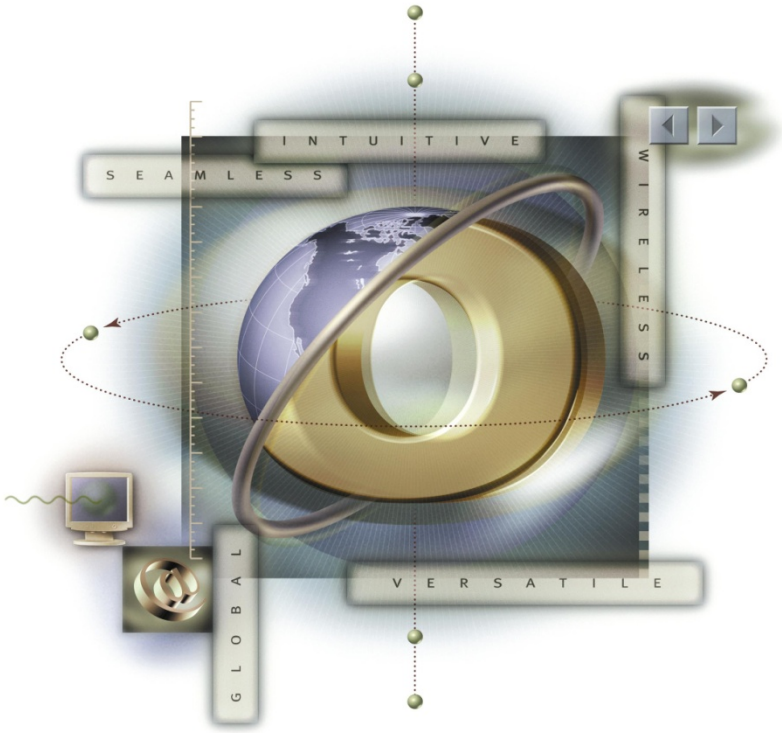
micros® | FIDELIO



Opera Hotel Edition

ORACLE

Opera Electronic Distribution Systems (OEDS)

Product Overview – Opera Web Suite Version 5.1

Document Version 6.7

*Updated **1/2/2013***

MICROS Systems, Inc.
Fidelio Technologies Inc.
2640 Golden Gate Parkway, Suite 211
Naples, FL 34105
Voice. (239) 643-7999 / Fax. (239) 643-7911
Voice. (239) 643-7999 / Fax. (239) 643-7911

Contents

Record of Changes

Date	Version	Description	Author
18 Dec 2004	1.0	Initial Release	C.Dimitrov
30 Jan 2007	2.0	Edits	S. Khomutetsky
18 Oct 2007	3.0	Edits & updated the link to the Micros Web site.	S. Khomutetsky
08 Dec 2007	3.1	Updated functional descriptions.	S. Khomutetsky
04 Mar 2008	3.2	Removed reference to the sample Web site.	S. Khomutetsky
05 November 2008	3.3	Added Meeting Room Web Service table listing all its functions	C. Hunter
05 June 2009	3.4	Added Room Type, Alternate Dates, and Membership search information under Enhanced Availability Features section. Also, updated Availability function description under Availability web service.	C. Hunter
11 June 2009	3.5	157539 – Added UpdateAppUserPassword web function to Description of Functions section.	C. Hunter
16 June 2009	3.6	159201 – TravelAgentLookup function added to Name Service Functions.	C. Hunter
22 June 2009	3.7	159200 – New function named FetchProfile added under Name Service Functions.	C. Hunter
13 Aug 2009	3.8	159443 – Updated ModifyBooking description under Reservation service. CreateBooking functional description also updated.	C. Hunter
02 Oct. 2009	3.9	163147 - New web functions added under Membership: FetchPointsExchange and TransferPoints.	C. Hunter
24 Nov. 2009	4.0	166420 – Added FetchSubscription web function. Activity web service added. V5.0.02.01	C. Hunter

04 Dec. 2009	4.1	165755 – The following web functions have been added to Name service: FetchClaimsStatus, InsertClaim and UpdateClaim. V5.0.02.01	C. Hunter
30 Dec. 2009	4.2	167459 – Added the following web functions to the Membership service: ConsumeECertificate, FavoriteGuest, FetchECertificates and IssueECertificate. Added FetchProfileBenefits web function to the Name service. V5.0.02.01	C. Hunter
10 Feb. 2010	4.3	170798 – QueryAwardsSchedules function added to Information service table. V5.0.02.02	C. Hunter
19 April 2010	4.4	170770 – GuestServices web functions added. V5.0.02.02	C. Hunter
06/18/10	4.5	178568 – New GetScreenItems function added to Information. V5.0.02.03	C. Hunter
07/19/10	4.6	176715 – New QueryProductItems function added to Information service. V5.0.02.03	C. Hunter
08/17/10	4.7	180516 – New QueryPackageItems function added to Information. V5.0.02.03	C. Hunter
08/31/10	4.8	178204 – New web functions added to Activity service: ActivityAvailability, CreateActivity, FetchActivity and CancelActivity. V5.0.02.03	C. Hunter
09/16/10	4.9	181813 – 2 new functions added to Reservation: FetchRoomUpgrades and Upgrade Reservation. V5.0.02.03	C. Hunter
12/16/10	5.0	184444 – AddBenefit and FetchBenefits web functions added to Membership. V5.0.02.04	C. Hunter
12/20/10	5.1	182175—FetchAvailableOffers and UpsellReservation web functions added to Reservation.V5.0.02.04	C. Hunter

01/07/11	5.2	188237 – CreateBlock web function added to Meeting Room service. V5.0.02.04	C. Hunter
03/22/11	5.3	182888 – FetchAvailableECertificates added to Membership. V5.0.02.04	C. Hunter
07/18/11	5.4	194987 – FetchExpectedCharges function added to Availability service. V5.0.03.01	C. Hunter
07/25/11	5.5	185516 – BreakShare and CombineShare web functions added to Reservation service. V5.0.03.01	C. Hunter
11/08/11	5.6	211156 – MeetingRoomCopyBlock function added to MeetingRoom service. V5.0.03.02	C. Hunter
11/09/11	5.7	208456, 211155, 200235 – MeetingRoomFetchBlock and MeetingRoomFetchMyBlocks web functions added to MeetingRoom service. 200235—GuestRequests function added to Reservation service. V5.0.03.02	C. Hunter
11/14/11	5.8	211482—CreateMultipleBookings function added to Reservation service. V5.0.03.02 211475—MeetingRoomCreateRelationship function added to MeetingRoom service.V5.0.03.02 211484—MeetingModifyPackageEvent function added to MeetingRoom service. V5.0.03.02 211478 – MeetingFetchContract function added to MeetingRoom service. MeetingFetchBEO function removed. V5.0.03.02 208527 – ModifyBlock function added to MeetingRoom web service. V5.0.03.02	C. Hunter
03/12/12	5.9	218247 – FetchMemberPoints function added to the Membership service.	C. Hunter

		V5.0.03.03	
04/26/12	6.0	220510—ReInstate Reservation function added to the Reservation service. V5.0.03.03	C. Hunter
05/22/12	6.1	225767—ReIssueMemberCard function added to Membership service. V5.0.03.03	C. Hunter
05/31/12	6.2	225696—Added AssignRoom and ReleaseRoom web functions to Reservation service. V5.0.03.03	C. Hunter
06/11/12	6.3	211501 — External Payment function added to the Reservation Advanced service. V5.0.03.03	C. Hunter
06/20/12	6.4	227414– CancelECertificate function added to Membership service. V5.0.03.03 225614 – MeetingFetchBlockDelegates, MeetingFetchMyRegisteredEvents and MeetingRegisterEventAttendees functions added to MeetingRoom service. V5.0.03.03	C. Hunter
8/20/12	6.5	232686 - Added 2 functions: AddAccompanyGuest to attach the accompanying guest. DeleteAccompanyGuest to detach the accompanying guest. V5.0.02.01	K. Anderson
12/7/12	6.6	228675 - Added new Housekeeping Web Service V5.0.04.00	K. Anderson
12/28/12	6.7	235668 - Added CreateContract, UpdateContract, Delete Contract in Unit Owners Message Specs. V5.0.04.00	K. Anderson

Audience

This document is designed for use by our potential interface partners and clients who wish to develop chain specific User Interfaces. It provides our partners with an overview of Opera Web Suite (OWS) and enables them to assess the knowledge and approximate time-scale that this interface entails.

Overview

This overview is designed to provide a high-level introduction to Opera Web Suites (OWS), including a brief introduction to Web Services technologies, as well as a description of how Web Services can be used to interface with the Central Reservation System® (CRS) on the Opera Reservation System® (ORS) platform.

OWS is designed to support data transfers between client travel applications and the Central Reservation System (CRS). OWS is a collection of Web Services that provide client applications with access to key functionality on the CRS/ORS.

What Are Web Services?

Web services are the fundamental building blocks in the move to distributed computing on the Internet. Open standards, and the focus on communication and collaboration among people and applications have created an environment where Web services are becoming the platform for application integration. Applications are constructed using multiple Web services from various sources that work together regardless of where they reside, or how they were implemented.

Web Services represent an *evolution* in communication between applications, rather than a *revolution*. Most of the technologies behind Web Services are not new. However, these existing technologies have been combined and refined to create a common system of communication. Most importantly, a set of common standards have been assigned to these technologies so that systems using Web Services have a single, explicitly defined method for communicating with each other.

Application Interfaces Before Web Services

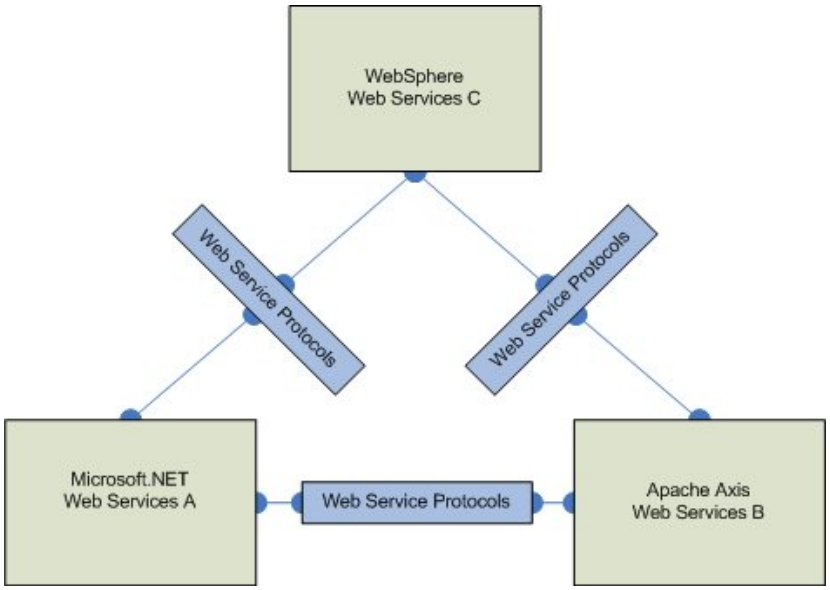
Web Services create greater flexibility because they remove the need for applications to programmatically understand each other. In some cases, pre-existing interfaces can be purchased to translate between disparate languages and platforms. In other cases, however, a proprietary application interface must be specifically designed and deployed to communicate with the other application. Such specialized design can often consume huge amounts of financial and employee resources, as well as add another layer of complication to product design, implementation, and maintenance.

Application Interfaces with Web Services

One of the primary advantages of the Web services architecture is that it allows programs written in different languages on different platforms to communicate with each other in a standardized way. The specific protocols for data transfer are less important than the fact that all of the

services and applications adhere to the same standard. See **Appendix A** for more details about the specific standards, protocols, and formats used in OWS and other Web Services.

In the below scenario, we have Web Services A, Web Services B, and Web Services C that use standard Web Service protocols to transfer information.



The fact that Application A uses Microsoft.NET on a Windows platform as the infrastructure for its Web Service, while Application C uses Web Sphere on UNIX platform for its Web Service and Application B uses Apache Axis on UNIX platform, is irrelevant. All Web Services still communicate outside of their infrastructure using the same protocols.

What is Opera Web Suite?

Opera Web Suite (OWS) uses Web Services technologies to support data transfer between client travel applications and the Opera Reservation System®. OWS is a collection of Web Services that provide access to key functionality on the CRS. The Web Services available through OWS provide access to a wide-variety of travel-related functionality. As OWS evolves, additional CRS business functions will be abstracted and encapsulated into separate Web Services.

A Brief Description of the Functions

OWS currently includes the following services and functions:

Activity Service Functions

The following functions are currently provided through the Activity Web Service:

Function	Description
ActivityLookup	Retrieves a list of activities the guest has reserved.
ActivityAvailability	Retrieves available activities.
CancelActivity	Cancels guest activities.
CreateActivity	Creates guest activities.
FetchActivity	Fetches guest activities.

For the functional specifications, refer to the OWS Activity Message Spec document.

Availability Service Functions

The following functions are currently provided through the Availability Web Service:

Function	Description
Availability	Retrieves lists of available rooms and rates. The user should provide at least a date range, hotel criteria, number of rooms, total guests, and number of children. Rate range or plans, rate tiers, alternate date searches, and search by membership number, specific room type or block criteria are also allowed. Setting the summaryOnly flag to true results in a General Availability request and setting the flag to false results in a Detail Availability request.
FetchAvailableItems	Retrieves a list of available add-on items by date. The user should provide at least the date range (a maximum of seven days), hotel criteria, and the item class or group. Specifying an item group retrieves a list of available items in that group. Setting the isPackage flag to true adds attached packages to each item in the list and setting the flag false ignores package information for items.

Function	Description
FetchAvailablePackages	Retrieves a list of available add-on packages and package groups by date. The user should provide at least a date range and hotel criteria. Confirmation identifier, leg confirmation identifier, number of rooms, and counts of adults and children are also allowed.
FetchCalendar	Retrieves a list with room occupancy, rate, and restriction details by date. The user should provide at least a date range and hotel criteria. Room type or block, adult or child guest counts, and rate plan information are also allowed.
FetchExpectedCharges	Retrieve a rate, taxes and package breakdown for stay for requested stay date, rate code, room type, property.
FetchItemGroups	Retrieves a list of available add-on items. Hotel criteria must be provided. Setting flag groupOnly to true (the default) retrieves the list of item groups and setting the flag to true retrieves a list of all available items.
GDSAreaAvailability	Fetches the availability status of a list of properties submitted in the request message. The response contains the property status, chain code, Property ID and the minimum and maximum rate ranges dynamically retrieved from the Opera database.
RegionalAvailability	Retrieves a list of hotels. The user must provide at least a geographic region code or hotel criteria, date range, and the number of guests. Number of rooms and a rate range are also allowed.
RegionalAvailabilityExt	Retrieves a list of and provides extended information about hotels. The user must provide at least a geographic region code or hotel criteria, date range, and number of guests. Number of rooms and a rate range are also allowed.

For the functional specifications, refer to the OWS Availability Message Spec document.

Brochure Service Functions

The following functions are currently provided through the Brochure Web Service:

Function	Description
SendBrochure	Sends a brochure, given the hotel criteria, brochure type, name identifier, and e-mail address.

For the functional specifications, refer to the OWS Brochure Message Spec document.

Guest Services Functions

The following functions are currently provided through the Guest Services Web Service:

Function	Description
FetchOptInSetup	Fetch Optin setup fields for requested resort.
UpdateReservationForOptIn	Update Reservation with OptIn Data.
UpdateRoomStatus	Update room status for requested reservation.
WakeUpCall	Retrieves the wakeup calls for a reservation. In addition, it can add wakeup calls and delete wakeup calls for the reservation. This function can only be utilized when a reservation is “IN HOUSE.”

For the functional specifications, refer to the OWS Guest Services Message Spec document.

Housekeeping Functions

The following functions are currently provided through the Housekeeping Web Service:
Allows for the inputting of housekeeping information via webservices.

Function	Description
FetchHouseKeepingDiscrepancies	Displays all rooms where the number status conflicts between Front Office and Housekeeping.
FetchHouseKeepingStatistics	Displays the housekeeping statistics information.
FetchHouseStatus	The House Status screen is a survey of all movements for the current date as well as any future date. Information available from this screen includes arrivals, departures, available rooms, housekeeping status, and expected occupancy for the date searched.
UpdateHouseKeepingDiscrepancies	Updates the HouseKeeping Discrepancies with actual status.

For the functional specifications, refer to the OWS Housekeeping Message Spec document.

Information Service Functions

The following functions are currently provided through the Information Web Service:

Function	Description
CurrencyConverter	Returns the converted amount, given the original currency code, original amount, converted currency code, and hotel criteria. An exchange type is also allowed.
GetScreenItems	Retrieves all the screen items.
QueryAwardsSchedules	Returns Awards Schedules for Rate, Product and Upgrade awards types.
QueryChainInformation	Returns information about the chain, given the chain code. Information returned includes addresses, emails, phone numbers, marketing text, loyalty program details, booking conditions, and frequent flyer program details. This functionality is only available when in MyFidelio mode.
QueryHotelInformation	Retrieves information about a hotel, given the hotel criteria. Contact and location details as well as information about facilities, amenities, services, and alternative properties are provided.
QueryLov	Retrieves a list of values, given the query type string. A wide variety of resort configuration details are viewable.
QueryPackageItems	Returns package groups/packages/item groups/items setup in the resort.
QueryProductItems	Returns product groups/products/item groups/items setup in the resort.
QueryRate	Retrieves rate information, given the hotel and rate codes. A date range is also allowed. The information includes policies, requirements, restrictions, and other descriptive details.

For the functional specifications, refer to the OWS Information Message Spec document.

Meeting Room Service Functions

The following functions are currently provided through the Meeting Room Web Service:

Function	Description
CreateBlock	Creates a block in OPERA.
MeetingAvailability	Single-property general or detailed availability for function space.
MeetingCancelEvent	(Not Implemented yet) Deletes existing event.
MeetingCreateEvent	Create an event which includes meeting room, miscellaneous Items, Menu, additional menu items and sleeping rooms.
MeetingCreatePackageEvent	Create a packaged event which includes Package ID, additional miscellaneous Items, menu items, and sleeping rooms.
MeetingFetchBlockDelegates	Retrieves the delegates (attendees) for a business block or event.
MeetingFetchContract	Fetches the CONTRACT, BEO or INQUIRY report and returns it as a link or emails it as a PDF.
MeetingFetchEvent	Fetch created event with the ability to return meeting room, miscellaneous Items, menu, and menu items.
MeetingFetchMenu	Returns configured menus and related menu items depending on the request. The data returned will be for one property.
MeetingFetchMenuItem	Depending on the request, returns configured menu items. The data returned will be for one property.
MeetingFetchMiscellaneousItem	Returns all configured miscellaneous items for one property.
MeetingFetchMyRegisteredEvents	Retrieves an attendee's registered events.
MeetingFetchPackageEvent	Fetch created packaged event will have the ability to return package, meeting room, miscellaneous Items, menus, additional menu items and miscellaneous Items.
MeetingModifyEvent	Modify created event.
MeetingModifyPackageEvent	Modifies the package event, which includes additional rooms and events, menus, items and catering packages.

Function	Description
MeetingMultiPropertyAvailability	Multi-property availability will return function space availability for more than one property depending on the request.
MeetingPackageAvailability	Returns general or detailed package availability for a property.
MeetingRegisterEventAttendees	Registers attendees for an event.
MeetingRoomCopyBlock	Copies an existing business block to a new date along with new business block data.
MeetingRoomCreateRelationship	Creates relationship between a contact profile and other profile types.
MeetingRoomFetchBlock	Returns the business block details including events, package events and sleeping rooms.
MeetingRoomFetchMyBlocks	Returns a list of business blocks by either ContactNameId or AccountNameId.
ModifyBlock	Modify a business block by adding notes, changing name, adding/deleting number of rooms, changing number of days, and changing room types.

For the functional specifications, refer to the OWS Meeting Room Message Spec document.

Membership Service Functions

The following functions are currently provided through the Membership Web Service:

Function	Description
AddBenefit	Applies benefits to the member who will be automatically upgraded to the level associated with the benefit code, given the membership ID.
AddPromoSubscription	Adds a promotional subscription to a membership, given the membership identifier and promotion information.
CancelECertificate	Cancel an e-certificate.
ConsumeECertificate	Consume an issued e-certificate.
ConsumePoints	Adds customer points and returns the transaction identifier, given the membership identifier and loyalty award program request details.
ConsumePointsOthers	Adds customer points and returns the transaction identifier,

Function	Description
	given the membership number and product award details.
DeletePromoSubscription	Deletes a promotional subscription from a membership, given the membership identifier and promotion information.
FavoriteGuest	Add/Delete/Fetch favorite guest to/from a profile.
FetchAvailableECertificates	Returns all configured E-Certificates for a date range.
FetchBenefits	Returns all benefit codes that are applied to a member.
FetchECertificates	Fetch all e-certificates issued to a profile.
FetchEnrollmentCode	Retrieves the enrollment code for a membership, given the membership identifier.
FetchMemberPoints	Returns member points information.
FetchMemberTierWizard	Returns information on the members upgrade requirements or the membership level on downgrade which will be evaluated based on the inputs of a future evaluation date and membership ID.
FetchMembershipTransactions	Retrieves a list of membership transactions, given the membership identifier or membership record.
FetchNextCardNumber	Retrieves a membership card number, given the membership type.
FetchPointsExchange	Retrieves the membership type points based on the exchange rate setup between the two membership types.
FetchProductAwards	Retrieves a list of product awards in effect, given the membership type and level, date range, and number of rooms. Details on the product award requirements are provided.
FetchPromoSubscriptions	Retrieves a list of promotional subscriptions attached to a membership, given the membership identifier.
FetchRateAwards	Retrieves a list of rate awards in effect, given the membership type and level, date range, number of rooms, points range minimum and maximum values, and a list of hotel references. Details on the rate award requirements are provided.
FetchStatement	Retrieves a hotel loyalty program statement for a membership, given the membership identifier and the statement identifier and date. Summary information and details on the membership transactions are provided.
FetchStatementRefs	Retrieves a list of statement records, given the membership identifier or membership record. Each record includes the

Function	Description
	statement identifier and date.
FetchUpgradeAwards	Retrieves a list of award upgrades in effect, given the membership type and level, date range, and number of rooms. Details on the upgrade and its requirements are provided.
IssueECertificate	Issue e-certificate to a profile.
ReIssueMemberCard	Member will be scheduled for next fulfillment export for code REISSUE NEW CARD.
TransferPoints	Transfer points from one membership type to another membership type.
UpdateEnrollmentCode	Adds or modifies the enrollment code of a membership, given the membership identifier and new enrollment code.

For the functional specifications, refer to the OWS Membership Message Spec document.

Name Service Functions

The following functions are currently provided through the Name Web Service:

Function	Description
DeleteAddress	Deletes a specific address from a profile, given the address identifier.
DeleteComment	Deletes a specific comment from a profile, given the name and comment identifiers.
DeleteCreditCard	Deletes a specific credit card record from a profile, given the credit card identifier.
DeleteEmail	Deletes a specific e-mail record from a profile, given the e-mail identifier.
DeleteGuestCard	Deletes a specific membership number from a profile, given the guest card identifier.
DeletePassport	Deletes the passport information from a profile, given the name identifier.
DeletePhone	Deletes a specific phone record from a profile, given the phone record identifier.
DeletePreference	Deletes a specific preference from a profile, given the name identifier and the preference.
DeletePrivacyOption	Deletes the privacy option settings for a profile, given the name

Function	Description
	identifier.
FetchAddressList	Retrieves a list of address records from a profile, given the name identifier.
FetchClaimsStatus	Fetches the status of the claim submitted by a profile.
FetchCommentList	Retrieves the list of comments from a profile, given the name identifier.
FetchCreditCardList	Retrieves the list of credit card records from a profile, given the name identifier.
FetchEmailList	Retrieves the list of e-mail address records from a profile, given the name identifier.
FetchGuestCardList	Retrieves the list of membership records from a profile, given the name identifier.
FetchName	Retrieves the name record from a profile, including any birthday and gender information, given the name identifier.
FetchNameUDFs	Retrieves the list of user-defined field values from a profile, given the name identifier.
FetchPhoneList	Retrieves the list of phone records from a profile, given the name identifier
FetchPreferenceList	Retrieves the list of preferences from a profile, given the name identifier.
FetchPrivacyOption	Retrieves the list of privacy option settings for a profile, given the name identifier.
FetchProfile	Sends profile request to Opera and Opera returns all of the profile information in the response message.
FetchProfileBenefits	Fetch all promotions and e-certificates associated with a profile.
FetchSubscription	Retrieves all the external system NameIds of the profile.

Function	Description
GetPassport	Retrieves the passport information from a profile, given the name identifier.
InsertAddress	Adds an address record to a profile, given the name identifier and address information to insert.
InsertClaim	Submits a claim for a profile.
InsertComment	Adds a comment to a profile, given the name identifier and comment information to insert.
InsertCreditCard	Adds a credit card record to a profile, given the name identifier and credit card information to insert.
InsertEmail	Adds an e-mail address record to a profile, given the name identifier and e-mail address to insert.
InsertGuestCard	Adds a membership record to a profile, given the name identifier and membership information to insert.
InsertPhone	Adds a phone record to a profile, given the name identifier and phone information to insert.
InsertPreference	Adds a preference to a profile, given the name identifier and preference description.
InsertUpdateNameUDFs	Adds to or modifies the list of user-defined field values in a profile, given the name identifier and a list of UDF records.
InsertUpdatePrivacyOption	Adds to or modifies the list of privacy option settings for a profile, given the name identifier and a list of privacy option setting records.
NameLookup	Retrieves a list of profile records, given the name look-up credit card or membership criteria.
RegisterName	Creates a new profile and its name identifier, given at least the person's name record. Native name record, birth date, gender, address record, phone record, passport record, login name, and login password are also allowed. Specifying the login

Function	Description
	information also creates a web user.
TravelAgentLookup	Allows the look up of travel agent given the NameID (IATA Number). Returns the travel agent's company name, address, phone number, email address, and IATA number in the response.
UpdateAddress	Modifies an address record in a profile, given the address information.
UpdateClaim	Adds comments to an existing claim.
UpdateComment	Modifies a comment record in a profile, given the name identifier and comment information.
UpdateCreditCard	Modifies a credit card record in a profile, given the credit card information.
UpdateEmail	Modifies an e-mail address record in a profile, given the-e-mail address information.
UpdateGuestCard	Modifies a membership record in a profile, given the membership information.
UpdateName	Modifies the name information in a profile, given at least the name identifier and person's name record. Native name record, birth date, and gender are also allowed.
UpdatePassport	Adds or modifies the passport record in a profile, given the name identifier and passport information.
UpdatePhone	Modifies a phone record in a profile, given the phone information.

For the functional specifications, refer to the OWS Name Message Spec document.

Reservation Service Functions

The following functions are currently provided through the Reservation Web Service:

Function	Description
AccompanyGuest	Attaches an accompanying guests to a booking. Requires 2 calls: The first call creates the booking and receive the response for the primary guest and the second call attaches or detaches the accompanying guests. This function requires the Reservation group application function ACCOMPANYING GUEST to be enabled in PMS.
AssignRoom	Assigns the room number for a reservation, given at least the hotel criteria and reservation identifier. Requested room number and station identifier are also allowed.
BookHoldItems	Attaches held inventory items to a reservation, given the hotel criteria, confirmation or reservation identifier, and the list of hold item identifiers to attach.
BreakShare	Break a shared reservation.
CancelBooking	Cancels a reservation and returns a cancellation identifier, given at least the hotel criteria, confirmation identifier, and cancellation type summary. GDS identifier and Leg confirmation identifier are also allowed.
ClearItemHold	Returns a held item to available inventory, given the hotel criteria and hold item identifier.
CombineShare	Share two reservations.
ConfirmBooking	Confirms a booking for a GDS channel type, given hotel reference information and the booking confirmation number, and the GDS ID (optional). If this is a multileg booking, the leg number is also allowed. The response includes the result and confirmation number. Used only when Session Control is active in Channel setup.
CreateBooking	Creates a new reservation and returns confirmation and reservation identifiers, given at least hotel reservation criteria including room stay, guest information, and confirmation instructions. Reservation history information, user defined values, and bill header are also allowed. Multiple rate codes, including multiple child rates, are supported. CreateBooking also allows packages to be attached to the reservation during the booking process.

Function	Description
CreateItemHold	Creates a hold item and returns a hold item identifier, given the hotel criteria, item code, quantity, and date range.
CreateMultipleBookings	Create multiple bookings.
DeleteInventoryItem	Removes inventory attached to a reservation, given at least the hotel criteria, confirmation or reservation identifier, and item or item group representing a collection of items to remove. Leg confirmation identifier is also allowed. If a single deleted item is the sole element of a package, the package is also removed from the reservation.
DeleteAccompanyGuest	Detaches the accompanying guest from the reservation. This function requires the Reservation group application function ACCOMPANYING GUEST to be enabled in PMS.
DeletePackages	Removes an add-on package from a reservation, given at least the hotel criteria, confirmation identifier, and product code. Leg confirmation identifier is also allowed.
EmailConfirmation	Sends a reservation confirmation, given the confirmation identifier and e-mail address.
FetchAvailableOffers	Fetch available Upsell offers for reservation.
FetchBookedInventoryItems	Retrieves the list of inventory items attached to a reservation, given at least the hotel criteria and confirmation or reservation identifier. Leg confirmation identifier is also allowed. Details including the item code and quantity are provided for each date.
FetchBookedPackages	Retrieves the list of packages and package groups attached to a reservation, given at least the hotel criteria, and confirmation identifier. Leg confirmation identifier is also allowed. Details about each package or package group and expected charges by date are provided.
FetchBooking	Retrieves a reservation, given at least the confirmation identifier or GDS identifier or customer reference identifier. Hotel criteria, leg confirmation identifier, and external hotel reference and leg confirmation number are also allowed. Details are provided on the room stay, guest profile, and confirmation instructions.
FetchBookingForPointUpdate	Retrieves a custom list of reservations for assigning consumed membership points, given the hotel criteria.
FetchHoldItems	Retrieves a list of items on hold, given the hotel criteria and a comma-separated list of hold item identifiers. The item code, quantity, and date range of the held items are provided.

Function	Description
FetchRoomUpgrades	Fetch Room Upgrade available for member reservation.
FetchSummary	Retrieves a reservation, given the confirmation identifier. Summary information is provided on the status, room stay, and guest profile.
FutureBookingSummary	Retrieves a list of reservations for future dates, given some filtering criteria. The filtering may be by date range (booked or arrival date), name or corporate identifier, last name, first name, credit card number, search level (subgroup, group, or booker), and/or other defined filters (creation date, contract identifier, membership record, hotel criteria, confirmation identifier, reservation identifier, and key track data). Source and Origin codes for reservations can also be returned in the response.
GuestRequests	Add or remove or update or fetch comments, traces, specials attached to a reservation.
IgnoreBooking	Ignores (cancels) a booking for a GDS channel type, given hotel reference information and the booking confirmation number, and the GDS ID (optional). If this is a multileg booking, the leg number is also allowed. The response includes the result and confirmation number. Used only when Session Control is active in Channel setup.
ModifyBooking	Modifies a reservation, given the updated reservation record. The confirmation identifier in the record is used to fetch the existing reservation. Key track data is also allowed. Multiple rate codes, including multiple child rates, are supported. ModifyBooking also allows packages to be attached or updated during modification of the booking. In addition, moving a reservation from one resort to another with the same change is also supported.
ModifyItemHold	Modifies a held item, given the hotel criteria, hold item identifier, and the new quantity and/or date range values.
ReInstateReservation	ReInstate a cancel/noshow reservation to reserved.
ReleaseRoom	Releases the assigned room number from a reservation, given at least the hotel criteria and reservation identifier. Station identifier is also allowed.
SetDailyPoints	Retrieves a reservation for assigning consumed membership points, given the hotel criteria, confirmation identifier, and list of earned points by date range or start date plus duration.
UpdateInventoryItem	Adds or updates inventory items attached to a reservation,

Function	Description
	given at least the hotel criteria, confirmation or reservation identifier, item code or item group, and desired quantity and/or date range. Leg confirmation identifier is also allowed. The quantity and date range apply to an item or all items in an item group.
UpdatePackages	Adds or updates a package or package group attached to a reservation, given the quantity. Hotel criteria, confirmation identifier, product code, leg number, and stay date range is also allowed. Details about the package or package group and expected charges by date are provided.
UpgradeReservation	Upgrade Reservation from one room type to another room type.
UpsellReservation	Upsell current reservation to selected offer by user.

For the functional specifications, refer to the OWS Reservation Message Spec document.

Reservation Advanced Service Functions

The following functions are currently provided through the Reservation Advanced Web Service:

Function	Description
AdditionalKeys	Generates additional room keys for an in-house guest, given at least the hotel criteria, reservation identifier or key track, station identifier, and desired number of keys. User identifier, key encoder, and get-key-track indicator are also allowed. If the latter indicator has value true, the key track is returned.
AlternateRooms	Generates a list of alternate room types and rooms for a guest, given the hotel criteria and either reservation identifier or key track. Rate information is also provided.
AssignRoom	Assigns the room number for a reservation, given at least the hotel criteria and reservation identifier. Requested room number and station identifier are also allowed.
CancelCheckIn	Converts the status of a reservation from checked-in to due-in, given at least the hotel criteria, and either reservation identifier or key track. Printer designation and cancellation type summary are also allowed.
CheckIn	Checks a guest in, given at least the hotel criteria and reservation identifier or key track. Credit card information, printer designation, number of keys, key encoder, approval code, get-print-out indicator, and get-key-track indicator are also allowed. If the indicators have value true, the registration is printed and

Function	Description
	the key track is returned. Check-in information including the invoice number and room number is provided.
CheckOut	Checks a guest out and generates the final bill, given at least the hotel criteria and reservation identifier or key track. Credit card information and printer designation are also allowed.
ExternalPayment	Calls an external payment provider and submits payment data to the provider for a start and finish payment process.
FetchKeyData	Retrieves KeyTrack data attached to the reservation, given the reservation information.
FetchPromotionCode	Retrieves a list of promotion codes available at the hotel.
FetchResPromotionCode	Retrieves promotion codes attached to the reservation.
FetchRoomSetup	Retrieves room setup information, given the room type and room number. Hotel reference information is also allowed. Returns the room type and number, suite type (if applicable), phone number, maximum occupancy, housekeeping section code, smoking preference, and room description, amongst other details about the room is setup.
FetchRoomStatus	Retrieves room status information for the requested room or room type. The response includes the room number and type, next reservation date, room status and the valid dates of the room status, front office status, occupancy condition and housekeeping status, and the service status.
GenerateRegistrationCard	Generates a registration card for the guest during checkin.
GuestMessages	Retrieves the messages not yet flagged as received for a guest, given at least the hotel criteria and reservation identifier or key track. Line length is also allowed. Arriving, departing, and in-house guests are considered.
Invoice	Retrieves an invoice, given at least the hotel criteria and reservation identifier or key track. Printer designation is also allowed. An itemized bill is provided.
MakePayment	Post a payment, given at least the hotel criteria, reservation identifier or key track, credit card information, and charge amount. Posting date, posting time, short comment, long comment, station identifier, and user identifier are also allowed.

Function	Description
PayRouting	Retrieves lists of routing instructions and payment methods, given the hotel criteria and reservation identifier or key track. Window number, window user, and room number are included. This information is provided only for notification that part of the stay may be handled by different pay windows.
PostCharge	Adds charges to a guest account, given at least the hotel criteria, reservation identifier or key track, and charge amount. Posting date, posting time, short comment, long comment, station identifier, user identifier, account number, and article number are also allowed.
PrintPreCheckOutBill	Generates a bill, given at least the hotel criteria and reservation identifier or key track. Printer designation is also allowed. The bill may either be generated as a PDF file that the kiosk may retrieve over the network and print, or it may be directly printed.
ReleaseRoom	Releases the assigned room number from a reservation, given at least the hotel criteria and reservation identifier. Station identifier is also allowed.
ReservationRequestCode	Retrieves a list of special requests with a reservation, including profile preferences, given the hotel criteria and reservation identifier or key track. The request type, code, source, and description are provided.
SetKeyData	Stores key track data in Opera, given the hotel reference information, reservation ID, and/or key track data.
SetResPromotionCode	Stores promotional information related to a reservation, given the reservation request information and the promotion code(s).
UpdateMethodOfPayment	Updates the method of payment, given the Hotel Code, Key track or Opera Reservation Id, Folio View Number, and Credit Card data. From a kiosk, the method of payment can be changed on individual folio windows, not just for the entire reservation.

For the functional specifications, refer to the OWS Reservation Advanced Message Spec document.

Security Service Functions

The following functions are currently provided through the Security Web Service:

Function	Description
AuthenticateNRUser	Authenticates a non-registered (no membership profile) user and returns the name identifier, given the user last name, confirmation identifier, and credit card number.
AuthenticateUser	Authenticates a user and returns the name identifier, given the membership number, last name, and password.
CreateAppUser	Creates an application user profile and returns the name identifier, given the profile details, and user group type, as well as a login name, password, and expiration date.
CreateUser	Creates a web user, given the name identifier and a login name and password.
DeleteAppUser	Deletes an application user profile, given the login name and password.
ExtAuthenticateUser	Authenticates a user and returns the login profile, given the login name and password.
FetchAppUser	Retrieves a list of application users, given the login name. Login and name profile details are also returned.
FetchQuestionList	Retrieves a list of predefined secret questions that users can choose from and answer for further authentication.
GeneratePassword	Returns an automatically generated password, given the login name.
LoginAppUser	Authenticates an application user using the login name and password credentials, and returns the security identifier (company name and user group type) and license key.
UpdateAppUser	Modifies an application user profile and returns the name identifier, given the login name, password, and expiration date, as well as the name profile and user group information.
UpdateAppUserPassword	Updates the user's password given the application user's loginName, oldPassword, and newPassword. This allows users to change their passwords without logging into the application. A Success or Error Result is returned.
UpdatePassword	Modifies the login password, given the membership number, last name, old password (if not a new user), and new password.
UpdateQuestion	Modifies the answer to a user's chosen secret question, given the name identifier, pre-defined question identifier, and

Function	Description
	new answer.
UserAccountLock	Performs a lock or unlock operation on a user account, given the login name and the lock action type.
ValidateQuestion	Validates a user answer to a chosen special question and returns the password, given the membership number, question identifier, and an answer.

For the functional specifications, refer to the OWS Security Message Spec document.

Stay History Service Functions

The following functions are currently provided through the Stay History Web Service:

Function	Description
StayHistory	Retrieves a list of past reservations, given the guest name and agent identifiers. A date range is also allowed. Only reservations with status checked-out or cancelled are returned. Considerable detail is provided about the guests, amenities, services, facilities, rates, charges, and communications.

For the functional specifications, refer to the OWS Stay History Message Spec document.

Unit Owners Service Functions

The following functions are currently provided through the Unit Owners Web Service:

Function	Description
AddProfileToContract	Adds a profile to a contract, given the guest name, requesting name, contract, and resort identifiers, and the relationship of the guest to the contract owner.
CreateContract	Create a contract.
DeleteContract	Delete a contract.
DeleteProfileFromContract	Removes a profile from a contract, given the guest name, requesting name, resort, and resort identifiers.
FetchAuthorizedProfiles	Retrieves a list of contracts, given the name, resort, and/or contract identifiers. Contracts of the owner, co-owners, and family/friends are included. Information about the contract type, unit, effective dates, rate plans, and attached profiles is provided.
FetchContract	Retrieves a contract, given at least the name identifier. The resort and contract identifiers and the requestor role (primary-owner, co-owner, or friends-and-family) are also allowed, and are returned. Information about the contract type, unit, effective dates, rate plans, and attached profiles is provided.
FetchContractDetails	Retrieves details of a contract, given the name and resort identifiers. The contract identifier is also allowed and is returned. Information includes financial arrangements, restrictions, contract documents, and insurance policies in addition to information about the contract type, unit, effective dates, rate plans, and attached profiles.
OwnedUnitAvailability	Retrieves unit availability by contract, given at least the name identifier and stay date range. Contract and resort identifiers as well as a list of guest counts and the room type code are also allowed. For each contract, information includes the contract identifier, type, unit, effective dates, rate plans, and attached profiles, as well as a list of room types with rate details.
OwnerStatementDetails	Retrieves the file names of the owner statement and summary information, given the name and resort identifiers and the stay date range.
UpdateContract	Update the contract with the contract details.
UpdateProfileInContract	Modifies a family/friend guest profile, given the guest name, requestor name, contract, and resort identifiers as well as the relationship of the requestor.

For the functional specifications, refer to the OWS Unit Owners Message Spec document.

What are the Benefits of Opera Web Suite?

By standardizing communications between our customers, Central Reservation Systems (CRS), Opera Reservation Systems (ORS), and Opera Web Suite (OWS) offer a number of potential benefits for designing, implementing, and marketing travel-related applications. By combining existing Opera products with new technology standards, OWS offers a more streamlined and robust method for designing travel applications.

The benefits of OWS include:

- Platform independence.
- Encapsulated business logic with the CRS.
- More connectivity options.
- CRS communications components hosted by Micros-Fidelio.
- Flexibility and extensibility
- International scope
- Enhanced availability features
- Flexible profile management
- Stay history services
- Membership services
- Advanced logging

Platform Independence

Because OWS adheres to cross-platform Web Services standards, OWS does not restrict the environment for developing or deploying client applications. OWS client applications can be deployed in any environment that supports Hypertext Transfer Protocol (HTTP), including Windows® and UNIX, without requiring bridges or specialized interfaces. In addition, a wide variety of languages and development developer toolkits can be used to design and deploy client applications for OWS.

Opera Business Logic Encapsulated With the CRS

As OWS evolves, more and more Web Services will encapsulate the business logic needed to communicate with the CRS into a more streamlined, intuitive framework. As a result, developers focus on writing their applications rather than communicating with the CRS. The streamlined usability of OWS could potentially reduce development time by 30 to 70%, depending on the type of application and development environment. The Web Services call on Opera logic, which means that an inventory query done via the hotel call center will return the same availability to the Web user. There is only one database to setup with a common set of booking rules, applicable to both Web and call center-originated reservations. There is only one database to set restrictions, inventory counts, and rate periods with associated amounts, which will apply across the board to different sources of availability and booking messages.

More Connectivity Options

Client applications can connect with OWS using either the Internet or a dedicated connection. Customers can select the type of connectivity that is most appropriate for their requirements and resources.

CRS Communications Components Hosted by Micros-Fidelio

Previously, developers needed to include complicated code in their applications for identifying and routing transactions with the CRS. OWS streamlines communications with the CRS for customer-hosted applications, and hosts any required components at Micros-Fidelio data center, which internally supports identifying, routing, and transforming transactions with the Micros Fidelio ORS system (CRS).

Flexibility and Extensibility

OWS specifications provide travel service providers with the flexibility they need to develop, test, and deploy new services, the minimum necessary functionality to provide reliable interactions between customers and the systems owned and maintained by the companies serving them.

As the versions of our ORS and Web Services evolve, there are plans in place to ensure that incompatibility issues are kept to a minimum.

International Scope

OWS was designed to be a multi-language platform that supports the Unicode character set standards, including ISO countries, and ISO currencies and city codes, based on the United Nations code for trade and transport locations.

Enhanced Availability Features

In addition to traditional availability for a single property, OWS offers back-end functionality within Opera to retrieve real time availability of hotels grouped by geographical regions, cities, CRO and Chain codes. The real-time regional availability response includes the available status of a property, a minimum and maximum rate range, and property information. Search by alternate dates enhances availability requests by providing the next available dates if the primary dates requested are unavailable. A typical request includes items such as arrival date, number of nights, number in party, and possibly a specific rate code or room type. OWS exposes Opera's availability search capability to a promotion code, a membership number, a group code and introduces the concept of allotment sales to Micros Systems electronic distribution functionality.

Flexible Profile Management

Allow users to register in your database! Give them the option to enter their preferences, credit cards, and contact information. All of the Opera functionality behind the OWS can be used independently, allowing hotels to control what information they would like to capture from their registered profiles!

Stay History Services

This feature allows guests to retrieve past booking information with the condition that they must be checked out or cancelled. The ability for a guest to look at the history of their business on a property or properties is a great offering from hotels to their frequent travellers.

Membership Services

OWS exposes the Opera's capability to list a member's summary of statements. It also allows members to drill into each statement for details. Furthermore, it allows members to retrieve transactions that are not yet part of any statement, which means they can have an up to date status of the points they have earned.

Advanced Logging

OWS incorporates advanced logic capabilities that allow users to have a distinct count of message types and processing times.

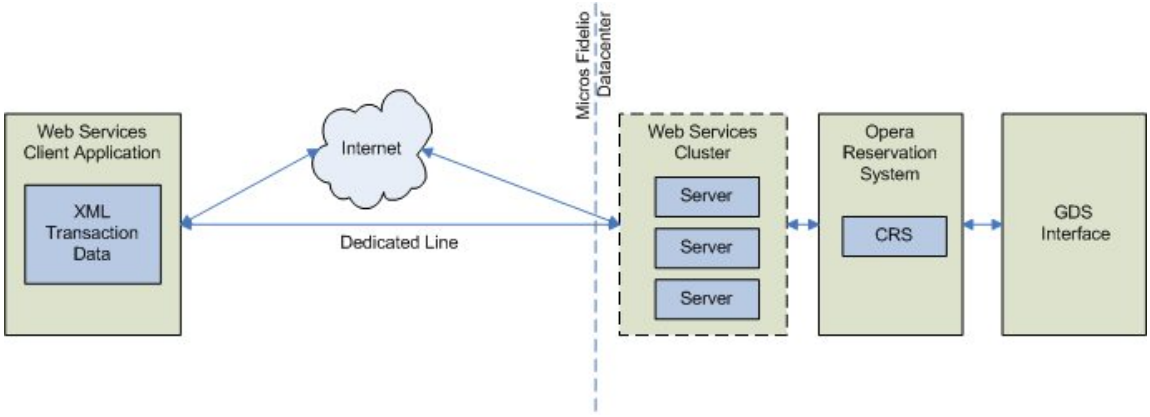
How Does Opera Web Suite Work?

Opera Web Suite (OWS) combines GDS and CRS interfaces from ORS with Web Services technologies. The OWS product performs two key functions:

- Manages communications between client applications and the ORS.
- Converts between XML data from client applications and ORS proprietary structured data formats, which are used by the CRS and/or GDS, depending on the channel used.

However, with OWS, the components are hosted by Micros-Fidelio on secured servers. By hosting their OWS interface at Micros-Fidelio, customers do not need to install, configure, or maintain components on-site.

The following graphic depicts the basic flow of information in a transaction that uses OWS to communicate with the CRS and travel applications. For additional information about the details of how data transfers in a Web Services environment, please refer to **Appendix A: Know-How on Web Services (page 19)**.



The following description explains how a message is sent from a client application to the CRS.

Sending Requests

The client application creates a message with the request data in an XML (Extensible Markup Language) format.

Client applications can be written in any language or platform that supports Web Services technologies or protocols.

Regardless of the programming language or platform, the application code must use XML for the actual request data that is sent to the CRS. This XML transaction is wrapped in a Simple Object Access Protocol (SOAP) envelope with a specific header and body. SOAP is XML data that directs the message to be delivered to the correct Web Service, and allows the message to be appropriately processed by the Web Service.

Network Connection: Public Internet or Private Dedicated Connection

The message is sent from the client application to the OWS servers.

Messages in Web Services are sent via Hypertext Transfer Protocol (HTTP), which is a common method of transferring HTML and XML data. OWS can be supported through HTTP used on either a public Internet connection or a private dedicated connection.

The amount of network bandwidth is the most important factor in planning for adequate capacity. Regardless of the type of network connection, the size of the connection has the most direct affect on the speed and capacity of the client application.

Web Services Servers

The Web Services servers receive and process the request message.

The SOAP message that was attached to the request message channels the request to the appropriate Web Service, and essentially instructs the Web Service in how to process the message. A special header is included with the XML application data to identify the sender of the transaction to the Web Service.

Some Web Services messages end at the CRS servers, while others are forwarded to the GDS servers to be converted and processed by the CRS. How messages are handled depends on the type of origin and destination. For example, the origin is specified as GDS and the destination is specified as Web, which then presents GDS availability on the Web channel. The decode request and response is handled completely within the service. However, a GDS availability request requires additional processing by the CRS and the GDS. In this case, the Web Services act as a sort of intermediary for processing the availability request and response data.

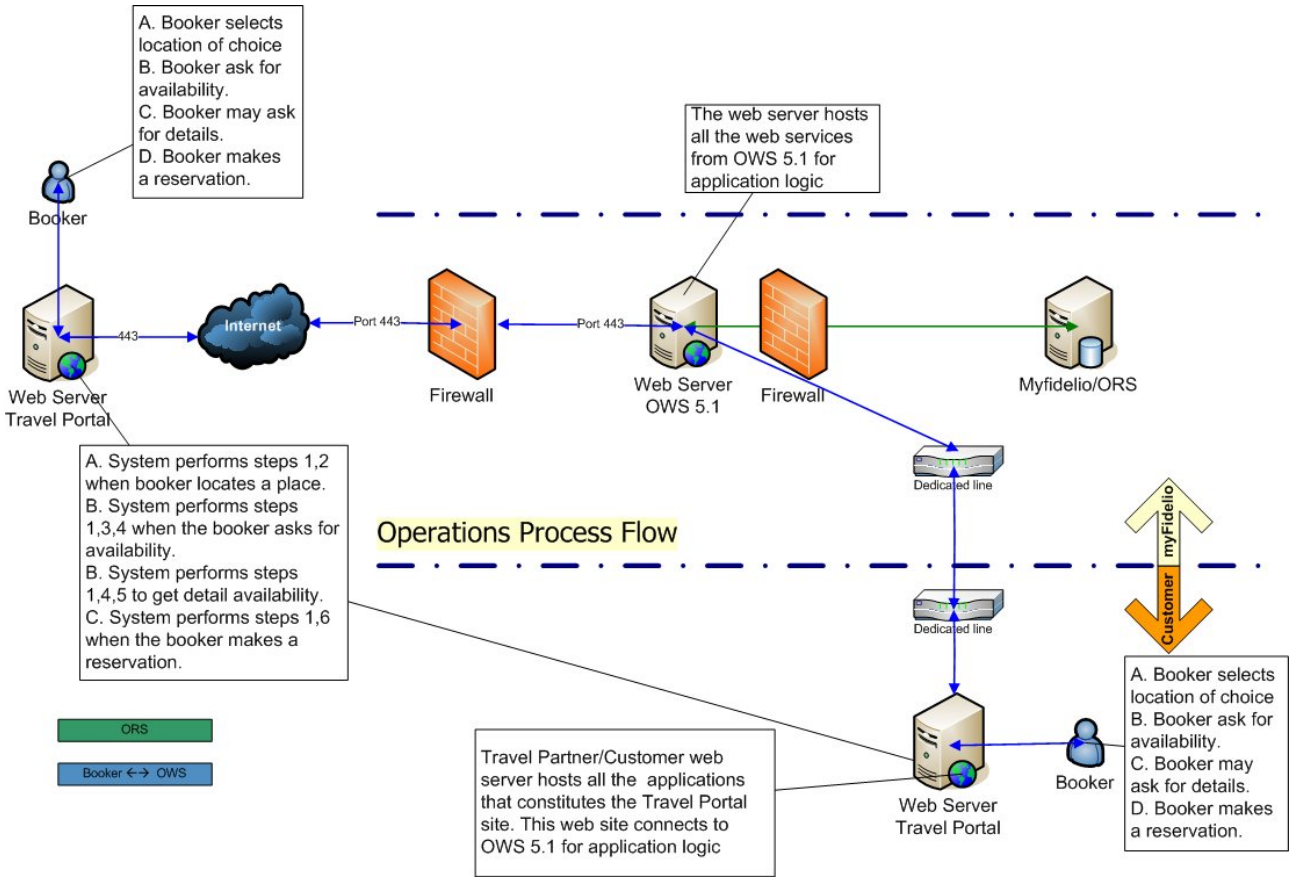
Retrieving Responses

The response message follows the same path as the request message, but in reverse. The proprietary structured data response is sent from the CRS or GDS to OWS, which wraps the response in a SOAP envelope and forwards the message back to the client application. Again, the response is sent via HTTP through either the Internet or a dedicated connection. After the client application receives the message, it takes XML response data and displays that data within the client application.

The Booking Process

A complete booking process from a client application to OWS is based on the request-response schema explained above, and follows the steps performed by the travel partner/customer's system requested by a booker:

1. The client application authenticates itself using the security service.
2. The client application selects city/region/chain by making a call to the information service.
3. The client application asks for a list of available properties by using the Availability service.
4. The client application selects a specific property and sends a General Availability Request using the Availability service.
5. The client application selects a chosen rate room for the property and sends the Detail Availability Request using the same Availability service.
6. The client application calls Create Booking using the Reservation Service, and retrieves the response indicating the booking details.



Requirements and Recommendations

Planning for OWS includes familiarity with not only the required technologies and skill sets, but also preliminary planning to design a client application that can appropriately support Web Services for the expected amount of traffic within the scope of available resources.

OWS Standards and Protocols

OWS conforms to the following industry standards and protocols.

Message Envelope	SOAP 1.1	http://www.w3.org/TR/SOAP/
Message Transfer	HTTP 1.1	http://www.w3.org/Protocols/
Encryption	SSL 3.0	http://home.netscape.com/eng/ssl3/index.html
Web Service Description Language	WSDL 1.1	http://www.w3.org/TR/wsdl
Data Transfer	XML 1.0	http://www.w3.org/XML

Tools and Platforms

Any development tools or platforms that are compatible with the above standards and protocols can be used. These tools and platforms include, but are not limited to:

- Microsoft.NET
- IBM WebSphere® Studio Application Developer
- Apache Axis

Developer Skills

Developers should have familiarity with XML Web Services, as well as the standards and protocols listed within this section. Particularly when using the XML Web Service, knowledge of the CRS/OTA business model is also helpful.

Connectivity

OWS can be supported through either an Internet connection or a dedicated connection. Internet connections must be secured via a Secured Socket Layer (SSL) protocol, as well as the security functions provided for OWS. Because Web Services formats and standards are not limited to use on the Internet, the same protocols apply to the message, regardless of the type of network connection.

Security

The OWS servers are secured, and access is granted only to the specific Web Services for which the client is licensed. The Internet connections to OWS must also be encrypted via an SSL protocol. For dedicated connections, security methods for the connection must be defined in the client application, or by using Point-To-Point tunnelling protocols with appropriate encryption methods. Because Web Services formats and standards are not limited to use on the Internet, the same protocols apply to the message, regardless of the type of network connection.

Capacity Planning

All capacity planning decisions should initiate from the projected scope and requirements of the client application. Capacity relates to the functionality of two basic components:

- The design and required functionality for the client application.
- The size and type of network connection.

The expected volume of Web Service calls is the key factor in capacity planning. Issues for determining expected volume include:

- The expected look-to-book ratio between requests to the CRS and actual bookings.
- The expected number of transactions.
- The types of Web Service calls planned, and their expected message sizes.

The most appropriate type of connection depends on the requirements of the client application and the available resources. Dedicated connections are generally faster and have more reliable availability, while Internet connections are typically less expensive. However, many of these factors vary depending on the environment. For example, in many locations dedicated connections are prohibitively expensive for some organizations, while in other locations, Internet connections may actually be more reliable than the local telecommunications required for a dedicated connection.

Support for Opera Web Suite

Technical support is provided during the development of new applications and modification to existing applications. Micros-Fidelio supports OWS only; the programming technique, language, or environment is not supported.

Online Help

OWS includes a Software Developers Kit that provides theoretical and practical information for using Opera Web Suite, including:

- A general overview of Web Services.
- Descriptions of each OWS Service.
- Recommendations and requirements for application design and security.
- Getting started guides and sample transactions.
- Detailed information about how to use each Web service, including the individual transactions.

Learn More About Opera Web Suite

There are several ways to learn more about Opera Web Suite.

Web Site

Additional information about Opera Web Suite is available at:

http://members.micros.com/members/product_support/micros_fidelio_products/opera/vendors/OWS/index.asp

Contact Us

Prospective customers can contact one of our Micros-Fidelio regional offices.

Customers in Europe, Africa & the Middle East should contact:

Micros-Fidelio Software Deutschland GmbH & Co.KG

Regional Operations center

Euro Center

Europadamm 2 –6

Neuss, Germany

Phone: +(49) (2131) 137-0
Fax: +(49) (2131) 137-702

Customers in Asia Pacific Region should contact:-

Micros Fidelio Software

Regional Operations Center

Sydney

Australia

Phone +61 2 9485 1000

Fax +61 2 9485 1099

Customers in the US should contact:-

Micros Systems Inc

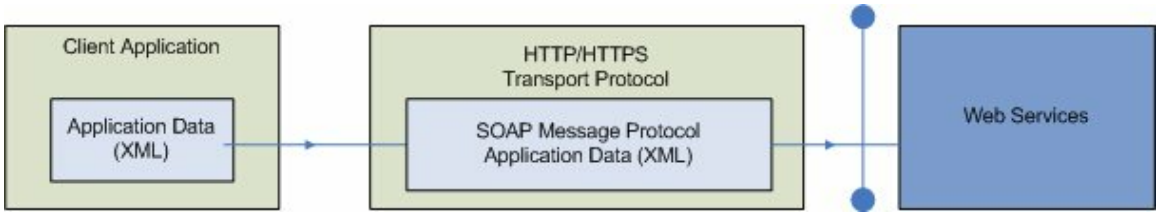
7031 Columbia Gateway Drive
Columbia, MD 21046-2289

Phone +1 443 285 6000

contact@micros.com

Appendix A: Know-How on Web Services

Understanding the underlying concepts behind Web Services helps to better illustrate why Web Services streamline and transform communication with the CRS. For all of the terms that surround Web Services, the basic premise of Web Services is not very complex. The following diagram shows the basic flow of data between a client application and a Web Service:



XML (Extensible Mark-up Language)

XML is the format for message data used with Web Services.

What is XML?

Like its "cousin" HTML, XML is not a programming language, but rather a system for tagging chunks of data so that they can be sent and received in a defined, self-descriptive fashion. HTML tells you how the data should *display*. XML not only controls the display of data, but also tells you *what* the data is and *how* to use the data.

XML is commonly used across the Internet, but can also be used to transport information in a number of other environments.

XML has several particularly useful qualities that can be applied to Web Services:

- *Customizable tags* that can be used to precisely define content for a specific application. The tags, together with the data described by the tags, form an XML *element*.
- Developers typically use an XML schema to describe specific XML data and define how to use that data.
- *Data storage* within the application itself, which can increase product usability, reduces transaction time, and decrease server usage.
- *Separation of display and content*, in which changing the display of content does not affect the content itself.

How do Web Services use XML?

XML is used in two main ways for Web Services:

- The actual application data that is sent between the systems is formatted in XML. For OWS, the data that is sent between a client travel application and the Opera Web Suite is in XML.
- Message protocols and supporting data that are used to send the application data are also written in XML.

Simple Object Access Protocol

Simple Object Access Protocol (SOAP) is a specific kind of XML message that is "wrapped around" the XML application data. SOAP directs the application data to the Web Services and defines how the Web Service processes the data.

Written in XML, the SOAP message is then wrapped in a *SOAP envelope* around the application data, which is also written in XML. SOAP indicates:

- How a message is being sent
- Where the message is being sent
- What kind of message is being sent

For example, in OWS, a SOAP message for a hotel general availability request would indicate that the XML data for the transaction should be sent:

- From the client application via HTTP (the standard for Web Services data transaction).
- Specifically to the OWS Availability Service (which processes the requests).

Web Services Definition Language

The Web Services Definition Language (WSDL) provides a compilation of the functions that can be performed by a specific Web Service.

The WSDL is another XML document that contains the XML schema about how to access and use the functions for a specific Web Service. For example, the WSDL for the Availability Web Service contains information about functions such as Regional Availability transactions, receiving transactions, and beginning sessions. When a SOAP message is sent to the Web Service from the client application, the data about how to use that service has been obtained from the WSDL.

Typically, the WSDL programmatically interfaces with the software developer toolkit. Before designing a client application, developers create a proxy (copy) of the WSDL, which then provides them with the Web Service parameters needed to design their application and data structures.

Hypertext Transfer Protocol

Hypertext Transfer Protocol (HTTP) is the method by which XML application data and SOAP data are sent between client applications and Web Services.

HTTP is a communications protocol for transferring data that is already commonly used on the Internet. HTTP is used extensively on the World Wide Web to transfer HTML from Web servers to Internet browsers. The **http://** prefix in front of a web site address indicates that HTTP is the protocol being used by that web site.

Because XML and HTML are related, XML can also use HTTP as a transfer protocol. Web Services do not strictly require the use of HTTP; however, it is currently the most commonly used transfer protocol for Web Services.

While HTTP is used most visibly on the World Wide Web, its use is also not limited to the Internet. For example, OWS clients also have the option of using HTTP across a dedicated connection or via the Internet.