IVI

Interchangeable
Virtual
Instruments

IVI-4.5: IviACPwr Class Specification

March 11, 2011 Edition
Revision 1.2

Important Information

IVI-4.5 IviACPwr Class Specification 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.

The IVI Foundation wants to receive your comments on this specification. You can contact the Foundation through the web site at www.ivifoundation.org.

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.

Table of Contents

Index of Tables

Index of Figures

IviACPwr Class Specification

IviACPwr Revision History

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

Table 1-1. IviACPwr Class Specification Revisions

Revision Number	Date of Revision	Revision Notes
Revision 0.1 draft	October 2006	Original draft.
Revision 0.2 draft	February 2007	Original draft revisions.
Revision 0.3 draft	May 2007	Revising Adding Extensions
Revision 4.1f draft	December 2007	Completed adding of all Extensions
Revision 4.1M draft	July 18, 2008	Specification ready for Driver code Prototyping
Revision 4.1N draft	August 29, 2008	Updated as needed to create baseline project file.
Revision 4.1P draft	October 27, 2008	Updated based on addition of Help Sections.
Revision 4.1R draft	November 6, 2008	Updated Ramp descriptions
Revision 4.1S draft	December 18, 2008	Waveform DC Offset, DC Generation
Revision 4.1T draft	March 4, 2009	Updated Current/Voltage Protection
Revision 4.1U draft	April 15, 2009	Changed repeated capability to OutputPhase
Revision 4.1V draft	May 23, 2009	Updated ramp sections, added Fetch Measurement Array function and separated enums.
Revision 4.1W draft	May 3, 2010	Updated DCGeneration extension group
Revision 4.1X draft	May 11, 2010	Added AbortFrequencyRamp, fixed RampCurrent
Revision 1.0	July 14, 2010	Posted for 45 day review.
Revision 1.0	October 8, 2010	QueryArbWaveformCatalog parameter order and misc minor cleanup.
Revision 1.1	January 27, 2011	Editorial changes in the following COM method prototypes: Query Arbitrary Waveform Catalog, Output Impedance Enabled, Output Impedance Resistive, Abort Voltage Ramp, Abort Current Ramp. Updates in the COM and C function hierarchy tables.
Revision 1.2	March 11, 2011	Misc minor editorial changes

1 Overview of the IviACPwr Specification

1.1 Introduction

This specification defines the IVI class for AC Power Sources. The IviACPwr class is designed to support the typical AC power source as well as common extended functionality found in more complex instruments. This section summarizes the *IviACPwr 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:

IviACPwr Class Overview

The definitions of terms and acronyms

References

1.2 IviACPwr Class Overview

This specification describes the IVI class for AC power sources. The IviACPwr class is designed to support the typical single or multi-phase AC power source as well as common extended functionality found in more complex instruments. The IviACPwr class conceptualizes an AC power source as an instrument capable of generating an AC power signal.

The IviACPwrBase capability group provides the ability to configure an AC power source for basic signal output. This includes setting the current limit, voltage level, frequency, and enabling the output. The IviACPwrBase capability group is described in Section 4, *IviACPwrBase Capability Group*.

The IviACPwrMeasurement extension group provides the ability to measure AC power source output signals such as voltage and current. This extension group is described in Section 5, *IviACPwrMeasurement Extension Group*.

The IviACPwrPhase extension group provides the ability to set the output phase relationships of a multiphase power source. This extension group is described in Section 6, *IviACPwrPhase Extension* Group.

The IviACPwrExternalSync extension provides the ability to synchronize the power source output frequency with an external signal. It also allows the querying the synchronization state of the power source. This extension group is described in Section 7, *IviACPwrExternalSync Extension* Group.

The IviACPwrCurrentProtection extension group provides programmable current protection to detect an over-current condition and automatically disable the output. It also allows the querying of the current protection state of the power source. This extension group is described in Section 8, *IviACPwrCurrentProtection Extension Group*.

The IviACPwrVoltageProtection extension group provides programmable voltage protection to detect an under/over voltage condition and automatically disable the output. It also allows the querying of the voltage protection state of the power source. This extension group is described in Section 9, *IviACPwrVoltageProtection Extension Group*.

The IviACPwrArbWaveform extension group provides the ability to create user-defined arbitrary waveforms. The IviACPwrArbWaveform extension group includes functions for Writing and Clearing arbitrary waveforms, and Querying arbitrary waveform catalogs. This extension group is described in Section 10, *IviACPwrArbWaveform Extension Group.*

The IviACPwrImpedance extension group provides the ability to set and query the power source output impedance. With this extension, the driver can configure the power source output resistance and inductance. This extension group is described in Section 11, *IviACPwrImpedance Extension Group*.

The IviACPwrDCGeneration extension group provides the ability to configure the output of the AC Power Source with an AC only output, DC only output or an AC signal with DC offset. This extension group is described in Section 12, *IviACPwrDCGeneration Extension Group*.

The IviACPwrVoltageRamp extension group provides the ability to configure a linear voltage ramp with starting and ending voltage in volts RMS and ramp duration in seconds. This extension group is described in Section 13, *IviACPwrVoltageRamp Extension Group*.

The IviACPwrCurrentRamp extension group provides the ability to configure a linear current ramp with starting and ending current in amps and ramp duration in seconds. This extension group is described in Section 14, *IviACPwrCurrentRamp Extension Group*.

The IviACPwrFrequencyRamp extension group provides the ability to configure a linear frequency ramp with starting and ending frequencies in hertz and ramp duration in seconds. This extension group is described in Section 15, *IviACPwrFrequencyRamp Extension Group*.

1.3 References

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

- IVI Charter Document
- IVI-3.1: Driver Architecture Specification
- IVI-3.2: Inherent Capabilities Specification
- IVI-3.3: Standard Cross-Class Capabilities Specification
- VPP-3.x - VXI*plug&play* Instrument Driver Specifications
- VPP-4.x - Virtual Instrument Software Architecture (VISA) Specifications
- IVI-5.0: Glossary

1.4 Definitions of Terms and Acronyms

This section defines terms acronyms that are specific to the IviACPwr class.

Constant Voltage Mode

The AC power source is said to be in the constant voltage mode when the output voltage reaches the Voltage Level value before the output current reaches the Current Limit value. In the constant voltage mode, the power source output voltage remains constant at the Voltage Level value and the output current can vary based on the load.

Constant Current Mode

The AC power source is said to be in the constant current mode when the output current reaches the Current Limit value before the output voltage reaches the Voltage Level value. In the constant current mode, the power source output current remains constant at the Current Limit value and the output voltage can vary based on the load.

Protection Mode

The AC power source is said to be in a protection mode when the output is disabled by the power source due to a voltage or current protection condition. Protection mode is intended to protect the load from over or under voltage or over current.

2 IviACPwr Class Capabilities

2.1 Introduction

The IviACPwr specification divides AC power source capabilities into a base capability group and multiple extension capability groups. Each capability group is discussed in a separate section. This section defines names for each capability group and gives a brief overview for each capability group.

2.2 IviACPwr Group Names

The capability group names for the IviACPwr 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 Group Capabilities attribute.

Table 2-1. IviACPwr Group Names

Group Name	Description
IviACPwrBase	Base: Complies with the IviACPwr base capabilities. This group supports the ability to generate an AC power signal, to specify output limits, and to control the behavior of the power source when the output is greater than or equal to one of the limits
IviACPwrMeasurement	Extension: Supports the ability to query the instrument for the measurement characteristics of the output signal provides the capability to take measurements on the output signal.
IviACPwrPhase	Extension: Supports the ability to set and measure output phase relationship.
IviACPwrExternalSync	Extension: Supports the ability of synchronizing an AC Power Source’s output frequency with an external signal.
IviACPwrCurrentProtection	Extension: Supports the ability to program current protection.
IviACPwrVoltageProtection	Extension: Supports the ability to program voltage protection.
IviACPwrArbWaveform	Extension: Supports the ability to produce waveforms other than sine waves on the output signal.
IviACPwrImpedance	Extension: Supports the ability to set and query the instrument for the output impedance characteristics of the output signal.
IviACPwrDCGeneration	Extension: Supports the ability to configure the output of the AC Power Source with DC Content.
IviACPwrVoltageRamp	Extension: Supports the ability to configure a linear Voltage ramp with starting and ending voltage in volts (RMS) and ramp duration in seconds.
IviACPwrCurrentRamp	Extension: Supports the ability to configure a Current ramp starting and ending Current and ramp duration in seconds.
IviACPwrFrequencyRamp	Extension: Supports the ability to configure a linear Frequency ramp with starting and ending Frequencies and ramp duration

2.3 Repeated Capability Names

The IviACPwr Class Specification defines one repeated capability. 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*, and Section 5.8, *Repeated Capability Identifiers and Selectors*.

OutputPhase

2.3.1 OutputPhase

In the configuration store, the name for the Output Phase repeated capability shall be exactly one of "OutputPhase" or "IviACPwrOutputPhase". Drivers that implement multiple repeated capabilities with the name "OutputPhase" shall use the latter form to disambiguate the names.

Phase names can be vendor unique but the recommended names are "PhaseA" for output phase index 1, "PhaseB" for output phase index 2 and so on. This method of phase naming is assumed throughout this document and is based on industry standard conventions for single and multiphase AC power sources.

In addition to individual phase names, drivers must also support the special phase name "AllPhases". When used with attributes or functions, this name sets all phases of that attribute or function to the same value.

2.3.2 Multi Phase Attributes

Instrument vendors must allow setting all phases of multi-phase attributes and functions to the same value using the "AllPhases" repeated capability name for the following capability groups:

- IviACPwrBase
- IviACPwrCurrentProtection
- IviACPwrVoltageProtection
- IviACPwrImpedance
- IviACPwrDCGeneration
- IviACPwrVoltageRamp
- IviACPwrCurrentRamp

Instrument vendors may prevent the user from setting the phases for a particular multi-phase attribute independently. In this case, only the "AllPhases" repeated capability name is allowed otherwise it returns the All Phases Required error.

If you attempt to read an attribute using the "AllPhases" repeated capability name and one of the phases contains a different value, the error Phase Values Different is returned. You must then query each phase separately to obtain the attribute value for each phase.

2.3.3 Phase Group

A driver session must only support one phase group. A phase group is defined as one or more output phases that have the same frequency and are synchronized to each other. If an AC power source supports more than one phase group, then separate driver sessions must be opened for each group.

If an AC power source supports more than one phase group, this specification assumes that each of these phase groups are independent of each other and behave as if they are completely separate devices. For the case where a device has multiple phase groups, the user initializes a separate instrument driver session to control each phase group. The Phase Group that is being used must be identified during initialization. The mechanism for identifying the Phase Group to be used during initialization is instrument specific. A driver developer can choose to use the configuration store, an initialization option string, or other mechanism. If there are synchronization or configuration requirements that are common across the multiple phase groups, they will be handled in an instrument specific manner.

2.3.4 Configuring Single/MultiPhase Operation

AC Power sources that support single or multi-phase operation must only be configured during driver initialization. In other words, once a driver session is opened, the number of output phases remains fixed and must match the number of output phase connections the user connects to their load. To change the number of phases, the user must close the driver session and open a new one. There are two mechanisms available to configure the number of output phases during driver initialization:

- Create a driver session in the configuration store and include the string “NumPhases:n” in the DriverSetup field.
- Pass the token “NumPhases:n” to the InitWithOptions DriverSetup parameter. Note that this overrides the value in the configuration store.

The initialize function will return an error if the AC power source does not support the requested number of phases. If neither of the above methods is used, the driver initializes the number of phases based on the current instrument state or vendor default number of phases . This sets an interchangeability warning.

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 IviACPwr Class Specification, a specific driver shall implement the inherent capabilities that *IVI-3.2: Inherent IVI Capabilities Specification* defines, and the IviACPwrBase 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 AC Power Source to apply the minimum amount of power possible at the output terminals. Setting the voltage to a value close to zero, setting the current limit to value close to zero, or physically disconnecting the power source from the output terminals meets this requirement. Other techniques are also allowed.

3.2 Capability Group Compliance

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

4 IviACPwrBase Capability Group

4.1 Overview

The IviACPwrBase capability group supports the basic AC power source capabilities. The user can specify the Voltage Level, Current Limit, Frequency and control the operation of the output relay/contactor.

This specification uses the following terms to describe the power source’s output: Voltage Level, Frequency, Phase Angle (IviACPwrPhase extension group), Waveform, Current limit, DC Voltage Level (IviACPwrDCGeneration extension group) and Impedance (IviACPwrImpedance extension group).

4.1.1 User Specified Settings

Voltage Level The RMS AC output voltage of the AC power source. The user configures the voltage level with the Voltage Level attribute.

Frequency The frequency of the output AC voltage waveform of the AC power source. The user configures the frequency with the Frequency attribute.

Waveform The waveform the power source uses to generate the output voltage. The user configures the waveform with the Waveform attribute.

Current Limit Sets the maximum allowable current the AC power source will deliver. The user configures the current limit with the Current Limit attribute.

Output Enabled Enables/Disables the output contactor/relay of the AC power source. The user configures the output enable with the Output Enabled attribute.

Voltage Range Specifies the output voltage range. AC power sources may support multiple voltage ranges to allow greater output voltage or current. The user configures the voltage range with the Voltage Range attribute.

Frequency Range Specifies the output frequency range. AC power sources may support multiple frequency ranges. The user configures the frequency range with the Frequency Range attribute.

4.1.2 User Queriable Settings

Number of Phases Returns the number of physical output phases.

Number of Voltage Ranges Returns the number of supported voltage ranges.

Voltage Range Capabilities Returns the minimum and maximum voltages supported by a particular voltage range.

Number of Frequency Ranges Returns the number of supported frequency ranges.

Frequency Range Capabilities Returns the minimum and maximum frequency supported by a particular frequency range

4.1.3 Power Source Voltage and Current Scenarios

The signal that the power source produces depends on the values of the voltage level and current limit that the user supplies, and the impedance of the load to which the power source is attached. Therefore, the power source might not produce the exact voltage or current that the user configures. Figure 4-1 shows the possible output scenarios.

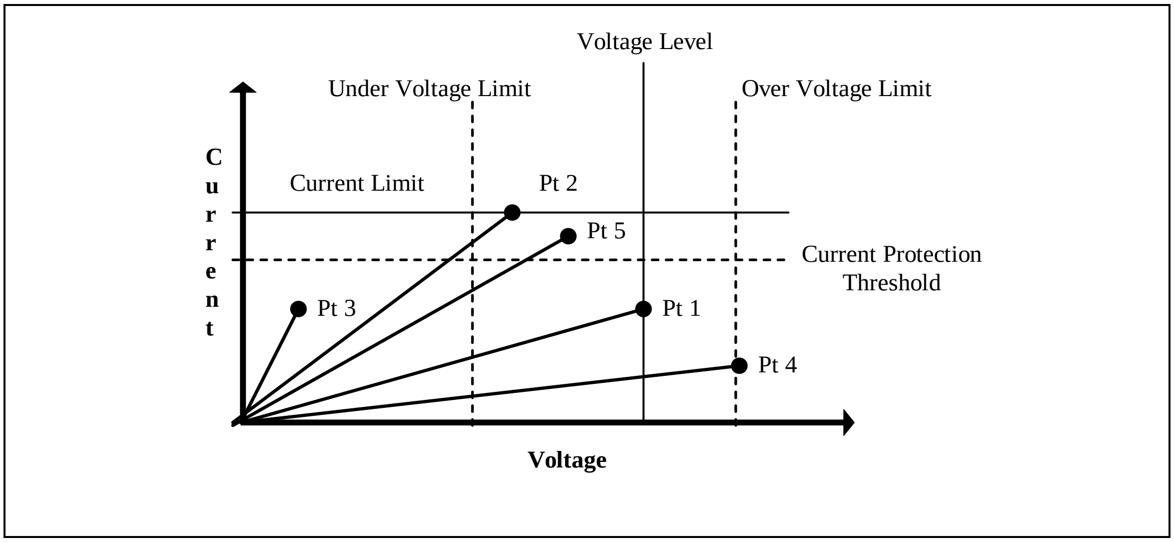


Figure 4-1 AC Power Source Operating Point

Point 1 The AC power source output has reached the voltage level before it reached the current limit. This is an example of the AC power source operating in the constant voltage mode. The voltage level is defined by the Voltage Level attribute in the IviACPwrBase capability group described in section 4

Point 2 The AC power source output has reached the current limit before it reached the voltage level. This is an example of the AC power source operating in the constant current mode. The current limit is defined by the Current Limit attribute in the IviACPwrBase capability group described in section 4.

Point 3, The AC power source output is below the under voltage protection limit. If under voltage protection is enabled, the power source disables the output. This mode is controlled by the *IviACPwrVoltageProtection Extension Group* described in section 9.

Point 4 The AC power source output exceeds the over voltage protection limit. If over voltage protection is enabled, the power source disables the output. This mode is controlled by the *IviACPwrVoltageProtection Extension Group* described in section 9.

Point 5 The AC power source output exceeds the current protection threshold. If current protection is enabled and the AC power source output current has reached or exceeds the current protection threshold for the time specified by the current protection delay, the AC power source disables the output. This mode is controlled by the *IviACPwrCurrentProtection Extension Group* described in section 8.

4.2 IviACPwrBase Attributes

The IviACPwrBase capability group defines the following attributes:

- Current Limit
- Frequency
- Number of Phases
- Output Enabled
- Voltage Level
- Waveform
- Number of Voltage Ranges
- Voltage Range
- Number of Frequency Ranges
- Frequency Range
- Output Phase Item (IVI-COM only)
- Output Phase Name (IVI-COM only)
- Output Phase Count

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in section 16, *IviACPwr Attribute ID* Definitions.

4.2.1 Current Limit

Data Type	Access	Applies to	Coercion	High Level Functions
`ViReal64`	`R/W`	OutputPhase	None	Configure Current Limit

COM Property Name

```
OutputPhases.Item().CurrentLimit
```

COM Enumeration Name

```
N/A
```

C Constant Name

```
IVIACPWR_ATTR_CURRENT_LIMIT
```

Description

Specifies the output current limit. The units are Amps. The RMS behavior of this attribute is vendor specific.

4.2.2 Frequency

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

COM Property Name

```
OutputPhases.Frequency
```

COM Enumeration Name

```
N/A
```

C Constant Name

```
IVIACPWR_ATTR_FREQUENCY
```

Description

Specifies the frequency the AC output voltage waveform. The units are Hertz.

4.2.3 Number of Phases

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

COM Property Name

```
OutputPhases.NumPhases
```

COM Enumeration Name

```
N/A
```

C Constant Name

```
IVIACPWR_ATTR_NUM_PHASES
```

Description

The number of physical output phases of the AC power source the user can connect to their load for this phase group (driver session). The number of phases returned by this attribute does not include the special “AllPhases” output phase name used to set all output phases.

AC power sources that support single or multi phase operation can only be configured when opening a new driver session using the “NumPhases” token in the DriverSetup string or in the configuration store. Otherwise, the present number of phases configured in the AC power source is used.

4.2.4 Output Enabled

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

COM Property Name

`OutputPhases.Item().Enabled`

COM Enumeration Name

`N/A`

C Constant Name

`IVIACPWR_ATTR_OUTPUT_ENABLED`

Description

Enables or disables the output of the AC power source. When Output Enabled is set to False. The output of the AC power source is effectively disconnected from the load.

Defined Values

Name	***Description***		
		Language	***Identifier***
True	The output contactor/relay is closed.		
		C	`VI_TRUE`
		COM	`True`
False	The output contactor/relay is open.		
		C	`VI_FALSE`
		COM	`False`

Return Values

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

Completion Codes	Description
Current Protection Tripped	The power source current protection has tripped. Refer to the ResetCurrentProtection function in the Current Protection extension group.
Voltage Protection Tripped	The power source voltage protection has tripped. Refer to the ResetVoltageProtection function in the Voltage Protection extension group.

Compliance Notes

If current protection or voltage protection has tripped, the driver must not allow setting Output Enabled to True and must return the Current Protection Tripped or Voltage Protection Tripped completion code. Refer to Section 8 : IviACPwrCurrentProtection Extension Group and Section 9 : IviACPwrVoltageProtection Extension Group for more information on resetting a current or voltage protection tripped condition.

4.2.5 Voltage Level

Data Type	Access	Applies to	Coercion	High Level Functions
`ViReal64`	`R/W`	OutputPhase	None	Configure Voltage Level

COM Property Name

```
OutputPhases.Item().VoltageLevel
```

COM Enumeration Name

```
N/A
```

C Constant Name

```
IVIACPWR_ATTR_VOLTAGE_LEVEL
```

Description

Specifies the Line to Neutral output voltage level of the AC power source when operating in constant voltage mode. The units are Volts RMS. The power source will reduce the actual output voltage to limit the output current to the value specified by the Current Limit attribute.

Use the Query Voltage Range Capabilities function to determine the minimum and maximum allowable values for the Voltage Level attribute for a particular voltage range and waveform.

4.2.6 Waveform

Data Type	Access	Applies to	Coercion	High Level Functions
`ViString`	`R/W`	OutputPhase	None	Configure Waveform

COM Property Name

```
OutputPhases.Item().Waveform
```

COM Enumeration Name

```
N/A
```

C Constant Name

```
IVIACPWR_ATTR_WAVEFORM
```

Description

Specifies the waveform name (refer to the *IviACPwrArbWaveform* capability group) used to generate output voltage.

Compliance Notes

All vendors must support ‘Sine’.

4.2.7 Output Phase Item (IVI-COM only)

Data Type	Access	Applies to	Coercion	High Level Functions
`IIviACPwrOutputPhase`	`RO`	OutputPhase	None	

COM Property Name

```
OutputPhases.Item(BSTR Name);
```

COM Enumeration Name

```
N/A
```

C Constant Name

```
N/A
```

Description

Gets an interface pointer for an interface to control a particular Output Phase.

Return Values

If the IVI-COM driver cannot recognize the Name parameter, it returns an Unknown Capability Name completion code as described in IVI-3.2: Inherent Capabilities Specification, Section 9.3.

4.2.8 Output Phase Name (IVI-COM only)

Data Type	Access	Applies to	Coercion	High Level Functions
`ViString`	`RO`	OutputPhase	None	

COM Property Name

```
OutputPhases.Name ([in] LONG Index)
```

COM Enumeration Name

```
N/A
```

C Constant Name

```
N/A
```

Description

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

4.2.9 Output Phase Count

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

COM Property Name

```
OutputPhases.Count
```

COM Enumeration Name

```
N/A
```

C Constant Name

```
IVIACPWR_ATTR_OUTPUT_PHASE_COUNT
```

Description

Returns the number of output phase names supported by the power source.

4.2.10 Number of Voltage Ranges

Data Type	Access	Applies to	Coercion	High Level Functions
`ViInt32`	`RO`	OutputPhase	None	None

COM Property Name

```
OutputPhases.Item().NumVoltageRanges
```

COM Enumeration Name

```
N/A
```

C Constant Name

```
IVIACPWR_ATTR_NUM_VOLTAGE_RANGES
```

Description

The number of output voltage ranges supported by the AC power source. Use the `IVIACPWR_ATTR_VOLTAGE_RANGE` attribute to read or change the output voltage range.

Compliance Notes

All vendors must support at least 1 voltage range.

4.2.11 Voltage Range

Data Type	Access	Applies to	Coercion	High Level Functions
`ViReal64`	`R/W`	OutputPhase	Up	Configure Voltage Range

COM Property Name

```
OutputPhases.Item().VoltageRange
```

COM Enumeration Name

```
N/A
```

C Constant Name

```
IVIACPWR_ATTR_VOLTAGE_RANGE
```

Description

Sets or returns the voltage range of the AC power source based on a specified maximum voltage. The driver coerces this value to the upper limit of the lowest range that supports that voltage.

Use the Query Voltage Range Capabilities function to determine the minimum and maximum RMS voltages supported by a particular range and waveform.

Compliance Notes

Changing the voltage range may require that the output be disabled. Refer to *Appendix A.5: Operations that Require the Output be Disabled* for more information.

4.2.12 Number of Frequency Ranges

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

COM Property Name

```
OutputPhases.NumFrequencyRanges
```

COM Enumeration Name

```
N/A
```

C Constant Name

```
IVIACPWR_ATTR_NUM_FREQUENCY_RANGES
```

Description

The number of output frequency ranges supported by the AC power source. Use the `IVIACPWR_ATTR_FREQUENCY_RANGE` attribute to read or change the output frequency range.

Compliance Notes

All vendors must support at least 1 frequency range.

4.2.13 Frequency Range

Data Type	Access	Applies to	Coercion	High Level Functions
`ViReal64`	`R/W`	N/A	Up	Configure Frequency Range

COM Property Name

```
OutputPhases.FrequencyRange
```

COM Enumeration Name

```
N/A
```

C Constant Name

```
IVIACPWR_ATTR_FREQUENCY_RANGE
```

Description

Sets or returns the frequency range of the AC power source based on a specified maximum frequency. The driver coerces this value to the upper limit of the lowest range that supports that frequency.

Use the Query Frequency Range Capabilities function to determine the minimum and maximum frequency supported by a particular range.

Compliance Notes

Changing the frequency range may require that output power to be disabled. Refer to *Appendix A.5: Operations that Require the Output be Disabled* for more information.

4.3 IviACPwrBase Functions

The IviACPwrBase capability group defines the following functions:

- Configure Current Limit (IVI-C only)
- Configure Output Enabled (IVI-C only)
- Configure Voltage Level (IVI-C only)
- Configure Voltage Range (IVI-C only)
- Configure Frequency (IVI-C only)
- Configure Frequency Range (IVI-C only)
- Configure Waveform (IVI-C only)
- Query Voltage Range Capabilities
- Query Frequency Range Capabilities
- Get Output Phase Name (IVI-C only)

This section describes the behavior and requirements of each function.

4.3.1 Configure Current Limit (IVI-C only)

Description

Configures the current limit. It specifies the output current limit value for constant current mode.

COM Method Prototype

```
N/A
```

(use the `OutputPhases.Item().CurrentLimit` property)

C Prototype

```
ViStatus IviACPwr_ConfigureCurrentLimit (ViSession Vi,
                                         ViConstString PhaseName,
                                         ViReal64 Limit);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle	`ViSession`
`PhaseName`	The name of the output on which to configure the current limit.	`ViConstString`
`Limit`	Specifies the output current limit. The driver uses this value to set the Current Limit attribute. See the attribute description for more details.	`ViReal64`

Return Values

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

4.3.2 Configure Output Enabled (IVI-C only)

Description

Enables/Disables the output contactor/relay on the power source output.

COM Method Prototype

```
N/A
```

(use the `OutputPhases.Item().Enabled` property)

C Prototype

```
ViStatus IviACPwr_ConfigureOutputEnabled (ViSession Vi,
                                          ViConstString PhaseName,
                                          ViBoolean Enabled);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle	`ViSession`
`PhaseName`	The name of the output to enable or disable.	`ViConstString`
`Enabled`	Specifies whether the signal the power source produces appears at the output connector. The driver uses this value to set the Output Enabled attribute. See the attribute description for more details.	`ViBoolean`

Return Values

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

4.3.3 Configure Voltage Level (IVI-C only)

Description

Configures the output Line to Neutral voltage level of the AC power source.

COM Method Prototype

```
N/A
```

(use the `OutputPhases.Item().VoltageLevel` property)

C Prototype

```
ViStatus IviACPwr_ConfigureVoltageLevel (ViSession Vi,
                                         ViConstString PhaseName,
                                         ViReal64 VoltageLevel);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle	`ViSession`
`PhaseName`	The name of the output on which to configure the voltage level.	`ViConstString`
`VoltageLevel`	Specifies the output voltage level of the power source. The driver uses this value to set the Voltage Level attribute. See the attribute description for more details.	`ViReal64`

Return Values

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

4.3.4 Configure Voltage Range (IVI-C only)

Description

Configures the voltage range of the AC power source.

COM Method Prototype

```
N/A
```

(use the `OutputPhases.Item().VoltageRange` property)

C Prototype

```
ViStatus IviACPwr_ConfigureVoltageRange (ViSession Vi,
                                         ViConstString PhaseName,
                                         ViReal64 Range);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle	`ViSession`
`PhaseName`	The name of the output on which to configure the voltage range.	`ViConstString`
`Range`	Specifies the voltage range of the power source. The driver uses this value to set the Voltage Range attribute. See the attribute description for more details.	`ViReal64`

Return Values

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

4.3.5 Configure Frequency (IVI-C only)

Description

Configures the frequency of a single phase power source or all phases of a multi-phase power source.

COM Method Prototype

```
N/A
```

(use the `OutputPhases.Frequency` property)

C Prototype

```
ViStatus IviACPwr_ConfigureFrequency (ViSession Vi,
                                      ViReal64 Frequency);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle	`ViSession`
`Frequency`	Specifies the output frequency of the power source. The driver uses this value to set the Frequency attribute. See the attribute description for more details.	`ViReal64`

Return Values

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

4.3.6 Configure Frequency Range (IVI-C only)

Description

Configures the frequency range.

COM Method Prototype

```
N/A
```

(use the `OutputPhases.FrequencyRange` property)

C Prototype

```
ViStatus IviACPwr_ConfigureFrequencyRange (ViSession Vi,
                                           ViReal64 Range);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle	`ViSession`
`Range`	Specifies the frequency range of the power source. The driver uses this value to set the Frequency Range attribute. See the attribute description for more details.	`ViReal64`

Return Values

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

4.3.7 Configure Waveform (IVI-C only)

Description

Configures the waveform name used to generate output voltage.

COM Method Prototype

```
N/A
```

(use the `OutputPhases.Item().Waveform` property)

C Prototype

```
ViStatus IviACPwr_ConfigureWaveform (ViSession Vi,
                                     ViConstString PhaseName,
                                     ViConstString WaveformName);
```

Parameters

Inputs	**Description**	**Base Type**
`Vi`	Instrument handle	`ViSession`
`PhaseName`	The name of the output on which to configure the waveform.	`ViConstString`
`WaveformName`	Specifies the waveform name (refer to the *IviACPwrArbWaveform* capability group) used to generate output voltage on the specified phase. The driver uses this value to set the Waveform attribute. See the attribute description for more details.	`ViConstString`

Return Values

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

Completion Codes	**Description**
Invalid Waveform Name	The waveform name is not currently stored in the AC power source

Compliance Notes

All vendors shall support a waveform name of ‘Sine’.

4.3.8 Query Voltage Range Capabilities

Description

Returns the minimum and maximum RMS voltage allowed for the user specified voltage range and waveform. The value assigned to the Voltage Level attribute must not be outside these minimum and maximum values. The returned voltages assume no load on the output.

Refer to section *A.6 Waveform RMS and Peak Voltage* for more information on how waveforms can affect the maximum voltage the power source can generate.

COM Method Prototype

```
HRESULT OutputPhases.Item().QueryVoltageRangeCapabilities([in] LONG Range,
                                        [in] BSTR WaveformName,
                                        [in, out] double *MinVoltage,
                                        [in, out] double *MaxVoltage);
```

C Prototype

```
ViStatus IviACPwr_QueryVoltageRangeCapabilities (ViSession Vi,
                                        ViConstString PhaseName,
                                        ViInt32 Range,
                                        ViConstString WaveformName,
                                        ViReal64 *MinVoltage,
                                        ViReal64 *MaxVoltage);
```

Parameters

Inputs	**Description**	**Base Type**
`Vi`	Instrument handle	`ViSession`
`PhaseName`	The name of the output on which to query the voltage range.	`ViConstString`
`Range`	The voltage range to query. This parameter must be between 1 and the value reported by the `IVIACPWR_ATTR_NUM_VOLTAGE_RANGES` attribute	`ViInt32`
`WaveformName`	The name of the waveform to query the range. If the waveform name parameter is VI_NULL, the range for a 'Sine' waveform is returned.	`ViConstString`

Outputs	**Description**	**Base Type**
`MinVoltage`	Returns the minimum RMS voltage supported by the specified range and waveform.	`ViReal64`
`MaxVoltage`	Returns the maximum RMS voltage supported by the specified range and waveform.	`ViReal64`

Return Values

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
Waveform Not Found	The waveform name is not currently stored in the AC Power Source.

Compliance Notes

All vendors must support a waveform name of ‘Sine’ and VI_NULL.

4.3.9 Query Frequency Range Capabilities

Description

Returns the minimum and maximum frequency in hertz allowed for the user specified frequency range. The value assigned to the Frequency attribute must not be outside these minimum and maximum values.

COM Method Prototype

```
HRESULT OutputPhases.QueryFrequencyRangeCapabilities([in] LONG range,
                                                [in, out] double *MinFrequency,
                                                [in, out] double *MaxFrequency);
```

C Prototype

```
ViStatus IviACPwr_QueryFrequencyRangeCapabilities (ViSession Vi,
                                                ViInt32 range,
                                                ViReal64 *MinFrequency,
                                                ViReal64 *MaxFrequency);
```

Parameters

Inputs	**Description**	**Base Type**
`Vi`	Instrument handle	`ViSession`
`range`	The frequency range to query. This parameter must be between 1 and the value reported by the `IVIACPWR_ATTR_NUM_FREQUENCY_RANGES` attribute	`ViInt32`

Outputs	**Description**	**Base Type**
`MinFrequency`	Returns the minimum frequency supported by the specified range.	`ViReal64`
`MaxFrequency`	Returns the maximum frequency supported by the specified range.	`ViReal64`

Return Values

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

4.3.10 Get Output Phase Name (IVI-C only)

Description

Returns the specific driver defined output phase name that corresponds to the one-based index that the user specifies. If the driver defines a qualified output phase name, this function returns the qualified name. If the value that the user passes for the Index parameter is less than one or greater than the value of the Output Phase Count attribute, the function returns an empty string in the Name parameter and returns the Invalid Value error.

COM Method Prototype

```
N/A
```

(use the `OutputPhases.Name()` property)

C Prototype

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

Parameters

Inputs	**Description**	**Base Type**
Vi	Instrument handle	ViSession
Index	A one-based index that defines which name to return.	ViInt32
NameBufferSize	The number of bytes in the ViChar array that the user specifies for the Name parameter.	ViInt32

Outputs	**Description**	**Base Type**
Name	The buffer into which the function returns the name that corresponds to the index the user specifies. The caller may pass VI_NULL for this parameter if the NameBufferSize parameter is 0.	ViChar[]

Return Values

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

Compliance Notes

1. For an instrument with only one Output Phase, that is the Output Phase Count attribute is one, the driver may return an empty string.
2. Refer to Section 3.1.2, Additional Compliance Rules for C Functions with ViChar Array Output Parameters, in IVI-3.2 Inherent Capabilities Specification, Section 3.1.2.1 for rules regarding the NameBufferSize and Name parameters.

4.4 IviACPwrBase Behavior Model

After the user calls the Initialize or Reset functions, the driver must immediately disable output power and set the Output Enabled attribute to False for all output phases. The driver must then set all extension capability groups to the disabled state as described in section A.2. It is the responsibility of the user to determine if the attribute values in the IviACPwrBase capability group are appropriate or safe for their load prior to setting the Output Enabled attribute to True.

All changes to the IviACPwrBase functions and attributes take place immediately.

Certain AC power sources may require that the output power be disabled in order to change the voltage or frequency range. Refer to *Appendix A.5: Operations that Require the Output be Disabled* for more information.

5 IviACPwrMeasurement Extension Group

5.1 IviACPwrMeasurement Overview

The IviACPwrMeasurement extension group defines extensions for AC Power Sources capable of measuring various output signals such as voltage and current.

5.2 IviACPwrMeasurement Functions

The IviACPwrMeasurement extension defines the following functions:

- Initiate Measurement
- Fetch Measurement
- Fetch Measurement Array

The Initiate Measurement function initiates a measurement of one or more measurement groups. To minimize the time to perform measurements and to correlate them to the same output cycle, measurement types are divided into groups. Ideally, for each call to the Initiate Measurement function, all measurement types for all phases for the groups specified are measured on the same cycle of the output, but individual vendors may implement this differently. The exact behavior must be documented by the instrument driver.

The Fetch Measurement function returns the measurement result of a previous call to the Initiate Measurement function. The Fetch Measurement function returns a single value of the specified measurement type. Call Fetch Measurement multiple times to retrieve values for the measurement types for each output phase in that measurement group.

The Fetch Measurement Array function returns an array of measured values like harmonics or waveform of a previous call to the Initiate Measurement function. The size of the user allocated measurement array is passed to the function and the function returns the actual number of values stored in the array. If you pass VI_NULL for the measurement pointer or 0 for ArraySize, the function will not perform the measurement but will return the array size needed (in the ActualSize parameter) to store the results for that measurement type. In cases where there are more values to return than the array size you specified, the function will fill the array and return the additional array size needed in the function result.

5.2.1 Initiate Measurement

Description

Initiates a measurement of one or more measurement groups for all output phases. You can specify multiple groups by ORing the group identifiers together.

COM Method Prototype

```
HRESULT OutputPhases.InitiateMeasurement (
                           [in] long Group);
```

C Prototype

```
ViStatus IviACPwr_InitiateMeasurement (ViSession Vi,
                           ViInt32 Group);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle	`ViSession`
`Group`	The group(s) to measure.	`ViInt32`

Defined Values for Group Parameter

Name	***Description***		
		Language	***Identifier***
Base	Measure all items in the Base measurement group		
		C	`IVIACPWR_VAL_MEASUREMENT_GROUP_BASE`
		COM	`IviACPwrMeasurementGroupBase`
Harmonic	Measure all items in the Harmonic measurement group		
		C	`IVIACPWR_VAL_MEASUREMENT_GROUP_HARMONIC,`
		COM	`IviACPwrMeasurementGroupHarmonic`
Distortion	Measure all items in the Distortion measurement group		
		C	`IVIACPWR_VAL_MEASUREMENT_GROUP_DISTORTION,`
		COM	`IviACPwrMeasurementGroupDistortion`
Waveform	Measure all items in the Waveform measurement group		
		C	`IVIACPWR_VAL_MEASUREMENT_GROUP_WAVEFORM,`
		COM	`IviACPwrMeasurementGroupWaveform`

Return Values

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

Completion Codes	Description
Group Not Supported	The AC power source does not support the requested measurement group(s).

5.2.2 Fetch Measurement

Description

Returns the value measured by a previous call to the Initiate Measurement function. The Fetch Measurement function returns a single measured value for the requested measurement type. Call this function multiple times to retrieve the available measurement types and output phases. The available measurement types are divided into measurement groups. Use the Initiate Measurement function to measure fresh values for that group.

COM Method Prototype

```
HRESULT OutputPhases.Item().FetchMeasurement([in] IviACPwrMeasurementTypeEnum
                           MeasurementType,
                           [out] double *Measurement);
```

C Prototype

```
ViStatus IviACPwr_FetchMeasurement (ViSession Vi,
                           ViConstString PhaseName,
                           ViInt32 MeasurementType,
                           ViReal64 *Measurement);
```

Parameters

Inputs	**Description**	**Base Type**
Vi	Instrument handle	ViSession
PhaseName	The name of the output phase to return the measurement.	ViConstString
MeasurementType	The type of measurement to return.	ViInt32

Outputs	**Description**	**Base Type**
Measurement	Returns the measured value.	ViReal64

Defined Values for MeasurementType Parameter – Base Group

Name	***Description***		
		Language	***Identifier***
Voltage RMS L-N	Voltage RMS line to neutral is measured.		
		C	`IVIACPWR_VAL_MEASURE_VOLTAGE_RMS_LN`
		COM	`IviACPwrMeasureVoltageRMS`
Current RMS	Current RMS is measured.		
		C	`IVIACPWR_VAL_MEASURE_CURRENT_RMS`
		COM	`IviACPwrMeasureCurrentRMS`

Frequency	Frequency is measured.		
		C	IVIACPWR_VAL_MEASURE_FREQUENCY
		COM	IviACPwrMeasureFrequency
Voltage DC	Voltage level is measured when in DC Only Mode.		
		C	IVIACPWR_VAL_MEASURE_VOLTAGE_DC
		COM	IviACPwrMeasureVoltageDC
Current DC	Current level is measured when in DC Only Mode.		
		C	IVIACPWR_VAL_MEASURE_CURRENT_DC
		COM	IviACPwrMeasureCurrentDC
Power Factor	Power Factor is measured.		
		C	IVIACPWR_VAL_MEASURE_POWER_FACTOR
		COM	IviACPwrMeasurePowerFactor
Crest Factor	Crest Factor is measured.		
		C	IVIACPWR_VAL_MEASURE_CREST_FACTOR
		COM	IviACPwrMeasureCrestFactor
Current Peak	Peak Current is measured.		
		C	IVIACPWR_VAL_MEASURE_CURRENT_PEAK
		COM	IviACPwrMeasureCurrentPeak
Power VA	Power is measured in Volts-Amps.		
		C	IVIACPWR_VAL_MEASURE_POWER_VA
		COM	IviACPwrMeasurePowerVA
Real Power	Real Power is measured in Watts.		
		C	IVIACPWR_VAL_MEASURE_POWER_REAL
		COM	IviACPwrMeasurePowerReal
Power DC	Power is measured in Watts when in DC Only Mode.		
		C	IVIACPWR_VAL_MEASURE_POWER_DC
		COM	IviACPwrMeasurePowerDC
Phase Angle	Phase Angle is measured in Degrees.		
		C	IVIACPWR_VAL_MEASURE_PHASE_ANGLE
		COM	IviACPwrMeasurePhaseAngle
Voltage RMS L-L	Voltage RMS (line-to-line) is measured.		
		C	IVIACPWR_VAL_MEASURE_VOLTAGE_RMS_LL
		COM	IviACPwrMeasureVoltageRMSLineToLine

Defined Values for MeasurementType Parameter – Distortion Group

Name	***Description***		
		Language	***Identifier***
Current OHD	Current Odd Harmonics Distortion is measured.		
		C	`IVIACPWR_VAL_MEASURE_CURRENT_OHD`
		COM	`IviACPwrMeasureCurrentOHD`
Current EHD	Current Even Harmonics Distortion is measured		
		C	`IVIACPWR_VAL_MEASURE_CURRENT_EHD`
		COM	`IviACPwrMeasureCurrentEHD`
Current THD	Current Total (Odd+Even) Harmonic Distortion is measured		
		C	`IVIACPWR_VAL_MEASURE_CURRENT_THD`
		COM	`IviACPwrMeasureCurrentTHD`
Voltage OHD	Voltage Odd Harmonics Distortion is measured.		
		C	`IVIACPWR_VAL_MEASURE_VOLTAGE_OHD`
		COM	`IviACPwrMeasureVoltageOHD`
Voltage EHD	Voltage Even Harmonics Distortion is measured		
		C	`IVIACPWR_VAL_MEASURE_VOLTAGE_EHD`
		COM	`IviACPwrMeasureVoltageEHD`
Voltage THD	Voltage Total (Odd+Even) Harmonic Distortion is measured		
		C	`IVIACPWR_VAL_MEASURE_VOLTAGE_THD`
		COM	`IviACPwrMeasureVoltageTHD`

Return Values

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

Completion Codes	**Description**
Measurement Not Initiated	You must use the Initiate Measurement function for that group prior to calling Fetch Measurement for that measurement type.
Measurement Not Supported	The AC Power source does not support the requested measurement type.

Compliance Notes

1. If an IVI-C class driver defines additional values for the MeasurementType parameter, the actual values shall be greater than or equal to IVIACPWR_VAL_MEASURE_CLASS_EXT_BASE and less than IVIACPWR_VAL_MEASURE_SPECIFIC_EXT_BASE.
2. IviACPwr specific drivers that implement the IviACPwrMeasurement Extension Group shall implement at least one of the following measurement types:
    - Voltage RMS L-N
    - Current RMS

5.2.1 Fetch Measurement Array

Description

Returns the values measured by a previous call to the Initiate Measurement function. The Fetch Measurement Array function returns an array of measured values of the requested measurement type. Call this function multiple times to retrieve the available measurement types and output phases. The available measurement types are divided into measurement groups. Use the Initiate Measurement function to measure fresh values for that group.

COM Method Prototype

```
HRESULT OutputPhases.Item().FetchMeasurementArray([in]
                           IviACPwrMeasurementTypeArrayEnum MeasurementType,
                           [in, out] SAFEARRAY(double) *Measurement);
```

C Prototype

```
ViStatus IviACPwr_FetchMeasurementArray (ViSession Vi,
                           ViConstString PhaseName,
                           ViInt32 MeasurementType,
                           ViInt32 MeasurementBufferSize,
                           ViReal64 Measurement[],
                           ViInt32 *MeasurementActualSize);
```

Parameters

Inputs	**Description**	**Base Type**
`Vi`	Instrument handle	`ViSession`
PhaseName	The name of the output phase to return the measurement	ViConstString
MeasurementType	The type of measurement to return.	ViInt32
MeasurementBufferSize	The size of the user allocated IVI-C measurement array.	ViInt32

Outputs	**Description**	**Base Type**
Measurement	User allocated (IVI-C) or driver allocated (IVI-COM) array to store the measured values.	ViReal64
MeasurementActualSize	The number of values stored in the measurement array.	ViInt32

Defined Values for MeasurementType Parameter – Harmonic Group

Name	***Description***		
		Language	***Identifier***
Array Current Harmonic Phase	Current Harmonic Phase Angles are measured in degrees		
		C	`IVIACPWR_VAL_MEASURE_ARRAY_CURRENT_HARMONIC_PHASE`
		COM	`IviACPwrMeasureArrayCurrentHarmonicPhase`
Array Current Harmonic Magnitude Absolute	Current Harmonic Magnitudes are measured as an absolute value in RMS Amps		
		C	`IVIACPWR_VAL_MEASURE_ARRAY_CURRENT_HARMONIC_ABS`
		COM	`IviACPwrMeasureArrayCurrentHarmonicAbs`
Array Current Harmonic Magnitude Percent Fundamental	Current Harmonic Magnitudes are measured as a Percent of fundamental.		
		C	`IVIACPWR_VAL_MEASURE_ARRAY_CURRENT HARMONIC_PCT`
		COM	`IviACPwrMeasureArrayCurrentHarmonicPct`
Array Voltage Harmonic Phase	Voltage Harmonic Phase Angles are measured in Degrees.		
		C	`IVIACPWR_VAL_MEASURE_ARRAY_VOLTAGE_HARMONIC_PHASE`
		COM	`IviACPwrMeasureArrayVoltageHarmonicPhase`
Array Voltage Harmonic Magnitude Absolute	Voltage Harmonic Magnitudes are measured as an absolute value in RMS Volts.		
		C	`IVIACPWR_VAL_MEASURE_ARRAY_VOLTAGE_HARMONIC_ABS`
		COM	`IviACPwrMeasureArrayVoltageHarmonicAbs`
Array Voltage Harmonic Magnitude Percent Fundamental	Voltage Harmonic Magnitudes are measured as a percent of the fundamental.		
		C	`IVIACPWR_VAL_MEASURE_ARRAY_VOLTAGE_HARMONIC_PCT`
		COM	`IviACPwrMeasureArrayVoltageHarmonicPct`

Defined Values for MeasurementType Parameter – Waveform Group

Name	***Description***		
		Language	***Identifier***
Array Current Cycle	Single cycle of the current waveform in Amps		
		C	`IVIACPWR_VAL_MEASURE_ARRAY_CURRENT_CYCLE`
		COM	`IviACPwrMeasureArrayCurrentCycle`
Array Voltage Cycle	Single cycle of the Voltage Waveform in Volts		
		C	`IVIACPWR_VAL_MEASURE_ARRAY_VOLTAGE_CYCLE`
		COM	`IviACPwrMeasureArrayVoltageCycle`

Return Values

The *IVI-3.2: Inherent Capabilities Specification* defines general status codes that this function can return. For IVI-C, in cases where there are more values to return than the user specified array size, the function will fill the array and return the additional array size needed in the function result.

Completion Codes	Description
Measurement Not Initiated	You must use the Initiate function for that group prior to calling Fetch for that measurement type.
Measurement Not Supported	The AC Power source does not support the requested measurement type.

Compliance Notes

1. If an IVI-C class driver defines additional values for the MeasurementType parameter, the actual values shall be greater than or equal to IVIACPWR_VAL_MEASURE_CLASS_EXT_BASE and less than IVIACPWR_VAL_MEASURE_SPECIFIC_EXT_BASE.
2. IviACPwr specific drivers that do not support any measurement types for the Fetch Measurement Array function should return the Measurement Not Supported status code.

5.3 IviACPwrMeasurement Behavior Model

The IviACPwrMeasurement Extension Group follows the behavior model of the IviACPwrBase capability group. The only modification to the behavior model from the IviACPwrBase capability group is the ability to measure output signals.

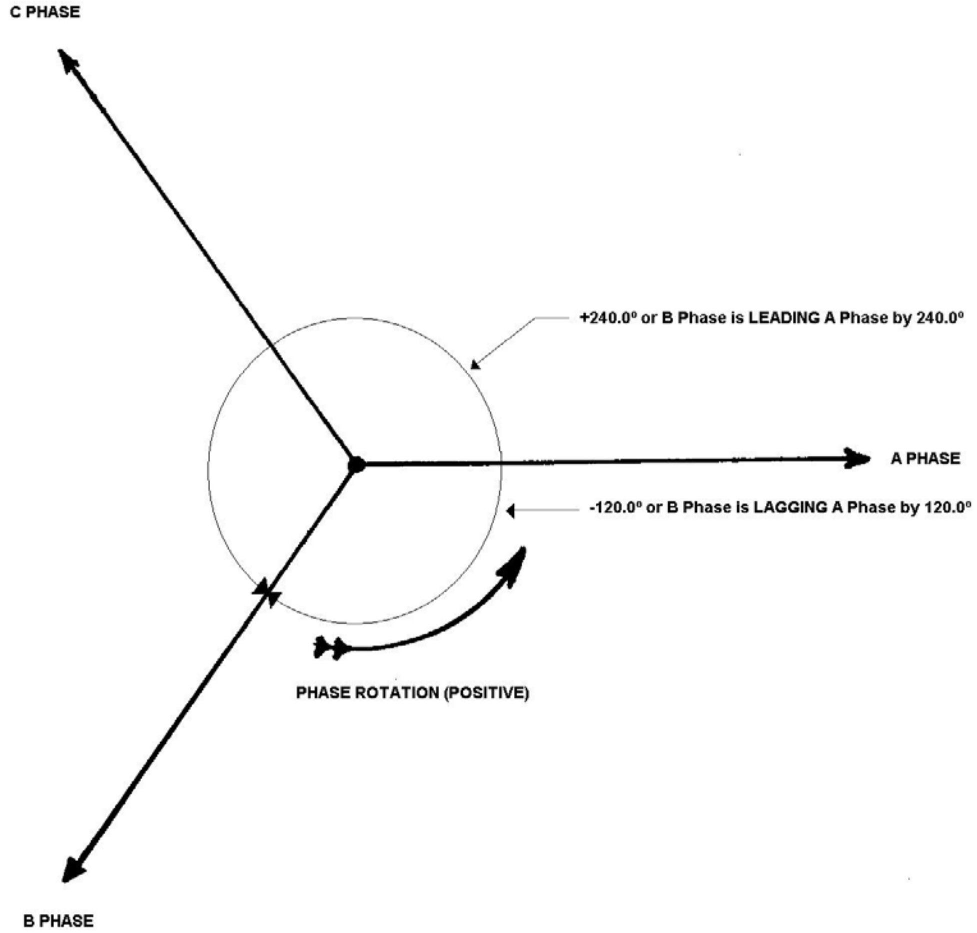
6 IviACPwrPhase Extension Group

6.1 IviACPwrPhase Overview

The IviACPwrPhase extension group defines extensions for AC Power Sources which support multiple phase operation. The Phase angle is given in reference to Phase A (refer to section 2.3.1 for information on how output phases map to phase letters). The functions and attributes in the IviACPwrPhase extension group will allow for positive leading and negative lagging phase angles relative to Phase A. If an instrument only allows positive phase angles or defines positive angles as lagging relative to phase A, the driver must translate the user value to the instrument value.

Example:

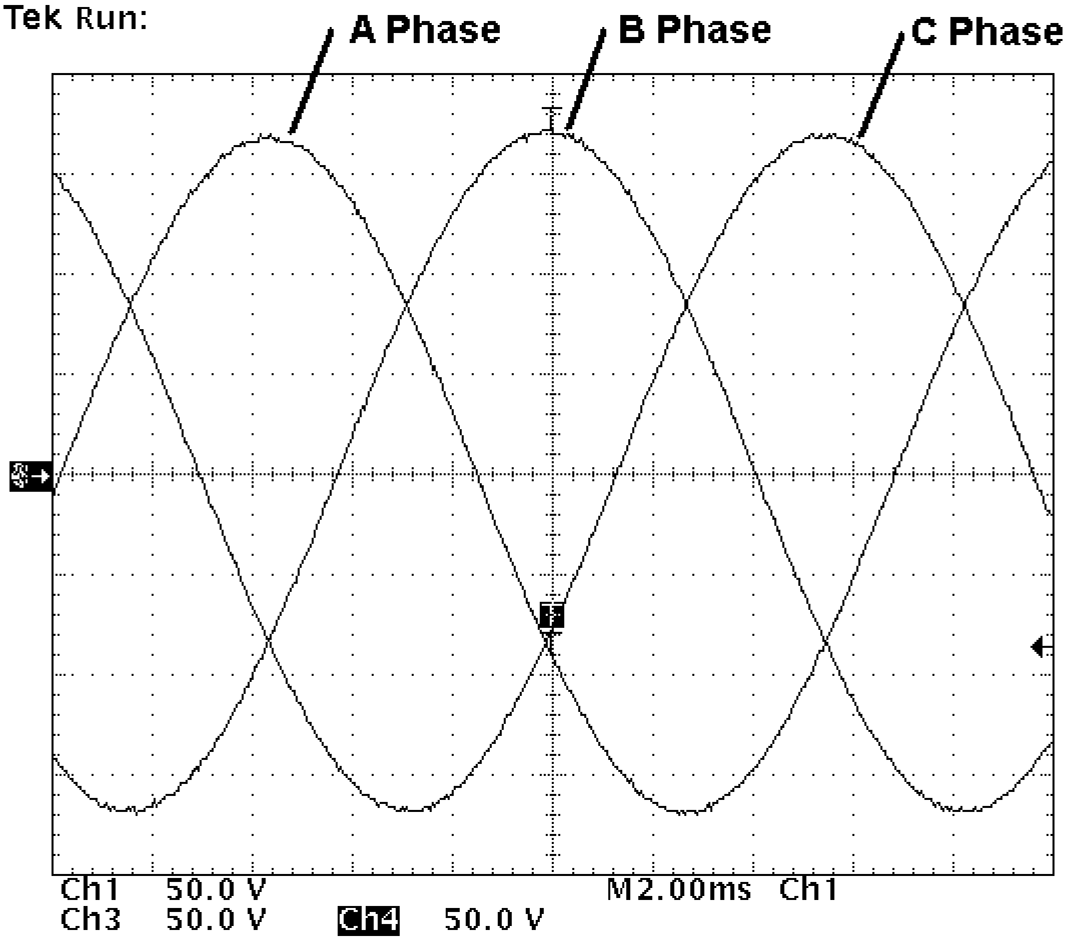
> Phase B set to negative 120 degrees lagging relative to Phase A translates to positive 240 degrees leading relative to Phase A.


C PHASE
+240.0° or B Phase is LEADING A Phase by 240.0°
A PHASE
-120.0° or B Phase is LAGGING A Phase by 120.0°
PHASE ROTATION (POSITIVE)
B PHASE

PHASOR DIAGRAM SHOWING LEADING vs. LAGGING

PHASE ANGLE FOR B PHASE OUTPUT

The definition of phase rotation or phase sequence is that the correct phase sequence will be A-B-C in all cases. This sequence is shown in oscilloscope capture:



PHASE ROTATION OR PHASE SEQUENCE

AS DISPLAYED ON AN OSCILLOSCOPE

6.2 IviACPwrPhase Attributes

The IviACPwrPhase extension group defines the following Attributes:

- Phase Angle

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in section 16, *IviACPwr Attribute ID Definitions*.

6.2.1 Phase Angle

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	OutputPhase	None	Configure Phase Angle

COM Property Name

```
OutputPhases.Item().PhaseAngle
```

COM Enumeration Name

```
N/A
```

C Constant Name

```
IVIACPWR_ATTR_PHASE_ANGLE
```

Description

Specifies the Phase Angle of the output phase in relation to Phase A. The units are degrees.

6.3 IviACPwrPhase Functions

The IviACPwrPhase extension defines the following function:

- Configure Phase Angle (IVI-C only)

This section describes the behavior and requirements of the function(s).

6.3.1 Configure Phase Angle (IVI-C only)

Description

Configures the phase angle of a multi-phase power source. It specifies the phase angle of each phase of the power source as it relates to the Phase A. The functions and attributes will allow for positive leading and negative lagging phase angles relative to Phase A.

COM Method Prototype

```
N/A
```

(use the `OutputPhases.Item().PhaseAngle` property)

C Prototype

```
ViStatus IviACPwr_ConfigurePhaseAngle (ViSession Vi,
                                       ViConstString PhaseName,
                                       ViReal64 PhaseAngle);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle	`ViSession`
`PhaseName`	The name of the output on which to configure the phase angle.	`ViConstString`
`PhaseAngle`	Sets the phase angle of the specified output phase of a multiphase power source relative to the Phase A. The driver uses this value to set the Phase Angle attribute. See the attribute description for more details.	`ViReal64`

Return Values

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

6.4 IviACPwrPhase Behavior Model

The IviACPwrPhase Extension Group follows the behavior model of the IviACPwrBase capability group. The only modification to the behavior from the IviACPwrBase capability group is the ability to configure IviACPwrPhase capabilities.

7 IviACPwrExternalSync Extension Group

7.1 IviACPwrExternalSync Overview

The IviACPwrExternalSync extension group defines extensions for AC Power Sources with the ability to use an external signal to synchronize the power source output frequency. These attributes and functions allow you to enable external synchronization, configure the phase offset and query the synchronization status of the power source with the external signal.

7.2 IviACPwrExternalSync Attributes

The IviACPwrExternalSync extension group defines the following Attributes:

- External Synchronization Enabled
- External Synchronization Phase Offset
- External Synchronization Locked

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in section 16, *IviACPwr Attribute ID Definitions*.

7.2.1 External Synchronization Enabled

Data Type	**Access**	**Applies to**	**Coercion**	**High Level Functions**
`ViBoolean`	`R/W`	N/A	None	Configure External Sync

COM Property Name

```
OutputPhases.ExternalSync.Enabled
```

COM Enumeration Name

```
N/A
```

C Constant Name

```
IVIACPWR_ATTR_EXTERNAL_SYNC_ENABLED
```

Description

Specifies whether to enable or disable synchronization to an external signal.

Defined Values

Name	***Description***		
		Language	***Identifier***
True	Enable synchronization to external signal.		
		C	`VI_TRUE`
		COM	`True`
False	Disable synchronization to external signal.		
		C	`VI_FALSE`
		COM	`False`

Compliance Notes

Instrument drivers shall support the value False.

7.2.2 External Synchronization Phase Offset

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

COM Property Name

```
OutputPhases.ExternalSync.PhaseOffset
```

COM Enumeration Name

```
N/A
```

C Constant Name

```
IVIACPWR_ATTR_EXTERNAL_SYNC_PHASE_OFFSET
```

Description

Specifies the phase angle offset (degrees) of the power sources output relative to the external synchronization signal.

A positive external sync phase offset value means the output of a single phase power source, or phase A of a multiphase power source, leads the external sync signal and a negative phase offset value means it lags the external sync signal. For multiphase power sources, the phase relationship between output phases is always exclusively determined by the IviACPwrPhase extension group.

7.2.3 External Synchronization Locked

Data Type	Access	Applies to	Coercion	High Level Functions
`ViBoolean`	`RO`	NA	None	Query External Synchronization Locked (IVI-C only)

COM Property Name

```
OutputPhases.ExternalSync.Locked
```

COM Enumeration Name

```
N/A
```

C Constant Name

```
IVIACPWR_ATTR_EXTERNAL_SYNC_LOCKED
```

Description

Indicates if the AC power source is locked to the external synchronization signal.

Defined Values

Name	***Description***		
		Language	***Identifier***
True	The AC power source is locked to the external synchronization signal.		
		C	`VI_TRUE`
		COM	`True`
False	The AC power source is not locked to the external synchronization signal.		
		C	`VI_FALSE`
		COM	`False`

7.3 IviACPwrExternalSync Functions

The IviACPwrExternalSync extension defines the following functions:

- Configure External Synchronization
- Query External Synchronization Locked (IVI-C only)

This section describes the behavior and requirements of the functions.

7.3.1 Configure External Synchronization

Description

Specifies whether to enable or disable external synchronization and the phase offset relative to the external synchronization signal.

COM Method Prototype

```
HRESULT OutputPhases.ExternalSync.Configure (
                            [in] BOOL Enabled
                           [in] double PhaseOffset);
```

C Prototype

```
ViStatus IviACPwr_ConfigureExternalSync (ViSession Vi,
                                         ViBoolean Enabled,
                                         ViReal64 PhaseOffset);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle	`ViSession`
`Enabled`	Specifies if the power source should synchronize its frequency to an external signal. The driver uses this value to set the External Sync Enabled attribute. See the attribute description for more details.	`ViBoolean`
`PhaseOffset`	Sets the phase angle offset of the specified output phase relative to the external synchronization signal. The driver uses this value to set the External Sync Phase Offset attribute. See the attribute description for more details.	`ViReal64`

Return Values

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

7.3.2 Query External Synchronization Locked (IVI-C only)

Description

Returns if the power source is locked to the external synchronization signal.

COM Method Prototype

`N/A`

(use the `OutputPhases.ExternalSync.Locked` property)

C Prototype

```
ViStatus IviACPwr_QueryExternalSyncLocked (ViSession Vi,
                                           ViBoolean *Locked);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle	`ViSession`

Outputs	Description	Base Type
`Locked`	Returns if the power source output is locked with the external synchronization signal.	`ViBoolean`

Defined Values for Locked Parameter

Name	***Description***		
		Language	***Identifier***
True	The power source is presently locked to the external synchronization signal.		
		C	`VI_TRUE`
		COM	`True`
False	The power source is not presently locked to the external synchronization signal.		
		C	`VI_FALSE`
		COM	`False`

Return Values

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

7.4 IviACPwrExternalSync Behavior Model

The IviACPwrExternalSync Extension Group follows the behavior model of the IviACPwrBase capability group. The only modification to the behavior from the IviACPwrBase capability group is the ability to configure IviACPwrExternalSync capabilities.

8 IviACPwrCurrentProtection Extension Group

8.1 IviACPwrCurrentProtection Overview

The IviACPwrCurrentProtection extension group defines extensions for AC Power Sources which allow programmable current protection. A method for resetting the current protection and querying the current protection state of the power source is also provided.

8.2 IviACPwrCurrentProtection Attributes

The IviACPwrCurrentProtection extension group defines the following attributes:

- Current Protection Threshold Value
- Current Protection Delay Value
- Current Protection Enabled
- Current Protection Tripped

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 16, *IviACPwr Attribute ID Definitions.*

8.2.1 Current Protection Threshold

Data Type	Access	Applies to	Coercion	High Level Functions
`ViReal64`	`R/W`	OutputPhase	None	Configure Current Protection

COM Property Name

```
OutputPhases.Item().CurrentProtection.Threshold
```

COM Enumeration Name

```
N/A
```

C Constant Name

```
IVIACPWR_ATTR_CURRENT_PROTECTION_THRESHOLD
```

Description

Specifies the output current threshold that triggers current protection. The units are Amps RMS.

Compliance Notes

When current protection is enabled, changing this attribute may invalidate the Current Limit attribute in IviACPwrBase. In other words, some instruments may share the same value for Current Limit and Current Protection Threshold.

8.2.2 Current Protection Delay

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	OutputPhase	None	Configure Current Protection

COM Property Name

```
OutputPhases.Item().CurrentProtection.Delay
```

COM Enumeration Name

```
N/A
```

C Constant Name

```
IVIACPWR_ATTR_CURRENT_PROTECTION_DELAY
```

Description

Specifies the time the output current must exceed the current protection threshold until current protection mode is triggered. The units are seconds.

8.2.3 Current Protection Enabled

Data Type	Access	Applies to	Coercion	High Level Functions
`ViBoolean`	`R/W`	OutputPhase	None	Configure Current Protection

COM Property Name

```
OutputPhases.Item().CurrentProtection.Enabled
```

COM Enumeration Name

```
N/A
```

C Constant Name

```
IVIACPWR_ATTR_CURRENT_PROTECTION_ENABLED
```

Description

When set to True, current protection mode is triggered when the output current exceeds the current protection threshold for the time specified by the current protection delay.

Defined Values

Name	***Description***		
		Language	***Identifier***
True	Current protection enabled.		
		C	`VI_TRUE`
		COM	`True`
False	Current protection disabled.		
		C	`VI_FALSE`
		COM	`False`

Compliance Notes

Instrument drivers shall support the value False.

8.2.4 Current Protection Tripped

Data Type	Access	Applies to	Coercion	High Level Functions
`ViBoolean`	`RO`	OutputPhase	None	Query Current Protection Tripped (IVI-C only)

COM Property Name

```
OutputPhases.Item().CurrentProtection.Tripped
```

COM Enumeration Name

```
N/A
```

C Constant Name

```
IVIACPWR_ATTR_CURRENT_PROTECTION_TRIPPED
```

Description

Indicates if current protection has tripped. Current protection trips when the output current exceeds the value of the current protection threshold attribute for the time specified by the current protection delay attribute. Use the Reset Current Protection function to reset the current protection state.

Defined Values

Name	***Description***		
		Language	***Identifier***
True	Current protection has tripped.		
		C	`VI_TRUE`
		COM	`True`
False	Current protection has not tripped.		
		C	`VI_FALSE`
		COM	`False`

8.3 IviACPwrCurrentProtection Functions

The IviACPwrCurrentProtection extension group defines the following functions:

- Query Current Protection Tripped (IVI-C only)
- Reset Current Protection
- Configure Current Protection

8.3.1 Query Current Protection Tripped (IVI-C only)

Description

Returns the current protection state of the AC power source.

Current protection trips when the output current exceeds the value of the current protection threshold attribute for the time specified by the current protection delay attribute.

When current protection trips, the power source enters the current protection state and disables the output. The user must call the Reset Current Protection function to reset the current protection state of the AC power source. Once current protection is reset, the user must then set the Output Enabled attribute to True for the AC power source to resume generating output voltage.

COM Method Prototype

`N/A`

(use the `OutputPhases.Item().CurrentProtection.Tripped` property)

C Prototype

```
ViStatus IviACPwr_QueryCurrentProtectionTripped (ViSession Vi,
                                                 ViConstString PhaseName,
                                                 ViBoolean *Tripped);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle	`ViSession`
`PhaseName`	The name of the output to return the current protection status.	`ViConstString`

Outputs	Description	Base Type
`Tripped`	Returns True if current protection has tripped.	`ViBoolean`

Defined Values for Tripped Parameter

Name	***Description***		
		Language	***Identifier***
True	Current protection has tripped.		
		C	`VI_TRUE`
		COM	`True`
False	Current protection has not tripped.		
		C	`VI_FALSE`
		COM	`False`

Return Values

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

8.3.2 Reset Current Protection

Description

Resets the AC power source current protection state after a current protection condition has occurred. The user must also set the Output Enabled attribute in IviACPwrBase to True to resume generating output power.

Use the Query Current Protection Tripped function to determine if current protection has tripped.

COM Method Prototype

```
HRESULT OutputPhases.Item().CurrentProtection.Reset();
```

C Prototype

```
ViStatus IviACPwr_ResetCurrentProtection (ViSession Vi,
                                          ViConstString PhaseName);
```

Parameters

Inputs	**Description**	**Base Type**
Vi	Instrument handle	ViSession
`PhaseName`	The name of the output on which to reset the current protection.	`ViConstString`

Return Values

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

8.3.3 Configure Current Protection

Description

Configures current protection. When the Enabled parameter is True, the power source enters the current protection state (tripped) and disables the output when the output current exceeds the current threshold for the current delay time.

When the Enabled parameter is False, the threshold and delay do not affect the instrument’s behavior and current protection is disabled.

COM Method Prototype

```
HRESULT OutputPhases.Item().CurrentProtection.Configure (
                           [in] BOOL Enabled,
                           [in] DOUBLE Threshold,
                           [in] DOUBLE Delay);
```

C Prototype

```
ViStatus IviACPwr_ConfigureCurrentProtection (ViSession Vi,
                             ViConstString PhaseName,
                             ViBoolean Enabled,
                             ViReal64 Threshold,
                             ViReal64 Delay);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
PhaseName	The name of the output on which to set the current protection values.	ViConstString
Enabled	Enables/disables current protection. The driver uses this value to set the Current Protection Enabled attribute. See the attribute description for more details.	ViBoolean
Threshold	Specifies the current threshold that triggers over current protection. The driver uses this value to set the Current Protection Threshold attribute. See the attribute description for more details.	ViReal64
Delay	Specifies the time the output current must exceed the over current protection threshold until over current protection is triggered. The driver uses this value to set the Current Protection Delay attribute. See the attribute description for more details.	ViReal64

Return Values

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

8.4 IviACPwrCurrentProtection Behavior Model

The IviACPwrCurrentProtection Extension Group follows the behavior model of the IviACPwrBase capability group. The only addition to the behavior model is the driver must set the Output Enabled attribute in the IviACPwrBase capability group to False when a current protection condition occurs.

In order to resume generating output power after current protection trips, the user must call the Reset Current Protection function then set the Output Enabled attribute in IviACPwrBase to True.

It is the responsibility of the user to determine if the circuit is appropriate or safe for resuming generating output voltage after a current protection trips.

9 IviACPwrVoltageProtection Extension Group

9.1 IviACPwrVoltageProtection Overview

The IviACPwrVoltageProtection extension group defines extensions for AC Power Sources which allow programmable voltage protection. A method for resetting the protection and querying the protection state of the power source is also provided. In order for an instrument to be compatible with this extension group, it must support either or both Under Voltage or Over Voltage protection.

9.2 IviACPwrVoltageProtection Attributes

The IviACPwrCurrentProtection extension group defines the following attributes:

- Under Voltage Limit
- Over Voltage Limit
- Over Voltage Enabled
- Under Voltage Enabled
- Voltage Protection Tripped

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 16, *IviACPwr Attribute ID Definitions.*

9.2.1 Under Voltage Limit

Data Type	Access	Applies to	Coercion	High Level Functions
`ViReal64`	`R/W`	OutputPhase	None	Configure Voltage Protection

COM Property Name

```
OutputPhases.Item().VoltageProtection.UnderLimit
```

COM Enumeration Name

```
N/A
```

C Constant Name

```
IVIACPWR_ATTR_UNDER_VOLTAGE_PROTECTION_LIMIT
```

Description

Specifies the output under voltage protection limit value. The units are Volts RMS.

9.2.2 Over Voltage Limit

Data Type	Access	Applies to	Coercion	High Level Functions
`ViReal64`	`R/W`	OutputPhase	None	Configure Voltage Protection

COM Property Name

```
OutputPhases.Item().VoltageProtection.OverLimit
```

COM Enumeration Name

```
N/A
```

C Constant Name

```
IVIACPWR_ATTR_OVER_VOLTAGE_PROTECTION_LIMIT
```

Description

Specifies the output over voltage protection limit value. The units are Volts RMS.

9.2.3 Over Voltage Enabled

Data Type	Access	Applies to	Coercion	High Level Functions
`ViBoolean`	`R/W`	OutputPhase	None	Configure Voltage Protection

COM Property Name

```
OutputPhases.Item().VoltageProtection.OverEnabled
```

COM Enumeration Name

```
N/A
```

C Constant Name

```
IVIACPWR_ATTR_OVER_VOLTAGE_PROTECTION_ENABLED
```

Description

When set to True, voltage protection mode is triggered when the output voltage is above the Over voltage limit.

Defined Values

Name	***Description***		
		Language	***Identifier***
True	Over voltage protection enabled.		
		C	`VI_TRUE`
		COM	`True`
False	Over voltage protection disabled.		
		C	`VI_FALSE`
		COM	`False`

Compliance Notes

Instrument drivers shall support the value False.

9.2.4 Under Voltage Enabled

Data Type	Access	Applies to	Coercion	High Level Functions
`ViBoolean`	`R/W`	OutputPhase	None	Configure Voltage Protection

COM Property Name

```
OutputPhases.Item().VoltageProtection.UnderEnabled
```

COM Enumeration Name

```
N/A
```

C Constant Name

```
IVIACPWR_ATTR_UNDER_VOLTAGE_PROTECTION_ENABLED
```

Description

When set to True, voltage protection mode is triggered when the output voltage is below the Under Voltage Limit.

Defined Values

Name	***Description***		
		Language	***Identifier***
True	Enable under voltage protection.		
		C	`VI_TRUE`
		COM	`True`
False	Disable under voltage protection.		
		C	`VI_FALSE`
		COM	`False`

Compliance Notes

Instrument drivers shall support the value False.

9.2.5 Voltage Protection Tripped

Data Type	Access	Applies to	Coercion	High Level Functions
ViBoolean	RO	OutputPhase	None	Query Voltage Protection Tripped (IVI-C only)

COM Property Name

```
OutputPhases.Item().VoltageProtection.Tripped
```

COM Enumeration Name

```
N/A
```

C Constant Name

```
IVIACPWR_ATTR_VOLTAGE_PROTECTION_TRIPPED
```

Description

Indicates if voltage protection has tripped. Use the Reset Voltage Protection function to reset the voltage protection state.

Defined Values

Name	***Description***		
		Language	***Identifier***
True	Voltage protection has tripped.		
		C	VI_TRUE
		COM	True
False	Voltage protection has not tripped.		
		C	VI_FALSE
		COM	False

9.3 IviACPwrVoltageProtection Functions

The IviACPwrVoltageProtection extension group defines the following functions:

- Query Voltage Protection Tripped (IVI-C only)
- Reset Voltage Protection
- Configure Voltage Protection

9.3.1 Query Voltage Protection Tripped (IVI-C only)

Description

Returns the voltage protection state of the AC power source.

Voltage protection trips when the output voltage is above the over-voltage protection limit attribute or below the under-voltage protection limit attributes and the corresponding voltage protection enabled attribute is set to True.

When voltage protection trips, the power source enters the voltage protection state and disables the output. The user must call the Reset Voltage Protection function to reset the voltage protection state of the AC power source. Once voltage protection is reset, the user must then set the Output Enabled attribute to True for the AC power source to resume generating output voltage.

COM Method Prototype

`N/A`

(use the `OutputPhases.Item().VoltageProtection.Tripped` property)

C Prototype

```
ViStatus IviACPwr_QueryVoltageProtectionTripped (ViSession Vi,
                                            ViConstString PhaseName,
                                            ViBoolean *Tripped);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle	`ViSession`
`PhaseName`	The name of the output on which to return the voltage protection status.	`ViConstString`

Outputs	Description	Base Type
`Tripped`	Returns True if voltage protection has tripped	`ViBoolean`

Defined Values for Tripped Parameter

Name	*Description*		
		Language	*Identifier*
True	Voltage protection has tripped.		
		C	`VI_TRUE`
		COM	`True`
False	Voltage protection has not tripped.		
		C	`VI_FALSE`
		COM	`False`

Return Values

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

9.3.2 Reset Voltage Protection

Description

Resets the AC power source voltage protection state after a voltage protection condition has occurred. The user must also set the Output Enabled attribute in IviACPwrBase to True to resume generating output power.

Use the Query Voltage Protection Tripped function to determine if voltage protection has tripped.

COM Method Prototype

```
HRESULT OutputPhases.Item().VoltageProtection.Reset();
```

C Prototype

```
ViStatus IviACPwr_ResetVoltageProtection (ViSession Vi,
                                          ViConstString PhaseName);
```

Parameters

Inputs	**Description**	**Base Type**
Vi	Instrument handle	ViSession
`PhaseName`	The name of the output on which to reset the voltage protection.	`ViConstString`

Return Values

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

9.3.3 Configure Voltage Protection

Description

Configures voltage protection. When the corresponding enable parameter is True, the AC power source enters the voltage protection state (tripped) and disables the output when the output voltage is below the Under Limit or above the Over Limit.

When the `OverEnabled` and/or `UnderEnabled` parameter is False, the corresponding limit does not affect the instrument's behavior and under and/or over voltage protection is disabled.

COM Method Prototype

```
HRESULT OutputPhases.Item().VoltageProtection.Configure(
                           [in} BOOL UnderEnabled,
                           [in} BOOL OverEnabled,
                           [in] DOUBLE UnderLimit,
                           [in] DOUBLE OverLimit);
```

C Prototype

```
ViStatus IviACPwr_ConfigureVoltageProtection (ViSession Vi,
                             ViConstString PhaseName,
                             ViBoolean UnderEnabled,
                             ViBoolean OverEnabled,
                             ViReal64 UnderLimit,
                             ViReal64 OverLimit);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle	`ViSession`
`PhaseName`	The name of the output on which to set the voltage protection values.	`ViConstString`
`OverEnabled`	The driver uses this value to set the Over Voltage Enabled attribute. See the attribute description for more details.	`ViBoolean`
`UnderEnabled`	The driver uses this value to set the Under Voltage Enabled attribute. See the attribute description for more details.	`ViBoolean`
`UnderLimit`	Specifies the under voltage protection level. The driver uses this value to set the Under Voltage Limit attribute. See the attribute description for more details.	`ViReal64`
`OverLimit`	Specifies the over voltage protection level. The driver uses this value to set the Over Voltage Limit attribute. See the attribute description for more details.	`ViReal64`

Return Values

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

9.4 IviACPwrVoltageProtection Behavior Model

The IviACPwrVoltageProtection Extension Group follows the behavior model of the IviACPwrBase capability group. The only addition to the behavior model is the driver must set the Output Enabled attribute in the IviACPwrBase capability group to False when a voltage protection condition occurs.

In order to resume generating output power after voltage protection trips, the user must call the Reset Voltage Protection function then set the Output Enabled attribute in IviACPwrBase to True.

It is the responsibility of the user to determine if the circuit is appropriate or safe for resuming generating output voltage after voltage protection trips.

10 IviACPwrArbWaveform Extension Group

10.1 IviACPwrArbWaveform Overview

The IviACPwrArbWaveform Extension Group defines an AC Power Source capable of producing user-defined arbitrary waveforms. The IviACPwrArbWaveform extension group includes functions for Writing, Clearing, and for returning information about arbitrary waveform capabilities and catalogs.

An Arbitrary Waveform is a user-defined series of sequential data points that describe an output waveform for a full cycle (360°). The frequency, phase angle, RMS amplitude and DC offset of the arbitrary waveform at the power source output is defined by the capability group and attribute shown in the following table.

Waveform Property	Capability Group	Attribute
Frequency	IviAcPwrBase	Frequency
Phase Angle	IviAcPwrPhase	Phase Angle
RMS Amplitude	IviAcPwrBase	Voltage Level
DC Offset	IviAcPwrDCGeneration	DC Offset

The amplitude of the waveform data array passed to the Write Waveform function is rescaled to the native waveform amplitude range supported by the AC power source. The number of data points in the waveform data array is automatically resampled to the native waveform data points supported by the AC power source.

The IviACPwrArbWaveform extension group includes read only attributes that define the maximum number of arbitrary waveforms the power source can store, the number of presently stored and available waveforms and the optimal number of data points of an arbitrary waveform. The functions include writing and clearing arbitrary waveforms and querying the arbitrary waveform catalog .

In addition to supporting user-defined arbitrary waveforms, the AC power source contains a base Sine waveform that is always present. Instrument vendors may also define additional fixed waveforms that cannot be cleared.

The driver must not allow user defined or fixed vendor defined waveforms to contain a DC Offset. This is because the DC offset of the power source output voltage must be exclusively controlled by the IviAcPwrDCGeneration extension capability group. The DC Offset can be determining by adding together all data points of the waveform. The absolute value of the sum must be less than 0.001% of the peak to peak amplitude.

Waveforms are referenced using a string name. This name can be used to clear a waveform from the power source memory (allowing other waveforms to be written) or specify which waveform the power source should use when generating the voltage on an output phase (Refer to Waveform attribute in the IviACPwrBase capability group).

The waveform data values may affect the maximum output voltage the power source can generate using that waveform. Refer to section *A.6 Waveform RMS and Peak Voltage* for more information.

10.2 IviACPwrArbWaveform Attributes

The IviACPwrArbWaveform capability group defines the following attributes:

- Num Waveforms Max
- Num Optimal Data Points
- Fixed Waveform Count
- User Waveform Count
- Available Waveform Count

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in Section 16, *IviACPwr Attribute ID Definitions*.

10.2.1 Num Waveforms Max

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViInt32	RO	N/A	None	None

COM Property Name

ArbWaveform.NumWaveformsMax

COM Enumeration Name

N/A

C Constant Name

IVIACPWR_ATTR_NUM_WAVEFORMS_MAX

Description

Returns the maximum number of arbitrary waveforms the power source can store including user defined waveforms and fixed vendor defined waveforms.

10.2.2 Num Optimal Data Points

Data Type	Access	Applies To	Coercion	High Level Function(s)
`ViInt32`	`RO`	`N/A`	`None`	None

COM Property Name

```
ArbWaveform.NumOptimalDataPoints
```

COM Enumeration Name

N/A

C Constant Name

```
IVIACPWR_ATTR_NUM_OPTIMAL_DATA_POINTS
```

Description

Returns the optimal (native) number of data points the AC power source uses for arbitrary waveforms.

The driver will automatically resample the waveform data array passed to the Write Arbitrary Waveform function to fit the optimal number of data points supported by the AC power source. To minimize the effects of resampling and to maximize the waveform resolution, allocate user defined waveform data using the optimal number of data points of the AC power source.

10.2.3 Fixed Waveform Count

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViInt32	RO	N/A	None	None

COM Property Name

```
ArbWaveform.FixedWaveformCount
```

COM Enumeration Name

N/A

C Constant Name

```
IVIACPWR_ATTR_FIXED_WAVEFORM_COUNT
```

Description

Returns the number of fixed vendor defined arbitrary waveforms stored in the power source.

10.2.4 User Waveform Count

Data Type	Access	Applies To	Coercion	High Level Function(s)
ViInt32	RO	N/A	None	None

COM Property Name

`ArbWaveform.UserWaveformCount`

COM Enumeration Name

N/A

C Constant Name

`IVIACPWR_ATTR_USER_WAVEFORM_COUNT`

Description

Returns the number of user defined arbitrary waveforms stored in the power source.

10.2.5 Available Waveform Count

Data Type	Access	Applies To	Coercion	High Level Function(s)
`ViInt32`	`RO`	`N/A`	`None`	None

COM Property Name

`ArbWaveform.AvailableWaveformCount`

COM Enumeration Name

N/A

C Constant Name

`IVIACPWR_ATTR_AVAILABLE_WAVEFORM_COUNT`

Description

Returns the number of additional arbitrary waveforms the user can store in the power source. Use the Clear Arbitrary Waveform function to increase the number user defined arbitrary waveforms the power source can store.

The number of available waveforms can also be calculated from the following attributes:

Available Waveforms = Num Waveforms Max – Fixed Waveform Count – User Waveform Count

10.3 IviACPwrArbWaveform Functions

The IviACPwrArbWaveform extension defines the following functions:

- Clear Arbitrary Waveform
- Write Arbitrary Waveform
- Query Arbitrary Waveform Catalog

This section describes the behavior and requirements of each function.

10.3.1 Clear Arbitrary Waveform

Description

Deletes individual user-defined waveforms from the power source's memory.

If the waveform cannot be cleared because it is currently being used to generate output voltage, this function returns the Waveform In Use error. If the waveform name is “Sine” or a fixed vendor defined waveform, this function returns the Waveform Reserved error.

COM Method Prototype

```
HRESULT ArbWaveform.Clear ([in] BSTR WaveformName );
```

C Prototype

```
ViStatus IviACPwr_ClearArbWaveform (ViSession Vi,
                                   ViConstString WaveformName);
```

Parameters

Inputs	**Description**	**Base Type**
`Vi`	Instrument handle	`ViSession`
`WaveformName`	Specifies the name of the user defined waveform to remove from the power source memory. If the waveform name parameter is VI_NULL, then all user defined waveforms are cleared.	`ViConstString`

Return Values

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**
Waveform In Use	The power source is currently configured to produce the specified waveform.
Waveform Reserved	The waveform name references a vendor defined fixed waveform.
Waveform Not Found	The waveform name is not currently stored in the AC Power Source.

Compliance Notes

1. Specific drivers shall implement VI_NULL for the WaveformName parameter to clear all user defined waveforms.

10.3.2 Write Arbitrary Waveform

Description

Writes an individual user-defined waveform to the AC power source's memory. If the power source cannot store any more arbitrary waveforms, this function returns the No Waveforms Available error.

If the waveform name already exists, then the function returns the Duplicate Waveform Name error.

If the absolute value of the sum of all waveform data points exceeds 0.001% of the peak to peak amplitude, the function returns the Waveform DC Offset error.

COM Method Prototype

```
HRESULT ArbWaveform.Write ([in] BSTR WaveformName,
                                  [in] SAFEARRAY(double) *WaveformData);
```

C Prototype

```
ViStatus IviACPwr_WriteArbWaveform (ViSession Vi,
                                   ViConstString WaveformName,
                                   ViInt32 WaveformDataBufferSize,
                                   ViReal64 WaveformData[] );
```

Parameters

Inputs	**Description**	**Base Type**
Vi	Instrument handle	ViSession
WaveformName	Specifies the waveform name that identifies the arbitrary waveform to write to the AC power source.	ViConstString
WaveformDataBufferSize	The size of the IVI-C user allocated waveform data buffer	ViInt32
WaveformData	Specifies the user allocated array of the user-defined waveform data to be stored in the AC power source.	ViReal64[]

Return Values

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
Duplicate Waveform Name	The specified waveform name is already stored in the AC power source.
No Waveforms Available	The waveform memory in the AC power source is full.
Invalid Waveform Name	The specified waveform name is invalid.
Waveform DC Offset	The waveform contains a DC offset that exceeds 0.001% of the peak to peak amplitude,

Compliance Notes

1. Waveform names cannot contain a comma character.

2. The driver can use the following code to detect DC offset in the user supplied waveform array.

```
ViReal64 sum=0, min, max, pkpk, y;
ViInt32 i;

// Calculate sum, min and max
min = WaveformData[0];
max=min;
for (i=0;i<WaveformDataBufferSize;i++) {
    y = WaveformData[i];
    if (y> max) max = y;
    if (y< min) min = y ;
    sum += y;
}

// Calculate 0.001% of peak to peak amplitude
pkpk = 0.001 * fabs(max - min) / 100;

// Return error if DC offset exceeds limit
if (fabs(sum) > pkpk) return IVIACPWR_ERROR_WAVEFORM_DC_OFFSET;
```

10.3.3 Query Arbitrary Waveform Catalog

Description

Returns a comma delimited string of user defined and/or fixed vendor defined waveform names currently stored in the AC power source.

The number of additional user defined waveforms the power source can store can be determined by the Available Waveform Count attribute.

COM Method Prototype

```
HRESULT ArbWaveform.QueryCatalog ([in] long CatalogType,
                                  [in, out] BSTR *Catalog);
```

C Prototype

```
ViStatus IviACPwr_QueryArbWaveformCatalog (ViSession Vi,
                                        ViInt32 CatalogType,
                                        ViInt32 CatalogBufferSize,
                                        ViChar Catalog[]);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
CatalogType	Specifies if the returned catalog string contains fixed vendor defined and/or user defined waveform names.	ViInt32
CatalogBufferSize	The size of the IVI-C user allocated buffer.	ViInt32

Outputs	Description	Base Type
Catalog	A user allocated buffer to store a comma delimited string of waveform names currently stored in the AC power source.	ViChar[]

Defined Values for CatalogType Parameter

Name	***Description***		
		Language	***Identifier***
Fixed	Request catalog of fixed vendor defined waveform names.		
		C	`IVIACPWR_VAL_WAVEFORM_CATALOG_FIXED`
		COM	`IviACPwrWaveformCatalogFixed`
User	Request catalog of user defined waveform names.		
		C	`IVIACPWR_VAL_WAVEFORM_CATALOG_USER`
		COM	`IviACPwrWaveformCatalogUser`
All	Request catalog of all waveform names.		
		C	`IVIACPWR_VAL_WAVEFORM_CATALOG_ALL`
		COM	`IviACPwrWaveformCatalogAll`

Return Values

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

10.4 IviACPwrArbWaveform Behavior Model

The IviACPwrArbWaveform Extension Group follows the behavior model of the IviACPwrBase capability group but modifies the Output Enabled attribute as described below.

Certain AC power sources may require that output power be disabled in order to write a waveform. Refer to *Appendix A.5: Operations that Require the Output be Disabled* for more information.

11 IviACPwrImpedance Extension Group

11.1 IviACPwrImpedance Overview

The IviACPwrImpedance extension group defines extensions for AC Power Sources with the ability to configure the output impedance. Typically the output impedance is set using firmware emulation of impedance or analog control loop.

11.2 IviACPwrImpedance Attributes

The IviACPwrImpedance extension defines the following attributes:

- Output Impedance Enabled
- Output Impedance Resistive
- Output Impedance Inductive

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in section 16, IviACPwr Attribute ID Definitions.

11.2.1 Output Impedance Enabled

Data Type	Access	Applies to	Coercion	High Level Functions
`ViBoolean`	`R/W`	OutputPhase	None	Configure Output Impedance

COM Property Name

```
OutputPhases.Item().Impedance.Enabled
```

COM Enumeration Name

```
N/A
```

C Constant Name

```
IVIACPWR_ATTR_OUTPUT_IMPEDANCE_ENABLED
```

Description

When set to True, this attribute allows the output impedance resistive and output impedance inductive attribute values to affect the power source output.

Defined Values

Name	***Description***		
		Language	***Identifier***
True	The output impedance values affect the AC source output		
		C	`VI_TRUE`
		COM	`True`
False	The output impedance feature is disabled		
		C	`VI_FALSE`
		COM	`False`

Compliance Notes

Instrument drivers shall support the value False.

11.2.2 Output Impedance Resistive

Data Type	Access	Applies to	Coercion	High Level Functions
`ViReal64`	`R/W`	OutputPhase	None	Configure Output Impedance

COM Property Name

```
OutputPhases.Item().Impedance.Resistive
```

COM Enumeration Name

```
N/A
```

C Constant Name

```
IVIACPWR_ATTR_OUTPUT_IMPEDANCE_RESISTIVE
```

Description

Specifies the resistive output impedance value in ohms. This value affects the power source output when the Output Impedance Enabled attribute is set to True.

Compliance Notes

Instrument drivers shall support the value 0 and may coerce this to the lowest values supported by the instrument. Use the Query Output Impedance Capabilities function to obtain a range of allowable values.

11.2.3 Output Impedance Inductive

Data Type	Access	Applies to	Coercion	High Level Functions
`ViReal64`	`R/W`	OutputPhase	None	Configure Output Impedance

COM Property Name

```
OutputPhases.Item().Impedance.Inductive
```

COM Enumeration Name

```
N/A
```

C Constant Name

```
IVIACPWR_ATTR_OUTPUT_IMPEDANCE_INDUCTIVE
```

Description

Specifies the inductive output impedance value in henries. This value affects the power source output when the Output Impedance Enabled attribute is set to True.

Compliance Notes

Instrument drivers shall support the value 0 and may coerce this to the lowest values supported by the instrument. Use the Query Output Impedance Capabilities function to obtain a range of allowable values.

11.3 IviACPwrImpedance Functions

The IviACPwrImpedance extension defines the following functions:

- Configure Output Impedance
- Query Output Impedance Capabilities

Configure Output Impedance configures the output impedance of the AC power source by setting the Output Impedance Enabled, Output Impedance Resistive and Output Impedance Inductive attributes.

Query Output Impedance Capabilities returns the minimum and maximum allowable resistive and inductive impedance the AC power source can be programmed for the present configuration.

11.3.1 Configure Output Impedance

Description

Configures the output impedance of the AC power source. The Enabled parameter must be set to True in order for the Resistive or Inductive values to affect the power source output.

COM Method Prototype

```
HRESULT OutputPhases.Item().Impedance.Configure (
                           [in} BOOL Enabled,
                           [in] DOUBLE ResistiveValue,
                           [in] DOUBLE InductiveValue );
```

C Prototype

```
ViStatus IviACPwr_ConfigureOutputImpedance (ViSession Vi,
                           ViConstString PhaseName,
                           ViBoolean Enabled,
                           ViReal64 ResistiveValue,
                           ViReal64 InductiveValue );
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
PhaseName	The name of the output on which to configure the output impedance.	ViConstString
Enabled	Enables/disables the output impedance capability of the AC power source. The driver uses this value to set the Output Impedance Enabled attribute. See the attribute description for more details.	ViBoolean
ResistiveValue	Sets the resistive part of the output impedance of the AC power source in Ohms. The driver uses this value to set the Output Impedance Resistive attribute. See the attribute description for more details.	ViReal64
InductiveValue	Sets the inductive part of the output impedance of the AC power source in Henries. The driver uses this value to set the Output Impedance Inductive attribute. See the attribute description for more details.	ViReal64

Return Values

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

11.3.2 Query Output Impedance Capabilities

Description

Queries the output impedance capabilities of the AC power source based on the present configuration.

COM Method Prototype

```
HRESULT OutputPhases.Item().Impedance.QueryCapabilities (
                          [in, out} DOUBLE *ResistiveMin,
                          [in, out] DOUBLE *ResistiveMax,
                          [in, out] DOUBLE *InductiveMin,
                          [in, out] DOUBLE *InductiveMax);
```

C Prototype

```
ViStatus IviACPwr_QueryOutputImpedanceCapabilities (ViSession Vi,
                          ViConstString PhaseName,
                          ViReal64 *ResistiveMin,
                          ViReal64 *ResistiveMax,
                         ViReal64 *InductiveMin,
                          ViReal64 *InductiveMax);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle	`ViSession`
`PhaseName`	The name of the output on which to return the output impedance capabilities.	`ViConstString`

Outputs	Description	Base Type
`ResistiveMin`	Returns the minimum resistive impedance value the power source can be programmed.	`ViReal64`
`ResistiveMax`	Returns the maximum resistive impedance value the power source can be programmed.	`ViReal64`
`InductiveMin`	Returns the minimum inductive impedance value the power source can be programmed.	`ViReal64`
`InductiveMax`	Returns the maximum inductive impedance value the power source can be programmed.	`ViReal64`

Return Values

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

11.4 IviACPwrImpedance Behavior Model

The IviACPwrImpedance Extension Group follows the behavior model of the IviACPwrBase capability group but modifies the Output Enabled attribute as described below.

Certain AC power sources may require that the output power be disabled in order to change the Output Impedance Enabled, Output Impedance Resistive or Output Impedance Inductive values. Refer to *Appendix A.5: Operations that Require the Output be Disabled* for more information.

12 IviACPwrDCGeneration Extension Group

12.1 IviACPwrDCGeneration Overview

The IviACPwrDCGeneration extension group defines extensions for AC Power Sources capable of producing a DC output.

There are three modes of operation; AC Only, DC Only and AC Plus DC. The mode of operation determines when the Voltage Level attribute in the IviACPwrBase capability group and the DC Voltage Level attribute in the IviACPwrDCGeneration are used as shown in the following table.

DC Mode	Voltage Level (IviACPwrBase)	DC Voltage Level (IviACPwrDCGeneration)
AC Only	AC RMS voltage	Not Used
DC Only	Not Used	DC Voltage
AC Plus DC	AC RMS voltage	DC Offset

The number of DC ranges the AC power supply supports are specified by the Number DC Ranges attribute. Each DC range supports a minimum and maximum DC voltage. Use the Query DC Capabilities function to determine the minimum and maximum DC voltage supported by each range.

Ranges may support unipolar or bipolar voltages. Ranges that support bipolar voltages may not be symmetrical. The DC Voltage Level attribute must not exceed the minimum and maximum DC voltages supported by the present range.

12.2 IviACPwrDCGeneration Attributes

The IviACPwrDCGeneration extension group defines the following attributes:

- DC Mode
- DC Voltage Level
- DC Range Min
- DC Range Max
- Number of DC Ranges

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in section 16, *IviACPwr Attribute ID Definitions*.

12.2.1 DC Mode

Data Type	**Access**	**Applies to**	**Coercion**	**High Level Functions**
`ViInt32`	`R/W`	OutputPhase	None	Configure DC

COM Property Name

`OutputPhases.Item().DCGeneration.Mode`

COM Enumeration Name

`N/A`

C Constant Name

`IVIACPWR_ATTR_DC_MODE`

Description

Configures AC Only, DC Only or a combination of AC Plus DC output voltage generation. When using AC Only or AC Plus DC generation mode, the Voltage Level attribute in IviACPwrBase sets the RMS value of the AC voltage.

Defined Values

Name	***Description***		
		Language	***Identifier***
AC Only	AC Only voltage generation		
		C	`IVIACPWR_VAL_MODE_AC_ONLY`
		COM	`IviACPwrDCGenerationModeACOnly`
DC Only	DC Only voltage generation		
		C	`IVIACPWR_VAL_MODE_DC_ONLY`
		COM	`IviACPwrDCGenerationModeDCOnly`
AC Plus DC	AC Plus DC voltage generation		
		C	`IVIACPWR_VAL_MODE_AC_DC`
		COM	`IviACPwrDCGenerationModeACPlusDC`

12.2.2 DC Voltage Level

Data Type	Access	Applies to	Coercion	High Level Functions
`Vireal64`	`R/W`	OutputPhase	None	Configure DC

COM Property Name

```
OutputPhases.Item().DCGeneration.DCVoltageLevel
```

COM Enumeration Name

```
N/A
```

C Constant Name

```
IVIACPWR_ATTR_DC_VOLTAGE_LEVEL
```

Description

The DC Voltage when the DC Mode attribute is set to DC Only or the DC Offset when the DC Mode attribute is set to AC Plus DC. Refer to the table in section 12.1 for more information on how this attribute affects the power source output.

12.2.3 DC Range Min

Data Type	Access	Applies to	Coercion	High Level Functions
`ViReal64`	`RO`	OutputPhase	None	Configure DC Range

COM Property Name

```
OutputPhases.Item().DCGeneration.RangeMin
```

COM Enumeration Name

```
N/A
```

C Constant Name

```
IVIACPWR_ATTR_DC_RANGE_MIN
```

Description

Returns the minimum DC voltage supported by the power source for the present DC voltage range. Use the Configure DC Range function to set the DC voltage range.

12.2.4 DC Range Max

Data Type	Access	Applies to	Coercion	High Level Functions
`ViReal64`	`RO`	OutputPhase	None	Configure DC Range

COM Property Name

```
OutputPhases.Item().DCGeneration.RangeMax
```

COM Enumeration Name

```
N/A
```

C Constant Name

```
IVIACPWR_ATTR_DC_RANGE_MAX
```

Description

Returns the maximum DC voltage supported by the power source for the present DC voltage range. Use the Configure DC Range function to set the DC voltage range.

12.2.5 Number of DC Ranges

Data Type	Access	Applies to	Coercion	High Level Functions
`ViInt32`	`RO`	OutputPhase	None	None

COM Property Name

```
OutputPhases.Item().DCGeneration.NumRanges
```

COM Enumeration Name

```
N/A
```

C Constant Name

```
IVIACPWR_ATTR_DC_NUM_RANGES
```

Description

The number of DC ranges supported by the power source.

Defined Values

N/A

Compliance Notes

All vendors that support the DC Generation extension group must support at least 1 DC voltage range

12.3 IviACPwrDCGeneration Functions

The IviACPwrDCGeneration extension group defines the following functions:

- Configure DC
- Configure DC Range
- Query DC Capabilities

This section describes the behavior and requirements of the functions.

12.3.1 Configure DC

Description

Configures the DC output generation.

COM Method Prototype

```
HRESULT OutputPhases.Item().DCGeneration.Configure(
                           [in] long Mode,
                           [in] double DCVoltageLevel);
```

C Prototype

```
ViStatus IviACPwr_ConfigureDC (ViSession Vi,
                                  ViConstString PhaseName,
                                 ViInt32 Mode,
                                  ViReal64 DCVoltageLevel);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle	`ViSession`
`PhaseName`	The name of the output on which to configure DC generation.	`ViConstString`
`Mode`	Configures AC Only, DC Only or AC Plus DC output generation. The driver uses this value to set the DC Mode attribute. See the attribute description for more details.	`ViInt32`
`DCVoltageLevel`	Specifies the power source DC output voltage when using the DC Only mode or the DC offset voltage when using the AC Plus DC mode. The driver uses this value to set the DC Voltage Level attribute. See the attribute description for more details.	`ViReal64`

Return Values

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

12.3.2 Configure DC Range

Description

Configures the DC voltage range of the AC power source. The driver sets the power source to the lowest range that supports the minimum and maximum voltages specified by the Minimum and Maximum parameters. It then updates the DC Range Min and DC Range Max attributes.

COM Method Prototype

```
HRESULT OutputPhases.Item().DCGeneration.ConfigureRange(
                                        [in] double Minimum
                                        [in] double Maximum);
```

C Prototype

```
ViStatus IviACPwr_ConfigureDCRange (ViSession Vi,
                                         ViConstString PhaseName,
                                         ViReal64 Minimum,
                                         ViReal64 Maximum);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
PhaseName	The name of the output on which to configure the DC Generation range.	ViConstString
Minimum	Specifies the minimum DC voltage. The driver coerces this value to the minimum voltage of the lowest range that supports this value.	ViReal64
Maximum	Specifies the maximum DC voltage. The driver coerces this value to the maximum voltage of the lowest range that supports this value.	ViReal64

Return Values

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

12.3.3 Query DC Capabilities

Description

Returns the minimum and maximum DC voltages the power source can generate for the specified range. Typically, the user calls the ConfigureDCRange function with the min and max voltages they want to use and it automatically selects the best range. The only time the Query DC Capabilities function would be useful is to eliminate unnecessary range changes by allowing the user to know in advance the range boundaries.

The DC Range Min and DC Range Max attributes return the min and max voltage of the present range.

COM Method Prototype

```
HRESULT OutputPhases.Item().DCGeneration.QueryCapabilities(
                           [in] long Range
                           [in, out] double *Minimum
                           [in, out] double *Maximum);
```

C Prototype

```
ViStatus IviACPwr_QueryDCCapabilities(ViSession Vi,
                                  ViConstString PhaseName,
                                  ViInt32 Range,
                                  ViReal64 *Minimum,
                                  ViReal64 *Maximum);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle	`ViSession`
`PhaseName`	The name of the output on which to query the DC generation capabilities.	`ViConstString`
`Range`	The DC range to query. This number must be from 1 to the values returned by the IVIACPWR_ATTR_DC_ NUM_RANGES attribute	`ViInt32`

Outputs	Description	Base Type
`Minimum`	Returns the minimum DC voltage the power source can generate	`ViReal64`
`Maximum`	Returns the maximum DC voltage the power source can generate	`ViReal64`

Return Values

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

12.4 IviACPwrDCGeneration Behavior Model

The IviACPwrDCGeneration Extension Group follows the behavior model of the IviACPwrBase capability group but modifies the Output Enabled attribute as described below.

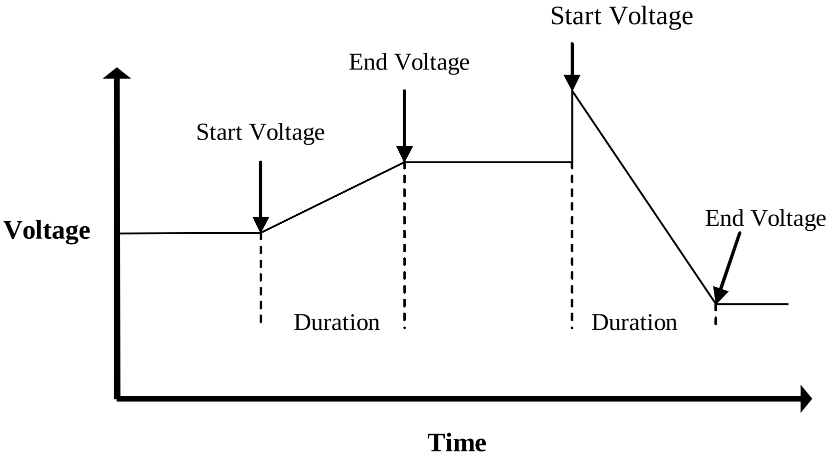
Certain AC power sources may require that the output power be disabled in order to change the DC Mode, DC Offset or DC Range. Refer to *Appendix A.5: Operations that Require the Output be Disabled* for more information.

13 IviACPwrVoltageRamp Extension Group

13.1 IviACPwrVoltageRamp Overview

The IviACPwrVoltageRamp extension group defines extensions for AC Power Sources with the capability to configure a linear voltage ramp with a starting and ending voltage in volts RMS and duration in seconds.

The Start Voltage value takes effect immediately then the voltage ramps linearly to the End Voltage value for the duration specified.



This figure above shows 2 voltage ramps. The Start Voltage of the first ramp was set to the same value as the Voltage Level attribute in IviACPwr base. The Start Voltage of the second ramp was set to a value higher than the present Voltage Level attribute, causing an immediate change in output voltage before linearly ramping to the End Voltage value.

13.2 IviACPwrVoltageRamp Attributes

The IviACPwrVoltageRamp extension group defines the following Attributes:

- Voltage Ramp Busy

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in section 16, *IviACPwr Attribute ID Definitions*.

13.2.1 Voltage Ramp Busy

Data Type	Access	Applies to	Coercion	High Level Functions
`ViBoolean`	`RO`	OutputPhase	None	Query Voltage Ramp Busy (IVI-C only)

COM Property Name

```
OutputPhases.Item().VoltageRampBusy
```

COM Enumeration Name

```
N/A
```

C Constant Name

```
IVIACPWR_ATTR_VOLTAGE_RAMP_BUSY
```

Description

Indicates if a voltage ramp is in progress (busy).

Defined Values

Name	***Description***		
		Language	***Identifier***
True	Voltage ramp in progress.		
		C	`VI_TRUE`
		COM	`True`
False	Voltage ramp not in progress.		
		C	`VI_FALSE`
		COM	`False`

13.3 IviACPwrVoltageRamp Functions

The IviACPwrVoltageRamp extension defines the following functions:

- Ramp Voltage
- Query Voltage Ramp Busy (IVI-C only)
- Abort Voltage Ramp

This section describes the behavior and requirements of these functions.

13.3.1 Ramp Voltage

Description

Starts a linear voltage ramp from a starting voltage to an ending voltage in volts RMS for a duration in seconds. Call the Query Voltage Ramp Busy function to determine when the ramp completes or call the Abort Voltage Ramp function to stop the voltage ramp.

The starting and ending ramp voltages must not exceed the values of the present voltage range. Refer to the Voltage Range attribute in the IviACPwrBase capability group.

COM Method Prototype

```
HRESULT OutputPhases.Item().RampVoltage(
                              [in] double StartVoltage);
                              [in] double EndVoltage);
                              [in] double Duration);
```

C Prototype

```
ViStatus IviACPwr_RampVoltage (ViSession Vi,
                                         ViConstString PhaseName,
                                         ViReal64 StartVoltage);
                                         ViReal64 EndVoltage);
                                         ViReal64 Duration);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
PhaseName	The name of the output on which to configure the voltage ramp.	ViConstString
StartVoltage	Specifies the power source’s starting output voltage in volts rms.	ViReal64
EndVoltage	Specifies the power source’s ending output voltage in volts rms.	ViReal64
Duration	Duration of the voltage ramp in seconds.	ViReal64

Return Values

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

Compliance Notes

Instruments that must complete the voltage ramp before returning are allowed to be compliant with the IviACPwrVoltageRamp extension group. For these instruments, the Ramp Voltage function waits for the ramp to complete before returning and the Query Voltage Ramp Busy function always returns False.

13.3.2 Query Voltage Ramp Busy (IVI-C only)

Description

Returns if a voltage ramp is in progress. Call this function after calling the Ramp Voltage function to determine when the ramp completes.

COM Method Prototype

```
N/A
```

(use the `OutputPhases.Item().VoltageRampBusy` property)

C Prototype

```
ViStatus IviACPwr_QueryVoltageRampBusy (ViSession Vi,
                                        ViConstString PhaseName,
                                        ViBoolean *Busy);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle	`ViSession`
`PhaseName`	The name of the output on which to query if the voltage ramp is busy.	`ViConstString`

Outputs	Description	Base Type
`Busy`	Returns True if a voltage ramp is in progress. Returns False when a Voltage Ramp completes or is not in progress.	`ViBoolean`

Defined Values for Busy Parameter

Name	*Description*		
		Language	*Identifier*
True	Voltage ramp in progress.		
		C	`VI_TRUE`
		COM	`True`
False	Voltage ramp not in progress		
		C	`VI_FALSE`
		COM	`False`

13.3.3 Abort Voltage Ramp

Description

Aborts a voltage ramp. If a voltage ramp is not in progress this function does nothing.

COM Method Prototype

```
HRESULT OutputPhases.Item().AbortVoltageRamp();
```

C Prototype

```
ViStatus IviACPwr_AbortVoltageRamp (ViSession Vi,
                                       ViConstString PhaseName);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle	`ViSession`
`PhaseName`	The name of the output on which to abort the voltage ramp.	`ViConstString`

Return Values

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

13.4 IviACPwrVoltageRamp Behavior Model

The IviACPwrVoltageRamp Extension Group follows the behavior model of the IviACPwrBase capability group. The only modification to the behavior model from the IviACPwrBase capability group is the ability to ramp the output voltage.

When the Ramp Voltage function returns immediately after the ramp has started, the user should only call the Query Voltage Ramp Busy or Abort Voltage Ramp functions or read the Voltage Ramp Busy attribute while the ramp is in progress. Calling any other function or reading or writing any other attribute may abort the ramp or cause other vendor specific instrument behavior. After the voltage ramp completes or is aborted, the driver must invalidate the Voltage Level attribute in the IviACPwrBase capability group.

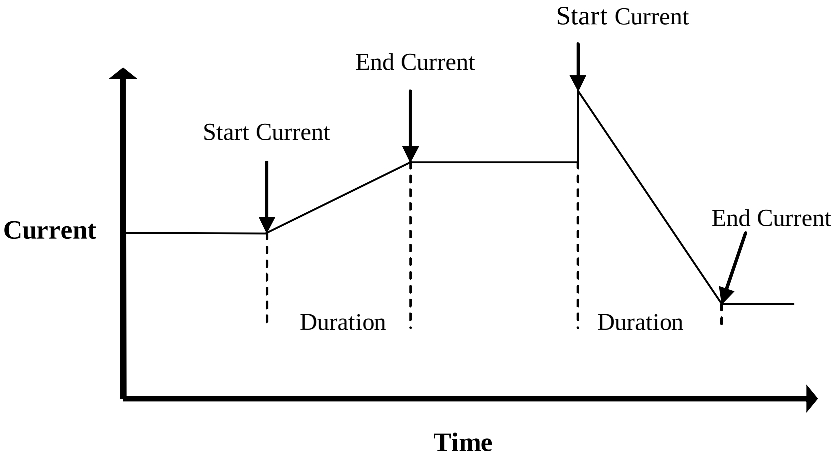
The actual voltage that appears at the power source output during a voltage ramp assumes the power source is operating in constant voltage mode. In other words, the output current during the voltage ramp is below the Current Limit attribute value in the IviACPwrBase capability group. If the power source operates in constant current mode during any portion of the ramp, the actual output voltage is reduced to maintain the current limit value but the actual ramp duration is not affected.

14 IviACPwrCurrentRamp Extension Group

14.1 IviACPwrCurrentRamp Overview

The IviACPwrCurrentRamp extension group defines extensions for AC Power Sources with the capability to configure a linear current ramp with a starting and ending current in amps and duration in seconds.

The Start Current value takes effect immediately then the current ramps linearly to the End Current value for the duration specified.



This figure above shows 2 current ramps. The Start Current of the first ramp was set to the same value as the Current Limit attribute in IviACPwr base. The Start Current of the second ramp was set to a value higher than the present Current Limit attribute, causing an immediate change in current limit before linearly ramping to the End Current value.

14.2 IviACPwrCurentRamp Attributes

The IviACPwrCurrentRamp extension group defines the following Attributes:

- Current Ramp Busy

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in section 16, *IviACPwr Attribute ID Definitions*.

14.2.1 Current Ramp Busy

Data Type	Access	Applies to	Coercion	High Level Functions
`ViBoolean`	`RO`	OutputPhase	None	Query Current Ramp Busy (IVI-C only)

COM Property Name

```
OutputPhases.Item().CurrentRampBusy
```

COM Enumeration Name

```
N/A
```

C Constant Name

```
IVIACPWR_ATTR_CURRENT_RAMP_BUSY
```

Description

Indicates if a current ramp is in progress (busy).

Defined Values

Name	***Description***		
		Language	***Identifier***
True	Current ramp in progress.		
		C	`VI_TRUE`
		COM	`True`
False	Current ramp not in progress.		
		C	`VI_FALSE`
		COM	`False`

14.3 IviACPwrCurrentRamp Functions

The IviACPwrCurrentRamp extension defines the following functions:

- Ramp Current
- Query Current Ramp Busy (IVI-C only)
- Abort Current Ramp

This section describes the behavior and requirements of these functions.

14.3.1 Ramp Current

Description

Starts a linear current ramp from a starting and ending current in amps for a duration in seconds. Call the Query Current Ramp Busy function to determine when the ramp completes or call the Abort Current Ramp function to stop the current ramp.

COM Method Prototype

```
HRESULT OutputPhases.Item().RampCurrent(
                              [in] double StartCurrent);
                              [in] double EndCurrent);
                              [in] double Duration);
```

C Prototype

```
ViStatus IviACPwr_RampCurrent (ViSession Vi,
                                         ViConstString PhaseName,
                                         ViReal64 StartCurrent);
                                         ViReal64 EndCurrent);
                                         ViReal64 Duration);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle	`ViSession`
`PhaseName`	The name of the output on which to configure the current ramp.	`ViConstString`
`StartCurrent`	Specifies the power source's starting output current in Amps.	`ViReal64`
`EndCurrent`	Specifies the power source's ending output current in Amps.	`ViReal64`
`Duration`	Duration of the current ramp in seconds	`ViReal64`

Return Values

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

Compliance Notes

Instruments that must complete the current ramp before returning are allowed to be compliant with the IviACPwrCurrentRamp extension group. For these instruments, the Ramp Current function waits for the ramp to complete before returning and the Query Current Ramp Busy function always returns False.

14.3.2 Query Current Ramp Busy (IVI-C only)

Description

Returns if a current ramp is in progress. Call this function after calling the Ramp Current function to determine when the ramp completes.

COM Method Prototype

```
N/A
```

(use the `OutputPhases.Item().CurrentRampBusy` property)

C Prototype

```
ViStatus IviACPwr_QueryCurrentRampBusy (ViSession Vi,
                                        ViConstString PhaseName,
                                        ViBoolean *Busy);
```

Parameters

Inputs	**Description**	**Base Type**
`Vi`	Instrument handle	`ViSession`
`PhaseName`	The name of the output on which to query if the current ramp is busy.	`ViConstString`

Outputs	**Description**	**Base Type**
`Busy`	Returns True if a current ramp is in progress. Returns False when a Current Ramp completes or is not in progress.	`ViBoolean`

Defined Values for Busy Parameter

Name	***Description***		
		Language	***Identifier***
True	Current ramp in progress.		
		C	`VI_TRUE`
		COM	`True`
False	Current ramp not in progress		
		C	`VI_FALSE`
		COM	`False`

14.3.3 Abort Current Ramp

Description

Aborts a current ramp. If a current ramp is not in progress this function does nothing.

COM Method Prototype

```
HRESULT OutputPhases.Item().AbortCurrentRamp();
```

C Prototype

```
ViStatus IviACPwr_AbortCurrentRamp (ViSession Vi,
                                       ViConstString PhaseName);
```

Parameters

Inputs	**Description**	**Base Type**
`Vi`	Instrument handle	`ViSession`
`PhaseName`	The name of the output on which to abort the current ramp.	`ViConstString`

Return Values

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

14.4 IviACPwrCurrentRamp Behavior Model

The IviACPwrCurrentRamp Extension Group follows the behavior model of the IviACPwrBase capability group. The only modification to the behavior model from the IviACPwrBase capability group is the ability to ramp the output current.

When the Ramp Current function returns immediately after the ramp has started, the user should only call the Query Current Ramp Busy or Abort Current Ramp functions or read the Current Ramp Busy attribute while the ramp is in progress. Calling any other function or reading or writing any other attribute may abort the ramp or cause other vendor specific instrument behavior. After the current ramp completes or is aborted, the driver must invalidate the Current Limit attribute in the IviACPwr capability group

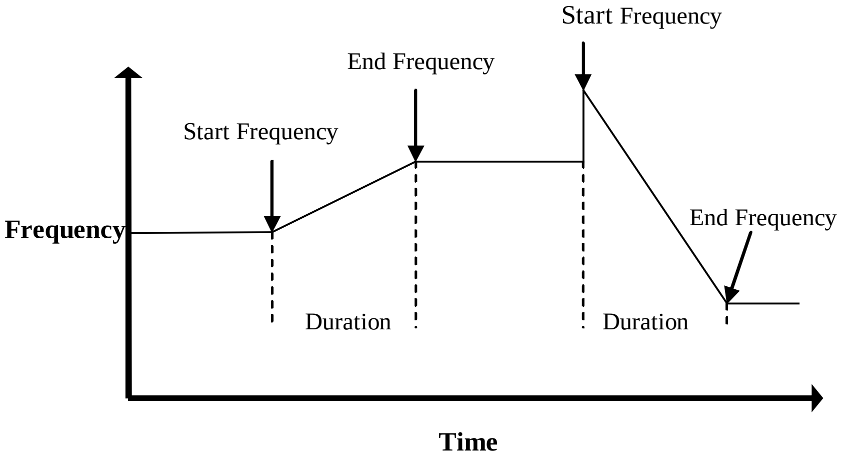
The actual current that appears at the power source output during a current ramp assumes the power source is operating in constant current mode. In other words, the output voltage during the current ramp is below the Voltage Level attribute value in the IviACPwrBase capability group. If the power source operates in constant voltage mode during any portion of the ramp, the output current is reduced to maintain the voltage level value but the actual ramp duration is not affected.

15 IviACPwrFrequencyRamp Extension Group

15.1 IviACPwrFrequencyRamp Overview

The IviACPwrFrequencyRamp extension group defines extensions for AC Power Sources with the capability to configure a linear Frequency ramp starting and ending Frequency in Hertz and duration in seconds.

The Start Frequency value takes effect immediately then the frequency ramps linearly to the End Frequency value for the duration specified.



This figure above shows 2 frequency ramps. The Start Frequency of the first ramp was set to the same value as the Frequency attribute in IviACPwr base. The Start Frequency of the second ramp was set to a value higher than the present Frequency attribute, causing an immediate change in output Frequency before linearly ramping to the End Frequency value.

15.2 IviACPwrFrequencyRamp Attributes

The IviACPwrFrequencyRamp extension group defines the following Attributes:

- Frequency Ramp Busy

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in section 16, *IviACPwr Attribute ID Definitions*.

15.2.1 Frequency Ramp Busy

Data Type	Access	Applies to	Coercion	High Level Functions
`ViBoolean`	`RO`	N/A	None	Query Frequency Ramp Busy (IVI-C only)

COM Property Name

`OutputPhases.FrequencyRampBusy`

COM Enumeration Name

`N/A`

C Constant Name

`IVIACPWR_ATTR_FREQUENCY_RAMP_BUSY`

Description

Indicates if a frequency ramp is in progress (busy).

Defined Values

Name	***Description***		
		Language	***Identifier***
True	Frequency ramp in progress.		
		C	`VI_TRUE`
		COM	`True`
False	Frequency ramp not in progress.		
		C	`VI_FALSE`
		COM	`False`

15.3 IviACPwrFrequencyRamp Functions

The IviACPwrFrequencyRamp extension defines the following functions:

- Ramp Frequency
- Query Frequency Ramp Busy (IVI-C only)
- Abort Frequency Ramp

This section describes the behavior and requirements of the functions.

15.3.1 Ramp Frequency

Description

Starts a the linear Frequency ramp then waits for it to complete. It specifies the starting and ending frequency in Hertz and duration in seconds. Call the Query Frequency Ramp Busy function to determine when the ramp completes or call the Abort Frequency Ramp function to stop the frequency ramp.

The starting and ending ramp frequencies must not exceed the values of the present frequency range. Refer to the Frequency Range attribute in the IviACPwrBase capability group.

COM Method Prototype

```
HRESULT OutputPhases.RampFrequency (
                              [in] double StartFrequency);
                              [in] double EndFrequency);
                              [in] double Duration);
```

C Prototype

```
ViStatus IviACPwr_RampFrequency (ViSession Vi,
                                         ViReal64 StartFrequency);
                                         ViReal64 EndFrequency);
                                         ViReal64 Duration);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
StartFrequency	Specifies the power source’s starting output frequency in Hertz.	ViReal64
EndFrequency	Specifies the power source’s ending output frequency in Hertz.	ViReal64
Duration	Duration of the Frequency ramp in seconds.	ViReal64

Return Values

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

Compliance Notes

Instruments that must complete the frequency ramp before returning are allowed to be compliant with the IviACPwrFrequencyRamp extension group. For these instruments, the Ramp Frequency function waits for the ramp to complete before returning and the Query Frequency Ramp Busy function always returns False.

15.3.2 Query Frequency Ramp Busy (IVI-C only)

Description

Returns if a frequency ramp is in progress. Call this function after calling the Ramp Frequency function to determine when the ramp completes.

COM Method Prototype

`N/A`

(use the `OutputPhases.FrequencyRampBusy` property)

C Prototype

```
ViStatus IviACPwr_QueryFrequencyRampBusy (ViSession Vi,
                                          ViBoolean *Busy);
```

Parameters

Inputs	**Description**	**Base Type**
`Vi`	Instrument handle	`ViSession`

Outputs	**Description**	**Base Type**
`Busy`	Returns True if a frequency ramp is in progress. Returns False when a Frequency Ramp completes or is not in progress.	`ViBoolean`

Defined Values for Busy Parameter

Name	***Description***		
		Language	***Identifier***
True	Frequency ramp in progress.		
		C	`VI_TRUE`
		COM	`True`
False	Frequency ramp not in progress		
		C	`VI_FALSE`
		COM	`False`

15.3.1 Abort Frequency Ramp

Description

Aborts a frequency ramp. If a frequency ramp is not in progress this function does nothing.

COM Method Prototype

```
HRESULT OutputPhases.AbortFrequencyRamp();
```

C Prototype

```
ViStatus IviACPwr_AbortFrequencyRamp (ViSession Vi);
```

Parameters

Inputs	Description	Base Type
`Vi`	Instrument handle	`ViSession`

Return Values

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

15.4 IviACPwrFrequencyRamp Behavior Model

The IviACPwrFrequencyRamp Extension Group follows the behavior model of the IviACPwrBase capability group. The only modification to the behavior model from the IviACPwrBase capability group is the ability to configure a frequency ramp.

When the Ramp Frequency function returns immediately after the ramp has started, the user should only call the Query Frequency Ramp Busy or Abort Frequency Ramp functions or read the Frequency Ramp Busy attribute while the ramp is in progress. Calling any other function or reading or writing any other attribute may abort the ramp or cause other vendor specific instrument behavior. After the frequency ramp completes or is aborted, the driver must invalidate the Frequency attribute in the IviACPwr capability group.

16 IviACPwr Attribute ID Definitions

The following table defines the ID value for all IviACPwr class attributes.

Table 16-1. IviACPwr Attributes ID Values

Attribute Name	ID Definition
IVIACPWR_ATTR_FREQUENCY	IVI_CLASS_ATTR_BASE+ 1
IVIACPWR_ATTR_NUM_PHASES	IVI_CLASS_ATTR_BASE+ 2
IVIACPWR_ATTR_NUM_FREQUENCY_RANGES	IVI_CLASS_ATTR_BASE+ 3
IVIACPWR_ATTR_FREQUENCY_RANGE	IVI_CLASS_ATTR_BASE+ 4
IVIACPWR_ATTR_NUM_WAVEFORMS_MAX	IVI_CLASS_ATTR_BASE+ 5
IVIACPWR_ATTR_NUM_OPTIMAL_DATA_POINTS	IVI_CLASS_ATTR_BASE+ 6
IVIACPWR_ATTR_FIXED_WAVEFORM_COUNT	IVI_CLASS_ATTR_BASE+ 7
IVIACPWR_ATTR_USER_WAVEFORM_COUNT	IVI_CLASS_ATTR_BASE+ 8
IVIACPWR_ATTR_AVAILABLE_WAVEFORM_COUNT	IVI_CLASS_ATTR_BASE+ 9
IVIACPWR_ATTR_EXTERNAL_SYNC_ENABLED	IVI_CLASS_ATTR_BASE+ 10
IVIACPWR_ATTR_EXTERNAL_SYNC_PHASE_OFFSET	IVI_CLASS_ATTR_BASE+ 11
IVIACPWR_ATTR_EXTERNAL_SYNC_LOCKED	IVI_CLASS_ATTR_BASE+ 12
IVIACPWR_ATTR_CURRENT_LIMIT	IVI_CLASS_ATTR_BASE+ 13
IVIACPWR_ATTR_OUTPUT_ENABLED	IVI_CLASS_ATTR_BASE+ 14
IVIACPWR_ATTR_NUM_VOLTAGE_RANGES	IVI_CLASS_ATTR_BASE+ 15
IVIACPWR_ATTR_VOLTAGE_LEVEL	IVI_CLASS_ATTR_BASE+ 16
IVIACPWR_ATTR_WAVEFORM	IVI_CLASS_ATTR_BASE+ 17
IVIACPWR_ATTR_VOLTAGE_RANGE	IVI_CLASS_ATTR_BASE+ 18
IVIACPWR_ATTR_PHASE_ANGLE	IVI_CLASS_ATTR_BASE+ 19
IVIACPWR_ATTR_OUTPUT_IMPEDANCE_INDUCTIVE	IVI_CLASS_ATTR_BASE+ 20
IVIACPWR_ATTR_OUTPUT_IMPEDANCE_ENABLED	IVI_CLASS_ATTR_BASE+ 21
IVIACPWR_ATTR_OUTPUT_IMPEDANCE_RESISTIVE	IVI_CLASS_ATTR_BASE+ 22
IVIACPWR_ATTR_CURRENT_PROTECTION_DELAY	IVI_CLASS_ATTR_BASE+ 23
IVIACPWR_ATTR_CURRENT_PROTECTION_ENABLED	IVI_CLASS_ATTR_BASE+ 24
IVIACPWR_ATTR_CURRENT_PROTECTION_THRESHOLD	IVI_CLASS_ATTR_BASE+ 25
IVIACPWR_ATTR_CURRENT_PROTECTION_TRIPPED	IVI_CLASS_ATTR_BASE+ 26
IVIACPWR_ATTR_OVER_VOLTAGE_PROTECTION_ENABLED	IVI_CLASS_ATTR_BASE+ 27
IVIACPWR_ATTR_OVER_VOLTAGE_PROTECTION_LIMIT	IVI_CLASS_ATTR_BASE+ 28
IVIACPWR_ATTR_UNDER_VOLTAGE_PROTECTION_ENABLED	IVI_CLASS_ATTR_BASE+ 29
IVIACPWR_ATTR_UNDER_VOLTAGE_PROTECTION_LIMIT	IVI_CLASS_ATTR_BASE+ 30
IVIACPWR_ATTR_VOLTAGE_PROTECTION_TRIPPED	IVI_CLASS_ATTR_BASE+ 31
IVIACPWR_ATTR_DC_MODE	IVI_CLASS_ATTR_BASE+ 32
IVIACPWR_ATTR_DC_VOLTAGE_LEVEL	IVI_CLASS_ATTR_BASE+ 33
IVIACPWR_ATTR_DC_NUM_RANGES	IVI_CLASS_ATTR_BASE+ 35
IVIACPWR_ATTR_OUTPUT_PHASE_COUNT	IVI_CLASS_ATTR_BASE+ 36
IVIACPWR_ATTR_VOLTAGE_RAMP_BUSY	IVI_CLASS_ATTR_BASE+ 37

Table 16-1. IviACPwr Attributes ID Values

Attribute Name	ID Definition
IVIACPWR_ATTR_CURRENT_RAMP_BUSY	IVI_CLASS_ATTR_BASE+ 38
IVIACPWR_ATTR_FREQUENCY_RAMP_BUSY	IVI_CLASS_ATTR_BASE+ 39
IVIACPWR_ATTR_DC_RANGE_MIN	IVI_CLASS_ATTR_BASE+ 40
IVIACPWR_ATTR_DC_RANGE_MAX	IVI_CLASS_ATTR_BASE+ 41

17 IviACPwr Attribute Value Definitions

This section specifies the actual value for each defined attribute value.

DC Mode

Value Name	*Language*	*Identifier*	*Actual Value*
AC Only	C	`IVIACPWR_VAL_MODE_AC_ONLY`	`0`
	COM	`IviACPwrDCGenerationModeACOnly`	`0`
DC Only	C	`IVIACPWR_VAL_MODE_DC_ONLY`	`1`
	COM	`IviACPwrDCGenerationModeDCOnly`	`1`
AC Plus DC	C	`IVIACPWR_VAL_MODE_AC_DC`	`2`
	COM	`IviACPwrDCGenerationModeACPlusDC`	`2`
DC Mode Class Extension Base	C	`IVIACPWR_VAL_DC_MODE_CLASS_EXT_BASE`	`500`
DC Mode Specific Extension Base	C	`IVIACPWR_VAL_DC_MODE_SPECIFIC_EXT_BASE`	`1000`

18 IviACPwr Function Parameter Value Definitions

This section specifies the actual values for each function parameter that defines values.

Fetch Measurement

Parameter: `MeasurementType`

COM Enumeration Name: `IviACPwrMeasurementTypeEnum`

Value Name	*Language*	*Identifier*	*Actual Value*
Voltage RMS L-N	C	`IVIACPWR_VAL_MEASURE_VOLTAGE_RMS_LN`	`0`
	COM	`IviACPwrMeasureVoltageRMS`	`0`
Current RMS	C	`IVIACPWR_VAL_MEASURE_CURRENT_RMS`	`1`
	COM	`IviACPwrMeasureCurrentRMS`	`1`
Frequency	C	`IVIACPWR_VAL_MEASURE_FREQUENCY`	`2`
	COM	`IviACPwrMeasureFrequency`	`2`
Voltage DC	C	`IVIACPWR_VAL_MEASURE_VOLTAGE_DC`	`3`
	COM	`IviACPwrMeasureVoltageDC`	`3`
Current DC	C	`IVIACPWR_VAL_MEASURE_CURRENT_DC`	`4`
	COM	`IviACPwrMeasureCurrentDC`	`4`
Power Factor	C	`IVIACPWR_VAL_MEASURE_POWER_FACTOR`	`5`
	COM	`IviACPwrMeasurePowerFactor`	`5`
Crest Factor	C	`IVIACPWR_VAL_MEASURE_CREST_FACTOR`	`6`
	COM	`IviACPwrMeasureCrestFactor`	`6`
Current Peak	C	`IVIACPWR_VAL_MEASURE_CURRENT_PEAK`	`7`
	COM	`IviACPwrMeasureCurrentPeak`	`7`
Power VA	C	`IVIACPWR_VAL_MEASURE_POWER_VA`	`8`
	COM	`IviACPwrMeasurePowerVA`	`8`
Real Power	C	`IVIACPWR_VAL_MEASURE_POWER_REAL`	`9`
	COM	`IviACPwrMeasurePowerReal`	`9`
Power DC	C	`IVIACPWR_VAL_MEASURE_POWER_DC`	`10`
	COM	`IviACPwrMeasurePowerDC`	`10`
Phase Angle	C	`IVIACPWR_VAL_MEASURE_PHASE_ANGLE`	`11`
	COM	`IviACPwrMeasurePhaseAngle`	`11`
Voltage RMS L-L	C	`IVIACPWR_VAL_MEASURE_VOLTAGE_RMS_LL`	`12`
	COM	`IviACPwrMeasureVoltageRMSLineToLine`	`12`

Current OHD	C	`IVIACPWR_VAL_MEASURE_CURRENT_OHD`	13
	COM	`IviACPwrMeasureCurrentOHD`	13
Current EHD	C	`IVIACPWR_VAL_MEASURE_CURRENT_EHD`	14
	COM	`IviACPwrMeasureCurrentEHD`	14
Current THD	C	`IVIACPWR_VAL_MEASURE_CURRENT_THD`	15
	COM	`IviACPwrMeasureCurrentTHD`	15
Voltage OHD	C	`IVIACPWR_VAL_MEASURE_VOLTAGE_OHD`	16
	COM	`IviACPwrMeasureVoltageOHD`	16
Voltage EHD	C	`IVIACPWR_VAL_MEASURE_VOLTAGE_EHD`	17
	COM	`IviACPwrMeasureVoltageEHD`	17
Voltage THD	C	`IVIACPWR_VAL_MEASURE_VOLTAGE_THD`	18
	COM	`IviACPwrMeasureVoltageTHD`	18

Fetch Measurement Array

Parameter: `MeasurementType`

COM Enumeration Name: `IviACPwrMeasurementTypeArrayEnum`

Value Name	***Language***	***Identifier***	***Actual Value***
Array Current Harmonic Phase	C	`IVIACPWR_VAL_MEASURE_ARRAY_CURRENT_HARMONIC_PHASE`	0
	COM	`IviACPwrMeasureArrayCurrentHarmonicPhase`	0
Array Current Harmonic Magnitude Absolute	C	`IVIACPWR_VAL_MEASURE_ARRAY_CURRENT_HARMONIC_ABS`	1
	COM	`IviACPwrMeasureArrayCurrentHarmonicAbs`	1
Array Current Harmonic Magnitude Percent Fundamental	C	`IVIACPWR_VAL_MEASURE_ARRAY_CURRENT_HARMONIC_PCT`	2
	COM	`IviACPwrMeasureArrayCurrentHarmonicPct`	2
Array Voltage Harmonic Phase	C	`IVIACPWR_VAL_MEASURE_ARRAY_VOLTAGE_HARMONIC_PHASE`	3
	COM	`IviACPwrMeasureArrayVoltageHarmonicPhase`	3
Array Voltage Harmonic Magnitude Absolute	C	`IVIACPWR_VAL_MEASURE_ARRAY_VOLTAGE_HARMONIC_ABS`	4
	COM	`IviACPwrMeasureArrayVoltageHarmonicAbs`	4
Array Voltage Harmonic Magnitude Percent Fundamental	C	`IVIACPWR_VAL_MEASURE_ARRAY_VOLTAGE_HARMONIC_PCT`	5
	COM	`IviACPwrMeasureArrayVoltageHarmonicPct`	5
Array Current Cycle	C	`IVIACPWR_VAL_MEASURE_ARRAY_CURRENT_CYCLE`	6
	COM	`IviACPwrMeasureArrayCurrentCycle`	6
Array Voltage Cycle	C	`IVIACPWR_VAL_MEASURE_ARRAY_VOLTAGE_CYCLE`	7
	COM	`IviACPwrMeasureArrayVoltageCycle`	7

DC Generation

Parameter: Mode

COM Enumeration Name: IviACPwrDCGenerationModeEnum

Value Name	*Language*	*Identifier*	*Actual Value*
AC Only	C	IVIACPWR_VAL_MODE_AC_ONLY	0
	COM	IviACPwrModeACOnly	0
DC Only	C	IVIACPWR_VAL_MODE_DC_ONLY	1
	COM	IviACPwrModeDCOnly	1
AC Plus DC	C	IVIACPWR_VAL_MODE_AC_DC	2
	COM	IviACPwrModeACPlusDC	2

Initiate Measurement

Parameter: Group

COM Enumeration Name: IviACPwrMeasurementGroupEnum

Value Name	*Language*	*Identifier*	*Actual Value*
Base	C	IVIACPWR_VAL_MEASUREMENT_GROUP_BASE	1
	COM	IviACPwrMeasurementGroupBase	1
Harmonic	C	IVIACPWR_VAL_MEASUREMENT_GROUP_HARMONIC	2
	COM	IviACPwrMeasurementGroupHarmonic	2
Distortion	C	IVIACPWR_VAL_MEASUREMENT_GROUP_DISTORTION	4
	COM	IviACPwrMeasurementGroupDistortion	4
Waveform	C	IVIACPWR_VAL_MEASUREMENT_GROUP_WAVEFORM	8
	COM	IviACPwrMeasurementGroupWaveform	8

Query Arbitrary Waveform Catalog

Parameter: `CatalogType`

COM Enumeration Name: `IviACPwrDCGenerationModeEnum`

Value Name	*Language*	*Identifier*	*Actual Value*
Catalog Fixed	C	IVIACPWR_VAL_WAVEFORM_CATALOG_FIXED	0
	COM	IviACPwrWaveformCatalogFixed	0
Catalog User	C	IVIACPWR_VAL_WAVEFORM_CATALOG_USER	1
	COM	IviACPwrWaveformCatalogUser	1
Catalog All	C	IVIACPWR_VAL_WAVEFORM_CATALOG_ALL	2
	COM	IviACPwrWaveformCatalogAll	2

19 Error and Completion Code Value Definitions

The IviACPwr class specification defines the additional status codes.

Error Name	*Language*	*Identifier*	*Actual Value*
All Phases Required	C	IVIACPWR_ERROR_ALL_PHASES_REQUIRED	0xBFFA2002
	COM	E_IVIACPWR_ALL_PHASES_REQUIRED	0x80042002
Current Protection Tripped	C	IVIACPWR_ERROR_CURRENT_PROTECTION	0xBFFA2003
	COM	E_IVIACPWR_CURRENT_PROTECTION	0x80042003
Duplicate Waveform Name	C	IVIACPWR_ERROR_DUPLICATE_WAVEFORM_NAME	0xBFFA2004
	COM	E_IVIACPWR_DUPLICATE_WAVEFORM_NAME	0x80042004
Invalid Waveform Name	C	IVIACPWR_ERROR_INVALID_WAVEFORM_NAME	0xBFFA2005
	COM	E_IVIACPWR_INVALID_WAVEFORM_NAME	0x80042005
Measurement Not Initiated	C	IVIACPWR_ERROR_MEASUREMENT_NOT_INITIATED	0xBFFA2006
	COM	E_IVIACPWR_MEASUREMENT_NOT_INITIATED	0x80042006
Measurement Not Supported	C	IVIACPWR_ERROR_MEASUREMENT_NOT_SUPPORTED	0xBFFA2007
	COM	E_IVIACPWR_MEASUREMENT_NOT_SUPPORTED	0x80042007
No Waveforms Available	C	IVIACPWR_ERROR_NO_WAVEFORMS_AVAILABLE	0xBFFA2008
	COM	E_IVIACPWR_NO_WAVEFORMS_AVAILABLE	0x80042008
Phase Values Different	C	IVIACPWR_ERROR_PHASE_VALUES_DIFFERENT	0xBFFA200A
	COM	E_IVIACPWR_PHASE_VALUES_DIFFERENT	0x8004200A
Unsupported Measurement Group	C	IVIACPWR_ERROR_UNSUPPORTED_MEASUREMENT_GROUP	0xBFFA200B
	COM	E_IVIACPWR_UNSUPPORTED_MEASUREMENT_GROUP	0x8004200B
Voltage Protection Tripped	C	IVIACPWR_ERROR_VOLTAGE_PROTECTION	0xBFFA200C
	COM	E_IVIACPWR_VOLTAGE_PROTECTION	0x8004200C
Waveform In Use	C	IVIACPWR_ERROR_WAVEFORM_IN_USE	0xBFFA200E
	COM	E_IVIACPWR_WAVEFORM_IN_USE	0x8004200E
Waveform Not Found	C	IVIACPWR_ERROR_WAVEFORM_NOT_FOUND	0xBFFA200F
	COM	E_IVIACPWR_WAVEFORM_NOT_FOUND	0x8004200F
Waveform Reserved	C	IVIACPWR_ERROR_WAVEFORM_RESERVED	0xBFFA2010
	COM	E_IVIACPWR_WAVEFORM_RESERVED	0x80042010
Waveform DC Offset	C	IVIACPWR_ERROR_WAVEFORM_DC_OFFSET	0xBFFA200D
	COM	E_IVIACPWR_WAVEFORM_DC_OFFSET	0x8004200D
All Phases Not Supported	C	IVIACPWR_ERROR_ALL_PHASES_NOT_SUPPORTED	0xBFFA2001
	COM	E_IVIACPWR_ALL_PHASES_NOT_SUPPORTED	0x80042001
Output Enabled	C	IVIACPWR_ERROR_OUTPUT_ENABLED	0xBFFA2009
	COM	E_IVIACPWR_OUTPUT_ENABLED	0x80042009

20 IviACPwr Hierarchies

20.1 IviACPwr COM Hierarchy

The full IviACPwr COM Hierarchy includes the Inherent Capabilities Hierarchy as defined in Section 4.1, *COM Inherent Capabilities* of *IVI-3.2: Inherent Capabilities Specification*. To avoid redundancy, the Inherent Capabilities are omitted here.

Table 20-1. IviACPwr COM Hierarchy

COM Interface Hierarchy	Generic Name	Type
OutputPhases		
Count	Output Phase Count	P
Name	Output Phase Name	P
Frequency	Frequency	P
FrequencyRange	Frequency Range	P
NumFrequencyRanges	Number of Frequency Ranges	P
NumPhases	Number of Phases	P
FrequencyRampBusy	Frequency Ramp Busy	P
QueryFrequencyRangeCapabilities	Query Frequency Range Capabilities	M
InitiateMeasurement	Initiate Measurement	M
RampFrequency	Ramp Frequency	M
AbortFrequencyRamp	Abort Frequency Ramp	M
Item		
Enabled	Output Enabled	P
VoltageLevel	Voltage Level	P
Waveform	Waveform	P
PhaseAngle	Phase Angle	P
VoltageRange	Voltage Range	P
NumVoltageRanges	Number of Voltage Ranges	P
CurrentLimit	Current Limit	P
VoltageRampBusy	Voltage Ramp Busy	P
CurrentRampBusy	Current Ramp Busy	P
FetchMeasurement	Fetch Measurement	M
FetchMeasurementArray	Fetch Measurement Array	M
QueryVoltageRangeCapabilities	Query Voltage Range Capabilities	M

RampVoltage	Ramp Voltage	M
AbortVoltageRamp	Abort Voltage Ramp	M
RampCurrent	Ramp Current	M
AbortCurrentRamp	Abort Current Ramp	M
Impedance		
Resistive	Output Impedance Resistive	P
Inductive	Output Impedance Inductive	P
Enabled	Output Impedance Enabled	P
Configure	Configure Output Impedance	M
QueryCapabilities	Query Output Impedance Capabilities	M
Current Protection		
Enabled	Current Protection Enabled	P
Delay	Current Protection Delay	P
Threshold	Current Protection Threshold	P
Tripped	Current Protection Tripped	P
Configure	Configure Current Protection	M
Reset	Reset Current Protection	M
Voltage Protection		
OverEnabled	Over Voltage Enabled	P
OverLimit	Over Voltage Limit	P
UnderEnabled	Under Voltage Enabled	P
UnderLimit	Under Voltage Limit	P
Tripped	Voltage Protection Tripped	P
Configure	Configure Voltage Protection	M
Reset	Reset Voltage Protection	M
DC Generation		
Mode	DC Mode	P
DCVoltageLevel	DC Voltage Level	P
RangeMin	DC Range Min	P
RangeMax	DC Range Max	P
NumRanges	Number of DC Ranges	P
Configure	Configure DC	M
ConfigureRange	Configure DC Range	M
QueryCapabilities	Query DC Capabilities	M
External Sync		
Enabled	Enabled	P
PhaseOffset	Phase Offset	P
Locked	Locked	P
Configure	Configure	M

Arb Waveform		
NumWaveformsMax	Num Waveforms Max	P
NumOpimalDataPoints	Num Optimal Data Points	P
FixedWaveformCount	Fixed Waveform Count	P
UserWaveformCount	User Waveform Count	P
AvailableWaveformCount	Available Waveform Count	P
Clear	Clear Waveform	M
Write	Write Waveform	M
QueryCatalog	Query Waveform Catalog	M

20.1.1 IviACPwr COM Interfaces

In addition to implementing IVI inherent capabilities interfaces, IviACPwr interfaces contain interface reference properties for accessing the following IviACPwr interfaces:

- IIviACPwrOutputPhases
- IIviACPwrArbWaveform

The IIviACPwrOutputPhases interface contains methods and properties for accessing a collection of objects that implement the IIviACPwrOutputPhase interface. The IIviACPwrOutputPhases interface contains interface reference properties for accessing the following IviACPwr interfaces:

- IIviACPwrExternalSync

The IIviACPwrOutputPhase interface contains interface reference properties for accessing the following IviACPwr interfaces:

- IIviACPwrCurrentProtection
- IIviACPwrVoltageProtection
- IIviACPwrImpedance
- IIviACPwrDCGeneration

Table 20-2 IviACPwr Interface GUIDs lists the interfaces that this specification defines and their GUIDs.

Table 20-2 IviACPwr Interface GUIDs

Interface	GUID
IIviACPwr	{ 47ed5440-a398-11d4-ba58-000064657374}
IIviACPwrOutputPhases	{ 47ed5441-a398-11d4-ba58-000064657374}
IIviACPwrOutputPhase	{ 47ed5442-a398-11d4-ba58-000064657374}
IIviACPwrArbWaveform	{ 47ed5443-a398-11d4-ba58-000064657374}
IIviACPwrExternalSync	{ 47ed5444-a398-11d4-ba58-000064657374}
IIviACPwrCurrentProtection	{ 47ed5445-a398-11d4-ba58-000064657374}
IIviACPwrVoltageProtection	{ 47ed5446-a398-11d4-ba58-000064657374}
IIviACPwrImpedance	{ 47ed5447-a398-11d4-ba58-000064657374}
IIviACPwrDCGeneration	{ 47ed5448-a398-11d4-ba58-000064657374}

20.1.2 IviACPwr COM Interface Reference Properties

Interface reference properties are used to navigate the IviACPwr COM hierarchy. This section describes the interface reference properties that the IIviACPwr interface defines.

20.1.2.1 OutputPhases

Data Type	Access
IIviACPwrOutputPhases*	RO

COM Property Name

`OutputPhases`

Description

Returns a pointer to the IIviACPwrOutputPhases interface.

20.1.2.2 OutputPhase

Data Type	Access
IIviACPwrOutputPhase*	RO

COM Property Name

`OutputPhase`

Description

Returns a pointer to the IIviACPwrOutputPhase interface.

20.1.2.3 Arb Waveform

Data Type	Access
IIviACPwrArbWaveform*	RO

COM Property Name

`ArbWaveform`

Description

Returns a pointer to the IIviACPwrArbWaveform interface.

20.1.2.4 External Sync

Data Type	Access
IIviACPwrExternalSync*	RO

COM Property Name

`ExternalSync`

Description

Returns a pointer to the IIviACPwrExternalSync interface.

20.1.2.5 Current Protection

Data Type	Access
IIviACPwrCurrentProtection*	RO

COM Property Name

`CurrentProtection`

Description

Returns a pointer to the IIviACPwrCurrentProtection interface.

20.1.2.6 Voltage Protection

Data Type	Access
IIviACPwrVoltageProtection*	RO

COM Property Name

`VoltageProtection`

Description

Returns a pointer to the IIviACPwrVoltageProtection interface.

20.1.2.7 Impedance

Data Type	Access
IIviACPwrImpedance*	RO

COM Property Name

`Impedance`

Description

Returns a pointer to the IIviACPwrImpedance interface.

20.1.2.8 DC Generation

Data Type	Access
IIviACPwrDCGeneration*	RO

COM Property Name

DCGeneration

Description

Returns a pointer to the IIviACPwrDCGeneration interface.

20.1.3 IviACPwr COM Category

The IviACPwr class COM Category shall be “IviACPwr”, and the Category ID (CATID) shall be {47ed5161-a398-11d4-ba58-000064657374}.

20.2 IviACPwr C Function Hierarchy

The IviACPwr class function hierarchy is shown in the following table. The full IviACPwr C Function Hierarchy includes the Inherent Capabilities Hierarchy as defined in Section 4.2, *C Inherent Capabilities* of *IVI-3.2: Inherent Capabilities Specification*. To avoid redundancy, the Inherent Capabilities are omitted here.

Table 20-3. IviACPwr C Function Hierarchy

Name or Class	Function Name
Configuration	
Configure Voltage Level	IviACPwr_ConfigureVoltageLevel
Configure Current Limit	IviACPwr_ConfigureCurrentLimit
Configure Output Enabled	IviACPwr_ConfigureOutputEnabled
Configure Waveform	IviACPwr_ConfigureWaveform
Configure Voltage Range	IviACPwr_ConfigureVoltageRange
Query Voltage Range Capabilities	IviACPwr_QueryVoltageRangeCapabilities
Get Output Phase Name	IviACPwr_GetOutputPhaseName
Configure Frequency	IviACPwr_ConfigureFrequency
Configure Frequency Range	IviACPwr_ConfigureFrequencyRange
Query Frequency Range Capabilities	IviACPwr_QueryFrequencyRangeCapabilities
Phase Angle	
Configure Phase Angle	IviACPwr_ConfigurePhaseAngle
Arb Waveform	
Clear Arbitrary Waveform	IviACPwr_ClearArbWaveform
Query Arbitrary Waveform Catalog	IviACPwr_QueryArbWaveformCatalog
Write Arbitrary Waveform	IviACPwr_WriteArbWaveform
Impedance	
Configure Output Impedance	IviACPwr_ConfigureOutputImpedance
Query Output Impedance Capabilities	IviACPwr_QueryOutputImpedanceCapabilities
Voltage Protection	
Configure Voltage Protection	IviACPwr_ConfigureVoltageProtection
Query Voltage Protection Tripped	IviACPwr_QueryVoltageProtectionTripped
Reset Voltage Protection	IviACPwr_ResetVoltageProtection
Current Protection	
Configure Current Protection	IviACPwr_ConfigureCurrentProtection
Query Current Protection Tripped	IviACPwr_QueryCurrentProtectionTripped
Reset Current Protection	IviACPwr_ResetCurrentProtection
DC Generation	
Configure DC	IviACPwr_ConfigureDC
Configure DC Range	IviACPwr_ConfigureDCRange
Query DC Capabilities	IviACPwr_QueryDCCapabilities
External Sync	
Configure External Sync	IviACPwr_ConfigureExternalSync
Query External Sync Locked	IviACPwr_QueryExternalSyncLocked

Action/Status	
Voltage Ramp	
Ramp Voltage	IviACPwr_RampVoltage
Abort Voltage Ramp	IviACPwr_AbortVoltageRamp
Query Voltage Ramp Busy	IviACPwr_QueryVoltageRampBusy
Current Ramp	
Ramp Current	IviACPwr_RampCurrent
Abort Current Ramp	IviACPwr_AbortCurrentRamp
Query Current Ramp Busy	IviACPwr_QueryCurrentRampBusy
Frequency Ramp	
Ramp Frequency	IviACPwr_RampFrequency
Abort Frequency Ramp	IviACPwr_AbortFrequencyRamp
Query Frequency Ramp Busy	IviACPwr_QueryFrequencyRampBusy
Measurement	
Initiate Measurement	IviACPwr_InitiateMeasurement
Fetch Measurement	IviACPwr_FetchMeasurement
Fetch Measurement Array	IviACPwr_FetchMeasurementArray

20.3 IviACPwr C Attribute Hierarchy

The IviACPwr class attribute hierarchy is shown in the following table. The full IviACPwr C Attribute Hierarchy includes the Inherent Capabilities Hierarchy as defined in Section 4.2, *C Inherent Capabilities* of *IVI-3.2: Inherent Capabilities Specification*. To avoid redundancy, the Inherent Capabilities are omitted here.

Table 20-4. IviACPwr C Attributes Hierarchy

Category or Generic Attribute Name	C Defined Constant
Output	
Current Limit	IVIACPWR_ATTR_CURRENT_LIMIT
Waveform	IVIACPWR_ATTR_WAVEFORM
Output Enabled	IVIACPWR_ATTR_OUTPUT_ENABLED
Voltage Range	IVIACPWR_ATTR_VOLTAGE_RANGE
Num Voltage Ranges	IVIACPWR_ATTR_NUM_VOLTAGE_RANGES
Voltage Level	IVIACPWR_ATTR_VOLTAGE_LEVEL
Phase Angle	
Phase Angle	IVIACPWR_ATTR_PHASE_ANGLE
Outputs	
Frequency	IVIACPWR_ATTR_FREQUENCY
Frequency Range	IVIACPWR_ATTR_FREQUENCY_RANGE
Num Frequency Ranges	IVIACPWR_ATTR_NUM_FREQUENCY_RANGES
Num Phases	IVIACPWR_ATTR_NUM_PHASES
Output Phase Count	IVIACPWR_ATTR_OUTPUT_PHASE_COUNT
Voltage Ramp	
Voltage Ramp Busy	IVIACPWR_ATTR_VOLTAGE_RAMP_BUSY
Current Ramp	
Current Ramp Busy	IVIACPWR_ATTR_CURRENT_RAMP_BUSY
Frequency Ramp	
Frequency Ramp Busy	IVIACPWR_ATTR_FREQUENCY_RAMP_BUSY
ArbWaveform	
Num Waveforms Max	IVIACPWR_ATTR_NUM_WAVEFORMS_MAX
Num Optimal Data Points	IVIACPWR_ATTR_NUM_OPTIMAL_DATA_POINTS
Fixed Waveform Count	IVIACPWR_ATTR_FIXED_WAVEFORM_COUNT
User Waveform Count	IVIACPWR_ATTR_USER_WAVEFORM_COUNT
Available Waveform Count	IVIACPWR_ATTR_AVAILABLE_WAVEFORM_COUNT
Impedance	
Impedance Inductive	IVIACPWR_ATTR_OUTPUT_IMPEDANCE_INDUCTIVE
Impedance Resistive	IVIACPWR_ATTR_OUTPUT_IMPEDANCE_RESISTIVE
Impedance Enabled	IVIACPWR_ATTR_OUTPUT_IMPEDANCE_ENABLED

Voltage Protection	
Over Voltage Protection Enabled	IVIACPWR_ATTR_OVER_VOLTAGE_PROTECTION_ENABLED
Over Voltage Protection Limit	IVIACPWR_ATTR_OVER_VOLTAGE_PROTECTION_LIMIT
Under Voltage Protection Enabled	IVIACPWR_ATTR_UNDER_VOLTAGE_PROTECTION_ENABLED
Under Voltage Protection Limit	IVIACPWR_ATTR_UNDER_VOLTAGE_PROTECTION_LIMIT
Voltage Protection Tripped	IVIACPWR_ATTR_VOLTAGE_PROTECTION_TRIPPED
Current Protection	
Current Protection Delay	IVIACPWR_ATTR_CURRENT_PROTECTION_DELAY
Current Protection Enabled	IVIACPWR_ATTR_CURRENT_PROTECTION_ENABLED
Current Protection Threshold	IVIACPWR_ATTR_CURRENT_PROTECTION_THRESHOLD
Current Protection Tripped	IVIACPWR_ATTR_CURRENT_PROTECTION_TRIPPED
DC Generation	
Mode	IVIACPWR_ATTR_DC_MODE
DCVoltageLevel	IVIACPWR_ATTR_DC_VOLTAGE_LEVEL
Range Min	IVIACPWR_ATTR_DC_RANGE_MIN
Range Max	IVIACPWR_ATTR_DC_RANGE_MAX
Num Ranges	IVIACPWR_ATTR_DC_NUM_RANGES
External Sync	
Enabled	IVIACPWR_ATTR_EXTERNAL_SYNC_EABLED
Phase Offset	IVIACPWR_ATTR_EXTERNAL_SYNC_PHASE_OFFSET
Locked	IVIACPWR_ATTR_EXTERNAL_SYNC_LOCKED

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 IviACPwr 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 Initialize or 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 must 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 IviACPwrMeasurement Extension Group

The IviACPwrMeasurement extension group affects the instrument behavior only when the user calls the measurement functions. Therefore, this specification does not recommend attribute values that disable the IviACPwrMeasurement extension group.

Disabling the IviACPwrPhase Extension Group

The Phase Angle attribute value in the IviACPwrPhase extension group for each output phase should be initialized to a known value based on the following equation:

Phase Angle = 360 – [(Phase Number – 1) * 360 / Number of Phases]

Where Phase Number is the 1 based output phase index and Number of Phases is the attribute defined in IviACPwrBase. For example, the initialized phase angle values for a three phase power source are shown in the following table.

Output Phase	Phase Angle Attribute Value
`2 (Phase B)`	240
`3 (Phase C)`	120

Disabling the IviACPwrExternalSync Extension Group

Attribute values that effectively disable the IviACPwrExternalSync extension group are shown in the following table.

Attribute	Value
`External Synchronization Enabled`	False

Disabling the IviACPwrCurrentProtection Extension Group

Attribute values that effectively disable the IviACPwrCurrentProtection extension group are shown in the following table.

Attribute	Value
`Current Protection Enabled`	False

The instrument driver must also make sure the AC power source is not in a current protection state

Disabling the IviACPwrVoltageProtection Extension Group

Attribute values that effectively disable the IviACPwrVoltageProtection extension group are shown in the following table.

Attribute	Value
`Under Voltage Enabled`	False
`Over Voltage Enabled`	False

The instrument driver must also make sure the AC power source is not in a voltage protection state

Disabling the IviACPwrArbWaveform Extension Group

The IviACPwrArbWaveform extension group affects the instrument behavior only when the Waveform attribute in IviACPwrBase group is set to a value other than ‘Sine’. Therefore, this specification recommends the attribute value for all output phases be set as shown in the following table.

Capability Group	Attribute	Value
`Base`	`Waveform`	‘Sine’

Disabling the IviACPwrImpedance Extension Group

Attribute values that effectively disable the IviACPwrImpedance extension group are shown in the following table.

Attribute	Value
`Output Impedance Enabled`	False

Disabling the IviACPwrDCGeneration Extension Group

Attribute values that effectively disable the IviACPwrDCGeneration extension group are shown in the following table.

Attribute	Value
`DC Mode`	AC Only

Disabling the IviACPwrVoltageRamp Extension Group

The IviACPwrVoltageRamp extension group affects the instrument behavior only when the user calls the voltage ramp functions. Therefore, this specification does not recommend attribute values that disable the IviACPwrVoltageRamp extension group.

Disabling the IviACPwrCurrentRamp Extension Group

The IviACPwrCurrentRamp extension group affects the instrument behavior only when the user calls the current ramp functions. Therefore, this specification does not recommend attribute values that disable the IviACPwrCurrentRamp extension group.

Disabling the IviACPwrFrequencyRamp Extension Group

The IviACPwrFrequencyRamp extension group affects the instrument behavior only when the user calls the frequency ramp functions. Therefore, this specification does not recommend attribute values that disable the IviACPwrFrequencyRamp extension group.

A.3 Special Repeated Capability Identifier

All attributes that apply to an OutputPhase and all functions that have a phase name parameter in the following IviACPwr groups must support the "AllPhases" repeated capability name:

- IviACPwrBase
- IviACPwrCurrentProtection
- IviACPwrVoltageProtection
- IviACPwrImpedance
- IviACPwrDCGeneration
- IviACPwrVoltageRamp
- IviACPwrCurrentRamp

These groups may or may not support repeated capability names for a specific phase. If they don't, they must return the IVIACPWR_ERROR_ALL_PHASES_REQUIRED completion code.

A.4 Current and Voltage Protection

When a current or voltage protection condition occurs (refer to the IviACPwrCurrentProtection and IviACPwrVoltageProtection extension groups), the Output Enabled attribute in IviACPwrBase is set to False. If you attempt to set Output Enabled to True prior to resetting the voltage or current protection state, the driver shall return `IVIACPWR_ERROR CURRENT_PROTECTION or IVIACPWR_ERROR VOLTAGE_PROTECTION.`

A.5 Operations that Require the Output be Disabled

Certain operations may require the power source output be reconfigured, requiring that the power source output first be disabled. This can adversely affect automated testing because it may require that the load be reinitialized. The table below shows examples of actions that may require that output power be disabled before they are performed.

Capability Group	Action
IviACPwrBase	Changing the voltage or frequency range
IviACPwrImpedance	Enabling or disabling the output impedance or changing the resistance or inductance value.
IviACPwrArbWaveform	Writing a waveform
IviACPwrDCGeneration	Changing the DC mode, DC Voltage Range, or DC Voltage Level

If the power source is presently supplying output power (Output Enabled attribute is True) and the user attempts to perform these or other actions that require output power be disabled, it is the responsibility of the driver to return the IVIACPWR_ERROR_OUTPUT_ENABLED completion code. The user is responsible for setting the Output Enabled attribute to False then reattempt the operation.

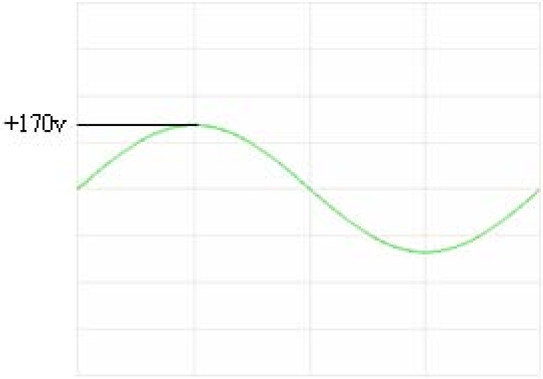
Operations requiring the output be disabled may also affect other attributes. For example, changing the voltage or frequency range may also invalidate or reset other attributes as described below:

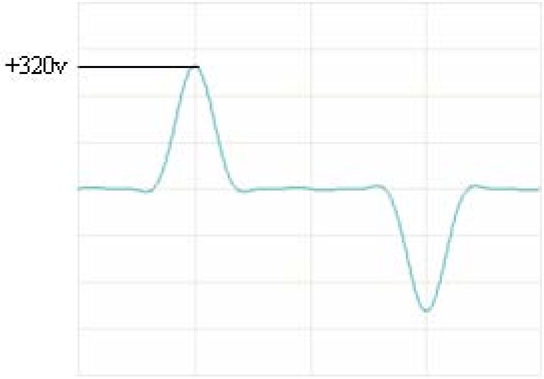
- Changing the voltage or frequency range may cause the previous Voltage Level , Frequency or Current Limit attribute values in IviACPwrBase to be outside the new range.
- Changing the voltage or frequency range may cause previous attribute values in IviACPwrImpedance to be outside the impedance capabilities of the new range.
- Changing the voltage or frequency range may cause previous attribute values in IviACPwrDCGeneration to be outside the capabilities of the new range.

It is the responsibility of a particular instrument driver to document which attributes must be checked following an operation requiring the output be disabled. It is the responsibility of the user to check and set these attributes to the desired values before setting the Output Enabled attribute to True.

A.6 Waveform RMS and Peak Voltage

The figures below show the power source output voltage with the Voltage Level attribute in IviACPwrBase set to 120. In order to achieve 120 volts RMS, the waveform on the left (a pure sine wave) requires a peak voltage of 170 volts and the waveform on the right requires a peak voltage of 320 volts. The peak voltage required to generate the user specified RMS voltage for a particular waveform may exceed the capabilities of the power source hardware. The driver must return the IVI_ERROR_VALUE_NOT_SUPPORTED completion code if the user attempts to set the Voltage Level attribute (or calls the Configure Voltage Level function) and the requested RMS voltage requires a peak voltage that exceeds the capabilities of the power source.





The Query Voltage Range Capabilities function in IviACPwrBase allows the user to determine the minimum and maximum RMS voltages supported by the power source for a particular voltage range and waveform. The driver must take into account the peak voltage limitations of the power source when returning the minimum and maximum RMS voltage for the Query Voltage Range Capabilities function.

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.

- Configure Output Enabled
- Configure Waveform
- Initiate Measurement

The user sets any of the attributes described in section *B.3 Interchangeability Checking Rules*.

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.

IviACPwrBase Capability Group

The driver performs interchangeability checking on the IviACPwrBase group only when the Output Enabled attribute is set to True.

The Current Limit, Voltage Level, Voltage Range, Frequency and Frequency Range attributes must be in a user defined state prior to setting the Output Enabled attribute to True.

When setting the Waveform attribute to a value other than ‘Sine’, the waveform must first be in a user defined state by calling the Write Arbitrary Waveform function in the IviACPwrArbWaveform capability group. Using a fixed vendor defined waveform other than ‘Sine’ must always generate an interchangeability warning because vendor defined waveform names and data may vary between instruments and vendors.

IviACPwrMeasurement Capability Group

The driver performs interchangeability checking on the IviACPwrMeasurement group when the user calls the Initiate Measurement function.

The driver returns an interchangeability warning if the instrument is not capable of performing all measurements in the requested measurement groups on the same cycle of the output.

IviACPwrPhase Capability Group

No additional interchangeability rules or exceptions are defined for the IviACPwrPhase capability group.

IviACPwrExternalSync Group

The driver performs interchangeability checking on the IviACPwrExternalSync group only when the External Synchronization Enabled attribute is set to True.

The External Synchronization Phase Offset attribute must be in a user defined state prior to setting the External Synchronization Enabled attribute to True.

IviACPwrCurrentProtection Capability Group

The driver performs interchangeability checking on the IviACPwrCurrentProtection group only when the Current Protection Enabled attribute is set to True.

The Current Protection Threshold and Current Protection Delay attributes must be in a user defined state prior to setting the Current Protection Enabled attribute to True.

IviACPwrVoltageProtection Capability Group

The driver performs interchangeability checking on the IviACPwrVoltageProtection group only when the Under Voltage Enabled or Over Voltage Enabled attribute is set to True.

The Under Voltage Limit attribute must be in a user defined state prior to setting the Under Voltage Enabled attribute to True.

The Over Voltage Limit attribute must be in a user defined state prior to setting the Over Voltage Enabled attribute to True.

IviACPwrArbWaveform Capability Group

The driver performs interchangeability checking on the IviACPwrArbWaveform group only when the user sets the Waveform attribute in the IviACPwrBase group to a value other than ‘Sine’. Waveforms other than ‘Sine’ must have been previously set to a user defined state by using the Write Arbitrary Waveform function in the IviACPwrArbWaveform group.

IviACPwrImpedance Capability Group

The driver performs interchangeability checking on the IviACPwrImpedance group only when the Output Impedance Enabled attribute is set to True.

The Output Impedance Resistive and Output Impedance Inductive attribute must be in a user defined state prior to setting the Output Impedance Enabled attribute to True.

IviACPwrDCGeneration Capability Group

The driver performs interchangeability checking on the IviACPwrDCGeneration group only when the DC Mode attribute is set to DC Only or AC Plus DC.

The DC Voltage Level attribute must be in a user defined state prior to setting the DC Mode attribute to DC Only or AC Plus DC.

IviACPwrVoltageRamp Capability Group

No additional interchangeability rules or exceptions are defined for the IviACPwrVoltageRamp capability group.

IviACPwrCurrentRamp Capability Group

No additional interchangeability rules or exceptions are defined for the IviACPwrCurrentRamp capability group.

IviACPwrFrequencyRamp Capability Group

No additional interchangeability rules or exceptions are defined for the IviACPwrFrequencyRamp capability group.