

Westcon Services Implementation Guide

Order Change

Version 2.0

Document Revision 1.0.4

September 8, 2016

Document Revision History			
Version	Date	Author	Description
1.0		Thomas Young	Draft
1.0.1	2016-06-24	Brad Ross	Format updates, revision history, branding, support team
1.0.2	2016-07-18	Brad Ross	Updated for Reserved Stock feature - using the availability API for special warehouse locations of reserved stock
1.0.3	2016-07-21	Brad Ross	Added IT Support email address
1.0.4	2016-09-08	Chakravarthy Sayani	Updated WCF End Points



Table of Contents

1. Overview	4
2. Getting Started	5
2.1 Working with Westcon Group	5
2.2 Accessing Westcon Group Services	5
3. OrderChange Service	6
3.1 Request Schema	7
3.2 Response Schema	7
4. Consuming the Service	7
4.1 Setting up .NET Client	8
4.1.1 Add a Service Reference	8
4.1.2 Configure bindings	8
4.1.3 Invoke service method.....	8
4.2 Setting up SoapUI client.....	11
4.2.1 Adding WSDL.....	11
4.2.2 Supplying Credentials	11
5. SOAP Failures and Exceptions.....	14
6. Reference Tables	14
6.1 Info Table (Sample).....	14

1. Overview

Westcon Services are a suite of API's that serve as a direct B2B electronic link between Westcon and its partners, complementing existing website portals and other B2C links. Westcon Services offer a variety of services, including Pricing, Availability, Order Placement (Purchase Order), Order Status, Order Change and Return Merchandise Authorisation (RMA).

Order Change service allows to change the Shipping Address and Shipping Condition of an Order before the Shipment is processed.

With Westcon Services, partners can seamlessly integrate information such as product pricing, stock availability obtained from Westcon in real-time – into both internal business and customer-facing services such as a shopping cart.

2. Getting Started

Welcome to Westcon Group partner integration! We will assist you along each step and answer questions throughout the process. We understand that each integration project is unique and you will be treated with the utmost attention and expediency.

2.1 Working with Westcon Group

Westcon will assemble a team to manage and implement your integration project.

The Application Integrations Team is available Monday – Friday, from 9:00 am EST to 5:00 pm EST.

For questions or guidance on your implementation, kindly reach out to Brad Ross, (914) 829-7631.

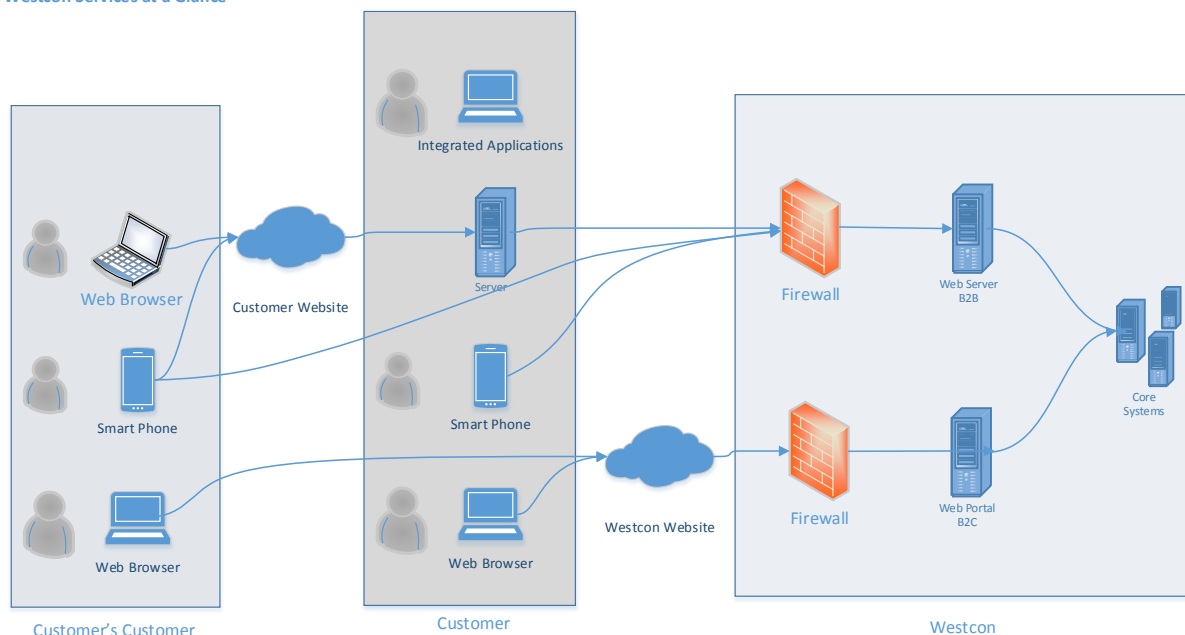
To contact our Support email address –

ITS_B2BSupport@westcon.com

2.2 Accessing Westcon Group Services

Westcon Services may be accessed via http/https, typically from a customer web server; it could be from anywhere including mobile devices via a customer-provided proxy service.


Westcon Services at a Glance



As a WEB service, the concrete WSDL is provided to the partner. The request and response structure of these operations are described in the associated sections. To address the security requirements, the services, IP Security is implemented and the traffic is over a secure HTTPS transport. Furthermore, there is a Username-Password authentication at message level.

3. OrderChange Service

Service	Operations	Description
OrderChange	submitOrderChange	Allow changes to ShipToAddress and ShippingCondition of an Order prior to the Shipment process.

Environment	Url
Test	https://webservicetest.westcon.com/B2B/WebServices/v2/OrderChange.svc?wsdl
Production	https://webservices.westcon.com/B2B/WebServices/v2/OrderChange.svc?wsdl
Service definition	 OrderChange.svc

For existing orders placed with Westcon, the following fields can be changed before orders are shipped:

- **Shipping Method:** One of the 2 char code representing shipping method. This is mandatory field.
- Name: Company name
- Street: Street Address
- City: City name
- **State:** This is mandatory. The 2 char representation of U.S State Code.
- **ZIP:** Zip code. This is a mandatory field.
- Telephone
- Fax
- Email

3.1 Request Schema

Element	Required	Repetitions (0+, 1, 1+)	Type/Length	Example	Comments
WestconOrderChange		1		<WestconOrderStatus>	
CustomerAccountNumber	Required	1	string(10)	1234567	Provide your username as it appears in the Subscriber's Details Document. (This value has to match the Username under UsernameToken in the Soap Header section)
OrderNumberType	Required	1	string(1)	W	C for Customer PO Number, W for Westcon PO Number (W is default)
OrderNumber	Required	1	String(40)	0002342342	Order Number for which the changes are being requested. You cannot mix and match Customer and Westcon Order Numbers.
ShippingMethod	Required	1	string(2)	S1	2 char shipping code to change on the order, see shipping condition codes
Name	Optional	1	String(35)	Cerium Networks	Company Name
Street	Optional	1	string(35)	520 White Plains	Street Address, or Line1 of address
City	Optional	1	string(35)	W Palm Beach	
State	Required	1	string(3)	FL	Mandatory. state (please use 2 character abbreviation)
ZIP	Required	1	String(9)	32212	postal code (with 5-4 or 5 character format)
Telephone	Optional	1	string(25)		1234232424
Fax	Optional	1	string (25)		1234242234
Email	Optional	1	string (64)		abc@def.com
WestconOrderChange		1			<WestconOrderStatus>

3.2 Response Schema

Element	Repetitions (0+, 1, 1+)	Type/Length	Example	Details
WestconOrderChange	1			
CustomerAccountNumber	1	string(10)	20001502	Your customer account number returned
Errors	0+			This block will be populated only in case of system errors
ErrorNo	1	1		Error Number
ErrorDetail	1	text	System Error. Please try again	Error Details
Errors	0+			
Orders	1			
Order	1+			
OrderChangeResponse	1	String	000235147 Sales Order was updated successfully	Message when order is successfully updated
Order				
Orders				
WestconOrderStatus				

4. Consuming the Service

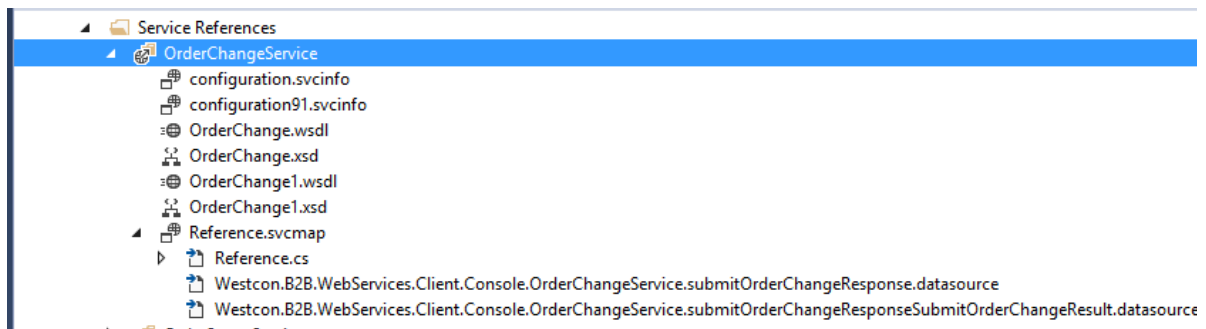
This section provides guidance to setup client and consume the service using Visual Studio for .NET clients and SoapUI to quickly test by providing XML inputs.

4.1 Setting up .NET Client

The following are the main steps to consume the service in .NET. Sample code is provided for guidance.

4.1.1 Add a Service Reference

Create Service Reference using the wsdl. For this sample, the name `AvailabilityService` is used for the Service Reference.



4.1.2 Configure bindings

Ensure the bindings are configured as below.

```
<system.serviceModel>
  <bindings>
    <wsHttpBinding>
      <binding name="OrderChangeWebServiceBinding">
        <security mode="TransportWithMessageCredential" >
          <transport clientCredentialType='None' />
          <message clientCredentialType='UserName' establishSecurityContext='false' />
        </security>
      </binding>
    </wsHttpBinding>
  </bindings>
  <client>
    <endpoint
      name="wsHttpAvailabilityWebService"
      address="https://webservicetest.westcon.com/B2B/WebServices/v2/OrderChange.svc"
      binding="wsHttpBinding"
      bindingConfiguration="OrderChangeWebServiceBinding"
      contract="OrderChangeService.WestconWebServicesOrderChangeWebService"/>
  </client>
</system.serviceModel>
```

4.1.3 Invoke service method

The following is a sample code to invoke the service. Comments within the code provide some details for quick reference. The following are the main steps to invoke.

Create the service proxy

The reference classes are in OrderChangeService Service Reference, you should have it referred in using statement

```
using OrderChangeService;
..
..
```



```
var serviceProxy = new WestconWebServicesOrderChangeWebServiceClient
{
    ClientCredentials =
    {
        //Replace UserName and Password values with those provided to you
        UserName =
        {
            UserName = "1XX1913",
            Password = "nXD4j7cdb68vPNHuMQLGCanMQXjqxXBJ1TRp"
        },
    },
};
```

Create the request

```
var request = new submitOrderChangeRequest
{
    WestconOrderChange = new submitOrderChangeRequestWestconOrderChange
    {
        CustomerAccountNumber = "1231963",
        OrderNumber = "550976",
        OrderNumberType = "W",
        ShippingMethod = "UF",
        Name = "Company Name",
        Street = "Changed Street Name",
        City = "Changed City Name",
        State = "NY",
        Email = "abc@xyz.com",
        ZIP = "10482",
        Fax = string.Empty,
        Telephone = string.Empty,
        WestconMetadata = new
        submitOrderChangeRequestWestconOrderChangeWestconMetadata
        {
            Culture = string.Empty,
            OperationType = string.Empty,
            SourceSystem = "B2BWEB",
            Timestamp = DateTime.Now,
            TransactionID = string.Empty
        }
    }
};
```

Invoke the service

```
//var authHeader = new AuthHeader {userName = "1XX1913", passWord =
"nXD4j7cdb68vPNHuMQLGCanMQXjqxXBJ1TRp"};

try
{
    //authHeader can be null... it is only there for backward compatibility. The
    credentials are passed
    var response = serviceProxy.submitOrderChange(null, request);
}
catch (Exception e)
{
    HandleException(e);
}
```

Sample code, all put together

```
using System;
using Westcon.B2B.WebServices.Client.Console.OrderChangeService;

namespace Westcon.B2B.WebServices.Client
{
    class Program
    {
        static void Main()
        {
            var serviceProxy = new WestconWebServicesOrderChangeWebServiceClient
            {
                ClientCredentials =
                {
                    //Replace UserName and Password values with those provided to you
                    UserName =
                    {
                        UserName = "1XX1913",
                        Password = "nXD4j7cdb68vPNHuMQLGCanMQXjqxXBj1TRp"
                    }
                },
            };

            var request = new submitOrderChangeRequest
            {
                WestconOrderChange = new submitOrderChangeRequestWestconOrderChange
                {
                    CustomerAccountNumber = "1001963",
                    OrderNumber = "550976",
                    OrderNumberType = "W",
                    ShippingMethod = "UF",
                    Name = "Company Name",
                    Street = "Changed Street Name",
                    City = "Changed City Name",
                    State = "NY",
                    Email = "abc@xyz.com",
                    ZIP = "10482",
                    Fax = string.Empty,
                    Telephone = string.Empty,
                    WestconMetadata = new
submitOrderChangeRequestWestconOrderChangeWestconMetadata
                {
                    Culture = string.Empty,
                    OperationType = string.Empty,
                    SourceSystem = "B2BWEB",
                    TimeStamp = DateTime.Now,
                    TransactionID = string.Empty
                }
            };

            //var authHeader = new AuthHeader { userName = "1XX1913",
            //                                passWord = "nXD4j7cdb68vPNHuMQLGCanMQXjqxXBj1TRp"};

            try
            {
                //authHeader can be null...
                //it is only there for backward compatibility.
                var response = serviceProxy.submitOrderChange (null, request);
            }
            catch (Exception e)
```

```

        {
            HandleException(e);
        }
    }

    private static void HandleException(Exception exception)
    {
    }
}

```

4.2 Setting up SoapUI client

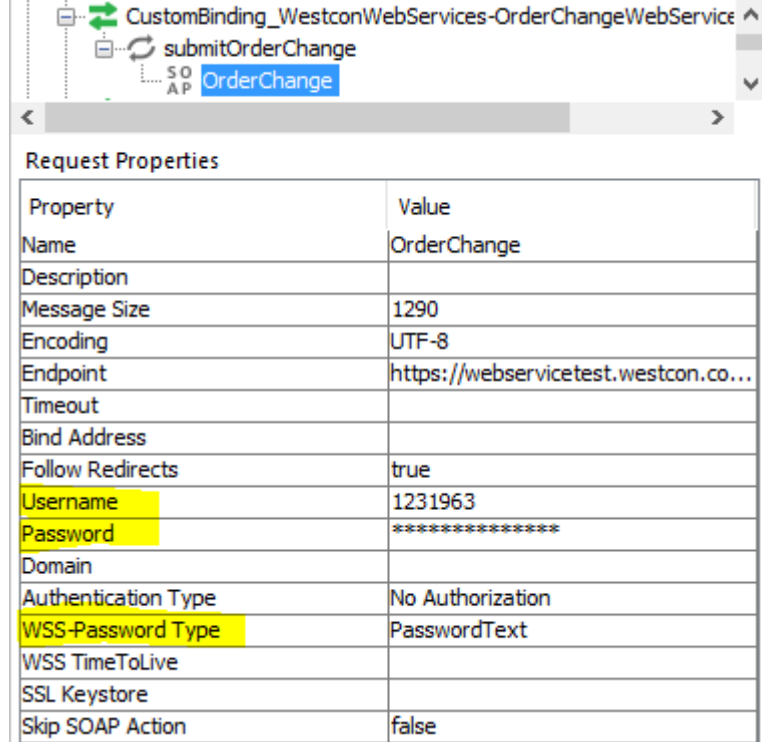
Use SoapUI tool to perform the initial tests quickly. SoapUI allows creating the client quickly and providing the XML input templates to quickly generate a client and test.

4.2.1 Adding WSDL

4.2.2 Supplying Credentials

In order to pass the credentials, the following 3 Request properties must be set as shown below.

Username and **Password** are provided to you separately, which are 2 of the 3 properties. For the 3rd property, **WSS-PasswordType**, select *PasswordText* from the drop down against this property.



Property	Value
Name	OrderChange
Description	
Message Size	1290
Encoding	UTF-8
Endpoint	https://webservicetest.westcon.co...
Timeout	
Bind Address	
Follow Redirects	true
Username	1231963
Password	*****
Domain	
Authentication Type	No Authorization
WSS-Password Type	PasswordText
WSS TimeToLive	
SSL Keystore	
Skip SOAP Action	false

The following is a working sample to invoke *submitOrderChange*.

*<wes:AuthHeader> section under <soap:Header> is there for backward compatibility. It can be ignored.

Request

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:wes="Westcon.B2B.WebServices.Wcf" xmlns:web="http://webservices.westcon.com">
  <soap:Header>
  </soap:Header>
  <soap:Body>
    <web:submitOrderChange>
      <web:request>
        <WestconOrderChange>
          <WestconMetadata>
            <SourceSystem>B2BWEB</SourceSystem>
            <TimeStamp>2016-09-06T09:43:29</TimeStamp>
            <OperationType>OrderChange</OperationType>
            <Culture>EN</Culture>
            <TransactionID/>
          </WestconMetadata>
          <OrderNumberType>W</OrderNumberType>
          <CustomerAccountNumber>1231963</CustomerAccountNumber>
          <OrderNumber>0000550976</OrderNumber>
          <ShippingMethod>UF</ShippingMethod>
          <Name>Lasalle</Name>
          <Street>520 Whiteplains Rd</Street>
          <City>Belleville</City>
          <State>IL</State>
          <ZIP>62226-6114</ZIP>
          <Telephone></Telephone>
          <Fax></Fax>
          <Email>abc@test.com</Email>
        </WestconOrderChange>
      </web:request>
    </web:submitOrderChange>
  </soap:Body>
</soap:Envelope>
```

Response

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd">
  <s:Header>
    <a:Action s:mustUnderstand="1">http://webservices.westcon.com/WestconWebServices-
OrderChangeWebService/submitOrderChangeResponse</a:Action>
    <o:Security s:mustUnderstand="1" xmlns:o="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd">
      <u:Timestamp u:Id="_0">
        <u:Created>2016-09-06T19:04:53.797Z</u:Created>
        <u:Expires>2016-09-06T19:09:53.797Z</u:Expires>
      </u:Timestamp>
    </o:Security>
  </s:Header>
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <submitOrderChangeResponse xmlns="http://webservices.westcon.com">
      <submitOrderChangeResult>
        <WestconMetadata xmlns="">
          <SourceSystem>B2BWEB</SourceSystem>
          <TimeStamp>2016-10-06T09:43:29</TimeStamp>
```

```
<OperationType>OrderChange</OperationType>
<Culture>EN</Culture>
<TransactionID/>
</WestconMetadata>
<Orders xmlns="">
  <OrderChangeResponse>0000550976 Sales Order was updated
successfully</OrderChangeResponse>
</Orders>
</submitOrderChangeResult>
</submitOrderChangeResponse>
</s:Body>
</s:Envelope>
```

5. SOAP Failures and Exceptions

We will respond back to the subscriber of the Web Services with appropriate messages. These messages are a good way to allow the subscriber to act accordingly when these errors are encountered. Most of the time, these exceptions will be thrown in the SAP Api layer and be forwarded to the subscriber by the Proxy. The subscriber should react based on the exception caught.

Error Description	Fault Location	Reason
Service is down	Proxy Service	The service is currently down for maintenance.
Unable to update Sales Order 0000550976 No jurisdiction code could be determined	Proxy Service	The address failed the jurisdiction code validation in SAP
Security key account number invalid	Proxy Service	The security key account number that has been supplied is invalid.
Timeout	Proxy Service	Service took too long to respond.

6. Reference Tables

6.1 Info Table (Sample)

This table contains customer specific details required for successful exchange with Westcon services. Customer should have received this during the on-boarding process.

Customer	SAMPLE COMPANY	This is your Company Name
Account: (username)	1011065	This is your Account number as it appears in our backend systems
Authentication Key: (password)	sN8DThqnkZEabGJtxPtcU8lIs3CjgTpDyFj1	This is a 36-character key with which you will be authenticated
Your Westcon Company	00011	This value is specific to your account and should be included with your web service call
RBU/HRBU	119920 (for Westcon) 119951 (for Vodaone) 119992 (for Comstor)	This value is specific to your account and should be included with your web service call
Currency	USD	
Route Code	U04	Used for PO
Shipping Method	1311	Used for PO
Main WWS-URL	https://webservices.westcon.com	