Interchangeable Virtual Instruments

IVI-4.4: IviDCPwr Class Specification

September 24, 2015 Edition
Revision 3.0

Important Information

The IviDCPwr Class Specification (IVI-4.4) 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.
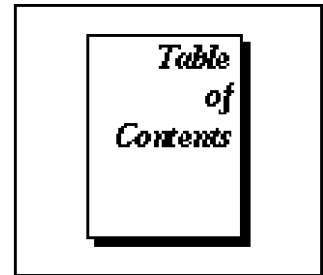
Warranty

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.

Appendix A   Specific Driver Development Guidelines................... 74

Appendix B.   Interchangeability Checking Rules........................... 77

IviDCPwr Class Specification

IviDCPwr Revision History

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

Table 1. IviDCPwr Class Specification Revisions

Revision Number	Date of Revision	Revision Notes
Revision 1.0 draft	October 1, 1999	First approved version. Original draft.
Revision 2.0a draft	April 31, 2001	First draft to include COM requirements
Revision 2.1vc1 draft	July 30, 2001	Voting candidate 1 review draft.
Revision 2.1vc2 draft	August 30, 2001	Voting candidate 2 review draft.
Revision 2.1vc3 draft	November, 2001	Voting candidate 3 review draft.
Revision 2.1vc5 draft	December, 2001	Voting candidate 5 review draft. Changes based on discussion at December IVI meeting.
Revision 2.0vc6 draft	January, 2002	Voting candidate 6 review draft. Changes based on comments from NI.
Revision 2.0vc7 draft	March, 2002	Added latest IDL.
Revision 2.0vc7 draft	March, 2002	`IVI_CLASS_PUBLIC_ATTR_BASE` changed to `IVI_CLASS_ATTR_BASE`.
Revision 2.0 Final	March, 2002	`Approved Specification`
Revision 2.0 Final	March, 2008	Editorial change to update the IVI Foundation contact information in the Important Information section to remove obsolete address information and refer only to the IVI Foundation web site.
Revision 2.0 Final	April 2009	Editorial change to update repeated capabilities section to include both qualified and unqualified repeated capability names.
Revision 3.0	June 9, 2010	Incorporated IVI.NET
Revision 3.0	August 25, 2011	Editorial IVI.NET change. Change references to process-wide locking to AppDomain-wide locking. Add an overload to the Create factory method that takes locking related parameters.
Revision 3.0	September 24, 2015	Editorial Change – Clarified the use of one-based index and COM, and zero-based index for .NET for repeated

Table 1. IviDCPwr Class Specification Revisions

		capabilities in section 4.2.9.

API Versions

Architecture	Drivers that comply with version 3.0 comply with all of the versions below
C	2.0, 3.0
COM	2.0, 3.0
.NET	3.0

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

1 Overview of the IviDCPwr Specification

1.1 Introduction

This specification defines the IVI class for DC power supplies. The IviDCPwr class is designed to support the typical DC power supply as well as common extended functionality found in more complex instruments. This section summarizes the *IviDCPwr 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:

- IviDCPwr Class Overview
- The definitions of terms and acronyms
- References

1.2 IviDCPwr Class Overview

This specification describes the IVI class for DC power supplies. The IviDCPwr class is designed to support the typical DC Power supply as well as common extended functionality found in more complex instruments. The IviDCPwr class conceptualizes a DC power supply as an instrument capable of generating a DC power signal.

The IviDCPwrBase capability group provides the capability to configure a power supply for basic signal output. This includes setting the output range, the output voltage level, the over-voltage protection level, the current limit, enabling or disabling over-voltage protection, setting the current limit behavior, and enabling or disabling outputs. The IviDCPwrBase capability group is described in section 4, *IviDCPwrBase Capability Group*.

The IviDCPwrTrigger extension provides the capability to make changes to the output signal based on a trigger event. The driver can configure the trigger source, the triggered voltage level, and the triggered current limit. This extension group is described in section 5, *IviDCPwrTrigger Extension Group*.

The IviDCPwrSoftwareTrigger extension provides the capability to make changes to the output signal based on a software trigger event. Sending a software trigger to the instrument causes it to change the output signal. This extension group is described in section 6, *IviDCPwrSoftwareTrigger Extension Group*.

The IviDCPwrMeasurement extension provides the capability to take measurements on the output signal. With this extension, the driver can take immediate measurements of values like the DC Voltage and DC Current. This extension group is described in section 7, IviDCPwrMeasurement 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
- IVI-3.18: IVI.NET Utility Classes and Interfaces Specification
- VPP-3.x - VXI*plug&play* Instrument Driver Specifications

- VPP-4.x - Virtual Instrument Software Architecture (VISA) Specifications

1.4 Definitions of Terms and Acronyms

Refer to *IVI-5: Glossary* for a description of the terms and acronyms used in this specification. This specification does not define any additional terms.

2 IviDCPwr Class Capabilities

2.1 Introduction

The IviDCPwr specification divides DC power supply 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 an overview of the information presented for each capability group.

2.2 IviDCPwr Group Names

The capability group names for the IviDCPwr 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. IviDCPwr Group Names

Group Name	Description
IviDCPwrBase	Base capabilities of the IviDCPwr specification. This group supports the ability to generate a DC power signal, to specify output limits, and to control the behavior of the power supply when the output is greater than or equal to one of the limits.
IviDCPwrTrigger	This group supports the ability to make changes to the output signal based on a trigger event.
IviDCPwrSoftwareTrigger	This group supports the ability to make changes to the output signal based on a software trigger event.
IviDCPwrMeasurement	This group supports the ability to query the instrument for the measurement characteristics of the output signal.

Refer to Section 16, *Class Specification Layout*, in *IVI-3.4: API Style Guide* for a description of the Capability Group Section Layout.

2.3 Repeated Capability Names

The IviDCPwr 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*, Section 4.3.9, *Repeated Capabilities*, and Section 5.8, *Repeated Capability Identifiers and Selectors*.

- OutputChannel

2.3.1 OutputChannel

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

2.4   Boolean Attribute and Parameter Values

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

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

2.5   .NET Namespace

The .NET namespace for the IviDCPwr class is Ivi.DCPwr.

2.6   .NET IviDCPwr Session Factory

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

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

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

 Refer to Section 4.3.11, *Multithread Safety*, of *IVI-3.1: Driver Architecture Specification* for a complete description of IVI.NET driver locking. Refer to Section 8, Table 8.2 *Required Lock Type Behavior for Drivers With the Same Access Key*, of *IVI-3.2, Inherent Capability Specification*, for an explanation of how the values for `lockType` and `accessKey` are used to determine the kind of multithreaded lock to use for the driver instance.

.NET Method Prototype

```
IIviDCPwr Ivi.DCPwr.Create(String name);

IIviDCPwr Ivi.DCPwr.Create(String name,
                           Boolean idQuery,
                           Boolean reset);

IIviDCPwr Ivi.DCPwr.Create(String name,
                           Boolean idQuery,
                           Boolean reset,
                           String options);

IIviDCPwr Ivi.DCPwr.Create(String resourceName,
                           Boolean idQuery,
                           Boolean reset,
                           LockType lockType,
                           String accessKey,
                           String options);
```

Parameters

Inputs	Description	Base Type
name	A session name or a logical name that points to a session that uses an IVI.NET IviDCPwr class-compliant driver.	String
idQuery	Specifies whether to verify the ID of the instrument. The default is False.	Boolean
reset	Specifies whether to reset the instrument. The default is False.	Boolean
lockType	Specifies whether to use AppDomain-wide locking or machine-wide locking.	Ivi.Driver.LockType

accessKey	Specifies a user-selectable access key to identify the lock. Driver instances that are created with the same accessKey will be protected from simultaneous access by multiple threads within an AppDomain or across AppDomains, depending upon the value of the lockType parameter.	String
options	A string that allows the user to specify the initial values of certain inherent attributes. The default is an empty string.	String

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

Defined Values

Name	Description		
		Language	Identifier
AppDomain	The lock is AppDomain-wide.		
		.NET	Ivi.Driver.LockType.AppDomain
Machine	The lock is machine-wide.		
		.NET	Ivi.Driver.LockType.Machine

.NET Exceptions

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

Usage

To create a driver that implements the IviDCPwr instrument class API from the logical name "My LogicalName" use the following code:

IIviDCPwr dcPwr = IviDCPwr.Create("MyLogicalName");

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

3   General Requirements

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

3.1   Minimum Class Compliance

To be compliant with the IviDCPwr Class Specification, a specific driver shall implement the inherent capabilities that *IVI-3.2: Inherent IVI Capabilities Specification* defines, and the IviDCPwrBase 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 DC Power Supply 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 supply 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   IviDCPwrBase Capability Group

4.1   Overview

The IviDCPwrBase capability group supports the most basic DC power supply capabilities. The user can enable or disable outputs, specify the DC voltage to generate, specify output limits, and control the behavior of the power supply when the output is greater than or equal to one of the limits.

This specification uses the following terms to describe the power supply's output: Voltage Level, OVP Limit, Current Limit, Current Limit Behavior, Constant Voltage Mode, Constant Current Mode, and Unregulated Mode.

Voltage Level – The DC voltage the power supply attempts to generate. The user configures the voltage level with the Voltage Level attribute.

OVP – OVP is an acronym for Over-Voltage Protection.

OVP Limit and OVP Enabled – If the OVP limit is enabled, the power supply disables the output when the output voltage is greater than or equal to the OVP limit. The user configures the OVP limit with the OVP Limit attribute, and enables or disables the OVP limit with the OVP Enabled attribute.

Current Limit and Current Limit Behavior – The current limit behavior determines the behavior of the instrument when the output current is greater than or equal to the current limit. When the current limit behavior is *trip,* the power supply disables the output when the output current is greater than or equal to the current limit. When the current limit behavior is *regulate,* the power supply restricts the output voltage such that the output current is not greater than the current limit. The user configures the current limit and current limit behavior with the Current Limit and Current Limit Behavior attributes.

Constant Voltage Mode – The power supply is said to be in the constant voltage mode when the power supply's output signal reaches the voltage level before it reaches the current limit. In the constant voltage mode, the power supply's output voltage remains constant at the voltage level and its output current can vary.

Constant Current Mode – The power supply is said to be in the constant current mode when the power supply's output signal reaches the current limit before it reaches the voltage level, and the current limit behavior is set to regulate. In the constant current mode, the power supply's output current remains constant at the current limit and its output voltage varies.

Unregulated Mode – The power supply is said to be in the unregulated mode when the power supply's output signal reaches neither the voltage level or the current limit. In the unregulated mode, the power supply's output current and output voltage vary.

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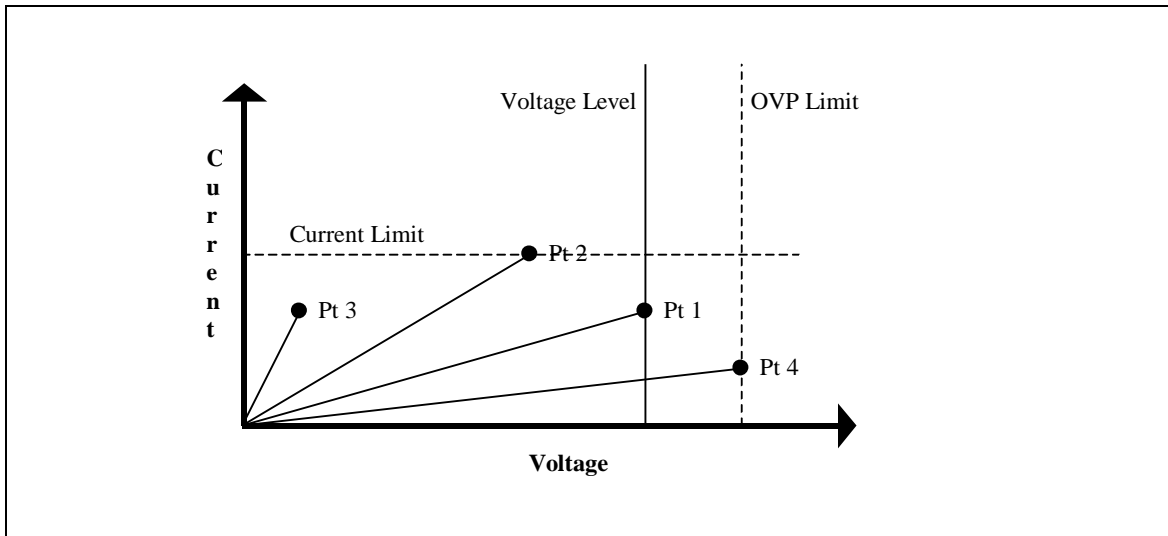
Figure 4-1. Power Supply Output Diagram

At Point 1, the power supply's output has reached the voltage level before it reached the current limit. This is an example of the power supply operating in the constant voltage mode. Note that any point on the vertical line defined by the voltage level would also cause the power supply to operate in the constant voltage mode.

At Point 2, the power supply's output has reached the current limit before it reached the voltage level. If the current limit behavior is set to regulate, this is an example of the power supply operating in the constant current mode. If the current limit behavior is set to trip, the power supply disables the output. Note that any point on the horizontal line defined by the current limit would also cause the power supply to operate in the constant current mode when the current limit behavior is set to regulate.

At point 3, the power supply's output has reached neither the voltage level or the current limit. This is an example of the power supply operating in the unregulated mode. Note that any point within the rectangle defined by the voltage level and current limit would also cause the power supply to operate in the unregulated mode.

At point 4, the power supply's output has reached the OVP limit. If OVP is enabled, the power supply disables the output. Note that any point on the vertical line defined by the OVP limit would also cause the power supply to disable the output when OVP is enabled.

4.2 IviDCPwrBase Attributes

The IviDCPwrBase capability group defines the following attributes:

- Current Limit

- Current Limit Behavior

- Output Enabled

- OVP Enabled

- OVP Limit

- Voltage Level

- Output Channel Count

- Output Channel Item (IVI-COM and IVI.NET only)

- Output Channel Name (IVI-COM and IVI.NET only)

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in section 8, *IVIDCPwr Attribute ID* Definitions.

4.2.1 Current Limit

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

.NET Property Name

```
Outputs[].CurrentLimit
```

COM Property Name

```
Outputs.Item().CurrentLimit
```

C Constant Name

```
IVIDCPWR_ATTR_CURRENT_LIMIT
```

Description

Specifies the output current limit. The units are Amps.

The value of the Current Limit Behavior attribute determines the behavior of the power supply when the output current is equal to or greater than the value of this attribute.

.NET Exceptions

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

4.2.2 Current Limit Behavior

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32	R/W	OutputChannel	None	Configure Current Limit

.NET Property Name

> Outputs[].CurrentLimitBehavior

.NET Enumeration Name

> Ivi.DCPwr.CurrentLimitBehavior

COM Property Name

> Outputs.Item().CurrentLimitBehavior

COM Enumeration Name

> IviDCPwrCurrentLimitEnum

C Constant Name

> IVIDCPWR_ATTR_CURRENT_LIMIT_BEHAVIOR

Description

Specifies the behavior of the power supply when the output current is equal to or greater than the value of the Current Limit attribute.

Defined Values

Name	Description	
	Language	*Identifier*
Current Trip	The power supply disables the output when the output current is equal to or greater than the value of the Current Limit attribute.	
	C	IVIDCPWR_VAL_CURRENT_TRIP
	COM	IviDcPwrCurrentLimitTrip
	.NET	Ivi.DCPwr.CurrentLimitBehavior.Trip
Current Regulate	The power supply restricts the output voltage such that the output current is not greater than the value of the Current Limit attribute.	
	C	IVIDCPWR_VAL_CURRENT_REGULATE
	COM	IviDcPwrCurrentLimitRegulate
	.NET	Ivi.DCPwr.CurrentLimitBehavior.Regulate

.NET Exceptions

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

Compliance Notes

1. If an IVI-C specific driver defines additional values for this attribute, the actual values must be greater than or equal to `IVIDCPWR_VAL_CURRENT_LIMIT_BEHAVIOR_SPECIFIC_EXT_BASE`.

2. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIDCPWR_VAL_CURRENT_LIMIT_BEHAVIOR_CLASS_EXT_BASE` and less than `IVIDCPWR_VAL_CURRENT_LIMIT_BEHAVIOR_SPECIFIC_EXT_BASE`.

3. When an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to Current Limit Behavior Specific Ext Base.

4.2.3 Output Enabled

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

.NET Property Name

> Outputs[].Enabled

COM Property Name

> Outputs.Item().Enabled

C Constant Name

> IVIDCPWR_ATTR_OUTPUT_ENABLED

Description

> If true, the signal the power supply produces appears at the output connector. If false, the signal the power supply produces does not appear at the output connector.

.NET Exceptions

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

Compliance Notes

> Instrument drivers shall support the value True on all outputs.

4.2.4 OVP Enabled

Data Type	Access	Applies to	Coercion	High Level Functions
ViBoolean	R/W	OutputChannel	None	Configure OVP

.NET Property Name

>   Outputs[].OvpEnabled

COM Property Name

>   Outputs.Item().OVPEnabled

C Constant Name

>   IVIDCPWR_ATTR_OVP_ENABLED

Description

> Specifies whether the power supply provides over-voltage protection. If this attribute is set to True, the power supply disables the output when the output voltage is greater than or equal to the value of the OVP Limit attribute.

.NET Exceptions

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

4.2.5 OVP Limit

Data Type	Access	Applies to	Coercion	High Level Functions
ViReal64	R/W	OutputChannel	None	Configure OVP

.NET Property Name

    Outputs[].OvpLimit

COM Property Name

    Outputs.Item().OVPLimit

C Constant Name

    IVIDCPWR_ATTR_OVP_LIMIT

Description

Specifies the voltage the power supply allows. The units are Volts.

If the OVP Enabled attribute is set to True, the power supply disables the output when the output voltage is greater than or equal to the value of this attribute.

If the OVP Enabled is set to False, this attribute does not affect the behavior of the instrument.

.NET Exceptions

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

4.2.6 Voltage Level

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

.NET Property Name

    Outputs[].VoltageLevel

COM Property Name

    Outputs.Item().VoltageLevel

C Constant Name

    IVIDCPWR_ATTR_VOLTAGE_LEVEL

Description

Specifies the voltage level the DC power supply attempts to generate. The units are Volts.

.NET Exceptions

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

4.2.7 Output Channel Count

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

.NET Property Name

    `Outputs.Count`

    This property is inherited from `IIviRepeatedCapabilityCollection`.

COM Property Name

    `Outputs.Count`

C Constant Name

    `IVIDCPWR_ATTR_CHANNEL_COUNT`

Description

    Returns the number of currently available outputs. The count returned includes any of the supported reserved repeated capability names defined in Section 2.3, *Repeated Capability Names* as well as any custom repeated capability identifiers. The C Constant Name uses CHANNEL while the COM Property Name is tied to the Outputs collection.

.NET Exceptions

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

4.2.8 Output Channel Item (IVI-COM and IVI.NET only)

Data Type	Access	Applies to	Coercion	High Level Functions
IIviDCPwrOutput*	RO	IviDCPwrOutput	None	N/A

.NET Property Name

```
Outputs[String name]
```

This indexer is inherited from `IIviRepeatedCapabilityCollection`. The name parameter uniquely identifies a particular output channel in the outputs collection.

COM Property Name

```
Outputs.Item([in] BSTR OutputName)
```

C Constant Name

```
N/A
```

Description

Output Channel Item uniquely identifies an output channel in the outputs collection. It returns an interface pointer which can be used to control the attributes and other functionality of that output channel.

The Item property takes an output name. If the user passes an invalid value for the output name parameter, the property returns an error.

Valid names include physical repeated capability identifiers and virtual repeated capability identifiers.

.NET Exceptions

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

4.2.9 Output Channel Name (IVI-COM and IVI.NET only)

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

.NET Property Name

Outputs[].Name

This property is inherited from IIviRepeatedCapabilityIdentification.

COM Property Name

Outputs.Name ([in] LONG Index)

C Constant Name

N/A

(Use the Get Output Channel Name function.)

Description

This attribute returns the repeated capability identifier defined by specific driver for the output channel that corresponds to the index that the user specifies. If the driver defines a qualified Output Channel name, this property returns the qualified name.

In COM, the index is one-based. In .NET, the index is zero-based.

For C and COM, valid values for the Index parameter are between one and the value of the Output Channel Count attribute, inclusive. If the user passes an invalid value for the Index parameter, the value of this attribute is an empty string, and the property returns an error.

If the value that the user passes for the Index parameter is less than one or greater than the value of the Output Channel Count, the attribute returns an empty string for the value and returns an error.

.NET Exceptions

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

4.3 IviDCPwrBase Functions

The IviDCPwrBase capability group defines the following functions:

- Configure Current Limit

- Configure Output Enabled (IVI-C only)

- Configure Output Range

- Configure OVP

- Configure Voltage Level (IVI-C only)

- Get Output Channel Name (IVI-C only)

- Query Current Limit Max

- Query Voltage Level Max

- Query Output State

- Reset Output Protection

This section describes the behavior and requirements of each function.

4.3.1 Configure Current Limit

Description

This function configures the current limit. It specifies the output current limit value and the behavior of the power supply when the output current is greater than or equal to that value.

See the definition of the Current Limit Behavior attribute for defined values for the behavior parameter.

.NET Method Prototype

```
void Outputs[].ConfigureCurrentLimit(CurrentLimitBehavior behavior,
                                     Double limit);
```

COM Method Prototype

```
HRESULT Outputs.Item().ConfigureCurrentLimit(
                        [in] IviDCPwrCurrentLimitBehaviorEnum Behavior,
                        [in] DOUBLE Limit);
```

C Prototype

```
ViStatus IviDCPwr_ConfigureCurrentLimit (ViSession Vi,
                                         ViConstString ChannelName,
                                         ViInt32 Behavior,
                                         ViReal64 Limit);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
ChannelName	The name of the output on which to configure the current limit.	ViConstString
Behavior	Specifies the behavior of the power supply when the output current is greater than or equal to the value of the Limit parameter. The driver uses this value to set the Current Limit Behavior attribute. See the attribute description for more details.	ViInt32
Limit	Specifies the power supply's output current limit. The driver uses this value to set the Current Limit attribute. See the attribute description for more details.	ViReal64

Return Values (C/COM)

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

.NET Exceptions

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

4.3.2 Configure Output Enabled (IVI-C only)

Description

Configures whether the signal that the power supply produces appears at the output connector.

.NET Method Prototype

```
N/A
```

(use the `Outputs[].Enabled` property)

COM Method Prototype

```
N/A
```

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

C Prototype

```
ViStatus IviDCPwr_ConfigureOutputEnabled (ViSession Vi,
                                          ViConstString ChannelName,
                                          ViBoolean Enabled);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
ChannelName	The name of the output to enable or disable.	ViConstString
Enabled	Specifies whether the signal the power supply 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 (C/COM)

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

4.3.3 Configure Output Range

Description

Configures the power supply's output range on an output. One parameter specifies whether to configure the voltage or current range, and the other parameter is the value to which to set the range.

Setting a voltage range can invalidate a previously configured current range. Setting a current range can invalidate a previously configured voltage range.

.NET Method Prototype

```
void Outputs[].ConfigureRange(Ivi.DCPwr.RangeType rangeType, Double range);
```

COM Method Prototype

```
HRESULT Outputs.Item().ConfigureRange([in] IviDCPwrRangeTypeEnum RangeType,
                                      [in] DOUBLE Range);
```

C Prototype

```
ViStatus IviDCPwr_ConfigureOutputRange (ViSession Vi,
                                        ViConstString ChannelName,
                                        ViInt32 RangeType,
                                        ViReal64 Range);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
ChannelName	The name of the output on which to configure the output range.	ViConstString
RangeType	The kind of range to be configured.	ViInt32
Range	The range in which the power supply operates. The instrument driver coerces this value to the closest value the instrument supports that is greater than or equal to the value specified.	ViReal64

Defined Values for RangeType Parameter

Name	Description		
		Language	*Identifier*
Voltage	Voltage range is set by the Range parameter.		
		.NET	RangeType.Voltage
		C	IVIDCPWR_VAL_RANGE_VOLTAGE
		COM	IviDCPwrRangeVoltage
Current	Current range is set by the Range parameter.		
		C	IVIDCPWR_VAL_RANGE_CURRENT
		.NET	RangeType.Current
		COM	IviDCPwrRangeCurrent

Return Values (C/COM)

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

.NET Exceptions

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

Compliance Notes

1. If an IVI-C class driver defines additional values for the `RangeType` parameter, the actual values shall be greater than or equal to `IVIDCPWR_VAL_RANGE_TYPE_CLASS_EXT_BASE` and less than `IVIDCPWR_VAL_RANGE_TYPE_SPECIFIC_EXT_BASE`.

2. If an IVI-C specific driver defines additional values for the `RangeType` parameter, the actual values shall be greater than or equal to `IVIDCPWR_VAL_RANGE_TYPE_SPECIFIC_EXT_BASE`.

4.3.4 Configure OVP

Description

Configures the over-voltage protection. It specifies the over-voltage limit and the behavior of the power supply when the output voltage is greater than or equal to that value.

When the `Enabled` parameter is False, the `Limit` parameter does not affect the instrument's behavior, and the driver does not set the OVP Limit attribute.

.NET Method Prototype

```
void Outputs[].ConfigureOvp(Boolean enabled, Double limit);
```

COM Method Prototype

```
HRESULT Outputs.Item().ConfigureOVP([in] VARIANT_BOOL Enabled,
                                    [in] DOUBLE Limit);
```

C Prototype

```
ViStatus IviDCPwr_ConfigureOVP (ViSession Vi,
                                ViConstString ChannelName,
                                ViBoolean Enabled,
                                ViReal64 Limit);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
ChannelName	The name of the output on which to configure over-voltage protection.	ViConstString
Enabled	Specifies the behavior of the power supply when the ouptut voltage is greater than or equal to the value of the `Limit` parameter. The driver uses this value to set the OVP Enabled attribute. See the attribute description for more details.	ViBoolean
Limit	Specifies the power supply's over-voltage protection limit. The driver uses this value to set the OVP Limit attribute. See the attribute description for more details.	ViReal64

Return Values (C/COM)

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

.NET Exceptions

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

4.3.5 Configure Voltage Level (IVI-C only)

Description

Configures the voltage level the power supply attempts to generate.

.NET Method Prototype

```
N/A
```

(use the `Outputs[].VoltageLevel` property)

COM Method Prototype

```
N/A
```

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

C Prototype

```
ViStatus IviDCPwr_ConfigureVoltageLevel (ViSession Vi,
                                         ViConstString ChannelName,
                                         ViReal64 Level);
```

Parameters

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

Return Values (C/COM)

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

4.3.6 Get Output Channel Name (IVI-C only)

Description

This function returns the specific driver defined output name that corresponds to the one-based index that the user specifies. If the driver defines a qualified Output Channel 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 Channel Count attribute, the function returns an empty string in the `Name` parameter and returns an error.

The C prototype uses Channel in the function name rather than the name of the repeated capability.

.NET Method Prototype

```
N/A
```

(use the `Outputs[].Name` property)

COM Method Prototype

```
N/A
```

(use the `Outputs.Name` property)

C Prototype

```
ViStatus IviDCPwr_GetChannelName (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	A user-allocated (for IVI-C) or driver-allocated (for IVI-COM) buffer into which the driver stores the output name. The caller may pass `VI_NULL` for this parameter if the `NameBufferSize` parameter is 0.	ViChar[]

Return Values (C/COM)

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

4.3.7 Query Current Limit Max

Description

This function returns the maximum programmable current limit that the power supply accepts for a particular voltage level on an output.

.NET Method Prototype

```
Double Outputs[].QueryCurrentLimitMax(Double voltageLevel);
```

COM Method Prototype

```
HRESULT Outputs.Item().QueryCurrentLimitMax(
                        [in] DOUBLE VoltageLevel,
                        [out, retval] DOUBLE *MaxCurrentLimit);
```

C Prototype

```
ViStatus IviDCPwr_QueryMaxCurrentLimit (ViSession  Vi,
                                        ViConstString ChannelName,
                                        ViReal64 VoltageLevel,
                                        ViReal64* MaxCurrentLimit);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
ChannelName	The name of the output on which to query the maximum current limit.	ViConstString
VoltageLevel	The voltage level for which to determine the maximum programmable current limit.	ViReal64

Outputs	Description	Base Type
Return value (.NET)	Returns the maximum programmable current limit for the specified voltage level.	ViReal64
MaxCurrentLimit (C/COM)	Returns the maximum programmable current limit for the specified voltage level.	ViReal64*

Return Values (C/COM)

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

.NET Exceptions

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

4.3.8 Query Voltage Level Max

Description

This function returns the maximum programmable voltage level that the power supply accepts for a particular current limit on an output.

.NET Method Prototype

```
Double Outputs[].QueryVoltageLevelMax(Double currentLimit);
```

COM Method Prototype

```
HRESULT Outputs.Item().QueryVoltageLevelMax(
                        [in] DOUBLE CurrentLimit,
                        [out, retval] DOUBLE *MaxVoltageLevel);
```

C Prototype

```
ViStatus IviDCPwr_QueryMaxVoltageLevel (ViSession Vi,
                                        ViConstString ChannelName,
                                        ViReal64 CurrentLimit,
                                        ViReal64* MaxVoltageLevel);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
ChannelName	The name of the output on which to query the maximum voltage level.	ViConstString
CurrentLimit	The current limit for which to determine the maximum programmable voltage level.	ViReal64

Outputs	Description	Base Type
Return value (.NET)	Returns the maximum programmable voltage level for the specified current limit.	ViReal64
MaxVoltageLevel (C/COM)	Returns the maximum programmable voltage level for the specified current limit.	ViReal64*

Return Values (C/COM)

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

.NET Exceptions

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

4.3.9 Query Output State

Description

This function returns whether the power supply is in a particular output state.

A constant voltage condition occurs when the output voltage is equal to the value of the Voltage Level attribute and the current is less than or equal to the value of the Current Limit attribute.

A constant current condition occurs when the output current is equal to the value of the Current Limit attribute and the Current Limit Behavior attribute is set to the Current Regulate defined value.

An unregulated condition occurs when the output voltage is less than the value of the Voltage Level attribute and the current is less than the value of the Current Limit attribute.

An over-voltage condition occurs when the output voltage is equal to or greater than the value of the OVP Limit attribute and the OVP Enabled attribute is set to True.

An over-current condition occurs when the output current is equal to or greater than the value of the Current Limit attribute and the Current Limit Behavior attribute is set to the Current Trip defined value.

When either an over-voltage condition or an over-current condition occurs, the power supply's output protection disables the output. If the power supply is in an over-voltage or over-current state, it does not produce power until the output protection is reset. The Reset Output Protection function resets the output protection. Once the output protection is reset, the power supply resumes generating a power signal.

.NET Method Prototype

```
Boolean Outputs[].QueryState(OutputState outputState);
```

COM Method Prototype

```
HRESULT Outputs.Item().QueryState([in] IviDCPwrOutputStateEnum OutputState,
                                  [out, retval] VARIANT_BOOL *InState);
```

C Prototype

```
ViStatus IviDCPwr_QueryOutputState (ViSession  Vi,
                                    ViConstString ChannelName,
                                    ViInt32 OutputState,
                                    ViBoolean* InState);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
ChannelName	The output on which to query the output state.	ViConstString
OutputState	The output state for which to query.	ViInt32

Outputs	Description	Base Type
InState (.NET)	Returns True if the power supply is currently in the state specified with the OutputState parameter, and False if it is not.	ViBoolean

InState (C/COM)	Returns True if the power supply is currently in the state specified with the `OutputState` parameter, and False if it is not.	`ViBoolean*`

Values for OutputState

Name	*Description*	
	Language	*Identifier*
Constant Voltage	Return value indicates whether a constant voltage condition exists.	
	C	`IVIDCPWR_VAL_OUTPUT_CONSTANT_VOLTAGE`
	.NET	`OutputState.ConstantVoltage`
	COM	`IviDCPwrOutputConstantVoltage`
Constant Current	Return value indicates whether a constant current condition exists.	
	C	`IVIDCPWR_VAL_OUTPUT_CONSTANT_CURRENT`
	.NET	`OutputState.ConstantCurrent`
	COM	`IviDCPwrOutputConstantCurrent`
Over Voltage	Return value indicates whether an over-voltage condition exists.	
	C	`IVIDCPWR_VAL_OUTPUT_OVER_VOLTAGE`
	.NET	`OutputState.OverVoltage`
	COM	`IviDCPwrOutputOverVoltage`
Over Current	Return value indicates whether an over-current condition exists.	
	C	`IVIDCPWR_VAL_OUTPUT_OVER_CURRENT`
	.NET	`OutputState.OverCurrent`
	COM	`IviDCPwrOutputOverCurrent`
Unregulated	Return value indicates whether an unregulated condition exists.	
	C	`IVIDCPWR_VAL_OUTPUT_UNREGULATED`
	.NET	`OutputState.Unregulated`
	COM	`IviDCPwrOutputUnregulated`

Return Values (C/COM)

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

.NET Exceptions

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

Compliance Notes

1. If an IVI-C class driver defines additional values for the `OutputState` parameter, the actual values shall be greater than or equal to `IVIDCPWR_VAL_OUTPUT_STATE_CLASS_EXT_BASE` and less than `IVIDCPWR_VAL_OUTPUT_STATE_SPECIFIC_EXT_BASE`.

2. If an IVI-C specific driver defines additional values for the `OutputState` parameter, the actual values shall be greater than or equal to `IVIDCPWR_VAL_OUTPUT_STATE_SPECIFIC_EXT_BASE`.

4.3.10 Reset Output Protection

Description

This function resets the power supply output protection after an over-voltage or over-current condition occurs.

An over-voltage condition occurs when the output voltage is equal to or greater than the value of the OVP Limit attribute and the OVP Enabled attribute is set to True.

An over-current condition occurs when the output current is equal to or greater than the value of the Current Limit attribute and the Current Limit Behavior attribute is set to Current Trip.

When either an over-voltage condition or an over-current condition occurs, the output protection of the power supply disables the output. Once the output protection is reset, the power supply resumes generating a power signal.

Use the Query Output State function to determine if the power supply is in an over-voltage or over-current state.

.NET Method Prototype

```
void Outputs[].ResetOutputProtection();
```

COM Method Prototype

```
HRESULT Outputs.Item().ResetOutputProtection();
```

C Prototype

```
ViStatus IviDCPwr_ResetOutputProtection (ViSession Vi,
                                          ViConstString ChannelName);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
ChannelName	The output on which to reset output protection	ViConstString

Return Values (C/COM)

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

.NET Exceptions

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

4.4   IviDCPwrBase Behavior Model

After the user calls the Initialize or Reset functions, the power supply produces a power signal based on its current configuration.

All changes to the power supply's IviDCPwrBase functions and attributes take place immediately.

5 IviDCPwrTrigger Extension Group

5.1 IviDCPwrTrigger Overview

The IviDCPwrTrigger extension group defines extensions for DC power supplies capable of changing the output signal based on a trigger event.

5.2 IviDCPwrTrigger Attributes

The IviDCPwrTrigger capability group defines the following attributes:

- Trigger Source

- Triggered Current Limit

- Triggered Voltage Level

This section describes the behavior and requirements of each attribute. The actual value for each attribute ID is defined in section 8, *IVIDCPwr Attribute ID Definitions*.

5.2.1 Trigger Source

Data Type	Access	Applies to	Coercion	High Level Functions
ViInt32 (C/COM)	R/W	OutputChannel	None	Configure Trigger Source
ViString (.NET)	R/W	OutputChannel	None	Configure Trigger Source

.NET Property Name

> `Outputs[].TriggerSource`

COM Property Name

> `Outputs.Item().TriggerSource`

COM Enumeration Name

> `IviDCPwrTriggerSourceEnum`

C Constant Name

> `IVIDCPWR_ATTR_TRIGGER_SOURCE`

Description

> Specifies the trigger source. After an Initiate call, the power supply waits for a trigger event from the source specified with this attribute. After a trigger event occurs, the power supply changes the voltage level to the value of the Triggered Voltage Level attribute and the current limit to the value of the Triggered Current Limit attribute.

Defined Values

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

Name	*Description*		
		Language	*Identifier*
Immediate	The power supply does not wait for a trigger before changing the output signal.		
		C	IVIDCPWR_VAL_TRIG_IMMEDIATE
		COM	IviDCPwrTriggerSourceImmediate
External	The power supply waits for an external trigger before changing the output signal.		
		C	IVIDCPWR_VAL_TRIG_EXTERNAL
		COM	IviDCPwrTriggerSourceExternal

Software Trigger	The power supply waits for the Send Software Trigger function to execute before changing the output signal.		
	C	IVIDCPWR_VAL_SOFTWARE_TRIG	
	COM	IviDCPwrTriggerSourceSwTrigFunc	
TTL0	The power supply waits for a trigger on the TTL 0 line before changing the output signal.		
	C	IVIDCPWR_VAL_TRIG_TTL0	
	COM	IviDCPwrTriggerSourceTTL0	
TTL1	The power supply waits for a trigger on the TTL 1 line before changing the output signal.		
	C	IVIDCPWR_VAL_TRIG_TTL1	
	COM	IviDCPwrTriggerSourceTTL1	
TTL2	The power supply waits for a trigger on the TTL 2 line before changing the output signal.		
	C	IVIDCPWR_VAL_TRIG_TTL2	
	COM	IviDCPwrTriggerSourceTTL2	
TTL3	The power supply waits for a trigger on the TTL 3 line before changing the output signal.		
	C	IVIDCPWR_VAL_TRIG_TTL3	
	COM	IviDCPwrTriggerSourceTTL3	
TTL4	The power supply waits for a trigger on the TTL 4 line before changing the output signal.		
	C	IVIDCPWR_VAL_TRIG_TTL4	
	COM	IviDCPwrTriggerSourceTTL4	
TTL5	The power supply waits for a trigger on the TTL 5 line before changing the output signal.		
	C	IVIDCPWR_VAL_TRIG_TTL5	
	COM	IviDCPwrTriggerSourceTTL5	
TTL6	The power supply waits for a trigger on the TTL 6 line before changing the output signal.		
	C	IVIDCPWR_VAL_TRIG_TTL6	
	COM	IviDCPwrTriggerSourceTTL6	
TTL7	The power supply waits for a trigger on the TTL 7 line before changing the output signal.		
	C	IVIDCPWR_VAL_TRIG_TTL7	
	COM	IviDCPwrTriggerSourceTTL7	
ECL0	The power supply waits for a trigger on the ECL 0 line before changing the output signal.		
	C	IVIDCPWR_VAL_TRIG_ECL0	
	COM	IviDCPwrTriggerSourceECL0	
ECL1	The power supply waits for a trigger on the ECL 1 line before changing the output signal.		
	C	IVIDCPWR_VAL_TRIG_ECL1	
	COM	IviDCPwrTriggerSourceECL1	
PXI Star	The power supply waits for a trigger on the PXI Star line before changing the output signal.		

	C	`IVIDCPWR_VAL_TRIG_PXI_STAR`
	COM	`IviDCPwrTriggerSourcePXIStar`
RTSI0	The power supply waits for a trigger on the RTSI 0 line before changing the output signal.	
	C	`IVIDCPWR_VAL_TRIG_RTSI_0`
	COM	`IviDCPwrTriggerSourceRTSI0`
RTSI1	The power supply waits for a trigger on the RTSI 1 line before changing the output signal.	
	C	`IVIDCPWR_VAL_TRIG_RTSI_1`
	COM	`IviDCPwrTriggerSourceRTSI1`
RTSI2	The power supply waits for a trigger on the RTSI 2 line before changing the output signal.	
	C	`IVIDCPWR_VAL_TRIG_RTSI_2`
	COM	`IviDCPwrTriggerSourceRTSI2`
RTSI3	The power supply waits for a trigger on the RTSI 3 line before changing the output signal.	
	C	`IVIDCPWR_VAL_TRIG_RTSI_3`
	COM	`IviDCPwrTriggerSourceRTSI3`
RTSI4	The power supply waits for a trigger on the RTSI 4 line before changing the output signal.	
	C	`IVIDCPWR_VAL_TRIG_RTSI_4`
	COM	`IviDCPwrTriggerSourceRTSI4`
RTSI5	The power supply waits for a trigger on the RTSI 5 line before changing the output signal.	
	C	`IVIDCPWR_VAL_TRIG_RTSI_5`
	COM	`IviDCPwrTriggerSourceRTSI5`
RTSI6	The power supply waits for a trigger on the RTSI 6 line before changing the output signal.	
	C	`IVIDCPWR_VAL_TRIG_RTSI_6`
	COM	`IviDCPwrTriggerSourceRTSI6`

.NET Exceptions

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

Compliance Notes

1. If an IVI-C class driver defines additional values for this attribute, the actual values shall be greater than or equal to `IVIDCPWR_VAL_TRIG_SRC_CLASS_EXT_BASE` and less than `IVIDCPWR_VAL_TRIG_SRC_SPECIFIC_EXT_BASE`.

2. If an IVI-C specific driver defines additional values for this attribute, the actual values must be greater than or equal to `IVIDCPWR_VAL_TRIG_SRC_SPECIFIC_EXT_BASE`.

3. If a specific driver implements any of the defined values in the following table, it must also implement the corresponding capability group:

Value	Required Capability Group

Software Trigger	IviDCPwrSoftwareTrigger

4. When an IVI-COM specific driver implements this attribute with additional elements in its instrument specific interfaces, the actual values of the additional elements shall be greater than or equal to Trigger Source Specific Ext Base.

5.2.2 Triggered Current Limit

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

.NET Property Name

Outputs[].TriggeredCurrentLimit

COM Property Name

Outputs.Item().TriggeredCurrentLimit

C Constant Name

IVIDCPWR_ATTR_TRIGGERED_CURRENT_LIMIT

Description

Specifies the value to which the power supply sets the current limit after a trigger event occurs. The units are Amps.

After an Initiate call, the power supply waits for a trigger event from the source specified with the Trigger Source attribute. After a trigger event occurs, the power supply sets the current limit to the value of this attribute.

After a trigger occurs, the value of the Current Limit attribute reflects the new value to which the current limit has been set.

.NET Exceptions

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

5.2.3 Triggered Voltage Level

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

.NET Property Name

    Outputs[].TriggeredVoltageLevel

COM Property Name

    Outputs.Item().TriggeredVoltageLevel

C Constant Name

    IVIDCPWR_ATTR_TRIGGERED_VOLTAGE_LEVEL

Description

Specifies the value to which the power supply sets the voltage level after a trigger event occurs. The units are Volts.

After an Initiate call, the power supply waits for a trigger event from the source specified with the Trigger Source attribute. After a trigger event occurs, the power supply sets the voltage level to the value of this attribute.

After a trigger occurs, the value of the Voltage Level attribute reflects the new value to which the voltage level has been set.

.NET Exceptions

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

5.3 IviDCPwrTrigger Functions

The IviDCPwrTrigger extension defines the following functions:

- Abort

- Configure Triggered Current Limit (IVI-C only)

- Configure Triggered Voltage Level (IVI-C only)

- Configure Trigger Source (IVI-C only)

- Initiate

This section describes the behavior and requirements of the function.

5.3.1 Abort

Description

If the power supply is currently waiting for a trigger to change the output signal, this function returns the power supply to the ignore triggers state.

If the power supply is not waiting for a trigger, this function does nothing and returns Success.

.NET Method Prototype

```
void Trigger.Abort();
```

COM Method Prototype

```
HRESULT Trigger.Abort();
```

C Prototype

```
ViStatus IviDCPwr_Abort (ViSession Vi);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession

Return Values (C/COM)

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

.NET Exceptions

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

5.3.2 Configure Triggered Current Limit (IVI-C only)

Description

Configures the current limit the power supply attempts to generate after it receives a trigger.

.NET Method Prototype

```
N/A
```
(use the `Outputs[].TriggeredCurrentLimit` property)

COM Method Prototype

```
N/A
```
(use the `Outputs.Item().TriggeredCurrentLimit` property)

C Prototype

```
ViStatus IviDCPwr_ConfigureTriggeredCurrentLimit (ViSession Vi,
                                    ViConstString ChannelName,
                                    ViReal64 Limit);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
ChannelName	The name of the output on which to configure the triggered current limit.	ViConstString
Limit	Specifies the current limit the power supply attempts to generate after it receives a trigger. The driver uses this value to set the Triggered Current Limit attribute. See the attribute description for more details.	ViReal64

Return Values (C/COM)

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

5.3.3 Configure Triggered Voltage Level (IVI-C only)

Description

Configures the voltage level the power supply attempts to generate after it receives a trigger.

.NET Method Prototype

```
N/A
```

(use the `Outputs[].TriggeredVoltageLevel` property)

COM Method Prototype

```
N/A
```

(use the `Outputs.Item().TriggeredVoltageLevel` property)

C Prototype

```
ViStatus IviDCPwr_ConfigureTriggeredVoltageLevel (ViSession Vi,
                                     ViConstString ChannelName,
                                     ViReal64 Level);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
ChannelName	The name of the output on which to configure the triggered voltage level.	ViConstString
Level	Specifies the voltage level the power supply attempts to generate after it receives a trigger. The driver uses this value to set the Triggered Voltage Level attribute. See the attribute description for more details.	ViReal64

Return Values (C/COM)

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

5.3.4 Configure Trigger Source (IVI-C only)

Description

Configures the trigger source the power supply waits for before changing the output signal.

.NET Method Prototype

```
N/A
```

(use the Trigger.Source property)

COM Method Prototype

```
N/A
```

(use the Trigger.Source property)

C Prototype

```
ViStatus IviDCPwr_ConfigureTriggerSource (ViSession Vi,
                                          ViConstString ChannelName,
                                          ViInt32 Source);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
ChannelName	The name of the output on which to configure the trigger source.	ViConstString
Source	Specifies the trigger source the power supply waits for before changing the output signal. The driver uses this value to set the Trigger Source attribute. See the attribute description for more details.	ViInt32

Return Values (C/COM)

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

5.3.5 Initiate

Description

If the power supply is not currently waiting for a trigger, this function causes the power supply to wait for a trigger.

If the power supply is already waiting for a trigger, this function does nothing and returns Success.

.NET Method Prototype

```
void Trigger.Initiate();
```

COM Method Prototype

```
HRESULT Trigger.Initiate();
```

C Prototype

```
ViStatus IviDCPwr_Initiate (ViSession Vi);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession

Return Values (C/COM)

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

.NET Exceptions

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

5.4 IviDCPwrTrigger Behavior Model

The following behavior model shows the relationship between the IviDCPwrTrigger capability group and power supply behavior.
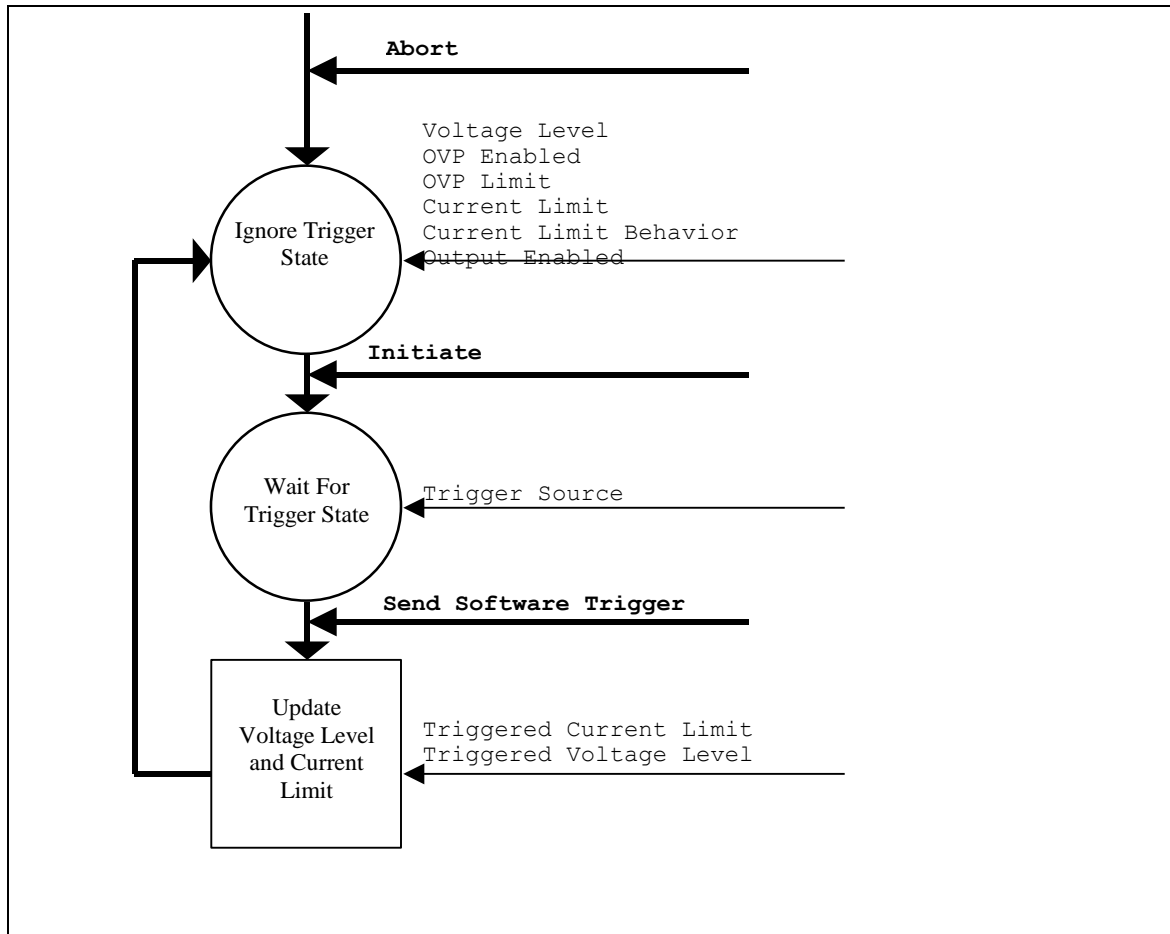
Figure 5-2. IviDCPwrTrigger Behavior Model

After the user calls the Initialize or Reset functions, the power supply enters the *ignore trigger* state.

In the *ignore trigger* state, the attributes of the IviDCPwrBase capability group determine the signal that the power supply produces. IviDCPwrTrigger attributes can be set, but do not affect the output signal.

Calling Initiate moves the power supply to the *wait for trigger* state.

In the *wait for trigger* state, the power supply waits for a trigger on the trigger source specified by the Trigger Source attribute. After the power supply receives a trigger, it sets the voltage level to the value of the Triggered Voltage Level attribute, and the current limit to the value of the Triggered Current Limit attribute. It then returns to the *ignore trigger* state.

After the changes in output occur, the Voltage Level and Current Limit attributes reflect the power supply's new configuration..

Calling the Abort function moves the power supply from its current state to the *ignore trigger* state. If the power supply has not yet responded to a trigger, no change occurs to the voltage level or current limit.

6 IviDCPwrSoftwareTrigger Extension Group

6.1 IviDCPwrSoftwareTrigger Overview

The IviDCPwrSoftwareTrigger extension group defines extensions for DC power supplies capable of changing the output signal based on a software trigger event.

6.2 IviDCPwrTrigger Functions

The IviDCPwrTrigger extension defines the following functions:

- Send Software Trigger

This section describes the behavior and requirements of this function.

6.2.1 Send Software Trigger

Refer to *IVI-3.3: Standard Cross Class Capabilities Specification, Section 2 Software Triggering Capability* for the prototype and complete description of this function.

6.3 IviDCPwrSoftwareTrigger Behavior Model

The IviDCPwrSoftwareTrigger Extension Group follows the behavior model of the IviDCPwrTrigger capability group. The only modification to the behavior model from the IviDCPwrTrigger capability group is the ability to send software triggers.

6.4 IviDCPwrSoftwareTrigger Compliance Notes

1. If an instrument driver implements the IviDCPwrSoftwareTrigger Extension Group, it shall implement the IviDCPwrTrigger Extension Group.

2. If an instrument driver implements the IviDCPwrSoftwareTrigger Extension Group, it shall implement the Software Trigger defined value for the Trigger Source attribute.

7   IviDCPwrMeasurement Extension Group

7.1   IviDCPwrMeasurement Overview

The IviDCPwrMeasurement extension group defines extensions for DC power supplies capable of calculating various measurements such as voltage and current from the output signal.

7.2   IviDCPwrMeasurement Functions

The IviDCPwrMeasurement extension defines the following function:

- Measure

This section describes the behavior and requirements of the function.

7.2.1 Measure

Description

Takes a measurement on the output signal and returns the measured value.

.NET Method Prototype

```
Double Outputs[].Measure(Ivi.DCPwr.MeasurementType measurementType);
```

COM Method Prototype

```
HRESULT Outputs.Item().Measure(
                        [in] IviDCPwrMeasurementTypeEnum MeasurementType,
                        [out, retval] DOUBLE *Measurement);
```

C Prototype

```
ViStatus IviDCPwr_Measure (ViSession  Vi,
                        ViConstString ChannelName,
                        ViInt32 MeasurementType,
                        ViReal64* Measurement);
```

Parameters

Inputs	Description	Base Type
Vi	Instrument handle	ViSession
ChannelName	The output on which to take a measurement	ViConstString
MeasurementType	The type of measurement to take.	ViInt32

Outputs	Description	Base Type
Return value (.NET)	The measured value.	ViReal64
Measurement (C/COM)	The measured value.	ViReal64*

Defined Values for MeasurementType Parameter

Name	Description	
	Language	*Identifier*
Voltage	Voltage is measured.	
	C	IVIDCPWR_VAL_MEASURE_VOLTAGE
	.NET	Ivi.DCPwr.MeasurementType.Voltage
	COM	IviDCPwrMeasurementVoltage
Current	Current is measured.	
	C	IVIDCPWR_VAL_MEASURE_CURRENT
	.NET	Ivi.DCPwr.MeasurementType.Current
	COM	IviDCPwrMeasurementCurrent

Return Values (C/COM)

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

.NET Exceptions

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

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 `IVIDCPWR_VAL_MEASURE_CLASS_EXT_BASE` and less than `IVIDCPWR_VAL_MEASURE_SPECIFIC_EXT_BASE`.

2. If an IVI-C specific driver defines additional values for the `rangeType` parameter, the actual values shall be greater than or equal to `IVIDCPWR_VAL_MEASURE_SPECIFIC_EXT_BASE`.

7.3 IviDCPwrMeasurement Behavior Model

The IviDCPwrMeasurement Extension Group follows the behavior model of the IviDCPwrBase capability group. The only modification to the behavior model from the IviDCPwrBase capability group is the ability to take measurement on the output signal.

8 IVIDCPwr Attribute ID Definitions

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

Table 8-1. IviDCPwr Attributes ID Values

Attribute Name	ID Definition
IVIDCPWR_ATTR_VOLTAGE_LEVEL	IVI_CLASS_ATTR_BASE + 1
IVIDCPWR_ATTR_OVP_ENABLED	IVI_CLASS_ATTR_BASE + 2
IVIDCPWR_ATTR_OVP_LIMIT	IVI_CLASS_ATTR_BASE + 3
IVIDCPWR_ATTR_CURRENT_LIMIT_BEHAVIOR	IVI_CLASS_ATTR_BASE + 4
IVIDCPWR_ATTR_CURRENT_LIMIT	IVI_CLASS_ATTR_BASE + 5
IVIDCPWR_ATTR_OUTPUT_ENABLED	IVI_CLASS_ATTR_BASE + 6
IVIDCPWR_ATTR_CHANNEL_COUNT	IVI_CLASS_ATTR_BASE + 203
IVIDCPWR_ATTR_TRIGGER_SOURCE	IVI_CLASS_ATTR_BASE + 101
IVIDCPWR_ATTR_TRIGGERED_CURRENT_LIMIT	IVI_CLASS_ATTR_BASE + 102
IVIDCPWR_ATTR_TRIGGERED_VOLTAGE_LEVEL	IVI_CLASS_ATTR_BASE + 103

In revision 1.0 of IVI 7: IviDCPwr Class Specification, the base attribute ID value was listed as: IVI_CLASS_PUBLIC_ATTR_BASE. The constant ID definition was renamed to IVI_CLASS_ATTR_BASE. The value of IVI_CLASS_ATTR_BASE is defined in IVI-3.1: Driver Architecture Specification.

9 IVIDCPwr Attribute Value Definitions

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

Current Limit Behavior

Value Name	Language	Identifier	Actual Value
Current Regulate	C	`IVIDCPWR_VAL_CURRENT_REGULATE`	0
	COM	`IviDCPwrCurrentLimitRegulate`	0
	.NET	`CurrentLimitBehavior.Regulate`	0
Current Trip	C	`IVIDCPWR_VAL_CURRENT_TRIP`	1
	COM	`IviDCPwrCurrentLimitTrip`	1
	.NET	`CurrentLimitBehavior.Trip`	1
Current Limit Behavior Class Ext Base	C	`IVIDCPWR_VAL_CURRENT_LIMIT_BEHAVIOR_CLASS_EXT_BASE`	500
	COM		
Current Limit Behavior Specific Ext Base	C	`IVIDCPWR_VAL_CURRENT_LIMIT_BEHAVIOR_SPECIFIC_EXT_BASE`	1000
	COM		1000

Trigger Source

Value Name	*Language*	*Identifier*	*Actual Value*
Immediate	C	IVIDCPWR_VAL_TRIG_IMMEDIATE	0
	COM	IviDCPwrTriggerSourceImmediate	0
External	C	IVIDCPWR_VAL_TRIG_EXTERNAL	1
	COM	IviDCPwrTriggerSourceExternal	1
Software Trigger	C	IVIDCPWR_VAL_SOFTWARE_TRIG	2
	COM	IviDCPwrTriggerSourceSwTrigFunc	2
TTL0	C	IVIDCPWR_VAL_TRIG_TTL0	3
	COM	IviDCPwrTriggerSourceTTL0	3
TTL1	C	IVIDCPWR_VAL_TRIG_TTL1	4
	COM	IviDCPwrTriggerSourceTTL1	4
TTL2	C	IVIDCPWR_VAL_TRIG_TTL2	5
	COM	IviDCPwrTriggerSourceTTL2	5
TTL3	C	IVIDCPWR_VAL_TRIG_TTL3	6
	COM	IviDCPwrTriggerSourceTTL3	6
TTL4	C	IVIDCPWR_VAL_TRIG_TTL4	7
	COM	IviDCPwrTriggerSourceTTL4	7
TTL5	C	IVIDCPWR_VAL_TRIG_TTL5	8
	COM	IviDCPwrTriggerSourceTTL5	8
TTL6	C	IVIDCPWR_VAL_TRIG_TTL6	9
	COM	IviDCPwrTriggerSourceTTL6	9
TTL7	C	IVIDCPWR_VAL_TRIG_TTL7	10
	COM	IviDCPwrTriggerSourceTTL7	10
ECL0	C	IVIDCPWR_VAL_TRIG_ECL0	11
	COM	IviDCPwrTriggerSourceECL0	11
ECL1	C	IVIDCPWR_VAL_TRIG_ECL1	12
	COM	IviDCPwrTriggerSourceECL1	12
PXI Star	C	IVIDCPWR_VAL_TRIG_PXI_STAR	13
	COM	IviDCPwrTriggerSourcePXIStar	13
RTSI0	C	IVIDCPWR_VAL_TRIG_RTSI_0	14
	COM	IviDCPwrTriggerSourceRTSI0	14
RTSI1	C	IVIDCPWR_VAL_TRIG_RTSI_1	15
	COM	IviDCPwrTriggerSourceRTSI1	15
RTSI2	C	IVIDCPWR_VAL_TRIG_RTSI_2	16
	COM	IviDCPwrTriggerSourceRTSI2	16
RTSI3	C	IVIDCPWR_VAL_TRIG_RTSI_3	17
	COM	IviDCPwrTriggerSourceRTSI3	17
RTSI4	C	IVIDCPWR_VAL_TRIG_RTSI_4	18
	COM	IviDCPwrTriggerSourceRTSI4	18
RTSI5	C	IVIDCPWR_VAL_TRIG_RTSI_5	19
	COM	IviDCPwrTriggerSourceRTSI5	19

RTSI6	C	`IVIDCPWR_VAL_TRIG_RTSI_6`	20
	COM	`IviDCPwrTriggerSourceRTSI6`	20
Trigger Source Class Ext Base	C	`IVIDCPWR_VAL_TRIG_SRC_CLASS_EXT_BASE`	500
	COM		
Trigger Source Specific Ext Base	C	`IVIDCPWR_VAL_TRIG_SRC_SPECIFIC_EXT_BASE`	1000
	COM		1000

10 IviDCPwr Function Parameter Value Definitions

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

Configure Output Range

Parameter: RangeType

Value Name	Language	Identifier	Actual Value
Current	C	IVIDCPWR_VAL_RANGE_CURRENT	0
	.NET	RangeType.Current	0
	COM	IviDCPwrRangeCurrent	0
Voltage	C	IVIDCPWR_VAL_RANGE_VOLTAGE	1
	.NET	RangeType.Voltage	1
	COM	IviDCPwrRangeVoltage	1
Configure Output Range Class Ext Base	C	IVIDCPWR_VAL_RANGE_TYPE_CLASS_EXT_BASE	500
	COM		
Configure Output Range Specific Ext Base	C	IVIDCPWR_VAL_RANGE_TYPE_SPECIFIC_EXT_BASE	1000
	COM		

Query Output State

Parameter: OutputState

COM Enumeration Name: IviDCPwrOutputStateEnum

Value Name	Language	Identifier	Actual Value
Constant Voltage	C	IVIDCPWR_VAL_OUTPUT_CONSTANT_VOLTAGE	0
	.NET	OuputState.ConstantVoltage	0
	COM	IviDCPwrOutputConstantVoltage	0
Constant Current	C	IVIDCPWR_VAL_OUTPUT_CONSTANT_CURRENT	1
	.NET	OuputState.ConstantCurrent	1
	COM	IviDCPwrOutputConstantCurrent	1
Over Voltage	C	IVIDCPWR_VAL_OUTPUT_OVER_VOLTAGE	2
	.NET	OuputState.OverVoltage	2
	COM	IviDCPwrOutputOverVoltage	2
Over Current	C	IVIDCPWR_VAL_OUTPUT_OVER_CURRENT	3
	.NET	OuputState.OverCurrent	3
	COM	IviDCPwrOutputOverCurrent	3
Unregulated	C	IVIDCPWR_VAL_OUTPUT_UNREGULATED	4
	.NET	OuputState.Unregulated	4
	COM	IviDCPwrOutputUnregulated	4
Query Output State Class Ext Base	C	IVIDCPWR_VAL_OUTPUT_STATE_CLASS_EXT_BASE	500
	COM		

Query Output State Specific Ext Base	C	IVIDCPWR_VAL_OUTPUT_STATE_SPECIFIC_EXT_BA SE	1000
	COM		

Measure

Parameter: MeasurementType

Value Name	*Language*	*Identifier*	*Actual Value*
Current	C	IVIDCPWR_VAL_MEASURE_CURRENT	0
	.NET	MeasurementType.Current	0
	COM	IviDCPwrMeasurementCurrent	0
Voltage	C	IVIDCPWR_VAL_MEASURE_VOLTAGE	1
	.NET	MeasurementType.Voltage	1
	COM	IviDCPwrMeasurementVoltage	1
Measure Class Ext Base	C	IVIDCPWR_VAL_MEASURE_CLASS_EXT_BASE	500
	COM		
Measure Specific Ext Base	C	IVIDCPWR_VAL_MEASURE_SPECIFIC_EXT_BAS E	1000
	COM		

11 Error and Completion Code Value Definitions

The table below specifies the actual value for each status code, and the actual exception class and warning GUID for each IVI.NET exception and warning that the IviDCPwr class specification defines. Note that warnings are raised as events in .NET. See IVI-3.1 and IVI-3.2 for details.

Table 11-1. IviDCPwr Error and Completion Codes

Error Name	Description		
	API	*Identifier*	*Value(hex)*
Trigger Not Software	The trigger source is not set to software trigger.		
	.NET	Ivi.Driver. TriggerNotSoftwareException	IVI defined exception (see IVI 3.2)
	C	IVIDCPWR_ERROR_TRIGGER_NOT_SOFTWARE	0xBFFA1001
	COM	E_IVIDCPWR_TRIGGER_NOT_SOFTWARE	0x80041001

12 IviDCPwr Hierarchies

12.1 IviDCPwr .NET Hierarchy

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

Table 12-1. IviDCPwr .NET Hierarchy

.NET Interface Hierarchy	Generic Name	Type
Outputs		
Count	Output Channel Count	P
Name	Output Channel Name	P
[]	**Item**	
ConfigureCurrentLimit	Configure Current Limit	M
ConfigureOvp	Configure OVP	M
ConfigureRange	Configure Output Range	M
CurrentLimit	Current Limit	P
CurrentLimitBehavior	Current Limit Behavior	P
Enabled	Output Enabled	P
Measure	Measure	M
OvpEnabled	OVP Enabled	P
OvpLimit	OVP Limit	P
QueryCurrentLimitMax	Query Current Limit Max	M
QueryState	Query Output State	M
QueryVoltageLevelMax	Query Voltage Level Max	M
ResetOutputProtection	Reset Output Protection	M
TriggeredCurrentLimit	Triggered Current Limit	P
TriggerSource	Trigger Source	P
TriggeredVoltageLevel	Triggered Voltage Level	P
VoltageLevel	Voltage Level	P
Trigger		
Abort	Abort	M
Initiate	Initiate	M
SendSoftwareTrigger	Send Software Trigger	M

12.1.1 IviDCPwr .NET Interface Reference Properties

Interface reference properties are used to navigate the IviDCPwr .NET hierarchy. This section describes the interface reference properties that the IIviDCPwr interface defines.

Data Type	.NET Property Name
IIviDCPwrOutputCollection	Outputs
IIviDCPwrOutput	Outputs[]
IIviDCPwrTrigger	Trigger

12.2 IviDCPwr COM Hierarchy

The full IviDCPwr COM Hierarchy includes the Inherent Capabilities Hierarchy as defined in Section 4.2, *COM Inherent Capabilities* of *IVI-3.2: Inherent Capabilities Specification*. To avoid redundancy, the Inherent Capabilities are omitted here.

Table 12-2. IviDCPwr COM Hierarchy

COM Interface Hierarchy	Generic Name	Type
Outputs		
Count	Output Channel Count	P
Name	Output Channel Name	P
Item		
Enabled	Output Enabled	P
VoltageLevel	Voltage Level	P
TriggerSource	Trigger Source	P
TriggeredVoltageLevel	Triggered Voltage Level	P
TriggeredCurrentLimit	Triggered Current Limit	P
ConfigureCurrentLimit	Configure Current Limit	M
CurrentLimit	Current Limit	P
CurrentLimitBehavior	Current Limit Behavior	P
ConfigureOVP	Configure OVP	M
OVPLimit	OVP Limit	P
OVPEnabled	OVP Enabled	P
ConfigureRange	Configure Output Range	M
Measure	Measure	M
ResetOutputProtection	Reset Output Protection	M
QueryCurrentLimitMax	Query Current Limit Max	M
QueryVoltageLevelMax	Query Voltage Level Max	M
QueryState	Query Output State	M
Trigger		
Abort	Abort	M
Initiate	Initiate	M
SendSoftwareTrigger	Send Software Trigger	M

12.2.1 IviDCPwr COM Interfaces

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

- IIviDCPwrOutputs
- IIviDCPwrTrigger

The IIviDCPwrOutputs interface contains methods and properties for accessing a collection of objects that implement the IIviDCPwrOutput interface.

Table 12-3**. IviDCPwr Interface GUIDs** lists the interfaces that this specification defines and their GUIDs.

Interface	GUID
IIviDCPwr	{ 47ed51b6-a398-11d4-ba58-000064657374 }
IIviDCPwrOutputs	{ 47ed51b7-a398-11d4-ba58-000064657374 }
IIviDCPwrOutput	{ 47ed51b8-a398-11d4-ba58-000064657374 }
IIviDCPwrTrigger	{ 47ed51b9-a398-11d4-ba58-000064657374 }

12.2.2 IviDCPwr COM Interface Reference Properties

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

Data Type	COM Property Name
IIviDCPwrOutputs	Outputs
IIviDCPwrTrigger	Trigger

12.2.3 IviDCPwr COM Category

The IviDCPwr class COM Category shall be "IviDCPwr", and the Category ID (CATID) shall be {47ed5153-a398-11d4-ba58-000064657374}.

12.3 IviDCPwr C Function Hierarchy

The IviDCPwr class function hierarchy is shown in the following table. The full IviDCPwr C Function Hierarchy includes the Inherent Capabilities Hierarchy as defined in Section 4.3, *C Inherent Capabilities* of *IVI-3.2: Inherent Capabilities Specification*. To avoid redundancy, the Inherent Capabilities are omitted here.

Table 12-4 IviDCPwr C Function Hierarchy

Name or Class	Function Name	
Configuration		
Configure Voltage Level	`IviDCPwr_ConfigureVoltageLevel`	
Configure OVP	`IviDCPwr_ConfigureOVP`	
Configure Current Limit	`IviDCPwr_ConfigureCurrentLimit`	
Configure Output Range	`IviDCPwr_ConfigureOutputRange`	
Configure Output Enabled	`IviDCPwr_ConfigureOutputEnabled`	
Query Current Limit Max	`IviDCPwr_QueryMaxCurrentLimit`	
Query Voltage Level Max	`IviDCPwr_QueryMaxVoltageLevel`	
Triggering		
Configure Trigger Source	`IviDCPwr_ConfigureTriggerSource`	
Configure Triggered Voltage Level	`IviDCPwr_ConfigureTriggeredVoltageLevel`	
Configure Triggered Current Limit	`IviDCPwr_ConfigureTriggeredCurrentLimit`	
Action		
Initiate	`IviDCPwr_Initiate`	
Abort	`IviDCPwr_Abort`	
Send SoftwareTrigger	`IviDCPwr_SendSoftwareTrigger`	
Query Output State	`IviDCPwr_QueryOutputState`	
Reset Output Protection	`IviDCPwr_ResetOutputProtection`	
Measure	`IviDCPwr_Measure`	
Get Output Channel Name	`IviDCPwr_GetChannelName`	

12.4 IviDCPwr C Attribute Hierarchy

The IviDCPwr class attribute hierarchy is shown in the following table. The full IviDCPwr C Attribute Hierarchy includes the Inherent Capabilities Hierarchy as defined in Section 4.3, *C Inherent Capabilities* of *IVI-3.2: Inherent Capabilities Specification*. To avoid redundancy, the Inherent Capabilities are omitted here.

Table 12-5. IviDCPwr C Attributes Hierarchy

Category or Generic Attribute Name	C Defined Constant
Output	
Current Limit	IVIDCPWR_ATTR_CURRENT_LIMIT
Current Limit Behavior	IVIDCPWR_ATTR_CURRENT_LIMIT_BEHAVIOR
Output Channel Count	IVIDCPWR_ATTR_CHANNEL_COUNT
Output Enabled	IVIDCPWR_ATTR_OUTPUT_ENABLED
OVP Enabled	IVIDCPWR_ATTR_OVP_ENABLED
OVP Limit	IVIDCPWR_ATTR_OVP_LIMIT
Voltage Level	IVIDCPWR_ATTR_VOLTAGE_LEVEL
Trigger	
Trigger Source	IVIDCPWR_ATTR_TRIGGER_SOURCE
Triggered Current Limit	IVIDCPWR_ATTR_TRIGGERED_CURRENT_LIMIT
Triggered Voltage Level	IVIDCPWR_ATTR_TRIGGERED_VOLTAGE_LEVEL

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 IviDCPwr 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 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 IviDCPwrTrigger Extension Group

The IviDCPwrTrigger extension group affects the instrument behavior only when the user calls the Initiate function. Therefore, this specification does not recommend attribute values that disable the IviDCPwrTrigger extension group.

Disabling the IviDCPwrSoftwareTrigger Extension Group

The IviDCPwrSoftwareTrigger extension group affects the instrument behavior only when the user calls the Send Software Trigger function. Therefore, this specification does not recommend attribute values that disable the IviDCPwrSoftwareTrigger extension group.

Disabling the IviDCPwrMeasurement Extension Group

The IviDCPwrMeasurement extension group affects the instrument behavior only when the user calls the Measure function. Therefore, this specification does not recommend attribute values that disable the IviDCPwrMeasurement extension group.

A.3 Special Considerations for Configure Output Range

Some DC power supplies do not allow the user to explicitly specify an output's range. Instead, they automatically change the range based on the values the user requests for the voltage level, OVP limit, and current limit. For instruments that automatically change the range, the IviDCPwr_ConfigureOutputRange function should perform range checking to verify that its input parameters are valid, but should not perform any communication with the instrument or set any attributes.

A.4 Special Considerations for Initiate

After an Initiate call, the values of the Voltage Level and Current Limit attributes might change. If a driver caches these attribute values, the driver needs to invalidate them when Initiate is called.

A.5 Special Considerations for Current Limit and Current Limit Behavior

Not all DC power supplies implement the current limit and current limit behavior settings in the same manner that IviDCPwr specification defines. Instead, some power supplies have a current level setting that is always active and an over-current limit setting that can be enabled or disabled. In order to develop a driver for these power supplies that complies with the IviDCPwr specification, the driver developer can implement the driver as described in this section.

It can be difficult to implement an IVI driver when the physical instrument settings differ from the attributes that the class specification defines. For these cases, a preferred approach can be to implement hidden attributes in the specific driver that control the physical instrument settings. The public attributes that the class specification defines manipulate the hidden attributes that correspond to the instrument settings. For this example the specific driver might implement the following hidden attributes

Instr Current Level – A hidden attribute that sets the instrument's current level.

Instr OCP Limit – A hidden attribute that sets the instrument's over-current protection limit.

Instr OCP Enable – A hidden attribute that enables and disables the instrument's over-current protection.

Using these hidden attributes, the driver developer implements the specific driver as follows:

- When the user sets the Current Limit attribute, the specific driver sets both the Instr Current Level attribute and the Instr OCP Limit to the value that the user requests.

- When the user sets the Current Limit Behavior attribute to Current Trip, the driver sets the Instr OCP Enable to True. When the user sets the Current Limit Behavior attribute to Current Regulate, the driver sets the Instr OCP Enable to False.

For one particular DC power supply, the over-current protection is disabled by setting the over-current limit to a value that is 120% of the maximum current that the power supply can produce. In this case the specific driver implementation changes as follows:

- The specific driver implements hidden attributes only for the current level and the over-current limit.

- When the user sets the Current Limit attribute the specific driver always sets the Instr Current Level attribute to the value that the user requests. If the Current Limit Behavior attribute is set to Trip, the specific driver also sets the Instr OCP Limit to the value that the user requests. Otherwise, the specific driver does not set the value of the Instr OCP Limit attribute.

- When the user sets the Current Limit Behavior attribute to Current Regulate, the driver sets the Instr OCP Limit to 120% of the maximum current that the power supply can produce. When the user sets the Current Limit Behavior attribute to Current Trip, the driver sets the Instr OCP Limit to the same value as the Current Level attribute.

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 were 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.
    - Measure
    - Initiate

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.

IviDCPwrBase Capability Group

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

IviDCPwrTrigger Capability Group

The driver needs to perform interchangeability checking on this group only when the function performing interchangeability checking is Initate.

IviDCPwrSoftwareTrigger Capability Group

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

IviDCPwrMeasurement Capability Group

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