IVI-3.6: COM Session Factory Specification

May 19, 2008 Edition
Revision 1.0

Important Information

The IVI-3.6: COM Session Factory Specification is authored by the IVI Foundation member companies. For a vendor membership roster list, please visit the IVI Foundation web site at `www.ivifoundation.org`.

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

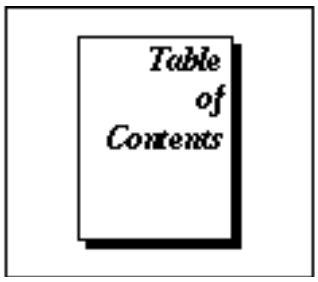
Warranty

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

Trademarks

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

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



COM Session Factory Specification

COM Session Factory Revision History

This section is an overview of the revision history of the IVI-COM Session Factory specification.

Table 1. COM Session Factory Specification Revisions

Revision Number	Date of Revision	Revision Notes
Revision 0.1	October, 1999	Original draft.
Revision 0.2	May 16, 2001	COM Driver Factory specification split from combined Config Store/Factory specification.
Revision 1.0 vc1	October 17, 2001	Final Draft: Voting Candidate 1
Revision 1.0 vc2	December 5, 2001	Final Draft: Voting Candidate 2
Revision 1.0 vc3	March 8, 2002	Final Draft: Voting Candidate 3
Revision 1.0	May 3, 2002	Voted and approved version 1.0
Revision 1.0	May 19 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.

1. Overview of the IVI-COM Session Factory Specification

1.1 Introduction

This document provides detailed documentation on the operation of the COM Session Factory. The purpose of the COM Session Factory is to provide to the client application a simple mechanism for creating instances of IVI-COM driver and rôle components. Additionally, the activation mechanism of the COM Session Factory provides a means of interchanging IVI-COM drivers without modifying the client program source code.

The present design of the COM Session Factory relies on the IVI Configuration Store to associate logical names of instruments with the specific drivers intended for use. The COM Session Factory relies on these mappings to resolve logical names provided by a client programmer to an assortment of initial settings provided in the configuration store. Once the COM Session Factory has instantiated a driver or rôle component according to the data in the configuration store, then the client program communicates directly with the component, with no intermediate run-time layer required to invoke methods.

1.2 COM Session Factory Overview

The following Visual Basic code snippet demonstrates the intended usage of the COM Session Factory:

```
Dim Factory As New IviSessionFactory
Dim Scope As IIviScope
Set Scope = Factory.CreateDriver("MyScope")
Scope.Initialize "MyScope", ...
Scope.Acquisition.NumberOfPointsMin = 1000
```

All of the work of the COM Session Factory is performed in the CreateDriver method shown in the Set statement above. The only parameter is the logical name `MyScope`, which is a user-specified name representing the instrument the client intends to use. The COM Session Factory accesses the current configuration store to see which IVI-COM driver is currently mapped to the logical name `MyScope`. If `MyScope` is found in the configuration store, then all of the required IVI-COM initial settings can be retrieved and used to instantiate the desired IVI-COM driver. As a result of this activity, the client receives an interface pointer, which may subsequently be used in the client program to execute methods supported by the IVI-COM driver. It is important to note that no changes are required in the client program when the user decides to use a different specific instrument model. Only the mappings in the configuration store must be changed, and the COM Session Factory will automatically instantiate the correct driver.

The above exposition holds true for use of the CreateSession method to instantiate rôle components.

1.3 References

Several other documents and specifications are related to this specification. These other related documents are as follows:

- IVI Charter Document
- IVI-3.1: Driver Architecture Specification
- IVI-3.2: Inherent Capabilities Specification
- IVI-3.5: Configuration Server Specification
- IVI-3.10: Measurement and Stimulus Subsystems Specification
- IVI-5: Glossary

1.4 Implementation

The current installation package for the IVI Foundation Shared Components, including the COM Session Factory, is available from the IVI Foundation website at http://www.ivifoundation.org.

2. COM Session Factory Interface Specification

2.1 CreateSession

Description

This function creates an instance of an IVI-COM rôle and returns an IUnknown pointer to the caller. The creation process uses the Configuration Server to map the provided LogicalName to the ProgID of the corresponding rôle. The ProgID used to activate the rôle is retrieved from the appropriate IviSession entry in the configuration store.

COM Method Prototype

```
HRESULT CreateSession([in] BSTR LogicalName, [out, retval] IUnknown** Session)
```

Parameters

Inputs	Description	Base Type
`LogicalName`	Logical name of the IVI-COM rôle to create.	`BSTR`

Outputs	Description	Base Type
`Session`	Interface pointer to the IVI-COM rôle represented by `LogicalName`.	`IUnknown **`

Return Values

Any error that can be returned from ::CoCreateInstance().

Any error that can be returned from IIviConfigStore::MasterLocation(), IIviConfigStore::Deserialize(), IIviConfigStore::GetSession(), or IIviSession::SoftwareModule().

HRESULT	Explanation
`S_OK`	Success
`E_IVIFACTORY_NO_CONFIG_STORE`	The COM Session Factory was unable to create an instance of the Configuration Server.
`E_IVIFACTORY_NO_SESSION`	The COM Session Factory did not find a Session corresponding to the given `LogicalName`.
`E_IVIFACTORY_NO_SW_MODULE`	The COM Session Factory did not find a SW Module corresponding to the given `LogicalName`.
`E_IVIFACTORY_NO_PROG_ID`	The COM Session Factory did not find a ProgID corresponding to the given `LogicalName`.
`E_IVIFACTORY_CLSID_NOT_REGISTERED`	The COM Session Factory was unable to translate the ProgID found in the configuration store to a registered CLSID.

2.2 CreateDriver

Description

This function creates an instance of an IVI-COM driver and returns an IUnknown pointer to the caller. The creation process uses the Configuration Server to map the provided LogicalName to the ProgID of the corresponding driver. The ProgID used to activate the driver is retrieved from the appropriate IviDriverSession entry in the configuration store.

COM Method Prototype

```
HRESULT CreateDriver([in] BSTR LogicalName, [out, retval] IUnknown** Driver)
```

Parameters

Inputs	Description	Base Type
LogicalName	Logical name of the IVI-COM driver to create.	BSTR

Outputs	Description	Base Type
Driver	Interface pointer to the IVI-COM driver represented by LogicalName.	IUnknown **

Return Values

Any error that can be returned from ::CoCreateInstance().

Any error that can be returned from IIviConfigStore::MasterLocation(), IIviConfigStore::Deserialize(), IIviConfigStore::GetDriverSession(), or IIviSession::SoftwareModule().

HRESULT	Explanation
S_OK	Success
E_IVIFACTORY_NO_CONFIG_STORE	The COM Session Factory was unable to create an instance of the Configuration Server.
E_IVIFACTORY_NO_SESSION	The COM Session Factory did not find a Driver Session corresponding to the given LogicalName.
E_IVIFACTORY_NO_SW_MODULE	The COM Session Factory did not find a SW Module corresponding to the given LogicalName.
E_IVIFACTORY_NO_PROG_ID	The COM Session Factory did not find a ProgID corresponding to the given LogicalName.
E_IVIFACTORY_CLSID_NOT_REGISTERED	The COM Session Factory was unable to translate the ProgID found in the configuration store to a registered CLSID.

3. Error and Completion Code Value Definitions

3.1 IVI-COM Error Codes

The table below specifies the actual value for each status code that the COM Session Factory specification defines.

Table 3-1. COM Session Factory Error and Completion Codes

Error Name	*Description*	
	Identifier	*Value(hex)*
No Config Store	The COM Session Factory was unable to create an instance of the Configuration Server.	
	E_IVIFACTORY_NO_CONFIG_STORE	0x80041100
No Driver Session/ No Session	The COM Session Factory did not find a Session or Driver Session corresponding to the given LogicalName.	
	E_IVIFACTORY_NO_SESSION	0x80041101
No Software Module	The COM Session Factory did not find a SW Module corresponding to the given LogicalName.	
	E_IVIFACTORY_NO_SW_MODULE	0x80041102
No Prog ID	The COM Session Factory did not find a ProgID corresponding to the given LogicalName.	
	E_IVIFACTORY_NO_PROG_ID	0x80041103
Driver Session Not Registered/ Session Not Registered	The COM Session Factory was unable to translate the ProgID found in the configuration store to a registered CLSID.	
	E_IVIFACTORY_CLSID_NOT_REGISTERED	0x80041104

Table 3-2 defines the recommended format of the message string associated with the errors. In COM, these strings are the description contained in the ErrorInfo object.

Note: In the description string table entries listed below, **%s** is always used to represent the component name. Where used, **%s1** represents context sensitive information.

Table 3-2. COM Session Factory Error Message Strings

Name	Message String
No Config Store	"%s: Problem initiating Configuration Server"
No Driver Session	"%s: Logical Name (%s1) references a non-existent Driver Session"
No Session	"%s: Logical Name (%s1) references a non-existent Session"
No Software Module	"%s: Driver Session (%s1) references a non-existent Software Module"
No Prog ID	"%s: Software Module (%s1) has no Prog ID"
Driver Session Not Registered	"%s: Driver Session CLSID (%s1) not registered"
Session Not Registered	"%s: Session CLSID (%s1) not registered"

Appendix A: COM Factory IDL file

```
// IviSessionFactory.idl : IDL source for IviSessionFactory.dll
// Copyright: (c) 2001-2002, Racal Instruments, All rights reserved
// Copyright: (c) 2001-2002, IVI Foundation, Inc, All rights reserved
// Original Author: David G McKay
//
// This file will be processed by the MIDL tool to
// produce the type library (IviSessionFactory.tlb) and marshalling code.

import "oaidl.idl";
import "ocidl.idl";

	[
		object,
		uuid(DE217CF0-2F0C-4EB5-B435-E69400C467EC),

		helpstring("IIviSessionFactory Interface"),
		pointer_default(unique)
	]
	interface IIviSessionFactory : IUnknown
	{
		[helpstring("method CreateDriver")] HRESULT CreateDriver([in] BSTR
LogicalName, [out, retval] IUnknown** Driver);
		[helpstring("method CreateSession")] HRESULT CreateSession([in] BSTR
LogicalName, [out,retval] IUnknown** Session);
	};
[
	uuid(2730EDC0-8DB0-445B-9BD2-B031416475C3),
	version(1.0),
	helpstring("IviSessionFactory 1.0 Type Library")
]
library IVISESSIONFACTORYLib
{
	importlib("stdole32.tlb");
	importlib("stdole2.tlb");

	[
		uuid(1EF38758-06AA-4CAD-9982-2CEA978C5855),
		helpstring("IviSessionFactory Class")
	]
	coclass IviSessionFactory
	{
		[default] interface IIviSessionFactory;
	};
};
```