



## **IVI-4.10: IviRFSigGen Class Specification**

June 9, 2010 Edition  
Revision 2.0

# Important Information

---

The IviRFSigGen Class Specification (IVI-4.10) is authored by the IVI Foundation member companies. For a vendor membership roster list, please visit the IVI Foundation web site at [www.ivifoundation.org](http://www.ivifoundation.org), or contact the IVI Foundation.

The IVI Foundation wants to receive your comments on this specification. See the IVI Foundation web site for contact information.

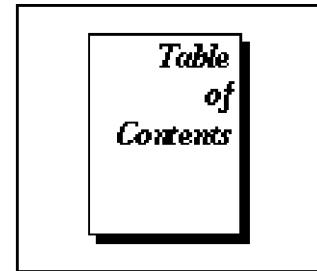
## Warranty

The IVI Foundation and its member companies make no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The IVI Foundation and its member companies shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

## Trademarks

Product and company names listed are trademarks or trade names of their respective companies.

No investigation has been made of common-law trademark rights in any work.



---

<b>1.</b>	<b>Overview of the IviRFSigGen Specification .....</b>	<b>13</b>
1.1	Introduction .....	13
1.2	IviRFSigGen Class Overview .....	13
1.3	References .....	13
1.4	Definitions of Terms and Acronyms .....	14
<b>2.</b>	<b>IviRFSigGen Class Capabilities.....</b>	<b>15</b>
2.1	Introduction .....	15
2.2	IviRFSigGen Group Names .....	15
2.3	Repeated Capability Names .....	16
2.3.1	LFGenerator .....	16
2.3.2	Analog Modulation Source .....	17
2.4	Boolean Attribute and Parameter Values.....	17
2.5	.NET Namespace.....	17
2.6	.NET IviRFSigGen Session Factory.....	17
<b>3.</b>	<b>General Requirements.....</b>	<b>19</b>
3.1	Minimum Class Compliance.....	19
3.1.1	Disable .....	19
3.2	Capability Group Compliance .....	19
<b>4.</b>	<b>IviRFSigGenBase Capabilities Group .....</b>	<b>20</b>
4.1	Overview .....	20
4.2	IviRFSigGenBase Attributes .....	20
4.2.1	Frequency.....	21
4.2.2	Power Level .....	22
4.2.3	ALC Enabled.....	23
4.2.4	Output Enabled.....	24
4.3	IviRFSigGenBase Functions .....	25
4.3.1	Configure RF.....	26
4.3.2	Configure ALC Enabled (IVI-C only) .....	27
4.3.3	Configure Output Enabled (IVI-C only).....	28
4.3.4	Disable All Modulation .....	29
4.3.5	Is Settled .....	30
4.3.6	Wait Until Settled .....	31
4.4	IviRFSigGenBase Behavior Model .....	33
<b>5.</b>	<b>IviRFSigGenModulateAM Extension Group .....</b>	<b>34</b>
5.1	IviRFSigGenModulateAM Extension Group Overview .....	34
5.2	IviRFSigGenModulateAM Attributes .....	34

5.2.1	AM Enabled .....	35
5.2.2	AM Source .....	36
5.2.3	AM Scaling .....	37
5.2.4	AM External Coupling .....	39
5.2.5	AM Nominal Voltage .....	41
5.2.6	AM Depth .....	42
5.3	IviRFSigGenModulateAM Functions .....	43
5.3.1	Configure AM Enabled (IVI-C only) .....	44
5.3.2	Configure AM External Coupling (IVI-C only) .....	45
5.3.3	Configure AM .....	46
5.4	IviRFSigGenModulateAM Behavior Model .....	47
5.5	IviRFSigGenModulateAM Compliance Notes .....	47

## 6. IviRFSigGenModulateFM Extension Group..... 48

6.1	IviRFSigGenModulateFM Extension Group Overview .....	48
6.2	IviRFSigGenModulateFM Attributes .....	48
6.2.1	FM Enabled .....	49
6.2.2	FM Source .....	50
6.2.3	FM External Coupling .....	51
6.2.4	FM Nominal Voltage .....	53
6.2.5	FM Deviation .....	54
6.3	IviRFSigGenModulateFM Functions .....	55
6.3.1	Configure FM Enabled (IVI-C only) .....	56
6.3.2	Configure FM External Coupling (IVI-C only) .....	57
6.3.3	Configure FM .....	58
6.4	IviRFSigGenModulateFM Behavior Model .....	59
6.5	IviRFSigGenModulateFM Compliance Notes .....	59

## 7. IviRFSigGenModulatePM Extension Group..... 60

7.1	IviRFSigGenModulatePM Extension Group Overview .....	60
7.2	IviRFSigGenModulatePM Attributes .....	60
7.2.1	PM Enabled .....	61
7.2.2	PM Source .....	62
7.2.3	PM External Coupling .....	63
7.2.4	PM Nominal Voltage .....	65
7.2.5	PM Deviation .....	66
7.3	IviRFSigGenModulatePM Functions .....	67
7.3.1	Configure PM Enabled (IVI-C only) .....	68
7.3.2	Configure PM External Coupling (IVI-C only) .....	69
7.3.3	Configure PM .....	70
7.4	IviRFSigGenModulatePM Behavior Model .....	71
7.5	IviRFSigGenModulatePM Compliance Notes .....	71

## 8. IviRFSigGenAnalogModulationSource Extension Group 72

8.1	IviRFSigGenAnalogModulationSource Extension Group Overview .....	72
8.2	IviRFSigGenAnalogModulationSource Attributes .....	72
8.2.1	Analog Modulation Source Count .....	73
8.2.2	Analog Modulation Source Name (IVI-COM Only) .....	74
8.3	IviRFSigGenAnalogModulationSource Functions .....	75
8.3.1	Get Analog Modulation Source Name (IVI-C & IVI.NET only) .....	76
8.4	IviRFSigGenAnalogModulationSource Behavior Model .....	78

8.5	IviRFSigGenAnalogModulationSource Compliance Notes .....	78
-----	--	----

## **9. IviRFSigGenModulatePulse Extension Group ..... 79**

9.1	IviRFSigGenModulatePulse Extension Group Overview .....	79
9.2	IviRFSigGenModulatePulse Attributes .....	79
9.2.1	Pulse Modulation Enabled .....	80
9.2.2	Pulse Modulation Source .....	81
9.2.3	Pulse Modulation External Polarity .....	83
9.3	IviRFSigGenModulatePulse Functions .....	85
9.3.1	Configure Pulse Modulation Enabled (IVI-C only) .....	86
9.3.2	Configure Pulse Modulation Source (IVI-C only) .....	87
9.3.3	Configure Pulse Modulation External Polarity (IVI-C only) .....	88
9.4	IviRFSigGenModulatePulse Behavior Model .....	89
9.5	IviRFSigGenModulatePulse Compliance Notes .....	89

## **10. IviRFSigGenLFGenerator Extension Group ..... 90**

10.1	IviRFSigGenLFGenerator Extension Group Overview .....	90
10.2	IviRFSigGenLFGenerator Attributes .....	90
10.2.1	Active LFGenerator .....	91
10.2.2	LFGenerator Count .....	92
10.2.3	LFGenerator Name (IVI-COM Only) .....	93
10.2.4	LFGenerator Frequency .....	94
10.2.5	LFGenerator Waveform .....	95
10.3	IviRFSigGenLFGenerator Functions .....	97
10.3.1	Get LFGenerator Name (IVI-C & IVI.NET only) .....	98
10.3.2	Set Active LFGenerator (IVI-C Only) .....	99
10.3.3	Configure LFGenerator .....	100
10.4	IviRFSigGenLFGenerator Behavior Model .....	101
10.5	IviRFSigGenLFGenerator Compliance Notes .....	101

## **11. IviRFSigGen LFGeneratorOutput Extension Group .... 102**

11.1	IviRFSigGenLFGeneratorOutput Extension Group Overview .....	102
11.2	IviRFSigGenLFGeneratorOutput Attributes .....	102
11.2.1	LFGenerator Output Amplitude .....	103
11.2.2	LFGenerator Output Enabled .....	104
11.3	IviRFSigGenLFGeneratorOutput Functions .....	105
11.3.1	Configure LFGenerator Output .....	106
11.4	IviRFSigGenLFGeneratorOutput Behavior Model .....	107
11.5	IviRFSigGenLFGeneratorOutput Compliance Notes .....	107

## **12. IviRFSigGenPulseGenerator Extension Group ..... 108**

12.1	IviRFSigGenPulseGenerator Extension Group Overview .....	108
12.2	IviRFSigGenPulseGenerator Attributes .....	108
12.2.1	Pulse Internal Trigger Period .....	109
12.2.2	Pulse Width .....	110
12.2.3	Pulse Gating Enabled .....	111
12.2.4	Pulse Trigger Source .....	112
12.2.5	Pulse External Trigger Slope .....	114
12.2.6	Pulse External Trigger Delay .....	116
12.3	IviRFSigGenPulseGenerator Functions .....	117
12.3.1	Configure Pulse External Trigger .....	118

12.3.2	Configure Pulse Internal Trigger (IVI-C only).....	119
12.3.3	Configure Pulse .....	120
12.4	IviRFSigGenPulseGenerator Behavior Model .....	121
12.5	IviRFSigGenPulseGenerator Compliance Notes.....	121

### **13. IviRFSigGenPulseDoubleGenerator Extension Group 122**

13.1	IviRFSigGenPulseDoubleGenerator Extension Group Overview .....	122
13.2	IviRFSigGenPulseDoubleGenerator Attributes .....	122
13.2.1	Pulse Double Enabled .....	123
13.2.2	Pulse Double Delay.....	124
13.3	IviRFSigGenPulseDoubleGenerator Functions .....	125
13.3.1	Configure Pulse Double .....	126
13.4	IviRFSigGenPulseDoubleGenerator Behavior Model .....	127
13.5	IviRFSigGenPulseDoubleGenerator Compliance Notes.....	127

### **14. IviRFSigGenPulseGeneratorOutput Extension Group 128**

14.1	IviRFSigGenPulseGeneratorOutput Extension Group Overview .....	128
14.2	IviRFSigGenPulseGeneratorOutput Attributes .....	128
14.2.1	Pulse Output Polarity.....	129
14.2.2	Pulse Output Enabled .....	131
14.3	IviRFSigGenPulseGeneratorOutput Functions .....	132
14.3.1	Configure Pulse Output .....	133
14.4	IviRFSigGenPulseGeneratorOutput Behavior Mode .....	134
14.5	IviRFSigGenPulseGeneratorOutput Compliance Notes .....	134

### **15. IviRFSigGenSweep Extension Group..... 135**

15.1	IviRFSigGenSweep Overview .....	135
15.2	IviRFSigGenSweep Attributes .....	135
15.2.1	Sweep Mode .....	136
15.2.2	Sweep Trigger Source .....	139
15.3	IviRFSigGenSweep Functions .....	141
15.3.1	Configure Sweep .....	142
15.4	IviRFSigGenSweep Behavior Model.....	143
15.4.1	Frequency Sweep / Power Sweep .....	143
15.4.2	Frequency Step / Power Step / List .....	144

### **16. IviRFSigGenFrequencySweep Extension Group ..... 146**

16.1	IviRFSigGenFrequencySweep Extension Group Overview .....	146
16.2	IviRFSigGenFrequencySweep Attributes .....	146
16.2.1	Frequency Sweep Start .....	147
16.2.2	Frequency Sweep Stop .....	148
16.2.3	Frequency Sweep Time.....	149
16.3	IviRFSigGenFrequencySweep Functions .....	150
16.3.1	Configure Frequency Sweep Start Stop.....	151
16.3.2	Configure Frequency Sweep Center Span .....	152
16.3.3	Configure Frequency Sweep Time (IVI-C only) .....	153
16.4	IviRFSigGenFrequencySweep Behavior Model.....	154
16.5	IviRFSigGenFrequencySweep Compliance Notes .....	154

<b>17.</b>	<b>IviRFSigGenPowerSweep Extension Group .....</b>	<b>155</b>
17.1	IviRFSigGenPowerSweep Extension Group Overview .....	155
17.2	IviRFSigGenPowerSweep Attributes .....	155
17.2.1	Power Sweep Start .....	156
17.2.2	Power Sweep Stop .....	157
17.2.3	Power Sweep Time.....	158
17.3	IviRFSigGenPowerSweep Functions .....	159
17.3.1	Configure Power Sweep Start Stop.....	160
17.3.2	Configure Power Sweep Time (IVI-C only).....	161
17.4	IviRFSigGenPowerSweep Behavior Model.....	162
17.5	IviRFSigGenPowerSweep Compliance Notes .....	162
<b>18.</b>	<b>IviRFSigGenFrequencyStep Extension Group.....</b>	<b>163</b>
18.1	IviRFSigGenFrequencyStep Extension Group Overview .....	163
18.2	IviRFSigGenFrequencyStep Attributes .....	163
18.2.1	Frequency Step Start .....	164
18.2.2	Frequency Step Stop.....	165
18.2.3	Frequency Step Scaling.....	166
18.2.4	Frequency Step Size .....	168
18.2.5	Frequency Step Single Step Enabled .....	169
18.2.6	Frequency Step Dwell.....	170
18.3	IviRFSigGenFrequencyStep Functions.....	171
18.3.1	Configure Frequency Step Start Stop.....	172
18.3.2	Configure Frequency Step Dwell .....	173
18.3.3	Reset Frequency Step .....	174
18.4	IviRFSigGenFrequencyStep Behavior Model.....	175
18.5	IviRFSigGenFrequencyStep Compliance Notes.....	175
<b>19.</b>	<b>IviRFSigGenPowerStep Extension Group .....</b>	<b>176</b>
19.1	IviRFSigGenPowerStep Extension Group Overview .....	176
19.2	IviRFSigGenPowerStep Attributes .....	176
19.2.1	Power Step Start .....	177
19.2.2	Power Step Stop.....	178
19.2.3	Power Step Size .....	179
19.2.4	Power Step Single Step Enabled .....	180
19.2.5	Power Step Dwell.....	181
19.3	IviRFSigGenPowerStep Functions.....	182
19.3.1	Configure Power Step Start Stop.....	183
19.3.2	Configure Power Step Dwell .....	184
19.3.3	Reset Power Step .....	185
19.4	IviRFSigGenPowerStep Behavior Model.....	186
19.5	IviRFSigGenPowerStep Compliance Notes.....	186
<b>20.</b>	<b>IviRFSigGenList Extension Group.....</b>	<b>187</b>
20.1	IviRFSigGenList Extension Group Overview.....	187
20.2	IviRFSigGenList Attributes .....	187
20.2.1	List Selected Name .....	188
20.2.2	List Single Step Enabled .....	189
20.2.3	List Dwell.....	190
20.3	IviRFSigGenList Functions.....	191
20.3.1	Create Frequency List .....	192
20.3.2	Create Power List.....	193

20.3.3	Create Frequency Power List .....	194
20.3.4	Select List (IVI-C only) .....	195
20.3.5	Clear All Lists .....	196
20.3.6	Configure List Dwell .....	197
20.3.7	Reset List .....	198
20.4	IviRFSigGenList Behavior Model .....	199
20.5	IviRFSigGenList Compliance Notes .....	199
<b>21.</b>	<b>IviRFSigGenALC Extension Group .....</b>	<b>200</b>
21.1	IviRFSigGenALC Overview .....	200
21.2	IviRFSigGenALC Attributes .....	200
21.2.1	ALC Source .....	201
21.2.2	ALC Bandwidth .....	203
21.3	IviRFSigGenALC Functions .....	204
21.3.1	Configure ALC .....	205
21.4	IviRFSigGenALC Behavior Model .....	206
<b>22.</b>	<b>IviRFSigGen ReferenceOscillator Extension Group ....</b>	<b>207</b>
22.1	IviRFSigGenReferenceOscillator Overview .....	207
22.2	IviRFSigGenReferenceOscillator Attributes .....	207
22.2.1	Reference Oscillator Source .....	208
22.2.2	Reference Oscillator External Frequency .....	210
22.3	IviRFSigGenReferenceOscillator Functions .....	211
22.3.1	Configure Reference Oscillator .....	212
22.4	IviRFSigGenReferenceOscillator Behavior Model .....	213
<b>23.</b>	<b>IviRFSigGenSoftwareTrigger Extension Group .....</b>	<b>214</b>
23.1	IviRFSigGenSoftwareTrigger Overview .....	214
23.2	IviRFSigGenSoftwareTrigger Functions .....	214
23.2.1	Send Software Trigger .....	214
23.3	IviRFSigGenSoftwareTrigger Behavior Model .....	214
23.4	IviRFSigGenSoftwareTrigger Compliance Notes .....	214
<b>24.</b>	<b>IviRFSigGenModulateIQ Extension Group .....</b>	<b>215</b>
24.1	IviRFSigGenModulateIQ Extension Group Overview .....	215
24.2	IviRFSigGenModulateIQ Attributes .....	215
24.2.1	IQ Enabled .....	216
24.2.2	IQ Nominal Voltage .....	217
24.2.3	IQ Source .....	218
24.2.4	IQ Swap Enabled .....	220
24.3	IviRFSigGenModulateIQ Functions .....	221
24.3.1	Configure IQ Enabled (IVI-C only) .....	222
24.3.2	Configure IQ .....	223
24.3.3	Calibrate IQ .....	224
24.4	IviRFSigGenModulateIQ Behavior Model .....	225
<b>25.</b>	<b>IviRFSigGenIQImpairment Extension Group .....</b>	<b>226</b>
25.1	IviRFSigGenIQImpairment Extension Group Overview .....	226
25.2	IviRFSigGenIQImpairment Attributes .....	226
25.2.1	IQ Impairment Enabled .....	227



25.2.2	IQ I-Offset.....	228
25.2.3	IQ Q-Offset .....	229
25.2.4	IQ Ratio .....	230
25.2.5	IQ Skew .....	231
25.3	IviRFSigGenIQImpairment Functions .....	232
25.3.1	Configure IQ Impairment Enabled (IVI-C only) .....	233
25.3.2	Configure IQ Impairment.....	234
25.4	IviRFSigGenIQImpairment Behavior Model .....	235
25.5	IviRFSigGenIQImpairment Compliance Notes.....	235

## **26. IviRFSigGenArbGenerator Extension Group ..... 236**

26.1	IviRFSigGenArbGenerator Extension Group Overview .....	236
26.2	IviRFSigGenArbGenerator Attributes.....	236
26.2.1	Arb Selected Waveform.....	237
26.2.2	Arb Clock Frequency.....	238
26.2.3	Arb Filter Frequency.....	239
26.2.4	Arb Max Number Waveforms.....	240
26.2.5	Arb Waveform Quantum.....	241
26.2.6	Arb Waveform Size Min .....	242
26.2.7	Arb Waveform Size Max .....	243
26.2.8	Arb Trigger Source.....	244
26.2.9	Arb External Trigger Slope.....	246
26.3	IviRFSigGenArbGenerator Functions .....	248
26.3.1	Configure Arb.....	249
26.3.2	Write Arb Waveform .....	250
26.3.3	Select Arb Waveform (IVI-C only) .....	251
26.3.4	Clear All Arb Waveforms.....	252
26.3.5	Query Arb Waveform Capabilities (IVI-C and IVI-COM Only).....	253
26.3.6	Configure Arb Trigger Source (IVI-C only).....	254
26.3.7	Configure Arb External Trigger Slope (IVI-C only) .....	255
26.4	IviRFSigGenArbGenerator Behavior Model .....	256
26.5	IviRFSigGenArbGenerator Compliance Notes .....	256

## **27. IviRFSigGenDigitalModulationBase Extension Group 257**

27.1	IviRFSigGenDigitalModulationBase Extension Group Overview .....	257
27.2	IviRFSigGenDigitalModulationBase Attributes.....	257
27.2.1	DigitalModulationBase Standard Names (.NET only) .....	258
27.2.2	DigitalModulationBase Standard Name (IVI-COM only).....	259
27.2.3	DigitalModulationBase Standard Count (IVI-C and IVI-COM only).....	260
27.2.4	DigitalModulationBase Selected Standard.....	261
27.2.5	DigitalModulationBase Data Source .....	262
27.2.6	DigitalModulationBase PRBS Type .....	264
27.2.7	DigitalModulationBase Selected Bit Sequence .....	266
27.2.8	DigitalModulationBase Clock Source .....	267
27.2.9	DigitalModulationBase External Clock Type.....	269
27.3	IviRFSigGenDigitalModulationBase Functions .....	271
27.3.1	Get DigitalModulationBase Standard Name (IVI-C only).....	272
27.3.2	Select DigitalModulationBase Standard (IVI-C only).....	273
27.3.3	Configure DigitalModulationBase Clock Source .....	274
27.3.4	Configure DigitalModulationBase Data Source (IVI-C only) .....	275
27.3.5	Configure DigitalModulationBase PRBS Type (IVI-C only) .....	276
27.3.6	Write DigitalModulationBase Bit Sequence .....	277
27.3.7	Select DigitalModulationBase Bit Sequence (IVI-C only) .....	278

27.3.8	Clear All DigitalModulationBase Bit Sequences .....	279
27.4	IviRFSigGenDigitalModulationBase Behavior Model .....	280
27.5	IviRFSigGenDigitalModulationBase Compliance Notes .....	280

## **28. IviRFSigGenCDMABase Extension Group ..... 281**

28.1	IviRFSigGenCDMABase Extension Group Overview .....	281
28.2	IviRFSigGenCDMABase Attributes .....	281
28.2.1	CDMA Standard Names (IVI.NET only) .....	282
28.2.2	CDMA Standard Name (IVI-COM only) .....	283
28.2.3	CDMA Standard Count (IVI-C and IVI-COM only) .....	284
28.2.4	CDMA Selected Standard .....	285
28.2.5	CDMA Trigger Source .....	286
28.2.6	CDMA External Trigger Slope .....	288
28.2.7	CDMA Test Model Names (IVI.NET only) .....	290
28.2.8	CDMA Test Model Name (IVI-COM only) .....	291
28.2.9	CDMA Test Model Count (IVI-C and IVI-COM only) .....	292
28.2.10	CDMA Selected Test Model .....	293
28.2.11	CDMA Clock Source .....	294
28.3	IviRFSigGenCDMABase Functions .....	296
28.3.1	Get CDMA Standard Name (IVI-C only) .....	297
28.3.2	Select CDMA Standard (IVI-C only) .....	298
28.3.3	Configure CDMA Clock Source (IVI-C only) .....	299
28.3.4	Configure CDMA Trigger Source (IVI-C only) .....	300
28.3.5	Configure CDMA External Trigger Slope (IVI-C only) .....	301
28.3.6	Get CDMA Test Model Name (IVI-C only) .....	302
28.3.7	Select CDMA Test Model (IVI-C only) .....	303
28.4	IviRFSigGenCDMABase Behavior Model .....	304
28.5	IviRFSigGenCDMABase Compliance Notes .....	304

## **29. IviRFSigGenTDMABase Extension Group ..... 305**

29.1	IviRFSigGenTDMABase Extension Group Overview .....	305
29.2	IviRFSigGenTDMABase Attributes .....	305
29.2.1	TDMA Standard Names (IVI.NET only) .....	306
29.2.2	TDMA Standard Name (IVI-COM only) .....	307
29.2.3	TDMA Standard Count (IVI-C and IVI-COM only) .....	308
29.2.4	TDMA Selected Standard .....	309
29.2.5	TDMA Trigger Source .....	310
29.2.6	TDMA External Trigger Slope .....	312
29.2.7	TDMA Frame Names (IVI.NET only) .....	314
29.2.8	TDMA Frame Name (IVI-COM only) .....	315
29.2.9	TDMA Frame Count (IVI-C and IVI-COM only) .....	316
29.2.10	TDMA Selected Frame .....	317
29.2.11	TDMA Clock Source .....	318
29.2.12	TDMA External Clock Type .....	320
29.3	IviRFSigGenTDMABase Functions .....	322
29.3.1	Get TDMA Standard Name (IVI-C only) .....	323
29.3.2	Select TDMA Standard (IVI-C only) .....	324
29.3.3	Configure TDMA Clock Source .....	325
29.3.4	Configure TDMA Trigger Source (IVI-C only) .....	326
29.3.5	Configure TDMA External Trigger Slope (IVI-C only) .....	327
29.3.6	Get TDMA Frame Name (IVI-C only) .....	328
29.3.7	Select TDMA Frame (IVI-C only) .....	329
29.4	IviRFSigGenTDMABase Behavior Model .....	330

29.5	IviRFSigGenTDMABase Compliance Notes .....	330
<b>30.</b>	<b>IviRFSigGen Attribute ID Definitions .....</b>	<b>331</b>
<b>31.</b>	<b>IviRFSigGen Attribute Value Definitions .....</b>	<b>335</b>
<b>32.</b>	<b>IviRFSigGen Function Parameter Value Definitions....</b>	<b>350</b>
<b>33.</b>	<b>IviRFSigGen Error and Completion Code Value</b>	
<b>Definitions</b>	<b>351</b>	
33.1	IVI.NET IviRFSigGen Exceptions and Warnings.....	352
33.1.1	ListUnknownException .....	352
<b>34.</b>	<b>IviRFSigGen Hierarchies .....</b>	<b>353</b>
34.1	IviRFSigGen .NET Hierarchy .....	353
34.1.1	IviRFSigGen COM Interfaces .....	359
34.1.2	.NET Interface Reference Properties .....	360
34.2	IviRFSigGen COM Hierarchy.....	361
34.2.1	IviRFSigGen COM Interfaces .....	367
34.2.2	COM Interface Reference Properties .....	369
34.2.3	IviRFSigGen COM Category .....	369
34.3	IviRFSigGen C Function Hierarchy.....	370
34.4	IviRFSigGen C Attribute Hierarchy .....	373
<b>Appendix A</b>	<b>Specific Driver Development Guidelines .....</b>	<b>378</b>
A.1	Introduction .....	378
A.2	Disabling Unused Extensions .....	378
A.3	Query Instrument Status .....	383
<b>Appendix B</b>	<b>Interchangeability Checking Rules.....</b>	<b>384</b>
B.1	Introduction .....	384
B.2	When to Perform Interchangeability Checking .....	384
B.3	Interchangeability Checking Rules .....	384

# IviRFSigGen Class Specification

---

## IviRFSigGen Revision History

---

This section is an overview of the revision history of the IviRFSigGen specification.

**Table 1.** IviRFSigGen Class Specification Revisions

Revision Number	Date of Revision	Revision Notes
Revision 1.0	March 2002	Released version 1.0 of the spec
Revision 1.1	February 2007	Editorial changes: Correction of misspellings on pages 32, 80,101, 107, 251, 266, 309, and 357 . Interchangeability checking rules for AM, FM and PM Source corrected
Revision 1.1	October 2008	Editorial change to update the IVI Foundation contact information in the Important Information section to remove obsolete address information and refer only to the IVI Foundation web site.
Revision 1.1	April 2009	Editorial change to update repeated capabilities section to include both qualified and unqualified repeated capability names.
Revision 2.0	June 9, 2010	Incorporated IVI.NET

### API Versions

Architecture	Drivers that comply with version 2.0 comply with all of the versions below.
C	1.0, 1.1, 2.0
COM	1.0, 1.1, 2.0
.NET	2.0

Drivers that comply with this version of the specification also comply with earlier, compatible versions of the specification as shown in the table above. The driver may benefit by advertising that it supports all the API versions listed in the table above.

# 1. Overview of the IviRFSigGen Specification

---

## 1.1 Introduction

This specification defines the IVI class for RF signal generators. The IviRFSigGen class is designed to support the typical RF signal generator as well as common extended functionality found in more complex instruments. This section summarizes the *IviRFSigGen Specification* itself and contains general information that the reader may need in order to understand, interpret, and implement aspects of this specification. These aspects include the following:

- IviRFSigGen Class Overview
- References
- Definitions of Terms and Acronyms

## 1.2 IviRFSigGen Class Overview

This specification defines the IVI class for RF signal generators (RF SigGens). The IviRFSigGen class is designed to support the typical RF signal generator as well as common extended functionality found in more complex instruments. The IviRFSigGen class conceptualizes a RF signal generator as an instrument that can generate a sinusoidal carrier waveform with a wide variety of modulation types.

The IviRFSigGen class is divided into a base capability group and several extension groups. The base capability group is used to configure the carrier waveform. This includes setting the output frequency and amplitude, enabling or disabling the ALC and the RF output. The IviRFSigGen base capability group is described in Section 4, *IviRFSigGenBase Capability Group*.

The extension groups for analog and pulse modulation are IviRFSigGenModulateAM, IviRFSigGenModulateFM, IviRFSigGenModulatePM, IviRFSigGenAnalogModulationSource, IviRFSigGenModulatePulse, IviRFSigGenLFGenerator, IviRFSigGenLFGeneratorOutput, IviRFSigGenPulseGenerator, IviRFSigGenPulseDoubleGenerator and IviRFSigGenPulseGeneratorOutput.

The extension groups for sweeping are IviRFSigGenSweep, IviRFSigGenFrequencySweep, IviRFSigGenPowerSweep, IviRFSigGenFrequencyStep, IviRFSigGenPowerStep and IviRFSigGenList.

The extension groups for digital modulation are IviRFSigGenModulateIQ, IviRFSigGenIQImpairments, IviRFSigGenArbGeneration, IviRFSigGenDigitalModulationBase, IviRFSigGenCDMABase and IviRFSigGenTDMABase.

The remaining extension groups are IviRFSigGenALC, IviRFSigGenReferenceOscillator and IviRFSigGenSoftwareTrigger.

## 1.3 References

Several other documents and specifications are related to this specification. These other related documents are the following:

- IVI-3.1: Architecture Overview Specification
- IVI-3.2: Inherent Capabilities Specification
- IVI-3.3: Standard Cross Class Capabilities
- IVI-3.18: IVI.NET Utility Classes and Interfaces Specification
- IVI-5.0: Glossary

- CCITT-V.52: Characteristics of distortion and error-rate measuring apparatus for data transmission

## 1.4 **Definitions of Terms and Acronyms**

This section defines terms and acronyms that are specific to the IviRFSigGen class. Terms of more general interest are defined in *IVI-5.0: Glossary*.

ALC	Automatic Level Control, Used for controlling the RF SigGens output signal on a constant amplitude.
TDMA	Time Division Multiple Access
CDMA	Code Division Multiple Access
LF	Low Frequency
AM	Amplitude Modulation
FM	Frequency Modulation
PM	Phase Modulation
PRBS	Pseudo Random Bit Sequence
ARB	Arbitrary (Waveform)

## 2. IviRFSigGen Class Capabilities

---

### 2.1 Introduction

The IviRFSigGen specification divides signal generator capabilities into a base capabilities group and multiple extension capabilities groups. Each capability group is discussed in a separate section. This section defines names for each capability group and gives an overview of the information presented in each section.

### 2.2 IviRFSigGen Group Names

The capability group names for the IviRFSigGen class are defined in the following table. The Group Name is used to represent a particular capability group and is returned as one of the possible group names from the IviRFSigGenGroupCapabilities attribute, see *IVI-3.3: Standard Cross Class Capabilities Specification*.

Table 2-1. IviRFSigGen Group Names

Group Name	Description
IviRFSigGenBaseCapabilities	Base Capabilities: For generators that at minimum support setting the output frequency and amplitude, enabling or disabling the ALC and the RF output.
IviRFSigGenModulateAM	Extension Group: For generators with the ability to apply amplitude modulation to an output signal.
IviRFSigGenModulateFM	Extension Group: For generators with the ability to apply frequency modulation to an output signal.
IviRFSigGenModulatePM	Extension Group: For generators with the ability to apply phase modulation to an output signal.
IviRFSigGenAnalogModulationSource	Extension Group: For generators with at least one modulation source.
IviRFSigGenModulatePulse	Extension Group: For generators with the ability to apply pulse modulation to an output signal.
IviRFSigGenLFGenerator	Extension Group: For generators with an internal LF generator.
IviRFSigGenLFGeneratorOutput	Extension Group: For generators with an internal LF generator and a configurable output.
IviRFSigGenPulseGenerator	Extension Group: For generators with an internal pulse generator.
IviRFSigGenPulseDoubleGenerator	Extension Group: For generators with an internal pulse generator and the ability to generate a double pulse.
IviRFSigGenPulseGeneratorOutput	Extension Group: For generators with an internal pulse generator and a configurable output.
IviRFSigGenSweep	Extension Group: For generators with the ability to sweep the output frequency and/or power by analog means, discrete steps, or a list of values.
IviRFSigGenFrequencySweep	Extension Group: For generators with the ability to sweep the output frequency.

Table 2-1. IviRFSigGen Group Names

IviRFSigGenPowerSweep	Extension Group: For generators with the ability to sweep the output power.
IviRFSigGenFrequencyStep	Extension Group: For generators with the ability to sweep the output frequency in discrete steps.
IviRFSigGenPowerStep	Extension Group: For generators with the ability to sweep the output power in discrete steps.
IviRFSigGenList	Extension Group: For generators with the ability to sweep the output frequency and/or power by alist of values.
IviRFSigGenALC	Extension Group: For generators with the ability to use an automatic level control.
IviRFSigGenReferenceOscillator	Extension Group: For generators with the ability to use an external reference frequency.
IviRFSigGenSoftwareTrigger	Extension Group: For generators with the ability to trigger signal generation.
IviRFSigGenModulateIQ	Extension Group: For generators with the ability to apply vector (IQ) modulation to an output signal.
IviRFSigGenIQImpairment	Extension Group: For generators with the ability to apply impairment to vector (IQ) modulation.
IviRFSigGenArbGenerator	Extension Group: For generators with an arbitrary waveform generator as source for vector (IQ) modulation.
IviRFSigGenDigitalModulationBase	Extension Group: For generators with the ability to apply basic Digital Modulation
IviRFSigGenCDMABase	Extension Group: For generators with the ability to apply digital CDMA (Code Division Multiple Access) modulation.
IviRFSigGenTDMABase	Extension Group: For generators with the ability to apply digital TDMA (Time Division Multiple Access) modulation.

## 2.3 Repeated Capability Names

The IviRFSigGen Class Specification defines two repeated capabilities. Refer to the sections of *IVI-3.1, Driver Architecture Specification* that deal with repeated capabilities. The relevant sections are Section 2.7, *Repeated Capabilities*, Section 4.1.9, *Repeated Capabilities*, Section 4.2.5, *Repeated Capabilities*, Section 4.3.9, *Repeated Capabilities*, and Section 5.9, *Repeated Capability Identifiers and Selectors*.

- LFGGenerator
- Analog Modulation Source

### 2.3.1 LFGGenerator

In the configuration store, the name for the LFGGenerator repeated capability shall be exactly one of "LFGGenerator" or "IviRFSigGenLFGGenerator". Drivers that implement multiple repeated capabilities with the name "LFGGenerator" shall use the latter form to disambiguate the names. The LFGGenerator capability is used



by the IviRFSigGenLFGenerator and IviRFSigGenLFGeneratorOutput extension groups and shall be available only if those groups are implemented.

When using the LFGenerator capability, it is necessary to set the Active LFGenerator attribute to an active repeated capability identifier before calling LFGenerator functions. All attributes and functions in the IviRFSigGenLFGenerator and IviRFSigGenLFGeneratorOutput extension groups operate on the repeated capability instance value set by the Active LFGenerator attribute.

### 2.3.2 Analog Modulation Source

In the configuration store, the name for the analog modulation source repeated capability shall be exactly one of “AnalogModulationSource” or “IviRFSigGenAnalogModulationSource”. Drivers that implement multiple repeated capabilities with the name “AnalogModulationSource” shall use the latter form to disambiguate the names.

The Analog Modulation Source capability is used by the IviRFSigGenAnalogModulationSource, IviRFSigGenModulateAM, IviRFSigGenModulateFM, and IviRFSigGenModulatePM extension groups and shall be available only if the IviRFSigGenAnalogModulationSource group is implemented.

## 2.4 Boolean Attribute and Parameter Values

This specification uses True and False as the values for Boolean attributes and parameters. The following table defines the identifiers that are used for True and False in the IVI.NET, IVI-COM, and IVI-C architectures.

Boolean Value	IVI.NET Identifier	IVI-COM Identifier	IVI-C Identifier
True	true	VARIANT_TRUE	VI_TRUE
False	false	VARIANT_FALSE	VI_FALSE

## 2.5 .NET Namespace

The .NET namespace for the IviRFSigGen class is `Ivi.RFSigGen`.

## 2.6 .NET IviRFSigGen Session Factory

The IviRFSigGen .NET assembly contains a factory method called `Create` for creating instances of IviRFSigGen class-compliant IVI.NET drivers from driver sessions and logical names. `Create` is a static method accessible from the static IviRFSigGen class.

Refer to *IVI-3.5: Configuration Server Specification* for a description of how logical names and session names are defined in the configuration store.

Refer to Section 8, *IVI.NET Specific Driver Constructor*, of *IVI-3.2: Inherent Capabilities Specification*, for more details on how the `idQuery`, `reset`, and `options` parameters affect the instantiation of the driver.

### .NET Method Prototype

```
IIviRFSigGen IviRFSigGen.Create(String name);
IIviRFSigGen IviRFSigGen.Create(String name,
                                Boolean idQuery,
                                Boolean reset);
IIviRFSigGen IviRFSigGen.Create(String name,
```

```
Boolean idQuery,
Boolean reset,
String options);
```

## Parameters

Inputs	Description	Base Type
name	A session name or a logical name that points to a session that uses an IVI.NET IviRFSigGen class-compliant driver.	String
idQuery	Specifies whether to verify the ID of the instrument. The default is False.	Boolean
reset	Specifies whether to reset the instrument. The default is False.	Boolean
options	A string that allows the user to specify the initial values of certain inherent attributes. The default is an empty string.	String

Outputs	Description	Base Type
Return Value	Interface pointer to the IIVI.RFSigGen interface of the driver referenced by session.	IIviRFSigGen

## .NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

## Usage

To create a driver that implements the IviRFSigGen instrument class API from the logical name “My LogicalName”, use the following:

```
IIviRFSigGen rfSigGen = IviRFSigGen.Create("MyLogicalName");
```

In this case, the ID of the instrument will not be verified, the instrument will not be reset, and options will be supplied from the configuration store and/or driver defaults.

## 3. General Requirements

---

This section describes the general requirements a specific driver shall meet in order to be compliant with this specification. In addition, it provides general requirements that specific drivers shall meet in order to comply with a capability group, attribute, or function.

### 3.1 **Minimum Class Compliance**

To be compliant with the IviRFSigGen Class Specification, an IVI specific driver shall conform to all of the requirements for an IVI class-compliant specific driver specified in *IVI-3.1: Driver Architecture Specification*. In addition it shall implement the inherent capabilities that *IVI-3.2: Inherent Capabilities Specification* defines, and the IviRFSigGenBase capability group.

#### 3.1.1 Disable

Refer to *IVI-3.2: Inherent Capabilities Specification* for the prototype of this function.

The Disable function shall cause the Signal Generator to apply the minimum amount of power possible at the output terminals. Setting the power level to a very small value or physically disconnecting the signal generator from the output terminals meets this requirement. Other techniques are also allowed.

### 3.2 **Capability Group Compliance**

*IVI-3.1: Architecture Overview Specification* defines the general rules for a specific driver to be compliant with a capability group.

## 4. IviRFSigGenBase Capabilities Group

---

### 4.1 Overview

The IviRFSigGenBase capabilities group supports the basic signal generator capabilities.

### 4.2 *IviRFSigGenBase Attributes*

The IviRFSigGenBase capabilities group defines the following attributes:

- Frequency
- Power Level
- ALC Enabled
- Output Enabled

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 30, *IviRFSigGen Attribute ID Definitions*.

### 4.2.1 Frequency

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	Configure RF

#### .NET Property Name

RF.Frequency

#### COM Property Name

RF.Frequency

#### C Constant Name

IVIRFSIGGEN\_ATTR\_FREQUENCY

#### Description

Specifies the frequency of the generated RF output signal. The units are Hertz.

#### .NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

### 4.2.2 Power Level

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	Configure RF

#### .NET Property Name

`RF.Level`

#### COM Property Name

`RF.Level`

#### C Constant Name

`IVIRFSIGGEN_ATTR_POWER_LEVEL`

#### Description

Specifies the power level of the RF output signal. The units are dBm.

#### .NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

### 4.2.3 ALC Enabled

Data Type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	Configure ALC Enabled

#### .NET Property Name

`Alc.Enabled`

#### COM Property Name

`ALC.Enabled`

#### C Constant Name

`IVIRFSIGGEN_ATTR_ALC_ENABLED`

#### Description

If set to True, enables Automatic Level Control (ALC). If set to False, disables Automatic Level Control (ALC). Additional ALC attributes are in the ALC extension group.

#### .NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

#### Compliance Notes

If a specific driver implements the value True for this attribute, it shall also implement the IviRFSigGenALC extension capability group

## 4.2.4 Output Enabled

Data Type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	Configure Output Enabled

### .NET Property Name

`RF.OutputEnabled`

### COM Property Name

`RF.OutputEnabled`

### C Constant Name

`IVIRFSIGGEN_ATTR_OUTPUT_ENABLED`

### Description

If set to True, the signal the RF signal generator produces appears at the output connector. If set to False, the signal the RF signal generator produces does not appear at the output connector.

### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

### Compliance Notes

1. Instrument drivers shall support the value True.



### **4.3 *IviRFSigGenBase Functions***

In addition to the IVI required functions, the IviRFSigGen Base capability group includes the following functions:

- Configure RF
- Configure ALC Enabled (IVI-C only)
- Configure Output Enabled (IVI-C only)
- Disable All Modulation
- Is Settled
- Wait Until Settled

This section describes the behavior and requirements of each function.

### 4.3.1 Configure RF

#### Description

Configures the frequency and the power level of the RF output signal.

#### .NET Method Prototype

```
void RF.Configure (Double frequency,  
                  Double powerLevel);
```

#### COM Method Prototype

```
HRESULT RF.Configure ([in] DOUBLE Frequency,  
                     [in] DOUBLE PowerLevel);
```

#### C Prototype

```
ViStatus IviRFSigGen_ConfigureRF (ViSession Vi,  
                                  ViReal64 Frequency,  
                                  ViReal64 PowerLevel);
```

#### Parameters

Inputs	Description
Vi	Instrument handle.
Frequency	Specifies the frequency of the generated RF signal. The driver uses this value to set the Frequency attribute. See the attribute description for more details.
PowerLevel	Specifies the power level of the generated RF signal. The driver uses this value to set the Power Level attribute. See the attribute description for more details.

#### Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

### 4.3.2 Configure ALC Enabled (IVI-C only)

**Description**

Enables or disables the Automatic Level Control (ALC).

**.NET Method Prototype**

N/A  
(use the `Alc.Enabled` property)

**COM Method Prototype**

N/A  
(use the `ALC.Enabled` property)

**C Prototype**

```
ViStatus IviRFSigGen_ConfigureALCEnabled (ViSession Vi, ViBoolean ALCEnabled);
```

**Parameters**

Inputs	Description
Vi	Instrument handle.
ALCEnabled	Specifies whether the ALC is used. The driver uses this value to set the ALC Enabled attribute. See the attribute description for more details.

**Return Values (C)**

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

### 4.3.3 Configure Output Enabled (IVI-C only)

#### Description

Enables or disables the RF output signal.

#### .NET Method Prototype

N/A

(use the `RF.OutputEnabled` property)

#### COM Method Prototype

N/A

(use the `RF.OutputEnabled` property)

#### C Prototype

```
ViStatus IviRFSigGen_ConfigureOutputEnabled (ViSession Vi, ViBoolean OutputEnabled);
```

#### Parameters

Inputs	Description
Vi	Instrument handle.
OutputEnabled	Specifies whether the RF output is enabled. The driver uses this value to set the Output Enabled attribute. See the attribute description for more details.

#### Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

### 4.3.4 Disable All Modulation

#### Description

Disables all currently enabled modulations (e.g. analog, pulse, IQ, and digital modulation).

#### .NET Method Prototype

```
void RF.DisableAllModulation();
```

#### COM Method Prototype

```
HRESULT RF.DisableAllModulation();
```

#### C Prototype

```
ViStatus IviRFSigGen_DisableAllModulation (ViSession Vi);
```

#### Parameters

Inputs	Description
Vi	Instrument handle.

#### Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

### 4.3.5 Is Settled

**Description**

Queries if the RF output signal is currently settled.

**.NET Method Prototype**

```
Boolean RF.IsSettled ();
```

**COM Method Prototype**

```
HRESULT RF.IsSettled ([out, retval] VARIANT_BOOL *Done);
```

**C Prototype**

```
ViStatus IviRFSigGen_IsSettled (ViSession Vi, ViBoolean * Done);
```

**Parameters**

Inputs	Description
Vi	Instrument handle

Outputs	Description
Done (C/COM)	Returns True if the output signal is in settled, False otherwise.
Return Value (.NET)	Returns true if the output signal is in settled, false otherwise.

**Return Values (C/COM)**

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

### 4.3.6 Wait Until Settled

#### Description

This function waits until the state of the RF output signal has settled.

#### .NET Method Prototype

```
void RF.WaitUntilSettled (PrecisionTimeSpan maximumTime);
```

#### COM Method Prototype

```
HRESULT RF.WaitUntilSettled ([in] LONG MaxTimeMilliseconds);
```

#### C Prototype

```
ViStatus IviRFSigGen_WaitUntilSettled (ViSession Vi, ViInt32 MaxTimeMilliseconds);
```

#### Parameters

Inputs	Description
Vi	Instrument handle.
MaxTimeMillise conds	Defines the maximum time the function waits for the output to be settled. If the maximum time is exceeded, this function returns the Max Time Exceeded error. The units are milliseconds.
MaximumTime	Defines the maximum time the function waits for the output to be settled. If the maximum time is exceeded, this function returns the Max Time Exceeded error.

#### Defined Values for MaxTimeMilliseconds Parameter (C and COM)

Name	Description	
	Language	Identifier
Max Time Immediate	Sets timeout to immediate. The function returns immediately.	
	C	IVIRFSIGGEN_VAL_MAX_TIME_IMMEDIATE
	COM	IviRFSigGenMaxTimeImmediate
Max Time Infinite	Sets timeout to infinite. The function waits indefinitely for the settling to complete.	
	C	IVIRFSIGGEN_VAL_MAX_TIME_INFINITE
	COM	IviRFSigGenMaxTimeInfinite

#### Defined Values for MaximumTime Parameter (.NET)

Name	Description	
	Language	Identifier
Zero	Sets timeout to immediate. The function returns immediately.	
	.NET	PrecisionTimeSpan.Zero

MaxValue	Sets timeout to infinite. The function waits indefinitely for the settling to complete.	
	C	PrecisionTimeSpan.MaxValue

### Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return. The table below specifies additional class-defined status codes for this function.

Completion Codes	Description
Max Time Exceeded	Error: Maximum time exceeded before the operation completed.

### .NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

Note that the .NET MaxTimeExceededException is defined in *IVI-3.2: Inherent Capabilities Specification*.

### Compliance Notes

An IviRFSigGen specific driver need not implement the Max Time Immediate or the Max Time Infinite defined values for the MaxTimeMilliseconds parameter to be compliant with the IviRFSigGenBase Capability group.



#### **4.4 *IviRFSigGenBase Behavior Model***

After the user calls the Initialize or Reset functions, the RF signal generator produces an output signal based on its current configuration. All changes to the RF signal generator's IviRFSigGenBase functions and attributes take place immediately.

If the user executes the Wait Until Settled function, the driver will block any further operation until the function completes (i.e. the signal at the output of the RF signal generator has settled).

## 5. IviRFSigGenModulateAM Extension Group

---

### 5.1 *IviRFSigGenModulateAM Extension Group Overview*

The IviRFSigGenModulateAM Extension Group supports signal generators that can apply amplitude modulation to the RF output signal. The user can enable or disable amplitude modulation, specify the source and coupling of the modulating signal and the modulation depth with linear or logarithmic attenuation.

### 5.2 *IviRFSigGenModulateAM Attributes*

The IviRFSigGenModulateAM extension group defines the following attributes:

- AM Enabled
- AM Source
- AM Scaling
- AM External Coupling
- AM Nominal Voltage
- AM Depth

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 30, *IviRFSigGen Attribute ID Definitions*.

### 5.2.1 AM Enabled

Data Type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	Configure AM Enabled

#### .NET Property Name

`AnalogModulation.AM.Enabled`

#### COM Property Name

`AnalogModulation.AM.Enabled`

#### C Constant Name

`IVIRFSIGGEN_ATTR_AM_ENABLED`

#### Description

If set to `True`, the RF signal generator applies amplitude modulation to the RF output signal. If set to `False`, the RF signal generator does not apply amplitude modulation to the output signal.

#### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

#### Compliance Notes

1. Instrument drivers shall support the values `True` and `False`.

### 5.2.2 AM Source

Data Type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	Configure AM

#### .NET Property Name

`AnalogModulation.AM.Source`

#### COM Property Name

`AnalogModulation.AM.Source`

#### C Constant Name

`IVIRFSIGGEN_ATTR_AM_SOURCE`

#### Description

Specifies the source of the signal that is used as the modulating signal. If more than one source is specified, the voltages of all specified sources (internal and external) are summed. Multiple source names are separated by commas.

#### Defined Values

To get the valid source names, use the `Get Modulation Source Name` function (IVI-C) or the `Analog Modulation Source Name` property (IVI-COM) from the `IviRFSigGenAnalogModulationSource` extension group.

#### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

### 5.2.3 AM Scaling

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	Configure AM

#### .NET Property Name

`AnalogModulation.AM.Scaling`

#### .NET Enumeration Name

`AMScaling`

#### COM Property Name

`AnalogModulation.AM.Scaling`

#### COM Enumeration Name

`IviRFSigGenAMScalingEnum`

#### C Constant Name

`IVIRFSIGGEN_ATTR_AM_SCALING`

#### Description

Specifies linear or logarithmic characteristic for amplitude modulation. The units of the AM Depth attribute is changed with this setting.

#### Defined Values

<i>Name</i>	<i>Description</i>		
		<i>Language</i>	<i>Identifier</i>
Linear	Enables linear attenuation for amplitude modulation.		
		.NET	<code>AMScaling.Linear</code>
		C	<code>IVIRFSIGGEN_VAL_AM_SCALING_LINEAR</code>
		COM	<code>IviRFSigGenAMScalingLinear</code>
Logarithmic	Enables logarithmic attenuation for amplitude modulation.		
		.NET	<code>AMScaling.Logarithmic</code>
		C	<code>IVIRFSIGGEN_VAL_AM_SCALING_LOGARITHMIC</code>
		COM	<code>IviRFSigGenAMScalingLogarithmic</code>

#### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## Compliance Notes

1. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_AM_SCALING_CLASS_EXT_BASE` and less than `IVIRFSIGGEN_VAL_AM_SCALING_SPECIFIC_EXT_BASE`.
2. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_AM_SCALING_SPECIFIC_EXT_BASE`.
3. When an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to AM Scale Specific Ext Base.

See Section Attribute Value Definitions, for the definitions of AM Scaling Specific Ext Base, `IVIRFSIGGEN_VAL_AM_SCALING_SPECIFIC_EXT_BASE` and `IVIRFSIGGEN_VAL_AM_SCALING_CLASS_EXT_BASE`.

## 5.2.4 AM External Coupling

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	Configure AM External Coupling

### .NET Property Name

`AnalogModulation.AM.ExternalCoupling`

### .NET Enumeration Name

`ExternalCoupling`

### COM Property Name

`AnalogModulation.AM.ExternalCoupling`

### COM Enumeration Name

`IviRFSigGenAMExternalCouplingEnum`

### C Constant Name

`IVIRFSIGGEN_ATTR_AM_EXTERNAL_COUPLING`

### Description

Specifies the coupling of the external source of the modulating signal.

### Defined Values

Name	Description		
		Language	Identifier
AC	The external source is coupled for AC only.		
		.NET	<code>ExternalCoupling.AC</code>
		C	<code>IVIRFSIGGEN_VAL_AM_EXTERNAL_COUPLING_AC</code>
		COM	<code>IviRFSigGenAMExternalCouplingAC</code>
DC	The external source is coupled for both DC and AC		
		.NET	<code>ExternalCoupling.DC</code>
		C	<code>IVIRFSIGGEN_VAL_AM_EXTERNAL_COUPLING_DC</code>
		COM	<code>IviRFSigGenAMExternalCouplingDC</code>

### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## Compliance Notes

1. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_AM_EXTERNAL_COUPLING_CLASS_EXT_BASE` and less than `IVIRFSIGGEN_VAL_AM_EXTERNAL_COUPLING_SPECIFIC_EXT_BASE`.
2. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_AM_EXTERNAL_COUPLING_SPECIFIC_EXT_BASE`.
3. When an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to AM External Coupling Specific Ext Base.

See Section Attribute Value Definitions, for the definitions of AM External Coupling Specific Ext Base, `IVIRFSIGGEN_VAL_AM_EXTERNAL_COUPLING_SPECIFIC_EXT_BASE` and `IVIRFSIGGEN_VAL_AM_EXTERNAL_COUPLING_CLASS_EXT_BASE`.



5.2.5 AM Nominal Voltage

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64	RO	N/A	None	None

**.NET Property Name**

`AnalogModulation.AM.NominalVoltage`

**COM Property Name**

`AnalogModulation.AM.NominalVoltage`

**C Constant Name**

`IVIRFSIGGEN_ATTR_AM_NOMINAL_VOLTAGE`

**Description**

Returns the voltage at which the instrument achieves the amount of modulation specified by the AM Depth attribute. The units are Volts.

**.NET Exceptions**

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## 5.2.6 AM Depth

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	Configure AM

### .NET Property Name

`AnalogModulation.AM.Depth`

### COM Property Name

`AnalogModulation.AM.Depth`

### C Constant Name

`IVIRFSIGGEN_ATTR_AM_DEPTH`

### Description

Specifies the extent of modulation the signal generator applies to the RF-signal (carrier waveform) with the modulating signal as a result of summing all sources, internal and external. The amount of the specified modulation depth is achieved with a modulating voltage of AM Nominal Voltage.

If the AM Scaling attribute is set to Linear, then the units are percent (%). If the AM Scaling attribute is set to logarithmic, then the units are dBm.

### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

### **5.3 *IviRFSigGenModulateAM Functions***

The IviRFSigGenModulateAM extension defines the following functions:

- Configure AM Enabled (IVI-C only)
- Configure AM External Coupling (IVI-C only)
- Configure AM

This section describes the behavior and requirements of each function.

### 5.3.1 Configure AM Enabled (IVI-C only)

#### Description

Configures the signal generator to apply amplitude modulation to the RF output signal.

#### .NET Method Prototype

N/A  
(use the `AnalogModulation.AM.Enabled` property)

#### COM Method Prototype

N/A  
(use the `AnalogModulation.AM.Enabled` property)

#### C Prototype

```
ViStatus IviRFSigGen_ConfigureAMEnabled (ViSession Vi, ViBoolean Enabled);
```

#### Parameters

Inputs	Description
Vi	Instrument handle
Enabled	Enables or disables amplitude modulation. The driver uses this value to set the <code>AM Enabled</code> attribute. See the attribute description for more details.

#### Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

### 5.3.2 Configure AM External Coupling (IVI-C only)

#### Description

Configures the coupling of the external source the signal generator uses for amplitude modulation of the output signal.

#### .NET Method Prototype

N/A

(use the `AnalogModulation.AM.ExternalCoupling` property)

#### COM Method Prototype

N/A

(use the `AnalogModulation.AM.ExternalCoupling` property)

#### C Prototype

```
ViStatus IviRFSigGen_ConfigureAMExternalCoupling (ViSession Vi,  
                                                    ViInt32 Coupling);
```

#### Parameters

Inputs	Description
Vi	Instrument handle
Coupling	Specifies the coupling of the external source. The driver uses this value to set the AM External Coupling attribute. See the attribute description for more details.

#### Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

### 5.3.3 Configure AM

#### Description

Configures the attributes that control the signal generator's amplitude modulation. These attributes are the AM modulation source, scaling and depth.

#### .NET Method Prototype

```
void AnalogModulation.AM.Configure (String source,  
                                     AMScaling scaling,  
                                     Double depth);
```

#### COM Method Prototype

```
HRESULT AnalogModulation.AM.Configure ([in] BSTR Source,  
                                         [in] IviRFSigGenAMScalingEnum Scaling,  
                                         [in] DOUBLE Depth);
```

#### C Prototype

```
ViStatus IviRFSigGen_ConfigureAM (ViSession Vi, ViConstString Source,  
                                   ViInt32 Scaling,  
                                   ViReal64 Depth);
```

#### Parameters

Inputs	Description
Vi	Instrument handle
Source	Specifies the source of the signal that the signal generator uses to modulate the output signal. The driver uses this value to set the AM Source attribute. See the attribute description for more details.
Scaling	Specifies the scaling of the modulation. The driver uses this value to set the AM Scaling attribute. See the attribute description for more details.
Depth	Specifies the extent of modulation. The driver uses this value to set the AM Depth attribute. See the attribute description for more details.

#### Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

#### .NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

## **5.4 *IviRFSigGenModulateAM Behavior Model***

The IviRFSigGenModulateAM Extension Group follows the behavior model of the IviRFSigGenBase capability group. The only modification to the behavior model from the IviRFSigGenBase capability group is the ability to configure IviRFSigGenModulateAM settings.

## **5.5 *IviRFSigGenModulateAM Compliance Notes***

1. If a specific driver implements the IviRFSigGenModulateAM Extension Group, it shall also implement the IviRFSigGenAnalogModulationSource Extension Group.
2. If the AM source attribute is not set to an external modulation source, the AM External Coupling attribute need not to be in a user specified state.

## 6. IviRFSigGenModulateFM Extension Group

---

### 6.1 *IviRFSigGenModulateFM Extension Group Overview*

The IviRFSigGenModulateFM Extension Group supports signal generators that can apply frequency modulation to the RF output signal. The user can enable or disable frequency modulation, specify the source and coupling of the modulating signal and the peak frequency deviation.

### 6.2 *IviRFSigGenModulateFM Attributes*

The IviRFSigGenModulateFM extension group defines the following attributes:

- FM Enabled
- FM Source
- FM External Coupling
- FM Nominal Voltage
- FM Deviation

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 30, *IviRFSigGen Attribute ID Definitions*.



## 6.2.1 FM Enabled

Data Type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	Configure FM Enabled

### .NET Property Name

`AnalogModulation.FM.Enabled`

### COM Property Name

`AnalogModulation.FM.Enabled`

### C Constant Name

`IVIRFSIGGEN_ATTR_FM_ENABLED`

### Description

If set to `True`, the RF signal generator applies frequency modulation to the RF output signal. If set to `False`, the RF signal generator does not apply frequency modulation to the RF output signal.

### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

### Compliance Notes

1. Instrument drivers shall support the values `True` and `False`.

## 6.2.2 FM Source

Data Type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	Configure FM

### .NET Property Name

`AnalogModulation.FM.Source`

### COM Property Name

`AnalogModulation.FM.Source`

### C Constant Name

`IVIRFSIGGEN_ATTR_FM_SOURCE`

### Description

Specifies the source of the signal that is used as the modulating signal. If more than one source is specified, the voltages of all sources (internal and external) are summed. Multiple source names are separated by commas.

### Defined Values

To get the values (names) use the Get Modulation Source Name function (IVI-C) or the Analog Modulation Source Name property (IVI-COM) from the IviRFSigGenAnalogModulationSource extension group.

### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

### 6.2.3 FM External Coupling

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	Configure FM External Coupling

#### .NET Property Name

`AnalogModulation.FM.ExternalCoupling`

#### .NET Enumeration Name

`ExternalCoupling`

#### COM Property Name

`AnalogModulation.FM.ExternalCoupling`

#### COM Enumeration Name

`IviRFSigGenFMExternalCouplingEnum`

#### C Constant Name

`IVIRFSIGGEN_ATTR_FM_EXTERNAL_COUPLING`

#### Description

Specifies the coupling of the external source of the modulating signal.

#### Defined Values

Name	Description		
		Language	Identifier
AC	The external source is coupled for AC only.		
		.NET	<code>ExternalCoupling.AC</code>
		C	<code>IVIRFSIGGEN_VAL_FM_EXTERNAL_COUPLING_AC</code>
		COM	<code>IviRFSigGenFMExternalCouplingAC</code>
DC	The external source is coupled for both DC and AC		
		.NET	<code>ExternalCoupling.DC</code>
		C	<code>IVIRFSIGGEN_VAL_FM_EXTERNAL_COUPLING_DC</code>
		COM	<code>IviRFSigGenFMExternalCouplingDC</code>

#### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## Compliance Notes

1. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_FM_EXTERNAL_COUPLING_CLASS_EXT_BASE` and less than `IVIRFSIGGEN_VAL_FM_EXTERNAL_COUPLING_SPECIFIC_EXT_BASE`.
2. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_FM_EXTERNAL_COUPLING_SPECIFIC_EXT_BASE`.
3. When an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to FM External Coupling Specific Ext Base.

See Section Attribute Value Definitions, for the definitions of FM External Coupling Specific Ext Base, `IVIRFSIGGEN_VAL_FM_EXTERNAL_COUPLING_SPECIFIC_EXT_BASE` and `IVIRFSIGGEN_VAL_FM_EXTERNAL_COUPLING_CLASS_EXT_BASE`.

## 6.2.4 FM Nominal Voltage

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64	RO	N/A	None	None

### .NET Property Name

`AnalogModulation.FM.NominalVoltage`

### COM Property Name

`AnalogModulation.FM.NominalVoltage`

### C Constant Name

`IVIRFSIGGEN_ATTR_FM_NOMINAL_VOLTAGE`

### Description

Returns the voltage at which the instrument achieves the amount of modulation specified by the FM Deviation attribute. The units are Volts.

### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## 6.2.5 FM Deviation

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	Configure FM

### .NET Property Name

`AnalogModulation.FM.Deviation`

### COM Property Name

`AnalogModulation.FM.Deviation`

### C Constant Name

`IVIRFSIGGEN_ATTR_FM_DEVIATION`

### Description

Specifies the extent of modulation (peak frequency deviation) the signal generator applies to the RF-signal (carrier waveform) with the modulating signal as a result of summing all sources, internal and external. The amount of the specified FM modulation deviation is achieved with a modulating voltage of FM Nominal Voltage. The units are Hertz.

### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

### **6.3 *IviRFSigGenModulateFM Functions***

The IviRFSigGenModulateFM extension group defines the following functions:

- Configure FM Enabled (IVI-C only)
- Configure FM External Coupling (IVI-C only)
- Configure FM

This section describes the behavior and requirements of each function.

### 6.3.1 Configure FM Enabled (IVI-C only)

#### Description

Configures the signal generator to apply frequency modulation to the RF output signal.

#### .NET Method Prototype

N/A

(use the `AnalogModulation.FM.Enabled` property)

#### COM Method Prototype

N/A

(use the `AnalogModulation.FM.Enabled` property)

#### C Prototype

```
ViStatus IviRFSigGen_ConfigureFMEnabled (ViSession Vi, ViBoolean Enabled);
```

#### Parameters

Inputs	Description
Vi	Instrument handle
Enabled	Enables or disables frequency modulation. The driver uses this value to set the FM Enabled attribute. See the attribute description for more details.

#### Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.



### 6.3.2 Configure FM External Coupling (IVI-C only)

**Description**

Configures the coupling of the external source the generator uses for frequency modulation of the output signal.

**.NET Method Prototype**

N/A  
(use the `AnalogModulation.FM.ExternalCoupling` property)

**COM Method Prototype**

N/A  
(use the `AnalogModulation.FM.ExternalCoupling` property)

**C Prototype**

```
ViStatus IviRFSigGen_ConfigureFMExternalCoupling (ViSession Vi,  
                                                    ViInt32 Coupling);
```

**Parameters**

Inputs	Description
Vi	Instrument handle
Coupling	Specifies the coupling of the external source. The driver uses this value to set the FM External Coupling attribute. See the attribute description for more details.

**Return Values (C)**

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

### 6.3.3 Configure FM

#### Description

Configures the attributes that control the signal generator's frequency modulation. The attributes are the modulation deviation and the modulating source(s).

#### .NET Method Prototype

```
void AnalogModulation.FM.Configure (String source,  
                                     Double deviation);
```

#### COM Method Prototype

```
HRESULT AnalogModulation.FM.Configure ([in] BSTR Source,  
                                         [in] DOUBLE Deviation);
```

#### C Prototype

```
ViStatus IviRFSigGen_ConfigureFM (ViSession Vi, ViConstString Source,  
                                   ViReal64 Deviation);
```

#### Parameters

Inputs	Description
Vi	Instrument handle
Source	Specifies the modulating source. The driver uses this value to set the FM Source attribute. See the attribute description for more details.
Deviation	Specifies the extent of modulation. The driver uses this value to set the FM Deviation attribute. See the attribute description for more details.

#### Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

#### .NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

## **6.4 *IviRFSigGenModulateFM Behavior Model***

The IviRFSigGenModulateFM Extension Group follows the behavior model of the IviRFSigGenBase capability group. The only modification to the behavior model from the IviRFSigGenBase capability group is the ability to configure IviRFSigGenModulateFM settings.

## **6.5 *IviRFSigGenModulateFM Compliance Notes***

1. If a specific driver implements the IviRFSigGenModulateFM Extension Group, it shall also implement the IviRFSigGenAnalogModulationSource Extension Group.
2. If the FM Source attribute is not set to an external modulation source, the FM External Coupling attribute need not to be in a user specified state.

## 7. IviRFSigGenModulatePM Extension Group

---

### 7.1 *IviRFSigGenModulatePM Extension Group Overview*

The IviRFSigGenModulatePM Extension Group supports signal generators that can apply phase modulation to the RF output signal. The user can enable or disable phase modulation, specify the source and coupling of the modulating signal and the peak phase deviation.

### 7.2 *IviRFSigGenModulatePM Attributes*

The IviRFSigGenModulatePM extension group defines the following attributes:

- PM Enabled
- PM Source
- PM External Coupling
- PM Nominal Voltage
- PM Deviation

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 30, *IviRFSigGen Attribute ID Definitions*.

### 7.2.1 PM Enabled

Data Type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	Configure PM Enabled

#### .NET Property Name

`AnalogModulation.PM.Enabled`

#### COM Property Name

`AnalogModulation.PM.Enabled`

#### C Constant Name

`IVIRFSIGGEN_ATTR_PM_ENABLED`

#### Description

If set to `True`, the RF signal generator applies phase modulation to the RF output signal. If set to `False`, the RF signal generator does not apply phase modulation to the RF output signal.

#### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

#### Compliance Notes

1. Instrument drivers shall support the values `True` and `False`.

### 7.2.2 PM Source

Data Type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	Configure PM

#### .NET Property Name

`AnalogModulation.PM.Source`

#### COM Property Name

`AnalogModulation.PM.Source`

#### C Constant Name

`IVIRFSIGGEN_ATTR_PM_SOURCE`

#### Description

Specifies the source of the signal that is used as the modulating signal. If more than one source is specified, the voltages of all sources (internal and external) are summed. Multiple source names are separated by commas.

#### Defined Values

To get the values (names) use the Get Modulation Source Name function (IVI-C) or the Analog Modulation Source Name property (IVI-COM) from the IviRFSigGenAnalogModulationSource extension group.

#### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

### 7.2.3 PM External Coupling

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	Configure PM External Coupling

#### .NET Property Name

`AnalogModulation.PM.ExternalCoupling`

#### .NET Enumeration Name

`ExternalCoupling`

#### COM Property Name

`AnalogModulation.PM.ExternalCoupling`

#### COM Enumeration Name

`IviRFSigGenPMExternalCouplingEnum`

#### C Constant Name

`IVIRFSIGGEN_ATTR_PM_EXTERNAL_COUPLING`

#### Description

Specifies the coupling of the external source of the modulating signal.

#### Defined Values

Name	Description		
		Language	Identifier
AC	The external source is coupled for AC only.		
		.NET	<code>ExternalCoupling.AC</code>
		C	<code>IVIRFSIGGEN_VAL_PM_EXTERNAL_COUPLING_AC</code>
		COM	<code>IviRFSigGenPMExternalCouplingAC</code>
DC	The external source is coupled for both DC and AC		
		.NET	<code>ExternalCoupling.DC</code>
		C	<code>IVIRFSIGGEN_VAL_PM_EXTERNAL_COUPLING_DC</code>
		COM	<code>IviRFSigGenPMExternalCouplingDC</code>

#### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## Compliance Notes

1. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_PM_EXTERNAL_COUPLING_CLASS_EXT_BASE` and less than `IVIRFSIGGEN_VAL_PM_EXTERNAL_COUPLING_SPECIFIC_EXT_BASE`.
2. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_PM_EXTERNAL_COUPLING_SPECIFIC_EXT_BASE`.
3. When an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to PM External Coupling Specific Ext Base.

See Section Attribute Value Definitions, for the definitions of AM External Coupling Specific Ext Base, `IVIRFSIGGEN_VAL_PM_EXTERNAL_COUPLING_SPECIFIC_EXT_BASE` and `IVIRFSIGGEN_VAL_PM_EXTERNAL_COUPLING_CLASS_EXT_BASE`.



## 7.2.4 PM Nominal Voltage

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64	RO	N/A	None	None

### .NET Property Name

`AnalogModulation.PM.NominalVoltage`

### COM Property Name

`AnalogModulation.PM.NominalVoltage`

### C Constant Name

`IVIRFSIGGEN_ATTR_PM_NOMINAL_VOLTAGE`

### Description

Returns the voltage at which the instrument achieves the amount of modulation specified by the PM Deviation attribute. The units are Volts.

### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## 7.2.5 PM Deviation

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	Configure PM

### .NET Property Name

`AnalogModulation.PM.Deviation`

### COM Property Name

`AnalogModulation.PM.Deviation`

### C Constant Name

`IVIRFSIGGEN_ATTR_PM_DEVIATION`

### Description

Specifies the extent of modulation (peak phase deviation) the signal generator applies to the RF-signal (carrier waveform) with the modulating signal as a result of summing all sources, internal and external. The amount of the specified PM modulation deviation is achieved with a modulating voltage of PM Nominal Voltage. The units are radians.

### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

### **7.3 *IviRFSigGenModulatePM Functions***

The IviRFSigGenModulatePM extension group defines the following functions:

- Configure PM Enabled (IVI-C only)
- Configure PM External Coupling (IVI-C only)
- Configure PM

This section describes the behavior and requirements of each function.

### 7.3.1 Configure PM Enabled (IVI-C only)

#### Description

Configures the signal generator to apply phase modulation to the RF output signal.

#### .NET Method Prototype

N/A  
(use the `AnalogModulation.PM.Enabled` property)

#### COM Method Prototype

N/A  
(use the `AnalogModulation.PM.Enabled` property)

#### C Prototype

```
ViStatus IviRFSigGen_ConfigurePMEnabled (ViSession Vi, ViBoolean Enabled);
```

#### Parameters

Inputs	Description
Vi	Instrument handle
Enabled	Enables or disables phase modulation. The driver uses this value to set the PM Enabled attribute. See the attribute description for more details.

#### Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

7.3.2 Configure PM External Coupling (IVI-C only)

Description

Configures the coupling of the external source for phase modulation.

.NET Method Prototype

N/A  
(use the `AnalogModulation.PM.ExternalCoupling` property)

COM Method Prototype

N/A  
(use the `AnalogModulation.PM.ExternalCoupling` property)

C Prototype

```
ViStatus IviRFSigGen_ConfigurePMExternalCoupling (ViSession Vi,  
                                                    ViInt32 Coupling);
```

Parameters

Inputs	Description
Vi	Instrument handle
Coupling	Specifies the coupling of the external source. The driver uses this value to set the PM External Coupling attribute. See the attribute description for more details.

Return Values (C)

### 7.3.3 Configure PM

#### Description

Configures the attributes that control the signal generator's phase modulation. The attributes are the modulation deviation and the modulating source(s).

#### .NET Method Prototype

```
void AnalogModulation.PM.Configure (String source,  
                                     Double deviation);
```

#### COM Method Prototype

```
HRESULT AnalogModulation.PM.Configure ([in] BSTR Source,  
                                         [in] DOUBLE Deviation);
```

#### C Prototype

```
ViStatus IviRFSigGen_ConfigurePM (ViSession Vi,  
                                   ViConstString Source,  
                                   ViReal64 Deviation);
```

#### Parameters

Inputs	Description
Vi	Instrument handle
Source	Specifies the source of the signal the signal generator uses to modulate the output signal. The driver uses this value to set the PM Source attribute. See the attribute description for more details.
Deviation	Specifies the extent of modulation. The driver uses this value to set the PM Deviation attribute. See the attribute description for more details.

#### Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

#### .NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

#### **7.4 *IviRFSigGenModulatePM Behavior Model***

The IviRFSigGenModulatePM Extension Group follows the behavior model of the IviRFSigGenBase capability group. The only modification to the behavior model from the IviRFSigGenBase capability group is the ability to configure IviRFSigGenModulatePM settings.

#### **7.5 *IviRFSigGenModulatePM Compliance Notes***

1. If a specific driver implements the IviRFSigGenModulatePM Extension Group, it shall also implement the IviRFSigGenAnalogModulationSource Extension Group.
2. If the PM source attribute is not set to an external modulation source, the PM External Coupling attribute need not to be in a user specified state.

## 8. IviRFSigGenAnalogModulationSource Extension Group

---

### 8.1 *IviRFSigGenAnalogModulationSource Extension Group Overview*

The source of the modulating signal is a repeated capability. One or more internal sources (LF Generators) and/or one and more external sources can be combined. The voltage of all signals is summed before modulating the RF Signal. This applies to the following extension groups:

- IviRFSigGenModulateAM
- IviRFSigGenModulateFM
- IviRFSigGenModulatePM

Typically, all the LFGenerators defined in the IviRFSigGenLFGenerator extension group have a corresponding Analog Modulation Source.

### 8.2 *IviRFSigGenAnalogModulationSource Attributes*

The IviRFSigGenAnalogModulationSource extension group defines the following attributes:

- Analog Modulation Source Count
- Analog Modulation Source Name (IVI-COM only)

This section describes the behavior and requirements of this attribute. The actual value for each attribute ID is defined in Section 30, *IviRFSigGen Attribute ID Definitions*.



### 8.2.1 Analog Modulation Source Count

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32	RO	N/A	None	

#### .NET Property Name

`AnalogModulation.Source.Count`

#### COM Property Name

`AnalogModulation.Source.Count`

#### C Constant Name

`IVIRFSIGGEN_ATTR_ANALOG_MODULATION_SOURCE_COUNT`

#### Description

Specifies how many analog modulation sources are available.

#### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## 8.2.2 Analog Modulation Source Name (IVI-COM Only)

Data Type	Access	Applies to	Coercion	High Level Functions
ViString	RO	N/A	None	None

### .NET Property Name

N/A  
(Use the Get Analog Modulation Source Name function.)

### COM Property Name

```
HRESULT AnalogModulation.Source.Name ([in] LONG Index,  
                                         [out,retval] BSTR* Name);
```

### C Constant Name

N/A  
(Use the Get Analog Modulation Source Name function.)

### Description

This property returns the Analog Modulation Source identifier that corresponds to the one-based index that the user specifies. If the driver defines a qualified analog modulation source name, this property returns the qualified name. If the value that the user pass for the Index parameter is less than one or greater than the value of the Analog Modulation Source Count attribute, the property returns an empty string in the Name parameter and returns the Invalid Value error.

### Parameters

Inputs	Description	Base Type
Index	A one-based index that defines which name to return.	ViInt32

Output/Return Value	Description	Base Type
Name	A driver-allocated buffer into which the driver stores the analog modulation source name.	ViChar[]

### **8.3 *IviRFSigGenAnalogModulationSource Functions***

The IviRFSigGenAnalogModulationSource extension group defines the following functions:

- Get Analog Modulation Source Name (IVI-C & IVI.NET only)

This section describes the behavior and requirements of each function.

8.3.1 Get Analog Modulation Source Name (IVI-C & IVI.NET only)

Description

This function returns the specific driver defined analog modulation source name that corresponds to the index that the user specifies. If the driver defines a qualified analog modulation source name, this property returns the qualified name. In C and COM, the index is one-based. In .NET, the index is zero-based.

.NET Method Prototype

```
String AnalogModulation.Source.GetName (Int32 index);
```

COM Method Prototype

```
N/A  
(Use AnalogModulation.Source.Name property)
```

C Prototype

```
ViStatus IviRFSigGen_GetAnalogModulationSourceName (ViSession Vi,  
                                                    ViInt32 Index,  
                                                    ViInt32 NameBufferSize,  
                                                    ViChar Name[]);
```

Parameters

Inputs	Description
Vi	Instrument handle
Index	A one-based index that defines which name to return.
NameBufferSize	The number of bytes in the viChar array that the user specifies for the Name parameter.

Outputs	Description
Name (C)	A user-allocated buffer into which the driver stores the Analog Modulation Source name. The caller may pass VI_NULL for this parameter if the NameBufferSize parameter is 0.
Return Value (.NET)	The Analog Modulation Source name corresponding to the user specified index.

Return Values (C)

The IVI-3.2: *Inherent Capabilities Specification* defines general status codes that this function can return.

.NET Exceptions

The IVI-3.2: *Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this property.

**Compliance Notes**

For an instrument with only one modulation source, that is the Analog Modulation Source Count attribute is one, the driver may return an empty string in the Name parameter.

#### **8.4 *IviRFSigGenAnalogModulationSource Behavior Model***

The IviRFSigGenAnalogModulationSource Extension Group follows the behavior model of the IviRFSigGenBase capability group.

#### **8.5 *IviRFSigGenAnalogModulationSource Compliance Notes***

If the driver supports this extension group it shall support the IviRFSigGenBase capabilities and at least one of the following extension groups:

- IviRFSigGenModulateAM
- IviRFSigGenModulateFM
- IviRFSigGenModulatePM

## 9. IviRFSigGenModulatePulse Extension Group

---

### 9.1 *IviRFSigGenModulatePulse Extension Group Overview*

The IviRFSigGenModulatePulse Extension Group supports signal generators that can apply pulse modulation to the RF output signal. The user can enable or disable pulse modulation, and specify the source and the polarity of the modulating signal.

### 9.2 *IviRFSigGenModulatePulse Attributes*

The IviRFSigGenModulatePulse extension group defines the following attributes:

- Pulse Modulation Enabled
- Pulse Modulation Source
- Pulse Modulation External Polarity

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 30, *IviRFSigGen Attribute ID Definitions*.

### 9.2.1 Pulse Modulation Enabled

Data Type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	Configure Pulse Modulation Enabled

#### .NET Property Name

`PulseModulation.Enabled`

#### COM Property Name

`PulseModulation.Enabled`

#### C Constant Name

`IVIRFSIGGEN_ATTR_PULSE_MODULATION_ENABLED`

#### Description

If set to True, enables pulse modulation of the RF output signal. If set to False, disables pulse modulation of the RF output signal .

#### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

#### Compliance Notes

1. Instrument drivers shall support the values `True` and `False`.



## 9.2.2 Pulse Modulation Source

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	Configure Pulse Modulation Source

### .NET Property Name

`PulseModulation.Source`

### .NET Enumeration Name

`PulseModulationSource`

### COM Property Name

`PulseModulation.Source`

### COM Enumeration Name

`IviRFSigGenPulseModulationSourceEnum`

### C Constant Name

`IVIRFSIGGEN_ATTR_PULSE_MODULATION_SOURCE`

### Description

Specifies the source of the signal that is used as the modulating signal.

### Defined Values

Name	Description		
		Language	Identifier
Internal	The internal pulse generator (IviRFSigGenPulseGenerator Extension Group) is used for modulation.		
		.NET	<code>PulseModulationSource.Internal</code>
		C	<code>IVIRFSIGGEN_VAL_PULSE_MODULATION_SOURCE_INTERNAL</code>
		COM	<code>IviRFSigGenPulseModulationSourceInternal</code>
External	An external generator is used for modulation.		
		.NET	<code>PulseModulationSource.External</code>
		C	<code>IVIRFSIGGEN_VAL_PULSE_MODULATION_SOURCE_EXTERNAL</code>
		COM	<code>IviRFSigGenPulseModulationSourceExternal</code>

### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## Compliance Notes

1. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_PULSE_MODULATION_SOURCE_CLASS_EXT_BASE` and less than `IVIRFSIGGEN_VAL_PULSE_MODULATION_SOURCE_SPECIFIC_EXT_BASE`.
2. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_PULSE_MODULATION_SOURCE_SPECIFIC_EXT_BASE`.
3. When an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to Pulse Modulation Source Specific Ext Base.

See Section Attribute Value Definitions, for the definitions of Pulse ModulationSource Specific Ext Base, `IVIRFSIGGEN_VAL_PULSE_MODULATION_SOURCE_SPECIFIC_EXT_BASE` and `IVIRFSIGGEN_VAL_PULSE_MODULATION_SOURCE_CLASS_EXT_BASE`.

### 9.2.3 Pulse Modulation External Polarity

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	Configure Pulse Modulation External Polarity

#### .NET Property Name

`PulseModulationExternalPolarity` `PulseModulation.ExternalPolarity`

#### .NET Enumeration Name

`PulseModulationExternalPolarity`

#### COM Property Name

`PulseModulation.ExternalPolarity`

#### COM Enumeration Name

`IviRFSigGenPulseModulationExternalPolarityEnum`

#### C Constant Name

`IVIRFSIGGEN_ATTR_PULSE_MODULATION_EXTERNAL_POLARITY`

#### Description

Specifies the polarity of the external source signal.

#### Defined Values

Name	Description	
	Language	Identifier
Normal	The signal generator modulates the carrier signal with normal pulse polarity. Increasing the positive pulse voltage level results in higher RF level.	
	.NET	<code>PulseModulationExternalPolarity.Normal</code>
	C	<code>IVIRFSIGGEN_VAL_PULSE_MODULATION_EXTERNAL_POLARITY_NORMAL</code>
	COM	<code>IviRFSigGenPulseModulationExternalPolarityNormal</code>
Inverse	The signal generator modulates the carrier signal with inverted pulse polarity. Increasing the positive pulse voltage level results in lower RF level.	
	.NET	<code>PulseModulationExternalPolarity.Inverse</code>
	C	<code>IVIRFSIGGEN_VAL_PULSE_MODULATION_EXTERNAL_POLARITY_INVERSE</code>
	COM	<code>IviRFSigGenPulseModulationExternalPolarityInverse</code>

#### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## Compliance Notes

1. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_PULSE_MODULATION_EXTERNAL_POLARITY_CLASS_EXT_BASE` and less than `IVIRFSIGGEN_VAL_PULSE_MODULATION_EXTERNAL_POLARITY_SPECIFIC_EXT_BASE`.
2. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_PULSE_MODULATION_EXTERNAL_POLARITY_SPECIFIC_EXT_BASE`.
3. When an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to Pulse Modulation External Polarity Specific Ext Base.

See Section Attribute Value Definitions, for the definitions of Pulse Modulation Specific Ext Base, `IVIRFSIGGEN_VAL_PULSE_MODULATION_EXTERNAL_POLARITY_SPECIFIC_EXT_BASE` and `IVIRFSIGGEN_VAL_PULSE_MODULATION_EXTERNAL_POLARITY_CLASS_EXT_BASE`.

### **9.3 *IviRFSigGenModulatePulse Functions***

The IviRFSigGenPulseModulation extension group defines the following functions:

- Configure Pulse Modulation Enabled (IVI-C only)
- Configure Pulse Modulation Source (IVI-C only)
- Configure Pulse Modulation External Polarity (IVI-C only)

This section describes the behavior and requirements of each function.

9.3.1 Configure Pulse Modulation Enabled (IVI-C only)

Description

Configures the signal generator to apply pulse modulation to the RF output signal.

.NET Method Prototype

N/A  
(use the `PulseModulation.Enabled` property)

COM Method Prototype

N/A  
(use the `PulseModulation.Enabled` property)

C Prototype

```
ViStatus IviRFSigGen_ConfigurePulseModulationEnabled (ViSession Vi, ViBoolean Enabled);
```

Parameters

Inputs	Description
Vi	Instrument handle
Enabled	True enables pulse modulation. The driver uses this value to set the Pulse Modulation Enabled attribute. See the attribute description for more details.

Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

9.3.2 Configure Pulse Modulation Source (IVI-C only)

Description

Configures the source the signal generator uses for pulse modulation of the RF output signal.

.NET Method Prototype

N/A  
(use the `PulseModulation.Source` property)

COM Method Prototype

N/A  
(use the `PulseModulation.Source` property)

C Prototype

```
ViStatus IviRFSigGen_ConfigurePulseModulationSource (ViSession Vi, ViInt32
Source);
```

Parameters

Inputs	Description
Vi	Instrument handle
Source	Specifies the source of the signal, the signal generator uses to modulate the output signal. The driver uses this value to set the Pulse Modulation Source attribute. See the attribute description for more details.

Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

### 9.3.3 Configure Pulse Modulation External Polarity (IVI-C only)

#### Description

Specifies the polarity of the external source signal.

#### .NET Method Prototype

N/A

(use the `PulseModulation.ExternalPolarity` property)

#### COM Method Prototype

N/A

(use the `PulseModulation.ExternalPolarity` property)

#### C Prototype

```
ViStatus IviRFSigGen_ConfigurePulseModulationExternalPolarity (ViSession Vi,  
                                                                ViInt32 Polarity);
```

#### Parameters

Inputs	Description
Vi	Instrument handle
Polarity	Specifies the polarity of the signal used as the external modulation source. The driver uses this value to set the Pulse Modulation External Polarity attribute. See the attribute description for more details.

#### Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.



#### **9.4 *IviRFSigGenModulatePulse Behavior Model***

The IviRFSigGenModulatePulse Extension Group follows the behavior model of the IviRFSigGenBase capability group. The only modification to the behavior model from the IviRFSigGenBase capability group is the ability to configure IviRFSigGenModulatePulse settings.

#### **9.5 *IviRFSigGenModulatePulse Compliance Notes***

1. If a specific driver supports the value Internal for the Pulse Modulation Source attribute, it shall also implement the IviRFSigGenPulseGenerator Extension Group.
2. If a specific driver does not support an external pulse modulation source, it need not support the Configure Pulse Modulation External Polarity function and the Pulse Modulation External Polarity attribute.

## 10. IviRFSigGenLFGenerator Extension Group

---

### 10.1 *IviRFSigGenLFGenerator Extension Group Overview*

The IviRFSigGenLFGenerator Extension Group supports the LF generator (within the RF signal generator), that is normally used as a source for the modulators. To use the LF generator as a modulation source, the modulation source in the ModulateAM extensions group, ModulateFM extension group or ModulatePM extension group should be configured. The user can set the frequency and the waveform of the LF signal.

### 10.2 *IviRFSigGenLFGenerator Attributes*

The IviRFSigGenLFGenerator extension group defines the following attributes:

- Active LFGenerator
- LFGenerator Count
- LFGenerator Name (IVI-COM only)
- LFGenerator Frequency
- LFGenerator Waveform

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 30, *IviRFSigGen Attribute ID Definitions*.

### 10.2.1 Active LFGenerator

Data Type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	Set Active LFGenerator

#### .NET Property Name

`LFGenerator.ActiveLFGenerator`

#### COM Property Name

`LFGenerator.ActiveLFGenerator`

#### C Constant Name

`IVIRFSIGGEN_ATTR_ACTIVE_LF_GENERATOR`

#### Description

Specifies the LF generator which is currently active. The values for this attribute correspond to the LFGenerator repeated capability. If the driver defines a qualified LF generator name, this property returns the qualified name.

#### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

10.2.2 LFGenerator Count

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32	RO	N/A	None	N/A

**.NET Property Name**

`LFGenerator.Count`

**COM Property Name**

`LFGenerator.Count`

**C Constant Name**

`IVIRFSIGGEN_ATTR_LF_GENERATOR_COUNT`

**Description**

Specifies the number of LF generator sources available for a particular instrument.

**.NET Exceptions**

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

### 10.2.3 LFGGenerator Name (IVI-COM Only)

Data Type	Access	Applies to	Coercion	High Level Functions
ViString	RO	N/A	None	None

#### .NET Property Name

N/A  
(Use the Get LFGGenerator Name function.)

#### COM Property Name

```
HRESULT LFGGenerator.Name ([in] LONG Index,  
                           [out,retval] BSTR* Name);
```

#### C Constant Name

N/A  
(Use the Get LFGGenerator Name function.)

#### Description

This property returns the LFGGenerator identifier that corresponds to the one-based index that the user specifies. If the driver defines a qualified LFGGenerator name, this property returns the qualified name. If the value that the user pass for the Index parameter is less than one or greater than the value of the LFGGenerator Count attribute, the property returns an empty string in the Name parameter and returns the Invalid Value error.

#### Parameters

Inputs	Description	Base Type
Index	A one-based index that defines which name to return.	ViInt32

Output/Return Value	Description	Base Type
Name	A driver-allocated buffer into which the driver stores the LFGGenerator name.	ViChar[]

10.2.4 LFGenerator Frequency

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	Configure LF Generator

**.NET Property Name**

`LFGenerator.Frequency`

**COM Property Name**

`LFGenerator.Frequency`

**C Constant Name**

`IVIRFSIGGEN_ATTR_LF_GENERATOR_FREQUENCY`

**Description**

Specifies the frequency of the active LF generator. The units are Hertz.

**.NET Exceptions**

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## 10.2.5 LFGenerator Waveform

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	Configure LF Generator

### .NET Property Name

`LFGenerator.Waveform`

### .NET Enumeration Name

`LFGeneratorWaveform`

### COM Property Name

`LFGenerator.Waveform`

### COM Enumeration Name

`IviRFSigGenLFGeneratorWaveformEnum`

### C Constant Name

`IVIRFSIGGEN_ATTR_LF_GENERATOR_WAVEFORM`

### Description

Specifies the waveform of the active LF generator.

### Defined Values

Name	Description		
		Language	Identifier
Sine	Configures the LF generator to produce a sinusoidal waveform		
		.NET	<code>LFGeneratorWaveform.Sine</code>
		C	<code>IVIRFSIGGEN_VAL_LF_GENERATOR_WAVEFORM_SINE</code>
		COM	<code>IviRFSigGenLFGeneratorWaveformSine</code>
Square	Configures the LF generator to produce a square waveform		
		.NET	<code>LFGeneratorWaveform.Square</code>
		C	<code>IVIRFSIGGEN_VAL_LF_GENERATOR_WAVEFORM_SQUARE</code>
		COM	<code>IviRFSigGenLFGeneratorWaveformSquare</code>
Triangle	Configures the LF generator to produce a triangle waveform		
		.NET	<code>LFGeneratorWaveform.Triangle</code>
		C	<code>IVIRFSIGGEN_VAL_LF_GENERATOR_WAVEFORM_TRIANGLE</code>
		COM	<code>IviRFSigGenLFGeneratorWaveformTriangle</code>
Ramp Up	Configures the LF generator to produce a rising ramp waveform		
		.NET	<code>LFGeneratorWaveform.RampUp</code>
		C	<code>IVIRFSIGGEN_VAL_LF_GENERATOR_WAVEFORM_RAMP_UP</code>

		COM	IviRFSigGenLFGeneratorWaveformRampUp
Ramp Down	Configures the LF generator to produce a falling ramp waveform		
		.NET	LFGeneratorWaveform.RampDown
		C	IVIRFSIGGEN_VAL_LF_GENERATOR_WAVEFORM_RAMP_DOWN
		COM	IviRFSigGenLFGeneratorWaveformRampDown

### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

### Compliance Notes

1. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_LF_GENERATOR_WAVEFORM_CLASS_EXT_BASE` and less than `IVIRFSIGGEN_VAL_LF_GENERATOR_WAVEFORM_SPECIFIC_EXT_BASE`.
2. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_LF_GENERATOR_WAVEFORM_SPECIFIC_EXT_BASE`.
3. When an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to `LFGenerator Waveform Specific Ext Base`.

See Section Attribute Value Definitions, for the definitions of `LFGenerator Specific Ext Base`, `IVIRFSIGGEN_VAL_LF_GENERATOR_WAVEFORM_SPECIFIC_EXT_BASE` and `IVIRFSIGGEN_VAL_LF_GENERATOR_WAVEFORM_CLASS_EXT_BASE`.



### **10.3 *IviRFSigGenLFGenerator Functions***

The IviRFSigGenLFGenerator extension group defines the following function:

- Get LFGenerator Name (IVI-C & IVI.NET)
- Set Active LFGenerator (IVI-C only)
- Configure LFGenerator

This section describes the behavior and requirements of this function.

### 10.3.1 Get LFGenerator Name (IVI-C & IVI.NET only)

#### Description

This function returns the specific driver defined LF generator source name that corresponds to the index that the user specifies. If the driver defines a qualified LF generator name, this property returns the qualified name. In C, the index is one-based. In .NET, the index is zero-based.

#### .NET Method Prototype

```
String LFGenerator.GetName (Int32 index);
```

#### COM Method Prototype

N/A

(Use LFGenerator.Name property)

#### C Prototype

```
ViStatus IviRFSigGen_GetLFGeneratorName (ViSession Vi,  
                                          ViInt32 Index,  
                                          ViInt32 NameBufferSize,  
                                          ViChar Name[]);
```

#### Parameters

Inputs	Description
Vi	Instrument handle
Index	A one-based index that defines which name to return.
NameBufferSize	The number of bytes in the ViChar array that the user specifies for the Name parameter.

Outputs	Description
Name (C)	A user-allocated buffer into which the driver stores the LFGenerator name.  The caller may pass VI_NULL for this parameter if the NameBufferSize parameter is 0.
Return Value (.NET)	The LFGenerator name corresponding to the user-specified index.

#### Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

#### .NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

#### Compliance Notes

For an instrument with only one LF generator source, that is the LFGenerator Count attribute is one, the driver may return an empty string in the Name parameter.

### 10.3.2 Set Active LFGenerator (IVI-C Only)

**Description**

This function selects one of the available LF generator sources, and makes it the active LFGenerator.

**.NET Method Prototype**

N/A  
(Use the Active LF Generator property.)

**COM Method Prototype**

N/A  
(Use the Active LF Generator property.)

**C Function Prototype**

```
ViStatus IviRFSigGen_SetActiveLFGenerator (ViSession Vi,  
                                           ViConstString ActiveLFGenerator);
```

**Parameters**

Inputs	Description
Vi	Instrument handle.
ActiveLFGenerator	LF generator source to be selected. The driver uses this value to set the Active LFGenerator attribute. See the attribute description for more details.

**Return Values (C)**

The IVI-3.2:*Inherent Capabilities Specification* defines general status codes that this function can return.

### 10.3.3 Configure LFGenerator

#### Description

Configures the LF generator output frequency and waveform.

#### .NET Method Prototype

```
void LFGenerator.Configure (Double frequency,  
                           LFGeneratorWaveform waveform);
```

#### COM Method Prototype

```
HRESULT LFGenerator.Configure ([in] DOUBLE Frequency,  
                              [in] IviRFSigGenLFGeneratorWaveformEnum Waveform);
```

#### C Prototype

```
ViStatus IviRFSigGen_ConfigureLFGenerator (ViSession Vi,  
                                           ViReal64 Frequency,  
                                           ViInt32 Waveform);
```

#### Parameters

Inputs	Description
Vi	Instrument handle
Frequency	Specifies the frequency of the LF generator. The driver uses this value to set the LFGenerator Frequency attribute. See the attribute description for more details.
Waveform	Specifies the waveform of the LF generator. The driver uses this value to set the LFGenerator Waveform attribute. See the attribute description for more details.

#### Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

#### .NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

#### **10.4 *IviRFSigGenLFGenerator Behavior Model***

The IviRFSigGenLFGenerator Extension Group follows the behavior model of the IviRFSigGenBase capability group. The only modification to the behavior model from the IviRFSigGenBase capability group is the ability to configure IviRFSigGenLFGenerator settings.

#### **10.5 *IviRFSigGenLFGenerator Compliance Notes***

1. If a specific driver implements the IviRFSigGenLFGenerator Extension Group, it shall also implement the IviRFSigGenAnalogModulationSource Extension Group.

## 11. IviRFSigGen LFGeneratorOutput Extension Group

---

### 11.1 *IviRFSigGenLFGeneratorOutput Extension Group Overview*

The IviRFSigGenLFGenerator Output Extension Group supports the output of the LF generator, if it is used as a source for external devices. The user can set the amplitude and enable or disable the output. This extension group requires the LFGenerator extension group.

### 11.2 *IviRFSigGenLFGeneratorOutput Attributes*

- LFGenerator Output Amplitude
- LFGenerator Output Enabled

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 30, *IviRFSigGen Attribute ID Definitions*.

11.2.1 LFGGenerator Output Amplitude

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	Configure LFGGenerator Output

**.NET Property Name**

`LFGGenerator.Output.Amplitude`

**COM Property Name**

`LFGGenerator.Output.Amplitude`

**C Constant Name**

`IVIRFSIGGEN_ATTR_LF_GENERATOR_OUTPUT_AMPLITUDE`

**Description**

Specifies the output voltage the of the LF generator. The units are Volts, peak-to-peak.

**.NET Exceptions**

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

### 11.2.2 LFGenerator Output Enabled

Data Type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	Configure LF Output

#### .NET Property Name

`LFGenerator.Output.Enabled`

#### COM Property Name

`LFGenerator.Output.Enabled`

#### C Constant Name

`IVIRFSIGGEN_ATTR_LF_GENERATOR_OUTPUT_ENABLED`

#### Description

If set to `True`, the LF generator applies a signal to the output. If set to `False`, the LF generator does not apply a signal to the output.

#### Compliance Notes

1. Instrument drivers shall support the value `True`.

#### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.



### **11.3 *IviRFSigGenLFGeneratorOutput Functions***

The IviRFSigGenLFGeneratorOutput extension group defines the following function:

- Configure LFGenerator Output

This section describes the behavior and requirements of this function.

### 11.3.1 Configure LFGenerator Output

#### Description

Configures the attributes of the LF generator output (within the RF signal generator).

#### .NET Method Prototype

```
void LFGenerator.Output.Configure (Double amplitude,  
                                   Boolean enabled );
```

#### COM Method Prototype

```
HRESULT LFGenerator.Output.Configure ([in] DOUBLE Amplitude,  
                                       [in] VARIANT_BOOL Enabled );
```

#### C Prototype

```
ViStatus IviRFSigGen_ConfigureLFGeneratorOutput (ViSession Vi,  
                                                  ViReal64 Amplitude,  
                                                  ViBoolean Enabled);
```

#### Parameters

Inputs	Description
Vi	Instrument handle
Amplitude	Specifies the output voltage of the LF generator output signal. The driver uses this value to set the LFGenerator Output Amplitude attribute. See the attribute description for more details.
Enabled	Specifies whether the LF generator applies a signal to the output. The driver uses this value to set the LFGenerator Output Enabled attribute. See the attribute description for more details.

#### Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

#### .NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

#### **11.4 *IviRFSigGenLFGeneratorOutput Behavior Model***

The IviRFSigGenLFGeneratorOutput Extension Group follows the behavior model of the IviRFSigGenBase capability group. The only modification to the behavior model from the IviRFSigGenBase capability group is the ability to configure IviRFSigGenLFGeneratorOutput settings.

#### **11.5 *IviRFSigGenLFGeneratorOutput Compliance Notes***

1. If a specific driver implements the IviRFSigGenLFGeneratorOutput Extension Group, it shall also implement the IviRFSigGenLFGenerator Extension Group.

## 12. IviRFSigGenPulseGenerator Extension Group

---

### 12.1 *IviRFSigGenPulseGenerator Extension Group Overview*

The IviRFSigGenPulseGenerator Extension Group supports the pulse generator within the signal generator that is normally used as a source for the pulse modulator. The user can set the pulse period, width and delay. The pulse can be triggered, so source and external trigger polarity can be set.

Double pulse can be set with the functions and attributes of the IviRFSigGenPulseDoubleGenerator extension group.

The output of the pulse generator can be set with the functions and attributes of the IviRFSigGenPulseGeneratorOutput extension group.

### 12.2 *IviRFSigGenPulseGenerator Attributes*

The IviRFSigGenPulseGenerator extension group defines the following attributes:

- Pulse Internal Trigger Period
- Pulse Width
- Pulse Gating Enabled
- Pulse Trigger Source
- Pulse External Trigger Slope
- Pulse External Trigger Delay

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 30, *IviRFSigGen Attribute ID Definitions*.

### 12.2.1 Pulse Internal Trigger Period

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64 (C/COM) Ivi.Driver.PrecisionTimeSpan (.NET)	R/W	N/A	None	Configure Pulse Internal Trigger

#### .NET Property Name

```
Ivi.Driver.PrecisionTimeSpan PulseGenerator.InternalTriggerPeriod
```

#### COM Property Name

```
PulseGenerator.InternalTriggerPeriod
```

#### C Constant Name

```
IVIRFSIGGEN_ATTR_PULSE_INTERNAL_TRIGGER_PERIOD
```

#### Description

Specifies the period of the pulse generator output signal when Pulse Trigger Source is set to Internal. For C and COM, the units are seconds. For .NET, the units are implicit in the definition of Precision Time Span.

#### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## 12.2.2 Pulse Width

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64 (C/COM) Ivi.Driver.PrecisionTimeSpan (.NET)	R/W	N/A	None	Configure Pulse

### .NET Property Name

`Ivi.Driver.PrecisionTimeSpan.PulseGenerator.Width`

### COM Property Name

`PulseGenerator.Width`

### C Constant Name

`IVIRFSIGGEN_ATTR_PULSE_WIDTH`

### Description

Specifies the width of the output pulse. For C and COM, the units are seconds. For .NET, the units are implicit in the definition of Precision Time Span.

### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

12.2.3 Pulse Gating Enabled

Data Type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	Configure Pulse

.NET Property Name

PulseGenerator.GatingEnabled

COM Property Name

PulseGenerator.GatingEnabled

C Constant Name

IVIRFSIGGEN\_ATTR\_PULSE\_GATING\_ENABLED

Description

If set to True, enables the pulse trigger gating. If set to False, disables the pulse trigger gating.

.NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

Compliance Notes

- 1. Instrument drivers shall support the values True and False.

## 12.2.4 Pulse Trigger Source

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32 (C/COM)	R/W	N/A	None	Configure Pulse
ViString (.NET)	R/W	N/A	None	Configure Pulse

### .NET Property Name

`PulseGenerator.TriggerSource`

### COM Property Name

`PulseGenerator.TriggerSource`

### COM Enumeration Name

`IviRFSigGenPulseTriggerSourceEnum`

### C Constant Name

`IVIRFSIGGEN_ATTR_PULSE_TRIGGER_SOURCE`

### Description

Specifies the source of the trigger signal the pulse generator uses to generate one pulse.

### Defined Values

In IVI.NET the trigger source is a string. If an IVI driver supports a trigger source and the trigger source is listed in IVI-3.3 *Cross Class Capabilities Specification*, Section 3 then the IVI driver shall accept the standard string for that trigger source. This attribute is case insensitive, but case preserving. That is the setting is case insensitive but when reading it back the programmed case is returned. IVI specific drivers may define new trigger source strings for trigger sources that are not defined by IVI-3.3 *Cross Class Capabilities Specification* if needed.

Name	Description	
	Language	Identifier
Internal	No external trigger is used. The pulse period is specified with Pulse Internal Trigger Period.	
	C	IVIRFSIGGEN_VAL_PULSE_TRIGGER_SOURCE_INTERNAL
	COM	IviRFSigGenPulseTriggerSourceInternal
External	The pulse is started with a trigger after the delay time specified with Pulse External Trigger Delay.	
	C	IVIRFSIGGEN_VAL_PULSE_TRIGGER_SOURCE_EXTERNAL
	COM	IviRFSigGenPulseTriggerSourceExternal

### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.



## Compliance Notes

1. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_PULSE_TRIGGER_SOURCE_CLASS_EXT_BASE` and less than `IVIRFSIGGEN_VAL_PULSE_TRIGGER_SOURCE_SPECIFIC_EXT_BASE`.
2. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_PULSE_TRIGGER_SOURCE_SPECIFIC_EXT_BASE`.
3. When an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to Pulse Trigger Source Specific Ext Base.

See Section Attribute Value Definitions, for the definitions of Pulse Trigger Source Specific Ext Base, `IVIRFSIGGEN_VAL_PULSE_TRIGGER_SOURCE_SPECIFIC_EXT_BASE` and `IVIRFSIGGEN_VAL_PULSE_TRIGGER_SOURCE_CLASS_EXT_BASE`.

## 12.2.5 Pulse External Trigger Slope

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32 (C/COM)	R/W	N/A	None	Configure Pulse External Trigger

### .NET Property Name

`PulseGenerator.ExternalTriggerSlope`

### .NET Enumeration Name

`Slope`

### COM Property Name

`PulseGenerator.ExternalTriggerSlope`

### COM Enumeration Name

`IviRFSigGenPulseExternalTriggerSlopeEnum`

### C Constant Name

`IVIRFSIGGEN_ATTR_PULSE_EXTERNAL_TRIGGER_SLOPE`

### Description

Specifies whether the event occurs on the rising or falling edge of the trigger input signal.

### Defined Values

Name	Description		
		Language	Identifier
Positive	Enables rising edge triggering.		
		.NET	<code>Slope.Positive</code>
		C	<code>IVIRFSIGGEN_VAL_PULSE_EXTERNAL_TRIGGER_SLOPE_POSITIVE</code>
		COM	<code>IviRFSigGenPulseExternalTriggerSlopePositive</code>
Negative	Enables falling edge triggering.		
		.NET	<code>Slope.Negative</code>
		C	<code>IVIRFSIGGEN_VAL_PULSE_EXTERNAL_TRIGGER_SLOPE_NEGATIVE</code>
		COM	<code>IviRFSigGenPulseExternalTriggerSlopeNegative</code>

### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## Compliance Notes

1. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_PULSE_EXTERNAL_TRIGGER_CLASS_EXT_BASE` and less than `IVIRFSIGGEN_VAL_PULSE_EXTERNAL_TRIGGER_SPECIFIC_EXT_BASE`.
2. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_PULSE_EXTERNAL_TRIGGER_SPECIFIC_EXT_BASE`.
3. When an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to Pulse External Trigger Specific Ext Base.

See Section Attribute Value Definitions, for the definitions of Pulse External Trigger Specific Ext Base, `IVIRFSIGGEN_VAL_PULSE_EXTERNAL_TRIGGER_SLOPE_SPECIFIC_EXT_BASE` and `IVIRFSIGGEN_VAL_PULSE_EXTERNAL_TRIGGER_SLOPE_CLASS_EXT_BASE`.

## 12.2.6 Pulse External Trigger Delay

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64 (C/COM)	R/W	N/A	None	Configure Pulse External Trigger
PrecisionTimeSpan (.NET)	R/W	N/A	None	Configure Pulse External Trigger

### .NET Property Name

```
PulseGenerator.ExternalTriggerDelay
```

### COM Property Name

```
PulseGenerator.ExternalTriggerDelay
```

### C Constant Name

```
IVIRFSIGGEN_ATTR_PULSE_EXTERNAL_TRIGGER_DELAY
```

### Description

Specifies the delay for starting the output pulse with respect to the trigger input. For C and COM, the units are seconds. For .NET, the units are implicit in the definition of PrecisionTimeSpan.

### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

### **12.3 *IviRFSigGenPulseGenerator Functions***

The IviRFSigGenPulseGenerator extension group defines the following functions:

- Configure Pulse External Trigger
- Configure Pulse Internal Trigger (IVI-C only)
- Configure Pulse

This section describes the behavior and requirements of each function.

## 12.3.1 Configure Pulse External Trigger

### Description

Configures the triggering of the pulse generator within the RF signal generator. Specifies the external trigger slope and the delay time for starting the pulse after the trigger pulse.

### .NET Method Prototype

```
void PulseGenerator.ConfigureExternalTrigger (
    Slope slope,
    Ivi.Driver.PrecisionTimeSpan delay);
```

### COM Method Prototype

```
HRESULT PulseGenerator.ConfigureExternalTrigger (
    [in] IviRFSigGenPulseExternalTriggerSlopeEnum Slope,
    [in] DOUBLE Delay);
```

### C Prototype

```
ViStatus IviRFSigGen_ConfigurePulseExternalTrigger (ViSession Vi,
    ViInt32 Slope,
    ViReal64 Delay);
```

### Parameters

Inputs	Description
Vi	Instrument handle
Slope	Specifies the polarity of the trigger input to start one pulse. The driver uses this value to set the Pulse External Trigger Slope attribute. See the attribute description for more details.
Delay	Specifies the delay for starting the output pulse with respect to the trigger input. The driver uses this value to set the Pulse External Trigger Delay attribute. See the attribute description for more details.

### Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

### .NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

### 12.3.2 Configure Pulse Internal Trigger (IVI-C only)

**Description**

Configures the pulse generator within the RF signal generator. Specifies the period time (repetition rate) in case of internal trigger (free run) mode.

**.NET Method Prototype**

N/A  
(use the `PulseGenerator.InternalTriggerPeriod` property)

**COM Method Prototype**

N/A  
(use the `PulseGenerator.InternalTriggerPeriod` property)

**C Prototype**

```
ViStatus IviRFSigGen_ConfigurePulseInternalTrigger (ViSession Vi,  
                                                    ViReal64 Period);
```

**Parameters**

Inputs	Description
Vi	Instrument handle
Period	Specifies the period (repetition rate) for the pulse. The driver uses this value to set the Pulse Internal Trigger Period attribute. See the attribute description for more details.

**Return Values (C)**

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

### 12.3.3 Configure Pulse

#### Description

Configures the trigger source, pulse width and gating enabled for the pulse generator.

#### .NET Method Prototype

```
void PulseGenerator.Configure (String pulseTriggerSource,  
                               Ivi.Driver.PrecisionTimeSpan pulseWidth,  
                               Boolean gatingEnabled);
```

#### COM Method Prototype

```
HRESULT PulseGenerator.Configure (  
    [in] IviRFSigGenPulseTriggerSourceEnum PulseTriggerSource,  
    [in] DOUBLE PulseWidth,  
    [in] VARIANT_BOOL GatingEnabled);
```

#### C Prototype

```
ViStatus IviRFSigGen_ConfigurePulse (ViSession Vi, ViInt32 PulseTriggerSource,  
                                       ViReal64 PulseWidth, ViBoolean GatingEnabled);
```

#### Parameters

Inputs	Description
Vi	Instrument handle
PulseTriggerSource	Specifies the source of the signal the pulse generator uses to generate one pulse. The driver uses this value to set the Pulse Trigger Source attribute. See the attribute description for more details.
PulseWidth	Specifies the width of the output pulse. The driver uses this value to set the Pulse Width attribute. See the attribute description for more details.
GatingEnabled	Enables or disables gating. The driver uses this value to set the Pulse Gating Enabled attribute. See the attribute description for more details.

#### Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

#### .NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.



## **12.4 *IviRFSigGenPulseGenerator Behavior Model***

The IviRFSigGenPulseGenerator Extension Group follows the behavior model of the IviRFSigGenBase capability group. The only modification to the behavior model from the IviRFSigGenBase capability group is the ability to configure IviRFSigGenPulseGenerator settings.

## **12.5 *IviRFSigGenPulseGenerator Compliance Notes***

1. If an instrument driver implements the IviRFSigGenPulseGenerator Extension Group, it shall also implement the IviRFSigGenModulatePulse Extension Group.
2. If an instrument driver does not implement the value Pulse Trigger Source External for the Pulse Trigger Source attribute, it shall not implement the Pulse External Trigger Delay attribute, the Pulse External Trigger Slope attribute and the Configure Pulse External Trigger function.
3. If an instrument driver does not implement the value Pulse Trigger Source Internal for the Pulse Trigger Source attribute, it shall not implement the Pulse Internal Trigger Period attribute and the Configure Internal Trigger function.

## 13. IviRFSigGenPulseDoubleGenerator Extension Group

---

### 13.1 *IviRFSigGenPulseDoubleGenerator Extension Group Overview*

The IviRFSigGenPulseDoubleGenerator Extension Group extends the IviRFSigGenPulseGenerator Extension Group to support double pulse generation.

### 13.2 *IviRFSigGenPulseDoubleGenerator Attributes*

The IviRFSigGenPulseDoubleGenerator extension group defines the following attributes:

- Pulse Double Enabled
- Pulse Double Delay

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 30, *IviRFSigGen Attribute ID Definitions*.

13.2.1 Pulse Double Enabled

Data Type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	Configure Pulse Double

**.NET Property Name**

`PulseGenerator.DoublePulse.Enabled`

**COM Property Name**

`PulseGenerator.Double.Enabled`

**C Constant Name**

`IVIRFSIGGEN_ATTR_PULSE_DOUBLE_ENABLED`

**Description**

If set to True, the double pulse mode is enabled. If set to False, the double pulse mode is disabled.

**.NET Exceptions**

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

**Compliance Notes**

Instrument drivers shall support the values `True` and `False`.

### 13.2.2 Pulse Double Delay

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64 (C/COM)	R/W	N/A	None	Configure Pulse Double
PrecisionTimeSpan (.NET)	R/W	N/A	None	Configure Pulse Double

#### .NET Property Name

```
PulseGenerator.DoublePulse.Delay
```

#### COM Property Name

```
PulseGenerator.Double.Delay
```

#### C Constant Name

```
IVIRFSIGGEN_ATTR_PULSE_DOUBLE_DELAY
```

#### Description

Specifies the delay of the second pulse from the falling edge of the first pulse. The second pulse has the same width as the first. For C and COM, the units are seconds. For .NET, the units are implicit in the definition of PrecisionTimeSpan.

#### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

### **13.3 *IviRFSigGenPulseDoubleGenerator Functions***

The IviRFSigGenPulseGenerator extension group defines the following function:

- Configure Pulse Double

This section describes the behavior and requirements of this function.

### 13.3.1 Configure Pulse Double

#### Description

Configures the pulse generator within the RF signal generator. Specifies double pulse state and delay.

#### .NET Method Prototype

```
void PulseGenerator.DoublePulse.Configure (Boolean enabled,  
                                           Ivi.Driver.PrecisionTimeSpan delay);
```

#### COM Method Prototype

```
HRESULT PulseGenerator.Double.Configure ([in] VARIANT_BOOL Enabled,  
                                         [in] DOUBLE Delay);
```

#### C Prototype

```
ViStatus IviRFSigGen_ConfigurePulseDouble (ViSession Vi, ViBoolean Enabled,  
                                           ViReal64 Delay);
```

#### Parameters

Inputs	Description
Vi	Instrument handle
Enabled	Enables or disables double pulse mode. The driver uses this value to set the Pulse Double Enabled attribute. See the attribute description for more details.
Delay	Specifies the delay of the second pulse. The driver uses this value to set the Pulse Double Delay attribute. See the attribute description for more details.

#### Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

#### .NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

### **13.4 *IviRFSigGenPulseDoubleGenerator Behavior Model***

The IviRFSigGenPulseDoubleGenerator Extension Group follows the behavior model of the IviRFSigGenBase capability group. The only modification to the behavior model from the IviRFSigGenBase capability group is the ability to configure IviRFSigGenPulseDoubleGenerator settings.

### **13.5 *IviRFSigGenPulseDoubleGenerator Compliance Notes***

1. If a specific driver implements the IviRFSigGenPulseDoubleGenerator Extension Group, it shall also implement the IviRFSigGenPulseGenerator Extension Group.

## 14. IviRFSigGenPulseGeneratorOutput Extension Group

---

### 14.1 *IviRFSigGenPulseGeneratorOutput Extension Group Overview*

The IviRFSigGenPulseGeneratorOutput Extension Group requires the IviRFSigGenPulseGenerator extension group. The output of the pulse generator can be enabled or disabled and its polarity set.

### 14.2 *IviRFSigGenPulseGeneratorOutput Attributes*

The IviRFSigGenPulseGeneratorOutput extension group defines the following attributes:

- Pulse Output Polarity
- Pulse Output Enabled

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 30, *IviRFSigGen Attribute ID Definitions*.



## 14.2.1 Pulse Output Polarity

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	Configure Pulse Output

### .NET Property Name

`PulseGenerator.Output.Polarity`

### .NET Enumeration Name

`PulseOutputPolarity`

### COM Property Name

`PulseGenerator.Output.Polarity`

### COM Enumeration Name

`IviRFSigGenPulseOutputPolarityEnum`

### C Constant Name

`IVIRFSIGGEN_ATTR_PULSE_OUTPUT_POLARITY`

### Description

Specifies the polarity of the output signal.

### Defined Values

Name	Description		
		Language	Identifier
Normal	Specifies normal polarity.		
		.NET	<code>PulseOutputPolarity.Normal</code>
		C	<code>IVIRFSIGGEN_VAL_PULSE_OUTPUT_POLARITY_NORMAL</code>
		COM	<code>IviRFSigGenPulseOutputPolarityNormal</code>
Inverse	Specifies inverted polarity.		
		.NET	<code>PulseOutputPolarity.Inverse</code>
		C	<code>IVIRFSIGGEN_VAL_PULSE_OUTPUT_POLARITY_INVERSE</code>
		COM	<code>IviRFSigGenPulseOutputPolarityInverse</code>

### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## Compliance Notes

1. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_PULSE_OUTPUT_POLARITY_CLASS_EXT_BASE` and less than `IVIRFSIGGEN_VAL_PULSE_OUTPUT_POLARITY_SPECIFIC_EXT_BASE`.
2. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_PULSE_OUTPUT_POLARITY_SPECIFIC_EXT_BASE`.
3. When an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to Pulse Output Polarity Specific Ext Base.

See Section Attribute Value Definitions, for the definitions of Pulse Output Polarity Specific Ext Base, `IVIRFSIGGEN_VAL_PULSE_OUTPUT_POLARITY_SPECIFIC_EXT_BASE` and `IVIRFSIGGEN_VAL_PULSE_OUTPUT_POLARITY_CLASS_EXT_BASE`.

### 14.2.2 Pulse Output Enabled

Data Type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	Configure Pulse Output

#### .NET Property Name

`PulseGenerator.Output.Enabled`

#### COM Property Name

`PulseGenerator.Output.Enabled`

#### COM Enumeration Name

N/A

#### C Constant Name

`IVIRFSIGGEN_ATTR_PULSE_OUTPUT_ENABLED`

#### Description

If set to True, the pulse generator's external output is enabled. If set to False, the pulse generator's external output is disabled.

#### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

#### Compliance Notes

Instrument drivers shall support the values True and False.

### **14.3 *IviRFSigGenPulseGeneratorOutput Functions***

The IviRFSigGenPulseGenerator extension group defines the following functions:

- Configure Pulse Output

This section describes the behavior and requirements of this function.

### 14.3.1 Configure Pulse Output

#### Description

Configures the output and polarity of the pulse generator within the RF signal generator.

#### .NET Method Prototype

```
void PulseGenerator.Output.Configure(PulseOutputPolarity polarity,  
                                     Boolean enabled);
```

#### COM Method Prototype

```
HRESULT PulseGenerator.Output.Configure(  
    [in] IviRFSigGenPulseOutputPolarityEnum Polarity,  
    [in] VARIANT_BOOL Enabled);
```

#### C Prototype

```
ViStatus IviRFSigGen_ConfigurePulseOutput (ViSession Vi, ViInt32 Polarity,  
                                           ViBoolean Enabled);
```

#### Parameters

Inputs	Description
Vi	Instrument handle
Polarity	Specifies the polarity of the output signal. The driver uses this value to set the Pulse Output Polarity attribute. See the attribute description for more details.
Enabled	Enables or disables the external output of the pulse generator. The driver uses this value to set the Pulse Output Enabled attribute. See the attribute description for more details.

#### Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

#### .NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

#### **14.4 IviRFSigGenPulseGeneratorOutput Behavior Mode**

The IviRFSigGenPulseGeneratorOutput Extension Group follows the behavior model of the IviRFSigGenBase capability group. The only modification to the behavior model from the IviRFSigGenBase capability group is the ability to configure IviRFSigGenPulseGeneratorOutput settings.

#### **14.5 IviRFSigGenPulseGeneratorOutput Compliance Notes**

1. If a specific driver implements the IviRFSigGenPulseGeneratorOutput Extension Group, it shall also implement the IviRFSigGenPulseGenerator Extension Group.

## 15. IviRFSigGenSweep Extension Group

---

### 15.1 *IviRFSigGenSweep Overview*

The IviRFSigGenSweep extension group supports signal generators with the ability to sweep (or step) the frequency or the power of the RF output signal.

### 15.2 *IviRFSigGenSweep Attributes*

The IviRFSigGenSweep extension group defines the following attributes:

- Sweep Mode
- Sweep Trigger Source

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 30, *IviRFSigGen Attribute ID Definitions*.

## 15.2.1 Sweep Mode

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	Configure Sweep

### .NET Property Name

`Sweep.Mode`

### .NET Enumeration Name

`SweepMode`

### COM Property Name

`Sweep.Mode`

### COM Enumeration Name

`IviRFSigGenSweepModeEnum`

### C Constant Name

`IVIRFSIGGEN_ATTR_SWEEP_MODE`

### Description

Specifies the sweep mode applied to the output signal.

### Defined Values

Name	Description		
		Language	Identifier
None	The RF output of the signal generator is a non-swept signal (Continuous Wave). Frequency and power level settings from the base capability group are used.		
		.NET	<code>SweepMode.None</code>
		C	<code>IVIRFSIGGEN_VAL_SWEEP_MODE_NONE</code>
		COM	<code>IviRFSigGenSweepModeNone</code>
Frequency Sweep	The signal generator sweeps the RF output signal's frequency in an analog form (non-stepped). Refer to <code>IviRFSigGenFrequencySweep</code> extension group.		
		.NET	<code>SweepMode.FrequencySweep</code>
		C	<code>IVIRFSIGGEN_VAL_SWEEP_MODE_FREQUENCY_SWEEP</code>
		COM	<code>IviRFSigGenSweepModeFrequencySweep</code>
Power Sweep	The signal generator sweeps the RF output signal's power in an analog form (non-stepped). Refer to <code>IviRFSigGenPowerSweep</code> extension group.		
		.NET	<code>SweepMode.PowerSweep</code>
		C	<code>IVIRFSIGGEN_VAL_SWEEP_MODE_POWER_SWEEP</code>
		COM	<code>IviRFSigGenSweepModePowerSweep</code>



Frequency Step	The signal generator sweeps the RF output signal's frequency in steps. Refer to IviRFSigGenFrequencyStep extension group.	
	.NET	SweepMode.FrequencyStep
	C	IVIRFSIGGEN_VAL_SWEEP_MODE_FREQUENCY_STEP
	COM	IviRFSigGenSweepModeFrequencyStep
Power Step	The signal generator sweeps the RF output signal's power level in steps. Refer to IviRFSigGenPowerStep extension group.	
	.NET	SweepMode.PowerStep
	C	IVIRFSIGGEN_VAL_SWEEP_MODE_POWER_STEP
	COM	IviRFSigGenSweepModePowerStep
List	The signal generator uses two lists with frequency and power level values to sweep the RF output signal. Refer to IviRFSigGenList extension group.	
	.NET	SweepMode.List
	C	IVIRFSIGGEN_VAL_SWEEP_MODE_LIST
	COM	IviRFSigGenSweepModeList

## .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## Compliance Notes

1. A specific driver shall implement the defined value None and at least one of the other defined values.
2. If a specific driver implements the defined value Frequency Sweep it shall implement the IviRFSigGenFrequencySweep Extension group.
3. If a specific driver implements the defined value Power Sweep it shall implement the IviRFSigGenPowerSweep Extension group.
4. If a specific driver implements the defined value Frequency Step it shall implement the IviRFSigGenFrequencyStep Extension group.
5. If a specific driver implements the defined value Power Step it shall implement the IviRFSigGenPowerStep extension group.
6. If a specific driver implements the defined value List it shall implement the IviRFSigGenList Extension group.
7. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to IVIRFSIGGEN\_VAL\_SWEEP\_MODE\_CLASS\_EXT\_BASE and less than IVIRFSIGGEN\_VAL\_SWEEP\_MODE\_SPECIFIC\_EXT\_BASE.
8. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to IVIRFSIGGEN\_VAL\_SWEEP\_MODE\_SPECIFIC\_EXT\_BASE.
9. When an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to Sweep Mode Specific Ext Base.

See Section Attribute Value Definitions, for the definitions of AM Sweep Mode Specific Ext Base, IVIRFSIGGEN\_VAL\_SWEEP\_MODE\_SPECIFIC\_EXT\_BASE and IVIRFSIGGEN\_VAL\_SWEEP\_MODE\_CLASS\_EXT\_BASE.

## 15.2.2 Sweep Trigger Source

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32 (C/COM)	R/W	N/A	None	Configure Sweep
ViString (.NET)	R/W	N/A	None	Configure Sweep

### .NET Property Name

`Sweep.TriggerSource`

### COM Property Name

`Sweep.TriggerSource`

### COM Enumeration Name

`IviRFSigGenSweepTriggerSourceEnum`

### C Constant Name

`IVIRFSIGGEN_ATTR_SWEEP_TRIGGER_SOURCE`

### Description

Specifies the trigger used to start a sweep operation.

### Defined Values

In IVI.NET the trigger source is a string. If an IVI driver supports a trigger source and the trigger source is listed in IVI-3.3 *Cross Class Capabilities Specification*, Section 3 then the IVI driver shall accept the standard string for that trigger source. This attribute is case insensitive, but case preserving. That is the setting is case insensitive but when reading it back the programmed case is returned. IVI specific drivers may define new trigger source strings for trigger inputs that are not defined by IVI-3.3 *Cross Class Capabilities Specification* if needed.

Name	Description	
	Language	Identifier
Immediate	The sweep system does not wait for a trigger of any kind, so it is running continuously.	
	C	IVIRFSIGGEN_VAL_SWEEP_TRIGGER_SOURCE_IMMEDIATE
	COM	IviRFSigGenSweepTriggerSourceImmediate
External	The sweep is started with an external signal.	
	C	IVIRFSIGGEN_VAL_SWEEP_TRIGGER_SOURCE_EXTERNAL
	COM	IviRFSigGenSweepTriggerSourceExternal
Software Trigger	The sweep is started with a software programmable trigger.	
	C	IVIRFSIGGEN_VAL_SWEEP_TRIGGER_SOURCE_SOFTWARE
	COM	IviRFSigGenSweepTriggerSourceSoftware

## .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## Compliance Notes

1. If a specific driver implements the value Software Trigger, it shall also implement the IviRFSigGenSoftwareTrigger Extension Group.
2. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_SWEEP_TRIGGER_SOURCE_SPECIFIC_EXT_BASE`.
3. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_SWEEP_TRIGGER_SOURCE_CLASS_EXT_BASE` and less than `IVIRFSIGGEN_VAL_SWEEP_TRIGGER_SOURCE_SPECIFIC_EXT_BASE`.
4. When an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to Sweep Trigger Source Specific Ext Base.

See Section Attribute Value Definitions, for the definitions of Sweep Trigger Source Specific Ext Base, `IVIRFSIGGEN_VAL_SWEEP_TRIGGER_SOURCE_SPECIFIC_EXT_BASE` and `IVIRFSIGGEN_VAL_SWEEP_TRIGGER_SOURCE_CLASS_EXT_BASE`.

### **15.3 *IviRFSigGenSweep Functions***

The IviRFSigGenSweep extension group defines the following functions:

- Configure Sweep

This section describes the behavior and requirements of each function.

## 15.3.1 Configure Sweep

### Description

Configures the signal generator whether the RF output signal is a continuous wave or the frequency, the power level or both are swept or stepped.

### .NET Method Prototype

```
void Sweep.Configure (SweepMode mode,  
                     String triggerSource);
```

### COM Method Prototype

```
HRESULT Sweep.Configure([in] IviRFSigGenSweepModeEnum Mode,  
                       [in] IviRFSigGenSweepTriggerSourceEnum TriggerSource);
```

### C Prototype

```
ViStatus IviRFSigGen_ConfigureSweep (ViSession Vi, ViInt32 Mode,  
                                     ViInt32 TriggerSource);
```

### Parameters

Inputs	Description
Vi	Instrument handle.
Mode	Specifies the sweep mode of the RF signal generator. The driver uses this value to set the Sweep Mode attribute. See the attribute description for more details.
TriggerSource	Specifies the way to start the sweep or running it continuously. The driver uses this value to set the Sweep Trigger Source attribute. See the attribute description for more details.

### Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

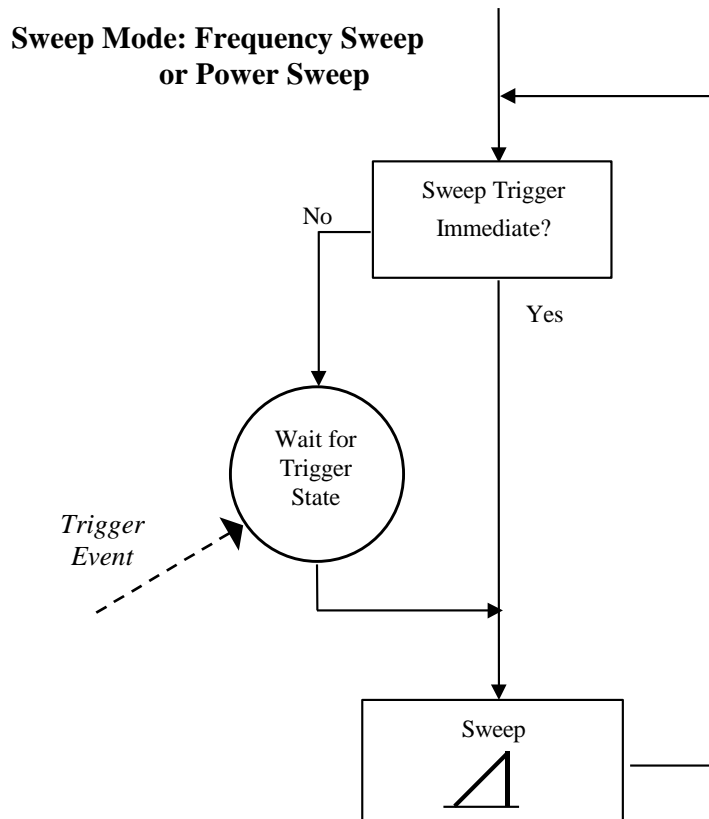
### .NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

## 15.4 IviRFSigGenSweep Behavior Model

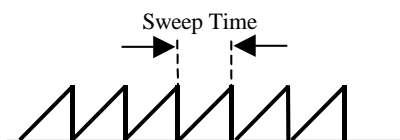
The following behavior models show the relationship between the IviRFSigGenSweep extension group and RF Signal Generator behavior.

### 15.4.1 Frequency Sweep / Power Sweep

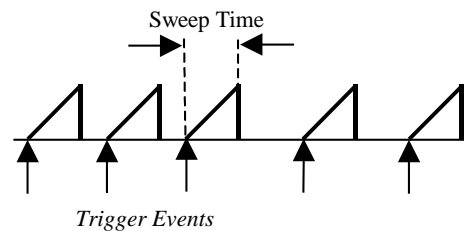


Setting the Sweep Trigger attribute to Immediate will continuously generate sweeps (either frequency or power sweeps). The duration of one sweep from start to stop is defined with Frequency Sweep Time or Power Sweep Time. Setting the Sweep Trigger attribute to External or Software Trigger will delay the start of the next sweep until the specified trigger event occurs.

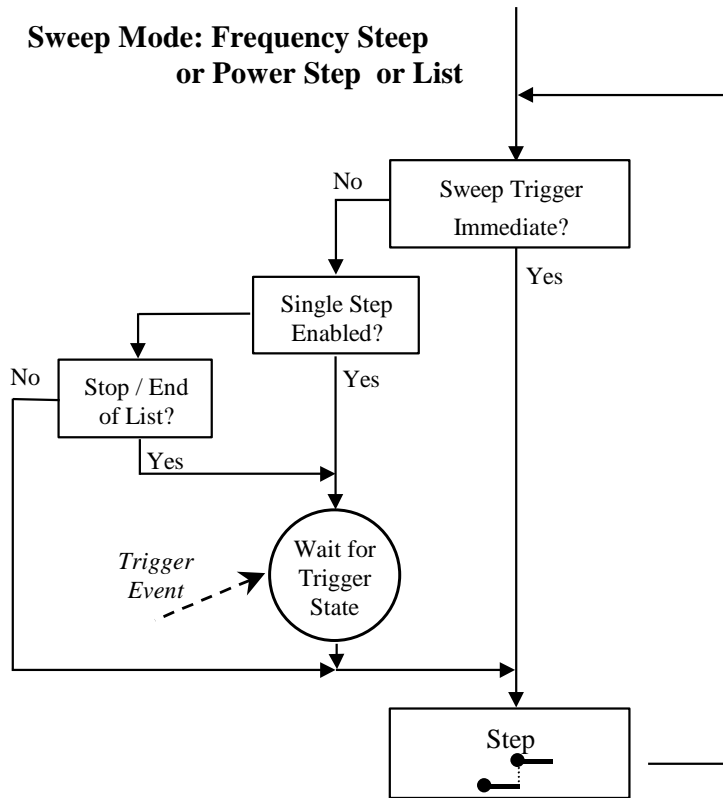
Trigger Immediate:



Trigger External / Software:

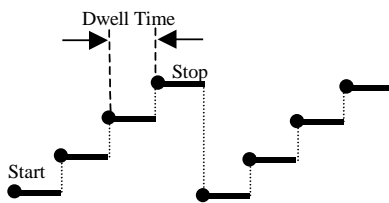


## 15.4.2 Frequency Step / Power Step / List

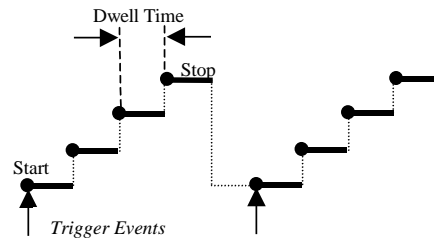


Setting the Sweep Trigger attribute to Immediate will continuously generate steps (frequency or power) either from start to stop with fixed increments (steps) or from a list of arbitrary values. The duration of one step is defined with Dwell Time. Setting the Sweep Trigger attribute to External or Software Trigger will delay the start of the next step until the specified trigger event occurs.

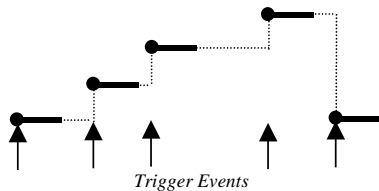
Trigger Immediate:



Trigger External / Software:



Single Step and Trigger External / Software:





In addition the following rules apply:

1. When a call to Configure RF of the base capability group is made, the Sweep Mode attribute is set to None, hence stopping any sweeps that may have been in progress.
2. When the Sweep Mode attribute is set to Frequency Sweep, Frequency Step or List Mode with frequency or frequency and power list selected, setting Frequency of the base capability group will set the Sweep Mode attribute to None. Setting the value of Power Level will control the power level of the swept signal.
3. When the Sweep Mode attribute is set to Power Sweep, Power Step or List Mode with power or frequency and power level list selected, setting Power Level of the base capability group will set the Sweep Mode attribute to None. Setting the value of Frequency will control the frequency of the swept signal.
4. When the Sweep Mode attribute is set to List and a frequency and power level list is selected, setting Frequency or Power Level of the base capability group will set the Sweep Mode attribute to None.
5. When the Sweep Mode attribute is changed from any sweep mode to None, the signal generator will generate the signal accordingly to the previously set values of Frequency and Power Level of the base capability group.

## 16. IviRFSigGenFrequencySweep Extension Group

---

### 16.1 *IviRFSigGenFrequencySweep Extension Group Overview*

The IviRFSigGenFrequencySweep Extension Group supports signal generators that can apply a frequency sweep to the output signal. The user may configure the sweep with start and stop, or center and span frequencies. The sweep time is also configurable.

In order to support this extension group, a driver shall first support the IviRFSigGenSweep Extension Group. This extension group is active when the Sweep Mode attribute is set to Frequency Sweep.

### 16.2 *IviRFSigGenFrequencySweep Attributes*

The IviRFSigGenFrequencySweep extension group defines the following attributes:

- Frequency Sweep Start
- Frequency Sweep Stop
- Frequency Sweep Time

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 30, *IviRFSigGen Attribute ID Definitions*.

16.2.1 Frequency Sweep Start

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	Configure Frequency Sweep Start Stop, Configure Frequency Sweep Center Span

**.NET Property Name**

`Sweep.FrequencySweep.Start`

**COM Property Name**

`Sweep.FrequencySweep.Start`

**C Constant Name**

`IVIRFSIGGEN_ATTR_FREQUENCY_SWEEP_START`

**Description**

Specifies the start frequency of the sweep. If the stop frequency is less than the start frequency, the frequency decreases during the sweep. The units are Hertz.

**.NET Exceptions**

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## 16.2.2 Frequency Sweep Stop

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	Configure Frequency Sweep Start Stop Configure Frequency Sweep Center Span

### .NET Property Name

`Sweep.FrequencySweep.Stop`

### COM Property Name

`Sweep.FrequencySweep.Stop`

### C Constant Name

`IVIRFSIGGEN_ATTR_FREQUENCY_SWEEP_STOP`

### Description

Specifies the stop frequency of the sweep. If the stop frequency is less than the start frequency, the frequency decreases during the sweep. The units are Hertz.

### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

### 16.2.3 Frequency Sweep Time

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64 (C/COM)	R/W	N/A	None	Configure Frequency Sweep Time
PrecisionTimeSpan (.NET)	R/W	N/A	None	Configure Frequency Sweep Time

#### .NET Property Name

`Sweep.FrequencySweep.Time`

#### COM Property Name

`Sweep.FrequencySweep.Time`

#### C Constant Name

`IVIRFSIGGEN_ATTR_FREQUENCY_SWEEP_TIME`

#### Description

Specifies the duration of one sweep from start to stop frequency. For C and COM, the units are seconds. For .NET, the units are implicit in the definition of PrecisionTimeSpan.

#### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

### **16.3 *IviRFSigGenFrequencySweep Functions***

The IviRFSigGenFrequencySweep extension group defines the following functions:

- Configure Frequency Sweep Start Stop
- Configure Frequency Sweep Center Span
- Configure Frequency Sweep Time (IVI-C only)

This section describes the behavior and requirements of each function.

## 16.3.1 Configure Frequency Sweep Start Stop

### Description

Configures the attributes that control the sweep frequencies of the generator's output signal. These attributes are start and stop frequency. If the stop frequency is less than the start frequency, the frequency decreases during the sweep.

### .NET Method Prototype

```
void Sweep.FrequencySweep.ConfigureStartStop (Double start,  
                                              Double stop);
```

### COM Method Prototype

```
HRESULT Sweep.FrequencySweep.ConfigureStartStop ([in] DOUBLE Start,  
                                              [in] DOUBLE Stop);
```

### C Prototype

```
ViStatus IviRFSigGen_ConfigureFrequencySweepStartStop (ViSession Vi,  
                                                      ViReal64 Start,  
                                                      ViReal64 Stop);
```

### Parameters

Inputs	Description
Vi	Instrument handle
Start	Specifies the start frequency of the sweep. The driver uses this value to set the Frequency Sweep Start attribute. See the attribute description for more details.
Stop	Specifies the stop frequency of the sweep. The driver uses this value to set the Frequency Sweep Stop attribute. See the attribute description for more details.

### Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

### .NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

## 16.3.2 Configure Frequency Sweep Center Span

### Description

Configures the attributes that control the sweep frequencies of the generator's output signal. These attributes are center and span frequency. This function modifies the attributes as follows:

Frequency Sweep Start = Center – Span / 2

Frequency Sweep Stop = Center + Span / 2

### .NET Method Prototype

```
void Sweep.FrequencySweep.ConfigureCenterSpan (Double center,  
                                                Double span);
```

### COM Method Prototype

```
HRESULT Sweep.FrequencySweep.ConfigureCenterSpan ([in] DOUBLE Center,  
                                                  [in] DOUBLE Span);
```

### C Prototype

```
ViStatus IviRFSigGen_ConfigureFrequencySweepCenterSpan (ViSession Vi,  
                                                         ViReal64 Center,  
                                                         ViReal64 Span);
```

### Parameters

Inputs	Description
Vi	Instrument handle
Center	Specifies the center frequency of the sweep. The driver uses this value to set the Frequency Sweep Start and Frequency Sweep Stop attribute. See the attribute description and function description for more details.
Span	Specifies the frequency span of the sweep. The driver uses this value to set the Frequency Sweep Start and Frequency Sweep Stop attribute. See the attribute description and function description for more details.

### Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

### .NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.



### 16.3.3 Configure Frequency Sweep Time (IVI-C only)

**Description**

Configures the duration of one frequency sweep.

**.NET Method Prototype**

N/A  
(use the `Sweep.FrequencySweep.Time` property)

**COM Method Prototype**

N/A  
(use the `Sweep.FrequencySweep.Time` property)

**C Prototype**

```
ViStatus IviRFSigGen_ConfigureFrequencySweepTime (ViSession Vi, ViReal64 Time);
```

**Parameters**

Inputs	Description
Vi	Instrument handle
Time	Specifies the duration of the frequency sweep. The driver uses this value to set the Frequency Sweep Time attribute. See the attribute description for more details.

**Return Values (C)**

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

#### **16.4 *IviRFSigGenFrequencySweep Behavior Model***

The IviRFSigGenFrequencySweep Extension Group follows the behavior model of the IviRFSigGenSweep capability group.

#### **16.5 *IviRFSigGenFrequencySweep Compliance Notes***

1. If a specific driver implements the IviRFSigGenFrequencySweep Extension Group, it shall also implement the IviRFSigGenSweep Extension Group and support the defined value Frequency Sweep for the Sweep Mode attribute.

## 17. IviRFSigGenPowerSweep Extension Group

---

### 17.1 *IviRFSigGenPowerSweep Extension Group Overview*

The IviRFSigGenPowerSweep Extension Group supports signal generators that can apply a power sweep to the output signal. The user may configure the sweep with start and stop power. The sweep time is also configurable.

In order to support this extension group, a driver shall first support the IviRFSigGenSweep Extension Group. This extension group is active when the Sweep Mode attribute is set to Power Sweep.

### 17.2 *IviRFSigGenPowerSweep Attributes*

The IviRFSigGenPowerSweep extension group defines the following attributes:

- Power Sweep Start
- Power Sweep Stop
- Power Sweep Time

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 30, *IviRFSigGen Attribute ID Definitions*.

17.2.1 Power Sweep Start

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	Configure Power Sweep Start Stop

**.NET Property Name**

`Sweep.PowerSweep.Start`

**COM Property Name**

`Sweep.PowerSweep.Start`

**C Constant Name**

`IVIRFSIGGEN_ATTR_POWER_SWEEP_START`

**Description**

Specifies the start power of the sweep. If the stop power is less than the start power, the power decreases in value during the sweep. The units are dBm.

**.NET Exceptions**

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

17.2.2 Power Sweep Stop

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	Configure Power Sweep Start Stop

**.NET Property Name**

`Sweep.PowerSweep.Stop`

**COM Property Name**

`Sweep.PowerSweep.Stop`

**C Constant Name**

`IVIRFSIGGEN_ATTR_POWER_SWEEP_STOP`

**Description**

Specifies the stop power of the sweep. If the stop power is less than the start power, the power decreases in value during the sweep. The units are dBm.

**.NET Exceptions**

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

### 17.2.3 Power Sweep Time

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64 (C/COM)	R/W	N/A	None	Configure Power Sweep Time
PrecisionTimeSpan (.NET)	R/W	N/A	None	Configure Power Sweep Time

#### .NET Property Name

`Sweep.PowerSweep.Time`

#### COM Property Name

`Sweep.PowerSweep.Time`

#### COM Enumeration Name

N/A

#### C Constant Name

`IVIRFSIGGEN_ATTR_POWER_SWEEP_TIME`

#### Description

Specifies the duration of one sweep from start to stop power. For C and COM the units are seconds. For .NET, the units are implicit in the definition of PrecisionTimeSpan.

#### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

### **17.3 *IviRFSigGenPowerSweep Functions***

The IviRFSigGenPowerSweep extension group defines the following functions:

- Configure Power Sweep Start Stop
- Configure Power Sweep Time (IVI-C only)

This section describes the behavior and requirements of each function.

## 17.3.1 Configure Power Sweep Start Stop

### Description

Configures the attributes that control the power sweep of the generator's output signal. These attributes are start and stop power. If the stop power is less than the start power, the power decreases in value during the sweep.

### .NET Method Prototype

```
void Sweep.PowerSweep.ConfigureStartStop (Double start,  
                                           Double stop);
```

### COM Method Prototype

```
HRESULT Sweep.PowerSweep.ConfigureStartStop ([in] DOUBLE Start,  
                                              [in] DOUBLE Stop);
```

### C Prototype

```
ViStatus IviRFSigGen_ConfigurePowerSweepStartStop (ViSession Vi,  
                                                    ViReal64 Start,  
                                                    ViReal64 Stop);
```

### Parameters

Inputs	Description
Vi	Instrument handle
Start	Specifies the start power of the sweep. The driver uses this value to set the Power Sweep Start attribute. See the attribute description for more details.
Stop	Specifies the stop power of the sweep. The driver uses this value to set the Power Sweep Stop attribute. See the attribute description for more details.

### Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

### .NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.



17.3.2 onfigure Power Sweep Time (IVI-C only)

Description

Configures the duration of one power sweep.

.NET Method Prototype

N/A  
(use the `Sweep.PowerSweep.Time` property)

COM Method Prototype

N/A  
(use the `Sweep.PowerSweep.Time` property)

C Prototype

```
ViStatus IviRFSigGen_ConfigurePowerSweepTime (ViSession Vi, ViReal64 Time);
```

Parameters

Inputs	Description
Vi	Instrument handle
Time	Specifies the duration of the power sweep. The driver uses this value to set the <code>Power Sweep Time</code> attribute. See the attribute description for more details.

Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

#### **17.4 IviRFSigGenPowerSweep Behavior Model**

The IviRFSigGenPowerSweep Extension Group follows the behavior model of the IviRFSigGenSweep capability group.

#### **17.5 IviRFSigGenPowerSweep Compliance Notes**

1. If a specific driver implements the IviRFSigGenPowerSweep Extension Group, it shall also implement the IviRFSigGenSweep Extension Group and support the defined value Power Sweep for the Sweep Mode attribute.

## 18. IviRFSigGenFrequencyStep Extension Group

---

### 18.1 *IviRFSigGenFrequencyStep Extension Group Overview*

The IviRFSigGenFrequencyStep Extension Group supports signal generators that can vary (sweep) the frequency of the RF output signal in steps. The user can specify the start, stop and step frequency and set linear or logarithmic spacing. Setting single step and dwell time are also included.

This extension group requires the Sweep Extension Group. Frequency stepping is enabled by setting the Sweep Mode to Frequency Step in the IviRFSigGenSweep Extension Group.

### 18.2 *IviRFSigGenFrequencyStep Attributes*

The IviRFSigGenFrequencyStep extension group defines the following attributes:

- Frequency Step Start
- Frequency Step Stop
- Frequency Step Scaling
- Frequency Step Size
- Frequency Step Single Step Enabled
- Frequency Step Dwell

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 30, *IviRFSigGen Attribute ID Definitions*.

18.2.1 Frequency Step Start

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	Configure Frequency Step Start Stop

**.NET Property Name**

`Sweep.FrequencyStep.Start`

**COM Property Name**

`Sweep.FrequencyStep.Start`

**C Constant Name**

`IVIRFSIGGEN_ATTR_FREQUENCY_STEP_START`

**Description**

Specifies the start frequency of the stepped sweep. If the stop frequency is less than the start frequency, the frequency decreases during the sweep. The units are Hertz.

**.NET Exceptions**

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

18.2.2 Frequency Step Stop

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	Configure Frequency Step Start Stop

**.NET Property Name**

`Sweep.FrequencyStep.Stop`

**COM Property Name**

`Sweep.FrequencyStep.Stop`

**C Constant Name**

`IVIRFSIGGEN_ATTR_FREQUENCY_STEP_STOP`

**Description**

Specifies the stop frequency of the stepped sweep. If the stop frequency is less than the start frequency, the frequency decreases during the sweep. The units are Hertz.

**.NET Exceptions**

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

### 18.2.3 Frequency Step Scaling

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	Configure Frequency Step Start Stop

#### .NET Property Name

`Sweep.FrequencyStep.Scaling`

#### COM Enumeration Name

`FrequencyStepScaling`

#### COM Property Name

`Sweep.FrequencyStep.Scaling`

#### COM Enumeration Name

`IviRFSigGenFrequencyStepScalingEnum`

#### C Constant Name

`IVIRFSIGGEN_ATTR_FREQUENCY_STEP_SCALING`

#### Description

Specifies the spacing of the steps.

#### Defined Values

Name	Description		
		Language	Identifier
Linear	Enables linear scaling.		
		.NET	<code>FrequencyStepScaling.Linear</code>
		C	<code>IVIRFSIGGEN_VAL_FREQUENCY_STEP_SCALING_LINEAR</code>
		COM	<code>IviRFSigGenFrequencyStepScalingLinear</code>
Logarithmic	Enables logarithmic scaling.		
		.NET	<code>FrequencyStepScaling.Logarithmic</code>
		C	<code>IVIRFSIGGEN_VAL_FREQUENCY_STEP_SCALING_LOGARITHMIC</code>
		COM	<code>IviRFSigGenFrequencyStepScalingLogarithmic</code>

#### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## Compliance Notes

1. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_FREQUENCY_STEP_SCALING_CLASS_EXT_BASE` and less than `IVIRFSIGGEN_VAL_FREQUENCY_STEP_SCALING_SPECIFIC_EXT_BASE`.
2. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_FREQUENCY_STEP_SCALING_SPECIFIC_EXT_BASE`.
3. When an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to Frequency Step Scaling Specific Ext Base.

See Section Attribute Value Definitions, for the definitions of Frequency Step Scaling Specific Ext Base, `IVIRFSIGGEN_VAL_FREQUENCY_STEP_SCALING_SPECIFIC_EXT_BASE` and `IVIRFSIGGEN_VAL_FREQUENCY_STEP_SCALING_CLASS_EXT_BASE`.

18.2.4 Frequency Step Size

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	Configure Frequency Step Start Stop

**.NET Property Name**

`Sweep.FrequencyStep.Size`

**COM Property Name**

`Sweep.FrequencyStep.Size`

**C Constant Name**

`IVIRFSIGGEN_ATTR_FREQUENCY_STEP_SIZE`

**Description**

Specifies the step size. The units are Hertz if Frequency Step Scaling attribute is set to Linear and is unitless (factor) if Frequency Step Scaling attribute is set to Logarithmic .

**.NET Exceptions**

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.



### 18.2.5 Frequency Step Single Step Enabled

Data Type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	Configure Frequency Step Dwell

#### .NET Property Name

`Sweep.FrequencyStep.SingleStepEnabled`

#### COM Property Name

`Sweep.FrequencyStep.SingleStepEnabled`

#### C Constant Name

`IVIRFSIGGEN_ATTR_FREQUENCY_STEP_SINGLE_STEP_ENABLED`

#### Description

If set to True, Frequency Step Single Step is enabled, and the frequency sweep will advance when the next trigger event occurs. If set to False, Frequency Step Single Step is disabled, and the frequency sweep will advance immediately after the dwell time ends.

#### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## 18.2.6 Frequency Step Dwell

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64 (C/COM)	R/W	N/A	None	Configure Frequency Step Dwell
PrecisionTimeSpan (.NET)	R/W	N/A	None	Configure Frequency Step Dwell

### .NET Property Name

`Sweep.FrequencyStep.Dwell`

### COM Property Name

`Sweep.FrequencyStep.Dwell`

### C Constant Name

`IVIRFSIGGEN_ATTR_FREQUENCY_STEP_DWELL`

### Description

Specifies the duration of one step. Dwell time starts immediately after trigger or next step; no settling time is added. This attribute is ignored if the Frequency Step Single Step Enabled attribute is set to True.

For C and COM, the units are seconds. For .NET, the units are implicit in the definition of PrecisionTimeSpan.

### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

### **18.3 *IviRFSigGenFrequencyStep Functions***

The IviRFSigGenFrequencyStep extension group defines the following functions:

- Configure Frequency Step Start Stop
- Configure Frequency Step Dwell
- Reset Frequency Step

This section describes the behavior and requirements of each function.

### 18.3.1 Configure Frequency Step Start Stop

#### Description

Configures the attributes that control the step frequencies of the generator's RF output signal. These attributes are start and stop frequency, step size and lin/log scaling. If the stop frequency is less than the start frequency, the frequency decreases during the sweep.

#### .NET Method Prototype

```
void Sweep.FrequencyStep.ConfigureStartStop (Double start,  
                                              Double stop),  
                                              FrequencyStepScaling scaling,  
                                              Double stepSize);
```

#### COM Method Prototype

```
HRESULT Sweep.FrequencyStep.ConfigureStartStop ([in] DOUBLE Start,  
                                                [in] DOUBLE Stop),  
                                                [in] IviRFSigGenFrequencyStepScalingEnum Scaling,  
                                                [in] DOUBLE StepSize);
```

#### C Prototype

```
ViStatus IviRFSigGen_ConfigureFrequencyStepStartStop (ViSession Vi,  
                                                      ViReal64 Start,  
                                                      ViReal64 Stop,  
                                                      ViInt32 Scaling,  
                                                      ViReal64 StepSize);
```

#### Parameters

Inputs	Description
Vi	Instrument handle
Start	Specifies the start frequency of the step sweep. The driver uses this value to set the Frequency Step Start attribute. See the attribute description for more details.
Stop	Specifies the stop frequency of the step sweep. The driver uses this value to set the Frequency Step Stop attribute. See the attribute description for more details.
Scaling	Specifies the scaling of the step sweep. The driver uses this value to set the Frequency Step Scaling attribute. See the attribute description for more details.
StepSize	Specifies the size of one step. The driver uses this value to set the Frequency Step Size attribute. See the attribute description for more details.

#### Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

#### .NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

## 18.3.2 Configure Frequency Step Dwell

### Description

Configures the attributes that control frequency stepping.

### .NET Method Prototype

```
void Sweep.FrequencyStep.ConfigureDwell (Boolean singleStepEnabled  
                                         PrecisionTimeSpan dwell);
```

### COM Method Prototype

```
HRESULT Sweep.FrequencyStep.ConfigureDwell ([in] VARIANT_BOOL SingleStepEnabled  
                                             [in] DOUBLE Dwell);
```

### C Prototype

```
ViStatus IviRFSigGen_ConfigureFrequencyStepDwell (ViSession Vi,  
                                                  ViBoolean SingleStepEnabled,  
                                                  ViReal64 Dwell);
```

### Parameters

Inputs	Description
Vi	Instrument handle
SingleStepEnabled	Specifies whether the trigger initiates the next step. The driver uses this value to set the Frequency Step Single Step Enabled attribute. See the attribute description for more details.
Dwell	Specifies the duration of one frequency step. The driver uses this value to set the Frequency Step Dwell attribute. See the attribute description for more details.

### Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

### .NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

### 18.3.3 Reset Frequency Step

**Description**

Resets the current frequency step to the frequency step start value.

**.NET Method Prototype**

```
void Sweep.FrequencyStep.Reset ();
```

**COM Method Prototype**

```
HRESULT Sweep.FrequencyStep.Reset ();
```

**C Prototype**

```
ViStatus IviRFSigGen_ResetFrequencyStep (ViSession Vi);
```

**Parameters**

Inputs	Description
Vi	Instrument handle

**Return Values (C/COM)**

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

**.NET Exceptions**

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

#### **18.4 *IviRFSigGenFrequencyStep Behavior Model***

The IviRFSigGenFrequencyStep Extension Group follows the behavior model of the IviRFSigGenSweep capability group.

#### **18.5 *IviRFSigGenFrequencyStep Compliance Notes***

1. If a specific driver implements the IviRFSigGenFrequencyStep Extension Group, it shall also implement the IviRFSigGenSweep Extension Group and support the defined value Frequency Step for the Sweep Mode attribute.

## 19. IviRFSigGenPowerStep Extension Group

---

### 19.1 *IviRFSigGenPowerStep Extension Group Overview*

The IviRFSigGenPowerStep Extension Group supports signal generators that can vary (sweep) the power of the RF output signal in steps. The user can enable or disable stepping, specify the start, stop and step power. Setting single step and dwell time are also included.

This extension group requires the Sweep Extension Group. Power stepping is enabled by setting the Sweep Mode to Power Step in the IviRFSigGenSweep Extension Group.

### 19.2 *IviRFSigGenPowerStep Attributes*

The IviRFSigGenPowerStep extension group defines the following attributes:

- Power Step Start
- Power Step Stop
- Power Step Size
- Power Step Single Step Enabled
- Power Step Dwell

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 30, *IviRFSigGen Attribute ID Definitions*.



19.2.1 Power Step Start

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	Configure Power Step Start Stop

**.NET Property Name**

`Sweep.PowerStep.Start`

**COM Property Name**

`Sweep.PowerStep.Start`

**C Constant Name**

`IVIRFSIGGEN_ATTR_POWER_STEP_START`

**Description**

Specifies the start power of the stepped sweep. If the stop power is less than the start power, the power decreases in value during the sweep. The units are dBm.

**.NET Exceptions**

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

19.2.2 Power Step Stop

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	Configure PowerStep Start Stop

**.NET Property Name**

`Sweep.PowerStep.Stop`

**COM Property Name**

`Sweep.PowerStep.Stop`

**C Constant Name**

`IVIRFSIGGEN_ATTR_POWER_STEP_STOP`

**Description**

Specifies the stop power of the stepped sweep. If the stop power is less than the start power, the power decreases in value during the sweep. The units are dBm.

**.NET Exceptions**

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

19.2.3 Power Step Size

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	Configure Power Step Start Stop

**.NET Property Name**

`Sweep.PowerStep.Size`

**COM Property Name**

`Sweep.PowerStep.Size`

**C Constant Name**

`IVIRFSIGGEN_ATTR_POWER_STEP_SIZE`

**Description**

Specifies the step size. The units are dBm.

**.NET Exceptions**

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## 19.2.4 Power Step Single Step Enabled

Data Type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	Configure Power Step Dwell

### .NET Property Name

`Sweep.PowerStep.SingleStepEnabled`

### COM Property Name

`Sweep.PowerStep.SingleStepEnabled`

### C Constant Name

`IVIRFSIGGEN_ATTR_POWER_STEP_SINGLE_STEP_ENABLED`

### Description

If set to True, the power sweep will advance when the next trigger event occurs. If set to False, the power sweep will advance immediately after the dwell time ends.

### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## 19.2.5 Power Step Dwell

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64 (C/COM)	R/W	N/A	None	Configure Power Step Dwell
PrecisionTimeSpan (.NET)	R/W	N/A	None	Configure Power Step Dwell

### .NET Property Name

`Sweep.PowerStep.Dwell`

### COM Property Name

`Sweep.PowerStep.Dwell`

### C Constant Name

`IVIRFSIGGEN_ATTR_POWER_STEP_DWELL`

### Description

Specifies the duration of one step. This attribute is ignored if `Power Step Single Step Enabled` is set to `True`.

For C and COM, the units are seconds. For .NET, the units are implicit in the definition of `PrecisionTimeSpan`.

### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

### **19.3 *IviRFSigGenPowerStep Functions***

The IviRFSigGenPowerStep extension group defines the following functions:

- Configure Power Step Start Stop
- Configure Power Step Dwell
- Reset Power Step

This section describes the behavior and requirements of each function.

### 19.3.1 Configure Power Step Start Stop

#### Description

Configures the attributes that control the power steps of the generator's RF output signal. These attributes are start and stop power and step size. If the stop power is less than the start power, the power decreases in value during the sweep.

#### .NET Method Prototype

```
void Sweep.PowerStep.ConfigureStartStop (Double start,  
                                         Double stop,  
                                         Double stepSize);
```

#### COM Method Prototype

```
HRESULT Sweep.PowerStep.ConfigureStartStop ([in] DOUBLE Start,  
                                             [in] DOUBLE Stop,  
                                             [in] DOUBLE StepSize);
```

#### C Prototype

```
ViStatus IviRFSigGen_ConfigurePowerStepStartStop (ViSession Vi,  
                                                  ViReal64 Start,  
                                                  ViReal64 Stop,  
                                                  ViReal64 StepSize);
```

#### Parameters

Inputs	Description
Vi	Instrument handle
Start	Specifies the start power of the stepping. The driver uses this value to set the PowerStep Start attribute. See the attribute description for more details.
Stop	Specifies the stop power of the stepping. The driver uses this value to set the Power Step Stop attribute. See the attribute description for more details.
StepSize	Specifies the size of one step. The driver uses this value to set the Power Step Size attribute. See the attribute description for more details.

#### Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

#### .NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

## 19.3.2 Configure Power Step Dwell

### Description

Configures the attributes that control power stepping.

### .NET Method Prototype

```
void Sweep.PowerStep.ConfigureDwell (Boolean singleStepEnabled,  
                                     PrecisionTimeSpan dwell);
```

### COM Method Prototype

```
HRESULT Sweep.PowerStep.ConfigureDwell ([in] VARIANT_BOOL SingleStepEnabled,  
                                         [in] DOUBLE Dwell);
```

### C Prototype

```
ViStatus IviRFSigGen_ConfigurePowerStepDwell (ViSession Vi,  
                                              ViBoolean SingleStepEnabled,  
                                              ViReal64 Dwell);
```

### Parameters

Inputs	Description
Vi	Instrument handle
SingleStepEnabled	Specifies whether the trigger initiates the next step. The driver uses this value to set the Power Step Single Step Enabled attribute. See the attribute description for more details.
Dwell	Specifies the duration of one power step. The driver uses this value to set the Power Step Dwell attribute. See the attribute description for more details.

### Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

### .NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.



### 19.3.3 Reset Power Step

**Description**

Resets the current power step to the power step start value.

**.NET Method Prototype**

```
void Sweep.PowerStep.Reset ();
```

**COM Method Prototype**

```
HRESULT Sweep.PowerStep.Reset ();
```

**C Prototype**

```
ViStatus IviRFSigGen_ResetPowerStep (ViSession Vi);
```

**Parameters**

Inputs	Description
Vi	Instrument handle

**Return Values (C/COM)**

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

**.NET Exceptions**

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

#### **19.4 *IviRFSigGenPowerStep Behavior Model***

The IviRFSigGenPowerStep Extension Group follows the behavior model of the IviRFSigGenSweep capability group.

#### **19.5 *IviRFSigGenPowerStep Compliance Notes***

1. If a specific driver implements the IviRFSigGenPowerStep Extension Group, it shall also implement the IviRFSigGenSweep Extension Group and support the defined value Power Step for the Sweep Mode attribute.

## 20. IviRFSigGenList Extension Group

---

### 20.1 *IviRFSigGenList Extension Group Overview*

The IviRFSigGenList Extension Group supports signal generators that can set the frequency and power of the RF output signal to values given as a list of values. The user can enable or disable stepping the frequency and power list, specify the name of the list and set its values. The active list can be selected using the list name. Setting single step and dwell time are also included.

This extension group requires the Sweep Extension Group. List stepping is enabled by setting the Sweep Mode to List in the IviRFSigGenSweep Extension Group.

### 20.2 *IviRFSigGenList Attributes*

The IviRFSigGenList extension group defines the following attributes:

- List Selected Name
- List Single Step Enabled
- List Dwell

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 30, *IviRFSigGen Attribute ID Definitions*.

### 20.2.1 List Selected Name

Data Type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	Select List

#### .NET Property Name

`Sweep.List.SelectedList`

#### COM Property Name

`Sweep.List.SelectedName`

#### C Constant Name

`IVIRFSIGGEN_ATTR_LIST_SELECTED_NAME`

#### Description

Specifies the name of the selected list to become active. The name shall be one of the lists created.

#### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## 20.2.2 List Single Step Enabled

Data Type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	Configure List Dwell

### .NET Property Name

`Sweep.List.SingleStepEnabled`

### COM Property Name

`Sweep.List.SingleStepEnabled`

### C Constant Name

`IVIRFSIGGEN_ATTR_LIST_SINGLE_STEP_ENABLED`

### Description

If set to True, the list will advance when the next trigger event occurs. If set to False, the list will advance immediately after the dwell time ends.

### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

### 20.2.3 List Dwell

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64 (C/COM)	R/W	N/A	None	Configure List Dwell
PrecisionTimeSpan (.NET)	R/W	N/A	None	Configure List Dwell

#### .NET Property Name

`Sweep.List.Dwell`

#### COM Property Name

`Sweep.List.Dwell`

#### C Constant Name

`IVIRFSIGGEN_ATTR_LIST_DWELL`

#### Description

Specifies the duration of one step. This attribute is ignored if the List Single Step Enabled attribute is set to True.

For C and COM, the units are seconds. For .NET, units are implicit in the definition of PrecisionTimeSpan.

#### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

### **20.3 *IviRFSigGenList Functions***

The IviRFSigGenList extension group defines the following functions:

- Create Frequency List
- Create Power List
- Create Frequency Power List
- Select List (IVI-C only)
- Clear All Lists
- Configure List Dwell
- Reset List

This section describes the behavior and requirements of each function.

## 20.3.1 Create Frequency List

### Description

Creates a named list of frequency values.

### .NET Method Prototype

```
void Sweep.List.CreateFrequency (String name,  
                                Double[] frequency);
```

### COM Method Prototype

```
HRESULT Sweep.List.CreateFrequency ([in] BSTR Name,  
                                     [in] SAFEARRAY(DOUBLE) *Frequency);
```

### C Prototype

```
ViStatus IviRFSigGen_CreateFrequencyList (ViSession Vi,  
                                           ViConstString Name,  
                                           ViInt32 Length,  
                                           ViReal64 Frequency[]);
```

### Parameters

Inputs	Description
Vi	Instrument handle
Name	Specifies the name of the list to be created.
Length	Specifies the number of values in the list array
Frequency[]	Specifies the array of frequency values to become elements of the list. The array must have at least as many elements as the value in the Length parameter. The units are Hertz.

### Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

### .NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.



### 20.3.2 Create Power List

#### Description

Creates a named list of power values.

#### .NET Method Prototype

```
void Sweep.List.CreatePower (String name,  
                             Double[] power);
```

#### COM Method Prototype

```
HRESULT Sweep.List.CreatePower ([in] BSTR Name,  
                                [in] SAFEARRAY(DOUBLE) *Power);
```

#### C Prototype

```
ViStatus IviRFSigGen_CreatePowerList (ViSession Vi,  
                                       ViConstString Name,  
                                       ViInt32 Length,  
                                       ViReal64 Power[]);
```

#### Parameters

Inputs	Description
Vi	Instrument handle
Name	Specifies the name of the list to be created.
Length	Specifies the number of values in the list array.
Power[]	Specifies the array of power values to become elements of the list. The array must have at least as many elements as the value in the Length parameter. The units are dBm.

#### Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

#### .NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

### 20.3.3 Create Frequency Power List

#### Description

Creates a named list of both frequency and power values.

#### .NET Method Prototype

```
void Sweep.List.CreateFrequencyPower (String name,
                                     Double[] frequency,
                                     Double[] power);
```

#### COM Method Prototype

```
HRESULT Sweep.List.CreateFrequencyPower ([in] BSTR Name,
                                         [in] SAFEARRAY(DOUBLE) *Frequency,
                                         [in] SAFEARRAY(DOUBLE) *Power);
```

#### C Prototype

```
ViStatus IviRFSigGen_CreateFrequencyPowerList (ViSession Vi,
                                              ViConstString Name,
                                              ViInt32 Length,
                                              ViReal64 Frequency[],
                                              ViReal64 Power[]);
```

#### Parameters

Inputs	Description
Vi	Instrument handle
Name	Specifies the name of the list to be created.
Length	Specifies the number of values in the list array
Frequency[]	Specifies the array of frequency values to become elements of the list. The array must have at least as many elements as the value in the Length parameter. The units are Hertz.
Power[]	Specifies the array of power values to become elements of the list. The array must have at least as many elements as the value in the Length parameter. The units are dBm.

#### Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

#### .NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

## 20.3.4 Select List (IVI-C only)

### Description

Selects one list by name to become active (out of the pool of list).

### .NET Method Prototype

N/A

(Use the List Selected Name property.)

### COM Method Prototype

N/A

(Use the List Selected Name property.)

### C Prototype

```
ViStatus IviRFSigGen_SelectList (ViSession Vi, ViConstString Name);
```

### Parameters

Inputs	Description
Vi	Instrument handle
Name	Specifies the name of the list to make this list active. The driver uses this value to set the List Selected Name attribute. See the attribute description for more details.

### Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

Completion Codes	Description
IVIRFSIGGEN_ERROR_LIST_UNKNOWN	The operation cannot be completed because the list is not defined.

## 20.3.5 Clear All Lists

### Description

Deletes all lists from the pool of defined lists.

### .NET Method Prototype

```
void Sweep.List.ClearAll ();
```

### COM Method Prototype

```
HRESULT Sweep.List.ClearAll ();
```

### C Prototype

```
ViStatus IviRFSigGen_ClearAllLists (ViSession Vi);
```

### Parameters

Inputs	Description
Vi	Instrument handle

### Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

### .NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

## 20.3.6 Configure List Dwell

### Description

Configures the attributes that control list stepping.

### .NET Method Prototype

```
void Sweep.List.ConfigureDwell (Boolean singleStepEnabled,  
                                Ivi.Driver.PrecisionTimeSpan dwell);
```

### COM Method Prototype

```
HRESULT Sweep.List.ConfigureDwell ([in] VARIANT_BOOL SingleStepEnabled,  
                                    [in] DOUBLE Dwell);
```

### C Prototype

```
ViStatus IviRFSigGen_ConfigureListDwell (ViSession Vi,  
                                          ViBoolean SingleStepEnabled,  
                                          ViReal64 Dwell);
```

### Parameters

Inputs	Description
Vi	Instrument handle
SingleStepEnabled	Specifies whether the trigger initiates transitions to the next list step. The driver uses this value to set the List Single Step Enabled attribute. See the attribute description for more details.
Dwell	Specifies the duration of one list step. The driver uses this value to set the List Dwell attribute. See the attribute description for more details.

### Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

### .NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

### 20.3.7 Reset List

**Description**

Resets the current list to the first entry value.

**.NET Method Prototype**

```
void Sweep.List.Reset ();
```

**COM Method Prototype**

```
HRESULT Sweep.List.Reset ();
```

**C Prototype**

```
ViStatus IviRFSigGen_ResetList (ViSession Vi);
```

**Parameters**

Inputs	Description
Vi	Instrument handle

**Return Values (C/COM)**

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

**.NET Exceptions**

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

## **20.4 IviRFSigGenList Behavior Model**

The IviRFSigGenList Extension Group follows the behavior model of the IviRFSigGenSweep capability group.

## **20.5 IviRFSigGenList Compliance Notes**

1. If a specific driver implements the IviRFSigGenList Extension Group, it shall also implement the IviRFSigGenSweep Extension Group and support the defined value List for the Sweep Mode attribute.

## 21. IviRFSigGenALC Extension Group

---

### 21.1 *IviRFSigGenALC Overview*

For generators with configurable Automatic Level Control.

### 21.2 *IviRFSigGenALC Attributes*

The IviRFSigGenALC extension group defines the following attributes:

- ALC Source
- ALC Bandwidth

The ALC Enabled attribute is defined in the IviRFSigGen Base Capabilities Group.

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 30, *IviRFSigGen Attribute ID Definitions*.



## 21.2.1 ALC Source

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32 (C/COM)	R/W	N/A	None	Configure ALC
ViString (.NET)	R/W	N/A	None	Configure ALC

### .NET Property Name

`Alc.Source`

### COM Property Name

`ALC.Source`

### COM Enumeration Name

`IviRFSigGenALCSourceEnum`

### C Constant Name

`IVIRFSIGGEN_ATTR_ALC_SOURCE`

### Description

Specifies the source of the controlling voltage for the Automatic Level Control. The RF level at the sensor point is held constant.

### Defined Values

In IVI.NET the ALC source is a string. If an IVI driver supports an ALC source and the ALC source is listed in IVI-3.3 *Cross Class Capabilities Specification*, Section 3 then the IVI driver shall accept the standard string for that ALC source. This attribute is case insensitive, but case preserving. That is the setting is case insensitive but when reading it back the programmed case is returned. IVI specific drivers may define new ALC source strings for ALC inputs that are not defined by IVI-3.3 *Cross Class Capabilities Specification* if needed.

Name	Description	
	Language	Identifier
Internal	The ALC is controlled by an internal measurement source.	
	C	IVIRFSIGGEN_VAL_ALC_SOURCE_INTERNAL
	COM	IviRFSigGenALCSourceInternal
External	The ALC is controlled by an external voltage.	
	C	IVIRFSIGGEN_VAL_ALC_SOURCE_EXTERNAL
	COM	IviRFSigGenALCSourceExternal

### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## Compliance Notes

1. An instrument driver shall support the defined value Internal.
2. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_ALC_SOURCE_CLASS_EXT_BASE` and less than `IVIRFSIGGEN_VAL_ALC_SOURCE_SPECIFIC_EXT_BASE`.
3. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_ALC_SOURCE_SPECIFIC_EXT_BASE`.
4. When an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to ALC Source Specific Ext Base.

See Section Attribute Value Definitions, for the definitions of ALC Source Specific Ext Base, `IVIRFSIGGEN_VAL_ALC_SOURCE_SPECIFIC_EXT_BASE` and `IVIRFSIGGEN_VAL_ALC_SOURCE_CLASS_EXT_BASE`.

## 21.2.2 ALC Bandwidth

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	Up	Configure ALC

### .NET Property Name

`Alc.Bandwidth`

### COM Property Name

`ALC.Bandwidth`

### C Constant Name

`IVIRFSIGGEN_ATTR_ALC_BANDWIDTH`

### Description

Specifies the bandwidth of the level control. Narrow bandwidth improves noise and allows AM with modulation frequencies beyond bandwidth frequency. The value is coerced to reflect the ability of the current RF signal generator. The units are Hertz.

### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

### **21.3 *IviRFSigGenALC Functions***

The IviRFSigGenALC extension group defines the following functions:

- Configure ALC

This section describes the behavior and requirements of this function.

### 21.3.1 Configure ALC

#### Description

Configures the ALC (Automatic Level Control) of the signal generator’s RF output.

#### .NET Method Prototype

```
void Alc.Configure (String source,  
                   Double bandwidth);
```

#### COM Method Prototype

```
HRESULT ALC.Configure ([in] IviRFSigGenALCSourceEnum Source,  
                       [in] DOUBLE Bandwidth);
```

#### C Prototype

```
ViStatus IviRFSigGen_ConfigureALC (ViSession Vi, ViInt32 Source,  
                                   ViReal64 Bandwidth);
```

#### Parameters

Inputs	Description
Vi	Instrument handle.
Source	The driver uses this value to set the ALC Source attribute. See the attribute description for more details.
Bandwidth	The driver uses this value to set the ALC Bandwidth attribute. See the attribute description for more details.

#### Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

#### .NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

## **21.4 IviRFSigGenALC Behavior Model**

The IviRFSigGenALC Extension Group follows the behavior model of the IviRFSigGenBase capability group. The only modification to the behavior model from the IviRFSigGenBase capability group is the ability to configure IviRFSigGenALC settings.

## 22. IviRFSigGen ReferenceOscillator Extension Group

---

### 22.1 *IviRFSigGenReferenceOscillator Overview*

The IviRFSigGenReferenceOscillator extension group supports signal generators with a configurable frequency reference.

### 22.2 *IviRFSigGenReferenceOscillator Attributes*

The IviRFSigGenReferenceOscillator extension group defines the following attributes:

- Reference Oscillator Source
- Reference Oscillator External Frequency

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 30, *IviRFSigGen Attribute ID Definitions*.

## 22.2.1 Reference Oscillator Source

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32 (C/COM)	R/W	N/A	None	Configure Reference Oscillator
ViString (.NET)	R/W	N/A	None	Configure Reference Oscillator

### .NET Property Name

`ReferenceOscillator.Source`

### COM Property Name

`ReferenceOscillator.Source`

### COM Enumeration Name

`IviRFSigGenReferenceOscillatorSourceEnum`

### C Constant Name

`IVIRFSIGGEN_ATTR_REFERENCE_OSCILLATOR_SOURCE`

### Description

Specifies the reference frequency source used to generate the exact RF output frequency.

### Defined Values

In IVI.NET the reference oscillator source is a string. If an IVI driver supports a reference oscillator source and the reference oscillator source is listed in IVI-3.3 *Cross Class Capabilities Specification*, Section 3 then the IVI driver shall accept the standard string for that reference oscillator source. This attribute is case insensitive, but case preserving. That is the setting is case insensitive but when reading it back the programmed case is returned. IVI specific drivers may define new reference oscillator source strings for reference oscillator sources that are not defined by IVI-3.3 *Cross Class Capabilities Specification* if needed.

Name	Description	
	Language	Identifier
Internal	The internal reference oscillator is used.	
	C	IVIRFSIGGEN_VAL_REFERENCE_OSCILLATOR_SOURCE_INTERNAL
	COM	IviRFSigGenReferenceOscillatorSourceInternal
External	An external reference oscillator is used.	
	C	IVIRFSIGGEN_VAL_REFERENCE_OSCILLATOR_SOURCE_EXTERNAL
	COM	IviRFSigGenReferenceOscillatorSourceExternal

### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.



## Compliance Notes

1. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_REFERENCE_OSCILLATOR_SOURCE_CLASS_EXT_BASE` and less than `IVIRFSIGGEN_VAL_REFERENCE_OSCILLATOR_SOURCE_SPECIFIC_EXT_BASE`.
2. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_REFERENCE_OSCILLATOR_SOURCE_SPECIFIC_EXT_BASE`.
3. When an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to Reference Oscillator Source Specific Ext Base.

See Section Attribute Value Definitions, for the definitions of Reference Oscillator Source Specific Ext Base, `IVIRFSIGGEN_VAL_REFERENCE_OSCILLATOR_SOURCE_SPECIFIC_EXT_BASE` and `IVIRFSIGGEN_VAL_REFERENCE_OSCILLATOR_SOURCE_CLASS_EXT_BASE`.

### 22.2.2 Reference Oscillator External Frequency

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	Configure Reference Oscillator

**.NET Property Name**

`ReferenceOscillator.ExternalFrequency`

**COM Property Name**

`ReferenceOscillator.ExternalFrequency`

**C Constant Name**

`IVIRFSIGGEN_ATTR_REFERENCE_OSCILLATOR_EXTERNAL_FREQUENCY`

**Description**

Specifies the frequency of the external signal, which is used as reference for internal RF frequency generation. This value is used only if Reference Oscillator Source is set to External. The units are Hertz.

**.NET Exceptions**

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

### **22.3 *IviRFSigGenReferenceOscillator Functions***

The IviRFSigGenReferenceOscillator extension group defines the following functions:

- Configure Reference Oscillator

This section describes the behavior and requirements of this function.

### 22.3.1 Configure Reference Oscillator

#### Description

Configures the signal generator's reference oscillator.

#### .NET Method Prototype

```
void ReferenceOscillator.Configure (String source,  
                                   Double frequency);
```

#### COM Method Prototype

```
HRESULT ReferenceOscillator.Configure ([in]  
IviRFSigGenReferenceOscillatorSourceEnum Source,  
                                       [in] DOUBLE Frequency);
```

#### C Prototype

```
ViStatus IviRFSigGen_ConfigureReferenceOscillator (ViSession Vi,  
                                                  ViInt32 Source,  
                                                  ViReal64 Frequency);
```

#### Parameters

Inputs	Description
Vi	Instrument handle.
Source	Specifies source of reference frequency signal. The driver uses this value to set the Reference Oscillator Source attribute. See the attribute description for more details.
Frequency	Specifies the frequency of the external reference oscillator. This parameter is only used if the Source is set to External. The driver uses this value to set the Reference Oscillator Frequency attribute. See the attribute description for more details.

#### Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

#### .NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

## **22.4 *IviRFSigGenReferenceOscillator Behavior Model***

The IviRFSigGenReferenceOscillator Extension Group follows the behavior model of the IviRFSigGenBase capability group. The only modification to the behavior model from the IviRFSigGenBase capability group is the ability to configure IviRFSigGenReferenceOscillator settings.

## 23. IviRFSigGenSoftwareTrigger Extension Group

---

### 23.1 IviRFSigGenSoftwareTrigger Overview

The IviRFSigGenSoftwareTrigger Extension Group supports RF signal generators that can generate output based on a software trigger signal. The user can send a software trigger to cause signal output to occur, to start a sweep (frequency or power), or to initiate the next step (frequency and/or power).

This extension affects instrument behavior when the Trigger Source attribute is set to Software Trigger.

### 23.2 IviRFSigGenSoftwareTrigger Functions

The IviRFSigGenSoftwareTrigger extension defines the following functions:

- Send Software Trigger

This section describes the behavior and requirements of this function.

#### 23.2.1 Send Software Trigger

Refer to *IVI-3.3: Standard Cross Class Capabilities, Section 2 Software Triggering Capability* for the prototype and complete description of this function.

Send Software Trigger is a member of the IviRFSigGen interface.

### 23.3 IviRFSigGenSoftwareTrigger Behavior Model

The IviRFSigGenSoftwareTrigger Extension Group follows the behavior model of the IviRFSigGenSweep capability group. The only modification to the behavior model from the IviRFSigGenSweep capability group is the ability to send software triggers.

### 23.4 IviRFSigGenSoftwareTrigger Compliance Notes

1. If a specific driver implements the IviRFSigGenSoftwareTrigger Extension Group, it shall implement at least one of the following extension groups:
  - IviRFSigGenSweep Extension Group
  - IviRFSigGenArbGenerator Extension Group
  - IviRFSigGenCDMABase Extension Group
  - IviRFSigGenTDMABase Extension Group
2. If a specific driver implements the IviRFSigGenSoftwareTrigger Extension Group together with the IviRFSigGenSweep Extension Group, it shall implement the value Software Trigger for the Sweep Trigger Source attribute.
3. If a specific driver implements the IviRFSigGenSoftwareTrigger Extension Group together with the IviRFSigGenArbGenerator Extension Group, it shall implement the value Software Trigger for the Arb Trigger Source attribute.
4. If a specific driver implements the IviRFSigGenSoftwareTrigger Extension Group together with the IviRFSigGenCDMABase Extension Group, it shall implement the value Software Trigger for the CDMA Trigger Source attribute.
5. If a specific driver implements the IviRFSigGenSoftwareTrigger Extension Group together with the IviRFSigGenTDMABase Extension Group, it shall implement the value Software Trigger for the TDMA Trigger Source attribute.

## 24. IviRFSigGenModulateIQ Extension Group

---

### 24.1 *IviRFSigGenModulateIQ Extension Group Overview*

The IviRFSigGenModulateIQ Extension Group supports signal generators that can apply IQ (vector) modulation to the RF output signal. The user can enable or disable IQ modulation and specify the source of the modulating signal. A calibration is executed with an event function.

### 24.2 *IviRFSigGenModulateIQ Attributes*

The IviRFSigGenModulateIQ extension group defines the following attributes:

- IQ Enabled
- IQ Source
- IQ Nominal Voltage
- IQ Swap Enabled

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 30, *IviRFSigGen Attribute ID Definitions*.

## 24.2.1 IQ Enabled

Data Type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	Configure IQ Enabled

### .NET Property Name

`IQ.Enabled`

### COM Property Name

`IQ.Enabled`

### C Constant Name

`IVIRFSIGGEN_ATTR_IQ_ENABLED`

### Description

If set to True, the RF signal generator applies IQ modulation to the RF output signal. If set to False, the RF signal generator does not apply IQ modulation to the RF output signal.

### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

### Compliance Notes

1. Instrument drivers shall support the values `True` and `False`.



24.2.2 IQ Nominal Voltage

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64	RO	N/A	None	None

**.NET Property Name**

IQ.NominalVoltage

**COM Property Name**

IQ.NominalVoltage

**C Constant Name**

IVIRFSIGGEN\_ATTR\_IQ\_NOMINAL\_VOLTAGE

**Description**

Returns the voltage at which the instrument achieves full modulation. The value is calculated by  $\text{SQRT}(I^2+Q^2)$ . The units are Volts.

**.NET Exceptions**

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

### 24.2.3 IQ Source

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	Configure IQ

#### .NET Property Name

`IQ.Source`

#### .NET Enumeration Name

`IQSource`

#### COM Property Name

`IQ.Source`

#### COM Enumeration Name

`IviRFSigGenIQSourceEnum`

#### C Constant Name

`IVIRFSIGGEN_ATTR_IQ_SOURCE`

#### Description

Specifies the source of the signal that the signal generator uses for IQ modulation.

#### Defined Values

Name	Description		
		Language	Identifier
DigitalModulationBase	The signal generator uses the internally generated digital modulation signal to apply IQ modulation to the output RF signal. – See chapter IviRFSigGenDigitalModulationBase Extension Group for configuration.		
		.NET	<code>IQSource.DigitalModulationBase</code>
		C	<code>IVIRFSIGGEN_VAL_IQ_SOURCE_DIGITAL_MODULATION_BASE</code>
		COM	<code>IviRFSigGenIQSourceDigitalModulationBase</code>
CDMABase	The signal generator uses the internally generated CDMA signal to apply IQ modulation to the output RF signal. – See chapter IviRFSigGenCDMABase Extension Group for configuration.		
		.NET	<code>IQSource.CdmaBase</code>
		C	<code>IVIRFSIGGEN_VAL_IQ_SOURCE_CDMA_BASE</code>
		COM	<code>IviRFSigGenIQSourceCDMABase</code>
TDMABase	The signal generator uses the internally generated TDMA signal to apply IQ modulation to the output RF signal. – See chapter IviRFSigGenTDMABase Extension Group for configuration.		
		.NET	<code>IQSource.TdmaBase</code>

		C	IVIRFSIGGEN_VAL_IQ_SOURCE_TDMA_BASE
		COM	IviRFSigGenIQSourceTDMABase
ArbGenerator	The signal generator uses the internally generated Arb signal to apply IQ modulation to the output RF signal. – See chapter IviRFSigGenArbGenerator Extension Group for configuration.		
		.NET	IQSource.ArbGenerator
		C	IVIRFSIGGEN_VAL_IQ_SOURCE_ARB_GENERATOR
		COM	IviRFSigGenIQSourceArbGenerator
External	The signal generator uses data from an external source for IQ modulation.		
		.NET	IQSource.External
		C	IVIRFSIGGEN_VAL_IQ_SOURCE_EXTERNAL
		COM	IviRFSigGenIQSourceExternal

## .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## Compliance Notes

1. If a specific driver implements the defined value DigitalModulationBase it shall implement the IviRFSigGenDigitalModulationBase Extension group.
2. If a specific driver implements the defined value CDMABase it shall implement the IviRFSigGenCDMABase Extension group.
3. If a specific driver implements the defined value TDMABase it shall implement the IviRFSigGenTDMABase Extension group.
4. If a specific driver implements the defined value ArbGenerator it shall implement the IviRFSigGenArbGenerator extension group.
5. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to IVIRFSIGGEN\_VAL\_IQ\_SOURCE\_CLASS\_EXT\_BASE and less than IVIRFSIGGEN\_VAL\_IQ\_SOURCE\_SPECIFIC\_EXT\_BASE.
6. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to IVIRFSIGGEN\_VAL\_IQ\_SOURCE\_SPECIFIC\_EXT\_BASE.
7. When an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to IQ Source Specific Ext Base.

See Section Attribute Value Definitions, for the definitions of IQ Source Specific Ext Base, IVIRFSIGGEN\_VAL\_IQ\_SOURCE\_SPECIFIC\_EXT\_BASE and IVIRFSIGGEN\_VAL\_IQ\_SOURCE\_CLASS\_EXT\_BASE.

## 24.2.4 IQ Swap Enabled

Data Type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	Configure IQ

### .NET Property Name

`IQ.SwapEnabled`

### COM Property Name

`IQ.SwapEnabled`

### C Constant Name

`IVIRFSIGGEN_ATTR_IQ_SWAP_ENABLED`

### Description

If set to `True`, the RF signal generator applies inverse phase rotation of the IQ signal. If set to `False`, the RF signal generator applies non-inverse phase rotation of the IQ signal.

### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

### Compliance Notes

1. Instrument drivers shall support the value `False`.

### **24.3 *IviRFSigGenModulateIQ Functions***

The IviRFSigGenModulateIQ extension group defines the following functions:

- Configure IQ Enabled (IVI-C only)
- Configure IQ
- Calibrate IQ

This section describes the behavior and requirements of each function.

### 24.3.1 Configure IQ Enabled (IVI-C only)

**Description**

Configures the signal generator to apply IQ (vector) modulation to the RF output signal.

**.NET Method Prototype**

N/A  
(use the `IQ.Enabled` property)

**COM Method Prototype**

N/A  
(use the `IQ.Enabled` property)

**C Prototype**

```
ViStatus IviRFSigGen_ConfigureIQEnabled (ViSession Vi, ViBoolean Enabled);
```

**Parameters**

Inputs	Description
Vi	Instrument handle
Enabled	True enables IQ (vector) modulation. The driver uses this value to set the Modulate IQ Enabled attribute. See the attribute description for more details.

**Return Values (C)**

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

## 24.3.2 Configure IQ

### Description

Configures the attributes that control the signal generator's IQ modulation. These attributes are the modulation source, impairment state and swap state.

### .NET Method Prototype

```
void IQ.Configure (IQSource source,  
                  Boolean swapEnabled);
```

### COM Method Prototype

```
HRESULT IQ.Configure ([in] IviRFSigGenIQSourceEnum Source,  
                     [in] VARIANT_BOOL SwapEnabled);
```

### C Prototype

```
ViStatus IviRFSigGen_ConfigureIQ (ViSession Vi, ViInt32 Source,  
                                  ViBoolean SwapEnabled);
```

### Parameters

Inputs	Description
Vi	Instrument handle
Source	Specifies the modulation source. The driver uses this value to set the Modulate IQ Source attribute. See the attribute description for more details.
SwapEnabled	Specifies the swap state. True swaps I and Q inputs, the result is an invers phase rotation. The driver uses this value to set the Modulate IQ Swap Enabled attribute. See the attribute description for more details.

### Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

### .NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

### 24.3.3 Calibrate IQ

**Description**

This function calibrates the IQ modulator

**.NET Method Prototype**

```
void IQ.Calibrate();
```

**COM Method Prototype**

```
HRESULT IQ.Calibrate();
```

**C Prototype**

```
ViStatus IviRFSigGen_CalibrateIQ (ViSession Vi) ;
```

**Parameters**

Inputs	Description
Vi	Instrument handle.

**Return Values (C/COM)**

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

**.NET Exceptions**

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.



## **24.4 *IviRFSigGenModulateIQ Behavior Model***

The IviRFSigGenModulateIQ Extension Group follows the behavior model of the IviRFSigGenBase capability group. The only modification to the behavior model from the IviRFSigGenBase capability group is the ability to configure IviRFSigGenModulateIQ settings.

## 25. IviRFSigGenIQImpairment Extension Group

---

### 25.1 *IviRFSigGenIQImpairment Extension Group Overview*

With IviRFSigGenIQImpairment Extension Group the user can simulate or correct impairment on IQ (vector) modulation. This group is an extension of the IviRFSigGenModulateIQ Extension Group.

### 25.2 *IviRFSigGenIQImpairment Attributes*

The IviRFSigGenIQImpairment extension group defines the following attributes:

- IQ Impairment Enabled
- IQ I-Offset
- IQ Q-Offset
- IQ Ratio
- IQ Skew

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 30, *IviRFSigGen Attribute ID Definitions*.

## 25.2.1 IQ Impairment Enabled

Data Type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	N/A	None	Configure IQ Impairment Enabled

### .NET Property Name

`IQ.Impairment.Enabled`

### COM Property Name

`IQ.Impairment.Enabled`

### C Constant Name

`IVIRFSIGGEN_ATTR_IQ_IMPAIRMENT_ENABLED`

### Description

If set to True, the RF signal generator applies all IQ impairment attributes to the IQ modulation. If set to False, the RF signal generator does not apply any IQ impairment attributes to the IQ modulation.

### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

### Compliance Notes

1. Instrument drivers shall support the values `True` and `False`.

25.2.2 IQ I-Offset

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	Configure IQ Impairment

.NET Property Name

```
IQ.Impairment.IOffset
```

COM Property Name

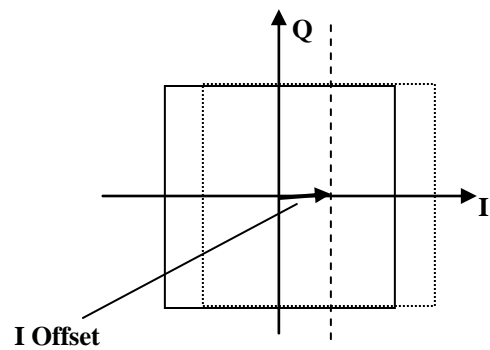
```
IQ.Impairment.IOffset
```

C Constant Name

```
IVIRFSIGGEN_ATTR_IQ_I_OFFSET
```

Description

Specifies an origin offset voltage to the I signal. The range of values allowed is -100% to +100%. The value is expressed as percentage (%).



.NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

25.2.3 IQ Q-Offset

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	Configure IQ Impairment

.NET Property Name

`IQ.Impairment.QOffset`

COM Property Name

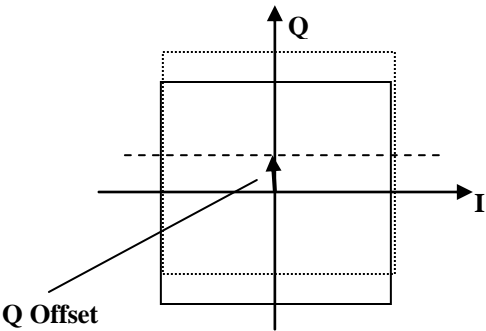
`IQ.Impairment.QOffset`

C Constant Name

`IVIRFSIGGEN_ATTR_IQ_Q_OFFSET`

Description

Specifies an origin offset voltage to the Q signal. The range of values allowed is –100% to +100%. The value is expressed as percentage (%).



.NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

25.2.4 IQ Ratio

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	Configure IQ Impairment

.NET Property Name

```
IQ.Impairment.Ratio
```

COM Property Name

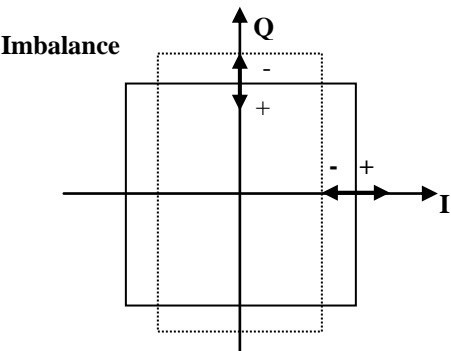
```
IQ.Impairment.Ratio
```

C Constant Name

```
IVIRFSIGGEN_ATTR_IQ_RATIO
```

Description

Specifies the gain imbalance between the I and Q channels. For no imbalance this value is set to 0 %. The value is expressed as percentage (%).



.NET Exceptions

The Ivi-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

25.2.5 IQ Skew

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	Configure IQ Impairment

.NET Property Name

`IQ.Impairment.Skew`

COM Property Name

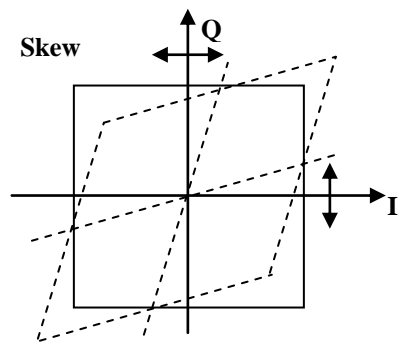
`IQ.Impairment.Skew`

C Constant Name

`IVIRFSIGGEN_ATTR_IQ_SKEW`

Description

Specifies the adjustment of the phase angle between the I and Q vectors. If this skew is zero, the phase angle is 90 degrees. The units are degrees.



.NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

### **25.3 *IviRFSigGenIQImpairment Functions***

The IviRFSigGenIQImpairment extension group defines the following functions:

- Configure IQ Impairment Enabled (IVI-C only)
- Configure IQ Impairment

This section describes the behavior and requirements of each function.



### 25.3.1 Configure IQ Impairment Enabled (IVI-C only)

**Description**

Configures the IQ modulation to allow controlled impairment for test or external corrections.

**.NET Method Prototype**

N/A  
(use the `IQ.Impairment.Enabled` property)

**COM Method Prototype**

N/A  
(use the `IQ.Impairment.Enabled` property)

**C Prototype**

```
ViStatus IviRFSigGen_ConfigureIQImpairmentEnabled (ViSession Vi,  
                                                    ViBoolean Enabled);
```

**Parameters**

Inputs	Description
Vi	Instrument handle
Enabled	True enables IQ (vector) modulation impairment. The driver uses this value to set the IQ Impairment Enabled attribute. See the attribute description for more details.

**Return Values (C)**

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

### 25.3.2 Configure IQ Impairment

#### Description

Configures the attributes that simulate or correct impairment for the signal generator's IQ modulation. These attributes are only used, if IQ Impairment Enabled attribute is set to True (see Configure IQ Impairment Enabled function).

#### .NET Method Prototype

```
void IQ.Impairment.Configure (Double iOffset,  
                             Double qOffset,  
                             Double ratio,  
                             Double skew);
```

#### COM Method Prototype

```
HRESULT IQ.Impairment.Configure ([in] DOUBLE IOffset,  
                                 [in] DOUBLE QOffset,  
                                 [in] DOUBLE Ratio,  
                                 [in] DOUBLE Skew);
```

#### C Prototype

```
ViStatus IviRFSigGen_ConfigureIQImpairment (ViSession Vi,  
                                             ViReal64 IOffset,  
                                             ViReal64 QOffset,  
                                             ViReal64 Ratio,  
                                             ViReal64 Skew);
```

#### Parameters

Inputs	Description
Vi	Instrument handle
IOffset	Specifies an offset to the I-path of IQ signals. The driver uses this value to set the IQ I Offset attribute. See the attribute description for more details.
QOffset	Specifies an offset to the Q-path of IQ signals. The driver uses this value to set the IQ Q Offset attribute. See the attribute description for more details.
Ratio	Specifies a gain imbalance to the IQ signals. The driver uses this value to set the IQ Ratio attribute. See the attribute description for more details.
Skew	Specifies an angle offset to the IQ signals. The driver uses this value to set the IQ Skew attribute. See the attribute description for more details.

#### Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

#### .NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

## **25.4 *IviRFSigGenIQImpairment Behavior Model***

The IviRFSigGenIQImpairment Extension Group follows the behavior model of the IviRFSigGenBase capability group. The only modification to the behavior model from the IviRFSigGenBase capability group is the ability to configure IviRFSigGenIQImpairment settings.

## **25.5 *IviRFSigGenIQImpairment Compliance Notes***

1. If a specific driver implements the IviRFSigGenIQImpairment Extension Group, it shall also implement the IviRFSigGenModulateIQ Extension Group.

## 26. IviRFSigGenArbGenerator Extension Group

---

### 26.1 *IviRFSigGenArbGenerator Extension Group Overview*

The IviRFSigGenArbGenerator Extension Group controls the internal arbitrary waveform generator. The two outputs of the Arb generator are used with the IQ (vector) modulation to simulate digital modulation. So this group is an extension of the IviRFSigGenModulateIQ Extension Group.

### 26.2 *IviRFSigGenArbGenerator Attributes*

The IviRFSigGenArbGenerator extension group defines the following attributes:

- Arb Selected Waveform
- Arb Clock Frequency
- Arb Filter Frequency
- Arb Max Number Waveforms
- Arb Waveform Quantum
- Arb Waveform Size Min
- Arb Waveform Size Max
- Arb Trigger Source
- Arb External Trigger Slope

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 30, *IviRFSigGen Attribute ID Definitions*.

### 26.2.1 Arb Selected Waveform

Data Type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	Select Arb Waveform

**.NET Property Name**

`DigitalModulation.Arb.SelectedWaveform`

**COM Property Name**

`DigitalModulation.Arb.SelectedWaveform`

**C Constant Name**

`IVIRFSIGGEN_ATTR_ARB_SELECTED_WAVEFORM`

**Description**

Specifies the selected waveform from the pool of available waveforms.

**.NET Exceptions**

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## 26.2.2 Arb Clock Frequency

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	None	Configure Arb

### .NET Property Name

`DigitalModulation.Arb.ClockFrequency`

### COM Property Name

`DigitalModulation.Arb.ClockFrequency`

### C Constant Name

`IVIRFSIGGEN_ATTR_ARB_CLOCK_FREQUENCY`

### Description

Specifies the sample frequency. The waveform is generated with this clock frequency. The units are Hertz.

### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

### 26.2.3 Arb Filter Frequency

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	N/A	UP	Configure Arb

#### .NET Property Name

`DigitalModulation.Arb.FilterFrequency`

#### COM Property Name

`DigitalModulation.Arb.FilterFrequency`

#### C Constant Name

`IVIRFSIGGEN_ATTR_ARB_FILTER_FREQUENCY`

#### Description

Specifies the cut-off frequency of the low pass filter. The waveform is filtered before output with this filter for antialiasing. The filter frequency normally is lower than the clock frequency. The units are Hertz.

#### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## 26.2.4 Arb Max Number Waveforms

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32	RO	N/A	None	Query Arb Waveform Capabilities

### .NET Property Name

`DigitalModulation.Arb.MaxNumberWaveforms`

### COM Property Name

`DigitalModulation.Arb.MaxNumberWaveforms`

### C Constant Name

`IVIRFSIGGEN_ATTR_ARB_MAX_NUMBER_WAVEFORMS`

### Description

Specifies the max number of waveforms the instrument can hold in the memory. The number may depend on the length of the waveform already in the pool of waveforms stored in the instrument.

### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.



## 26.2.5 Arb Waveform Quantum

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32	RO	N/A	None	Query Arb Waveform Capabilities

### .NET Property Name

`DigitalModulation.Arb.WaveformQuantum`

### COM Property Name

`DigitalModulation.Arb.WaveformQuantum`

### C Constant Name

`IVIRFSIGGEN_ATTR_ARB_WAVEFORM_QUANTUM`

### Description

The waveform length (the number of samples) shall be a multiple of this quantum. If it is 1, there is no restriction on the length - other than min and max size - for the waveform.

### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## 26.2.6 Arb Waveform Size Min

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32	RO	N/A	None	Query Arb Waveform Capabilities

### .NET Property Name

`DigitalModulation.Arb.WaveformSizeMin`

### COM Property Name

`DigitalModulation.Arb.WaveformSizeMin`

### C Constant Name

`IVIRFSIGGEN_ATTR_ARB_WAVEFORM_SIZE_MIN`

### Description

The waveform length (the number of samples) shall be equal or greater than min size. If it is 1, there is no restriction on the length - other than max size and quantum - for the waveform.

### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## 26.2.7 Arb Waveform Size Max

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32	RO	N/A	None	Query Arb Waveform Capabilities

### .NET Property Name

`DigitalModulation.Arb.WaveformSizeMax`

### COM Property Name

`DigitalModulation.Arb.WaveformSizeMax`

### C Constant Name

`IVIRFSIGGEN_ATTR_ARB_WAVEFORM_SIZE_MAX`

### Description

The waveform length (the number of samples) shall be equal or less than max size.

### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## 26.2.8 Arb Trigger Source

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32 (C/COM)	R/W	N/A	None	Configure Arb Trigger Source
ViString (.NET)				

### COM Property Name

`DigitalModulation.Arb.TriggerSource`

### COM Enumeration Name

`IviRFSigGenArbTriggerSourceEnum`

### C Constant Name

`IVIRFSIGGEN_ATTR_ARB_TRIGGER_SOURCE`

### Description

Specifies the way to start the Arb waveform or running it continuously.

### Defined Values

In IVI.NET the trigger source is a string. If an IVI driver supports a trigger source and the trigger source is listed in IVI-3.3 *Cross Class Capabilities Specification*, Section 3 then the IVI driver shall accept the standard string for that trigger source. This attribute is case insensitive, but case preserving. That is the setting is case insensitive but when reading it back the programmed case is returned. IVI specific drivers may define new trigger source strings for trigger sources that are not defined by IVI-3.3 *Cross Class Capabilities Specification* if needed.

Name	Description		
		Language	Identifier
Immediate	The Arb generator system does not wait for a trigger of any kind, so it is running continuously.		
		C	IVIRFSIGGEN_VAL_ARB_TRIGGER_SOURCE_IMMEDIATE
		COM	IviRFSigGenArbTriggerSourceImmediate
External	The sweep is started with an external signal.		
		C	IVIRFSIGGEN_VAL_ARB_TRIGGER_SOURCE_EXTERNAL
		COM	IviRFSigGenArbTriggerSourceExternal
Software Trigger	The sweep is started with a software programmable trigger.		
		C	IVIRFSIGGEN_VAL_ARB_TRIGGER_SOURCE_SOFTWARE
		COM	IviRFSigGenArBbtTriggerSourceSoftware

### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## Compliance Notes

1. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_ARB_TRIGGER_SOURCE_CLASS_EXT_BASE` and less than `IVIRFSIGGEN_VAL_ARB_TRIGGER_SOURCE_SPECIFIC_EXT_BASE`.
2. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_ARB_TRIGGER_SOURCE_SPECIFIC_EXT_BASE`.
3. When an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to Arb Trigger Source Specific Ext Base.

See Section Attribute Value Definitions, for the definitions of Arb Trigger Source Specific Ext Base, `IVIRFSIGGEN_VAL_ARB_TRIGGER_SOURCE_SPECIFIC_EXT_BASE` and `IVIRFSIGGEN_VAL_ARB_TRIGGER_SOURCE_CLASS_EXT_BASE`.

## 26.2.9 Arb External Trigger Slope

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	Configure Arb External Trigger Slope

### .NET Property Name

`DigitalModulation.Arb.ExternalTriggerSlope`

### .NET Enumeration Name

`Slope`

### COM Property Name

`DigitalModulation.Arb.ExternalTriggerSlope`

### COM Enumeration Name

`IviRFSigGenArbExternalTriggerSlopeEnum`

### C Constant Name

`IVIRFSIGGEN_ATTR_ARB_EXTERNAL_TRIGGER_SLOPE`

### Description

Specifies whether the trigger event occurs on the rising or falling edge of the input signal.

### Defined Values

Name	Description		
		Language	Identifier
Positive	Enables rising edge triggering.		
		.NET	<code>Slope.Positive</code>
		C	<code>IVIRFSIGGEN_VAL_ARB_EXTERNAL_TRIGGER_SLOPE_POSITIVE</code>
		COM	<code>IviRFSigGenArbExternalTriggerSlopePositive</code>
Negative	Enables falling edge triggering.		
		.NET	<code>Slope.Negative</code>
		C	<code>IVIRFSIGGEN_VAL_ARB_EXTERNAL_TRIGGER_SLOPE_NEGATIVE</code>
		COM	<code>IviRFSigGenArbExternalTriggerSlopeNegative</code>

### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## Compliance Notes

1. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_ARB_EXTERNAL_TRIGGER_SLOPE_CLASS_EXT_BASE` and less than `IVIRFSIGGEN_VAL_ARB_EXTERNAL_TRIGGER_SLOPE_SPECIFIC_EXT_BASE`.
2. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_ARB_EXTERNAL_TRIGGER_SLOPE_SPECIFIC_EXT_BASE`.
3. When an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to Arb External Trigger Slope Specific Ext Base.

See Section Attribute Value Definitions, for the definitions of Arb External Trigger Slope Specific Ext Base, `IVIRFSIGGEN_VAL_ARB_EXTERNAL_TRIGGER_SLOPE_SPECIFIC_EXT_BASE` and `IVIRFSIGGEN_VAL_ARB_EXTERNAL_TRIGGER_SLOPE_CLASS_EXT_BASE`.

### **26.3 *IviRFSigGenArbGenerator Functions***

The IviRFSigGenArbGenerator extension group defines the following functions:

- Configure Arb
- Write Arb Waveform
- Select Arb Waveform (IVI-C only)
- Clear All Arb Waveforms
- Query Arb Waveform Capabilities
- Configure Arb Trigger Source (IVI-C only)
- Configure Arb External Trigger Slope (IVI-C only)

This section describes the behavior and requirements of each function.



### 26.3.1 Configure Arb

#### Description

Configures the Arb generator. Specified is the sample frequency the waveform is generated with together with the cut-off frequency of the low pass filter which is used for antialiasing the output waveform. The filter frequency normally is lower than the clock frequency.

#### .NET Method Prototype

```
void DigitalModulation.Arb.Configure (Double clockFrequency,  
                                     Double filterFrequency);
```

#### COM Method Prototype

```
HRESULT DigitalModulation.Arb.Configure ([in] DOUBLE ClockFrequency,  
                                         [in] DOUBLE FilterFrequency);
```

#### C Prototype

```
ViStatus IviRFSigGen_ConfigureArb (ViSession Vi, ViReal64 ClockFrequency,  
                                   ViReal64 FilterFrequency);
```

#### Parameters

Inputs	Description
Vi	Instrument handle
ClockFrequency	The driver uses this value to set the Arb Clock Frequency attribute. See the attribute description for more details.
FilterFrequency	The driver uses this value to set the Arb Filter Frequency attribute. See the attribute description for more details.

#### Return Values (C/COM)

The **IVI-3.2: Inherent Capabilities Specification** defines general status codes that this function can return..NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

## 26.3.2 Write Arb Waveform

### Description

Stores the transmitted waveform in the drivers's or instrument's memory.

### .NET Method Prototype

```
void DigitalModulation.Arb.WriteWaveform (String name,  
                                           Double[] iData,  
                                           Double[] qData,  
                                           Boolean moreDataPending);
```

### COM Method Prototype

```
HRESULT DigitalModulation.Arb.WriteWaveform ([in] BSTR Name,  
                                              [in] SAFEARRAY(DOUBLE) *IData,  
                                              [in] SAFEARRAY(DOUBLE) *QData,  
                                              [in] VARIANT_BOOL MoreDataPending);
```

### C Prototype

```
ViStatus IviRFSigGen_WriteArbWaveform (ViSession Vi, ViConstString Name,  
                                       ViInt32 NumberOfSamples, ViReal64 IData[],  
                                       ViReal64 QData[], ViBoolean MoreDataPending);
```

### Parameters

Inputs	Description
Vi	Instrument handle
NumberOfSamples	The number of samples in both I and Q array. Not used in COM environment.
Name	The driver uses this value to name the stored waveform.
IData	The driver uses the values of this array as I part of the waveform . The voltages are normalized to {-1, +1} to the full I range.
QData	The driver uses the values of this array as Q part of the waveform . The voltages are normalized to {-1, +1} to the full Q range.
MoreDataPending	The value is set to True if more data is coming. To complete the waveform, the next data shall have the same name. False means the waveform has no more data to come.

### Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

### .NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

### 26.3.3 Select Arb Waveform (IVI-C only)

**Description**

Selects a waveform from the pool of waveforms to become active.

**.NET Method Prototype**

N/A  
(use the `DigitalModulation.Arb.SelectedWaveform` property)

**COM Method Prototype**

N/A  
(use the `DigitalModulation.Arb.SelectedWaveform` property)

**C Prototype**

```
ViStatus IviRFSigGen_SelectArbWaveform (ViSession Vi, ViConstString Name);
```

**Parameters**

Inputs	Description
Vi	Instrument handle
Name	The driver uses this value to set the Arb Selected Waveform attribute. See the attribute description for more details.

**Return Values (C)**

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

### 26.3.4 Clear All Arb Waveforms

**Description**

Deletes all waveforms from the pool of waveforms currently defined.

**.NET Method Prototype**

```
void DigitalModulation.Arb.ClearAllWaveforms ();
```

**COM Method Prototype**

```
HRESULT DigitalModulation.Arb.ClearAllWaveforms ();
```

**C Prototype**

```
ViStatus IviRFSigGen_ClearAllArbWaveforms (ViSession Vi);
```

**Parameters**

Inputs	Description
Vi	Instrument handle

**Return Values (C/COM)**

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

**.NET Exceptions**

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

### 26.3.5 Query Arb Waveform Capabilities (IVI-C and IVI-COM Only)

#### Description

Returns the capabilities of the Arb generator related to the waveform.

#### .NET Method Prototype

N/A  
(use the `DigitalModulation.Arb.MaxNumberWaveforms`,  
`DigitalModulation.Arb.WaveformQuantum`, `DigitalModulation.Arb.MinWaveformSize`, and  
`DigitalModulation.Arb.MaxWaveformSize` properties.)

#### COM Method Prototype

```
HRESULT DigitalModulation.Arb.QueryWaveformCapabilities (  
    [in, out] LONG *MaxNumberWaveforms,  
    [in, out] LONG *WaveformQuantum,  
    [in, out] LONG *MinWaveformSize,  
    [in, out] LONG *MaxWaveformSize);
```

#### C Prototype

```
ViStatus IviRFSigGen_QueryArbWaveformCapabilities (ViSession Vi,  
    ViInt32 *MaxNumberWaveforms,  
    ViInt32 *WaveformQuantum,  
    ViInt32 *MinWaveformSize,  
    ViInt32 *MaxWaveformSize);
```

#### Parameters

Inputs	Description
Vi	Instrument handle

Outputs	Description
MaxNumberWaveforms	Returns the Arb Max Number Waveforms attribute. See the attribute description for more details.
WaveformQuantum	Returns the Arb Waveform Quantum attribute. See the attribute description for more details.
MinWaveformSize	Returns the Arb Min Waveform Size attribute. See the attribute description for more details.
MaxWaveformSize	Returns the Arb Max Waveform Size attribute. See the attribute description for more details.

#### Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

### 26.3.6 Configure Arb Trigger Source (IVI-C only)

**Description**

Configures the trigger source for waveform generation. The output waveform may be generated continuously or once based on the value of the Arb Trigger Source attribute.

**.NET Method Prototype**

N/A  
(Use the Arb Trigger Source property.)

**COM Method Prototype**

N/A  
(Use the Arb Trigger Source property.)

**C Prototype**

```
ViStatus IviRFSigGen_ConfigureArbTriggerSource (ViSession Vi, ViInt32 Source);
```

**Parameters**

Inputs	Description
Vi	Instrument handle
Source	The driver uses this value to set the Arb Trigger Source attribute. See the attribute description for more details.

**Return Values (C)**

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

### 26.3.7 Configure Arb External Trigger Slope (IVI-C only)

**Description**

Configures the external trigger slope.

**.NET Method Prototype**

N/A  
(Use the Arb External Trigger Slope property.)

**COM Method Prototype**

N/A  
(Use the Arb External Trigger Slope property.)

**C Prototype**

```
ViStatus IviRFSigGen_ConfigureArbExternalTriggerSlope (ViSession Vi,  
ViInt32 Slope);
```

**Parameters**

Inputs	Description
Vi	Instrument handle
Slope	The driver uses this value to set the Arb External Trigger Slope attribute. See the attribute description for more details.

**Return Values (C)**

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

## **26.4 IviRFSigGenArbGenerator Behavior Model**

The IviRFSigGenArbGenerator Extension Group follows the behavior model of the IviRFSigGenBase capability group. The only modification to the behavior model from the IviRFSigGenBase capability group is the ability to configure IviRFSigGenArbGenerator settings.

## **26.5 IviRFSigGenArbGenerator Compliance Notes**

If a specific driver implements the IviRFSigGenArbGenerator Extension Group, it shall also implement the IviRFSigGenModulateIQ Extension Group and support the defined value ArbGenerator for the IQ Source attribute.

1. If a specific driver does not support the value External for the Arb Trigger Source attribute, it need not support the Arb External Trigger Slope attribute or the Configure Arb External Trigger Slope function.



## 27. IviRFSigGenDigitalModulationBase Extension Group

---

### 27.1 *IviRFSigGenDigitalModulationBase Extension Group Overview*

With IviRFSigGenDigitalModulationBase Extension Group the user can generate signals conforming to wireless communication standards (e.g. mobile cellular standards). The generated signals do not have TDMA framing nor CDMA channel coding.

The functionality covers basic modulation properties such as IQ constellation, symbol mapping, etc. within a specified communication standard.

### 27.2 *IviRFSigGenDigitalModulationBase Attributes*

The IviRFSigGenDigitalModulationBase Extension Group defines the following attributes:

- DigitalModulationBase Standard Names (IVI.NET only)
- DigitalModulationBase Standard Name (IVI-COM only)
- DigitalModulationBase Standard Count (IVI-C and IVI-COM only)
- DigitalModulationBase Selected Standard
- DigitalModulationBase Data Source
- DigitalModulationBase PRBS Type
- DigitalModulationBase Selected Bit Sequence
- DigitalModulationBase Clock Source
- DigitalModulationBase External Clock Type

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 30, *IviRFSigGen Attribute ID Definitions*.

27.2.1 DigitalModulationBase Standard Names (.NET only)

Data Type	Access	Applies to	Coercion	High Level Functions
ReadOnlyCollection<String>	RO	N/A	None	None

**.NET Property Name**

`ReadOnlyCollection<String> DigitalModulation.Base.StandardNames`

**COM Property Name**

N/A  
(Use the DigitalModulationBase Standard Name property.)

**C Constant Name**

N/A  
(Use the Get DigitalModulationBase Standard Name function.)

**Description**

This property returns a collection of the currently defined Digital Modulation Base Standard identifiers.

**.NET Exceptions**

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

### 27.2.2 DigitalModulationBase Standard Name (IVI-COM only)

Data Type	Access	Applies to	Coercion	High Level Functions
ViString	RO	N/A	None	None

#### .NET Property Name

N/A

(Use the DigitalModulationBase Standard Names property.)

#### COM Property Name

```
HRESULT DigitalModulation.Base.StandardName ([in] long Index,  
                                              [out, retval] BSTR* Name);
```

#### C Constant Name

N/A

(Use the Get DigitalModulationBase Standard Name function.)

#### Description

This property returns the DigitalModulationBase Standard identifier that corresponds to the one-based index that the user specifies. If the value that the user pass for the Index parameter is less than one or greater than the value of the DigitalModulationBase Standard Count attribute, the property returns an empty string in the Name parameter and returns the Invalid Value error.

#### Parameters

Inputs	Description	Base Type
Index	A one-based index that defines which name to return.	ViInt32

Output/Return Value	Description	Base Type
Name	A driver-allocated buffer into which the driver stores the DigitalModulationBase standard name.	ViChar[]

### 27.2.3 DigitalModulationBase Standard Count (IVI-C and IVI-COM only)

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32	RO	N/A	None	N/A

#### .NET Property Name

N/A  
(Use DigitalModulationBase Standard Names Count property.)

#### COM Property Name

DigitalModulation.Base.StandardCount

#### C Constant Name

IVIRFSIGGEN\_ATTR\_DIGITAL\_MODULATION\_BASE\_STANDARD\_COUNT

#### Description

Specifies the number of DigitalModulationBase standards available for a particular instrument.

27.2.4 DigitalModulationBase Selected Standard

Data Type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	Select DigitalModulationBase Standard

**.NET Property Name**

DigitalModulation.Base.SelectedStandard

**COM Property Name**

DigitalModulation.Base.SelectedStandard

**C Constant Name**

IVIRFSIGGEN\_ATTR\_DIGITAL\_MODULATION\_BASE\_SELECTED\_STANDARD

**Description**

Specifies the actual standard used by the instrument. The coding, mapping, symbol rate or bit clock frequency, filter together with the according filter, parameters, FSK deviation or ASK depth (in case of FSK or ASK modulation) are set as defined in the selected standard.

**.NET Exceptions**

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## 27.2.5 DigitalModulationBase Data Source

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	Configure DigitalModulationBase Data Source

### .NET Property Name

DigitalModulation.Base.DataSource

### .NET Enumeration Name

DigitalModulationBaseDataSource

### COM Property Name

DigitalModulation.Base.DataSource

### COM Enumeration Name

IviRFSigGenDigitalModulationBaseDataSourceEnum

### C Constant Name

IVIRFSIGGEN\_ATTR\_DIGITAL\_MODULATION\_BASE\_DATA\_SOURCE

### Description

Specifies the source of data. The data is used to modulate the RF signal according to the standard selected with the DigitalModulationBase Selected Standard attribute.

### Defined Values

Name	Description	
	Language	Identifier
External	The data from an external device connected to the instrument is used.	
	.NET	DigitalModulationBaseDataSource.External
	C	IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_DATA_SOURCE_EXTERNAL
	COM	IviRFSigGenDigitalModulationBaseDataSourceExternal
PRBS	The internal PRBS (Pseudo Random Binary Sequence) generator is used as data source.	
	.NET	DigitalModulationBaseDataSource.Prbs
	C	IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_DATA_SOURCE_PRBS
	COM	IviRFSigGenDigitalModulationBaseDataSourcePRBS
Bit Sequence	A constant bit sequence is used as data source and repeated continuously.	
	.NET	DigitalModulationBaseDataSource.BitSequence

	C	IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_DATA_SOURCE_BIT_SEQUENCE
	COM	IviRFSigGenDigitalModulationBaseDataSourceBitSequence

## .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## Compliance Notes

1. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to IVIRFSIGGEN\_VAL\_DIGITAL\_MODULATION\_BASE\_DATA\_SOURCE\_CLASS\_EXT\_BASE and less than IVIRFSIGGEN\_VAL\_DIGITAL\_MODULATION\_BASE\_DATA\_SOURCE\_SPECIFIC\_EXT\_BASE.
2. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to IVIRFSIGGEN\_VAL\_DIGITAL\_MODULATION\_BASE\_DATA\_SOURCE\_SPECIFIC\_EXT\_BASE.
3. When an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to Digital Modulation Base Data Source Specific Ext Base.

See Section Attribute Value Definitions, for the definitions of Digital Modulation Base Data Source Specific Ext Base, IVIRFSIGGEN\_VAL\_DIGITAL\_MODULATION\_BASE\_DATA\_SOURCE\_SPECIFIC\_EXT\_BASE and IVIRFSIGGEN\_VAL\_DIGITAL\_MODULATION\_BASE\_DATA\_SOURCE\_CLASS\_EXT\_BASE.

## 27.2.6 DigitalModulationBase PRBS Type

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	Configure DigitalModulationBase PRBS Type

### .NET Property Name

`DigitalModulation.Base.PrbsType`

### .NET Enumeration Name

`DigitalModulationBasePrbsType`

### COM Property Name

`DigitalModulation.Base.PRBSType`

### COM Enumeration Name

`IviRFSigGenDigitalModulationBasePRBSTypeEnum`

### C Constant Name

`IVIRFSIGGEN_ATTR_DIGITAL_MODULATION_BASE_PRBS_TYPE`

### Description

Specifies the type of the PRBS as defined in the CCITT-V.52 standard. The PRBS (Pseudo Random Binary Sequence) is used only if DigitalModulationBase Data Source is set to PRBS.

### Defined Values

Name	Description	
	Language	Identifier
PRBS9	Length of PRBS sequence is $2^9-1$ .	
	.NET	<code>DigitalModulationBasePrbsType.Prbs9</code>
	C	<code>IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_PRBS_TYPE_PRBS9</code>
	COM	<code>IviRFSigGenDigitalModulationBasePRBSTypePRBS9</code>
PRBS11	Length of PRBS sequence is $2^{11}-1$ .	
	.NET	<code>DigitalModulationBasePrbsType.Prbs11</code>
	C	<code>IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_PRBS_TYPE_PRBS11</code>
	COM	<code>IviRFSigGenDigitalModulationBasePRBSTypePRBS11</code>
PRBS15	Length of PRBS sequence is $2^{15}-1$ .	
	.NET	<code>DigitalModulationBasePrbsType.Prbs15</code>
	C	<code>IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_PRBS_TYPE_PRBS15</code>



		COM	IviRFSigGenDigitalModulationBasePRBSTypePRBS15
PRBS16	Length of PRBS sequence is $2^{16}-1$ .		
		.NET	DigitalModulationBasePrbsType.Prbs16
		C	IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_PRBS_TYPE_PRBS16
		COM	IviRFSigGenDigitalModulationBasePRBSTypePRBS16
PRBS20	Length of PRBS sequence is $2^{20}-1$ .		
		.NET	DigitalModulationBasePrbsType.Prbs20
		C	IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_PRBS_TYPE_PRBS20
		COM	IviRFSigGenDigitalModulationBasePRBSTypePRBS20
PRBS21	Length of PRBS sequence is $2^{21}-1$ .		
		.NET	DigitalModulationBasePrbsType.Prbs21
		C	IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_PRBS_TYPE_PRBS21
		COM	IviRFSigGenDigitalModulationBasePRBSTypePRBS21
PRBS23	Length of PRBS sequence is $2^{23}-1$ .		
		.NET	DigitalModulationBasePrbsType.Prbs23
		C	IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_PRBS_TYPE_PRBS23
		COM	IviRFSigGenDigitalModulationBasePRBSTypePRBS23

## .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## Compliance Notes

1. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_PRBS_TYPE_CLASS_EXT_BASE` and less than `IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_PRBS_TYPE_SPECIFIC_EXT_BASE`.
2. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_PRBS_TYPE_SPECIFIC_EXT_BASE`.
3. When an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to Digital Modulation Base PRBS Type Specific Ext Base.

See Section Attribute Value Definitions, for the definitions of Digital Modulation Base PRBS Type Specific Ext Base, `IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_PRBS_TYPE_SPECIFIC_EXT_BASE` and `IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_PRBS_TYPE_CLASS_EXT_BASE`.

### 27.2.7 DigitalModulationBase Selected Bit Sequence

Data Type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	Select DigitalModulationBase Bit Sequence

#### .NET Property Name

`DigitalModulation.Base.SelectedBitSequence`

#### COM Property Name

`DigitalModulation.Base.SelectedBitSequence`

#### C Constant Name

`IVIRFSIGGEN_ATTR_DIGITAL_MODULATION_BASE__BIT_SEQUENCE`

#### Description

Specifies name of the bit sequence (stream) used as data for digital modulation. The sequence is used only if DigitalModulationBase Data Source is set to BitSequence.

#### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## 27.2.8 DigitalModulationBase Clock Source

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32 (C/COM)	R/W	N/A	None	Configure DigitalModulationBase Clock Source
ViString (.NET)	R/W	N/A	None	Configure DigitalModulationBase Clock Source

### .NET Property Name

`DigitalModulation.Base.ClockSource`

### COM Property Name

`DigitalModulation.Base.ClockSource`

### COM Enumeration Name

`IviRFSigGenDigitalModulationBaseClockSourceEnum`

### C Constant Name

`IVIRFSIGGEN_ATTR_DIGITAL_MODULATION_BASE_CLOCK_SOURCE`

### Description

Specifies the source of the clock signal used to generate the digital modulation according to the selected standard.

### Defined Values

In IVI.NET the clock source is a string. If an IVI driver supports a clock source and the clock source is listed in IVI-3.3 *Cross Class Capabilities Specification*, Section 3 then the IVI driver shall accept the standard string for that clock source. This attribute is case insensitive, but case preserving. That is the setting is case insensitive but when reading it back the programmed case is returned. IVI specific drivers may define new clock source strings for clock sources that are not defined by IVI-3.3 *Cross Class Capabilities Specification* if needed.

Name	Description	
	Language	Identifier
Internal	The internal clock generator is used.	
	C	IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_CLOCK_SOURCE_INTERNAL
	COM	IviRFSigGenDigitalModulationBaseClockSourceInternal
External	An connected external clock generator (bit or symbol clock frequency) is used.	
	C	IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_CLOCK_SOURCE_EXTERNAL
	COM	IviRFSigGenDigitalModulationBaseClockSourceExternal

## .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## Compliance Notes

1. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_CLOCK_SOURCE_CLASS_EXT_BASE` and less than `IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_CLOCK_SOURCE_SPECIFIC_EXT_BASE`.
2. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_CLOCK_SOURCE_SPECIFIC_EXT_BASE`.
3. When an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to Digital Modulation Base Clock Source Specific Ext Base.

See Section Attribute Value Definitions, for the definitions of Digital Modulation Base Clock Source Specific Ext Base, `IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_CLOCK_SOURCE_SPECIFIC_EXT_BASE` and `IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_CLOCK_SOURCE_CLASS_EXT_BASE`.

## 27.2.9 DigitalModulationBase External Clock Type

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	Configure DigitalModulationBase Clock Source

### .NET Property Name

`DigitalModulation.Base.ExternalClockType`

### .NET Enumeration Name

`DigitalModulationBaseExternalClockType`

### COM Property Name

`DigitalModulation.Base.ExternalClockType`

### COM Enumeration Name

`IviRFSigGenDigitalModulationBaseExternalClockTypeEnum`

### C Constant Name

`IVIRFSIGGEN_ATTR_DIGITAL_MODULATION_BASE_EXTERNAL_CLOCK_TYPE`

### Description

Specifies the type of the external clock signal used to generate the digital modulation. This value is used only if the DigitalModulationBase Clock Source attribute is set to External.

### Defined Values

Name	Description		
		Language	Identifier
Bit	The external clock frequency is equal to the bit clock frequency of the digital modulation.		
		.NET	<code>DigitalModulationBaseExternalClockType.Bit</code>
		C	<code>IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_EXTERNAL_CLOCK_TYPE_BIT</code>
		COM	<code>IviRFSigGenDigitalModulationBaseExternalClockTypeBit</code>
Symbol	The external clock frequency is equal to the symbol clock frequency of the digital modulation.		
		.NET	<code>DigitalModulationBaseExternalClockType.Symbol</code>
		C	<code>IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_EXTERNAL_CLOCK_TYPE_SYMBOL</code>
		COM	<code>IviRFSigGenDigitalModulationBaseExternalClockTypeSymbol</code>

## .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## Compliance Notes

1. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to  
`IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_EXTERNAL_CLOCK_TYPE_CLASS_EXT_BASE` and less than  
`IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_EXTERNAL_CLOCK_TYPE_SPECIFIC_EXT_BASE`.
2. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to  
`IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_EXTERNAL_CLOCK_TYPE_SPECIFIC_EXT_BASE`.
3. When an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to Digital Modulation Base External Clock Type Specific Ext Base.

See Section Attribute Value Definitions, for the definitions of Digital Modulation Base External Clock Type Specific Ext Base,  
`IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_EXTERNAL_CLOCK_TYPE_SPECIFIC_EXT_BASE` and  
`IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_EXTERNAL_CLOCK_TYPE_CLASS_EXT_BASE`.

### **27.3 IviRFSigGenDigitalModulationBase Functions**

The IviRFSigGenDigitalModulationBaseBase Extension Group defines the following functions:

- Get DigitalModulationBaseStandard Name (IVI-C only)
- Select DigitalModulationBase Standard (IVI-C only)
- Configure DigitalModulationBase Clock Source
- Configure DigitalModulationBase Data Source (IVI-C only)
- Configure DigitalModulationBase PRBS Type (IVI-C only)
- Write DigitalModulationBase Bit Sequence
- Select DigitalModulationBase Bit Sequence (IVI-C only)
- Clear All DigitalModulationBase Bit Sequences

This section describes the behavior and requirements of each function.

### 27.3.1 Get DigitalModulationBase Standard Name (IVI-C only)

#### Description

This function returns the specific driver defined DigitalModulationBase standard name that corresponds to the one-based index that the user specifies.

#### .NET Method Prototype

N/A  
(Use DigitalModulation Base Standard Names property.)

#### COM Method Prototype

N/A  
(Use DigitalModulation Base Standard Name property.)

#### C Prototype

```
ViStatus IviRFSigGen_GetDigitalModulationBaseStandardName (ViSession Vi,
                                                           ViInt32 Index,
                                                           ViInt32 NameBufferSize,
                                                           ViChar Name[]);
```

#### Parameters

Inputs	Description
Vi	Instrument handle
Index	A one-based index that defines which name to return.
NameBufferSize	The number of bytes in the ViChar array that the user specifies for the Name parameter.

Outputs	Description
Name	A user-allocated buffer into which the driver stores the DigitalModulationBase standard name.  The caller may pass VI_NULL for this parameter if the NameBufferSize parameter is 0.

#### Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

#### Compliance Notes

For an instrument with only one DigitalModulationBase standard, that is the DigitalModulationBase Standard Count attribute is one, the driver may return an empty string in the Name parameter.



### 27.3.2 Select DigitalModulationBase Standard (IVI-C only)

#### Description

Selects the actual standard used by the instrument. The standard is selected with one of the names of the list of names queried with the Get DigitalModulationBase Standard Names function. The coding, mapping, symbol rate or bit clock frequency, filter together with the according filter. parameters, FSK deviation or ASK depth (in case of FSK or ASK modulation) are set as defined in the selected standard.

#### .NET Method Prototype

N/A  
(use the `DigitalModulation.Base.SelectedStandard` property)

#### COM Method Prototype

N/A  
(use the `DigitalModulation.Base.SelectedStandard` property)

#### C Prototype

```
ViStatus IviRFSigGen_SelectDigitalModulationBaseStandard (ViSession Vi,  
                                                         ViConstString Name);
```

#### Parameters

Inputs	Description
Vi	Instrument handle
Name	The driver uses this value to set the DigitalModulationBase Selected Standard attribute. See the attribute description for more details.

#### Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

### 27.3.3 Configure DigitalModulationBase Clock Source

#### Description

Configures the DigitalModulationBase clock source.

#### .NET Method Prototype

```
void DigitalModulation.Base.ConfigureClockSource(String source,  
DigitalModulationBaseExternalClockType type);
```

#### COM Method Prototype

```
HRESULT DigitalModulation.Base.ConfigureClockSource ([in]  
IviRFSigGenDigitalModulationBaseClockSourceEnum Source,  
[in]  
IviRFSigGenDigitalModulationBaseExternalClockTypeEnum Type);
```

#### C Prototype

```
ViStatus IviRFSigGen_ConfigureDigitalModulationBaseClockSource (ViSession Vi,  
ViInt32 Source, ViInt32 Type);
```

#### Parameters

Inputs	Description
Vi	Instrument handle
Source	The driver uses this value to set the DigitalModulationBase Clock Source attribute. See the attribute description for more details.
Type	Type is ignored if Source is set to Internal. The driver uses this value to set the DigitalModulationBase External Clock Type attribute. See the attribute description for more details.

#### Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

#### .NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

27.3.4 Configure DigitalModulationBase Data Source (IVI-C only)

Description

Configures the source of the data for the digital modulation.

.NET Method Prototype

N/A  
(use the `DigitalModulation.Base.DataSource` property)

COM Method Prototype

N/A  
(use the `DigitalModulation.Base.DataSource` property)

C Prototype

```
ViStatus IviRFSigGen_ConfigureDigitalModulationBaseDataSource (ViSession Vi,  
                                                                ViInt32 Source);
```

Parameters

Inputs	Description
Vi	Instrument handle
Source	The driver uses this value to set the DigitalModulationBase Data Source attribute. See the attribute description for more details.

Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

### 27.3.5 Configure DigitalModulationBase PRBS Type (IVI-C only)

**Description**

Configures the type of the PRBS used as data for the digital modulation. The setting is used only if DigitalModulationBase Data Source is set to PRBS.

**.NET Method Prototype**

N/A  
(use the `DigitalModulation.Base.PrbsType` property)

**COM Method Prototype**

N/A  
(use the `DigitalModulation.Base.PRBSType` property)

**C Prototype**

```
ViStatus IviRFSigGen_ConfigureDigitalModulationBasePRBSType (ViSession Vi,  
                                                             ViInt32 Type);
```

**Parameters**

Inputs	Description
Vi	Instrument handle
Type	The driver uses this value to set the DigitalModulationBase PRBS Type attribute. See the attribute description for more details.

**Return Values (C)**

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

### 27.3.6 Write DigitalModulationBase Bit Sequence

#### Description

Writes the bit sequence used as data for the digital modulation. The sequence is repeated continuously. The string consists of binary values (8 bit in 1 char/byte).

#### .NET Method Prototype

```
void DigitalModulation.Base.CreateBitSequence (String name,  
                                              Long bitCount,  
                                              Byte[] sequence,  
                                              Boolean moreDataPending);
```

#### COM Method Prototype

```
HRESULT DigitalModulation.Base.WriteBitSequence ([in] BSTR Name,  
                                                [in] LONG BitCount,  
                                                [in] SAFEARRAY (BYTE) *Sequence,  
                                                [in] VARIANT_BOOL MoreDataPending);
```

#### C Prototype

```
ViStatus IviRFSigGen_WriteDigitalModulationBaseBitSequence (ViSession Vi,  
                                                           ViConstString Name, ViInt32 BitCount,  
                                                           ViChar Sequence[],  
                                                           ViBoolean MoreDataPending);
```

#### Parameters

Inputs	Description
Vi	Instrument handle
Name	The driver uses this value to name the stored bit sequence.
BitCount	The number of bits in the sequence. One char/byte has 8 bits, but the stream of bits doesn't end necessarily on byte boundaries.
Sequence	The driver uses this array as stream of bits. The size of the array is bitCount.
MoreDataPending	The value is set to True if more data is coming. To complete the sequence, the next data shall have the same name. False means the waveform has no more data to come.

#### Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

#### .NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

### 27.3.7 Select DigitalModulationBase Bit Sequence (IVI-C only)

#### Description

Selects the bit sequence used as data for the digital modulation. The bit sequence is selected with one of the names in the list of names queried with the Get DigitalModulationBase Bit Sequence Names function. The setting is used only if the DigitalModulationBase Data Source attribute is set to BitSequence.

#### .NET Method Prototype

N/A  
(use the `DigitalModulation.Base.SelectBitSequence` property)

#### COM Method Prototype

N/A  
(use the `DigitalModulation.Base.SelectBitSequence` property)

#### C Prototype

```
ViStatus IviRFSigGen_SelectDigitalModulationBaseBitSequence (ViSession Vi,  
                                                             ViConstString Name);
```

#### Parameters

Inputs	Description
Vi	Instrument handle
Name	The driver uses this value to set the DigitalModulationBase Selected Bit Sequence attribute. See the attribute description for more details.

#### Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

### 27.3.8 Clear All DigitalModulationBase Bit Sequences

**Description**

Clears (deletes) all named bit sequences.

**.NET Method Prototype**

```
void DigitalModulation.Base.ClearAllBitSequences ();
```

**COM Method Prototype**

```
HRESULT DigitalModulation.Base.ClearAllBitSequences ();
```

**C Prototype**

```
ViStatus IviRFSigGen_ClearAllDigitalModulationBaseBitSequences (ViSession Vi);
```

**Parameters**

Inputs	Description
Vi	Instrument handle

**Return Values (C/COM)**

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

**.NET Exceptions**

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

## **27.4 IviRFSigGenDigitalModulationBase Behavior Model**

The IviRFSigGenDigitalModulationBase Extension Group follows the behavior model of the IviRFSigGenBase capability group. The only modification to the behavior model from the IviRFSigGenBase capability group is the ability to configure IviRFSigGenDigitalModulationBase settings.

## **27.5 IviRFSigGenDigitalModulationBase Compliance Notes**

If a specific driver implements the IviRFSigGenDigitalModulationBase Extension Group, it shall also implement the IviRFSigGenModulateIQ Extension Group and support the defined value DigitalModulationBase for the IQ Source attribute.

1. If a specific driver does not support the value External for the DigitalModulationBase Clock Source, it shall not support the External Clock Type attribute.
2. If a specific driver does not support the value PRBS for the DigitalModulationBase Data Source attribute, it shall not implement the DigitalModulationBase PRBS Type attribute and the Configure DigitalModulationBase PRBS Type function.
3. If a specific driver does not support the value Bit Sequence for the DigitalModulationBase attribute, it shall not implement the DigitalModulationBase Selected Bit Sequence attribute and the Write DigitalModulationBase Bit Sequence, Select DigitalModulation Bit Sequence and Clear All DigitalModulationBase Bit Sequences functions.



## 28. IviRFSigGenCDMABase Extension Group

---

### 28.1 *IviRFSigGenCDMABase Extension Group Overview*

With IviRFSigGenCDMABase Extension Group the user can simulate mobile (cell) phone modulation based on Code Division Multiple Access (CDMA) standards.

### 28.2 *IviRFSigGenCDMABase Attributes*

The IviRFSigGenCDMABase extension group defines the following attributes:

- CDMA Standard Names (IVI.NET only)
- CDMA Standard Name (IVI-COM only)
- CDMA Standard Count (IVI-C and IVI-COM only)
- CDMA Selected Standard
- CDMA Trigger Source
- CDMA External Trigger Slope
- CDMA Test Model Names (IVI.NET only)
- CDMA Test Model Name (IVI-COM only)
- CDMA Test Model Count (IVI-C and IVI-COM only)
- CDMA Selected Test Model
- CDMA Clock Source

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 30, *IviRFSigGen Attribute ID Definitions*.

## 28.2.1 CDMA Standard Names (IVI.NET only)

Data Type	Access	Applies to	Coercion	High Level Functions
ReadOnlyCollection<String>	RO	N/A	None	None

### .NET Constant Name

`ReadOnlyCollection<String> DigitalModulation.CDMA.StandardNames`

### COM Property Name

N/A

(Use the CDMA Standard Name property.)

### C Constant Name

N/A

(Use the Get CDMA Standard Name function.)

### Description

This property returns a collection of the currently defined Digital Modulation CDMA Standard identifiers.

### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## 28.2.2 CDMA Standard Name (IVI-COM only)

Data Type	Access	Applies to	Coercion	High Level Functions
ViString	RO	N/A	None	None

### .NET Constant Name

N/A

(Use the CDMA Standard Names property.)

### COM Property Name

```
HRESULT DigitalModulation.CDMA.StandardName ([in] long Index,  
                                              [out, retval] BSTR* Name);
```

### C Constant Name

N/A

(Use the Get CDMA Standard Name function.)

### Description

This property returns the CDMA Standard identifier that corresponds to the one-based index that the user specifies. If the value that the user pass for the Index parameter is less than one or greater than the value of the CDMA Standard Count attribute, the property returns an empty string in the Name parameter and returns the Invalid Value error.

### Parameters

Inputs	Description	Base Type
Index	A one-based index that defines which name to return.	ViInt32

Output/Return Value	Description	Base Type
Name	A driver-allocated buffer into which the driver stores the CDMA standard name.	ViChar[]

### 28.2.3 CDMA Standard Count (IVI-C and IVI-COM only)

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32	RO	N/A	None	N/A

#### .NET Property Name

N/A  
(Use CDMA Standard Names Count property.)

#### COM Property Name

DigitalModulation.CDMA.StandardCount

#### C Constant Name

IVIRFSIGGEN\_ATTR\_CDMA\_STANDARD\_COUNT

#### Description

Specifies the number of CDMA standards available for a particular instrument.

## 28.2.4 CDMA Selected Standard

Data Type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	Select CDMA Standard

### .NET Property Name

`DigitalModulation.Cdma.SelectedStandard`

### COM Property Name

`DigitalModulation.CDMA.SelectedStandard`

### C Constant Name

`IVIRFSIGGEN_ATTR_CDMA_SELECTED_STANDARD`

### Description

Specifies the actual standard used by the instrument. The modulation type, bit clock frequency and filter together with the according filter. parameters are set as defined in the selected standard.

### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## 28.2.5 CDMA Trigger Source

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32 (C/COM)	R/W	N/A	None	Configure CDMA Trigger Source
ViString (.NET)	R/W	N/A	None	Configure CDMA Trigger Source

### .NET Property Name

`DigitalModulation.Cdma.TriggerSource`

### COM Property Name

`DigitalModulation.CDMA.TriggerSource`

### COM Enumeration Name

`IviRFSigGenCDMATriggerSourceEnum`

### C Constant Name

`IVIRFSIGGEN_ATTR_CDMA_TRIGGER_SOURCE`

### Description

Specifies the source of the trigger signal that starts the channel coding generation.

### Defined Values

In IVI.NET the trigger input is a string. If an IVI driver supports a trigger input and the trigger input is listed in IVI-3.3 *Cross Class Capabilities Specification*, Section 3 then the IVI driver shall accept the standard string for that trigger input. This attribute is case insensitive, but case preserving. That is the setting is case insensitive but when reading it back the programmed case is returned. IVI specific drivers may define new trigger input strings for trigger inputs that are not defined by IVI-3.3 *Cross Class Capabilities Specification* if needed.

Name	Description	
	Language	Identifier
Immediate	The CDMA generator system does not wait for a trigger of any kind, so it is running the channel codings continuously.	
	C	IVIRFSIGGEN_VAL_CDMA_TRIGGER_SOURCE_IMMEDIATE
	COM	IviRFSigGenCDMATriggerSourceImmediate
External	Each channel coding is started with an external signal.	
	C	IVIRFSIGGEN_VAL_CDMA_TRIGGER_SOURCE_EXTERNAL
	COM	IviRFSigGenCDMATriggerSourceExternal
Software Trigger	Each channel coding is started with a software programmable trigger.	
	C	IVIRFSIGGEN_VAL_CDMA_TRIGGER_SOURCE_SOFTWARE
	COM	IviRFSigGenCDMATriggerSourceSoftware

## .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## Compliance Notes

1. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_CDMA_TRIGGER_SOURCE_CLASS_EXT_BASE` and less than `IVIRFSIGGEN_VAL_CDMA_TRIGGER_SOURCE_SPECIFIC_EXT_BASE`.
2. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_CDMA_TRIGGER_SOURCE_SPECIFIC_EXT_BASE`.
3. When an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to CDMA Trigger Source Specific Ext Base.

See Section Attribute Value Definitions, for the definitions of CDMA Trigger Source Specific Ext Base, `IVIRFSIGGEN_VAL_CDMA_TRIGGER_SOURCE_SPECIFIC_EXT_BASE` and `IVIRFSIGGEN_VAL_CDMA_TRIGGER_SOURCE_CLASS_EXT_BASE`.

## 28.2.6 CDMA External Trigger Slope

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	Configure CDMA External Trigger Slope

### .NET Property Name

`DigitalModulation.Cdma.ExternalTriggerSlope`

### .NET Enumeration Name

`Slope`

### COM Property Name

`DigitalModulation.CDMA.ExternalTriggerSlope`

### COM Enumeration Name

`IviRFSigGenCDMAExternalTriggerSlopeEnum`

### C Constant Name

`IVIRFSIGGEN_ATTR_CDMA_EXTERNAL_TRIGGER_SLOPE`

### Description

Specifies whether the trigger event occurs on the rising or falling edge of the input signal.

### Defined Values

Name	Description		
		Language	Identifier
Positive	Enables rising edge triggering.		
		.NET	<code>Slope.Positive</code>
		C	<code>IVIRFSIGGEN_VAL_CDMA_EXTERNAL_TRIGGER_SLOPE_POSITIVE</code>
		COM	<code>IviRFSigGenCDMAExternalTriggerSlopePositive</code>
Negative	Enables falling edge triggering.		
		.NET	<code>Slope.Negative</code>
		C	<code>IVIRFSIGGEN_VAL_CDMA_EXTERNAL_TRIGGER_SLOPE_NEGATIVE</code>
		COM	<code>IviRFSigGenCDMAExternalTriggerSlopeNegative</code>

### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.



## Compliance Notes

1. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_CDMA_EXTERNAL_TRIGGER_SLOPE_CLASS_EXT_BASE` and less than `IVIRFSIGGEN_VAL_CDMA_EXTERNAL_TRIGGER_SLOPE_SPECIFIC_EXT_BASE`.
2. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_CDMA_EXTERNAL_TRIGGER_SLOPE_SPECIFIC_EXT_BASE`.
4. When an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to CDMA Trigger External Slope Specific Ext Base.

See Section Attribute Value Definitions, for the definitions of CDMA Trigger External Slope Specific Ext Base, `IVIRFSIGGEN_VAL_CDMA_TRIGGER_EXTERNAL_SLOPE_SPECIFIC_EXT_BASE` and `IVIRFSIGGEN_VAL_CDMA_TRIGGER_EXTERNAL_SLOPE_CLASS_EXT_BASE`.

28.2.7 CDMA Test Model Names (IVI.NET only)

Data Type	Access	Applies to	Coercion	High Level Functions
ReadOnlyCollection<String>	RO	N/A	None	None

**.NET Property Name**

ReadOnlyCollection<String> DigitalModulation.CDMA.TestModelNames

**COM Property Name**

(Use the CDMA Test Model Name property.)

**C Constant Name**

N/A  
(Use the Get CDMA Test Model Name function.)

**Description**

This property returns a collection of the currently defined Digital Modulation CDMA Test Model identifiers.

**.NET Exceptions**

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## 28.2.8 CDMA Test Model Name (IVI-COM only)

Data Type	Access	Applies to	Coercion	High Level Functions
ViString	RO	N/A	None	None

### .NET Property Name

N/A

(Use the CDMA Test Model Names property.)

### COM Property Name

```
HRESULT DigitalModulation.CDMA.TestModelName ([in] long Index,  
                                                [out, retval] BSTR* Name);
```

### C Constant Name

N/A

(Use the Get CDMA Test Model Name function.)

### Description

This property returns the CDMA Test Model identifier that corresponds to the one-based index that the user specifies. If the value that the user pass for the Index parameter is less than one or greater than the value of the CDMA Test Model Count attribute, the property returns an empty string in the Name parameter and returns the Invalid Value error.

### Parameters

Inputs	Description	Base Type
Index	A one-based index that defines which name to return.	ViInt32

Output/Return Value	Description	Base Type
Name	A driver-allocated buffer into which the driver stores the CDMA test model name.	ViChar[]

### 28.2.9 CDMA Test Model Count (IVI-C and IVI-COM only)

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32	RO	N/A	None	N/A

#### .NET Property Name

N/A  
(Use CDMA Test Model Names Count property.)

#### COM Property Name

DigitalModulation.CDMA.TestModelCount

#### C Constant Name

IVIRFSIGGEN\_ATTR\_CDMA\_TEST\_MODEL\_COUNT

#### Description

Specifies the number of CDMA test models available for a particular instrument.

## 28.2.10 CDMA Selected Test Model

Data Type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	Select CDMA Test Model

### .NET Property Name

`DigitalModulation.Cdma.SelectedTestModel`

### COM Property Name

`DigitalModulation.CDMA.SelectedTestModel`

### C Constant Name

`IVIRFSIGGEN_ATTR_CDMA_SELECTED_TEST_MODEL`

### Description

Specifies the actual CDMA test model used by the instrument.

### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## 28.2.11 CDMA Clock Source

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32 (C/COM)	R/W	N/A	None	Configure CDMA Clock Source
ViString (.NET)	R/W	N/A	None	Configure CDMA Clock Source

### .NET Property Name

`DigitalModulation.Cdma.ClockSource`

### COM Property Name

`DigitalModulation.CDMA.ClockSource`

### COM Enumeration Name

`IviRFSigGenCDMAClockSourceEnum`

### C Constant Name

`IVIRFSIGGEN_ATTR_CDMA_CLOCK_SOURCE`

### Description

Specifies the source of the clock signal used to generate the digital modulation according to the selected standard.

### Defined Values

In IVI.NET the CDMA clock source is a string. If an IVI driver supports a CDMA clock source and the CDMA clock source is listed in IVI-3.3 *Cross Class Capabilities Specification*, Section 3 then the IVI driver shall accept the standard string for that CDMA clock source. This attribute is case insensitive, but case preserving. That is the setting is case insensitive but when reading it back the programmed case is returned. IVI specific drivers may define new trigger input strings for CDMA clock sources that are not defined by IVI-3.3 *Cross Class Capabilities Specification* if needed.

Name	Description		
		Language	Identifier
Internal	The internal clock generator is used.		
		C	IVIRFSIGGEN_VAL_CDMA_CLOCK_SOURCE_INTERNAL
		COM	IviRFSigGenCDMAClockSourceInternal
External	A connected external clock generator (bit or symbol clock frequency) is used.		
		C	IVIRFSIGGEN_VAL_CDMA_CLOCK_SOURCE_EXTERNAL
		COM	IviRFSigGenCDMAClockSourceExternal

### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## Compliance Notes

1. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_CDMA_CLOCK_SOURCE_CLASS_EXT_BASE` and less than `IVIRFSIGGEN_VAL_CDMA_CLOCK_SOURCE_SPECIFIC_EXT_BASE`.
2. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_CDMA_CLOCK_SOURCE_SPECIFIC_EXT_BASE`.
3. When an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to CDMA Clock Source Specific Ext Base.

See Section Attribute Value Definitions, for the definitions of CDMA Clock Source Specific Ext Base, `IVIRFSIGGEN_VAL_CDMA_CLOCK_SOURCE_SPECIFIC_EXT_BASE` and `IVIRFSIGGEN_VAL_CDMA_CLOCK_SOURCE_CLASS_EXT_BASE`.

### **28.3 IviRFSigGenCDMABase Functions**

The IviRFSigGenCDMABase extension group defines the following functions:

- Get CDMA Standard Name (IVI-C only)
- Select CDMA Standard (IVI-C only)
- Configure CDMA Clock Source (IVI-C only)
- Configure CDMA Trigger Source (IVI-C only)
- Configure CDMA External Trigger Slope (IVI-C only)
- Get CDMA Test Model Name (IVI-C only)
- Select CDMA Test Model (IVI-C only)

This section describes the behavior and requirements of each function.



### 28.3.1 Get CDMA Standard Name (IVI-C only)

#### Description

This function returns the specific driver defined CDMA standard name that corresponds to the one-based index that the user specifies.

#### .NET Method Prototype

(Use the CDMA Standard Names property.)

#### COM Method Prototype

N/A

(Use the CDMA Standard Name property.)

#### C Prototype

```
ViStatus IviRFSigGen_GetCDMAStandardName (ViSession Vi,  
                                           ViInt32 Index,  
                                           ViInt32 NameBufferSize,  
                                           ViChar Name[]);
```

#### Parameters

Inputs	Description
Vi	Instrument handle
Index	A one-based index that defines which name to return.
NameBufferSize	The number of bytes in the ViChar array that the user specifies for the Name parameter.

Outputs	Description
Name	A user-allocated buffer into which the driver stores the CDMA standard name. The caller may pass VI_NULL for this parameter if the NameBufferSize parameter is 0.

#### Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

#### Compliance Notes

For an instrument with only one CDMA standard, that is the CDMA Standard Count attribute is one, the driver may return an empty string in the Name parameter.

### 28.3.2 Select CDMA Standard (IVI-C only)

**Description**

Selects the actual standard used by the instrument. The standard is selected from the list of names queried with the Get CDMA Standard Names function. The modulation type, clock frequency and filter together with the according filter. parameters are set as defined in the selected standard.

**.NET Method Prototype**

N/A  
(use the `DigitalModulation.Cdma.SelectedStandard` property)

**COM Method Prototype**

N/A  
(use the `DigitalModulation.CDMA.SelectedStandard` property)

**C Prototype**

```
ViStatus IviRFSigGen_SelectCDMAStandard (ViSession Vi, ViConstString Name);
```

**Parameters**

Inputs	Description
Vi	Instrument handle
Name	The driver uses this value to set the CDMA Selected Standard attribute. See the attribute description for more details.

**Return Values (C)**

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

### 28.3.3 Configure CDMA Clock Source (IVI-C only)

**Description**

Configures the CDMA clock source.

**.NET Method Prototype**

N/A  
(use the `DigitalModulation.Cdma.ClockSource` property)

**COM Method Prototype**

N/A  
(use the `DigitalModulation.CDMA.ClockSource` property)

**C Prototype**

```
ViStatus IviRFSigGen_ConfigureCDMAClockSource (ViSession Vi, ViInt32 Source);
```

**Parameters**

Inputs	Description
Vi	Instrument handle
Source	The driver uses this value to set the CDMA Clock Source attribute. See the attribute description for more details.

**Return Values (C)**

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

### 28.3.4 Configure CDMA Trigger Source (IVI-C only)

**Description**

Configures the CDMA trigger source for starting or synchronising the generation of the channel codings.

**.NET Method Prototype**

N/A  
(use the `DigitalModulation.Cdma.TriggerSource` property)

**COM Method Prototype**

N/A  
(use the `DigitalModulation.CDMA.TriggerSource` property)

**C Prototype**

```
ViStatus IviRFSigGen_ConfigureCDMATriggerSource (ViSession Vi, ViInt32 Source);
```

**Parameters**

Inputs	Description
Vi	Instrument handle
Source	The driver uses this value to set the CDMA Trigger Source attribute. See the attribute description for more details.

**Return Values (C)**

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

### 28.3.5 Configure CDMA External Trigger Slope (IVI-C only)

**Description**

Configures whether the trigger event occurs on the rising or falling edge of the input signal. This setting is used only if CDMA Trigger Source is set to External.

**.NET Method Prototype**

N/A  
(use the `DigitalModulation.Cdma.ExternalTriggerSlope` property)

**COM Method Prototype**

N/A  
(use the `DigitalModulation.CDMA.ExternalTriggerSlope` property)

**C Prototype**

```
ViStatus IviRFSigGen_ConfigureCDMAExternalTriggerSlope (ViSession Vi,  
                                                         ViInt32 Slope);
```

**Parameters**

Inputs	Description
Vi	Instrument handle
Slope	The driver uses this value to set the CDMA External Trigger Slope attribute. See the attribute description for more details.

**Return Values (C)**

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

### 28.3.6 Get CDMA Test Model Name (IVI-C only)

#### Description

This function returns the specific driver defined CDMA test model name that corresponds to the one-based index that the user specifies.

#### .NET Method Prototype

N/A  
(Use the CDMA Test Model Names property.)

#### COM Method Prototype

N/A  
(Use the CDMA Test Model Name property.)

#### C Prototype

```
ViStatus IviRFSigGen_GetCDMATestModelName (ViSession Vi,  
                                             ViInt32 Index,  
                                             ViInt32 NameBufferSize,  
                                             ViChar Name[]);
```

#### Parameters

Inputs	Description
Vi	Instrument handle
Index	A one-based index that defines which name to return.
BufferSize	The number of bytes in the ViChar array that the user specifies for the Name parameter.

Outputs	Description
Name	A user-allocated buffer into which the driver stores the CDMA test model name. The caller may pass VI_NULL for this parameter if the NameBufferSize parameter is 0.

#### Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

#### Compliance Notes

For an instrument with only one CDMA test model, that is the CDMA Test Model Count attribute is one, the driver may return an empty string in the Name parameter.

### 28.3.7 Select CDMA Test Model (IVI-C only)

#### Description

Selects the channel coding used for the digital modulation. The channel coding is selected with one of the names in the list of names queried with the Get CDMA Test Model Names function.

#### .NET Method Prototype

N/A  
(use the `DigitalModulation.Cdma.SelectedTestModel` property)

#### COM Method Prototype

N/A  
(use the `DigitalModulation.CDMA.SelectedTestModel` property)

#### C Prototype

```
ViStatus IviRFSigGen_SelectCDMATestModel (ViSession Vi, ViConstString Name);
```

#### Parameters

Inputs	Description
Vi	Instrument handle
Name	The driver uses this value to set the CDMA Selected Test Model attribute. See the attribute description for more details. See the attribute description for more details.

#### Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

## **28.4 *IviRFSigGenCDMABase Behavior Model***

The IviRFSigGenCDMABase Extension Group follows the behavior model of the IviRFSigGenBase capability group. The only modification to the behavior model from the IviRFSigGenBase capability group is the ability to configure IviRFSigGenCDMABase settings.

## **28.5 *IviRFSigGenCDMABase Compliance Notes***

For a specific driver to comply with the IviRFSigGenCDMABase extension, it shall also implement the IviRFSigGenModulateIQ Extension Group and support the defined value CDMABase for the IQ Source attribute.

1. If a specific driver does not support the defined value External for the CDMA Trigger Source attribute, it shall not support the CDMA External Trigger Slope attribute or the Configure CDMA External Trigger Slope function.



## 29. IviRFSigGenTDMABase Extension Group

---

### 29.1 *IviRFSigGenTDMABase Extension Group Overview*

With IviRFSigGenTDMABase Extension Group the user can simulate mobile (cell) phone modulation based on Time Division Multiple Access (TDMA) standards.

### 29.2 *IviRFSigGenTDMABase Attributes*

The IviRFSigGenTDMABase extension group defines the following attributes:

- TDMA Standard Names (IVI.NET only)
- TDMA Standard Name (IVI-COM only)
- TDMA Standard Count (IVI-C and IVI-COM only)
- TDMA Selected Standard
- TDMA Trigger Source
- TDMA External Trigger Slope
- TDMA Frame Names (IVI.NET only)
- TDMA Frame Name (IVI-COM only)
- TDMA Frame Count (IVI-C and IVI-COM only)
- TDMA Selected Frame
- TDMA Clock Source
- TDMA External Clock Type

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 30, *IviRFSigGen Attribute ID Definitions*.

29.2.1 TDMA Standard Names (IVI.NET only)

Data Type	Access	Applies to	Coercion	High Level Functions
ReadOnlyCollection<String>	RO	N/A	None	None

**.NET Property Name**

ReadOnlyCollection<String> DigitalModulation.TDMA.StandardNames

**COM Property Name**

N/A  
(Use the TDMA Standard Name property.)

**C Constant Name**

N/A  
(Use the Get TDMA Standard Name function.)

**Description**

This property returns a collection of the currently defined Digital Modulation TDMA Standard identifiers.

**.NET Exceptions**

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

### 29.2.2 TDMA Standard Name (IVI-COM only)

Data Type	Access	Applies to	Coercion	High Level Functions
ViString	RO	N/A	None	None

#### .NET Property Name

N/A

(Use the TDMA Standard Names property.)

#### COM Property Name

```
HRESULT DigitalModulation.TDMA.StandardName ([in] long Index,  
                                              [out, retval] BSTR* Name);
```

#### C Constant Name

N/A

(Use the Get TDMA Standard Names function.)

#### Description

This property returns the TDMA Standard identifier that corresponds to the one-based index that the user specifies. If the value that the user pass for the Index parameter is less than one or greater than the value of the TDMA Standard Count attribute, the property returns an empty string in the Name parameter and returns the Invalid Value error.

#### Parameters

Inputs	Description	Base Type
Index	A one-based index that defines which name to return.	ViInt32

Output/Return Value	Description	Base Type
Name	A driver-allocated buffer into which the driver stores the TDMA standard name.	ViChar[]

### 29.2.3 TDMA Standard Count (IVI-C and IVI-COM only)

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32	RO	N/A	None	N/A

#### .NET Property Name

N/A  
(Use TDMA Standard Names Count property.)

#### COM Property Name

DigitalModulation.TDMA.StandardCount

#### C Constant Name

IVIRFSIGGEN\_ATTR\_TDMA\_STANDARD\_COUNT

#### Description

Specifies the number of TDMA standards available for a particular instrument.

## 29.2.4 TDMA Selected Standard

Data Type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	Select TDMA Standard

### .NET Property Name

`DigitalModulation.Tdma.SelectedStandard`

### COM Property Name

`DigitalModulation.TDMA.SelectedStandard`

### C Constant Name

`IVIRFSIGGEN_ATTR_TDMA_SELECTED_STANDARD`

### Description

Specifies the actual standard used by the instrumen. The coding, mapping, symbol rate or bit clock frequency, filter together with the according filter, parameters, FSK deviation or ASK depth (in case of FSK or ASK modulation) are set as defined in the selected standard.

### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## 29.2.5 TDMA Trigger Source

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32 (C/COM)	R/W	N/A	None	Configure TDMA Trigger Source
ViString (.NET)	R/W	N/A	None	Configure TDMA Trigger Source

### .NET Property Name

`DigitalModulation.Tdma.TriggerSource`

### COM Property Name

`DigitalModulation.TDMA.TriggerSource`

### COM Enumeration Name

`IviRFSigGenTDMATriggerSourceEnum`

### C Constant Name

`IVIRFSIGGEN_ATTR_TDMA_TRIGGER_SOURCE`

### Description

Specifies the source of the trigger signal that starts the frame/slot generation.

### Defined Values

In IVI.NET the trigger source is a string. If an IVI driver supports a trigger source and the trigger source is listed in IVI-3.3 *Cross Class Capabilities Specification*, Section 3 then the IVI driver shall accept the standard string for that trigger source. This attribute is case insensitive, but case preserving. That is the setting is case insensitive but when reading it back the programmed case is returned. IVI specific drivers may define new trigger source strings for trigger sources that are not defined by IVI-3.3 *Cross Class Capabilities Specification* if needed.

Name	Description	
	Language	Identifier
Immediate	The TDMA generator system does not wait for a trigger of any kind, so it is running the frames continuously.	
	C	IVIRFSIGGEN_VAL_TDMA_TRIGGER_SOURCE_IMMEDIATE
	COM	IviRFSigGenTDMATriggerSourceImmediate
External	Each frame is started with an external signal.	
	C	IVIRFSIGGEN_VAL_TDMA_TRIGGER_SOURCE_EXTERNAL
	COM	IviRFSigGenTDMATriggerSourceExternal
Software Trigger	Each frame is started with a software programmable trigger.	
	C	IVIRFSIGGEN_VAL_TDMA_TRIGGER_SOURCE_SOFTWARE
	COM	IviRFSigGenTDMATriggerSourceSoftware

## **.NET Exceptions**

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## **Compliance Notes**

1. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_TDMA_TRIGGER_SOURCE_CLASS_EXT_BASE` and less than `IVIRFSIGGEN_VAL_TDMA_TRIGGER_SOURCE_SPECIFIC_EXT_BASE`.
2. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_TDMA_TRIGGER_SOURCE_SPECIFIC_EXT_BASE`.
3. When an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to `TDMA Trigger Source Specific Ext Base`.

See Section Attribute Value Definitions, for the definitions of `TDMA Trigger Source Specific Ext Base`, `IVIRFSIGGEN_VAL_TDMA_TRIGGER_SOURCE_SPECIFIC_EXT_BASE` and `IVIRFSIGGEN_VAL_TDMA_TRIGGER_SOURCE_CLASS_EXT_BASE`.

## 29.2.6 TDMA External Trigger Slope

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	Configure TDMA External Trigger Slope

### .NET Property Name

`DigitalModulation.Tdma.ExternalTriggerSlope`

### .NET Enumeration Name

`Slope`

### COM Property Name

`DigitalModulation.TDMA.ExternalTriggerSlope`

### COM Enumeration Name

`IviRFSigGenTDMAExternalTriggerSlopeEnum`

### C Constant Name

`IVIRFSIGGEN_ATTR_TDMA_EXTERNAL_TRIGGER_SLOPE`

### Description

Specifies whether the trigger event occurs on the rising or falling edge of the input signal.

### Defined Values

Name	Description		
		Language	Identifier
Positive	Enables rising edge triggering.		
		.NET	<code>Slope.Positive</code>
		C	<code>IVIRFSIGGEN_VAL_TDMA_EXTERNAL_TRIGGER_SLOPE_POSITIVE</code>
		COM	<code>IviRFSigGenTDMAExternalTriggerSlopePositive</code>
Negative	Enables falling edge triggering.		
		.NET	<code>Slope.Negative</code>
		C	<code>IVIRFSIGGEN_VAL_TDMA_EXTERNAL_TRIGGER_SLOPE_NEGATIVE</code>
		COM	<code>IviRFSigGenTDMAExternalTriggerSlopeNegative</code>

### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.



## Compliance Notes

1. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_TDMA_EXTERNAL_TRIGGER_SLOPE_CLASS_EXT_BASE` and less than `IVIRFSIGGEN_VAL_TDMA_EXTERNAL_TRIGGER_SLOPE_SPECIFIC_EXT_BASE`.
2. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_TDMA_EXTERNAL_TRIGGER_SLOPE_SPECIFIC_EXT_BASE`.
5. When an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to TDMA External Trigger Slope Specific Ext Base.

See Section Attribute Value Definitions, for the definitions of TDMA External Trigger Slope Specific Ext Base, `IVIRFSIGGEN_VAL_TDMA_EXTERNAL_TRIGGER_SLOPE_SPECIFIC_EXT_BASE` and `IVIRFSIGGEN_VAL_TDMA_EXTERNAL_TRIGGER_SLOPE_CLASS_EXT_BASE`.

29.2.7 TDMA Frame Names (IVI.NET only)

Data Type	Access	Applies to	Coercion	High Level Functions
ReadOnlyCollection<String>	RO	N/A	None	None

**.NET Property Name**

`ReadOnlyCollection<String> DigitalModulation.TDMA.FrameNames`

**COM Property Name**

N/A  
(Use the TDMA Frame Name property.)

**C Constant Name**

N/A  
(Use the Get TDMA Frame Name function.)

**Description**

This property returns a collection of the currently defined Digital Modulation TDMA Frame identifiers.

**.NET Exceptions**

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

### 29.2.8 TDMA Frame Name (IVI-COM only)

Data Type	Access	Applies to	Coercion	High Level Functions
ViString	RO	N/A	None	None

#### .NET Property Name

N/A

(Use the TDMA Frame Names property.)

#### COM Property Name

```
HRESULT DigitalModulation.TDMA.FrameName ([in] long Index,  
                                             [out, retval] BSTR* Name);
```

#### C Constant Name

N/A

(Use the Get TDMA Frame Name function.)

#### Description

This property returns the TDMA Frame identifier that corresponds to the one-based index that the user specifies. If the value that the user pass for the Index parameter is less than one or greater than the value of the TDMA Frame Count attribute, the property returns an empty string in the Name parameter and returns the Invalid Value error.

#### Parameters

Inputs	Description	Base Type
Index	A one-based index that defines which name to return.	ViInt32

Output/Return Value	Description	Base Type
Name	A driver-allocated buffer into which the driver stores the TDMA frame name.	ViChar[]

### 29.2.9 TDMA Frame Count (IVI-C and IVI-COM only)

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32	RO	N/A	None	N/A

#### .NET Property Name

N/A  
(Use TDMA Frame Names Count property.)

#### COM Property Name

DigitalModulation.TDMA.FrameCount

#### C Constant Name

IVIRFSIGGEN\_ATTR\_TDMA\_FRAME\_COUNT

#### Description

Specifies the number of TDMA frames available for a particular instrument.

29.2.10 TDMA Selected Frame

Data Type	Access	Applies to	Coercion	High Level Functions
ViString	R/W	N/A	None	Select TDMA Frame

**.NET Property Name**

`DigitalModulation.Tdma.SelectedFrame`

**COM Property Name**

`DigitalModulation.TDMA.SelectedFrame`

**C Constant Name**

`IVIRFSIGGEN_ATTR_TDMA_SELECTED_FRAME`

**Description**

Specifies the actual frame used by the instrument. It is selected from the list queried with the function Get TDMA Frame Names.

**.NET Exceptions**

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## 29.2.11 TDMA Clock Source

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32 (C/COM)	R/W	N/A	None	Configure TDMA Clock Source
ViString (.NET)	R/W	N/A	None	Configure TDMA Clock Source

### .NET Property Name

`DigitalModulation.Tdma.ClockSource`

### COM Property Name

`DigitalModulation.TDMA.ClockSource`

### COM Enumeration Name

`IviRFSigGenTDMAClockSourceEnum`

### C Constant Name

`IVIRFSIGGEN_ATTR_TDMA_CLOCK_SOURCE`

### Description

Specifies the source of the clock signal used to generate the digital modulation according to the selected standard.

### Defined Values

In IVI.NET the TDMA clock source is a string. If an IVI driver supports a TDMA clock source and the TDMA clock source is listed in IVI-3.3 *Cross Class Capabilities Specification*, Section 3 then the IVI driver shall accept the standard string for that TDMA clock source. This attribute is case insensitive, but case preserving. That is the setting is case insensitive but when reading it back the programmed case is returned. IVI specific drivers may define new TDMA clock source strings for TDMA clock sources that are not defined by IVI-3.3 *Cross Class Capabilities Specification* if needed.

Name	Description	
	Language	Identifier
Internal	The internal clock generator is used.	
	C	IVIRFSIGGEN_VAL_TDMA_CLOCK_SOURCE_INTERNAL
	COM	IviRFSigGenTDMAClockSourceInternal
External	A connected external clock generator (bit or symbol clock frequency) is used.	
	C	IVIRFSIGGEN_VAL_TDMA_CLOCK_SOURCE_EXTERNAL
	COM	IviRFSigGenTDMAClockSourceExternal

### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.

## Compliance Notes

1. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_TDMA_CLOCK_SOURCE_CLASS_EXT_BASE` and less than `IVIRFSIGGEN_VAL_TDMA_CLOCK_SOURCE_SPECIFIC_EXT_BASE`.
2. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_TDMA_CLOCK_SOURCE_SPECIFIC_EXT_BASE`.
3. When an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to TDMA Clock Source Specific Ext Base.

See Section Attribute Value Definitions, for the definitions of TDMA Clock Source Specific Ext Base, `IVIRFSIGGEN_VAL_TDMA_CLOCK_SOURCE_SPECIFIC_EXT_BASE` and `IVIRFSIGGEN_VAL_TDMA_CLOCK_SOURCE_CLASS_EXT_BASE`.

## 29.2.12 TDMA External Clock Type

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	N/A	None	Configure TDMA Clock Source

### .NET Property Name

`DigitalModulation.Tdma.ExternalClockType`

### .NET Enumeration Name

`TdmaExternalClockType`

### COM Property Name

`DigitalModulation.TDMA.ExternalClockType`

### COM Enumeration Name

`IviRFSigGenTDMAExternalClockTypeEnum`

### C Constant Name

`IVIRFSIGGEN_ATTR_TDMA_EXTERNAL_CLOCK_TYPE`

### Description

Specifies the type of the external clock signal used to generate the digital modulation. This value is used only if TDMA Clock Source is set to External.

### Defined Values

Name	Description		
		Language	Identifier
Bit	The external clock frequency is equal to the bit clock frequency of the digital modulation.		
		.NET	<code>TdmaExternalClockType.Bit</code>
		C	<code>IVIRFSIGGEN_VAL_TDMA_EXTERNAL_CLOCK_TYPE_BIT</code>
		COM	<code>IviRFSigGenTDMAExternalClockTypeBit</code>
Symbol	The external clock frequency is equal to the symbol clock frequency of the digital modulation.		
		.NET	<code>TdmaExternalClockType.Symbol</code>
		C	<code>IVIRFSIGGEN_VAL_TDMA_EXTERNAL_CLOCK_TYPE_SYMBOL</code>
		COM	<code>IviRFSigGenTDMAExternalClockTypeSymbol</code>

### .NET Exceptions

The IVI-3.2: Inherent Capabilities Specification defines general exceptions that may be thrown, and warning events that may be raised, by this property.



## Compliance Notes

1. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_TDMA_EXTERNAL_CLOCK_TYPE_CLASS_EXT_BASE` and less than `IVIRFSIGGEN_VAL_TDMA_EXTERNAL_CLOCK_TYPE_SPECIFIC_EXT_BASE`.
2. If an IVI-C specific driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIRFSIGGEN_VAL_TDMA_EXTERNAL_CLOCK_TYPE_SPECIFIC_EXT_BASE`.
3. When an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to TDMA External Clock Type Specific Ext Base.

See Section Attribute Value Definitions, for the definitions of TDMA External Clock Type Specific Ext Base, `IVIRFSIGGEN_VAL_TDMA_EXTERNAL_CLOCK_TYPE_SPECIFIC_EXT_BASE` and `IVIRFSIGGEN_VAL_TDMA_EXTERNAL_CLOCK_TYPE_CLASS_EXT_BASE`.

### **29.3 *IviRFSigGenTDMABase Functions***

The IviRFSigGenTDMABase extension group defines the following functions:

- Get TDMA Standard Name (IVI-C only)
- Select TDMA Standard (IVI-C only)
- Configure TDMA Clock Source
- Configure TDMA Trigger Source (IVI-C only)
- Configure TDMA External Trigger Slope (IVI-C only)
- Get TDMA Frame Name (IVI-C only)
- Select TDMA Frame (IVI-C only)

This section describes the behavior and requirements of each function.

### 29.3.1 Get TDMA Standard Name (IVI-C only)

#### Description

This function returns the specific driver defined TDMA standard name that corresponds to the one-based index that the user specifies.

#### .NET Method Prototype

N/A  
(Use the TDMA Standard Names property.)

#### COM Method Prototype

N/A  
(Use the TDMA Standard Name property.)

#### C Prototype

```
ViStatus IviRFSigGen_GetTDMAStandardName (ViSession Vi,  
                                           ViInt32 Index,  
                                           ViInt32 NameBufferSize,  
                                           ViChar Name[]);
```

#### Parameters

Inputs	Description
Vi	Instrument handle
Index	A one-based index that defines which name to return.
NameBufferSize	The number of bytes in the ViChar array that the user specifies for the Name parameter.

Outputs	Description
Name	A user-allocated buffer into which the driver stores the TDMA standard name. The caller may pass VI_NULL for this parameter if the NameBufferSize parameter is 0.

#### Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

#### Compliance Notes

For an instrument with only one TDMA standard, that is the TDMA Standard Count attribute is one, the driver may return an empty string in the Name parameter.

### 29.3.2 Select TDMA Standard (IVI-C only)

#### Description

Selects the actual standard used by the instrument. The standard is selected from the list of names queried with the Get TDMA Standard Names function. The coding, mapping, symbol rate or bit clock frequency, filter together with the according filter. parameters, FSK deviation or ASK depth (in case of FSK or ASK modulation) are set as defined in the selected standard.

#### .NET Method Prototype

N/A  
(use the `DigitalModulation.Tdma.SelectedStandard` property)

#### COM Method Prototype

N/A  
(use the `DigitalModulation.TDMA.SelectedStandard` property)

#### C Prototype

```
ViStatus IviRFSigGen_SelectTDMAStandard (ViSession Vi, ViConstString Name);
```

#### Parameters

Inputs	Description
Vi	Instrument handle
Name	The driver uses this value to set the TDMA Selected Standard attribute. See the attribute description for more details.

#### Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

### 29.3.3 Configure TDMA Clock Source

#### Description

Configures the TDMA clock source.

#### .NET Method Prototype

```
void DigitalModulation.Tdma.ConfigureClockSource (String source,  
                                                  TdmaExternalClockType type);
```

#### COM Method Prototype

```
HRESULT DigitalModulation.TDMA.ConfigureClockSource (  
    [in] IviRFSigGenTDMAClockSourceEnum Source,  
    [in] IviRFSigGenTDMAExternalClockTypeEnum Type);
```

#### C Prototype

```
ViStatus IviRFSigGen_ConfigureTDMAClockSource (ViSession Vi, ViInt32 Source,  
                                              ViInt32 Type);
```

#### Parameters

Inputs	Description
Vi	Instrument handle
Source	The driver uses this value to set the TDMA Clock Source attribute. See the attribute description for more details.
Type	Type is ignored if Source is set to Internal. The driver uses this value to set the TDMA External Clock Type attribute. See the attribute description for more details.

#### Return Values (C/COM)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

#### .NET Exceptions

The *IVI-3.2: Inherent Capabilities Specification* defines general exceptions that may be thrown, and warning events that may be raised, by this method.

### 29.3.4 Configure TDMA Trigger Source (IVI-C only)

**Description**

Configures the TDMA trigger source for starting or synchronising the generation of the frames/slots.

**.NET Method Prototype**

N/A  
(use the `DigitalModulation.Tdma.TriggerSource` property)

**COM Method Prototype**

N/A  
(use the `DigitalModulation.TDMA.TriggerSource` property)

**C Prototype**

```
ViStatus IviRFSigGen_ConfigureTDMATriggerSource (ViSession Vi,  
                                                ViInt32 Source);
```

**Parameters**

Inputs	Description
Vi	Instrument handle
Source	The driver uses this value to set the TDMA Trigger Source attribute. See the attribute description for more details.

**Return Values (C)**

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

### 29.3.5 Configure TDMA External Trigger Slope (IVI-C only)

**Description**

Configures whether the trigger event occurs on the rising or falling edge of the input signal. This setting is used only if the TDMA Trigger Source attribute is set to External.

**.NET Method Prototype**

N/A  
(use the `DigitalModulation.Tdma.ExternalTriggerSlope` property)

**COM Method Prototype**

N/A  
(use the `DigitalModulation.TDMA.ExternalTriggerSlope` property)

**C Prototype**

```
ViStatus IviRFSigGen_ConfigureTDMAExternalTriggerSlope (ViSession Vi,  
                                                         ViInt32 Slope);
```

**Parameters**

Inputs	Description
Vi	Instrument handle
Slope	The driver uses this value to set the TDMA ExternalTrigger Slope attribute. See the attribute description for more details.

**Return Values (C)**

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

### 29.3.6 Get TDMA Frame Name (IVI-C only)

#### Description

This function returns the specific driver defined TDMA frame name that corresponds to the one-based index that the user specifies.

#### .NET Method Prototype

N/A  
(Use the TDMA Frame Names property.)

#### COM Method Prototype

N/A  
(Use the TDMA Frame Name property.)

#### C Prototype

```
ViStatus IviRFSigGen_GetTDMAFrameName (ViSession Vi,  
                                         ViInt32 Index,  
                                         ViInt32 NameBufferSize,  
                                         ViChar Name[]);
```

#### Parameters

Inputs	Description
Vi	Instrument handle
Index	A one-based index that defines which name to return.
NameBufferSize	The number of bytes in the ViChar array that the user specifies for the Name parameter.

Outputs	Description
Name	A user-allocated buffer into which the driver stores the TDMA frame name. The caller may pass VI_NULL for this parameter if the NameBufferSize parameter is 0.

#### Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

#### Compliance Notes

For an instrument with only one TDMA frame, that is the TDMA Frame Count attribute is one, the driver may return an empty string in the Name parameter.



### 29.3.7 Select TDMA Frame (IVI-C only)

#### Description

Selects the frame used for the framed digital modulation. The frame is selected from the list of names queried with the Get TDMA Framed Names function.

#### .NET Method Prototype

N/A  
(use the `DigitalModulation.Tdma.SelectedFrame` property)

#### COM Method Prototype

N/A  
(use the `DigitalModulation.TDMA.SelectedFrame` property)

#### C Prototype

```
ViStatus IviRFSigGen_SelectTDMAFrame (ViSession Vi, ViConstString Name);
```

#### Parameters

Inputs	Description
Vi	Instrument handle
Name	The driver uses this value to set the TDMA Selected Frame attribute. See the attribute description for more details.

#### Return Values (C)

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return.

## **29.4 *IviRFSigGenTDMABase Behavior Model***

The IviRFSigGenTDMABase Extension Group follows the behavior model of the IviRFSigGenBase capability group. The only modification to the behavior model from the IviRFSigGenBase capability group is the ability to configure IviRFSigGenTDMABase settings.

## **29.5 *IviRFSigGenTDMABase Compliance Notes***

For a specific driver to comply with the IviRFSigGenTDMABase extension, it shall also implement the IviRFSigGenModulateIQ Extension Group and support the defined value TDMABase for the IQ Source attribute.

1. If a specific driver does not support the defined value External for the TDMA Trigger Source attribute, it shall not support the TDMA External Trigger Slope attribute, the TDMA External Clock Type attribute or the Configure TDMA External Trigger Slope function.

## 30. IviRFSigGen Attribute ID Definitions

Table 30-1 defines the ID values for all IviRFSigGen class attributes.

**Table 30-1** IviRFSigGen Attributes ID Values (C only)

Attribute Name	ID Definition
IVIRFSIGGEN_ATTR_FREQUENCY	IVI_CLASS_ATTR_BASE + 1
IVIRFSIGGEN_ATTR_POWER_LEVEL	IVI_CLASS_ATTR_BASE + 2
IVIRFSIGGEN_ATTR_ALC_ENABLED	IVI_CLASS_ATTR_BASE + 3
IVIRFSIGGEN_ATTR_OUTPUT_ENABLED	IVI_CLASS_ATTR_BASE + 4
IVIRFSIGGEN_ATTR_AM_ENABLED	IVI_CLASS_ATTR_BASE + 11
IVIRFSIGGEN_ATTR_AM_SOURCE	IVI_CLASS_ATTR_BASE + 12
IVIRFSIGGEN_ATTR_AM_SCALING	IVI_CLASS_ATTR_BASE + 13
IVIRFSIGGEN_ATTR_AM_EXTERNAL_COUPLING	IVI_CLASS_ATTR_BASE + 14
IVIRFSIGGEN_ATTR_AM_NOMINAL_VOLTAGE	IVI_CLASS_ATTR_BASE + 15
IVIRFSIGGEN_ATTR_AM_DEPTH	IVI_CLASS_ATTR_BASE + 16
IVIRFSIGGEN_ATTR_FM_ENABLED	IVI_CLASS_ATTR_BASE + 21
IVIRFSIGGEN_ATTR_FM_SOURCE	IVI_CLASS_ATTR_BASE + 22
IVIRFSIGGEN_ATTR_FM_EXTERNAL_COUPLING	IVI_CLASS_ATTR_BASE + 23
IVIRFSIGGEN_ATTR_FM_NOMINAL_VOLTAGE	IVI_CLASS_ATTR_BASE + 24
IVIRFSIGGEN_ATTR_FM_DEVIATION	IVI_CLASS_ATTR_BASE + 25
IVIRFSIGGEN_ATTR_PM_ENABLED	IVI_CLASS_ATTR_BASE + 31
IVIRFSIGGEN_ATTR_PM_SOURCE	IVI_CLASS_ATTR_BASE + 32
IVIRFSIGGEN_ATTR_PM_EXTERNAL_COUPLING	IVI_CLASS_ATTR_BASE + 33
IVIRFSIGGEN_ATTR_PM_NOMINAL_VOLTAGE	IVI_CLASS_ATTR_BASE + 34
IVIRFSIGGEN_ATTR_PM_DEVIATION	IVI_CLASS_ATTR_BASE + 35
IVIRFSIGGEN_ATTR_ANALOG_MODULATION_SOURCE_COUNT	IVI_CLASS_ATTR_BASE + 41
IVIRFSIGGEN_ATTR_PULSE_MODULATION_ENABLED	IVI_CLASS_ATTR_BASE + 51
IVIRFSIGGEN_ATTR_PULSE_MODULATION_SOURCE	IVI_CLASS_ATTR_BASE + 52
IVIRFSIGGEN_ATTR_PULSE_MODULATION_EXTERNAL_POLARITY	IVI_CLASS_ATTR_BASE + 53
IVIRFSIGGEN_ATTR_ACTIVE_LF_GENERATOR	IVI_CLASS_ATTR_BASE + 101
IVIRFSIGGEN_ATTR_LF_GENERATOR_COUNT	IVI_CLASS_ATTR_BASE + 102
IVIRFSIGGEN_ATTR_LF_GENERATOR_FREQUENCY	IVI_CLASS_ATTR_BASE + 103
IVIRFSIGGEN_ATTR_LF_GENERATOR_WAVEFORM	IVI_CLASS_ATTR_BASE + 104
IVIRFSIGGEN_ATTR_LF_GENERATOR_OUTPUT_AMPLITUDE	IVI_CLASS_ATTR_BASE + 111
IVIRFSIGGEN_ATTR_LF_GENERATOR_OUTPUT_ENABLED	IVI_CLASS_ATTR_BASE + 112
IVIRFSIGGEN_ATTR_PULSE_INTERNAL_TRIGGER_PERIOD	IVI_CLASS_ATTR_BASE + 121
IVIRFSIGGEN_ATTR_PULSE_WIDTH	IVI_CLASS_ATTR_BASE + 122
IVIRFSIGGEN_ATTR_PULSE_GATING_ENABLED	IVI_CLASS_ATTR_BASE + 123

**Table 30-1** IviRFSigGen Attributes ID Values (C only)

Attribute Name	ID Definition
IVIRFSIGGEN_ATTR_PULSE_TRIGGER_SOURCE	IVI_CLASS_ATTR_BASE + 124
IVIRFSIGGEN_ATTR_PULSE_EXTERNAL_TRIGGER_SLOPE	IVI_CLASS_ATTR_BASE + 125
IVIRFSIGGEN_ATTR_PULSE_EXTERNAL_TRIGGER_DELAY	IVI_CLASS_ATTR_BASE + 126
IVIRFSIGGEN_ATTR_PULSE_DOUBLE_ENABLED	IVI_CLASS_ATTR_BASE + 131
IVIRFSIGGEN_ATTR_PULSE_DOUBLE_DELAY	IVI_CLASS_ATTR_BASE + 132
IVIRFSIGGEN_ATTR_PULSE_OUTPUT_POLARITY	IVI_CLASS_ATTR_BASE + 141
IVIRFSIGGEN_ATTR_PULSE_OUTPUT_ENABLED	IVI_CLASS_ATTR_BASE + 142
IVIRFSIGGEN_ATTR_SWEEP_MODE	IVI_CLASS_ATTR_BASE + 201
IVIRFSIGGEN_ATTR_SWEEP_TRIGGER_SOURCE	IVI_CLASS_ATTR_BASE + 202
IVIRFSIGGEN_ATTR_FREQUENCY_SWEEP_START	IVI_CLASS_ATTR_BASE + 211
IVIRFSIGGEN_ATTR_FREQUENCY_SWEEP_STOP	IVI_CLASS_ATTR_BASE + 212
IVIRFSIGGEN_ATTR_FREQUENCY_SWEEP_TIME	IVI_CLASS_ATTR_BASE + 213
IVIRFSIGGEN_ATTR_POWER_SWEEP_START	IVI_CLASS_ATTR_BASE + 221
IVIRFSIGGEN_ATTR_POWER_SWEEP_STOP	IVI_CLASS_ATTR_BASE + 222
IVIRFSIGGEN_ATTR_POWER_SWEEP_TIME	IVI_CLASS_ATTR_BASE + 223
IVIRFSIGGEN_ATTR_FREQUENCY_STEP_START	IVI_CLASS_ATTR_BASE + 241
IVIRFSIGGEN_ATTR_FREQUENCY_STEP_STOP	IVI_CLASS_ATTR_BASE + 242
IVIRFSIGGEN_ATTR_FREQUENCY_STEP_SCALING	IVI_CLASS_ATTR_BASE + 243
IVIRFSIGGEN_ATTR_FREQUENCY_STEP_SIZE	IVI_CLASS_ATTR_BASE + 244
IVIRFSIGGEN_ATTR_FREQUENCY_STEP_SINGLE_STEP_ENABLED	IVI_CLASS_ATTR_BASE + 245
IVIRFSIGGEN_ATTR_FREQUENCY_STEP_DWELL	IVI_CLASS_ATTR_BASE + 246
IVIRFSIGGEN_ATTR_POWER_STEP_START	IVI_CLASS_ATTR_BASE + 261
IVIRFSIGGEN_ATTR_POWER_STEP_STOP	IVI_CLASS_ATTR_BASE + 262
IVIRFSIGGEN_ATTR_POWER_STEP_SIZE	IVI_CLASS_ATTR_BASE + 263
IVIRFSIGGEN_ATTR_POWER_STEP_SINGLE_STEP_ENABLED	IVI_CLASS_ATTR_BASE + 264
IVIRFSIGGEN_ATTR_POWER_STEP_DWELL	IVI_CLASS_ATTR_BASE + 265
IVIRFSIGGEN_ATTR_LIST_SELECTED_NAME	IVI_CLASS_ATTR_BASE + 281
IVIRFSIGGEN_ATTR_LIST_SINGLE_STEP_ENABLED	IVI_CLASS_ATTR_BASE + 282
IVIRFSIGGEN_ATTR_LIST_DWELL	IVI_CLASS_ATTR_BASE + 283
IVIRFSIGGEN_ATTR_ALC_SOURCE	IVI_CLASS_ATTR_BASE + 301
IVIRFSIGGEN_ATTR_ALC_BANDWIDTH	IVI_CLASS_ATTR_BASE + 302
IVIRFSIGGEN_ATTR_REFERENCE_OSCILLATOR_SOURCE	IVI_CLASS_ATTR_BASE + 321
IVIRFSIGGEN_ATTR_REFERENCE_OSCILLATOR_EXTERNAL_FREQUENCY	IVI_CLASS_ATTR_BASE + 322
IVIRFSIGGEN_ATTR_IQ_ENABLED	IVI_CLASS_ATTR_BASE + 401
IVIRFSIGGEN_ATTR_IQ_NOMINAL_VOLTAGE	IVI_CLASS_ATTR_BASE + 402
IVIRFSIGGEN_ATTR_IQ_SOURCE	IVI_CLASS_ATTR_BASE + 403
IVIRFSIGGEN_ATTR_IQ_SWAP_ENABLED	IVI_CLASS_ATTR_BASE + 404

**Table 30-1** IviRFSigGen Attributes ID Values (C only)

Attribute Name	ID Definition
IVIRFSIGGEN_ATTR_IQ_IMPAIRMENT_ENABLED	IVI_CLASS_ATTR_BASE + 421
IVIRFSIGGEN_ATTR_IQ_I_OFFSET	IVI_CLASS_ATTR_BASE + 422
IVIRFSIGGEN_ATTR_IQ_Q_OFFSET	IVI_CLASS_ATTR_BASE + 423
IVIRFSIGGEN_ATTR_IQ_RATIO	IVI_CLASS_ATTR_BASE + 424
IVIRFSIGGEN_ATTR_IQ_SKEW	IVI_CLASS_ATTR_BASE + 425
IVIRFSIGGEN_ATTR_ARB_SELECTED_WAVEFORM	IVI_CLASS_ATTR_BASE + 451
IVIRFSIGGEN_ATTR_ARB_CLOCK_FREQUENCY	IVI_CLASS_ATTR_BASE + 452
IVIRFSIGGEN_ATTR_ARB_FILTER_FREQUENCY	IVI_CLASS_ATTR_BASE + 453
IVIRFSIGGEN_ATTR_ARB_MAX_NUMBER_WAVEFORMS	IVI_CLASS_ATTR_BASE + 454
IVIRFSIGGEN_ATTR_ARB_WAVEFORM_QUANTUM	IVI_CLASS_ATTR_BASE + 455
IVIRFSIGGEN_ATTR_ARB_WAVEFORM_SIZE_MIN	IVI_CLASS_ATTR_BASE + 456
IVIRFSIGGEN_ATTR_ARB_WAVEFORM_SIZE_MAX	IVI_CLASS_ATTR_BASE + 457
IVIRFSIGGEN_ATTR_ARB_TRIGGER_SOURCE	IVI_CLASS_ATTR_BASE + 458
IVIRFSIGGEN_ATTR_ARB_EXTERNAL_TRIGGER_SLOPE	IVI_CLASS_ATTR_BASE + 459
IVIRFSIGGEN_ATTR_DIGITAL_MODULATION_BASE_STANDARD_COUNT	IVI_CLASS_ATTR_BASE + 501
IVIRFSIGGEN_ATTR_DIGITAL_MODULATION_BASE_SELECTED_STANDARD	IVI_CLASS_ATTR_BASE + 502
IVIRFSIGGEN_ATTR_DIGITAL_MODULATION_BASE_DATA_SOURCE	IVI_CLASS_ATTR_BASE + 503
IVIRFSIGGEN_ATTR_DIGITAL_MODULATION_BASE_PRBS_TYPE	IVI_CLASS_ATTR_BASE + 504
IVIRFSIGGEN_ATTR_DIGITAL_MODULATION_BASE_BIT_SEQUENCE	IVI_CLASS_ATTR_BASE + 505
IVIRFSIGGEN_ATTR_DIGITAL_MODULATION_BASE_CLOCK_SOURCE	IVI_CLASS_ATTR_BASE + 506
IVIRFSIGGEN_ATTR_DIGITAL_MODULATION_BASE_EXTERNAL_CLOCK_TYPE	IVI_CLASS_ATTR_BASE + 507
IVIRFSIGGEN_ATTR_CDMA_STANDARD_COUNT	IVI_CLASS_ATTR_BASE + 601
IVIRFSIGGEN_ATTR_CDMA_SELECTED_STANDARD	IVI_CLASS_ATTR_BASE + 602
IVIRFSIGGEN_ATTR_CDMA_TRIGGER_SOURCE	IVI_CLASS_ATTR_BASE + 603
IVIRFSIGGEN_ATTR_CDMA_EXTERNAL_TRIGGER_SLOPE	IVI_CLASS_ATTR_BASE + 604
IVIRFSIGGEN_ATTR_CDMA_TEST_MODEL_COUNT	IVI_CLASS_ATTR_BASE + 605
IVIRFSIGGEN_ATTR_CDMA_SELECTED_TEST_MODEL	IVI_CLASS_ATTR_BASE + 606
IVIRFSIGGEN_ATTR_CDMA_CLOCK_SOURCE	IVI_CLASS_ATTR_BASE + 607
IVIRFSIGGEN_ATTR_TDMA_STANDARD_COUNT	IVI_CLASS_ATTR_BASE + 701
IVIRFSIGGEN_ATTR_TDMA_SELECTED_STANDARD	IVI_CLASS_ATTR_BASE + 702
IVIRFSIGGEN_ATTR_TDMA_TRIGGER_SOURCE	IVI_CLASS_ATTR_BASE + 703
IVIRFSIGGEN_ATTR_TDMA_EXTERNAL_TRIGGER_SLOPE	IVI_CLASS_ATTR_BASE + 704
IVIRFSIGGEN_ATTR_TDMA_FRAME_COUNT	IVI_CLASS_ATTR_BASE + 705
IVIRFSIGGEN_ATTR_TDMA_SELECTED_FRAME	IVI_CLASS_ATTR_BASE + 706
IVIRFSIGGEN_ATTR_TDMA_CLOCK_SOURCE	IVI_CLASS_ATTR_BASE + 707

**Table 30-1** IviRFSigGen Attributes ID Values (C only)

Attribute Name	ID Definition
IVIRFSIGGEN_ATTR_TDMA_EXTERNAL_CLOCK_TYPE	IVI_CLASS_ATTR_BASE + 708

## 31. IviRFSigGen Attribute Value Definitions

This section specifies the actual value for each defined attribute value.

### AM Scaling

<i>Value Name</i>	<i>Language</i>	<i>Identifier</i>	<i>Actual Value</i>
AM Scaling Linear	.NET	AMScaling.Linear	0
	C	IVIRFSIGGEN_VAL_AM_SCALING_LINEAR	0
	COM	IviRFSigGenAMScalingLinear	0
AM Scaling Logarithmic	.NET	AMScaling.Logarithmic	1
	C	IVIRFSIGGEN_VAL_AM_SCALING_LOGARITHMIC	1
	COM	IviRFSigGenAMScalingLogarithmic	1
AM Scaling Class Ext Base	C	IVIRFSIGGEN_VAL_AM_SCALING_CLASS_EXT_BASE	500
AM Scaling Specific Ext Base	C	IVIRFSIGGEN_VAL_AM_SCALING_SPECIFIC_EXT_BASE	1000
	COM		1000

### AM External Coupling

<i>Value Name</i>	<i>Language</i>	<i>Identifier</i>	<i>Actual Value</i>
AM External Coupling AC	.NET	ExternalCoupling.AC	0
	C	IVIRFSIGGEN_VAL_AM_EXTERNAL_COUPLING_AC	0
	COM	IviRFSigGenAMExternalCouplingAC	0
AM External Coupling DC	.NET	ExternalCoupling.DC	1
	C	IVIRFSIGGEN_VAL_AM_EXTERNAL_COUPLING_DC	1
	COM	IviRFSigGenAMExternalCouplingDC	1
AM External Coupling Class Ext Base	C	IVIRFSIGGEN_VAL_AM_EXTERNAL_COUPLING_CLASS_EXT_BASE	500
AM External Coupling Specific Ext Base	C	IVIRFSIGGEN_VAL_AM_EXTERNAL_COUPLING_SPECIFIC_EXT_BASE	1000
	COM		1000

### FM External Coupling

<i>Value Name</i>	<i>Language</i>	<i>Identifier</i>	<i>Actual Value</i>
FM External Coupling AC	.NET	ExternalCoupling.AC	0

<i>Value Name</i>	<i>Language</i>	<i>Identifier</i>	<i>Actual Value</i>
	C	IVIRFSIGGEN_VAL_FM_EXTERNAL_COUPLING_AC	0
	COM	IviRFSigGenFMExternalCouplingAC	0
FM External Coupling DC	.NET	ExternalCoupling.DC	1
	C	IVIRFSIGGEN_VAL_FM_EXTERNAL_COUPLING_DC	1
	COM	IviRFSigGenFMExternalCouplingDC	1
FM External Coupling Class Ext Base	C	IVIRFSIGGEN_VAL_FM_EXTERNAL_COUPLING_CLASS_EXT_BASE	500
FM External Coupling Specific Ext Base	C	IVIRFSIGGEN_VAL_FM_EXTERNAL_COUPLING_SPECIFIC_EXT_BASE	1000
	COM		1000

## PM External Coupling

<i>Value Name</i>	<i>Language</i>	<i>Identifier</i>	<i>Actual Value</i>
PM External Coupling AC	.NET	ExternalCoupling.AC	0
	C	IVIRFSIGGEN_VAL_PM_EXTERNAL_COUPLING_AC	0
	COM	IviRFSigGenPMExternalCouplingAC	0
PM External Coupling DC	.NET	ExternalCoupling.DC	1
	C	IVIRFSIGGEN_VAL_PM_EXTERNAL_COUPLING_DC	1
	COM	IviRFSigGenPMExternalCouplingDC	1
PM External Coupling Class Ext Base	C	IVIRFSIGGEN_VAL_PM_EXTERNAL_COUPLING_CLASS_EXT_BASE	500
PM External Coupling Specific Ext Base	C	IVIRFSIGGEN_VAL_PM_EXTERNAL_COUPLING_SPECIFIC_EXT_BASE	1000
	COM		1000

## Pulse Modulation Source

<i>Value Name</i>	<i>Language</i>	<i>Identifier</i>	<i>Actual Value</i>
Pulse Modulation Source Internal	.NET	PulseModulationSource.Internal	0
	C	IVIRFSIGGEN_VAL_PULSE_MODULATION_SOURCE_INTERNAL	0
	COM	IviRFSigGenPulseModulationSourceInternal	0
Pulse Modulation Source External	.NET	PulseModulationSource.External	1
	C	IVIRFSIGGEN_VAL_PULSE_MODULATION_SOURCE_EXTERNAL	1
	COM	IviRFSigGenPulseModulationSourceExternal	1



<i>Value Name</i>	<i>Language</i>	<i>Identifier</i>	<i>Actual Value</i>
Pulse Modulation Source Class Ext Base	C	IVIRFSIGGEN_VAL_PULSE_MODULATION_SOURCE_CLASS_EXT_BASE	500
Pulse Modulation Source Specific Ext Base	C	IVIRFSIGGEN_VAL_PULSE_MODULATION_SOURCE_SPECIFIC_EXT_BASE	1000
	COM		1000

### Pulse Modulation External Polarity

<i>Value Name</i>	<i>Language</i>	<i>Identifier</i>	<i>Actual Value</i>
Pulse Modulation External Polarity Normal	.NET	PulseModulationExternalPolarity.Normal	0
	C	IVIRFSIGGEN_VAL_PULSE_MODULATION_EXTERNAL_POLARITY_NORMAL	0
	COM	IviRFSigGenPulseModulationExternalPolarityNormal	0
Pulse Modulation External Polarity Inverse	.NET	PulseModulationExternalPolarity.Inverse	1
	C	IVIRFSIGGEN_VAL_PULSE_MODULATION_EXTERNAL_POLARITY_INVERSE	1
	COM	IviRFSigGenPulseModulationExternalPolarityInverse	1
Pulse Modulation External Polarity Class Ext Base	C	IVIRFSIGGEN_VAL_PULSE_MODULATION_EXTERNAL_POLARITY_CLASS_EXT_BASE	500
Pulse Modulation External Polarity Specific Ext Base	C	IVIRFSIGGEN_VAL_PULSE_MODULATION_EXTERNAL_POLARITY_SPECIFIC_EXT_BASE	1000
	COM		1000

### LFGenerator Waveform

<i>Value Name</i>	<i>Language</i>	<i>Identifier</i>	<i>Actual Value</i>
LFGenerator Waveform Sine	.NET	LFGeneratorWaveform.Sine	0
	C	IVIRFSIGGEN_VAL_LF_GENERATOR_WAVEFORM_SINE	0
	COM	IviRFSigGenLFGeneratorWaveformSine	0
LFGenerator Waveform Square	.NET	LFGeneratorWaveform.Square	1
	C	IVIRFSIGGEN_VAL_LF_GENERATOR_WAVEFORM_SQUARE	1
	COM	IviRFSigGenLFGeneratorWaveformSquare	1
LFGenerator Waveform Triangle	.NET	LFGeneratorWaveform.Triangle	2

<i>Value Name</i>	<i>Language</i>	<i>Identifier</i>	<i>Actual Value</i>
	C	IVIRFSIGGEN_VAL_LF_GENERATOR_WAVEFORM_TRIANGLE	2
	COM	IviRFSigGenLFGeneratorWaveformTriangle	2
LFGenerator Waveform Ramp Up	.NET	LFGeneratorWaveform.RampUp	3
	C	IVIRFSIGGEN_VAL_LF_GENERATOR_WAVEFORM_RAMP_UP	3
	COM	IviRFSigGenLFGeneratorWaveformRampUp	3
LFGenerator Waveform Ramp Down	.NET	LFGeneratorWaveform.RampDown	4
	C	IVIRFSIGGEN_VAL_LF_GENERATOR_WAVEFORM_RAMP_DOWN	4
	COM	IviRFSigGenLFGeneratorWaveformRampDown	4
LFGenerator Waveform Class Ext Base	C	IVIRFSIGGEN_VAL_LF_GENERATOR_WAVEFORM_CLASSES_EXT_BASE	500
LFGenerator Waveform Specific Ext Base	C	IVIRFSIGGEN_VAL_LF_GENERATOR_WAVEFORM_SPECIFIC_EXT_BASE	1000
	COM		1000

## Pulse Trigger Source

<i>Value Name</i>	<i>Language</i>	<i>Identifier</i>	<i>Actual Value</i>
Pulse Trigger Source Internal	C	IVIRFSIGGEN_VAL_PULSE_TRIGGER_SOURCE_INTERNAL	0
	COM	IviRFSigGenPulseTriggerSourceInternal	0
Pulse Trigger Source External	C	IVIRFSIGGEN_VAL_PULSE_TRIGGER_SOURCE_EXTERNAL	1
	COM	IviRFSigGenPulseTriggerSourceExternal	1
Pulse Trigger Source Class Ext Base	C	IVIRFSIGGEN_VAL_PULSE_TRIGGER_SOURCE_CLASSES_EXT_BASE	500
Pulse Trigger Source Specific Ext Base	C	IVIRFSIGGEN_VAL_PULSE_TRIGGER_SOURCE_SPECIFIC_EXT_BASE	1000
	COM		1000

## Pulse External Trigger Slope

<i>Value Name</i>	<i>Language</i>	<i>Identifier</i>	<i>Actual Value</i>
Pulse External Trigger Slope Positive	.NET	Slope.Positive	0
	C	IVIRFSIGGEN_VAL_PULSE_EXTERNAL_TRIGGER_SLOPE_POSITIVE	0

<i>Value Name</i>	<i>Language</i>	<i>Identifier</i>	<i>Actual Value</i>
	COM	IviRFSigGenPulseExternalTriggerSlopePositive	0
Pulse External Trigger Slope Negative	.NET	Slope.Negative	1
	C	IVIRFSIGGEN_VAL_PULSE_EXTERNAL_TRIGGER_SLOPE_NEGATIVE	1
	COM	IviRFSigGenPulseExternalTriggerSlopeNegative	1
	C	IVIRFSIGGEN_VAL_PULSE_EXTERNAL_TRIGGER_SLOPE_CLASS_EXT_BASE	500
Pulse External Trigger Slope Class Ext Base	C	IVIRFSIGGEN_VAL_PULSE_EXTERNAL_TRIGGER_SLOPE_SPECIFIC_EXT_BASE	1000
Pulse External Trigger Slope Specific Ext Base	COM		1000

## Pulse Output Polarity

<i>Value Name</i>	<i>Language</i>	<i>Identifier</i>	<i>Actual Value</i>
Pulse Output Polarity Normal	.NET	PulseOutputPolarity.Normal	0
	C	IVIRFSIGGEN_VAL_PULSE_OUTPUT_POLARITY_NORMAL	0
	COM	IviRFSigGenPulseOutputPolarityNormal	0
Pulse Output Polarity Inverse	.NET	PulseOutputPolarity.Inverse	1
	C	IVIRFSIGGEN_VAL_PULSE_OUTPUT_POLARITY_INVERSE	1
	COM	IviRFSigGenPulseOutputPolarityInverse	1
Pulse Output Polarity Class Ext Base	C	IVIRFSIGGEN_VAL_PULSE_OUTPUT_POLARITY_CLASSES_EXT_BASE	500
Pulse Output Polarity Specific Ext Base	C	IVIRFSIGGEN_VAL_PULSE_OUTPUT_POLARITY_SPECIFIC_EXT_BASE	1000
	COM		1000

## Sweep Mode

<i>Value Name</i>	<i>Language</i>	<i>Identifier</i>	<i>Actual Value</i>
Sweep Mode None	.NET	SweepMode.None	0
	C	IVIRFSIGGEN_VAL_SWEEP_MODE_NONE	0
	COM	IviRFSigGenSweepModeNone	0

<i>Value Name</i>	<i>Language</i>	<i>Identifier</i>	<i>Actual Value</i>
Sweep Mode Frequency Sweep	.NET	SweepMode.FrequencySweep	1
	C	IVIRFSIGGEN_VAL_SWEEP_MODE_FREQUENCY_SWEEP	1
	COM	IviRFSigGenSweepModeFrequencySweep	1
Sweep Mode Power Sweep	.NET	SweepMode.PowerSweep	2
	C	IVIRFSIGGEN_VAL_SWEEP_MODE_POWER_SWEEP	2
	COM	IviRFSigGenSweepModePowerSweep	2
Sweep Mode Frequency Step	.NET	SweepMode.FrequencyStep	3
	C	IVIRFSIGGEN_VAL_SWEEP_MODE_FREQUENCY_STEP	3
	COM	IviRFSigGenSweepModeFrequencyStep	3
Sweep Mode Power Step	.NET	SweepMode.PowerStep	4
	C	IVIRFSIGGEN_VAL_SWEEP_MODE_POWER_STEP	4
	COM	IviRFSigGenSweepModePowerStep	4
Sweep Mode List	.NET	SweepMode.List	5
	C	IVIRFSIGGEN_VAL_SWEEP_MODE_LIST	5
	COM	IviRFSigGenSweepModeList	5
Sweep Mode Class Ext Base	C	IVIRFSIGGEN_VAL_SWEEP_MODE_CLASS_EXT_BASE	500
Sweep Mode Specific Ext Base	C	IVIRFSIGGEN_VAL_SWEEP_MODE_SPECIFIC_EXT_BASE	1000
	COM		1000

### Sweep Trigger Source

<i>Value Name</i>	<i>Language</i>	<i>Identifier</i>	<i>Actual Value</i>
Sweep Trigger Source Immediate	C	IVIRFSIGGEN_VAL_SWEEP_TRIGGER_SOURCE_IMMEDIATE	0
	COM	IviRFSigGenSweepTriggerSourceImmediate	0
Sweep Trigger Source External	C	IVIRFSIGGEN_VAL_SWEEP_TRIGGER_SOURCE_EXTERNAL	1
	COM	IviRFSigGenSweepTriggerSourceExternal	1
Sweep Trigger Source Software	C	IVIRFSIGGEN_VAL_SWEEP_TRIGGER_SOURCE_SOFTWARE	2
	COM	IviRFSigGenSweepTriggerSourceSoftware	2
Sweep Trigger Source Class Ext Base	C	IVIRFSIGGEN_VAL_SWEEP_TRIGGER_SOURCE_CLASS_EXT_BASE	500
Sweep Trigger Source Specific Ext Base	C	IVIRFSIGGEN_VAL_SWEEP_TRIGGER_SOURCE_SPECIFIC_EXT_BASE	1000
	COM		1000

## Frequency Step Scaling

<i>Value Name</i>	<i>Language</i>	<i>Identifier</i>	<i>Actual Value</i>
Frequency Step Scaling Linear	.NET	FrequencyStepScaling.Linear	0
	C	IVIRFSIGGEN_VAL_FREQUENCY_STEP_SCALING_LINEAR	0
	COM	IviRFSigGenFrequencyStepScalingLinear	0
Frequency Step Scaling Logarithmic	.NET	FrequencyStepScaling.Logarithmic	1
	C	IVIRFSIGGEN_VAL_FREQUENCY_STEP_SCALING_LOGARITHMIC	1
	COM	IviRFSigGenFrequencyStepScalingLogarithmic	1
Frequency Step Scaling Class Ext Base	C	IVIRFSIGGEN_VAL_FREQUENCY_STEP_SCALING_CLASS_EXT_BASE	500
Frequency Step Scaling Specific Ext Base	C	IVIRFSIGGEN_VAL_FREQUENCY_STEP_SCALING_SPECIFIC_EXT_BASE	1000
	COM		1000

## ALC Source

<i>Value Name</i>	<i>Language</i>	<i>Identifier</i>	<i>Actual Value</i>
ALC Source Internal	C	IVIRFSIGGEN_VAL_ALC_SOURCE_INTERNAL	0
	COM	IviRFSigGenALCSourceInternal	0
ALC Source External	C	IVIRFSIGGEN_VAL_ALC_SOURCE_EXTERNAL	1
	COM	IviRFSigGenALCSourceExternal	1
ALC Source Class Ext Base	C	IVIRFSIGGEN_VAL_ALC_SOURCE_CLASS_EXT_BASE	500
ALC Source Specific Ext Base	C	IVIRFSIGGEN_VAL_ALC_SOURCE_SPECIFIC_EXT_BASE	1000
	COM		1000

## Reference Oscillator Source

<i>Value Name</i>	<i>Language</i>	<i>Identifier</i>	<i>Actual Value</i>
Reference Oscillator Source Internal	C	IVIRFSIGGEN_VAL_REFERENCE_OSCILLATOR_SOURCE_INTERNAL	0
	COM	IviRFSigGenReferenceOscillatorSourceInternal	0
Reference Oscillator Source External	C	IVIRFSIGGEN_VAL_REFERENCE_OSCILLATOR_SOURCE_EXTERNAL	1
	COM	IviRFSigGenReferenceOscillatorSourceExternal	1

<i>Value Name</i>	<i>Language</i>	<i>Identifier</i>	<i>Actual Value</i>
Reference Oscillator Source Class Ext Base	C	IVIRFSIGGEN_VAL_REFERENCE_OSCILLATOR_SOURCE_CLASS_EXT_BASE	500
Reference Oscillator Source Specific Ext Base	C	IVIRFSIGGEN_VAL_REFERENCE_OSCILLATOR_SOURCE_SPECIFIC_EXT_BASE	1000
	COM		1000

## IQ Source

<i>Value Name</i>	<i>Language</i>	<i>Identifier</i>	<i>Actual Value</i>
IQ Source DigitalModulation Base	.NET	IQSource.DigitalModulationBase	0
	C	IVIRFSIGGEN_VAL_IQ_SOURCE_DIGITAL_MODULATION_BASE	0
	COM	IviRFSigGenIQSourceDigitalModulationBase	0
IQ Source CDMA Base	.NET	IQSource.CdmaBase	1
	C	IVIRFSIGGEN_VAL_IQ_SOURCE_CDMA_BASE	1
	COM	IviRFSigGenIQSourceCDMABase	1
IQ Source TDMA Base	.NET	IQSource.TdmaBase	2
	C	IVIRFSIGGEN_VAL_IQ_SOURCE_TDMA_BASE	2
	COM	IviRFSigGenIQSourceTDMABase	2
IQ Source Arb Generator	.NET	IQSource.ArbGenerator	3
	C	IVIRFSIGGEN_VAL_IQ_SOURCE_ARB_GENERATOR	3
	COM	IviRFSigGenIQSourceArbGenerator	3
IQ Source External	.NET	IQSource.External	4
	C	IVIRFSIGGEN_VAL_IQ_SOURCE_EXTERNAL	4
	COM	IviRFSigGenIQSourceExternal	4
IQ Source Class Ext Base	C	IVIRFSIGGEN_VAL_IQ_SOURCE_CLASS_EXT_BASE	500
IQ Source Specific Ext Base	C	IVIRFSIGGEN_VAL_IQ_SOURCE_SPECIFIC_EXT_BASE	1000
	COM		1000

## Arb Trigger Source

<i>Value Name</i>	<i>Language</i>	<i>Identifier</i>	<i>Actual Value</i>
Arb Trigger Source Immediate	C	IVIRFSIGGEN_VAL_ARB_TRIGGER_SOURCE_IMMEDIATE	0
	COM	IviRFSigGenArbTriggerSourceImmediate	0

<i>Value Name</i>	<i>Language</i>	<i>Identifier</i>	<i>Actual Value</i>
Arb Trigger Source External	C	IVIRFSIGGEN_VAL_ARB_TRIGGER_SOURCE_EXTERNAL	1
	COM	IviRFSigGenArbTriggerSourceExternal	1
Arb Trigger Source Software	C	IVIRFSIGGEN_VAL_ARB_TRIGGER_SOURCE_SOFTWARE	2
	COM	IviRFSigGenArbTriggerSourceSoftware	2
Arb Trigger Source Class Ext Base	C	IVIRFSIGGEN_VAL_ARB_TRIGGER_SOURCE_CLASS_EXT_BASE	500
Arb Trigger Source Specific Ext Base	C	IVIRFSIGGEN_VAL_ARB_TRIGGER_SOURCE_SPECIFIC_EXT_BASE	1000
	COM		1000

### Arb External Trigger Slope

<i>Value Name</i>	<i>Language</i>	<i>Identifier</i>	<i>Actual Value</i>
Arb External Trigger Slope Positive	.NET	Slope.Positive	0
	C	IVIRFSIGGEN_VAL_ARB_EXTERNAL_TRIGGER_SLOPE_POSITIVE	0
	COM	IviRFSigGenArbExternalTriggerSlopePositive	0
Arb External Trigger Slope Negative	.NET	Slope.Negative	1
	C	IVIRFSIGGEN_VAL_ARB_EXTERNAL_TRIGGER_SLOPE_NEGATIVE	1
	COM	IviRFSigGenArbExternalTriggerSlopeNegative	1
Arb External Trigger Slope Class Ext Base	C	IVIRFSIGGEN_VAL_ARB_EXTERNAL_TRIGGER_SLOPE_CLASS_EXT_BASE	500
Arb External Trigger Slope Specific Ext Base	C	IVIRFSIGGEN_VAL_ARB_EXTERNAL_TRIGGER_SLOPE_SPECIFIC_EXT_BASE	1000
	COM		1000

### Digital Modulation Base Data Source

<i>Value Name</i>	<i>Language</i>	<i>Identifier</i>	<i>Actual Value</i>
Digital Modulation Base Data Source External	.NET	DigitalModulationBaseDataSource.External	0
	C	IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_DATA_SOURCE_EXTERNAL	0
	COM	IviRFSigGenDigitalModulationBaseDataExternal	0

<i>Value Name</i>	<i>Language</i>	<i>Identifier</i>	<i>Actual Value</i>
Digital Modulation Base Data Source PRBS	.NET	DigitalModulationBaseDataSource.Prbs	1
	C	IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_DATA_SOURCE_PRBS	1
	COM	IviRFSigGenDigitalModulationBaseDataPRBS	1
Digital Modulation Base Data Source Bit Sequence	.NET	DigitalModulationBaseDataSource.BitSequence	2
	C	IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_DATA_SOURCE_BIT_SEQUENCE	2
	COM	IviRFSigGenDigitalModulationBaseDataBitSequence	2
Digital Modulation Base Data Source Class Ext Base	C	IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_DATA_SOURCE_CLASS_EXT_BASE	500
Digital Modulation Base Data Source Specific Ext Base	C	IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_DATA_SOURCE_SPECIFIC_EXT_BASE	1000
	COM		1000

### Digital Modulation Base PRBS Type

<i>Value Name</i>	<i>Language</i>	<i>Identifier</i>	<i>Actual Value</i>
Digital Modulation Base PRBS Type PRBS9	.NET	DigitalModulationBasePrbsType.Prbs9	0
	C	IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_PRBS_TYPE_PRBS9	0
	COM	IviRFSigGenDigitalModulationBasePRBS9	0
Digital Modulation Base PRBS Type PRBS11	.NET	DigitalModulationBasePrbsType.Prbs11	1
	C	IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_PRBS_TYPE_PRBS11	1
	COM	IviRFSigGenDigitalModulationBasePRBS11	1
Digital Modulation Base PRBS Type PRBS15	.NET	DigitalModulationBasePrbsType.Prbs15	2
	C	IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_PRBS_TYPE_PRBS15	2
	COM	IviRFSigGenDigitalModulationBasePRBS15	2
Digital Modulation Base PRBS Type PRBS16	.NET	DigitalModulationBasePrbsType.Prbs16	3
	C	IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_PRBS_TYPE_PRBS16	3
	COM	IviRFSigGenDigitalModulationBasePRBS16	3



<i>Value Name</i>	<i>Language</i>	<i>Identifier</i>	<i>Actual Value</i>
Digital Modulation Base PRBS Type PRBS20	.NET	DigitalModulationBasePrbsType.Prbs20	4
	C	IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_PRBS_TYPE_PRBS20	4
	COM	IviRFSigGenDigitalModulationBasePRBS20	4
Digital Modulation Base PRBS Type PRBS21	.NET	DigitalModulationBasePrbsType.Prbs21	5
	C	IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_PRBS_TYPE_PRBS21	5
	COM	IviRFSigGenDigitalModulationBasePRBS21	5
Digital Modulation Base PRBS Type PRBS23	.NET	DigitalModulationBasePrbsType.Prbs23	6
	C	IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_PRBS_TYPE_PRBS23	6
	COM	IviRFSigGenDigitalModulationBasePRBS23	6
Digital Modulation Base PRBS Type Class Ext Base	C	IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_PRBS_TYPE_CLASS_EXT_BASE	500
Digital Modulation Base PRBS Type Specific Ext Base	C	IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_PRBS_TYPE_SPECIFIC_EXT_BASE	1000
	COM		1000

### Digital Modulation Base Clock Source

<i>Value Name</i>	<i>Language</i>	<i>Identifier</i>	<i>Actual Value</i>
Digital Modulation Base Clock Source Internal	C	IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_CLOCK_SOURCE_INTERNAL	0
	COM	IviRFSigGenDigitalModulationBaseClockSourceInternal	0
Digital Modulation Base Clock Source External	C	IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_CLOCK_SOURCE_EXTERNAL	1
	COM	IviRFSigGenDigitalModulationBaseClockSourceExternal	1
Digital Modulation Base Clock Source Class Ext Base	C	IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_CLOCK_SOURCE_CLASS_EXT_BASE	500
Digital Modulation Base Clock Source Specific Ext Base	C	IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_CLOCK_SOURCE_SPECIFIC_EXT_BASE	1000
	COM		1000

### Digital Modulation Base External Clock Type

<i>Value Name</i>	<i>Language</i>	<i>Identifier</i>	<i>Actual Value</i>
Digital Modulation Base External Clock Type Bit	.NET	DigitalModulationBaseExternalClockType.Bit	0
	C	IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_EXTERNAL_CLOCK_TYPE_BIT	0
	COM	IviRFSigGenDigitalModulationBaseExternalClockTypeBit	0
Digital Modulation Base External Clock Type Symbol	.NET	DigitalModulationBaseExternalClockType.Symbol	1
	C	IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_EXTERNAL_CLOCK_TYPE_SYMBOL	1
	COM	IviRFSigGenDigitalModulationBaseExternalClockTypeSymbol	1
Digital Modulation Base External Clock Type Class Ext Base	C	IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_EXTERNAL_CLOCK_TYPE_CLASS_EXT_BASE	500
Digital Modulation Base External Clock Type Specific Ext Base	C	IVIRFSIGGEN_VAL_DIGITAL_MODULATION_BASE_EXTERNAL_CLOCK_TYPE_SPECIFIC_EXT_BASE	1000
	COM		1000

## CDMA Trigger Source

<i>Value Name</i>	<i>Language</i>	<i>Identifier</i>	<i>Actual Value</i>
CDMA Trigger Source Immediate	C	IVIRFSIGGEN_VAL_CDMA_TRIGGER_SOURCE_IMMEDIATE	0
	COM	IviRFSigGenCDMATriggerSourceImmediate	0
CDMA Trigger Source External	C	IVIRFSIGGEN_VAL_CDMA_TRIGGER_SOURCE_EXTERNAL	1
	COM	IviRFSigGenCDMATriggerSourceExternal	1
CDMA Trigger Source Software	C	IVIRFSIGGEN_VAL_CDMA_TRIGGER_SOURCE_SOFTWARE	2
	COM	IviRFSigGenCDMATriggerSourceSoftware	2
CDMA Trigger Source Class Ext Base	C	IVIRFSIGGEN_VAL_CDMA_TRIGGER_SOURCE_CLASS_EXT_BASE	500
CDMA Trigger Source Specific Ext Base	C	IVIRFSIGGEN_VAL_CDMA_TRIGGER_SOURCE_SPECIFIC_EXT_BASE	1000
	COM		1000

## CDMA External Trigger Slope

<i>Value Name</i>	<i>Language</i>	<i>Identifier</i>	<i>Actual Value</i>
CDMA External Trigger Slope Positive	.NET	Slope.Positive	0
	C	IVIRFSIGGEN_VAL_CDMA_EXTERNAL_TRIGGER_SLOPE_POSITIVE	0
	COM	IviRFSigGenCDMAExternalTriggerSlopePositive	0
CDMA External Trigger Slope Negative	.NET	Slope.Negative	1
	C	IVIRFSIGGEN_VAL_CDMA_EXTERNAL_TRIGGER_SLOPE_NEGATIVE	1
	COM	IviRFSigGenCDMAExternalTriggerSlopeNegative	1
CDMA External Trigger Slope Class Ext Base	C	IVIRFSIGGEN_VAL_CDMA_EXTERNAL_TRIGGER_SLOPE_CLASS_EXT_BASE	500
CDMA External Trigger Slope Specific Ext Base	C	IVIRFSIGGEN_VAL_CDMA_EXTERNAL_TRIGGER_SLOPE_SPECIFIC_EXT_BASE	1000
	COM		1000

## CDMA Clock Source

<i>Value Name</i>	<i>Language</i>	<i>Identifier</i>	<i>Actual Value</i>
CDMA Clock Source Internal	C	IVIRFSIGGEN_VAL_CDMA_CLOCK_SOURCE_INTERNAL	0
	COM	IviRFSigGenCDMAClockSourceInternal	0
CDMA Clock Source External	C	IVIRFSIGGEN_VAL_CDMA_CLOCK_SOURCE_EXTERNAL	1
	COM	IviRFSigGenCDMAClockSourceExternal	1
CDMA Clock Source Class Ext Base	C	IVIRFSIGGEN_VAL_CDMA_CLOCK_SOURCE_CLASS_EXT_BASE	500
CDMA Clock Source Specific Ext Base	C	IVIRFSIGGEN_VAL_CDMA_CLOCK_SOURCE_SPECIFIC_EXT_BASE	1000
	COM		1000

## TDMA Trigger Source

<i>Value Name</i>	<i>Language</i>	<i>Identifier</i>	<i>Actual Value</i>
TDMA Trigger Source Immediate	C	IVIRFSIGGEN_VAL_TDMA_TRIGGER_SOURCE_IMMEDIATE	0
	COM	IviRFSigGenTDMATriggerSourceImmediate	0
TDMA Trigger Source External	C	IVIRFSIGGEN_VAL_TDMA_TRIGGER_SOURCE_EXTERNAL	1
	COM	IviRFSigGenTDMATriggerSourceExternal	1

<i>Value Name</i>	<i>Language</i>	<i>Identifier</i>	<i>Actual Value</i>
TDMA Trigger Source Software	C	IVIRFSIGGEN_VAL_TDMA_TRIGGER_SOURCE_SOFTWARE	2
	COM	IviRFSigGenTDMATriggerSourceSoftware	2
TDMA Trigger Source Class Ext Base	C	IVIRFSIGGEN_VAL_TDMA_TRIGGER_SOURCE_CLASS_EXT_BASE	500
TDMA Trigger Source Specific Ext Base	C	IVIRFSIGGEN_VAL_TDMA_TRIGGER_SOURCE_SPECIFIC_EXT_BASE	1000
	COM		1000

### TDMA External Trigger Slope

<i>Value Name</i>	<i>Language</i>	<i>Identifier</i>	<i>Actual Value</i>
TDMA External Trigger Slope Positive	.NET	Slope.Positive	0
	C	IVIRFSIGGEN_VAL_TDMA_EXTERNAL_TRIGGER_SLOPE_POSITIVE	0
	COM	IviRFSigGenTDMAExternalTriggerSlopePositive	0
TDMA External Trigger Slope Negative	.NET	Slope.Negative	1
	C	IVIRFSIGGEN_VAL_TDMA_EXTERNAL_TRIGGER_SLOPE_NEGATIVE	1
	COM	IviRFSigGenTDMAExternalTriggerSlopeNegative	1
TDMA External Trigger Slope Class Ext Base	C	IVIRFSIGGEN_VAL_TDMA_EXTERNAL_TRIGGER_SLOPE_CLASS_EXT_BASE	500
TDMA External Trigger Slope Specific Ext Base	C	IVIRFSIGGEN_VAL_TDMA_EXTERNAL_TRIGGER_SLOPE_SPECIFIC_EXT_BASE	1000
	COM		1000

### TDMA Clock Source

<i>Value Name</i>	<i>Language</i>	<i>Identifier</i>	<i>Actual Value</i>
TDMA Clock Source Internal	C	IVIRFSIGGEN_VAL_TDMA_CLOCK_SOURCE_INTERNAL	0
	COM	IviRFSigGenTDMAClockSourceInternal	0
TDMA Clock Source External	C	IVIRFSIGGEN_VAL_TDMA_CLOCK_SOURCE_EXTERNAL	1
	COM	IviRFSigGenTDMAClockSourceExternal	1
TDMA Clock Source Class Ext Base	C	IVIRFSIGGEN_VAL_TDMA_CLOCK_SOURCE_CLASS_EXT_BASE	500
TDMA Clock Source Specific Ext Base	C	IVIRFSIGGEN_VAL_TDMA_CLOCK_SOURCE_SPECIFIC_EXT_BASE	1000

<i>Value Name</i>	<i>Language</i>	<i>Identifier</i>	<i>Actual Value</i>
	COM		1000

## TDMA External Clock Type

<i>Value Name</i>	<i>Language</i>	<i>Identifier</i>	<i>Actual Value</i>
TDMA External Clock Type Bit	.NET	TdmaExternalClockType.Bit	0
	C	IVIRFSIGGEN_VAL_TDMA_EXTERNAL_CLOCK_TYPE_BIT	0
	COM	IviRFSigGenTDMAExternalClockTypeBit	0
TDMA External Clock Type Symbol	.NET	TdmaExternalClockType.Symbol	1
	C	IVIRFSIGGEN_VAL_TDMA_EXTERNAL_CLOCK_TYPE_SYMBOL	1
	COM	IviRFSigGenTDMAExternalClockTypeSymbol	1
TDMA External Clock Type Class Ext Base	C	IVIRFSIGGEN_VAL_TDMA_EXTERNAL_CLOCK_TYPE_CLASS_EXT_BASE	500
TDMA External Clock Type Specific Ext Base	C	IVIRFSIGGEN_VAL_TDMA_EXTERNAL_CLOCK_TYPE_SPECIFIC_EXT_BASE	1000
	COM		1000

## 32. IviRFSigGen Function Parameter Value Definitions

This section specifies the actual values for each function parameter that defines values.

### Wait Until Settled

**Parameter:** MaxTimeMilleseconds

<i>Value Name</i>	<i>Language</i>	<i>Identifier</i>	<i>Actual Value</i>
Max Time Immediate	C	IVIRFSIGGEN_VAL_MAX_TIME_IMMEDIATE	0x0
	COM	IviRFSigGenMaxTimeImmediate	0x0
Max Time Infinite	C	IVIRFSIGGEN_VAL_MAX_TIME_INFINITE	0xFFFFFFFFFUL
	COM	IviRFSigGenMaxTimeInfinite	0xFFFFFFFFFUL

**Parameter:** MaximumTime

<i>Value Name</i>	<i>Language</i>	<i>Identifier</i>	<i>Actual Value</i>
Max Time Immediate	.NET	PrecisionTimeSpan.Zero	PrecisionTimeSpan.Zero
Max Time Infinite	.NET	PrecisionTimeSpan.MaxValue	PrecisionTimeSpan.MaxValue

### 33. IviRFSigGen Error and Completion Code Value Definitions

Table 33-1 specifies the actual value for each status code that the IviRFSigGen class specification defines.

**Table 33-1** IviRFSigGen Completion Codes

<i>Error Name</i>	<i>Description</i>		
	<i>Language</i>	<i>Identifier</i>	<i>Value(hex)</i>
Trigger Not Software	The trigger source is not set to software trigger.		
	.NET	Ivi.Driver.TriggerNotSoftwareException	IVI Defined Exception (See IVI-3.2)
	C	IVIRFSIGGEN_ERROR_TRIGGER_NOT_SOFTWARE	0xBFFA1001
	COM	E_IVIRFSIGGEN_TRIGGER_NOT_SOFTWARE	0x80041001
Max Time Exceeded	Maximum time exceeded before the operation completed.		
	.NET	Ivi.Driver.MaxTimeExceededException	IVI Defined Exception (See IVI-3.2)
	C	IVIRFSIGGEN_ERROR_MAX_TIME_EXCEEDED	0xBFFA2003
	COM	E_IVIRFSIGGEN_MAX_TIME_EXCEEDED	0x80042003
List Unknown	The selected list is not defined.		
	.NET	ListUnknownException	
	C	IVIRFSIGGEN_ERROR_LIST_UNKNOWN	0xBFFA200A
	COM	E_IVIRFSIGGEN_LIST_UNKNOWN	0x8004200A

Table 33-2 defines the recommended format of the message string associated with the errors. In C, these strings are returned by the Get Error function. In COM, these strings are the description contained in the ErrorInfo object.

**Note:** In the description string table entries listed below, %s is always used to represent the component name.

**Table 33-2.** IviRFSigGen Error Message Strings

<b>Name</b>	<b>Message String</b>
Trigger Not Software	“%s: Trigger source is not set to software trigger”
Max Time Exceeded	“%s: Max Time Exceeded”
List Unknown	“%s: List Unknown”

### 33.1 Ivi.NET IviRFSigGen Exceptions and Warnings

This section defines the list of Ivi.NET exceptions and warnings that are specific to the IviRFSigGen class. For general information on Ivi.NET exceptions and warnings, refer to *IVI-3.1: Driver Architecture Specification* and section 12, *Common Ivi.NET Exceptions and Warnings*, of *IVI-3.2: Inherent Capabilities Specification*.

The Ivi.NET exceptions defined in this specification are declared in the Ivi.RFSigGen namespace.

- ListUnknownException

#### 33.1.1 ListUnknownException

##### Description

This exception is used when the driver finds that a specified list is not defined.

##### Constructors

```
Ivi.RFSigGen.FrequencyListUnknownException(String message,
                                           String listName);

Ivi.RFSigGen.FrequencyListUnknownException();

Ivi.RFSigGen.FrequencyListUnknownException(String message);

Ivi.RFSigGen.FrequencyListUnknownException(String message,
                                           System.Exception innerException);
```

##### Message String

The selected list is not defined.  
List name: <listName>

##### Parameters

Inputs	Description	Base Type
listName	The name of the list that is undefined.	String

##### Usage

If driver developers use constructors that take a message string, they are responsible for message string localization.



## 34. IviRFSigGen Hierarchies

### 34.1 IviRFSigGen .NET Hierarchy

**Table 34-1.** IviRFSigGen .NET Hierarchy

.NET Interface Hierarchy	Generic Name	Type
SendSoftwareTrigger	Send Software Trigger	M
<b>RF</b>		
Configure	Configure RF	M
IsSettled	Is Settled	M
WaitUntilSettled	Wait Until Settled	M
DisableAllModulation	Disable All Modulation	M
Frequency	Frequency	P
Level	Power Level	P
OutputEnabled	Output Enabled	P
<b>AnalogModulation</b>		
<b>Source</b>		
GetName	Analog Modulation Source Name	M
Count	Analog Modulation Source Count	P
<b>AM</b>		
Configure	Configure AM	M
Enabled	AM Enabled	P
Source	AM Source	P
Scaling	AM Scaling	P
ExternalCoupling	AM External Coupling	P
NominalVoltage	AM Nominal Voltage	P
Depth	AM Depth	P
<b>FM</b>		
Configure	Configure FM	M
Enabled	FM Enabled	P
Source	FM Source	P
ExternalCoupling	FM External Coupling	P
NominalVoltage	FM Nominal Voltage	P
Deviation	FM Deviation	P

**Table 34-1. IviRFSigGen .NET Hierarchy**

<b>.NET Interface Hierarchy</b>	<b>Generic Name</b>	<b>Type</b>
<b>PM</b>		
Configure	Configure PM	M
Enabled	PM Enabled	P
Source	PM Source	P
ExternalCoupling	PM External Coupling	P
NominalVoltage	PM Nominal Voltage	P
Deviation	PM Deviation	P
<b>PulseModulation</b>		
Enabled	Pulse Modulation Enabled	P
Source	Pulse Modulation Source	P
ExternalPolarity	Pulse Modulation External Polarity	P
<b>LFGenerator</b>		
Configure	Configure LF Generator	M
ActiveLFGenerator	Active LFGenerator	P
GetName	LFGenerator Name	M
Count	LFGenerator Count	P
Frequency	LFGenerator Frequency	P
Waveform	LFGenerator Waveform	P
<b>Output</b>		
Configure	Configure LFGenerator Output	M
Enabled	LFGenerator Output Enabled	P
Amplitude	LFGenerator Output Amplitude	P
<b>PulseGenerator</b>		
Configure	Configure Pulse	M
ConfigureExternalTrigger	Configure Pulse External Trigger	M
InternalTriggerPeriod	Pulse Internal Trigger Period	P
Width	Pulse Width	P
GatingEnabled	Pulse Gating Enabled	P
TriggerSource	Pulse Trigger Source	P
ExternalTriggerSlope	Pulse External Trigger Slope	P
ExternalTriggerDelay	Pulse External Trigger Delay	P
<b>DoublePulse</b>		
Configure	Configure Pulse Double	M
Enabled	Pulse Double Enabled	P
Delay	Pulse Double Delay	P

**Table 34-1. IviRFSigGen .NET Hierarchy**

.NET Interface Hierarchy	Generic Name	Type
<b>Output</b>		
Configure	Configure Pulse Output	M
Polarity	Pulse Output Polarity	P
Enabled	Pulse Output Enabled	P
<b>Sweep</b>		
Configure	Configure Sweep	M
Mode	Sweep Mode	P
TriggerSource	Sweep Trigger Source	P
<b>FrequencySweep</b>		
ConfigureStartStop	Configure Frequency Sweep Start Stop	M
ConfigureCenterSpan	Configure Frequency Sweep Center Span	M
Start	Frequency Sweep Start	P
Stop	Frequency Sweep Stop	P
Time	Frequency Sweep Time	P
<b>PowerSweep</b>		
ConfigureStartStop	Configure Power Sweep Start Stop	M
Start	Power Sweep Start	P
Stop	Power Sweep Stop	P
Time	Power Sweep Time	P
<b>FrequencyStep</b>		
ConfigureStartStop	Configure Frequency Step Start Stop	M
ConfigureDwell	Configure Frequency Step Dwell	M
Reset	Frequency Step Reset	M
Start	Frequency Step Start	P
Stop	Frequency Step Stop	P
Scaling	Frequency Step Scaling	P
Size	Frequency Step Size	P
SingleStepEnabled	Frequency Step Single Step Enabled	P
Dwell	Frequency Step Swell	P
<b>PowerStep</b>		
ConfigureStartStop	Configure Power Step Start Stop	M
ConfigureDwell	Configure Power Step Dwell	M
Reset	Power Step Reset	M
Start	Power Step Start	P
Stop	Power Step Stop	P
Size	Power Step Size	P
SingleStepEnabled	Power Step Single Step Enabled	P
Dwell	Power Step Dwell	P

**Table 34-1. IviRFSigGen .NET Hierarchy**

<b>.NET Interface Hierarchy</b>	<b>Generic Name</b>	<b>Type</b>
<b>List</b>		
CreateFrequency	Create Frequency List	M
CreatePower	Create Power List	M
CreateFrequencyPower	Create Frequency Power List	M
ClearAll	Clear All List	M
ConfigureDwell	Configure List Dwell	M
Reset	List Reset	M
SelectedName	List Selected Name	P
SingleStepEnabled	List Single Step Enabled	P
Dwell	List Dwell	P
<b>Alc</b>		
Configure	Configure ALC	M
Enabled	ALC Enabled	P
Source	ALC Source	P
Bandwidth	ALC Bandwidth	P
<b>ReferenceOscillator</b>		
Configure	Configure Reference Oscillator	M
Source	Reference Oscillator Source	P
ExternalFrequency	Reference Oscillator External Frequency	P
<b>IQ</b>		
Configure	Configure IQ	M
Calibrate	Calibrate IQ	M
Enabled	IQ Enabled	P
Source	IQ Source	P
NominalVoltage	IQ Nominal Voltage	P
SwapEnabled	IQ Swap Enabled	P
<b>Impairment</b>		
Configure	Configure IQ Impairment	M
Enabled	IQ Impairment Enabled	P
IOffset	IQ I-Offset	P
QOffset	IQ Q-Offset	P
Ratio	IQ Ratio	P
Skew	IQ Skew	P

**Table 34-1. IviRFSigGen .NET Hierarchy**

.NET Interface Hierarchy	Generic Name	Type
<b>DigitalModulation</b>		
<b>Arb</b>		
Configure	Configure Arb	M
WriteWaveform	Write Arb Waveform	M
ClearAllWaveforms	Clear All Arb Waveforms	M
SelectedWaveform	Arb Selected Waveform	P
ClockFrequency	Arb Clock Frequency	P
FilterFrequency	Arb Filter Frequency	P
MaxNumberWaveforms	Arb Max Number Waveforms	P
WaveformQuantum	Arb Waveform Quantum	P
WaveformSizeMin	Arb Waveform Size Min	P
WaveformSizeMax	Arb Waveform Size Max	P
TriggerSource	Arb Trigger Source	P
ExternalTriggerSlope	Arb External Trigger Slope	P
<b>Base</b>		
StandardNames	DigitalModulationBase Standard Names	M
StandardCount	DigitalModulationBase Standard Count	P
ConfigureClockSource	Configure DigitalModulationBase Clock Source	M
WriteBitSequence	Write DigitalModulationBase Bit Sequence	M
ClearAllBitSequences	Clear DigitalModulationBase Bit Sequence	M
SelectedStandard	DigitalModulationBase Selected Standard	P
DataSource	DigitalModulationBase Data Source	P
PrbsType	DigitalModulationBase PRBS TYPe	P
SelectedBitSequence	DigitalModulationBase Selected Bit Sequence	P
ClockSource	DigitalModulationBase Clock Source	P
ExternalClockType	DigitalModulationBase External Clock Type	P
<b>Cdma</b>		
StandardNames	CDMA Standard Names	M
StandardCount	CDMA Standard Count	P
TestModelNames	CDMA Test Model Names	M
TestModelCount	CDMA Test Model Count	P
SelectedStandard	CDMA Selected Standard	P
TriggerSource	CDMA Trigger Source	P
ExternalTriggerSlope	CDMA External Trigger Slope	P
SelectedTestModel	CDMA Selected Test Model	P
ClockSource	CDMA Clock Source	P
<b>Tdma</b>		

**Table 34-1. IviRFSigGen .NET Hierarchy**

<b>.NET Interface Hierarchy</b>	<b>Generic Name</b>	<b>Type</b>
StandardNames	TDMA Standard Names	M
StandardCount	TDMA Standard Count	P
ConfigureClockSource	Configure TDMA Clock Source	M
FrameNames	TDMA Frame Names	M
FrameCount	TDMA Frame Count	P
SelectedStandard	TDMA Selected Standard	P
TriggerSource	TDMA Trigger Source	P
ExternalTriggerSlope	TDMA External Trigger Slope	P
SelectedFrame	TDMA Selected Frame	P
ClockSource	TDMA Clock Source	P
ExternalClockType	TDMA External Clock Type	P

### 34.1.1 IviRFSigGen COM Interfaces

In addition to implementing IVI inherent capabilities interfaces, IviRFSigGen interfaces contain interface reference properties for accessing the following IviRFSigGen interfaces:

- IiviRFSigGenRF
- IiviRFSigGenAnalogModulation
- IiviRFSigGenPulseModulation
- IiviRFSigGenLFGenerator
- IiviRFSigGenPulseGenerator
- IiviRFSigGenSweep
- IiviRFSigGenAfc
- IiviRFSigGenReferenceOscillator
- IiviRFSigGenIQ
- IiviRFSigGenDigitalModulation

The IiviRFSigGenAnalogModulation interface contains interface reference properties for accessing additional the following IviRFSigGen AnalogModulation interfaces:

- IiviRFSigGenAnalogModulationSource
- IiviRFSigGenAM
- IiviRFSigGenFM
- IiviRFSigGenPM

The IiviRFSigGenLFGenerator interface contains interface reference properties for accessing additional the following IviRFSigGen LFGenerator interfaces:

- IiviRFSigGenLFGeneratorOutput

The IiviRFSigGenPulseGenerator interface contains interface reference properties for accessing additional the following IviRFSigGen PulseGenerator interfaces:

- IiviRFSigGenPulseGeneratorDouble
- IiviRFSigGenPulseGeneratorOutput

The IiviRFSigGenSweep interface contains interface reference properties for accessing additional the following IviRFSigGen Sweep interfaces:

- IiviRFSigGenFrequencySweep
- IiviRFSigGenPowerSweep
- IiviRFSigGenFrequencyStep
- IiviRFSigGenPowerStep
- IiviRFSigGenList

The IiviRFSigGenIQ interface contains interface reference properties for accessing additional the following IviRFSigGen IQ interfaces:

- IiviRFSigGenIQImpairment

The IiviRFSigGenDigitalModulation interface contains interface reference properties for accessing additional the following IviRFSigGen DigitalModulation interfaces:

- IiviRFSigGenArb
- IiviRFSigGenDigitalModulationBase
- IiviRFSigGenCdma
- IiviRFSigGenTdma

### 34.1.2 .NET Interface Reference Properties

Interface reference properties are used to navigate the IviRFSigGen .NET hierarchy. This section describes the interface reference properties that the IviRFSigGen interfaces define. All interface reference properties are read-only.

**Table 34-2.** IviRFSigGen Interface Reference Properties

Interface	Reference Property Name
IviRFSigGenRF	RF
IviRFSigGenAlc	Alc
IviRFSigGenReferenceOscillator	ReferenceOscillator
IviRFSigGenLFGenerator	LFGenerator
IviRFSigGenLFGeneratorOutput	LFGenerator.Output
IviRFSigGenPulseGenerator	PulseGenerator
IviRFSigGenPulseGeneratorDouble	PulseGenerator.DoublePulse
IviRFSigGenPulseGeneratorOutput	PulseGenerator.Output
IviRFSigGenAnalogModulation	AnalogModulation
IviRFSigGenAnalogModulationSource	AnalogModulation.Source
IviRFSigGenAM	AnalogModulation.AM
IviRFSigGenFM	AnalogModulation.FM
IviRFSigGenPM	AnalogModulation.PM
IviRFSigGenPulseModulation	PulseModulation
IviRFSigGenSweep	Sweep
IviRFSigGenFrequencySweep	Sweep.FrequencySweep
IviRFSigGenPowerSweep	Sweep.PowerSweep
IviRFSigGenFrequencyStep	Sweep.FrequencyStep
IviRFSigGenPowerStep	Sweep.PowerStep
IviRFSigGenList	Sweep.List
IviRFSigGenIQ	IQ
IviRFSigGenIQImpairment	IQ.Impairment
IviRFSigGenDigitalModulation	DigitalModulation
IviRFSigGenArb	DigitalModulation.Arb
IviRFSigGenDigitalModulationBase	DigitalModulation.Base
IviRFSigGenCdma	DigitalModulation.Cdma
IviRFSigGenTdma	DigitalModulation.Tdma



## 34.2 IviRFSigGen COM Hierarchy

**Table 34-3.** IviRFSigGen COM Hierarchy

COM Interface Hierarchy	Generic Name	Type
SendSoftwareTrigger	Send Software Trigger	M
<b>RF</b>		
Configure	Configure RF	M
IsSettled	Is Settled	M
WaitUntilSettled	Wait Until Settled	M
DisableAllModulation	Disable All Modulation	M
Frequency	Frequency	P
Level	Power Level	P
OutputEnabled	Output Enabled	P
<b>AnalogModulation</b>		
<b>Source</b>		
Name	Analog Modulation Source Name	P
Count	Analog Modulation Source Count	P
<b>AM</b>		
Configure	Configure AM	M
Enabled	AM Enabled	P
Source	AM Source	P
Scaling	AM Scaling	P
ExternalCoupling	AM External Coupling	P
NominalVoltage	AM Nominal Voltage	P
Depth	AM Depth	P
<b>FM</b>		
Configure	Configure FM	M
Enabled	FM Enabled	P
Source	FM Source	P
ExternalCoupling	FM External Coupling	P
NominalVoltage	FM Nominal Voltage	P
Deviation	FM Deviation	P

**Table 34-3. IviRFSigGen COM Hierarchy**

COM Interface Hierarchy	Generic Name	Type
<b>PM</b>		
Configure	Configure PM	M
Enabled	PM Enabled	P
Source	PM Source	P
ExternalCoupling	PM External Coupling	P
NominalVoltage	PM Nominal Voltage	P
Deviation	PM Deviation	P
<b>PulseModulation</b>		
Enabled	Pulse Modulation Enabled	P
Source	Pulse Modulation Source	P
ExternalPolarity	Pulse Modulation External Polarity	P
<b>LFGenerator</b>		
Configure	Configure LF Generator	M
ActiveLFGenerator	Active LFGenerator	P
Name	LFGenerator Name	P
Count	LFGenerator Count	P
Frequency	LFGenerator Frequency	P
Waveform	LFGenerator Waveform	P
<b>Output</b>		
Configure	Configure LFGenerator Output	M
Enabled	LFGenerator Output Enabled	P
Amplitude	LFGenerator Output Amplitude	P
<b>PulseGenerator</b>		
Configure	Configure Pulse	M
ConfigureExternalTrigger	Configure Pulse External Trigger	M
InternalTriggerPeriod	Pulse Internal Trigger Period	P
Width	Pulse Width	P
GatingEnabled	Pulse Gating Enabled	P
TriggerSource	Pulse Trigger Source	P
ExternalTriggerSlope	Pulse External Trigger Slope	P
ExternalTriggerDelay	Pulse External Trigger Delay	P
<b>Double</b>		
Configure	Configure Pulse Double	M
Enabled	Pulse Double Enabled	P
Delay	Pulse Double Delay	P

**Table 34-3. IviRFSigGen COM Hierarchy**

COM Interface Hierarchy	Generic Name	Type
<b>Output</b>		
Configure	Configure Pulse Output	M
Polarity	Pulse Output Polarity	P
Enabled	Pulse Output Enabled	P
<b>Sweep</b>		
Configure	Configure Sweep	M
Mode	Sweep Mode	P
TriggerSource	Sweep Trigger Source	P
<b>FrequencySweep</b>		
ConfigureStartStop	Configure Frequency Sweep Start Stop	M
ConfigureCenterSpan	Configure Frequency Sweep Center Span	M
Start	Frequency Sweep Start	P
Stop	Frequency Sweep Stop	P
Time	Frequency Sweep Time	P
<b>PowerSweep</b>		
ConfigureStartStop	Configure Power Sweep Start Stop	M
Start	Power Sweep Start	P
Stop	Power Sweep Stop	P
Time	Power Sweep Time	P
<b>FrequencyStep</b>		
ConfigureStartStop	Configure Frequency Step Start Stop	M
ConfigureDwell	Configure Frequency Step Dwell	M
Reset	Frequency Step Reset	M
Start	Frequency Step Start	P
Stop	Frequency Step Stop	P
Scaling	Frequency Step Scaling	P
Size	Frequency Step Size	P
SingleStepEnabled	Frequency Step Single Step Enabled	P
Dwell	Frequency Step Swell	P
<b>PowerStep</b>		
ConfigureStartStop	Configure Power Step Start Stop	M
ConfigureDwell	Configure Power Step Dwell	M
Reset	Power Step Reset	M
Start	Power Step Start	P
Stop	Power Step Stop	P
Size	Power Step Size	P
SingleStepEnabled	Power Step Single Step Enabled	P
Dwell	Power Step Dwell	P

**Table 34-3. IviRFSigGen COM Hierarchy**

COM Interface Hierarchy	Generic Name	Type
<b>List</b>		
CreateFrequency	Create Frequency List	M
CreatePower	Create Power List	M
CreateFrequencyPower	Create Frequency Power List	M
ClearAll	Clear All List	M
ConfigureDwell	Configure List Dwell	M
Reset	List Reset	M
SelectedName	List Selected Name	P
SingleStepEnabled	List Single Step Enabled	P
Dwell	List Dwell	P
<b>ALC</b>		
Configure	Configure ALC	M
Enabled	ALC Enabled	P
Source	ALC Source	P
Bandwidth	ALC Bandwidth	P
<b>ReferenceOscillator</b>		
Configure	Configure Reference Oscillator	M
Source	Reference Oscillator Source	P
ExternalFrequency	Reference Oscillator External Frequency	P
<b>IQ</b>		
Configure	Configure IQ	M
Calibrate	Calibrate IQ	M
Enabled	IQ Enabled	P
Source	IQ Source	P
NominalVoltage	IQ Nominal Voltage	P
SwapEnabled	IQ Swap Enabled	P
<b>Impairment</b>		
Configure	Configure IQ Impairment	M
Enabled	IQ Impairment Enabled	P
IOffset	IQ I-Offset	P
QOffset	IQ Q-Offset	P
Ratio	IQ Ratio	P
Skew	IQ Skew	P

**Table 34-3. IviRFSigGen COM Hierarchy**

COM Interface Hierarchy	Generic Name	Type
<b>DigitalModulation</b>		
<b>Arb</b>		
Configure	Configure Arb	M
WriteWaveform	Write Arb Waveform	M
ClearAllWaveforms	Clear All Arb Waveforms	M
QueryWaveformCapabilities	Query Arb Waveform Capabilities	M
SelectedWaveform	Arb Selected Waveform	P
ClockFrequency	Arb Clock Frequency	P
FilterFrequency	Arb Filter Frequency	P
MaxNumberWaveforms	Arb Max Number Waveforms	P
WaveformQuantum	Arb Waveform Quantum	P
WaveformSizeMin	Arb Waveform Size Min	P
WaveformSizeMax	Arb Waveform Size Max	P
TriggerSource	Arb Trigger Source	P
ExternalTriggerSlope	Arb External Trigger Slope	P
<b>Base</b>		
StandardName	DigitalModulationBase Standard Name	P
StandardCount	DigitalModulationBase Standard Count	P
ConfigureClockSource	Configure DigitalModulationBase Clock Source	M
WriteBitSequence	Write DigitalModulationBase Bit Sequence	M
ClearAllBitSequences	Clear DigitalModulationBase Bit Sequence	M
SelectedStandard	DigitalModulationBase Selected Standard	P
DataSource	DigitalModulationBase Data Source	P
PRBSType	DigitalModulationBase PRBS TYPe	P
SelectedBitSequence	DigitalModulationBase Selected Bit Sequence	P
ClockSource	DigitalModulationBase Clock Source	P
ExternalClockType	DigitalModulationBase External Clock Type	P
<b>CDMA</b>		
StandardName	CDMA Standard Name	P
StandardCount	CDMA Standard Count	P
TestModelName	CDMA Test Model Name	P
TestModelCount	CDMA Test Model Count	P
SelectedStandard	CDMA Selected Standard	P
TriggerSource	CDMA Trigger Source	P
ExternalTriggerSlope	CDMA External Trigger Slope	P
SelectedTestModel	CDMA Selected Test Model	P
ClockSource	CDMA Clock Source	P

**Table 34-3. IviRFSigGen COM Hierarchy**

COM Interface Hierarchy	Generic Name	Type
<b>TDMA</b>		
StandardName	TDMA Standard Name	P
StandardCount	TDMA Standard Count	P
ConfigureClockSource	Configure TDMA Clock Source	M
FrameName	TDMA Frame Name	P
FrameCount	TDMA Frame Count	P
SelectedStandard	TDMA Selected Standard	P
TriggerSource	TDMA Trigger Source	P
ExternalTriggerSlope	TDMA External Trigger Slope	P
SelectedFrame	TDMA Selected Frame	P
ClockSource	TDMA Clock Source	P
ExternalClockType	TDMA External Clock Type	P

### 34.2.1 IviRFSigGen COM Interfaces

In addition to implementing IVI inherent capabilities interfaces, IviRFSigGen interfaces contain interface reference properties for accessing the following IviRFSigGen interfaces:

- IiviRFSigGenRF
- IiviRFSigGenAnalogModulation
- IiviRFSigGenPulseModulation
- IiviRFSigGenLFGenerator
- IiviRFSigGenPulseGenerator
- IiviRFSigGenSweep
- IiviRFSigGenALC
- IiviRFSigGenReferenceOscillator
- IiviRFSigGenIQ
- IiviRFSigGenDigitalModulation

The IiviRFSigGenAnalogModulation interface contains interface reference properties for accessing additional the following IviRFSigGen AnalogModulation interfaces:

- IiviRFSigGenAnalogModulationSource
- IiviRFSigGenAM
- IiviRFSigGenFM
- IiviRFSigGenPM

The IiviRFSigGenLFGenerator interface contains interface reference properties for accessing additional the following IviRFSigGen LFGenerator interfaces:

- IiviRFSigGenLFGeneratorOutput

The IiviRFSigGenPulseGenerator interface contains interface reference properties for accessing additional the following IviRFSigGen PulseGenerator interfaces:

- IiviRFSigGenPulseGeneratorDouble
- IiviRFSigGenPulseGeneratorOutput

The IiviRFSigGenSweep interface contains interface reference properties for accessing additional the following IviRFSigGen Sweep interfaces:

- IiviRFSigGenFrequencySweep
- IiviRFSigGenPowerSweep
- IiviRFSigGenFrequencyStep
- IiviRFSigGenPowerStep
- IiviRFSigGenList

The IiviRFSigGenIQ interface contains interface reference properties for accessing additional the following IviRFSigGen IQ interfaces:

- IiviRFSigGenIQImpairment

The IiviRFSigGenDigitalModulation interface contains interface reference properties for accessing additional the following IviRFSigGen DigitalModulation interfaces:

- IiviRFSigGenArb
- IiviRFSigGenDigitalModulationBase
- IiviRFSigGenCDMA
- IiviRFSigGenTDMA

Table 34-4. IviRFSigGen Interface GUIDs lists the interfaces that this specification defines and their GUIDs.

**Table 34-4.** IviRFSigGen Interface GUIDs

Interface	GUID
IviRFSigGen	{47ed52e2-a398-11d4-ba58-000064657374}
IviRFSigGenRF	{47ed52e3-a398-11d4-ba58-000064657374}
IviRFSigGenALC	{47ed52e4-a398-11d4-ba58-000064657374}
IviRFSigGenReferenceOscillator	{47ed52e5-a398-11d4-ba58-000064657374}
IviRFSigGenLFGenerator	{47ed52e6-a398-11d4-ba58-000064657374}
IviRFSigGenLFGeneratorOutput	{47ed52e7-a398-11d4-ba58-000064657374}
IviRFSigGenPulseGenerator	{47ed52e8-a398-11d4-ba58-000064657374}
IviRFSigGenPulseGeneratorDouble	{47ed52e9-a398-11d4-ba58-000064657374}
IviRFSigGenPulseGeneratorOutput	{47ed52ea-a398-11d4-ba58-000064657374}
IviRFSigGenAnalogModulation	{47ed52eb-a398-11d4-ba58-000064657374}
IviRFSigGenAnalogModulationSource	{47ed52ec-a398-11d4-ba58-000064657374}
IviRFSigGenAM	{47ed52ed-a398-11d4-ba58-000064657374}
IviRFSigGenFM	{47ed52ee-a398-11d4-ba58-000064657374}
IviRFSigGenPM	{47ed52ef-a398-11d4-ba58-000064657374}
IviRFSigGenPulseModulation	{47ed52f0-a398-11d4-ba58-000064657374}
IviRFSigGenSweep	{47ed52f1-a398-11d4-ba58-000064657374}
IviRFSigGenFrequencySweep	{47ed52f2-a398-11d4-ba58-000064657374}
IviRFSigGenPowerSweep	{47ed52f3-a398-11d4-ba58-000064657374}
IviRFSigGenFrequencyStep	{47ed52f4-a398-11d4-ba58-000064657374}
IviRFSigGenPowerStep	{47ed52f5-a398-11d4-ba58-000064657374}
IviRFSigGenList	{47ed52f6-a398-11d4-ba58-000064657374}
IviRFSigGenIQ	{47ed52f8-a398-11d4-ba58-000064657374}
IviRFSigGenIQImpairment	{47ed52f9-a398-11d4-ba58-000064657374}
IviRFSigGenDigitalModulation	{47ed52f7-a398-11d4-ba58-000064657374}
IviRFSigGenArb	{47ed52fa-a398-11d4-ba58-000064657374}
IviRFSigGenDigitalModulationBase	{47ed52fb-a398-11d4-ba58-000064657374}
IviRFSigGenCDMA	{47ed52fc-a398-11d4-ba58-000064657374}
IviRFSigGenTDMA	{47ed52fd-a398-11d4-ba58-000064657374}



### 34.2.2 COM Interface Reference Properties

Interface reference properties are used to navigate the IviRFSigGen .NET hierarchy. This section describes the interface reference properties that the IviRFSigGen interfaces define. All interface reference properties are read-only.

**Table 34-5.** IviRFSigGen Interface Reference Properties

Interface	Reference Property Name
IviRFSigGenRF	RF
IviRFSigGenALC	ALC
IviRFSigGenReferenceOscillator	ReferenceOscillator
IviRFSigGenLFGenerator	LFGenerator
IviRFSigGenLFGeneratorOutput	LFGenerator.Output
IviRFSigGenPulseGenerator	PulseGenerator
IviRFSigGenPulseGeneratorDouble	PulseGenerator.Double
IviRFSigGenPulseGeneratorOutput	PulseGenerator.Output
IviRFSigGenAnalogModulation	AnalogModulation
IviRFSigGenAnalogModulationSource	AnalogModulation.Source
IviRFSigGenAM	AnalogModulation.AM
IviRFSigGenFM	AnalogModulation.FM
IviRFSigGenPM	AnalogModulation.PM
IviRFSigGenPulseModulation	PulseModulation
IviRFSigGenSweep	Sweep
IviRFSigGenFrequencySweep	Sweep.FrequencySweep
IviRFSigGenPowerSweep	Sweep.PowerSweep
IviRFSigGenFrequencyStep	Sweep.FrequencyStep
IviRFSigGenPowerStep	Sweep.PowerStep
IviRFSigGenList	Sweep.List
IviRFSigGenIQ	IQ
IviRFSigGenIQImpairment	IQ.Impairment
IviRFSigGenDigitalModulation	DigitalModulation
IviRFSigGenArb	DigitalModulation.Arb
IviRFSigGenDigitalModulationBase	DigitalModulation.Base
IviRFSigGenCDMA	DigitalModulation.CDMA
IviRFSigGenTDMA	DigitalModulation.TDMA

### 34.2.3 IviRFSigGen COM Category

The IviRFSigGen class COM Category shall be “IviRFSigGen”, and the Category ID (CATID) shall be {47ed5159-a398-11d4-ba58-000064657374}.

### 34.3 IviRFSigGen C Function Hierarchy

The IviRFSigGen class function hierarchy is shown in the following table.

**Table 34-6.** IviRFSigGen Function Hierarchy

Name or Class	Function Name
<b>Configuration...</b>	
<b>RF...</b>	
Configure RF	IviRFSigGen_ConfigureRF
Configure ALC Enabled	IviRFSigGen_ConfigureALCEnabled
Configure Output Enabled	IviRFSigGen_ConfigureOutputEnabled
<b>AnalogModulation...</b>	
Configure AM Enabled	IviRFSigGen_ConfigureAMEnabled
Configure AM External Coupling	IviRFSigGen_ConfigureAMEXternalCoupling
Configure AM	IviRFSigGen_ConfigureAM
Configure FM Enabled	IviRFSigGen_ConfigureFMEnabled
Configure FM External Coupling	IviRFSigGen_ConfigureFMExternalCoupling
Configure FM	IviRFSigGen_ConfigureFM
Configure PM Enabled	IviRFSigGen_ConfigurePMEnabled
Configure PM External Coupling	IviRFSigGen_ConfigurePMExternalCoupling
Configure PM	IviRFSigGen_ConfigurePM
Get Analog Modulation Source Name	IviRFSigGen_GetAnalogModulationSourceName
<b>PulseModulation...</b>	
Configure Pulse Modulation Enabled	IviRFSigGen_ConfigurePulseModulationEnabled
Configure Pulse Modulation Source	IviRFSigGen_ConfigurePulseModulationSource
Configure Pulse Modulation External Polarity	IviRFSigGen_ConfigurePulseModulationExternalPolarity
<b>LFGenerator...</b>	
Configure LFGenerator	IviRFSigGen_ConfigureLFGenerator
Get LFGenerator Name	IviRFSigGen_GetLFGeneratorName
Set Active LFGenerator	IviRFSigGen_SetActiveLFGenerator
Configure LFGenerator Output	IviRFSigGen_ConfigureLFGeneratorOutput
<b>Pulse Generator...</b>	
Configure Pulse External Trigger	IviRFSigGen_ConfigurePulseExternalTrigger
Configure Pulse Internal Trigger	IviRFSigGen_ConfigurePulseInternalTrigger
Configure Pulse	IviRFSigGen_ConfigurePulse
Configure Pulse Double	IviRFSigGen_ConfigurePulseDouble
Configure Pulse Output	IviRFSigGen_ConfigurePulseOutput

**Table 34-6. IviRFSigGen Function Hierarchy**

Name or Class	Function Name
<b>Sweep...</b>	
Configure Sweep	IviRFSigGen_ConfigureSweep
Configure Frequency Sweep Start Stop	IviRFSigGen_ConfigureFrequencySweepStartStop
Configure Frequency Sweep Center Span	IviRFSigGen_ConfigureFrequencySweepCenterSpan
Configure Frequency Sweep Time	IviRFSigGen_ConfigureFrequencySweepTime
Configure Power Sweep Start Stop	IviRFSigGen_ConfigurePowerSweepStartStop
Configure Power Sweep Time	IviRFSigGen_ConfigurePowerSweepTime
Configure Frequency Step Start Stop	IviRFSigGen_ConfigureFrequencyStepStartStop
Configure Frequency Step Dwell	IviRFSigGen_ConfigureFrequencyStepDwell
Reset Frequency Step	IviRFSigGen_ResetFrequencyStep
Configure Power Step Start Stop	IviRFSigGen_ConfigurePowerStepStartStop
Configure Power Step Dwell	IviRFSigGen_ConfigurePowerStepDwell
Reset Power Step	IviRFSigGen_ResetPowerStep
Create Frequency List	IviRFSigGen_CreateFrequencyList
Create Power List	IviRFSigGen_CreatePowerList
Create Frequency Power List	IviRFSigGen_CreateFrequencyPowerList
Select List	IviRFSigGen_SelectList
Clear All Lists	IviRFSigGen_ClearAllLists
Configure List Dwell	IviRFSigGen_ConfigureListDwell
Reset List	IviRFSigGen_ResetList
<b>ALC...</b>	
Configure ALC	IviRFSigGen_ConfigureALC
<b>ReferenceOscillator...</b>	
Configure Reference Oscillator	IviRFSigGen_ConfigureReferenceOscillator
<b>IQ...</b>	
Configure IQ Enabled	IviRFSigGen_ConfigureIQEnabled
Configure IQ	IviRFSigGen_ConfigureIQ
Calibrate IQ	IviRFSigGen_CalibrateIQ
Configure IQ Impairment Enabled	IviRFSigGen_ConfigureIQImpairmentEnabled
Configure IQ Impairment	IviRFSigGen_ConfigureIQImpairment
<b>ARB Generator...</b>	
Configure Arb	IviRFSigGen_ConfigureArb
Write Arb Waveform	IviRFSigGen_WriteArbWaveform
Select Arb Waveform	IviRFSigGen_SelectArbWaveform
Clear All Arb Waveforms	IviRFSigGen_ClearAllArbWaveforms
Query Arb Waveform Capabilities	IviRFSigGen_QueryArbWaveformCapabilities
Configure Arb Trigger Source	IviRFSigGen_ConfigureArbTriggerSource
Configure Arb External Trigger Slope	IviRFSigGen_ConfigureArbExternalTriggerSlope

**Table 34-6. IviRFSigGen Function Hierarchy**

Name or Class	Function Name
<b>Digital Modulation Base...</b>	
Get DigitalModulationBase Standard Name	IviRFSigGen_GetDigitalModulationBaseStandardName
Select DigitalModulationBase Standard	IviRFSigGen_SelectDigitalModulationBaseStandard
Configure DigitalModulationBase Clock Source	IviRFSigGen_ConfigureDigitalModulationBaseClockSource
Configure DigitalModulationBase Data Source	IviRFSigGen_ConfigureDigitalModulationBaseDataSource
Configure DigitalModulationBase PRBS Type	IviRFSigGen_ConfigureDigitalModulationBasePRBSType
Write DigitalModulationBase Bit Sequence	IviRFSigGen_WriteDigitalModulationBaseBitSequence
Select DigitalModulationBase Bit Sequence	IviRFSigGen_SelectDigitalModulationBaseBitSequence
Clear All DigitalModulationBase Bit Sequences	IviRFSigGen_ClearAllDigitalModulationBaseBitSequences
<b>CDMA...</b>	
Get CDMA Standard Name	IviRFSigGen_GetCDMAStandardName
Select CDMA Standard	IviRFSigGen_SelectCDMAStandard
Configure CDMA Clock Source	IviRFSigGen_ConfigureCDMAClockSource
Configure CDMA Trigger Source	IviRFSigGen_ConfigureCDMATriggerSource
Configure CDMA External Trigger Slope	IviRFSigGen_ConfigureCDMAExternalTriggerSlope
Get CDMA Test Model Name	IviRFSigGen_GetCDMATestModelName
Select CDMA Test Model	IviRFSigGen_SelectCDMATestModel
<b>TDMA...</b>	
Get TDMA Standard Name	IviRFSigGen_GetTDMAStandardName
Select TDMA Standard	IviRFSigGen_SelectTDMAStandard
Configure TDMA Clock Source	IviRFSigGen_ConfigureTDMAClockSource
Configure TDMA Trigger Source	IviRFSigGen_ConfigureTDMATriggerSource
Configure TDMA External Trigger Slope	IviRFSigGen_ConfigureTDMAExternalTriggerSlope
Get TDMA Frame Name	IviRFSigGen_GetTDMAFrameName
Select TDMA Frame	IviRFSigGen_SelectTDMAFrame
<b>Action...</b>	
Disable All Modulation	IviRFSigGen_DisableAllModulation
IsSettled	IviRFSigGen_IsSettled
WaitUntilSettled	IviRFSigGen_WaitUntilSettled
Send Software Trigger	IviRFSigGen_SendSoftwareTrigger

### 34.4 IviRFSigGen C Attribute Hierarchy

The IviRFSigGen C attribute hierarchy is shown in the following table.

**Table 34-7. IviRFSigGen C Attributes Hierarchy**

Category or Generic Attribute Name	C Defined Constant
<i>RF</i>	
Frequency	IVIRFSIGGEN_ATTR_FREQUENCY
Power Level	IVIRFSIGGEN_ATTR_POWER_LEVEL
ALC Enabled	IVIRFSIGGEN_ATTR_ALC_ENABLED
Output Enabled	IVIRFSIGGEN_ATTR_OUTPUT_ENABLED
<i>Analog Modulation</i>	
<i>AM</i>	
AM Enabled	IVIRFSIGGEN_ATTR_AM_ENABLED
AM Source	IVIRFSIGGEN_ATTR_AM_SOURCE
AM Scaling	IVIRFSIGGEN_ATTR_AM_SCALING
AM External Coupling	IVIRFSIGGEN_ATTR_AM_EXTERNAL_COUPLING
AM Nominal Voltage	IVIRFSIGGEN_ATTR_AM_NOMINAL_VOLTAGE
AM Depth	IVIRFSIGGEN_ATTR_AM_DEPTH
<i>FM</i>	
FM Enabled	IVIRFSIGGEN_ATTR_FM_ENABLED
FM Source	IVIRFSIGGEN_ATTR_FM_SOURCE
FM External Coupling	IVIRFSIGGEN_ATTR_FM_EXTERNAL_COUPLING
FM Nominal Voltage	IVIRFSIGGEN_ATTR_FM_NOMINAL_VOLTAGE
FM Deviation	IVIRFSIGGEN_ATTR_FM_DEVIATION
<i>PM</i>	
PM Enabled	IVIRFSIGGEN_ATTR_PM_ENABLED
PM Source	IVIRFSIGGEN_ATTR_PM_SOURCE
PM External Coupling	IVIRFSIGGEN_ATTR_PM_EXTERNAL_COUPLING
PM Nominal Voltage	IVIRFSIGGEN_ATTR_PM_NOMINAL_VOLTAGE
PM Deviation	IVIRFSIGGEN_ATTR_PM_DEVIATION
<i>Source</i>	
Analog Modulation Source Count	IVIRFSIGGEN_ATTR_ANALOG_MODULATION_SOURCE_COUNT
<i>Pulse Modulation</i>	
Pulse Modulation Enabled	IVIRFSIGGEN_ATTR_PULSE_MODULATION_ENABLED
Pulse Modulation Source	IVIRFSIGGEN_ATTR_PULSE_MODULATION_SOURCE
Pulse Modulation External Polarity	IVIRFSIGGEN_ATTR_PULSE_MODULATION_EXTERNAL_POLARITY

**Table 34-7. IviRFSigGen C Attributes Hierarchy**

Category or Generic Attribute Name	C Defined Constant
<i>LF Generator</i>	
Active LFGenerator	IVIRFSIGGEN_ATTR_ACTIVE_LF_GENERATOR
LFGenerator Count	IVIRFSIGGEN_ATTR_LF_GENERATOR_COUNT
LFGenerator Frequency	IVIRFSIGGEN_ATTR_LF_GENERATOR_FREQUENCY
LFGenerator Waveform	IVIRFSIGGEN_ATTR_LF_GENERATOR_WAVEFORM
<i>LF Generator Outputs</i>	
LFGenerator Output Amplitude	IVIRFSIGGEN_ATTR_LF_GENERATOR_OUTPUT_AMPLITUDE
LFGenerator Output Enabled	IVIRFSIGGEN_ATTR_LF_GENERATOR_OUTPUT_ENABLED
<i>Pulse Generator</i>	
Pulse Internal Trigger Period	IVIRFSIGGEN_ATTR_PULSE_INTERNAL_TRIGGER_PERIOD
Pulse Width	IVIRFSIGGEN_ATTR_PULSE_WIDTH
Pulse Gating Enabled	IVIRFSIGGEN_ATTR_PULSE_GATING_ENABLED
Pulse Trigger Source	IVIRFSIGGEN_ATTR_PULSE_TRIGGER_SOURCE
Pulse External Trigger Slope	IVIRFSIGGEN_ATTR_PULSE_EXTERNAL_TRIGGER_SLOPE
Pulse External Trigger Delay	IVIRFSIGGEN_ATTR_PULSE_EXTERNAL_TRIGGER_DELAY
<i>Double Pulse Generators</i>	
Pulse Double Enabled	IVIRFSIGGEN_ATTR_PULSE_DOUBLE_ENABLED
Pulse Double Delay	IVIRFSIGGEN_ATTR_PULSE_DOUBLE_DELAY
<i>Pulse Generator Output</i>	
Pulse Output Polarity	IVIRFSIGGEN_ATTR_PULSE_OUTPUT_POLARITY
Pulse Output Enabled	IVIRFSIGGEN_ATTR_PULSE_OUTPUT_ENABLED
<i>Sweep</i>	
Sweep Mode	IVIRFSIGGEN_ATTR_SWEEP_MODE
Sweep Trigger Source	IVIRFSIGGEN_ATTR_SWEEP_TRIGGER_SOURCE
<i>Frequency Sweep</i>	
Frequency Sweep Start	IVIRFSIGGEN_ATTR_FREQUENCY_SWEEP_START
Frequency Sweep Stop	IVIRFSIGGEN_ATTR_FREQUENCY_SWEEP_STOP
Frequency Sweep Time	IVIRFSIGGEN_ATTR_FREQUENCY_SWEEP_TIME
<i>Power Sweep</i>	
Power Sweep Start	IVIRFSIGGEN_ATTR_POWER_SWEEP_START
Power Sweep Stop	IVIRFSIGGEN_ATTR_POWER_SWEEP_STOP
Power Sweep Time	IVIRFSIGGEN_ATTR_POWER_SWEEP_TIME
<i>Frequency Step</i>	
Frequency Step Start	IVIRFSIGGEN_ATTR_FREQUENCY_STEP_START
Frequency Step Stop	IVIRFSIGGEN_ATTR_FREQUENCY_STEP_STOP

**Table 34-7. IviRFSigGen C Attributes Hierarchy**

Category or Generic Attribute Name	C Defined Constant
Frequency Step Scaling	IVIRFSIGGEN_ATTR_FREQUENCY_STEP_SCALING
Frequency Step Size	IVIRFSIGGEN_ATTR_FREQUENCY_STEP_SIZE
Frequency Step Single Step Enabled	IVIRFSIGGEN_ATTR_FREQUENCY_STEP_SINGLE_STEP_ENABLED
Frequency Step Dwell	IVIRFSIGGEN_ATTR_FREQUENCY_STEP_DWELL
<i>Power Step</i>	
Power Step Start	IVIRFSIGGEN_ATTR_POWER_STEP_START
Power Step Stop	IVIRFSIGGEN_ATTR_POWER_STEP_STOP
Power Step Size	IVIRFSIGGEN_ATTR_POWER_STEP_SIZE
Power Step Single Step Enabled	IVIRFSIGGEN_ATTR_POWER_STEP_SINGLE_STEP_ENABLED
Power Step Dwell	IVIRFSIGGEN_ATTR_POWER_STEP_DWELL
<i>List</i>	
List Selected Name	IVIRFSIGGEN_ATTR_LIST_SELECTED_NAME
List Single Step Enabled	IVIRFSIGGEN_ATTR_LIST_SINGLE_STEP_ENABLED
List Dwell	IVIRFSIGGEN_ATTR_LIST_DWELL
<i>ALC</i>	
ALC Source	IVIRFSIGGEN_ATTR_ALC_SOURCE
ALC Bandwidth	IVIRFSIGGEN_ATTR_ALC_BANDWIDTH
<i>Reference Oscillator</i>	
Reference Oscillator Source	IVIRFSIGGEN_ATTR_REFERENCE_OSCILLATOR_SOURCE
Reference Oscillator External Frequency	IVIRFSIGGEN_ATTR_REFERENCE_OSCILLATOR_EXTERNAL_FREQUENCY
<i>IQ</i>	
IQ Enabled	IVIRFSIGGEN_ATTR_IQ_ENABLED
IQ Nominal Voltage	IVIRFSIGGEN_ATTR_IQ_NOMINAL_VOLTAGE
IQ Source	IVIRFSIGGEN_ATTR_IQ_SOURCE
IQ Swap Enabled	IVIRFSIGGEN_ATTR_IQ_SWAP_ENABLED
<i>IQ Impairment</i>	
IQ Impairment Enabled	IVIRFSIGGEN_ATTR_IQ_IMPAIRMENT_ENABLED
IQ I-Offset	IVIRFSIGGEN_ATTR_IQ_I_OFFSET
IQ Q-Offset	IVIRFSIGGEN_ATTR_IQ_Q_OFFSET
IQ Ratio	IVIRFSIGGEN_ATTR_IQ_RATIO
IQ Skew	IVIRFSIGGEN_ATTR_IQ_SKEW
<i>ARB Generator</i>	
Arb Clock Frequency	IVIRFSIGGEN_ATTR_ARB_CLOCK_FREQUENCY
Arb Filter Frequency	IVIRFSIGGEN_ATTR_ARB_FILTER_FREQUENCY
<i>Waveform</i>	

**Table 34-7. IviRFSigGen C Attributes Hierarchy**

Category or Generic Attribute Name	C Defined Constant
Arb Selected Waveform	IVIRFSIGGEN_ATTR_ARB_SELECTED_WAVEFORM
Arb Max Number Waveforms	IVIRFSIGGEN_ATTR_ARB_MAX_NUMBER_WAVEFORMS
Arb Waveform Quantum	IVIRFSIGGEN_ATTR_ARB_WAVEFORM_QUANTUM
Arb Waveform Size Min	IVIRFSIGGEN_ATTR_ARB_WAVEFORM_SIZE_MIN
Arb Waveform Size Max	IVIRFSIGGEN_ATTR_ARB_WAVEFORM_SIZE_MAX
<i>Trigger</i>	
Arb Trigger Source	IVIRFSIGGEN_ATTR_ARB_TRIGGER_SOURCE
Arb External Trigger Slope	IVIRFSIGGEN_ATTR_ARB_EXTERNAL_TRIGGER_SLOPE
<i>Digital Modulation</i>	
DigitalModulationBase Standard Count	IVIRFSIGGEN_ATTR_DIGITAL_MODULATION_BASE_STANDARD_COUNT
DigitalModulationBase Selected Standard	IVIRFSIGGEN_ATTR_DIGITAL_MODULATION_BASE_SELECTED_STANDARD
DigitalModulationBase Data Source	IVIRFSIGGEN_ATTR_DIGITAL_MODULATION_BASE_DATA_SOURCE
DigitalModulationBase PRBS Type	IVIRFSIGGEN_ATTR_DIGITAL_MODULATION_BASE_PRBS_TYPE
DigitalModulationBase Bit Sequence	IVIRFSIGGEN_ATTR_DIGITAL_MODULATION_BASE_BIT_SEQUENCE
DigitalModulationBase Clock Source	IVIRFSIGGEN_ATTR_DIGITAL_MODULATION_BASE_CLOCK_SOURCE
DigitalModulationBase External Clock Type	IVIRFSIGGEN_ATTR_DIGITAL_MODULATION_BASE_EXTERNAL_CLOCK_TYPE
<i>CDMA</i>	
CDMA Standard Count	IVIRFSIGGEN_ATTR_CDMA_STANDARD_COUNT
CDMA Selected Standard	IVIRFSIGGEN_ATTR_CDMA_SELECTED_STANDARD
CDMA Test Model Count	IVIRFSIGGEN_ATTR_CDMA_TEST_MODEL_COUNT
CDMA Selected Test Model	IVIRFSIGGEN_ATTR_CDMA_SELECTED_TEST_MODEL
CDMA Clock Source	IVIRFSIGGEN_ATTR_CDMA_CLOCK_SOURCE
<i>Trigger</i>	
CDMA Trigger Source	IVIRFSIGGEN_ATTR_CDMA_TRIGGER_SOURCE
CDMA External Trigger Slope	IVIRFSIGGEN_ATTR_CDMA_EXTERNAL_TRIGGER_SLOPE
<i>TDMA</i>	
TDMA Standard Count	IVIRFSIGGEN_ATTR_TDMA_STANDARD_COUNT
TDMA Selected Standard	IVIRFSIGGEN_ATTR_TDMA_SELECTED_STANDARD
TDMA Frame Count	IVIRFSIGGEN_ATTR_TDMA_FRAME_COUNT
TDMA Selected Frame	IVIRFSIGGEN_ATTR_TDMA_SELECTED_FRAME
TDMA Clock Source	IVIRFSIGGEN_ATTR_TDMA_CLOCK_SOURCE
TDMA External Clock Type	IVIRFSIGGEN_ATTR_TDMA_EXTERNAL_CLOCK_TYPE
<i>Trigger</i>	
TDMA Trigger Source	IVIRFSIGGEN_ATTR_TDMA_TRIGGER_SOURCE



**Table 34-7.** IviRFSigGen C Attributes Hierarchy

Category or Generic Attribute Name	C Defined Constant
TDMA External Trigger Slope	IVIRFSIGGEN_ATTR_TDMA_EXTERNAL_TRIGGER_SLOPE

## Appendix A Specific Driver Development Guidelines

### A.1 Introduction

This section describes situations driver developers should be aware of when developing a specific instrument driver that complies with the IviRFSigGen class.

### A.2 Disabling Unused Extensions

Specific drivers are required to disable extension capability groups that an application program does not explicitly use. The specific driver can do so by setting the attributes of an extension capability group to the values that this section recommends. A specific driver can set these values for all extension capability groups when the `IviRFSigGen_init`, `IviRFSigGen_InitWithOptions`, or `IviRFSigGen_reset` functions execute. This assumes that the extension capability groups remain disabled until the application program explicitly uses them. For the large majority of instruments, this assumption is true.

Under certain conditions, a specific driver might have to implement a more complex approach. For some instruments, configuring a capability group might affect instrument settings that correspond to an unused extension capability group. If these instrument settings affect the behavior of the instrument, then this might result in an interchangeability problem. If this can occur, the specific driver shall take appropriate action so that the instrument settings that correspond to the unused extension capability group do not affect the behavior of the instrument when the application program performs an operation that might be affected by those settings.

The remainder of this section recommends attribute values that effectively disable each extension capability group.

#### Disabling the IviRFSigGenModulateAM Extension Group

Attribute values that effectively disable the IviRFSigGenModulateAM extension group are shown in the following table. This functionality is also encapsulated in the Disable All Modulation function.

**Table A.1.** Values for Disabling the IviRFSigGenModulateAM Extension Group

Attribute	Value
AM Enabled	False

#### Disabling the IviRFSigGenModulateFM Extension Group

Attribute values that effectively disable the IviRFSigGenModulateFM extension group are shown in the following table. This functionality is also encapsulated in the Disable All Modulation function.

**Table A.2.** Values for Disabling the IviRFSigGenModulateFM Extension Group

Attribute	Value
FM Enabled	False

#### Disabling the IviRFSigGenModulatePM Extension Group

Attribute values that effectively disable the IviRFSigGenModulatePM extension group are shown in the following table. This functionality is also encapsulated in the Disable All Modulation function.

**Table A.3.** Values for Disabling the IviRFSigGenModulatePM Extension Group

Attribute	Value
PM Enabled	False

#### Disabling the IviRFSigGenModulatePulse Extension Group

Attribute values that effectively disable the IviRFSigGenModulatePulse extension group are shown in the following table. This functionality is also encapsulated in the Disable All Modulation function.

**Table A.4.** Values for Disabling the IviRFSigGenModulatePulse Extension Group

Attribute	Value
Pulse Modulation Enabled	False

#### Disabling the IviRFSigGenLFGenerator Extension Group

The IviRFSigGenLFGenerator extension group affects the instrument behavior only when its attributes are used. Therefore, this specification does not recommend attribute values that disable the IviRFSigGenLFGenerator extension group.

#### Disabling the IviRFSigGenLFGeneratorOutput Extension Group

Attribute values that effectively disable the IviRFSigGenLFGeneratorOutput extension group are shown in the following table.

**Table A.5.** Values for Disabling the IviRFSigGenLFGeneratorOutput Extension Group

Attribute	Value
LFGenerator Output Enabled	False

#### Disabling the IviRFSigGenPulseGenerator Extension Group

The IviRFSigGenPulseGenerator extension group affects the instrument behavior only when its attributes are used. Therefore, this specification does not recommend attribute values that disable the IviRFSigGenPulseGenerator extension group.

#### Disabling the IviRFSigGenPulseDoubleGenerator Extension Group

Attribute values that effectively disable the IviRFSigGenPulseDoubleGenerator extension group are shown in the following table.

**Table A.6.** Values for Disabling the IviRFSigGenPulseDoubleGenerator Extension Group

Attribute	Value
Pulse Double Enabled	False

#### **Disabling the IviRFSigGenPulseGeneratorOutput Extension Group**

Attribute values that effectively disable the IviRFSigGenPulseGeneratorOutput extension group are shown in the following table.

**Table A.7.** Values for Disabling the IviRFSigGenPulseGeneratorOutput Extension Group

Attribute	Value
Pulse Output Enabled	False

#### **Disabling the IviRFSigGenSweep Extension Group**

Attribute values that effectively disable the IviRFSigGenSweep extension group are shown in the following table.

**Table A.8.** Values for Disabling the IviRFSigGenSweep Extension Group

Attribute	Value
Sweep Mode	None

#### **Disabling the IviRFSigGenFrequencySweep Extension Group**

Attribute values that effectively disable the IviRFSigGenFrequencySweep extension group are shown in the following table.

**Table A.9.** Values for Disabling the IviRFSigGenFrequencySweep Extension Group

Attribute	Value
Sweep Mode	None

#### **Disabling the IviRFSigGenPowerSweep Extension Group**

Attribute values that effectively disable the IviRFSigGenPowerSweep extension group are shown in the following table.

**Table A.10.** Values for Disabling the IviRFSigGenPowerSweep Extension Group

Attribute	Value
Sweep Mode	None

### Disabling the IviRFSigGenFrequencyStep Extension Group

Attribute values that effectively disable the IviRFSigGenFrequencyStep extension group are shown in the following table.

**Table A.11.** Values for Disabling the IviRFSigGenFrequencyStep Extension Group

Attribute	Value
Sweep Mode	None

### Disabling the IviRFSigGenPowerStep Extension Group

Attribute values that effectively disable the IviRFSigGenPowerStep extension group are shown in the following table.

**Table A.12.** Values for Disabling the IviRFSigGenPowerStep Extension Group

Attribute	Value
Sweep Mode	None

### Disabling the IviRFSigGenList Extension Group

Attribute values that effectively disable the IviRFSigGenList extension group are shown in the following table.

**Table A.13.** Values for Disabling the IviRFSigGenList Extension Group

Attribute	Value
Sweep Mode	None

### Disabling the IviRFSigGenALC Extension Group

Attribute values that effectively disable the IviRFSigGenALC extension group are shown in the following table.

**Table A.14.** Values for Disabling the IviRFSigGenALC Extension Group

Attribute	Value
ALC Enabled	False

### Disabling the IviRFSigGenReferenceOscillator Extension Group

The IviRFSigGenReferenceOscillator extension group affects the instrument behavior only when the Reference Oscillator Source attribute is set to External. Therefore, this specification does not recommend attribute values that disable the IviRFSigGenReferenceOscillator extension group.

### Disabling the IviRFSigGenSoftwareTrigger Extension Group

The IviRFSigGenSoftwareTrigger extension group affects the instrument behavior only when the Trigger Source attribute is set to Software Trigger. Therefore, this specification does not recommend attribute values that disable the IviRFSigGenSoftwareTrigger extension group.

#### Disabling the IviRFSigGenModulateIQ Extension Group

Attribute values that effectively disable the IviRFSigGenModulateIQ extension group are shown in the following table. This functionality is also encapsulated in the Disable All Modulation function.

**Table A.15.** Values for Disabling the IviRFSigGenModulateIQ Extension Group

Attribute	Value
IQ Enabled	False

#### Disabling the IviRFSigGenIQImpairment Extension Group

Attribute values that effectively disable the IviRFSigGenIQImpairment extension group are shown in the following table.

**Table A.16.** Values for Disabling the IviRFSigGenModulateIQ Extension Group

Attribute	Value
IQ Impairment Enabled	False

#### Disabling the IviRFSigGenArbGenerator Extension Group

The IviRFSigGenArbGenerator extension group affects the instrument behavior only when the end user calls any of its functions. Therefore, this specification does not recommend attribute values that disable the IviRFSigGenArbGenerator extension group.

#### Disabling the IviRFSigGenDigitalModulationBase Extension Group

The IviRFSigGenDigitalModulationBase extension group affects the instrument behavior only when the end user calls any of its functions. Therefore, this specification does not recommend attribute values that disable the IviRFSigGenDigitalModulationBase extension group. This functionality is also encapsulated in the Disable All Modulation function.

#### Disabling the IviRFSigGenCDMABase Extension Group

The IviRFSigGenCDMABase extension group affects the instrument behavior only when the end user calls any of its functions. Therefore, this specification does not recommend attribute values that disable the IviRFSigGenCDMABase extension group. This functionality is also encapsulated in the Disable All Modulation function.

#### Disabling the IviRFSigGenTDMABase Extension Group

The IviRFSigGenTDMABase extension group affects the instrument behavior only when the end user calls any of its functions. Therefore, this specification does not recommend attribute values that disable the IviRFSigGenTDMABase extension group. This functionality is also encapsulated in the Disable All Modulation function.

### **A.3 Query Instrument Status**

Based on the value of Query Instrument Status, the specific driver may check the status of the instrument to see if it has encountered an error.

## Appendix B Interchangeability Checking Rules

### B.1 Introduction

IVI drivers have a feature called interchangeability checking. Interchangeability checking returns a warning when it encounters a situation where the application program might not produce the same behavior when the user attempts to use a different instrument.

### B.2 When to Perform Interchangeability Checking

Interchangeability checking occurs when all of the following conditions are met:

- The `Interchange Check` attribute is set to `True`
- The user calls one of the following functions.
  - `IsSettled`
  - `WaitUntilSettled`

### B.3 Interchangeability Checking Rules

Interchangeability checking is performed on a capability group basis. When enabled, interchangeability checking is always performed on the base capability group. In addition, interchangeability checking is performed on extension capability groups for which the user has ever set any of the attributes of the group. If the user has never set any attributes of an extension capability group, interchangeability checking is not performed on that group.

In general interchangeability warnings are generated if the following conditions are encountered:

- An attribute that affects the behavior of the instrument is not in a state that the user specifies.
- The user sets a class driver defined attribute to an instrument-specific value.
- The user configures the value of an attribute that the class defines as read-only. In a few cases the class drivers define read-only attributes that specific drivers might implement as read/write.

The remainder of this section defines additional rules and exceptions for each capability group.

#### IviRFSigGenBase Capability Group

1. If the `Output Enabled` attribute is set to `False`, all other attributes in the `IviRFSigGenBase` extension group need not be in a user-specified state.

#### IviRFSigGenModulateAM Capability Group

1. If the `Output Enabled` attribute is set to `False`, attributes in the `IviRFSigGenModulateAM` extension group need not be in a user-specified state.
2. If the `AM Source` attribute does not include the `??external modulation source??`, the `AM External Coupling` attribute need not be in a user specified state.
3. If the `AM Enabled` attribute is set to `False`, all other attributes in the `IviRFSigGenModulateAM` extension group need not be in a user-specified state.



### **IviRFSigGenModulateFM Capability Group**

1. If the Output Enabled attribute is set to False, attributes in the IviRFSigGenModulateFM extension group need not be in a user-specified state.
2. If the FM Source attribute does not include the ??external modulation source??. the FM External Coupling attribute need not be in a user specified state.
3. If the FM Enabled attribute is set to False, all other attributes in the IviRFSigGenModulateFM extension group need not be in a user-specified state.

### **IviRFSigGenModulatePM Capability Group**

1. If the Output Enabled attribute is set to False, attributes in the IviRFSigGenModulatePM extension group need not be in a user-specified state.
2. If the PM Source attribute does not include the ??external modulation source??. the PM External Coupling attribute need not be in a user specified state.
3. If the PM Enabled attribute is set to False, all other attributes in the IviRFSigGenModulatePM extension group need not be in a user-specified state.

### **IviRFSigGenAnalogModulationSources Capability Group**

No additional interchangeability rules or exceptions are defined for the IviRFSigGenAnalogModulationSources capability group.

### **IviRFSigGenModulatePulse Capability Group**

1. If the Output Enabled attribute is set to False, attributes in the IviRFSigGenModulatePulse extension group need not be in a user-specified state.
2. If the Pulse Modulation Source attribute is not set to External, the Pulse Modulation External Polarity attribute need not be in a user specified state.
3. If the Pulse Modulation Enabled attribute is set to False, all other attributes in the IviRFSigGenModulatePulse extension group need not be in a user-specified state.

### **IviRFSigGenLFGenerator Capability Group**

1. If the LFGenerator Output Enabled attribute is set to True  
OR  
If Output Enabled set to True AND one of the following sources includes an internal source
  - AM Source
  - FM Source
  - PM SourceAND any of the following attributes are set to True
  - AM Enabled
  - FM Enabled
  - PM Enabledthen the attributes in the IviRFSigGenLFGenerator extension group shall be in a user-specified state.

### **IviRFSigGenLFGeneratorOutput Capability Group**

1. If the LFGenerator Output Enabled attribute is set to False, the LFGenerator Output Amplitude attribute need not be in a user-specified state.

### **IviRFSigGenPulseGenerator Capability Group**

1. If the Pulse Output Enabled attribute is set to True  
OR  
If Output Enabled set to True AND the Pulse Generator Source is set to Internal AND the Pulse Modulation Enabled attribute is set to True, the attributes in the IviRFSigGenPulseGenerator extension group shall be in a user-specified state.
2. If the Pulse Trigger Source attribute is not set to Internal, the Pulse Internal Trigger Period attribute and the need not be in a user-specified state.
3. If the Pulse Trigger Source attribute is not set to External, the Pulse External Trigger Delay attribute and the Pulse External Trigger Slope attribute need not be in a user-specified state.

### **IviRFSigGenPulseDoubleGenerator Capability Group**

1. If the Pulse Double Enabled attribute is set to False, attributes in the IviRFSigGenPulseDoubleGenerator extension group need not be in a user-specified state.
2. If the IviRFSigGenPulseGenerator extension group does not need to be checked for interchangeability, then attributes in the IviRFSigGenPulseDoubleGenerator extension group need not be in a user-specified state.

### **IviRFSigGenPulseGeneratorOutput Capability Group**

1. If the Pulse Output Enabled attribute is set to False, all other attributes in the IviRFSigGenPulseGeneratorOutput extension group need not be in a user-specified state.

### **IviRFSigGenSweep Capability Group**

1. If the Sweep Mode attribute is set to None, all other attributes in the IviRFSigGenSweep extension group need not be in a user-specified state.
2. If the Output Enabled attribute is set to False, all the attributes in the IviRFSigGenSweep extension group need not be in a user-specified state.

### **IviRFSigGenFrequencySweep Capability Group**

1. If the Sweep Mode attribute is not set to Frequency Sweep, attributes in the IviRFSigGenFrequencySweep extension group need not be in a user-specified state.
2. If the IviRFSigGenSweep extension group does not need to be checked for interchangeability, then attributes in the IviRFSigGenFrequencySweep extension group need not be in a user-specified state.

### **IviRFSigGenPowerSweep Capability Group**

1. If the Sweep Mode attribute is not set to Power Sweep, attributes in the IviRFSigGenPowerSweep extension group need not be in a user-specified state.

2. If the IviRFSigGenSweep extension group does not need to be checked for interchangeability, then attributes in the IviRFSigGenPowerSweep extension group need not be in a user-specified state.

#### **IviRFSigGenFrequencyStep Capability Group**

1. If the Sweep Mode attribute is not set to Frequency Step, attributes in the IviRFSigGenFrequencyStep extension group need not be in a user-specified state.
2. If the IviRFSigGenSweep extension group does not need to be checked for interchangeability, then attributes in the IviRFSigGenFrequencyStep extension group need not be in a user-specified state.

#### **IviRFSigGenPowerStep Capability Group**

1. If the Sweep Mode attribute is not set to Power Step, attributes in the IviRFSigGenPowerStep extension group need not be in a user-specified state.
2. If the IviRFSigGenSweep extension group does not need to be checked for interchangeability, then attributes in the IviRFSigGenPowerStep extension group need not be in a user-specified state.

#### **IviRFSigGenList Capability Group**

1. If the Sweep Mode attribute is not set to List, attributes in the IviRFSigGenList extension group need not be in a user-specified state.
2. If the IviRFSigGenSweep extension group does not need to be checked for interchangeability, then attributes in the IviRFSigGenList extension group need not be in a user-specified state.

#### **IviRFSigGenALC Capability Group**

1. If the ALC Enabled attribute is set to False, attributes in the IviRFSigGenALC extension group need not be in a user-specified state.
2. If the Output Enabled attribute is set to False, all the attributes in the IviRFSigGenALC extension group need not be in a user-specified state.

#### **IviRFSigGenReferenceOscillator Capability Group**

1. If the Output Enabled attribute is set to False, all the attributes in the IviRFSigGenReferenceOscillator extension group need not be in a user-specified state.

#### **IviRFSigGenSoftwareTrigger Capability Group**

No additional interchangeability rules or exceptions are defined for the IviRFSigGenSoftwareTrigger capability group.

#### **IviRFSigGenModulateIQ Capability Group**

1. If the Output Enabled attribute is set to False, attributes in the IviRFSigGenModulateIQ extension group need not be in a user-specified state.
2. If the IQ Enabled attribute is set to False, attributes in the IviRFSigGenModulateIQ extension group need not be in a user-specified state.

### **IviRFSigGenIQImpairment Capability Group**

1. If the IQ Impairment Enabled attribute is set to False, attributes in the IviRFSigGenIQImpairment extension group need not be in a user-specified state.
2. If the IviRFSigGenModulateIQ extension group does not need to be checked for interchangeability, then attributes in the IviRFSigGenIQImpairment extension group need not be in a user-specified state.

### **IviRFSigGenArbGenerator Capability Group**

1. If the IQ Source attribute is not set to ArbGenerator, attributes in the IviRFSigGenArbGenerator extension group need not be in a user-specified state.
2. If the Arb Trigger Source attribute is not set to External, the Arb External Trigger Slope attribute need not be in a user-specified state.
3. If the IviRFSigGenModulateIQ extension group does not need to be checked for interchangeability, then attributes in the IviRFSigGenArbGenerator extension group need not be in a user-specified state.

### **IviRFSigGenDigitalModulationBase Capability Group**

1. If the Output Enabled attribute is set to False, attributes in the IviRFSigGenDigitalModulationBase extension group need not be in a user-specified state.
2. If the IQ Source attribute is not set to DigitalModulationBase, attributes in the IviRFSigGenDigitalModulationBase extension group need not be in a user-specified state.
3. If the DigitalModulationBase Clock Source attribute is not set to External, the External Clock Type attribute need not be in a user-specified state.
4. If the DigitalModulationBase Data Source attribute is not set to PRBS, the DigitalModulationBase PRBS Type attribute need not be in a user-specified state.
5. If the DigitalModulationBase Data Source attribute is not set to Bit Sequence, the DigitalModulationBase Selected Bit Sequence attribute need not be in a user-specified state.
6. If the IviRFSigGenModulateIQ extension group does not need to be checked for interchangeability, then attributes in the IviRFSigGenDigitalModulationBase extension group need not be in a user-specified state.

### **IviRFSigGenCDMABase Capability Group**

1. If the Output Enabled attribute is set to False, attributes in the IviRFSigGenCDMABase extension group need not be in a user-specified state.
2. If the IQ Source attribute is not set to CDMABase, attributes in the IviRFSigGenCDMABase extension group need not be in a user-specified state.
3. If the CDMA Clock Source is not set to External, the CDMA External Trigger Slope attribute need not be in a user-specified state.
4. If the IviRFSigGenModulateIQ extension group does not need to be checked for interchangeability, then attributes in the IviRFSigGenCDMABase extension group need not be in a user-specified state.

## **IviRFSigGenTDMABase Capability Group**

1. If the Output Enabled attribute is set to False, attributes in the IviRFSigGenTDMABase extension group need not be in a user-specified state.
2. If the IQ Source attribute is not set to TDMABase, attributes in the IviRFSigGenTDMABase extension group need not be in a user-specified state.
3. If the TDMA Clock Source is not set to External, the TDMA External Trigger Slope attribute need not be in a user-specified state.
4. If the IviRFSigGenModulateIQ extension group does not need to be checked for interchangeability, then attributes in the IviRFSigGenTDMABase extension group need not be in a user-specified state.