**Initialize**

**C Function Prototype**

ViStatus rsscope\_init (ViRsrc resourceName, ViBoolean idQuery, ViBoolean resetDevice, ViSession\* instrumentHandle);

**Basic Function Prototype**

Function rsscope\_init (ByVal resourceName As ViRsrc, ByVal idQuery As ViBoolean, ByVal resetDevice As ViBoolean, instrumentHandle As ViSession) As ViStatus

**Purpose**

This function performs the following initialization actions:  
  
- Creates a new IVI instrument driver session.  
  
- Opens a session to the specified device using the interface and address you specify for the Resource Name parameter.  
  
- If the ID Query parameter is set to VI\_TRUE, this function queries the instrument ID and checks that it is valid for this instrument driver.  
  
- If the Reset parameter is set to VI\_TRUE, this function resets the instrument to a known state.  
  
- Sends initialization commands to set the instrument to the state necessary for the operation of the instrument driver.  
  
- Returns a ViSession handle that you use to identify the instrument in all subsequent instrument driver function calls.  
  
Note: This function creates a new session each time you invoke it. Although you can open more than one VISA session for the same resource, it is best not to do so. You can use the same session in multiple program threads.

**Parameters**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| **resourceName** | ViRsrc | Pass the resource name of the device to initialize.  You can also pass the name of a virtual instrument or logical name that you configure with the VISA Configuration utility. The virtual instrument identifies a specific device and specifies the initial settings for the session. A logical Name identifies a particular virtual instrument.  Refer to the following table below for the exact grammar to use for this parameter. Optional fields are shown in square brackets ([]).  Syntax ------------------------------------------------------ GPIB[board]::<primary address>[::secondary address]::INSTR VXI[board]::<logical address>::INSTR GPIB-VXI[board]::<logical address>::INSTR ASRL<port>::INSTR <LogicalName> <DriverSession>  If you do not specify a value for an optional field, the following values are used:  Optional Field - Value ------------------------------------------------------ board - 0 secondary address - none (31)  The following table contains example valid values for this parameter.  "Valid Value" - Description ------------------------------------------------------ "GPIB::22::INSTR" - GPIB board 0, primary address 22 no secondary address "GPIB::22::5::INSTR" - GPIB board 0, primary address 22 secondary address 5 "GPIB1::22::5::INSTR" - GPIB board 1, primary address 22 secondary address 5 "VXI::64::INSTR" - VXI board 0, logical address 64 "VXI1::64::INSTR" - VXI board 1, logical address 64 "GPIB-VXI::64::INSTR" - GPIB-VXI board 0, logical address 64 "GPIB-VXI1::64::INSTR" - GPIB-VXI board 1, logical address 64 "ASRL2::INSTR" - COM port 2 "SampleInstr" - Logical name "SampleInstr" "xyz432" - Driver Session "xyz432"   Default Value: "GPIB::20::INSTR" |
| **idQuery** | ViBoolean | Specify whether you want the instrument driver to perform an ID Query.  Valid Range: VI\_TRUE (1) - Perform ID Query (Default Value) VI\_FALSE (0) - Skip ID Query  When you set this parameter to VI\_TRUE, the driver verifies that the instrument you initialize is a type that this driver supports.   Circumstances can arise where it is undesirable to send an ID Query command string to the instrument. When you set this parameter to VI\_FALSE, the function initializes the instrument without performing an ID Query. |
| **resetDevice** | ViBoolean | Specify whether you want the to reset the instrument during the initialization procedure.  Valid Range: VI\_TRUE (1) - Reset Device (Default Value) VI\_FALSE (0) - Don't Reset |
| **instrumentHandle** | ViSession | Returns a ViSession handle that you use to identify the instrument in all subsequent instrument driver function calls.  Notes:  (1) This function creates a new session each time you invoke it. This is useful if you have multiple physical instances of the same type of instrument.   (2) Avoid creating multiple concurrent sessions to the same physical instrument. Although you can create more than one VISA session for the same resource, it is best not to do so. A better approach is to use the same VISA session in multiple execution threads. |
| **Return** | | |
| **status** | ViStatus | Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.  To obtain a text description of the status code, call the rsscope\_error\_message function. To obtain additional information about the error condition, call the rsscope\_GetError function. To clear the error information from the driver, call the rsscope\_ClearError function.  The general meaning of the status code is as follows:  Value Meaning ------------------------------- 0 Success Positive Values Warnings Negative Values Errors  This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:  Numeric Range (in Hex) Status Code Types ------------------------------------------------- 3FFF0000 to 3FFFFFFF VISA Warnings 3FFC0000 to 3FFCFFFF VXIPnP Driver Warnings  BFFF0000 to BFFFFFFF VISA Errors BFFC0000 to BFFCFFFF VXIPnP Driver Errors |