information of the given IR node into the given AIX element-wise add (ewadd) layer object.*
- `def _emit_aix_layer_softmax`  
*This method emits some tensorflow-specific information of the given IR node into the given AIX softmax layer object.*
- `def _emit_aix_layer_biasadd`  
*This method emits some tensorflow-specific information of the given IR node into the given AIX biasadd layer object.*
- `def _emit_aix_layer_activation`  
*This method emits some tensorflow-specific information of the given IR node into the given AIX activation layer object.*
- `def _emit_aix_tensor_input`  
*This method emits an AIX tensor of an input type from the given IR node.*
- `def _emit_aix_tensor_filter`  
*This method emits an AIX tensor of an filter type from the given IR node.*
- `def _emit_aix_tensor_bias`  
*This method emits an AIX tensor of an bias type from the given IR node.*
- `def _emit_aix_tensor_scale`  
*This method emits an AIX tensor of an scale type from the given IR node.*
- `def _emit_aix_tensor_mean`  
*This method emits an AIX tensor of an mean type from the given IR node.*
- `def _emit_aix_tensor_variance`  
*This method emits an AIX tensor of an variance type from the given IR node.*
- `def _emit_aix_tensor_output`  
*This method emits an AIX tensor of an output type from the given IR node.*
- `def _emit_aix_convolution_desc`  
*This method emits the AIX convolution description of the given IR node.*
- `def _emit_aix_sampling_desc`  
*This method emits the AIX sampling description of the given IR node.*

## Additional Inherited Members

### 8.14.1 Detailed Description

[AxfcTFIRTranslator](#) class.

Definition at line 45 of file AxfcTFIRTranslator.py.

### 8.14.2 Constructor & Destructor Documentation

#### 8.14.2.1 \_\_init\_\_()

```
def AxfcTFIRTranslator.AxfcTFIRTranslator.__init__ (
    self,
    md )
```

The constructor.

Definition at line 48 of file AxfcTFIRTranslator.py.

### 8.14.3 Member Function Documentation

#### 8.14.3.1 \_\_get\_aix\_data\_type()

```
def AxfcTFIRTranslator.AxfcTFIRTranslator.__get_aix_data_type (
    self,
    tf_node_def ) [private]
```

private methods

This method returns the data type of the given node\_def

#### Parameters

<i>tf_node_def</i>	input node_def
--------------------	----------------

#### Returns

error info.

Definition at line 59 of file AxfcTFIRTranslator.py.

### 8.14.3.2 `__get_aix_tensor_dims()`

```
def AxfcTFIRTranslator.AxfcTFIRTranslator.__get_aix_tensor_dims (
    self,
    aix_tensor ) [private]
```

This method get aixtensor dims from format as dictionary.

#### Parameters

<i>self</i>	this object
<i>AIXTensor</i>	an an AIX tensor data contains dims, data format, dtype, size and ptr

#### Returns

a dictionary object has key as element of data format. e.g input['H'] = 2

Definition at line 106 of file AxfcTFIRTranslator.py.

### 8.14.3.3 `__get_aix_tensor_format()`

```
def AxfcTFIRTranslator.AxfcTFIRTranslator.__get_aix_tensor_format (
    self,
    tf_node_def ) [private]
```

This method returns the tensor format of the given node\_def.

#### Parameters

<i>tf_node_def</i>	input node_def
--------------------	----------------

#### Returns

error info.

Definition at line 83 of file AxfcTFIRTranslator.py.

### 8.14.3.4 `__get_values_of_format()`

```
def AxfcTFIRTranslator.AxfcTFIRTranslator.__get_values_of_format (
    self,
    values ) [private]
```

This method get data from aix\_tensor\_format format as dictionary.



**Parameters**

<i>self</i>	this object
<i>values</i>	an list of input values
<i>tensor_format</i>	an AIX tensor format

**Returns**

a dictionary object has key as element of data format. e.g input['H'] = 2

Definition at line 116 of file AxfcTFIRTranslator.py.

**8.14.3.5 \_emit\_aix\_convolution\_desc()**

```
def AxfcTFIRTranslator.AxfcTFIRTranslator._emit_aix_convolution_desc (
    self,
    ir_node ) [private]
```

This method emits the AIX convolution description of the given IR node.

**Parameters**

<i>self</i>	this object
<i>ir_node</i>	an IR node to be emitted as an AIX tensor

**Returns**

an AIX convolution description

Definition at line 1257 of file AxfcTFIRTranslator.py.

**8.14.3.6 \_emit\_aix\_layer\_activation()**

```
def AxfcTFIRTranslator.AxfcTFIRTranslator._emit_aix_layer_activation (
    self,
    ir_node ) [private]
```

This method emits some tensorflow-specific information of the given IR node into the given AIX activation layer object.

The information includes layer inputs, layer outputs, and so on.

**Parameters**

<i>self</i>	this object
<i>ir_node</i>	an IR node to be emitted

**Returns**

an output AIX activation layer

Definition at line 853 of file AxfcTFIRTranslator.py.

**8.14.3.7 \_emit\_aix\_layer\_avgpool()**

```
def AxfcTFIRTranslator.AxfcTFIRTranslator._emit_aix_layer_avgpool (
    self,
    ir_node ) [private]
```

This method emits some tensorflow-specific information of the given IR node into the given AIX avgpool layer object.

The information includes layer inputs, layer outputs, and so on.

**Parameters**

<i>self</i>	this object
<i>ir_node</i>	an IR node to be emitted

**Returns**

an output AIX avgpool layer

Definition at line 488 of file AxfcTFIRTranslator.py.

**8.14.3.8 \_emit\_aix\_layer\_batchnorm()**

```
def AxfcTFIRTranslator.AxfcTFIRTranslator._emit_aix_layer_batchnorm (
    self,
    ir_node ) [private]
```

This method emits some tensorflow-specific information of the given IR node into the given AIX batchnorm layer object.

The information includes layer inputs, layer outputs, and so on.

**Parameters**

<i>self</i>	this object
<i>ir_node</i>	an IR node to be emitted

**Returns**

an output AIX batchnorm layer

Definition at line 400 of file AxfcTFIRTranslator.py.

#### 8.14.3.9 \_emit\_aix\_layer\_biasadd()

```
def AxfcTFIRTranslator.AxfcTFIRTranslator._emit_aix_layer_biasadd (
    self,
    ir_node ) [private]
```

This method emits some tensorflow-specific information of the given IR node into the given AIX biasadd layer object.

The information includes layer inputs, layer outputs, and so on.

##### Parameters

<i>self</i>	this object
<i>ir_node</i>	an IR node to be emitted

##### Returns

an output AIX avgpool layer

Definition at line 801 of file AxfcTFIRTranslator.py.

#### 8.14.3.10 \_emit\_aix\_layer\_convolution()

```
def AxfcTFIRTranslator.AxfcTFIRTranslator._emit_aix_layer_convolution (
    self,
    ir_node ) [private]
```

protected methods

This method emits some tensorflow-specific information of the given IR node into the given AIX convolution layer object. The information includes layer inputs, layer outputs, and so on.

##### Parameters

<i>self</i>	this object
<i>ir_node</i>	an IR node to be emitted

##### Returns

an output AIX convolution layer

Definition at line 139 of file AxfcTFIRTranslator.py.

#### 8.14.3.11 `_emit_aix_layer_ewadd()`

```
def AxfcTFIRTranslator.AxfcTFIRTranslator._emit_aix_layer_ewadd (
    self,
    ir_node ) [private]
```

This method emits some tensorflow-specific information of the given IR node into the given AIX element-wise add (ewadd) layer object.

The information includes layer inputs, layer outputs, and so on.

##### Parameters

<i>self</i>	this object
<i>ir_node</i>	an IR node to be emitted

##### Returns

an output AIX avgpool layer

Definition at line 658 of file AxfcTFIRTranslator.py.

#### 8.14.3.12 `_emit_aix_layer_group_conv()`

```
def AxfcTFIRTranslator.AxfcTFIRTranslator._emit_aix_layer_group_conv (
    self,
    ir_node ) [private]
```

This method emits some tensorflow-specific information of the given IR node into the given AIX group convolution layer object.

The information includes layer inputs, layer outputs, and so on.

##### Parameters

<i>self</i>	this object
<i>ir_node</i>	an IR node to be emitted

##### Returns

an output AIX convolution layer

Definition at line 268 of file AxfcTFIRTranslator.py.

#### 8.14.3.13 \_emit\_aix\_layer\_maxpool()

```
def AxfcTFIRTranslator.AxfcTFIRTranslator._emit_aix_layer_maxpool (
    self,
    ir_node ) [private]
```

This method emits some tensorflow-specific information of the given IR node into the given AIX maxpool layer object.

The information includes layer inputs, layer outputs, and so on.

##### Parameters

<i>self</i>	this object
<i>ir_node</i>	an IR node to be emitted

##### Returns

an output AIX avgpool layer

Definition at line 568 of file AxfcTFIRTranslator.py.

#### 8.14.3.14 \_emit\_aix\_layer\_softmax()

```
def AxfcTFIRTranslator.AxfcTFIRTranslator._emit_aix_layer_softmax (
    self,
    ir_node ) [private]
```

This method emits some tensorflow-specific information of the given IR node into the given AIX softmax layer object.

The information includes layer inputs, layer outputs, and so on.

##### Parameters

<i>self</i>	this object
<i>ir_node</i>	an IR node to be emitted

##### Returns

an output AIX softmax layer

Definition at line 733 of file AxfcTFIRTranslator.py.

#### 8.14.3.15 \_emit\_aix\_sampling\_desc()

```
def AxfcTFIRTranslator.AxfcTFIRTranslator._emit_aix_sampling_desc (
    self,
    ir_node ) [private]
```

This method emits the AIX sampling description of the given IR node.

**Parameters**

<i>self</i>	this object
<i>ir_node</i>	an IR node to be emitted as an AIX tensor

**Returns**

an AIX sampling description

Definition at line 1378 of file AxfcTFIRTranslator.py.

**8.14.3.16 \_emit\_aix\_tensor\_bias()**

```
def AxfcTFIRTranslator.AxfcTFIRTranslator._emit_aix_tensor_bias (
    self,
    ir_node ) [private]
```

This method emits an AIX tensor of an bias type from the given IR node.

**Parameters**

<i>self</i>	this object
<i>ir_node</i>	an IR node to be emitted as an AIX tensor
<i>is_default</i>	indicates if default values are used to emit

**Returns**

an AIX tensor of an bias type

Definition at line 1051 of file AxfcTFIRTranslator.py.

**8.14.3.17 \_emit\_aix\_tensor\_filter()**

```
def AxfcTFIRTranslator.AxfcTFIRTranslator._emit_aix_tensor_filter (
    self,
    ir_node ) [private]
```

This method emits an AIX tensor of an filter type from the given IR node.

**Parameters**

<i>self</i>	this object
<i>ir_node</i>	an IR node to be emitted as an AIX tensor
<i>is_default</i>	indicates if default values are used to emit

**Returns**

an AIX tensor of an filter type

Definition at line 990 of file AxfcTFIRTranslator.py.

**8.14.3.18 \_emit\_aix\_tensor\_input()**

```
def AxfcTFIRTranslator.AxfcTFIRTranslator._emit_aix_tensor_input (
    self,
    ir_node ) [private]
```

This method emits an AIX tensor of an input type from the given IR node.

**Parameters**

<i>self</i>	this object
<i>ir_node</i>	an IR node to be emitted as an AIX tensor

**Returns**

an AIX tensor of an input type

Definition at line 928 of file AxfcTFIRTranslator.py.

**8.14.3.19 \_emit\_aix\_tensor\_mean()**

```
def AxfcTFIRTranslator.AxfcTFIRTranslator._emit_aix_tensor_mean (
    self,
    ir_node ) [private]
```

This method emits an AIX tensor of an mean type from the given IR node.

**Parameters**

<i>self</i>	this object
<i>ir_node</i>	an IR node to be emitted as an AIX tensor
<i>is_default</i>	indicates if default values are used to emit

**Returns**

an AIX tensor of an mean type

Definition at line 1139 of file AxfcTFIRTranslator.py.



8.14.3.20 `_emit_aix_tensor_output()`

```
def AxfcTFIRTranslator.AxfcTFIRTranslator._emit_aix_tensor_output (
    self,
    ir_node ) [private]
```

This method emits an AIX tensor of an output type from the given IR node.

**Parameters**

<i>self</i>	this object
<i>ir_node</i>	an IR node to be emitted as an AIX tensor
<i>output_dims</i>	output dimensions

**Returns**

an AIX tensor of an output type

Definition at line 1229 of file AxfcTFIRTranslator.py.

8.14.3.21 `_emit_aix_tensor_scale()`

```
def AxfcTFIRTranslator.AxfcTFIRTranslator._emit_aix_tensor_scale (
    self,
    ir_node ) [private]
```

This method emits an AIX tensor of an scale type from the given IR node.

**Parameters**

<i>self</i>	this object
<i>ir_node</i>	an IR node to be emitted as an AIX tensor
<i>is_default</i>	indicates if default values are used to emit

**Returns**

an AIX tensor of an scale type

Definition at line 1094 of file AxfcTFIRTranslator.py.

8.14.3.22 `_emit_aix_tensor_variance()`

```
def AxfcTFIRTranslator.AxfcTFIRTranslator._emit_aix_tensor_variance (
    self,
    ir_node ) [private]
```

This method emits an AIX tensor of an variance type from the given IR node.

**Parameters**

<i>self</i>	this object
<i>ir_node</i>	an IR node to be emitted as an AIX tensor
<i>is_default</i>	indicates if default values are used to emit

**Returns**

an AIX tensor of an variance type

Definition at line 1184 of file AxfcTFIRTranslator.py.

The documentation for this class was generated from the following file:

- /home/youngsun/Project/SKT-AIX/Development/aixc/src/[AxfcTFIRTranslator.py](#)

## Chapter 9

# File Documentation

### 9.1 /home/youngsun/Project/SKT-AIX/Development/aixc/README.md File Reference

### 9.2 /home/youngsun/Project/SKT-AIX/Development/aixc/src/aixh\_pb2.py File Reference

#### Namespaces

- [aixh\\_pb2](#)

#### Variables

- `int aixh_pb2._b = sys.version_info[0] < 3 and (lambda x: x) or (lambda x: x.encode('latin1'))`
- `aixh_pb2._sym_db = _symbol_database.Default()`
- `aixh_pb2.DESRIPTOR`
- `aixh_pb2._AIXLAYER_AIXLAYERTYPE`
- `aixh_pb2._AIXLAYER_AIXACTIVATIONMODE`
- `aixh_pb2._AIXLAYER_AIXSAMPLINGMODE`
- `aixh_pb2._AIXLAYER_AIXDATATYPE`
- `aixh_pb2._AIXLAYER_AIXTENSORFORMAT`
- `aixh_pb2._AIXLAYER_AIXCONVOLUTIONDESC`
- `aixh_pb2._AIXLAYER_AIXSAMPLINGDESC`
- `aixh_pb2._AIXLAYER_AIXEWADDDDESC`
- `aixh_pb2._AIXLAYER_AIXTENSOR`
- `aixh_pb2._AIXLAYER`
- `aixh_pb2._AIXGRAPH`
- `aixh_pb2.enum_type`
- `aixh_pb2.containing_type`
- `aixh_pb2.message_type`
- `aixh_pb2.AIXLayer`
- `aixh_pb2.AIXGraph`
- `aixh_pb2.has_options`
- `aixh_pb2._options`

### 9.3 /home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcError.py File Reference

#### Classes

- class [AxfcError.AxfcError](#)  
*AxfcError* enum class.

#### Namespaces

- [AxfcError](#)

### 9.4 /home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcFrontendCompiler.py File Reference

#### Classes

- class [AxfcFrontendCompiler.AxfcFrontendCompiler](#)  
*AxfcFrontendCompiler*.

#### Namespaces

- [AxfcFrontendCompiler](#)

### 9.5 /home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcGraphWriter.py File Reference

#### Classes

- class [AxfcGraphWriter.AxfcGraphWriter](#)  
*AxfcGraphWriter* class.

#### Namespaces

- [AxfcGraphWriter](#)

### 9.6 /home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcIRBlock.py File Reference

#### Classes

- class [AxfcIRBlock.AxfcIRBlock](#)  
*AxfcIRBlock* class.

## Namespaces

- [AxfclRBlock](#)

## 9.7 /home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfclRBuilder.py File Reference

### Classes

- class [AxfclRBuilder.AxfclRBuilder](#)  
*AxfclRBuilder class.*

## Namespaces

- [AxfclRBuilder](#)

## 9.8 /home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfclRGraph.py File Reference

### Classes

- class [AxfclRGraph.AxfclRGraph](#)  
*AxfclRGraph class.*

## Namespaces

- [AxfclRGraph](#)

## 9.9 /home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfclRNode.py File Reference

### Classes

- class [AxfclRNode.AxfclRNode](#)  
*AxfclRNode.*

## Namespaces

- [AxfclRNode](#)

## 9.10 /home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfclRTranslator.py File Reference

### Classes

- class [AxfclRTranslator.AIXTensorType](#)  
*AIXInputType enum class.*
- class [AxfclRTranslator.AxfclRTranslator](#)

### Namespaces

- [AxfclRTranslator](#)

## 9.11 /home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcLauncherWriter.py File Reference

### Classes

- class [AxfcLauncherWriter.AxfcLauncherWriter](#)  
*AxfcLauncherWriter class.*

### Namespaces

- [AxfcLauncherWriter](#)

## 9.12 /home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcMachineDesc.py File Reference

### Classes

- class [AxfcMachineDesc.AxfcMachineDesc](#)  
*AxfcMachineDesc class.*
- class [AxfcMachineDesc.AxfcMachineDesc.AIXLayerInfo](#)  
*AIXLayerInfo inner class.*

### Namespaces

- [AxfcMachineDesc](#)

## 9.13 /home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcMain.py File Reference

### Namespaces

- [AxfcMain](#)

## Functions

- def [AxfcMain.\\_\\_main](#) (vargs)  
*main function*

## Variables

- [AxfcMain.parser](#)
- [AxfcMain.metavar](#)
- [AxfcMain.type](#)
- [AxfcMain.str](#)
- [AxfcMain.required](#)
- [AxfcMain.help](#)
- [AxfcMain.args](#) = parser.parse\_args()

## 9.14 /home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcTFIRBuilder.py File Reference

### Classes

- class [AxfcTFIRBuilder.AxfcTFIRBuilder](#)  
*AxfcTFIRBuilder class.*

### Namespaces

- [AxfcTFIRBuilder](#)

## 9.15 /home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcTFIRTranslator.py File Reference

### Classes

- class [AxfcTFIRTranslator.AxfcTFIRTranslator](#)  
*AxfcTFIRTranslator class.*

### Namespaces

- [AxfcTFIRTranslator](#)

### Variables

- dictionary [AxfcTFIRTranslator.aix\\_data\\_type\\_tbl](#)  
*Global tables for AIXDataType and AIXTensorFormat.*
- dictionary [AxfcTFIRTranslator.aix\\_tensor\\_format\\_tbl](#)  
*AIXTensorFormat table.*

## 9.16 /home/youngsun/Project/SKT-AIX/Development/aixc/tst/aix\_tf.md File Reference





# Index

/home/youngsun/Project/SKT-AIX/Development/aixc/↔ README.md, 93  
/home/youngsun/Project/SKT-AIX/Development/aixc/src/↔ AixfError.py, 94  
/home/youngsun/Project/SKT-AIX/Development/aixc/src/↔ AixfFrontendCompiler.py, 94  
/home/youngsun/Project/SKT-AIX/Development/aixc/src/↔ AixfGraphWriter.py, 94  
/home/youngsun/Project/SKT-AIX/Development/aixc/src/↔ AixfIRBlock.py, 94  
/home/youngsun/Project/SKT-AIX/Development/aixc/src/↔ AixfIRBuilder.py, 95  
/home/youngsun/Project/SKT-AIX/Development/aixc/src/↔ AixfIRGraph.py, 95  
/home/youngsun/Project/SKT-AIX/Development/aixc/src/↔ AixfIRNode.py, 95  
/home/youngsun/Project/SKT-AIX/Development/aixc/src/↔ AixfIRTranslator.py, 96  
/home/youngsun/Project/SKT-AIX/Development/aixc/src/↔ AixfLauncherWriter.py, 96  
/home/youngsun/Project/SKT-AIX/Development/aixc/src/↔ AixfMachineDesc.py, 96  
/home/youngsun/Project/SKT-AIX/Development/aixc/src/↔ AixfMain.py, 96  
/home/youngsun/Project/SKT-AIX/Development/aixc/src/↔ AixfTFIRBuilder.py, 97  
/home/youngsun/Project/SKT-AIX/Development/aixc/src/↔ AixfTFIRTranslator.py, 97  
/home/youngsun/Project/SKT-AIX/Development/aixc/src/aixh↔ \_pb2.py, 93  
/home/youngsun/Project/SKT-AIX/Development/aixc/tst/aixh↔ \_tf.md, 97  
\_AIXGRAPH  
  aixh\_pb2, 13  
\_AIXLAYER  
  aixh\_pb2, 13  
\_AIXLAYER\_AIXACTIVATIONMODE  
  aixh\_pb2, 14  
\_AIXLAYER\_AIXCONVOLUTIONDESC  
  aixh\_pb2, 14  
\_AIXLAYER\_AIXDATATYPE  
  aixh\_pb2, 14  
\_AIXLAYER\_AIXEWADDDDESC  
  aixh\_pb2, 14  
\_AIXLAYER\_AIXLAYERTYPE  
  aixh\_pb2, 14  
\_AIXLAYER\_AIXSAMPLINGDESC  
  aixh\_pb2, 15  
\_AIXLAYER\_AIXSAMPLINGMODE  
  aixh\_pb2, 15  
  \_AIXLAYER\_AIXTENSOR  
  \_AIXLAYER\_AIXTENSORFORMAT  
  \_\_aix\_layer\_info\_tbl  
  AixfMachineDesc::AixfMachineDesc, 74  
  \_\_aix\_model\_info\_tbl  
  AixfMachineDesc::AixfMachineDesc, 74  
  \_\_analyse\_inout  
  AixfIRBlock::AixfIRBlock, 42  
  \_\_append\_node\_def  
  AixfTFIRBuilder::AixfTFIRBuilder, 77  
  \_\_del\_\_  
  AixfIRNode::AixfIRNode, 55  
  \_\_edge\_id  
  AixfGraphWriter::AixfGraphWriter, 40  
  \_\_emit\_aix\_layer\_tbl  
  AixfIRTranslator::AixfIRTranslator, 67  
  \_\_emit\_aixh\_block  
  AixfIRTranslator::AixfIRTranslator, 62  
  \_\_emit\_aixh\_node  
  AixfIRTranslator::AixfIRTranslator, 62  
  \_\_eq\_\_  
  AixfIRNode::AixfIRNode, 56  
  \_\_find\_aixh\_blocks  
  AixfIRBuilder::AixfIRBuilder, 46  
  \_\_get\_aix\_data\_type  
  AixfTFIRTranslator::AixfTFIRTranslator, 81  
  \_\_get\_aix\_tensor\_dims  
  AixfTFIRTranslator::AixfTFIRTranslator, 81  
  \_\_get\_aix\_tensor\_format  
  AixfTFIRTranslator::AixfTFIRTranslator, 82  
  \_\_get\_values\_of\_format  
  AixfTFIRTranslator::AixfTFIRTranslator, 82  
  \_\_graph  
  AixfGraphWriter::AixfGraphWriter, 40  
  \_\_hash\_\_  
  AixfIRNode::AixfIRNode, 56  
  \_\_init\_\_  
  AixfFrontendCompiler::AixfFrontendCompiler, 35  
  AixfGraphWriter::AixfGraphWriter, 38  
  AixfIRBlock::AixfIRBlock, 42  
  AixfIRBuilder::AixfIRBuilder, 46  
  AixfIRGraph::AixfIRGraph, 51  
  AixfIRNode::AixfIRNode, 55  
  AixfIRTranslator::AixfIRTranslator, 61  
  AixfLauncherWriter::AixfLauncherWriter, 69  
  AixfMachineDesc::AixfMachineDesc, 71

- AxfcMachineDesc::AxfcMachineDesc::AIXLayer↔  
Info, 24
- AxfcTFIRBuilder::AxfcTFIRBuilder, 77
- AxfcTFIRTranslator::AxfcTFIRTranslator, 81
- \_\_ir\_builder
  - AxfcFrontendCompiler::AxfcFrontendCompiler, 37
- \_\_ir\_graph
  - AxfcLauncherWriter::AxfcLauncherWriter, 70
- \_\_ir\_translator
  - AxfcFrontendCompiler::AxfcFrontendCompiler, 37
- \_\_main
  - AxfcMain, 19
- \_\_md
  - AxfcFrontendCompiler::AxfcFrontendCompiler, 37
  - AxfcLauncherWriter::AxfcLauncherWriter, 70
- \_\_nodes
  - AxfcGraphWriter::AxfcGraphWriter, 40
- \_\_perform\_maximal\_munch
  - AxfcIRBuilder::AxfcIRBuilder, 47
- \_\_prune\_ir\_nodes
  - AxfcTFIRBuilder::AxfcTFIRBuilder, 77
- \_\_str\_\_
  - AxfcFrontendCompiler::AxfcFrontendCompiler, 35
  - AxfcIRBlock::AxfcIRBlock, 42
  - AxfcIRBuilder::AxfcIRBuilder, 47
  - AxfcIRGraph::AxfcIRGraph, 51
  - AxfcIRNode::AxfcIRNode, 56
  - AxfcIRTranslator::AxfcIRTranslator, 63
  - AxfcLauncherWriter::AxfcLauncherWriter, 69
  - AxfcMachineDesc::AxfcMachineDesc, 72
  - AxfcMachineDesc::AxfcMachineDesc::AIXLayer↔  
Info, 24
  - AxfcTFIRBuilder::AxfcTFIRBuilder, 77
- \_\_x\_axis
  - AxfcGraphWriter::AxfcGraphWriter, 40
- \_\_y\_axis
  - AxfcGraphWriter::AxfcGraphWriter, 41
- \_aix\_graph
  - AxfcIRTranslator::AxfcIRTranslator, 67
- \_b
  - aixh\_pb2, 15
- \_build\_naive\_ir
  - AxfcIRBuilder::AxfcIRBuilder, 47
  - AxfcTFIRBuilder::AxfcTFIRBuilder, 78
- \_emit\_aix\_convolution\_desc
  - AxfcIRTranslator::AxfcIRTranslator, 63
  - AxfcTFIRTranslator::AxfcTFIRTranslator, 83
- \_emit\_aix\_layer\_activation
  - AxfcIRTranslator::AxfcIRTranslator, 63
  - AxfcTFIRTranslator::AxfcTFIRTranslator, 83
- \_emit\_aix\_layer\_avgpool
  - AxfcIRTranslator::AxfcIRTranslator, 63
  - AxfcTFIRTranslator::AxfcTFIRTranslator, 84
- \_emit\_aix\_layer\_batchnorm
  - AxfcIRTranslator::AxfcIRTranslator, 63
  - AxfcTFIRTranslator::AxfcTFIRTranslator, 84
- \_emit\_aix\_layer\_biasadd
  - AxfcIRTranslator::AxfcIRTranslator, 64
- AxfcTFIRTranslator::AxfcTFIRTranslator, 85
- \_emit\_aix\_layer\_convolution
  - AxfcIRTranslator::AxfcIRTranslator, 64
  - AxfcTFIRTranslator::AxfcTFIRTranslator, 85
- \_emit\_aix\_layer\_ewadd
  - AxfcIRTranslator::AxfcIRTranslator, 64
  - AxfcTFIRTranslator::AxfcTFIRTranslator, 85
- \_emit\_aix\_layer\_group\_conv
  - AxfcIRTranslator::AxfcIRTranslator, 64
  - AxfcTFIRTranslator::AxfcTFIRTranslator, 86
- \_emit\_aix\_layer\_maxpool
  - AxfcIRTranslator::AxfcIRTranslator, 64
  - AxfcTFIRTranslator::AxfcTFIRTranslator, 86
- \_emit\_aix\_layer\_softmax
  - AxfcIRTranslator::AxfcIRTranslator, 65
  - AxfcTFIRTranslator::AxfcTFIRTranslator, 87
- \_emit\_aix\_sampling\_desc
  - AxfcIRTranslator::AxfcIRTranslator, 65
  - AxfcTFIRTranslator::AxfcTFIRTranslator, 87
- \_emit\_aix\_tensor\_bias
  - AxfcIRTranslator::AxfcIRTranslator, 65
  - AxfcTFIRTranslator::AxfcTFIRTranslator, 89
- \_emit\_aix\_tensor\_filter
  - AxfcIRTranslator::AxfcIRTranslator, 65
  - AxfcTFIRTranslator::AxfcTFIRTranslator, 89
- \_emit\_aix\_tensor\_input
  - AxfcIRTranslator::AxfcIRTranslator, 65
  - AxfcTFIRTranslator::AxfcTFIRTranslator, 90
- \_emit\_aix\_tensor\_mean
  - AxfcIRTranslator::AxfcIRTranslator, 66
  - AxfcTFIRTranslator::AxfcTFIRTranslator, 90
- \_emit\_aix\_tensor\_output
  - AxfcIRTranslator::AxfcIRTranslator, 66
  - AxfcTFIRTranslator::AxfcTFIRTranslator, 90
- \_emit\_aix\_tensor\_scale
  - AxfcIRTranslator::AxfcIRTranslator, 66
  - AxfcTFIRTranslator::AxfcTFIRTranslator, 91
- \_emit\_aix\_tensor\_variance
  - AxfcIRTranslator::AxfcIRTranslator, 66
  - AxfcTFIRTranslator::AxfcTFIRTranslator, 91
- \_get\_emitted\_input\_nodes
  - AxfcIRTranslator::AxfcIRTranslator, 66
- \_ir\_graph
  - AxfcIRBuilder::AxfcIRBuilder, 49
- \_ir\_symtab
  - AxfcIRBuilder::AxfcIRBuilder, 49
  - AxfcIRTranslator::AxfcIRTranslator, 68
- \_md
  - AxfcIRBuilder::AxfcIRBuilder, 49
  - AxfcIRTranslator::AxfcIRTranslator, 68
- \_options
  - aixh\_pb2, 16
- \_read\_model\_graph
  - AxfcIRBuilder::AxfcIRBuilder, 48
  - AxfcTFIRBuilder::AxfcTFIRBuilder, 78
- \_sym\_db
  - aixh\_pb2, 16
- \_tf\_graph

- AxfcIRBuilder::AxfcIRBuilder, 49
- AxfcTFIRBuilder::AxfcTFIRBuilder, 79
- AIX\_TENSOR\_BIAS
  - AxfcIRTranslator::AIXTensorType, 26
- AIX\_TENSOR\_FILTER
  - AxfcIRTranslator::AIXTensorType, 26
- AIX\_TENSOR\_INPUT
  - AxfcIRTranslator::AIXTensorType, 27
- AIX\_TENSOR\_MEAN
  - AxfcIRTranslator::AIXTensorType, 27
- AIX\_TENSOR\_OUTPUT
  - AxfcIRTranslator::AIXTensorType, 27
- AIX\_TENSOR\_SCALE
  - AxfcIRTranslator::AIXTensorType, 27
- AIX\_TENSOR\_UNKNOWN
  - AxfcIRTranslator::AIXTensorType, 27
- AIX\_TENSOR\_VARIANCE
  - AxfcIRTranslator::AIXTensorType, 27
- AIXGraph
  - aixh\_pb2, 16
- AIXLayer
  - aixh\_pb2, 16
- activation
  - AxfcMachineDesc::AxfcMachineDesc::AIXLayer←  
Info, 24
- add\_edge
  - AxfcGraphWriter::AxfcGraphWriter, 39
- add\_node
  - AxfcGraphWriter::AxfcGraphWriter, 39
- aix\_data\_type\_tbl
  - AxfcTFIRTranslator, 22
- aix\_graph
  - AxfcIRBlock::AxfcIRBlock, 44
- aix\_graphs
  - AxfcIRTranslator::AxfcIRTranslator, 68
- aix\_layer
  - AxfcIRNode::AxfcIRNode, 57
- aix\_tensor\_format\_tbl
  - AxfcTFIRTranslator, 22
- aixh\_pb2, 13
  - \_AIXGRAPH, 13
  - \_AIXLAYER, 13
  - \_AIXLAYER\_AIXACTIVATIONMODE, 14
  - \_AIXLAYER\_AIXCONVOLUTIONDESC, 14
  - \_AIXLAYER\_AIXDATATYPE, 14
  - \_AIXLAYER\_AIXEWADDDDESC, 14
  - \_AIXLAYER\_AIXLAYERTYPE, 14
  - \_AIXLAYER\_AIXSAMPLINGDESC, 15
  - \_AIXLAYER\_AIXSAMPLINGMODE, 15
  - \_AIXLAYER\_AIXTENSOR, 15
  - \_AIXLAYER\_AIXTENSORFORMAT, 15
  - \_b, 15
  - \_options, 16
  - \_sym\_db, 16
  - AIXGraph, 16
  - AIXLayer, 16
  - containing\_type, 16
  - DESCRIPTOR, 17
  - enum\_type, 17
  - has\_options, 17
  - message\_type, 17
- aixh\_profit
  - AxfcIRBlock::AxfcIRBlock, 44
  - AxfcIRNode::AxfcIRNode, 57
- analyse\_liveness
  - AxfcIRBlock::AxfcIRBlock, 43
  - AxfcIRGraph::AxfcIRGraph, 51
- analyze\_profit
  - AxfcIRBlock::AxfcIRBlock, 43
  - AxfcIRNode::AxfcIRNode, 56
- append\_block
  - AxfcIRGraph::AxfcIRGraph, 52
- append\_node
  - AxfcIRGraph::AxfcIRGraph, 52
- args
  - AxfcMain, 20
- AxfcError, 17
- AxfcError.AxfcError, 28
- AxfcError::AxfcError
  - DUMP\_IR\_GRAPH\_ERROR, 29
  - EMPTY\_IR\_BLOCK, 29
  - INVALID\_ACTIVATION\_LAYER, 29
  - INVALID\_AIX\_GRAPH, 29
  - INVALID\_AIX\_LAYER\_TYPE, 30
  - INVALID\_AIX\_TENSOR\_FORMAT, 30
  - INVALID\_AIX\_TENSOR\_INPUT, 30
  - INVALID\_BATCHNORM\_LAYER, 30
  - INVALID\_CONVOLUTION\_LAYER, 30
  - INVALID\_EWADD\_LAYER, 30
  - INVALID\_FILE\_PATH, 31
  - INVALID\_GROUP\_CONV\_LAYER, 31
  - INVALID\_IDENTITY\_LAYER, 31
  - INVALID\_INPUT\_TYPE, 31
  - INVALID\_IR\_GRAPH, 31
  - INVALID\_MAXPOOL\_LAYER, 31
  - INVALID\_MD\_FORMAT, 32
  - INVALID\_PAD\_LAYER, 32
  - INVALID\_PARAMETER, 32
  - INVALID\_TF\_GRAPH, 32
  - INVALID\_BIASADD\_LAYER, 32
  - NOT\_AIXH\_SUPPORT, 32
  - NOT\_IMPLEMENTED, 33
  - PRED\_NODE\_NOT\_FOUND, 33
  - SUCCESS, 33
  - UNKNOWN\_TENSOR\_TYPE, 33
  - UNREMOVED\_IDENTITY, 33
  - UNSUPPORTED\_AIX\_LAYER\_EMIT, 33
- AxfcFrontendCompiler, 18
- AxfcFrontendCompiler.AxfcFrontendCompiler, 34
- AxfcFrontendCompiler::AxfcFrontendCompiler
  - \_\_init\_\_, 35
  - \_\_ir\_builder, 37
  - \_\_ir\_translator, 37
  - \_\_md, 37
  - \_\_str\_\_, 35
  - compile, 35

- dump\_aix\_graphs, 35
- dump\_launcher, 36
- get\_ir\_graph, 36
- read\_md\_file, 36
- AxflIRGraphWriter, 18
- AxflIRGraphWriter.AxflIRGraphWriter, 38
- AxflIRGraphWriter::AxflIRGraphWriter
  - \_\_edge\_id, 40
  - \_\_graph, 40
  - \_\_init\_\_, 38
  - \_\_nodes, 40
  - \_\_x\_axis, 40
  - \_\_y\_axis, 41
  - add\_edge, 39
  - add\_node, 39
  - write\_file, 39
- AxflIRBlock, 18
- AxflIRBlock.AxflIRBlock, 41
- AxflIRBlock::AxflIRBlock
  - \_\_analyse\_inout, 42
  - \_\_init\_\_, 42
  - \_\_str\_\_, 42
  - aix\_graph, 44
  - aixh\_profit, 44
  - analyse\_liveness, 43
  - analyse\_profit, 43
  - id, 44
  - is\_aixh\_support, 44
  - live\_in, 44
  - live\_out, 44
  - nodes, 45
- AxflIRBuilder, 18
- AxflIRBuilder.AxflIRBuilder, 45
- AxflIRBuilder::AxflIRBuilder
  - \_\_find\_aixh\_blocks, 46
  - \_\_init\_\_, 46
  - \_\_perform\_maximal\_munch, 47
  - \_\_str\_\_, 47
  - \_build\_naive\_ir, 47
  - \_ir\_graph, 49
  - \_ir\_syntab, 49
  - \_md, 49
  - \_read\_model\_graph, 48
  - \_tf\_graph, 49
  - build\_ir, 48
- AxflIRGraph, 18
- AxflIRGraph.AxflIRGraph, 50
- AxflIRGraph::AxflIRGraph
  - \_\_init\_\_, 51
  - \_\_str\_\_, 51
  - analyse\_liveness, 51
  - append\_block, 52
  - append\_node, 52
  - blocks, 53
  - dump\_to\_file, 52
  - nodes, 53
  - root\_node, 53
  - syntab, 53
- AxflIRNode, 18
- AxflIRNode.AxflIRNode, 54
- AxflIRNode::AxflIRNode
  - \_\_del\_\_, 55
  - \_\_eq\_\_, 56
  - \_\_hash\_\_, 56
  - \_\_init\_\_, 55
  - \_\_str\_\_, 56
  - aix\_layer, 57
  - aixh\_profit, 57
  - analyse\_profit, 56
  - block\_ref, 57
  - eval\_flag, 57
  - id, 58
  - is\_aixh\_support, 58
  - is\_input, 58
  - is\_output, 58
  - layer\_id, 58
  - name, 59
  - node\_def, 59
  - op, 59
  - preds, 59
  - succs, 59
- AxflIRTranslator, 19
- AxflIRTranslator.AIXTensorType, 26
- AxflIRTranslator.AxflIRTranslator, 60
- AxflIRTranslator::AIXTensorType
  - AIX\_TENSOR\_BIAS, 26
  - AIX\_TENSOR\_FILTER, 26
  - AIX\_TENSOR\_INPUT, 27
  - AIX\_TENSOR\_MEAN, 27
  - AIX\_TENSOR\_OUTPUT, 27
  - AIX\_TENSOR\_SCALE, 27
  - AIX\_TENSOR\_UNKNOWN, 27
  - AIX\_TENSOR\_VARIANCE, 27
- AxflIRTranslator::AxflIRTranslator
  - \_\_emit\_aix\_layer\_tbl, 67
  - \_\_emit\_aixh\_block, 62
  - \_\_emit\_aixh\_node, 62
  - \_\_init\_\_, 61
  - \_\_str\_\_, 63
  - \_aix\_graph, 67
  - \_emit\_aix\_convolution\_desc, 63
  - \_emit\_aix\_layer\_activation, 63
  - \_emit\_aix\_layer\_avgpool, 63
  - \_emit\_aix\_layer\_batchnorm, 63
  - \_emit\_aix\_layer\_biasadd, 64
  - \_emit\_aix\_layer\_convolution, 64
  - \_emit\_aix\_layer\_ewadd, 64
  - \_emit\_aix\_layer\_group\_conv, 64
  - \_emit\_aix\_layer\_maxpool, 64
  - \_emit\_aix\_layer\_softmax, 65
  - \_emit\_aix\_sampling\_desc, 65
  - \_emit\_aix\_tensor\_bias, 65
  - \_emit\_aix\_tensor\_filter, 65
  - \_emit\_aix\_tensor\_input, 65
  - \_emit\_aix\_tensor\_mean, 66
  - \_emit\_aix\_tensor\_output, 66

- [\\_emit\\_aix\\_tensor\\_scale, 66](#)
- [\\_emit\\_aix\\_tensor\\_variance, 66](#)
- [\\_get\\_emitted\\_input\\_nodes, 66](#)
- [\\_ir\\_syntab, 68](#)
- [\\_md, 68](#)
- [aix\\_graphs, 68](#)
- [emit\\_aixh\\_graphs, 67](#)
- [AxfcLauncherWriter, 19](#)
- [AxfcLauncherWriter.AxfcLauncherWriter, 68](#)
- [AxfcLauncherWriter::AxfcLauncherWriter](#)
  - [\\_\\_init\\_\\_, 69](#)
  - [\\_\\_ir\\_graph, 70](#)
  - [\\_\\_md, 70](#)
  - [\\_\\_str\\_\\_, 69](#)
  - [emit\\_aixh\\_launcher, 69](#)
- [AxfcMachineDesc, 19](#)
- [AxfcMachineDesc.AxfcMachineDesc, 70](#)
- [AxfcMachineDesc.AxfcMachineDesc.AIXLayerInfo, 23](#)
- [AxfcMachineDesc::AxfcMachineDesc](#)
  - [\\_\\_aix\\_layer\\_info\\_tbl, 74](#)
  - [\\_\\_aix\\_model\\_info\\_tbl, 74](#)
  - [\\_\\_init\\_\\_, 71](#)
  - [\\_\\_str\\_\\_, 72](#)
  - [DEFAULT\\_PROFIT\\_THRESHOLD, 74](#)
  - [get\\_aixh\\_support, 72](#)
  - [get\\_in\\_type, 72](#)
  - [get\\_layer\\_info, 73](#)
  - [get\\_profit\\_threshold, 73](#)
  - [read\\_file, 73](#)
  - [TYPE\\_MXNET, 75](#)
  - [TYPE\\_PYTORCH, 75](#)
  - [TYPE\\_TENSORFLOW, 75](#)
  - [TYPE\\_UNKNOWN, 75](#)
- [AxfcMachineDesc::AxfcMachineDesc::AIXLayerInfo](#)
  - [\\_\\_init\\_\\_, 24](#)
  - [\\_\\_str\\_\\_, 24](#)
  - [activation, 24](#)
  - [is\\_conv, 24](#)
  - [is\\_group, 24](#)
  - [layer, 25](#)
  - [op, 25](#)
  - [profit, 25](#)
- [AxfcMain, 19](#)
  - [\\_\\_main\\_\\_, 19](#)
  - [args, 20](#)
  - [help, 20](#)
  - [metavar, 20](#)
  - [parser, 20](#)
  - [required, 20](#)
  - [str, 21](#)
  - [type, 21](#)
- [AxfcTFIRBuilder, 21](#)
- [AxfcTFIRBuilder.AxfcTFIRBuilder, 76](#)
- [AxfcTFIRBuilder::AxfcTFIRBuilder](#)
  - [\\_\\_append\\_node\\_def, 77](#)
  - [\\_\\_init\\_\\_, 77](#)
  - [\\_\\_prune\\_ir\\_nodes, 77](#)
  - [\\_\\_str\\_\\_, 77](#)
  - [\\_\\_build\\_naive\\_ir, 78](#)
  - [\\_\\_read\\_model\\_graph, 78](#)
  - [\\_\\_tf\\_graph, 79](#)
- [AxfcTFIRTranslator, 21](#)
  - [aix\\_data\\_type\\_tbl, 22](#)
  - [aix\\_tensor\\_format\\_tbl, 22](#)
- [AxfcTFIRTranslator.AxfcTFIRTranslator, 79](#)
- [AxfcTFIRTranslator::AxfcTFIRTranslator](#)
  - [\\_\\_get\\_aix\\_data\\_type, 81](#)
  - [\\_\\_get\\_aix\\_tensor\\_dims, 81](#)
  - [\\_\\_get\\_aix\\_tensor\\_format, 82](#)
  - [\\_\\_get\\_values\\_of\\_format, 82](#)
  - [\\_\\_init\\_\\_, 81](#)
  - [\\_emit\\_aix\\_convolution\\_desc, 83](#)
  - [\\_emit\\_aix\\_layer\\_activation, 83](#)
  - [\\_emit\\_aix\\_layer\\_avgpool, 84](#)
  - [\\_emit\\_aix\\_layer\\_batchnorm, 84](#)
  - [\\_emit\\_aix\\_layer\\_biasadd, 85](#)
  - [\\_emit\\_aix\\_layer\\_convolution, 85](#)
  - [\\_emit\\_aix\\_layer\\_ewadd, 85](#)
  - [\\_emit\\_aix\\_layer\\_group\\_conv, 86](#)
  - [\\_emit\\_aix\\_layer\\_maxpool, 86](#)
  - [\\_emit\\_aix\\_layer\\_softmax, 87](#)
  - [\\_emit\\_aix\\_sampling\\_desc, 87](#)
  - [\\_emit\\_aix\\_tensor\\_bias, 89](#)
  - [\\_emit\\_aix\\_tensor\\_filter, 89](#)
  - [\\_emit\\_aix\\_tensor\\_input, 90](#)
  - [\\_emit\\_aix\\_tensor\\_mean, 90](#)
  - [\\_emit\\_aix\\_tensor\\_output, 90](#)
  - [\\_emit\\_aix\\_tensor\\_scale, 91](#)
  - [\\_emit\\_aix\\_tensor\\_variance, 91](#)
- [block\\_ref](#)
  - [AxfcIRNode::AxfcIRNode, 57](#)
- [blocks](#)
  - [AxfcIRGraph::AxfcIRGraph, 53](#)
- [build\\_ir](#)
  - [AxfcIRBuilder::AxfcIRBuilder, 48](#)
- [compile](#)
  - [AxfcFrontendCompiler::AxfcFrontendCompiler, 35](#)
- [containing\\_type](#)
  - [aixh\\_pb2, 16](#)
- [DEFAULT\\_PROFIT\\_THRESHOLD](#)
  - [AxfcMachineDesc::AxfcMachineDesc, 74](#)
- [DESCRIPTOR](#)
  - [aixh\\_pb2, 17](#)
- [DUMP\\_IR\\_GRAPH\\_ERROR](#)
  - [AxfcError::AxfcError, 29](#)
- [dump\\_aix\\_graphs](#)
  - [AxfcFrontendCompiler::AxfcFrontendCompiler, 35](#)
- [dump\\_launcher](#)
  - [AxfcFrontendCompiler::AxfcFrontendCompiler, 36](#)
- [dump\\_to\\_file](#)
  - [AxfcIRGraph::AxfcIRGraph, 52](#)
- [EMPTY\\_IR\\_BLOCK](#)
  - [AxfcError::AxfcError, 29](#)

- emit\_aixh\_graphs
  - AxfoIRTranslator::AxfoIRTranslator, 67
- emit\_aixh\_launcher
  - AxfoLauncherWriter::AxfoLauncherWriter, 69
- enum\_type
  - aixh\_pb2, 17
- eval\_flag
  - AxfoIRNode::AxfoIRNode, 57
- get\_aixh\_support
  - AxfoMachineDesc::AxfoMachineDesc, 72
- get\_in\_type
  - AxfoMachineDesc::AxfoMachineDesc, 72
- get\_ir\_graph
  - AxfoFrontendCompiler::AxfoFrontendCompiler, 36
- get\_layer\_info
  - AxfoMachineDesc::AxfoMachineDesc, 73
- get\_profit\_threshold
  - AxfoMachineDesc::AxfoMachineDesc, 73
- has\_options
  - aixh\_pb2, 17
- help
  - AxfoMain, 20
- INVALID\_ACTIVATION\_LAYER
  - AxfoError::AxfoError, 29
- INVALID\_AIX\_GRAPH
  - AxfoError::AxfoError, 29
- INVALID\_AIX\_LAYER\_TYPE
  - AxfoError::AxfoError, 30
- INVALID\_AIX\_TENSOR\_FORMAT
  - AxfoError::AxfoError, 30
- INVALID\_AIX\_TENSOR\_INPUT
  - AxfoError::AxfoError, 30
- INVALID\_BATCHNORM\_LAYER
  - AxfoError::AxfoError, 30
- INVALID\_CONVOLUTION\_LAYER
  - AxfoError::AxfoError, 30
- INVALID\_EWADD\_LAYER
  - AxfoError::AxfoError, 30
- INVALID\_FILE\_PATH
  - AxfoError::AxfoError, 31
- INVALID\_GROUP\_CONV\_LAYER
  - AxfoError::AxfoError, 31
- INVALID\_IDENTITY\_LAYER
  - AxfoError::AxfoError, 31
- INVALID\_INPUT\_TYPE
  - AxfoError::AxfoError, 31
- INVALID\_IR\_GRAPH
  - AxfoError::AxfoError, 31
- INVALID\_MAXPOOL\_LAYER
  - AxfoError::AxfoError, 31
- INVALID\_MD\_FORMAT
  - AxfoError::AxfoError, 32
- INVALID\_PAD\_LAYER
  - AxfoError::AxfoError, 32
- INVALID\_PARAMETER
  - AxfoError::AxfoError, 32
- INVALID\_TF\_GRAPH
  - AxfoError::AxfoError, 32
- INVALID\_BIASADD\_LAYER
  - AxfoError::AxfoError, 32
- id
  - AxfoIRBlock::AxfoIRBlock, 44
  - AxfoIRNode::AxfoIRNode, 58
- is\_aixh\_support
  - AxfoIRBlock::AxfoIRBlock, 44
  - AxfoIRNode::AxfoIRNode, 58
- is\_conv
  - AxfoMachineDesc::AxfoMachineDesc::AIXLayer↔  
Info, 24
- is\_group
  - AxfoMachineDesc::AxfoMachineDesc::AIXLayer↔  
Info, 24
- is\_input
  - AxfoIRNode::AxfoIRNode, 58
- is\_output
  - AxfoIRNode::AxfoIRNode, 58
- layer
  - AxfoMachineDesc::AxfoMachineDesc::AIXLayer↔  
Info, 25
- layer\_id
  - AxfoIRNode::AxfoIRNode, 58
- live\_in
  - AxfoIRBlock::AxfoIRBlock, 44
- live\_out
  - AxfoIRBlock::AxfoIRBlock, 44
- message\_type
  - aixh\_pb2, 17
- metavar
  - AxfoMain, 20
- NOT\_AIXH\_SUPPORT
  - AxfoError::AxfoError, 32
- NOT\_IMPLEMENTED
  - AxfoError::AxfoError, 33
- name
  - AxfoIRNode::AxfoIRNode, 59
- node\_def
  - AxfoIRNode::AxfoIRNode, 59
- nodes
  - AxfoIRBlock::AxfoIRBlock, 45
  - AxfoIRGraph::AxfoIRGraph, 53
- op
  - AxfoIRNode::AxfoIRNode, 59
  - AxfoMachineDesc::AxfoMachineDesc::AIXLayer↔  
Info, 25
- PRED\_NODE\_NOT\_FOUND
  - AxfoError::AxfoError, 33
- parser
  - AxfoMain, 20
- preds
  - AxfoIRNode::AxfoIRNode, 59

profit  
    AxfcMachineDesc::AxfcMachineDesc::AIXLayer↔  
        Info, 25

read\_file  
    AxfcMachineDesc::AxfcMachineDesc, 73

read\_md\_file  
    AxfcFrontendCompiler::AxfcFrontendCompiler, 36

required  
    AxfcMain, 20

root\_node  
    AxfcIRGraph::AxfcIRGraph, 53

SUCCESS  
    AxfcError::AxfcError, 33

str  
    AxfcMain, 21

succs  
    AxfcIRNode::AxfcIRNode, 59

symtab  
    AxfcIRGraph::AxfcIRGraph, 53

TYPE\_MXNET  
    AxfcMachineDesc::AxfcMachineDesc, 75

TYPE\_PYTORCH  
    AxfcMachineDesc::AxfcMachineDesc, 75

TYPE\_TENSORFLOW  
    AxfcMachineDesc::AxfcMachineDesc, 75

TYPE\_UNKNOWN  
    AxfcMachineDesc::AxfcMachineDesc, 75

type  
    AxfcMain, 21

UNKNOWN\_TENSOR\_TYPE  
    AxfcError::AxfcError, 33

UNREMOVED\_IDENTITY  
    AxfcError::AxfcError, 33

UNSUPPORTED\_AIX\_LAYER\_EMIT  
    AxfcError::AxfcError, 33

write\_file  
    AxfcGraphWriter::AxfcGraphWriter, 39