CME Group
A CME/Chicago Board of Trade Company

CME Clearing360 FIXML API

Access and Development Guide

Version: 1.1
12/03/08

Contents

1.0 Overview

CME Clearing360 is CME Group's OTC Clearing platform. The CME Clearing360 Access and Development Guide describes access mechanisms and development guidelines for using the CME Clearing360 API. There are multiple methods to access or interact with the CME Clearing360 platform including:

- FTP
- CME EOS Trader
- WebSphere MQ
- Web Services Gateway (WSG)

Customers can choose, based on their environment, which method they use to access the CME Clearing 360 Platform.

2.0 FTP

Product reference information for instruments traded on the CME Clearing360 platform is available on the CME Group FTP site.
Customers must obtain and load the Security Definition Message and the Derivative Security List Message. These messages are available as static file downloads at 6:00pm daily.
This file is located on the CME Group FTP site:
ftp://ftp.cme.com/pub/settle/

Note: The Derivative Security List Message file size is very large.

3.0 CME EOS Trader

CME EOS Trader™ is CME Group's web-based trading interface. Any instrument listed on CME Globex, including CME, NYMEX and COMEX products can be traded using CME EOS Trader. CME EOS Trader also includes support for Trade Reporter which allows customers to submit Block, Exchange For Physical (EFP), Exchange For Risk (EFR), or OPNT for OTC products to CME Group Clearing.

For more information on CME EOS Trader please go to:
http://www.cmegroup.com/globex/introduction/features-and-functionality/eos-trader.html

4.0 WebSphere MQ

WebSphere MQ is a messaging system from IBM that provides the facility for asynchronous communication between two systems by reading and writing messages to queues or topics. Queues are used for point-to-point communication while topics are used for publish-subscribe communication.

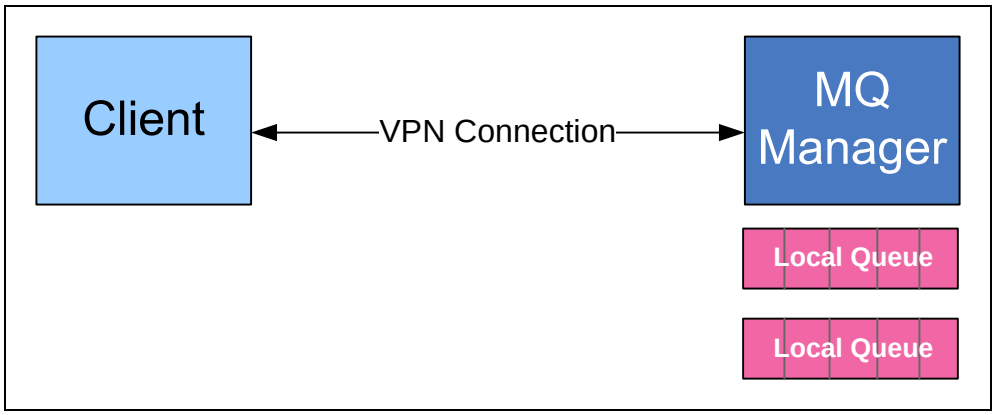
4.1 System Architecture

At the heart of WebSphere MQ system is a Queue Manager that holds queues and topics for applications to read or write message. Client systems connect to this Queue Manager generally via TCP/IP protocol and either write messages to a queue or read messages from a queue.

Basic information that is required to establish a connection with MQ Manager is:

- Host Name
- Port
- Channel Name
- Connection Type
- Queue names

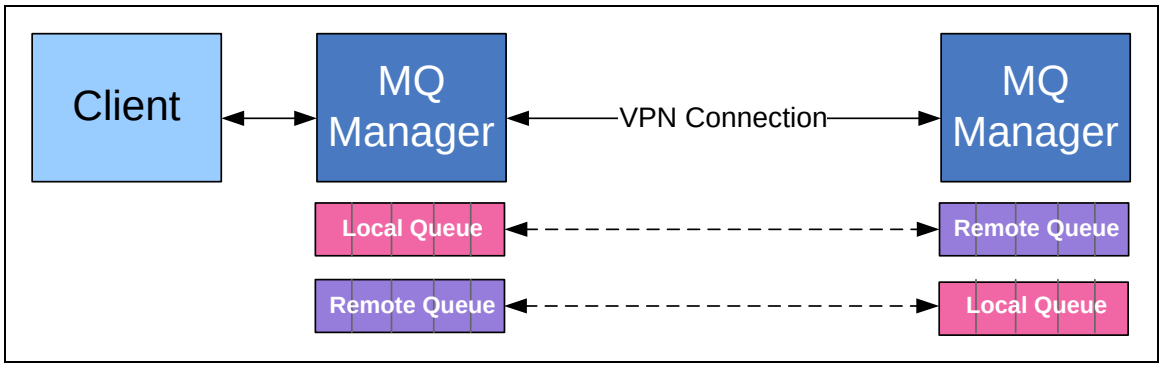
4.1.1 Using the Client Connection



In this model, the client system connects directly to a CME Group MQ manager. The client must configure a VPN connection to communicate with CME Group resources, which in this case, is the MQ manager. After establishing a connection, the client system can write messages to queues or read from queues. The queues are defined by CME Group.

The benefit of this approach is that the client system does not require any special infrastructure on its end other than a VPN connection. However, the resource consumption on the server side is higher than the alternative model.

4.1.2 Using a Queue Manager on the Client Side



In this model, the client system connects to an MQ queue manager that is located on the client side. The client side MQ manager communicates with the CME Group MQ manager using the VPN connection.

In this configuration, a pair of local and remote queues is defined. For example, if the client needs to send a message to CME Group, a local queue would be defined on CME Group MQ manager and a corresponding remote queue would be defined on the Client MQ manager.

The benefit of this configuration is the increased reliability for reading and writing messages to the queues defined on CME Group MQ manager. Two queue managers communicate with each other using proprietary protocol that ensures guaranteed delivery of messages.

4.2 Programming Guidelines

4.2.1 Major Languages

WebSphere MQ supports clients written in major languages and provides appropriate libraries.

Please refer to the following site that containsdocumentation provided by IBM:
http://publib.boulder.ibm.com/infocenter/wmqv6/v6r0/index.jsp?topic=/com.ibm.mq.branding.doc/help_home_wmq.htm

These documents provide a comprehensive reference for programmers to understand the API and mechanisms for connecting to and using messaging with WebSphere MQ.

The languages covered include Java, C++, and .NET.

4.2.2 Java

This section describes how to use the WebSphere MQ client libraries for Java to connect with CME Group Clearing to exchange FIXML messages. The client Java application must include the **com.ibm.mq.jar** and **com.ibm.mqjms.jar** client libraries in its CLASSPATH.

Current version of MQ libraries to be used will be specified elsewhere. The standard JMS API can be used to do most of the work, but the JMS specification does not specify a mechanism for passing configuration information to the JMS objects.

The programming guidelines to utilize WebSphere MQ is described in the following simple steps:

1. Obtain a QueueConnectionFactory. , The JMS API does not specify how to do this. For WebSphere MQ, use the following code to construct a QueueConnectionFactory that is able to bind to a queue manager hosted on another machine.
   - MQQueueConnectionFactory qcf = new MQQueueConnectionFactory();
   - qcf.setHostName(“host_name");
   - qcf.setPort(1414);
   - qcf.setChannel(“channel_name");
   - qcf.setQueueManager(“queue_manager_name");
   - qcf.setTransportType(JMSC.MQJMS_TP_CLIENT_MQ_TCPIP);

   The JMSC.MQJMS_TP_CLIENT_MQ_TCPIP transport type is used to connect to a remote queue manager.

   CME Group will provide values for host name, port, channel name, queue manager name, and inbound and outbound queue names for different environments including QA testing, certification, production, and disaster recovery.

2. The JMS specification also does not specify how to obtain a Queue. For WebSphere MQ, this construct an MQQueue object.
   Queue q = new MQQueue(“queue_name”);

   - The code used to send and receive messages via MQ uses the standard JMS API. First, a session is established.

     QueueConnection conn = qcf.createQueueConnection();
     QueueSession qsession = conn.createQueueSession(true,
     Session.AUTO_ACKNOWLEDGE);
     conn.start();

   The boolean parameter of createQueueSession method indicates whether to use a transacted session. A transacted session is recommended in order to prevent any message loss during failure scenarios.

   - Next, QueueSender and QueueReceiver objects are created.
     QueueSender qsender = qsession.createSender(qoutbound);
     QueueReceiver qreceiver = qsession.createReceiver(qinbound);

- To send a FIXML message, a TextMessage object is constructed and is passed to the QueueSender.
  String fixmlMessage;
  TextMessage msg = qsession.createTextMessage(fixmlMessage);
  qsender.send(msg);

- To receive messages, an implementation of the MessageListener interface may be registered with the QueueReceiver.
  qreceiver.setMessageListener(new MyMessageListener());
  public class MyMessageListener implements MessageListener {
  public void onMessage(Message msg) { }
  }
  Alternatively, the QueueReceiver's receive method may be called. A timeout value in milliseconds may be passed if desired.
  TextMessage msg = (TextMessage) qreceiver.receive(1000);
- Next, the FIXML string is extracted from the message.
  String fixmlMessage = msg.getText();

5.0 Web Services Gateway (WSG)

Web Services Gateway provides access to CME Group's clearing processes over the internet. The system allows customers to submit off-exchange (privately negotiated) trades and receive business confirmations from clearing systems.

CME Clearing360 (clearing platform for off-exchange trades) services can be accessed via the Web Services Gateway (WSG). WSG provides SOAP/HTTPS web services to client systems.

5.1 WSG Architecture

This section describes the WSG architecture that is of customer importance.

5.1.1 Architecture Features

WSG architecture has a number of features that ensure the trade submission process is straight forward, reliable and secure.

5.1.2 Usability

Web services architecture for building service-oriented applications is now widely understood and implemented. The WSG system utilizes non-proprietary, industry-standard technology that allows customers to access WSG with ease, providing them with a simple way to integrate with CME Group's industry-leading clearing system. Use of a standards based approach also isolates customers from the WSG architecture and platform specificity.

5.1.3 Reliability of Trade Submission Service

System reliability is crucial when submitting trades. In any complex system, multiple systems/components interact during a transaction and there is always a possibility that one or more of these components may fail. In the case of submitting trades, such a failure may result in a message loss.

To cope with partial system failure and message losses, WSG provides a Reliable Submit service that utilizes an industry standard specification called WebServices-ReliableMessaging (WS-RM). WS-RM is an open source specification that lays out the protocols for achieving reliable communication between a web services provider and a web service consumer. WS-RM makes sure that a web service request is successfully completed provided there is no return value expected by the service consumer. For more information, please use the reference provided in this document.
WS-RM has been implemented by multiple vendors in their suite of Application Server products. Our offering is based on WebLogic's implementation. However, other vendors such as Microsoft and Apache also support this specification.

Note: A WS-RM web service can be consumed only by another web service that is WS-RM compliant. Ramifications of this requirement are important in that a simple client not running in an Application Server cannot consume reliable web services.

Note: Customers who wish to ensure the reliability of trade submissions by some other programmatic or operational means may submit trades directly to the WSG using a plain non WS-RM Submit service.

5.1.4 Reliability of Business Confirmation Service

WSG architecture also ensures that customers reliably receive business confirmations. Customers can easily recover business confirmation messages lost due to the failure of any system component. Unlike with the trade submissions, WS-RM cannot be used for this purpose. Instead, for the poll web service, WSG utilizes a plain SOAP/HTTPS web service that can be invoked from a standalone application or from an application running in an application server. Messages lost due to component failure can be easily recovered and provided for re-polling via one of the backend systems participating in the business transaction.

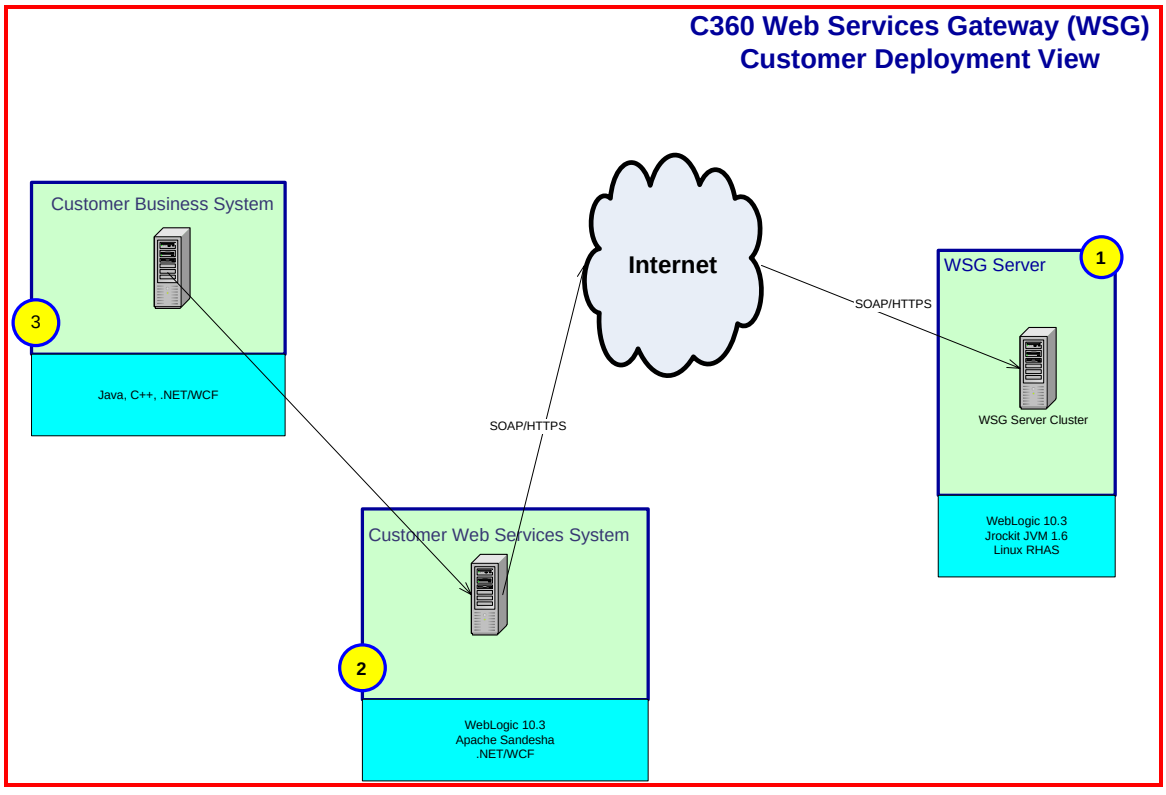
5.1.5 Security

The security of transactions carried out through the WSG is assured through a number of means. WSG secures its services by means of SSL and user certificates.

- WSG web services can be accessed only through HTTPS, which ensures that communication channels are properly encrypted between customer system and WSG.
- WSG also ensures that only authorized customers have access to the WSG system. Customers must use CME Group issued certificates in their web service calls. These certificates carry X509 tokens, a standard format for public key (identity) certificates, attribute certificates, and certificate revocation lists. These tokens are stored in CME Group's LDAP system and WSG uses them to authenticate the service calls.

5.1.6 Customer View of Architecture

The diagram in this section provides a representative view of the system and its interaction with client systems. WSG also interacts with other clearing systems (not shown here) to complete transactions and provide business confirmations back to customer systems.



5.1.7 Customer Business System

The Customer Business System is a client system that manages business transactions involving trades. In this context, the Customer Business System can consume web services provided by WSG for submitting trades and receiving business confirmation.

A Customer Business system can be implemented on any platform that supports consumption of web services. Any systems implemented in Java, C++, C#, etc. are capable of consuming web services, and can be used as a Customer Business System.

5.1.8 Customer Web Services System

If a customer wishes to utilize the reliable trade submission services of WSG, they must connect to WSG using a WS-RM enabled system such as the Customer Web Services System, which can handle invocation of a WS-RM web service. In this case, Customer Business Systems would interact with the Customer Web Services System to utilize reliable trade submission service of WSG indirectly. Customer Business System can also utilize the Customer Web Service system to access business confirmations via poll service provided by WSG. Since the poll service does not use WS-RM, Customer Business System can also access that service directly without using Customer Web Services System as an intermediary.

Customer Web Services System must be implemented using an application server (container) that supports WS-RM. There are a number of such servers commercially available; however, CME Group will only test a handful of them internally, such asWebLogic. It is the customers responsibility to select an application server that meets the required WS-RM specifications.

5.2 WSG Usage

This section describes the main use cases for WSG from a customer's point of view. WSG provides an API for customers to interact with the CME Clearing360 platform and these use cases describe the interaction between customer system and WSG using the API.

5.2.1 Submitting Trades

This section describes the process of a WSG Customer submitting off-exchange trades to CME Clearing360 via WSG.

1. A customer submits a trade using the Customer Business System.
2. The Customer Business System invokes the Trade Submission web service provided by the Customer Web Service System.
3. The Customer Web Service System invokes the Reliable Messaging based Trade Submission web service provided by WSG.
4. The WSG authenticates the submission based on pre-assigned credentials to the WSG Customer. If the authentication succeeds, the WSG queues up the trade to be processed by CME Group clearing systems.

5. The Customer Web Service System (and in turn the Customer Business System) does not receive a notification about the validity of the submitted trade immediately. At this point, no confirmation of the trade is received from the WSG.

Note: Alternatively, the Customer Business System can directly invoke the Trade Submission web service provided by WSG. Note that this method of submitting trades does not provide the reliability of the WS-RM based web service.

5.2.2 Polling for Confirmation Messages

This section describes the process of a WSG Customer polling confirmation messages via WSG.

1. A customer submits a poll request using the Customer Business System. The Customer Business System consumes the confirmation messages for back-end processing.
2. Customer Business System invokes the Confirmation Polling web service provided by the Customer Web Service System.
3. Customer Web Service System invokes the Confirmation Polling web service provided by WSG.
4. Upon request to poll confirmation messages, WSG authenticates the submission based on pre-assigned credentials CME Group issues to the WSG Customer. If the authentication succeeds, WSG retrieves the confirmation message and sends it to the customer.
5. After receiving the confirmation message, Customer Business System may correlate it with the trade that it previously submitted to WSG. At this point, the Customer Business System may persist the message, forward it to another system, or use in any other way according to its backend processing needs.
6. In a standard implementation of WSG Architecture, the Customer Business System continuously polls for confirmation messages. The Customer Business System can choose to receive one or all of the confirmation messages queued for it by CME Group clearing systems. However, WSG will impose a limit on number of confirmation messages returned to the Customer Business System.

Note: Alternatively, the Customer Business System can directly invoke the Confirmation Polling web service provided by WSG. If a customer wishes to maintain a uniform processing model via trade submission, business confirmation messages can be polled via Customer Web Service System.

5.3 Service Features

The WSG web services provide a basic trade submission and confirmation polling API. Additional services for obtaining reference information such as products, accounts, and prices will be provided over time. The following sections capture main service features and responsibilities of CME Group in providing the service and of customers in using the provided services.

5.3.1 Trade Submission

Service Feature	CME Group Responsibility	Customer Responsibility
Availability	This service will be available from Sunday 5:00 PM CST through Friday 5:00 PM CST.	Customer system must be able to handle HTTP error codes due to non-availability of service during off-hours.
Reliability	If the customer is using Reliable Submit web service, this service guarantees that the submitted message is processed by CME Group clearing systems. There is a small possibility of duplicate submission of trades due to failure in the Customer Web Service System. WSG is capable of handling the duplicates by persisting but ignoring those submissions. WSG will ignore a duplicate submission whether caused by a system failure or submitted by a Customer Business System. If the customer chooses to use the non-reliable Submit web service, there is a probability of duplicates or message loss. Duplicate messages will be ignored while lost messages must be resubmitted by the customer system.	Customer must work out recovery scenarios when failure occurs in their components before a successful call is made to WSG. Since WSG will enable duplicate checking for submitted messages, the customer system must be aware of the fact that any duplicate message (with identical FIXML payload) will be ignored. In case of a lost message, it is customer responsibility to resubmit the trade.

Service Feature	CME Group Responsibility	Customer Responsibility
Performance	There are no instantaneous business level acknowledgements for a trade submission and hence the performance is generally constrained by the latencies due to LAN as well as internet conditions. Under most conditions, customer calls for business confirmation messages will be processed in under a second. WSG is able to process requests from multiple HTTP sessions and provides a sufficient level of concurrency for each customer.	The customer system should be designed with an appropriate level of concurrency keeping in mind constraints imposed by Customer Web Service System and Customer Business System.
Monitoring & Troubleshooting	Any errors that can be trapped and logged by WSG will be logged and monitored by CME Group Technical Operations. Corrective action will immediately be initiated and escalation procedures followed according to CME Group procedures. CME Group will also contact the customer immediately if it is determined that the problem directly affects the customer's systems.	If a customer observes errors on his end that could be attributed to WSG operations, the error must be reported to CME Group. The customer must provide contact information for both their operations and technical resources to CME Group.

5.3.2 Confirmation Polling

Service Feature	CME Group Responsibility	Customer Responsibility
Availability	This service will be available from Sunday 5:00 PM CST through Friday 5:00 PM CST.	Customer system must be able to handle HTTP error codes due to non-availability of service during off-hours.

Service Feature	CME Group Responsibility	Customer Responsibility
Reliability	This service is not completely reliable - a polled message has a small chance of getting lost due to failure in the WSG or network under certain conditions. If the customer can detect the loss of a message, CME Group can make that message available for polling again.	Customer must work out recovery scenarios when failure occurs in their components after the polled message is returned to the Customer Web Service system or Customer Business System. The customer must be able to detect a missing confirmation message and contact CME Group for replay of that message. If a customer wishes to maintain a uniform processing model for both trade submission and confirmation polling, business confirmation messages can be polled via Customer Web Service System
Performance	This service returns business level acknowledgements in the form of a FIXML message or a collection of messages. Performance will depend on the number of messages returned. WSG is able to process requests from multiple HTTP sessions and provides a sufficient level of concurrency for each customer. WSG will throttle client requests for confirmation messages to manage its concurrency levels. WSG will also control the number of messages returned at once for one request.	The customer system should be designed with an appropriate polling interval. A very small polling interval would increase load on the WSG and may also hurt the performance of customer systems. The Polling interval should also be based on whether polling is done for one or multiple messages.

Service Feature	CME Group Responsibility	Customer Responsibility
Monitoring & Troubleshooting	Any errors that can be trapped and logged by WSG will be logged and monitored by CME Group Technical Operations. Corrective action will immediately be initiated and escalation procedures followed according to CME Group procedures. CME Group will also contact the customer immediately if it is determined that the problem directly affects the customer's systems.	If a customer observes errors on his end that could be attributed to WSG, the error must be reported to CME Group. The customer must provide contact information for both their operations and technical resources to CME Group.

5.4 Technical Specifications

5.4.1 Software Specifications

Standard/Specification	Description/Reference	Source	Version
WebLogic Server (WLS)	We prescribe use of this version of WebLogic server if customer chooses to use WebLogic as the App Server.	Oracle	10.x or above
Java	JVM/JDK used for building a java based client system.	Sun or Oracle	1.6 or above
Web Services Definition Language (WSDL)	A language to describe web services	W3C	1.1
Simple Object Access Protocol (SOAP)	A lightweight XML-based protocol used to exchange information in a decentralized, distributed environment.	W3C	1.2
Java API for XML Web Services (JAX-RPC)	A specification that defines the Java APIs for making XML-based remote procedure calls (RPC).	Sun	1.1
Web Services Metadata for Java Platform	Defines annotation framework for easily constructing and programming web services.	Sun	JSR 181
Web Services – Reliable Messaging (WSRM)	A protocol to ensure that web service invocation withstands network and other system failures.	OASIS	1.1

Standard/Specification	Description/Reference	Source	Version
Web Services – Security (WS-Security)	Basic security specification for secure messaging needed for web services.	OASIS	1.1
Web Services – Trust (WS-Trust)	The goal of this specification is to enable applications to construct trusted [SOAP] message exchanges. This trust is represented through the exchange and brokering of security tokens. This specification provides a protocol agnostic way to issue, renew, and validate these security tokens.	OASIS	1.3
Web Services – Secure Conversation (WS-SC)	Web Services Secure Conversation Language specification defines extensions that build on Web Services Security (WS-Security) 1.1 and Web Services Trust Language (WS-Trust) 1.3 to provide secure communication across one or more messages. This specification defines mechanisms for establishing and sharing security contexts and deriving keys from established security contexts (or any shared secret).	OASIS	1.3

5.4.2 Service Specifications

Service Interface	Description	Arguments	Return Value
submit	Used for submitting trades to CME Clearing360 platform. This method takes in Trade in a FIXML format.	String (trade in FIXML format)	None
submitReliably	Used for submitting trades to CME Clearing360 platform with guaranteed delivery to the WSG server. This method takes in Trade in a FIXML format.	String (trade in FIXML format)	None
poll	Used for polling confirmation messages waiting for a customer. This method returns waiting confirmation messages based on the requested number of messages . This method has to be called repeatedly to obtain confirmations.	Integer (number of confirmation messages requested to be returned)	String [ ]

5.5 Development Guidelines

This section is targeted for developers that are planning to develop client systems to interact with and use web services provided by WSG.

The starting point for coding to interact with WSG is the WSDL (Web Services Definition Language) that CME Group will provide for different web services. There are two ways client systems can access and use WSDL:

- via a physical file
- via a URL provided by CME Group.

5.5.1 Web Services Definition Language

The following WSDL is provided for building client code and accessing WSG services.

WSDL	Location	Services
WSG_Service	File: WSG_Service.wsdl URL: https://c360wsg.cmegroup.com\WSG_Service\WSG_Service?WSDL	Submit Poll
WSG_SubmitReliably	File: WSG_SubmitReliably.wsdl URL: https://c360wsg.cmegroup.com\WSG_Service\WSG_SubmitReliably?WSDL	SubmitReliably

5.5.2 Security

Client systems need to work with three different security requirements to utilize WSG web services:

- To communicate using HTTPS/SSL with WSG, the client system will need a root certificate from Verisign. Verisign is the Certification Authority (CA) that CMEG uses. This can be easily obtained from Verisign website.
- To identify itself as a customer that is recognized by WSG, the client system needs a User Certificate (X509) issued by CME Group.
- If the client system plans to utilize WS-RM web service provided by WSG, it also needs the server certificate (public key) used by WSG. This will be provided by CME Group.

5.5.3 General Coding Procedures

Depending upon your choice of language, you may need to complete the following:

- Compile WSDL to create client side stubs.
- In your code, use these stubs to access WSG web services and carry out your business processing.
- In your code, include security related programming segments to establish SSL connection with the server as well as to pass in certificate (X509 tokens) information for authentication.

- In case you are using an Application Server to host proxy web services, you will carry out a number of security related configuration steps. This configuration will most likely reduce the programming steps related to security in your code.
- In case you are using WS-RM web service, you would also have a programming segment in your implementation of proxy web service that sets up the server certificate.

5.5.4 Build Steps

Build steps would vary based on your choice of language and platform; however, here are major steps that a typical Java environment uses.

A typical example of an ANT task for creating client side stub source is:

```
<clientgen>
        wsdl="file:///${basedir}/WSG_Submit.xml"
        destDir="${client.ear.dir}/APP-INF/classes"
        packageName="com.xxx.yyy.zzz.client"
        classpath="${java.class.path}"

<clientgen>
```

Once a stub source is created, the following ANT task can compile those stubs in Java classes:

```
<javac srcdir="${client.ear.dir}/APP-INF/classes"
        destdir="${client.ear.dir}/APP-INF/classes"
        includes="com/xxx/yyy/client/**/*.java" />
```

If you are creating a Web Services System on your end, you would then use these complied stubs to create a deployable component for your Java Application Server:

```
<jwsc
        srcdir="SOURCEDIR"
        destdir="${client.ear.dir}" keepGenerated="true"
        deprecation="${deprecation}" debug="${debug}">
        <classpath refid="project.class.path" />
        <jws file="WSG_ClientImpl.java" />
</jwsc>
```

If you are not using a Web Service System, you would then use the stub Java classes in your client code and compile your client code to create components that would access the WSG services directly.

5.5.5 Use of Annotations

Annotations provide a useful way of reducing programming effort in Java. Java 1.5 introduced the notion of annotations.

JSR 181 defines annotation specifications for web services. Many Application Servers leverage this specification to provide compilers that would compile annotated web service files to runtime code, reducing the effort required to code web services.

5.5.6 Using Submit Service

Please consider the following issues when coding for the Submit Service.

If you are using the reliable version, it can only be invoked by another web service in an Application Server that supports WS-RM specifications. In this case you are guaranteed that a submitted message has been received by WSG and a business confirmation message will soon be provided for the submission.

If you choose to use the regular version of this service, you may have to programmatically ensure that WSG has received your submission by polling for confirmations and resubmitting in case a corresponding confirmation cannot be found within a reasonable time. The re-submit interval depends on the individual customer's backend processing needs and policies. Duplicate trade submissions will be ignored by the WSG.

5.5.7 Using Polling Service

Please consider the following issues when coding for using the Poll Service.

To process one message at a time, use the single poll version by setting the number of requested messages to 1. If you want to receive multiple confirmation messages, you can set the number of requested messages to more than 1.

Please ensure that you use a polling interval that meets your needs and does not overburden the WSG server. WSG will use a throttling mechanism to manage load on its servers.

If you have a heavy volume profile, it is better to request multiple messages at a time to decrease network overhead. However, a trade off associated with reliability must be kept in mind since recovering a large batch of messages lost due to system failures could be

more challenging. WSG will restrict the number of messages it returns for a multi-poll web service call.

5.5.8 Reference Implementation

A reference implementation in Java can be made available upon request. This implementation can help you understand how a web service is created to act as a proxy for the web service provided by WSG. This will also include a client that uses your web services to access WSG web services and a client that directly uses WSG web services.

5.6 Configuration Guidelines

If you intend to leverage WSG Reliable Submit web service, you must create a corresponding web service in your system that can invoke a Reliable Web Service in WSG. Web Services run in application containers, so you will need to create and deploy your web service in an Application Server.

Choice of Application Server is determined by the individual user of WSG. If the Application Server can support specifications defined in previous section, WSG services can be used.

5.6.1 General Configuration Requirements

The configuration steps required for an Application Server will depend on that Application Server. In general, there are a number of configuration steps that must be taken:

1. Create and configure a server instance according to your Application Server guidelines.
2. Configure your App Server to be able to communicate with WSG using HTTPS. This requires a Root CA (Certification Authority) certificate prescribed by CME Group.
3. Configure your App Server to be able to provide authentication information via X509 tokens to the target service hosted by WSG. For this you will be provided with a User Certificate from CME Group.
4. If you intend to use WS-RM web services provided by WSG, you may have to configure some mechanism to persist messages on your App Server. For example, WebLogic server uses Store and Forward Agents.
5. You may also have to configure the runtime environment of your App Server to enable security for making web service calls into WSG.
6. There may be additional steps required depending on your choice of App Server.

5.6.2 WebLogic Server Configuration

This section lays out specific steps needed to configure a WebLogic server for invoking web services provided by WSG via proxy web services.

5.6.3 Regular (Non-RM based Web Services)

- WebLogic Server Installation
  - Create a WebLogic domain.
  - You can choose to deploy web services on the Admin server or on a cluster of Managed servers. If you choose the latter, you must create both a managed servers and a WebLogic cluster.
  - Create the addition directories based on your deployment directory structure.
- Create and configure keys and certificates
  - Create the identity key store using either Java keytool utility or WebLogic utility. This key store is used for configuring an X509 Credential Provider that lets WSG authenticate the client.
  - Create the trust key stores using Java keytool utility. Then import root CA certificate from Verisign. This is used for establishing SSL with WSG.
  - Obtain the user certificate and private key from CME Group. This is used for authentication via X509 Credential Provider.
  - Create the user key store using WebLogic utility. This requires the provided user certificate and private key from CME Group.
- Configure key stores and SSL on WebLogic Server
  - Configure Identity Key Store with the previously generated Identity key store and with correct keystore pass-phrase.
  - Configure Trust Key Store with the previously generated key store and with correct keystore pass-phrase.
  - Configure SSL using Identity Key Store generated in previous step.
- Configure Web Services Security for WebLogic Domain
  - Create default_wss (use exactly as specified) web services security. This configuration item contains credential providers and token handlers for enabling web services security.
  - Create X509 token handler. This is needed for web service calls to be identified with an X509 token.
    - TokenType=x509
    - ClassName=weblogic.xml.crypto.wss.BinarySecurityTokenHandler
  - Create X509 Credential Provider.
    - TokenType=x509
    - ClassName=weblogic.wsee.security.bst.ServerBSTCredentialProvider

        - Configure X509 Credential Provider with the Identity Key Store values.
            - ConfidentialityKeyAlias, ConfidentialityKeyPassword, ConfidentialityKeyStore, ConfidentialityKeyStorePassword
            - IntegrityKeyAlias, IntegrityKeyPassword, IntegrityKeyStore, IntegrityKeyStorePassword
- Configure Security Realm for WebLogic Domain
    - Configure DefaultIdentityAsserter to support X509 token. This is needed to provide authentication information to WSG server.
    - Create PKI Credential Mapper using User Key Store info created in a previous step. Then create PKI Credential Mappings using the same Key Store information. This is needed to access the resources on WSG.
- Configure Runtime Environment
    - Start WebLogic server script should add the following variables:
        - -Djava.protocol.handler.pkgs=javax.net.ssl
        - -Dweblogic.webservice.client.ssl.strictcertchecking=false
        - -Dweblogic.security.SSL.ignoreHostnameVerification=true

5.6.4 Reliable Messaging based Web Services

Additional configuration is required to invoke WS-RM web services.

- WebLogic Server Installation
    - Update domain with web service template (*wlshome/common/templates/applications/wls_webservices.jar*) to support the reliable messaging.
- Configure Web Services Security for WebLogic Domain
    - Create Credential Providers for Secure Conversation needed for Reliable Messaging based Web Service calls.
        - Create SCT (Secure Credential Token) Provider.
            - TokenType=SCT
            - ClassName=weblogic.wsee.security.wssc.v13.sct.ServerSCCredentialProvider
        - Create DK (Derived Keys) Credential Provider.
            - TokenType=DK
            - ClassName=weblogic.wsee.security.wssc.v13.dk.DKCredentialProvider
- Configure Store And Forward Agent

- Create a new SAF agent or use the default one. This is used by WS-RM infrastructure to persist service call (or messages) to guarantee the delivery to the target web service host.

5.7 Message Specification

All the messages submitted to CME Group via the WSG must follow FIXML specifications. If the messages are not structured according to published specs, they will be rejected, and the Customer Business System will receive reject message through the poll web service.

There are two main message specifications relevant to WSG services:

- Trade Capture Report Message
- Trade Capture Report Acknowledgement Message

These are available at http://www.cmegroup.com/education/C360/index.html.

For formatting a valid message for trade submission via WSG, the following two FIXML fields must carry the Common Name (CN) issued to the customer by CME Group. This common name is defined on the User Certificate and is used to authenticate the sender. Without a valid value in these fields, the submission will be rejected.

- SID (Sender Comp ID)
- InptSrc (Trade Input Source)

6.0 Revision History

Version	Date	Author	Description
1.0	11/12/08	LM	Initial release of document.
1.1	12/03/08	CR	Update section 5.4.1, Web Services – Reliable Messaging (WSRM) to 1.1.