CISCO SYSTEMS

Cisco Element Management Framework CORBA Gateway v1.0 to v2.1 Porting Guide

Corporate Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
      800 553-NETS (6387)
Fax: 408 526-4100

Text Part Number: OL-4362-01

Cisco Element Management Framework CORBA Gateway v1.0 to v2.1 Porting Guide

C O N T E N T S

Cisco Element Management Framework CORBA Gateway v1.0 to v2.1 Porting Guide

Porting Cisco EMF CORBA Gateway v1.0 to v2.1

This document provides information for the CORBA Gateway Developer who requires to port existing Version 1.0 code to Version 2.1. Details of system and software requirements can be found in the Cisco EMF CORBA Gateway Developer Guide v2.1, caveats are detailed in the Release Note.

Version 2.1 of the Cisco EMF CORBA Gateway gives the following improvements:

- It extends the suite of CORBA services available to CORBA developers so that virtually all published APIs are available.

- The Interface Definition Language (IDL) conforms more strictly to OMG standard, so that most IDL compiler can compile it. Version 2.1 removes the use of restricted names, which can produce errors on various compilers.

- It extends the IDL where synchronous calls are required, to allow developers to implement multiple-threaded clients.

- It allows developers to implement clients to CORBA's Notification Service so that they can receive Cisco EMF events.

- It allows CORBA integration with Cisco EMF 3.2 Patch 5.4 and beyond.

- It allows developers to implement clients to CORBA Object Group Service so they can manipulate Cisco EMF Object Groups.

New IDLs

The following additional files are included in Version 2.1 to support Object Group and Event Channel services in the Developer's Toolkit:

- CosEventChannelAdmin.idl
- CosEventComm.idl
- CosNotification.idl
- CosNotifyChannelAdmin.idl
- CosNotifyComm.idl
- CosNotifyFilter.idl
- CosTrading.idl
- EventChannelManager.idl
- ObjectGroups.idl

Cisco Element Management Framework CORBA Gateway v1.0 to v2.1 Porting Guide

- Query.idl

- Session.idl

- StructuredEvents.idl

✎

Note   The IDL is included in the Cisco EMF CORBA Gateway Developer Toolkit. For installation, refer to the *Cisco Element Management Framework CORBA Gateway v2.1 Installation Instructions*.

IDL Modifications

Changes to the IDL attributes, struct names and interfaces fulfill the aim of making the IDL compilable with most IDL compilers, by making it adhere more strictly to the OMG standard on symbol usage.

1. In the module **OBJ_ATL** the structure **Attribute** is now called **Attr**, the reason for this is Attribute is a restricted name, which may or may not compile with some compilers.

2. Table 1-1 to Table 1-4 list the symbols that have been changed so that they are unique whatever their context.

Table 1-1   IDL Filename `ActionLauncher.idl`

Old Symbol	New Symbol
`ATL_ACTION::Action`	`ATL_ACTION::ActionStruct`
`ATL_ACTION::Annotations`	`ATL_ACTION::AnnotationsList`

Table 1-2   IDL Filename `Attribute.idl`

Old Symbol	New Symbol
`ATL_OBJ::ObjectIDNamePair::objectID`	`ATL_OBJ::ObjectIDNamePair::objID`
`ATL_OBJ::GenericIDSetValue`	`ATL_OBJ::GenericIDSet`
`ATL_OBJ::GenericIDVectorValue`	`ATL_OBJ::GenericIDVector`
`ATL_OBJ::ObjectStateValue`	`ATL_OBJ::ObjectState`
`ATL_OBJ::ObjectStateUpdateValue`	`ATL_OBJ::ObjectStateUpdate`
`ATL_OBJ::ContainmentPathValue`	`ATL_OBJ::ContainmentPath`
`ATL_OBJ::ContainmentPathListValue`	`ATL_OBJ::ContainmmentPathList`
`ATL_OBJ::ContainmentScopeValue`	`ATL_OBJ::ContainmentScope`
`ATL_OBJ::IdNamePairSetValue`	`ATL_OBJ::IdNamePairSet`
`ATL_OBJ::Int64Value`	`ATL_OBJ::Int64`
`ATL_OBJ::UInt64Value`	`ATL_OBJ::UInt64`
`ATL_OBJ::AlarmQuerySpecifier::timeFilter*`	`ATL_OBJ::AlarmQuerySpecifier::timeStampFilter`
`ATL_OBJ::AttributeValueUnion::objectStateQuerySpecifier*`	`ATL_OBJ::AttributeValueUnion::objectStateQuerySpecifierValue`
`ATL_OBJ::SNMPSpecific::snmpSpecific*`	`ATL_OBJ::SNMPSpecific::snmpStruct`
`ATL_OBJ::ChoiceValue::attr*`	`ATL_OBJ::ChoiceValue::choiceAttr`

Additions to the Attribute.idl file for setting the vector value as follows:

```
struct AVVSimple
    {
        ObjectID moid;
        ObjectID attrid;
        VectorValue value;
    };

    typedef sequence<AVVSimple> AVVSimpleList;

    struct AVVSimpleFailure
    {
        ObjectID moid;
        ObjectID attrid;
    };

    typedef sequence<AVVSimpleFailure> AVVSimpleFailureList;
```

Additions to the DataAbstractor.idl are as follows:

```
module ATL_DABS
{
    interface DataAbstractorCallbackExtended;

    interface DataAbstractorCallbackExtended :  DataAbstractorCallback {

    oneway void setAVVSimpleCB( in ATL_OBJ::AVVSimpleList success,
                                in ATL_OBJ::AVVSimpleFailureList failures,
                                in unsigned long userData );

    };

};
```

Table 1-3   IDL Filename `Query.idl`

Old Symbol	New Symbol
`ATL_Query::QueryScope_d::containment*`	`ATL_Query::QueryScope_d::contScope`
`ATL_Query::QueryScope_d::object*`	`ATL_Query::QueryScope_d::objScope`

Table 1-4   IDL Filename `StructuresEvents.idl`

Old Symbol	New Symbol
`ATL_STRUCTURED_EVENT::MetadataDistributionEvent::metadataList*`	`ATL_STRUCTURED_EVENT::MetadataDistributionEvent::metaList`

Note   Symbols marked with * are not available in Version 1.0, but represents a change from the Early Field Trial Version

3. In the Participation.idl file, parameter names in interface methods which have the same name as their structure have been changed as follows:

Note   Version 2.1 examples are shown in bold text.

```
ATL_PART::ParticipationCoordinator::registerParticipant(
    in RegistrationSpec regSpec,
    in ContextFilter contextFilter,
    in Participant participant,
    in unsigned long userData)
```

```
ATL_PART::ParticipationCoordinator::registerParticipant(
    in RegistrationSpec regSpec,
    in ContextFilter filter,
    in Participant part,
    in unsigned long userData)
```

```
ATL_PART::ParticipationCoordinator::deregisterParticipant(
    in RegistrationSpec regSpec,
    in ContextFilter contextFilter,
    in Participant participant,
    in unsigned long userData)
```

```
ATL_PART::ParticipationCoordinator::deregisterParticipant(
    in RegistrationSpec regSpec,
    in ContextFilter filter,
    in Participant part,
    in unsigned long userData)
```

```
ATL_PART::ParticipationCoordinator::startContext(
    in CorbaParticipationContext cpc,
    in Boolean progressReportFlag,
    in long startlevel,
    in long stoplevel,
    in Initiator initiator,
    in unsigned long userData)
```

```
ATL_PART::ParticipationCoordinator::startContext(
    in CorbaParticipationContext cpc,
    in Boolean progressReportFlag,
    in long startlevel,
    in long stoplevel,
    in Initiator init,
    in unsigned long userData)
```

```
ATL_PART::ParticipationCoordinator::stopContext(
    in long contextID,
    in Initiator initiator,
    in unsigned long userData)
```

```
ATL_PART::ParticipationCoordinator::stopContext(
    in long contextID,
    in Initiator init,
    in unsigned long userData)
```

4. CorbaGateway Manager API now includes exceptions for authorization failures, this is for use when a client provides incorrect username and password combinations.

5. In "The DeploymentContext", the text and code has been changed to reflect that deployment context is now available via the CORBA Gateway rather than the ParticipationCoordinator.

6. Participation Levels are now in the Deployment context.

7. The DataAbstractor IDL now has two additional methods which provide synchronous get and set calls:

```
short get(inout ATL_OBJ::ObjectAttributeList requests);
short set(inout ATL_OBJ::ObjectAttributeList requests);
```

8. In ActionLauncher IDL the method `actionResult` is changed to `actionInvocationResult`

9. Systems attributes are now defined in an IDL.

Retrieving CORBA Gateway Reference

The use of the Naming Service by the CORBA Gateway has changed in version 2.1. The kind field of each CosNaming object is an empty string for each placeholder object, and the reference holder object for the CORBA Gateway Manager has a kind field value of ServerObject. To illustrate this change the two code examples below show how a reference to the CORBA Gateway Manager is retrieved in CORBA Gateway version 1.0 and 2.1.

Note The code changed is in bold.

Firstly in version 1.0:

```
CosNaming::Name_var name = new CosNaming::Name(2);
name->length(2);
name[0].id = CORBA::string_dup("atl");
name[0].kind = CORBA::string_dup("atl");
name[1].id = CORBA::string_dup("CorbaGatewayManager");
name[1].kind = CORBA::string_dup("CorbaGatewayManager");

// Resolve the CorbaGatewayManager
CORBA::Object_var tempObj = nsrc->resolve(name);
cgm = ATL_GATE::CorbaGatewayManager::_narrow(tempObj);
```

In version 2.1 the following code is used to retrieve a reference to the CORBA Gateway Manager:

```
CosNaming::Name_var name = new CosNaming::Name(2);
name->length(2);
name[0].id = CORBA::string_dup("atl");
name[0].kind = CORBA::string_dup("");
name[1].id = CORBA::string_dup("CorbaGatewayManager");
name[1].kind = CORBA::string_dup("ServerObject");

// Resolve the CorbaGatewayManager
CORBA::Object_var tempObj = nsrc->resolve(name);
cgm = ATL_GATE::CorbaGatewayManager::_narrow(tempObj);
```