IBM Emptoris Contract Management

Web Services Integration Guide

Version 10.0.2.5

SC27-5348-03

IBM Emptoris Contract Management

Web Services Integration Guide

Version 10.0.2.5

SC27-5348-03

Copyright

Note: Before using this information and the product it supports, read the information in “Notices” on page 35.

Contents

Chapter 1. Introduction

IBM® Emptoris® Web Services represent functional interfaces or services that expose the IBM Emptoris Contract Management logical object model and the associated process models to external access.

This publication describes how to design integration code to use IBM Emptoris Web Services.

About this Guide

This publication describes how a system integrator can call the IBM Emptoris Contract Management interface through IBM Emptoris Web Services, and gain access to the object and process model.

Intended Audience

This publication is designed for use by system integrators and Professional Services Representatives.

A basic understanding of Emptoris Contract Management functionality and J2EE Webbased environments, as well as SOAP, WSDL, XSD, and XML syntax is required.

Prerequisites for Using Web Services

The effective use of web services requires an understanding of the Emptoris Contract Management application, the statuses and locking mechanism of a contract, the security model, and the requirements for performing any action in Emptoris Contract Management—whether using the UI or a web service. Familiarity with the IBM Emptoris Contract Management User Guide, IBM Emptoris Contract Management Administration Guide, and IBM Emptoris Contract Management Security Administration Guide is required.

Not all possible data that can be passed to Emptoris Contract Management using a web service is necessarily accepted by the application. In such a case, although an error issues, the underlying causes of the failure are not necessarily explicit. To understand the data or prerequisites for the action, an understanding of the equivalent action in the Emptoris Contract Management UI is recommended.

The information in this guide is provided under the following assumptions:

- Emptoris Contract Management is correctly installed and configured as described in the IBM Emptoris Suite Installation Guide.
- Adequate network permissions are granted.
- Network access to the Emptoris Contract Management application server is available.
- The ISDK version (see “Web Services ISDK ToolKit” on page 4) matches the version of the installed Emptoris Contract Management application.

Where to Find Information in this Guide

The chapters of this publication contain the following information:

Chapter 1, “Introduction,” on page 1 provides an overview of web services, WSDLs, and communication protocols.

Chapter 2, “Integration,” on page 7 describes IBM Emptoris Web Services integration processes and principles, and provides sample files that illustrate request and response messages for organization-related web services.

Chapter 3, “Appendix: List of Web Services,” on page 27 provides a list of available web services.

Web Services Overview

Web services are modular enterprise applications that can be published, discovered, and invoked dynamically across the web.

These services use open, XML-based standards and transport protocols to achieve data transfer and integration between different applications. Web services are platform- and language-independent.

A Web Service is an open programmatic interface that allows you to perform a standard UI task or action, without accessing the UI. It can be expressed and accessed using XML and XML messaging. Each web service has a service, input type, output type, and exceptions or "faults", and comprises some content, a process, or both.

Web services perform functions which can vary from simple requests to complicated business processes. Once a web service is deployed, other applications (and other web services) can discover and invoke the deployed service.

The web services are defined in Java™ while web service types are defined in WSDL (Web Service Definition Language)/XML schema language. The syntax of each web service is defined in XML schema language in order to allow a more explicit description of properties than is possible with Java. For example, it can specify minimum and maximum entry length of attributes, which is not possible with Java alone.

About Emptoris Contract Management Web Services

IBM Emptoris Web Services allow you to access various contract management and administration functionality across the Emptoris Contract Management object model, which are otherwise accessible through the UI. The web services operations perform security permission checks similar to that performed by the Emptoris Contract Management application.

You can use the web services to retrieve, query, create, add, update, or delete information from the Emptoris Contract Management Repository. The operations you can perform depend on your security permissions within Emptoris Contract Management. For more details on security permissions, see the IBM Emptoris Contract Management Security Administration Guide.

For each object that has a secondary object, secondary functions apply. For example, addTerm, updateTerm, and removeTerm apply to the list of Term objects referenced by a contract. A get(id) web service returns a descriptor or a representation of the object, whereas a get for a contract also returns a reference to each related object (a get operation on a contract returns references to both the internal and external organizations for that particular contract).

A descriptor contains details of the object's attributes. For example, a get for a contract returns a descriptor that contains all contract attributes such as contract name, status, class, and so on. A reference is a unique identifier—either an ID or an IDplus- revision. For example, Terms, Lines, and Clauses each have a revision number that specifies the number of times it was revised in the contract. In order to return the details of each related object, you need to iterate a get for each ID or ID-plus-revision returned.

Each object or "type" ID is defined using XSD (XML Schema Definition). This type describes the attributes of the object.

The web service client can be a Java application using stubs, or any application using SOAP. The web service EJBs (Enterprise Java Beans) perform security validations before passing data onto the Service Object layer. The Service Object layer is the most important layer of the Web Service integration process. It performs additional validations before permitting access to the Business Object layer. The Service Object layer is also the transformation engine for the web services. This layer transforms data as prescribed in the WSDL to communicate with the BO layer, in formats that the BO layer can understand.

The following table lists the standard operations you can perform using IBM Emptoris Web Services.

Table 1. Web Services Standard Operations

Operation	Description
get	Expects the ID of the object as its input and returns the corresponding object data. For example, if the input is an Address ID or Organization ID, it returns Address data or Organization data, respectively.
search	Expects search parameters as its input and returns a list of corresponding IDs. For example, if the input is a search for addresses based on a country, it returns a list of Address IDs. To obtain the properties of the IDs returned in the search, a get operation needs to be called for that ID.
create	Expects the minimum required values to create the new object (such as a Contract, an Address, or an Organization) as its input, and returns an ID for the new object.
update	Expects the ID of the object and other attributes that need to be updated, as its input. If the update is not successful, a fault results.
delete	Expects the ID of the object as its input. If the delete is not successful, a fault results.

The IBM Emptoris Contract Management Bulk Load Utilities enable an IBM Emptoris Customer to load data from external applications or sources into Emptoris Contract Management. These utilities can be used to load contract data or legacy contracts of any category—"Buy" (Purchase Agreement), "Sell" (Sales Agreement) or "Other"— in bulk quantities. For more information on bulk loading of data, see the IBM Emptoris Contract Management Bulk Load Utilities Guide.

Time Zones, Locales, and Dates

Emptoris Contract Management web services support the ability to configure dates according to locale and time zone. The web services assume a yyyy-mm-dd and HH:mm:ss format wherever dates and times are included in request and response messages.

Web Services ISDK ToolKit

The IBM Emptoris Web Services package is distributed in the form of an ISDK (Integration Software Development Kit).

You can download the toolkit from the Emptoris Contract Management Online Help. In the online help, go to the Contract Management BLU and ISDK Toolkits topic to download the isdk.tar file.

You can also access the toolkit from the following location:

```
/opt/emptoris/apps/<dateandtimestamp>/ECM_<release & build number>/LIB
```

The ISDK toolkit contains the following items:

- The wsdl directory containing the complete WSDL defining the Emptoris Contract Management web services.
- The webservices-client.jar file located in the jars directory. This is a library of Java classes which provide a client with access to the Emptoris Contract Management web services through a java interface.
- The SharedSecretTokenGenerator.sh utility located in the bin directory. This script provides for the generation of shared secret tokens to facilitate web service authentication using Shared Secret Authentication.

Communication Protocols and Languages used by Web Services

The basic platform of a web service is XML over HTTP. For platform support services such as discovery, transactions, security, and authentication, other communication protocols and languages apply.

IBM Emptoris Web Services use the following standard protocols and languages:

- Java
- SOAP1
- XML
- XSD
- WSDL
- UDDI

To send inbound messages to a Emptoris Contract Management application server, post your messages to the following URL:

```
https://<hostname>.<domain>:<SSLport>/webservices/services/<ObjectServices>
```

Where <ObjectServices> is the name of the web service. For example, OrganizationServices.

Note: The web service name is case-sensitive.

For example, if the login page for your Emptoris Contract Management installation is:

http://www.mycompany.com:7012/contracts/login.adv

And you want to use Web services of the type, “organization”, then the URL for posting messages is:

http://www.mycompany.com:7012/contracts/webservices/services/OrganizationServices

Any message posted to this address is presumed to be in XML format, and is handled as such.

For more details on accessing the web services using SOAP or Java client stubs, see “Calling Web Services Using SOAP over HTTP” on page 7, and “Calling Web Services Using Java Client Stubs” on page 12.

Web Services Description Language (WSDL)

The WSDL is the key element enabling web services. It is an XML file that describes the web service’s interfaces (address, messages, operations, etc.) to the outside world.

It is also a key to interoperability, since different clients using different languages can use the same service, regardless of the underlying technology.

The WSDLs provided with the IBM Emptoris Web Services package are grouped according to Emptoris logical objects and processes. Each web service provides a specific functional interface or service that exposes the Emptoris Contract Management logical object model and the associated process models to external access.

Note: WSDL is not generated if web services are accessed through the PWS URL. Access to web services is blocked at the CWS layer. Access to the WSDL is allowed only directly from the application server.

WSDL Components

Each WSDL contains the categories of components listed here. To view the components in a WSDL definition, append ?WSDL to the web service URL as follows:

```
https://&lthostname&gt;.&ltdomain&gt;:<SSLport>/webservices/services/
<ObjectServices>?WSDL
```

WSDL components are categorized as follows:

Types. The types element encloses data type definitions that are relevant for the exchanged messages. For maximum interoperability and platform neutrality, WSDL prefers the use of XSD as the canonical type system, and treats it as the intrinsic type system.

Messages. Input and output parameters of the operations. Messages consist of one or more logical parts. Each part is associated with a type from some type system using a message-typing attribute. The set of message-typing attributes is extensible. WSDL defines several such message-typing attributes for use with XSD such as element (an XSD element using a QName) and type (an XSD simpleType or complexType using a QName).

Operations. Abstract description of an action supported by the service.

Port Type. Named set or logical grouping of abstract operations and the abstract messages involved.

Bindings. Protocol to be used for accessing the operations. A binding defines message format and protocol details for operations and messages defined by a particular port type. There may be any number of bindings for a given port type.

Service. Address of the service. A service groups a set of related ports together.

Web Services Categories

The following table lists the various categories of IBM Emptoris Web Services and provides a brief description of each.

Table 2. Web Services Categories

Category	Description
Contract	Describes the operations for a Contract.
Contract Class	Describes the operations for a Contract Class.
Contract Template	Describes the operations for a Contract Template.
Clause Template	Describes the operations for a Clause Template.
Currency	Describes the operations for a Currency.
Custom Property Definition	Describes the operations for a Custom Property.
Interview Instance	Describes the interview-related activities for an Interview Instance.
Organization	Describes the operations for an Organization, Address, or Individual.
Price List	Describes the operations for a Price List.
Product	Describes the operations for a Product.
Relationship Type	Describes the operations for a Contract Relationship.
Security	Describes the operations for a User Group and its Users.
Single Sign-On	Describes the single sign-on operations to the Emptoris Contract Management Home Page or a single Contract.
Task	Describes the operations for a Contract Task.
Term Definition	Describes the operations for a Term Definition.
Value List	Describes the operations for a Value List.

For a complete listing of IBM Emptoris Web Services, see Chapter 3, “Appendix: List of Web Services,” on page 27. To view the WSDL, XSD, and HTML documentation of each web service, see the ISDK (described in “Web Services ISDK ToolKit” on page 4).

Chapter 2. Integration

This chapter describes the IBM Emptoris Web Services integration technologies and procedures, as well as data management considerations.

Integration Process

The web services engine uses Apache Axis, an implementation of SOAP—a Java to XML binding framework. It provides for the creation of Java classes to serialize or unserialize XML-based data. The Axis engine is deployed as a WAR (Web Module Archive), to work as a servlet over HTTP.

Axis generates a skeletal implementation of the web service from WSDL as a Java class. An Axis service redirects the call to this generated service, performs authentication, calls the EJB for authorization, and then calls the service object corresponding to the target business object. The service object performs validations, and then calls the business object, where additional validations may take place.

Calling Web Services Using SOAP over HTTP

The SOAP request message consists of a dicartaInvocation tag that provides user and system authentication information, the action to be performed, and the object to be affected by the action. The elements and attributes contained within this tag are sourced from the WSDL and XSD files provided in the ISDK. This section describes the SOAP request message format expected by Emptoris Contract Management, and the resulting response.

SOAP Request Message Format

A dicartaInvocation message consists of a Header and a Data section. The Header section contains user login and message-related information. The Data section contains the message action and the associated object.

The example below demonstrates the SOAP request message syntax in the form of a request to retrieve a contract from Emptoris Contract Management.

Example: SOAP Request message

```
<SOAP-ENV:Envelope

xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"

xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"

xmlns:xsd="http://www.w3.org/1999/XMLSchema">

<SOAP-ENV:Body>

<get-properties-request

xmlns="http://www.dicarta.com/contracts/services/contract"

xmlns:ns1="http://www.mycompany.com/contracts/types/auth"
```

```
xmlns:domain-contract="http://www.mycompany.com/contracts/types/domain/
contract"

xmlns:domain-contract-template="http://www.mycompany.com/contracts/types/
domain/contract-template"

xmlns:domain-common="http://www.mycompany.com/contracts/types/domain/
common"

xmlns:domain-party="http://www.mycompany.com/contracts/types/domain/party"

xmlns:contract="http://www.mycompany.com/contracts/services/contract"

xsi:schemaLocation="rsrc/webservices/services/contract/contract-services-
types.xsd">

xmlns="http://www.dicarta.com/contracts/services/contract"

xmlns:ns1="http://www.mycompany.com/contracts/types/auth"

xmlns:domain-contract="http://www.mycompany.com/contracts/types/domain/
contract"

xmlns:domain-contract-template="http://www.mycompany.com/contracts/types/
domain/contract-template"

xmlns:domain-common="http://www.mycompany.com/contracts/types/domain/
common"

xmlns:domain-party="http://www.mycompany.com/contracts/types/domain/party"

xmlns:contract="http://www.mycompany.com/contracts/services/contract"

xsi:schemaLocation="rsrc/webservices/services/contract/contract-services-
types.xsd">

<authentication>

<ns1:user>dicarta-user</ns1:user>

<ns1:credential>

<ns1:shared-secret>

<ns1:source> </ns1:source>

<ns1:date>2005-06-15T00:00:00</ns1:date>

<ns1:algorithm>MD5</ns1:algorithm>

<ns1:token>XaQyXkmsVpfcbNa0HyNrRQ==</ns1:token>

</ns1:shared-secret>

</ns1:credential>
```

```
</authentication>

<request-data>

<domain-contract:id>eca63398864342c2baa455099d5f06ce</domain-contract:id>

</request-data>

</get-properties-request>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>
```

SOAP Response Message Format

The example in table below is a sample SOAP response message in response to a request to retrieve a contract from Emptoris Contract Management.

Example: SOAP Response message

```
<?xml version="1.0" encoding="utf-8"?>

<soapenv:Envelope

xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"

xmlns:xsd="http://www.w3.org/1999/XMLSchema"

xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance">

<soapenv:Header>

<ns1:sessionID

soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"

soapenv:mustUnderstand="0"

xmlns:ns1="http://xml.apache.org/axis/session">-159494204549955334

</ns1:sessionID>

</soapenv:Header>

<soapenv:Body>

<get-properites-response xmlns="">

<ns1:id xmlns:ns1="http://www.mycompany.com/contracts/types/domain/
contract">eca63398864342c2baa455099d5f06ce</ns1:id>

<ns2:revision xmlns:ns2="http://www.mycompany.com/contracts/types/domain/
contract">2</ns2:revision>
```

```
<ns3:template-id xmlns:ns3="http://www.mycompany.com/contracts/types/
domain/contract">2dcdc3d30d024a979c0dfedea98b94fa</ns3:template-id>

<ns4:name xmlns:ns4="http://www.mycompany.com/contracts/types/domain/
contract">Contract Created Using WebServices</nprint singles4:name>

<ns5:contract-number xmlns:ns5="http://www.mycompany.com/contracts/types/
domain/contract">569</ns5:contract-number>

<ns6:title xmlns:ns6="http://www.mycompany.com/contracts/types/domain/
contract">WebServices template</ns6:title>

<ns7:category xmlns:ns7="http://www.mycompany.com/contracts/types/domain/
contract">Buy</ns7:category>

<ns8:class xmlns:ns8="http://www.mycompany.com/contracts/types/domain/
contract">ARTA</ns8:class>

<ns9:status xmlns:ns9="http://www.mycompany.com/contracts/types/domain/
contract">Draft</ns9:status>

<ns10:origin xmlns:ns10="http://www.mycompany.com/contracts/types/domain/
contract">AUTHORED</ns10:origin>

<ns11:substatus xsi:null="true" xmlns:ns11="http://www.mycompany.com/
contracts/types/domain/contract"/>

<ns12:region xsi:null="true" xmlns:ns12="http://www.mycompany.com/
contracts/types/domain/contract"/>

<ns13:notes xmlns:ns13="http://www.mycompany.com/contracts/types/domain/
contract"/>

<ns14:negotiation-type xmlns:ns14="http://www.mycompany.com/contracts/
types/domain/contract">AcceptOnly</ns14:negotiation-type>

<ns15:negotiation-status xsi:null="true" xmlns:ns15="http://
www.mycompany.com/contracts/types/domain/contract"/>

<ns16:internal-organization xmlns:ns16="http://www.mycompany.com/contracts/
types/domain/contract">

<ns17:name xmlns:ns17="http://www.mycompany.com/contracts/types/domain/
organization">diCarta</ns17:name>

<ns18:reference xmlns:ns18="http://www.mycompany.com/contracts/types/
domain/organization"><ns18:id>b39e6d737b994febb7946d3f45261596</ns18:id>

</ns18:reference>

</ns16:internal-organization>

<ns19:internal-contact xmlns:ns19="http://www.mycompany.com/contracts/
types/domain/contract">
```

```
<ns20:name xmlns:ns20="http://www.mycompany.com/contracts/types/domain/
individual">Internal User1</ns20:name>

<ns21:reference xmlns:ns21="http://www.mycompany.com/contracts/types/
domain/individual">

<ns21:id>a7c9ae0827b84140bf9e69592791543a</ns21:id>

</ns21:reference>

</ns19:internal-contact>

<ns22:external-party xmlns:ns22="http://www.mycompany.com/contracts/types/
domain/contract">

<ns23:name xmlns:ns23="http://www.mycompany.com/contracts/types/domain/
party">ExtComp1</ns23:name>

<ns24:reference xmlns:ns24="http://www.mycompany.com/contracts/types/
domain/party">

<ns24:id>98d2fddb571241938bd3ad90844962b2</ns24:id>

<ns24:type>organization</ns24:type>

</ns24:reference>

</ns22:external-party>

<ns25:external-contact xmlns:ns25="http://www.mycompany.com/contracts/
types/domain/contract">

<ns26:namexmlns:ns26="http://www.mycompany.com/contracts/types/domain/
individual">Jennifer Kettering</ns26:name>

<ns27:reference xmlns:ns27="http://www.mycompany.com/contracts/types/
domain/individual">

<ns27:id>a7c9ae0827b84140bf9e69592791543a</ns27:id>

</ns27:reference>

</ns25:external-contact>

<ns28:effective-start-date xmlns:ns28="http://www.mycompany.com/contracts/
types/domain/contract"/>

<ns29:effective-end-date xmlns:ns29="http://www.mycompany.com/contracts/
types/domain/contract"/>

<ns30:audit-info xmlns:ns30="http://www.mycompany.com/contracts/types/
domain/contract">

ns31:created-by xmlns:ns31="http://www.mycompany.com/contracts/types/
domain/common">aaddd8174bf9464fae8f908e089b1320</ns31:created-by>
```

```
<ns32:created-on xmlns:ns32="http://www.mycompany.com/contracts/types/
domain/common">2005-06-08T13:52:24.000Z</ns32:created-on>

<ns33:modified-by xmlns:ns33="http://www.mycompany.com/contracts/types/
domain/common">

aaddd8174bf9464fae8f908e089b1320</ns33:modified-by>

<ns34:modified-on xmlns:ns34="http://www.mycompany.com/contracts/types/
domain/common">2005-06-08T13:52:24.000Z</ns34:modified-on>

</ns30:audit-info>

</get-properites-response>

</soapenv:Body>

</soapenv:Envelope>
```

Calling Web Services Using Java Client Stubs

You can also call IBM Emptoris Web Services using the Java client stubs provided with the ISDK. The example in the table below illustrates how to call a web service using a Java client stub. All classes required to make this call are available in the jar file webservices-client.jar, available in the jars directory of the ISDK.

Example: Calling the getContract web service using a Java Client Stub

```
import java.net.*;

import java.io.*;

import java.lang.*;

import java.math.*;

import java.util.*;

import com.dicarta.webservices.types.common.Credential;

import com.dicarta.webservices.types.common.Authentication;

import com.dicarta.webservices.services.contract.*;

import com.dicarta.webservices.types.domain.*;

import com.dicarta.webservices.faults.*;

public class GetContract {

public static void main(String[] args) throws Exception {

if (args.length != 1) {

usage();
```

```
}

String hostUrl = args[0];

getContract(hostUrl);

}

public static void getContract(String hostUrl) {

ContractServices_BindingStub binding = null;

/* The following code obtains the binding stub for a given webservice, for
example, Contract or Organization:

*/

try {

java.net.URL endpoint;

/*This address must contain the host:port on which the diCarta Contracts
service is deployed.

*/

String serviceAddress = hostUrl +

"/webservices/services/ContractServices";

endpoint = new java.net.URL(serviceAddress);

binding = (ContractServices_BindingStub)

new ContractServices_ServiceLocator

().getContractServices(

endpoint);

} catch (Exception e) {

e.printStackTrace();

System.exit(1);

}

//Time out after a minute.

binding.setTimeout(60000);

binding.setMaintainSession(false);

/*To maintain this session in order to make multiple webservice calls, set
binding.setMaintainSession to true.
```

```
*/

//Make the webservice call:

try {

//Build the Authentication piece:

Credential credential = new Credential();

credential.setPassword("dicarta-user");

Authentication auth = new Authentication("intusr1", credential);//Build the
request data piece:

ContractReference reference = new

ContractReference();

reference.setId("AContractUUID");

GetRequest request = new GetRequest();

request.setAuthentication(auth);

request.setRequestData(reference);

Contract contract = binding.get(request);

} catch (AuthenticationFault e) {

System.err.println ("Unable to authentication");

System.err.println ("Unable to authentication");

} catch (AuthorizationFault e) {

System.err.println ("Invalid security permission for operation");

} catch (ObjectNotFoundFault e) {

System.err.println ("Invalid contract specified");

} catch (SystemFault e) {

System.err.println ("Unknown fault occured");

} catch (Exception e) {

System.err.println ("Unknown error occured");

}

}

private static void usage() {
```

```
System.out.println ("Usage: java GetContract <hostUrl>");

System.exit(1);
```

Handling Authentication

The Emptoris Contract Management XML protocol offers two types of credentials for authentication: password and shared secret. The password approach has the disadvantage of including the plain text password in the message.

The shared secret approach is more secure since validation requires access to information (the shared secret) that is not included in the message.

The <Token> in the shared secret message is the key to its approach. During integration, both systems are set up to know a secret authentication key. The secret is then included in a data packet that is encrypted using a message digest algorithm, and then encoded so that it can be embedded in an XML message. The remainder of the data is passed in the XML request message as shown in the example in table "Calling Web Services Using SOAP over HTTP" on page 7.

When Emptoris Contract Management receives the message, it constructs the same data packet from the XML content and appends the locally known shared secret. After performing the same encryption and encoding, it compares the result to the <Token> in the XML message. If it matches the external system, it considers the key to be valid and the user authentic.

The <Token> in the shared secret XML message is defined as the base 64 encoded string of the result delivered by the application of a message digest algorithm.

The Shared Secret Token Generator utility generates Shared Secret tokens to be used for single sign-on based user authentication. To use this utility, execute the `bin/SharedSecretTokenGenerator.sh` script with the following parameters:

```
bin/SharedSecretTokenGenerator.sh -user <user> -secret <source> [-host
<host>] [-source<source] [-algorithm<algorithm>]
```

The following table describes the values to be specified in this command.

Table 3. SharedSecretTokenGenerator.sh Parameters

Parameter	Description
User	The Emptoris user to authenticate.
Host	The host in the case of a multi-hosted installation of the Emptoris Contract Management server. For a single hosted installation, use " ".
Source	The source associated with the secret. By default, use " ".
Date	When generating a new token, use the current date.
Algorithm	The message digest algorithm (for example, MD5).
Secret	The shared secret.

Note: For a default Emptoris Contract Management server installation, the host, source, and algorithm parameters are optional.

The following example illustrates the shared secret token generation algorithm used by Emptoris Contract Management authentication.

Example: Shared Secret Credential

```
public class SharedSecretAlgorithm {

public class SharedSecretAlgorithm {

static public DateFormat SHARED_SECRET_DATE_FORMAT =

new SimpleDateFormat("MMMMMMMM dd, yyyy hh:mm:ss aa");

public String generateToken(

String user, String host, String source, Date date, String algorithm,
String secret) {

MessageDigest md = MessageDigest.getInstance(algorithm);

md.reset();

md.update(user);

md.update( (byte)':');

md.update(host);

md.update( (byte)':');

md.update(source);

md.update( (byte)':');

String dateStr = SHARED_SECRET_DATE_FORMAT.format(date);

md.update( dateStr.getBytes() );

md.update( (byte)':');

md.update( cred.getDigestAlgorithm().getBytes() );

md.update( (byte)':');

md.update( sharedSecret.getBytes() );

byte[] digestBytes = md.digest();

String token = new String( Base64.encode( digestBytes ) );

return token; }

}
```

Password credentials are less desirable than the shared secret since they take the form shown in the example below where <Password> contains the actual password of the user specified within the <Sender> tag.

Example: Password Credential

```
<Credential type="Password">

<Password>password<Password>

/Credential>
```

Integration Principles

The actions specified in the `verb=action` attribute are used for data management operations or back-end integration, and for seamless invocations from the external application UI or front-end integration.

Back-end and Front-end Integration Principles

Common scenarios usually require the following two types of integration support:

- **Back-end integration**.Occurs when an external application performs a data management operation in Emptoris Contract Management, without invoking the UI. An example back-end operation is to create a new contract based on a template and populate Terms with values.

  IBM Emptoris Web Services support the following back-end operations: *Create*, *Delete*, *Search*, *Get*, and *Update*.
- **Front-end integration**.Occurs when an external application's UI requires the invocation of certain elements of the Emptoris Contract Management UI (such as Contract Editor window pages). The user is not required to log in separately to Emptoris Contract Management. Instead, a single sign-on mechanism is used. Thus, in front-end integration, a user of an external application is able to work with Emptoris Contract Management UI features as if they were an integral part of the external application.

  IBM Emptoris Web Services support the following front-end action: *getURL*. This returns a hyperlink that is embedded as an element within the external application's UI or that is immediately opened by the external application.

  A getURL web service is also provided to allow access to the Emptoris Contract Management Home Page.

Data Management Principles

The Emptoris Contract Management system handles sensitive, transactional data where historical integrity and security are paramount. Changing or deleting data can have an impact on negotiations and renewal generation. Complex business logic controls and restricts the updates allowed to data that is referenced within contract objects. This logic often requires significant user interaction to validate the safety of any required modifications.

The web services therefore support a rich set of data population and retrieval actions but a comparatively conservative set of modification capabilities. Data can be easily added and retrieved, but updates and deletes are restricted. Updates and deletes are best performed within the Emptoris Contract Management application under interactive user control. Updates and deletes using IBM Emptoris Web Services are limited to only certain master objects.

System Configuration for Integrating Web Services

The ecm.properties file contains properties that you can use to configure the interpretation of web service request messages:

diCarta.SharedSecret<SOURCE>=<SHARED SECRET>

<SOURCE>	Specify the name of the external system with which you want to share the specified shared secret. To use a shared secret with an unspecified source, use the value DEFAULT.
<SHARED SECRET>	Shared secret for this source. Only one shared secret may be defined for each source.

You can supply multiple instances of this property in the ecm.properties file in order to define multiple source=shared secret pairs.

diCarta.xml.validateMode determines whether the system will validate XML tags:

=true	In the case of an invalid tag, the system will throw an exception and abort the operation. This is the default.
=false	In the case of an invalid tag, the system will issue a warning message only.

diCarta.xml.caseInSensitiveMode determines whether the system will ignore the case of XML tag names:

=true	The system will treat tag names as case-insensitive. This is the default.
=false	The system will require the case of each character in each tag name to match exactly the case shown in the XSD.

Sample: Web Services

The examples provided in this section illustrate how to use organization-related web services in order to get a contract, select an organization and retrieve or update organization information. Sample request and response messages are provided for each web service.

Get a Contract

You can use the get Web service to get the complete contents of a contract from Emptoris Contract Management when given a Contract ID. The example below is a sample request message for this Web service.

Example: getContract Request Message

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"

xmlns:con="http://www.dicarta.com/contracts/services/
contract"xmlns:auth="http://www.dicarta.com/contracts/types/auth"

xmlns:con1="http://www.dicarta.com/contracts/types/domain/contract">

<soapenv:Header/>

<soapenv:Body>

<con:get-request>

<con:authentication>
```

```
<auth:user>kmg</auth:user>

<auth:credential><auth:password>contracts</auth:password></auth:credential>

</con:authentication>

<con:request-data>

<con1:id>39cccff67d888</con1:id>

</con:request-data>

</con:get-request>

</soapenv:Body>

</soapenv:Envelope>
```

Retrieve an Organization

You can use the get Web service to retrieve an organization from Emptoris Contract Management. The example below is a sample request message for this Web service.

Example: getOrganization Request Message

```
<SOAP-ENV:Envelope

xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"

xmlns:xsd="http://www.w3.org/1999/XMLSchema">

<SOAP-ENV:Body>

<organization:get-request

xmlns:ns1="http://www.mycompany.com/contracts/types/auth"

xmlns:organization="http://www.mycompany.com/contracts/services/
organization/organization"

xmlns:domain-organization="http://www.mycompany.com/contracts/types/
domain/organization"

xmlns:domain-common="http://www.mycompany.com/contracts/types/
domain/common"

xsi:schemaLocation="rsrc/webservices/services/organization/
organization/organization-services-types.xsd">

<organization:authentication>

<ns1:user>dicarta-user</ns1:user>

<ns1:credential>
```

```
<ns1:shared-secret>

<ns1:source> </ns1:source>

<ns1:date>2005-06-

15T00:00:00</ns1:date>

<ns1:algorithm>MD5</ns1:algorithm>

<ns1:token>XaQyXkmsVpfcbNa0HyNrRQ==</ns1:token>

</ns1:shared-secret>

</ns1:credential>

</organization:authentication>

<organization:request-data>1c0fda4433144222bfaeb9a7992ecf12</
organization:request-data>

</organization:get-request>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>
```

The example below shows the expected response message for the *get* request.

Example: getOrganization Response Message

```
<?xml version="1.0" encoding="utf-8"?>

<soapenv:Envelope

xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"

xmlns:xsd="http://www.w3.org/1999/XMLSchema"

xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance">

<soapenv:Header>

<ns1:sessionID

soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"

soapenv:mustUnderstand="0"

xmlns:ns1="http://xml.apache.org/axis/session">6941874468357592732</
ns1:sessionID>

</soapenv:Header>

<soapenv:Body>
```

```
<get-response xmlns="http://www.dicarta.com/contracts/services/
organization/organization">

<response-data>

<ns1:id xmlns:ns1="http://www.dicarta.com/contracts/types/domain/
organization">cbb595f8815d4a5b9239376c522139a0</ns1:id>

<ns2:number xsi:null="true" xmlns:ns2="http://www.dicarta.com/contracts/
types/domain/organization"/>

<ns3:email xsi:null="true" xmlns:ns3="http://www.dicarta.com/contracts/
types/domain/organization"/>

<ns4:phone-number xsi:null="true" xmlns:ns4="http://www.dicarta.com/
contracts/types/domain/organization"/>

<ns5:is-legal-entity xmlns:ns5="http://www.dicarta.com/contracts/types/
domain/organization">false</ns5:is-legal-entity>

<ns6:notes xsi:null="true" xmlns:ns6="http://www.dicarta.com/contracts/
types/domain/organization"/>

<ns7:external-system-info xmlns:ns7="http://www.dicarta.com/contracts/
types/domain/organization"/>

<ns8:expiration-data xmlns:ns8="http://www.dicarta.com/contracts/types/
domain/organization">

<ns9:start-date xmlns:ns9="http://www.dicarta.com/contracts/types/domain/
common">2004-06-22</ns9:start-date>

<ns10:start-time xmlns:ns10="http://www.dicarta.com/contracts/types/domain/
common">18:38:11.000Z</ns10:start-time>

<ns11:end-date xsi:null="true" xmlns:ns11="http://www.dicarta.com/
contracts/types/domain/common"/></ns8:expiration-data>

<ns12:audit-info xmlns:ns12="http://www.dicarta.com/contracts/types/domain/
organization">

<ns13:created-by xmlns:ns13="http://www.dicarta.com/contracts/types/domain/
common">UserNotKnown</ns13:created-by>

<ns14:created-on xmlns:ns14="http://www.dicarta.com/contracts/types/domain/
common">2004-06-22T17:38:11.000Z</ns14:created-on>

<ns15:modified-by xsi:null="true" xmlns:ns15="http://www.dicarta.com/
contracts/types/domain/common"/>

<ns16:modified-on xsi:null="true" xmlns:ns16="http://www.dicarta.com/
contracts/types/domain/common"/></ns12:audit-info>

<ns17:time-zone xmlns:ns17="http://www.dicarta.com/contracts/types/domain/
organization">PST</ns17:time-zone>
```

```
<ns18:parent-party-id xmlns:ns18="http://www.dicarta.com/contracts/types/
domain/organization">b39e6d737b994febb7946d3f45261596</ns18:parent-party-
id>

<ns19:name xmlns:ns19="http://www.dicarta.com/contracts/types/domain/
organization">Engineering</ns19:name>

<ns20:type xsi:null="true" xmlns:ns20="http://www.dicarta.com/contracts/
types/domain/organization"/>

<ns21:organization-level xsi:null="true" xmlns:ns21="http://
www.dicarta.com/contracts/types/domain/organization"/>

<ns22:currency xmlns:ns22="http://www.dicarta.com/contracts/types/domain/
organization">US Dollar</ns22:currency>

<ns23:duns-number xsi:null="true" xmlns:ns23="http://www.dicarta.com/
contracts/types/domain/organization"/>

<ns24:website-url xsi:null="true" xmlns:ns24="http://www.dicarta.com/
contracts/types/domain/organization"/>

<ns25:federal-tax-id xsi:null="true" xmlns:ns25="http://www.dicarta.com/
contracts/types/domain/organization"/>

<ns26:naics-code xsi:null="true" xmlns:ns26="http://www.dicarta.com/
contracts/types/domain/organization"/>

<ns27:dashboard-flag xmlns:ns27="http://www.dicarta.com/contracts/types/
domain/organization">false</ns27:dashboard-flag>

<ns28:dashboard-folder xsi:null="true" xmlns:ns28="http://www.dicarta.com/
contracts/types/domain/organization"/>

<ns29:dashboard-usergroup-id xsi:null="true" xmlns:ns29="http://
www.dicarta.com/contracts/types/domain/organization"/>

<ns30:phones xmlns:ns30="http://www.dicarta.com/contracts/types/domain/
organization"/>

<ns31:addresses xmlns:ns31="http://www.dicarta.com/contracts/types/domain/
organization"/>

<ns32:aliases xmlns:ns32="http://www.dicarta.com/contracts/types/domain/
organization">

<ns33:alias xmlns:ns33="http://www.dicarta.com/contracts/types/domain/
party">

<ns33:id>63bfe0c6be374eeb9b8e0060c1bdf6bf</ns33:id>

<ns33:name>Engineering</ns33:name></ns33:alias></ns32:aliases>

<ns34:contacts xmlns:ns34="http://www.dicarta.com/contracts/types/domain/
organization">
```

```
<ns34:contact>

<ns34:contact-id>35f3a42e4896477d81cba281111af2ca</ns34:contact-id>

<ns34:individual-id>ada2afdb5e014ef48894532921448fe6</ns34:individual-id>

<ns34:name>WebServices_Individual_126 WebServices_Individual_126</
ns34:name></ns34:contact>

<ns34:contact><ns34:contact-id>c043d4afd489445999106912a3edbeec</
ns34:contact-id>

<ns34:individual-id>99007622434046f8aa21cfc776fb7a82</ns34:individual-id>

<ns34:name>WebServices_Individual_116 WebServices_Individual_116</
ns34:name></ns34:contact>

<ns34:contact>

<ns34:contact-id>be730c13d3b149d68710dd882e9c0118</ns34:contact-id>

<ns34:individual-id>f98af9cb53eb408cbba33a373edae7ce</ns34:individual-id>

<ns34:name>WebServices_Individual_115 WebServices_Individual_115</
ns34:name></ns34:contact>

<ns34:contact>

<ns34:contact-id>a83446e697284145b08e2af803c6c927</ns34:contact-id>

<ns34:individualid> 27d88acb9ec54154a3d2ccddc90a0747</ns34:individual-
id><ns34:name>SystemAdmin</ns34:name></ns34:contact></ns34:contacts>

<ns35:relationships xmlns:ns35="http://www.dicarta.com/contracts/types/
domain/organization"/>

<ns36:custom-properties xmlns:ns36="http://www.dicarta.com/contracts/types/
domain/organization"/>

</response-data>

</get-response>

</soapenv:Body>

</soapenv:Envelope>
```

Update Properties of an Organization

You can use the updateProperties Web service to update or modify the properties of the organization object. The example below is a sample request message for this Web service.

Example: updateOrganizationProperties Request Message

```
<SOAP-ENV:Envelope
```

```
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"

xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"

xmlns:xsd="http://www.w3.org/1999/XMLSchema">

<SOAP-ENV:Body>

<update-properties-request

xmlns="http://www.dicarta.com/contracts/services/organization/organization"

xmlns:ns1="http://www.dicarta.com/contracts/types/auth"

xmlns:domain-common="http://www.dicarta.com/contracts/types/domain/common"

xsi:schemaLocation="rsrc/webservices/services/organization/organization/
organization-services-types.xsd">

<authentication>

<ns1:user>dicarta-user</ns1:user>

<ns1:credential>

<ns1:shared-secret>

 <ns1:source> </ns1:source>

<ns1:date>2005-06-15T00:00:00</ns1:date>

<ns1:algorithm>MD5</ns1:algorithm>

<ns1:token>XaQyXkmsVpfcbNa0HyNrRQ==</ns1:token>

</ns1:shared-secret>

</ns1:credential>

</authentication>

<request-data>

<id>cbb595f8815d4a5b9239376c522139a0</id>

<name>Engineering</name>

<!-- diCarta -->

<parent-id>b39e6d737b994febb7946d3f45261596</parent-id>

<!-- qa_intorg_1

<parent-id>7f26ff779cf5491b9b57221621eab617</parent-id>-->

<!-- ExtComp1
```

```
<parent-id>98d2fddb571241938bd3ad90844962b2</parent-id> -->

<number>9</number>

<organization-level>LevelGold</organization-level>

<type>Project</type>

<!-- External System Info -->

</request-data>

</update-properties-request>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>
```

The example below shows the expected response message for the updateProperties request.

Example: updateOrganizationProperties Request Message

```
<?xml version="1.0" encoding="utf-8"?>

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"

xmlns:xsd="http://www.w3.org/1999/XMLSchema"

xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance">

<soapenv:Header>

<ns1:sessionID

soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"

soapenv:mustUnderstand="0"

xmlns:ns1="http://xml.apache.org/axis/session">-143886908318900494</
ns1:sessionID>

</soapenv:Header>

<soapenv:Body/>

</soapenv:Envelope>
```

Select an Organization

You can use the orgSelector Web service to select an organization from Emptoris Contract Management. The example below is a sample request message for this Web service.

Example: orgSelector Request Message

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:org="http://www.dicarta.com/contracts/services/organization"
xmlns:auth="http://www.dicarta.com/contracts/types/auth">

<soapenv:Header/>

<soapenv:Body>

<org:org-selector-request>

<org:authentication>

<auth:user>kmg</auth:user>

<auth:credential><auth:password>contracts</auth:password></auth:credential>
</org:authentication>

<org:request-data>

<org:name>Emptoris</org:name><org:number>1007</org:number><org:is-
external>true</org:is-external><org:organization-id>563456tyi</
org:organization-id><org:external-party-unique-id>343</org:external-party-
unique-id>

</org:request-data>

</org:org-selector-request>

</soapenv:Body>

</soapenv:Envelope>
```

Chapter 3. Appendix: List of Web Services

This chapter provides a list of the IBM Emptoris Web Services Definition Language files (WSDLs) that you can avail to access the Emptoris Contract Management database.

The WSDLs are arranged according to Emptoris Contract Management logical objects and processes. The ISDK contains the XML schema documentation for each Web service. For examples of usage, see Chapter 2, "Integration," on page 7.

The following table lists the Web services supported by Emptoris Contract Management 10.0 in alphabetical order, along with the associated business objects.

Table 4. List of Web Services

Web Service	Operation	Description
Contract	addAddress	Adds an address to a contract.
	addAttachment	Adds an attachment to a contract.
	addAttachmentToLanguage	
	addComment	Adds a comment to a contract.
	addTerm	Adds a Term to a contract.
	addLines	Adds one or more contract lines to a contract. It supports maximum of 750 lines.
	addBulkLines	Adds multiple lines to a contract. It supports maximum of 20,000 lines. You are recommended to use this web service over the 'addLines' web service.
	createAmendment	Creates an amendment contract for an existing executed contract.
	createAuthoredContract	Creates an authored contract, which can optionally include contract lines.
	createAuthoredContractWithLines Attachment	Creates an authored contract with contract lines in integrated or standalone environment.
	createFiledAmendmentContract	Creates a filed amendment in the contract.
	createFiledContract	Creates a filed contract.
	createQuote	Creates a quote contract.
	executeContract	Executes a contract. **Note:** If the clause visibility feature is enabled, and you do not have permissions to view all the clauses in the contract, you cannot execute the contract.
	executeAmendmentContract	Executes an amendment contract.
	executeFiledAmendmentContract	Executes a filed amendment contract.

Table 4. List of Web Services (continued)

Web Service	Operation	Description
	executeFiledContract	Executes a filed contract.
	executeQuoteContract	Executes a quote contract.
	get	Returns the complete contents of a contract when given a Contract ID.
	getAttachment	Requests attachment information using the contract reference and attachment ID as inputs, then returns the complex type attachment including all its corresponding elements.
	getComment	Uses the contract reference and comment reference complex types to return contract comment data.
	getProperties	Returns the core properties of a contract given its ID and revision. The core properties include only the simple data type attributes and excludes any fields that are links to other objects, such as clauses, Terms, attachments, etc.
	getTerm	Gets a Term and returns it for use within a contract.
	removeAddress	Removes an address from a contract.
	removeAttachment	Removes an attachment from a contract.
	removeTerm	Removes a Term from a contract.
	search	Returns a list of references of contracts that match the given search criteria.
	setExternalParty	Sets a contract's external party. The external party may be an external organization or an individual. If it is an organization, you can set the contract's external contact by passing an individual ID. The individual ID must represent a contact of the same external party organization. If it is an individual, the external contact uses the same individual ID as the external party.
	setInternalParty	Sets the internal organization and internal contact or just the internal contact. The internal contact must be a member of the contracting organization.
	removeParties	Removes internal or external parties from a contract.
	updateAttachment	Replaces an attachment associated with a contract.

Table 4. List of Web Services (continued)

Web Service	Operation	Description
	updateCustomProperty	Updates a custom property of a contract.
	updateProperties	Uses the contract ID to locate a contract and update one or many parameters on the contract.
	updateSubstatus	Uses the contract ID to update the contract substatus (system value list), notes, and notes category fields on the contract object.
	updateTerm	Updates a non-system Term within a contract.
	updateTerms	Updates multiple non-system Terms within a contract.
	withdrawContract	Withdraws an existing contract.
	deleteContracts	Deletes the specified contracts.
	getLanguage	Reads the language of a contract. **Note:** If the clause visibility feature is enabled, depending on the permissions assigned to your user account, you may or may not be able to view certain clauses within the contract.
	getRelationship	Retrieves the relationship details for a contract.
	setContractsRelationship	Adds multiple relationships to a contract object.
	searchAttachment	Retrieves the contract attachments of the latest contract revision.
	updatePrimaryContact	Updates the primary contact of an internal or external party of a contract.
	deleteAllLines	Deletes all Lines form a contract.
	getAttachmentFromLanguage	Gets the attachment from Language tab for Filed contract and Filed Amendment contract.
	getContractApprovalActivity	Identifies contract approval activity for a particular revision.
	getLines	Returns Line Definition Properties list and line field data list.
	updateCustomProperties	Updates a custom properties of a contract.
Contract Class	create	Creates a contract class.
	getContractClasses	Uses the request data string to return the contract class information.
Contract Instance	addPermissionUsers	Adds users to permissions at contract level.

Table 4. List of Web Services (continued)

Web Service	Operation	Description
	addUserPermissions	Adds permissions to users at contract level.
	readUserPermissions	Reads the users belonging to permission at a contract level.
	readPermissionUsers	Reads the user's permission at contract level.
	removePermissionUsers	Removes users from permissions at contract level.
	removeUserPermissions	Removes permissions from users at contract level.
Contract Template	search	Uses various searches for contract templates using one or many elements from the contract template.
	searchWithLineDefinitions	Searches for contract templates that contain specified line definitions.
Currency	search	Returns a list of references of currencies that match the search criteria.
Custom Property	get	Returns a custom property definition given a custom property name.
	search	Uses a dummy string as input to return a list of references of custom property definitions that match the search criteria.
	create	Creates a custom property definition.
Interview Instance	createInterviewInstance	Creates an interview instance for a given interview template.
	search	Returns descriptors of interview instance that matches the given search criteria.
	get	Returns the complete contents of an interview instance when given an Interview ID.
Folder	addContracts	Adds contracts to a folder.
	copyContractsToFolder	Copies selected contracts from one folder to another.
	createFolder	Creates a folder
	deleteFolder	Deletes the specified folder.
	getContractsInFolder	Retrieves contracts from a folder.
	getFoldersForContract	Retrieves folders that contain a particular contract.
	moveContractsToFolder	Moves selected contracts from one folder to another.
	removeContracts	Removes contracts from a folder.

Table 4. List of Web Services (continued)

Web Service	Operation	Description
	search	Returns folders that match the search criteria.
Organization: Address	get	Returns an organizational address given the UUID of the organizational address.
	search	Returns a list of references of organizational addresses that match the search criteria.
	update	Set an existing address for an organization as the primary address.
Organization: Individual	addAlias	Updates the properties of the individual object.
	removeAlias	Removes an individual's alias.
	get	Returns an individual given the UUID of the individual.
	getProperties	Returns an individual properties complex type given the UUID of the individual.
	search	Returns a list of individual references that match the search criteria.
Organization: Organization	addAlias	Adds an alias to the organization object.
	get	Returns an organization given the UUID of the organization.
	getProperties	Returns an organization properties complex type given the UUID of the organization.
	orgSelector	Returns the list of organizations.
	removeAlias	Removes an organization alias.
	search	Returns a list of organization references that match the search criteria.
	updateCustomProperty	Updates the custom properties of the organization object.
	updateCustomProperties	Updates the custom properties of the organization object.
	updateProperties	Updates the properties of the organization object.
Price List	addEntry	Adds an entry to the price list object.
	create	Creates a price list.
	get	Returns a price list given the UUID of the price list.
	search	Returns a list of price list references that match the search criteria.

Table 4. List of Web Services (continued)

Web Service	Operation	Description
Product	create	Creates a product.
	get	Returns a product given the UUID of the product.
	search	Returns a list of product references that match the search criteria.
	updateProperties	Updates the properties of a product.
Relationship Type	get	Returns a relationship type given the UUID of the relationship type.
	search	Returns a list of relationship type ' that match the search criteria.
Security: Permission	search	Reads the individual permission id provided individual permission name.
Security: Permission Group	addPermisssions	Adds permissions to the permission group object.
	create	Creates a permission group.
	get	Returns a permission group given the UUID of the permission group.
	removePermissions	Removes permissions from a permission group object.
	search	Returns a list of permission group references that match the search criteria.
Security: User	get	Returns a user given the UUID of the user.
Security: User Group	get	Returns a user group given the UUID of the user group.
	search	Returns a list of user group references that match the search criteria.
Single Sign-on	getURL	Returns a URL given certain parameters.
Task	get	Returns a task given the task's UUID.
	reassignTask	Reassigns the task to specific user.
	search	Returns a list of task references that match the search criteria.
	searchDetail	Returns all the tasks for a user.
	searchDetailbyUserId	Returns all the tasks for a user given the User ID.
Term Definition	create	Creates a Term definition.
	get	Returns the term definition given the UUID of a term definition name.

Table 4. List of Web Services (continued)

Web Service	Operation	Description
	search	Returns a list of Term definitions that match the search criteria.
	addLovs	Adds the list of values to the Term definition.
	getLovs	Returns the list of values for the specified Term definitions.
	deleteLovs	Deletes the list of values from the Term definition.
	getByLabel	Returns a list of Term definition that match the search criteria.
Value List	addValue	Adds a values to the list.
	get	Returns a list of values.
	getTypes	Returns a list of value list types.
	updateProperties	Updates the properties.
	updateValues	Updates the values to the list.

Notices

This information was developed for products and services offered on a worldwide basis.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing

IBM Corporation

North Castle Drive

Armonk, NY 10504-1785

U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing

Legal and Intellectual Property Law

IBM Japan Ltd.

1623-14, Shimotsuruma, Yamato-shi

Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation

Somers, NY

U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml.

Emptoris, EMPTORIS® and device, RIVERMINE®, and XCITEC® and device are trademarks or registered trademarks of IBM Corporation.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Privacy Policy Considerations

IBM Software products, including software as a service solution, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes.

Depending upon the configurations deployed, this Software Offering may use session and persistent cookies that collect each user's name, profile name for purposes of session management, authentication, enhanced user usability, single sign-on configuration or other usage tracking or functional purposes. These cookies can be disabled, but disabling them will also eliminate the functionality they enable.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, See IBM's Privacy Policy at http://www.ibm.com/privacy and IBM's Online Privacy Statement at http://www.ibm.com/privacy/ details the section entitled "Cookies, Web Beacons and Other Technologies" and the "IBM Software Products and Software-as-a-Service Privacy Statement" at http://www.ibm.com/software/info/product-privacy.

Readers' Comments — We'd Like to Hear from You

IBM Emptoris Contract Management
Web Services Integration Guide
Version 10.0.2.5

Publication No. SC27-5348-03

We appreciate your comments about this publication. Please comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. The comments you send should pertain to only the information in this manual or product and the way in which the information is presented.

For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you. IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you state on this form.

Comments:

Thank you for your support.

Send your comments to the address on the reverse side of this form.

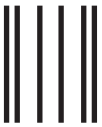
If you would like a response from IBM, please fill in the following information:

Name	Address
Company or Organization	
Phone No.	Email address

Readers' Comments — We'd Like to Hear from You
SC27-5348-03

Cut or Fold
Along Line

Fold and Tape | **Please do not staple** | Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM
ATTN: Dept 542
3905 37th Street NW
Rochester, MN
USA 55901-6666

Fold and Tape | **Please do not staple** | Fold and Tape

SC27-5348-03

Cut or Fold
Along Line

Printed in USA

SC27-5348-03