

# Oracle® Hospitality Cruise Shipboard Property Management System OHC WebServices Technical Specification



Release 23.1  
F84841-02  
November 2023

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

Oracle Hospitality Cruise Shipboard Property Management System OHC WebServices Technical Specification, Release 23.1

F84841-02

Copyright © 1995, 2023, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, and MySQL are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

# Contents

## Preface

---

## 1 General Overview

---

Handling for JSON Format	1-1
Passing Date Variable using XML format	1-1
Connecting to OHC Web Services using Microsoft .NET	1-3
Recommended EndPoint Setting	1-3

## 2 General Process Flow

---

## 3 Web Methods

---

FidelioSPMSWSJsonGet	3-1
FidelioSPMSWSJsonPost	3-6
FidelioSPMSWSXML	3-9

## 4 Functions

---

ActivateCreditCard	4-1
AddHKTask	4-1
AddRouting	4-2
AdvanceGuestSearch	4-3
AmenityCompleted	4-5
ApproveCrewWorkedHoursDaily	4-6
ChangeBoardStatus	4-6
ChangePassword	4-7
CheckCabinAvailable	4-8
Checkin	4-8
CheckSag	4-9
ClockInMobile	4-10
ClockInOutGetStatus	4-10

ClockOutMobile	4-11
DeactivateCreditCard	4-12
DeleteAmenity	4-12
DeleteComment	4-13
DeleteGuestMessage	4-13
DeleteLuggageLink	4-14
DeleteResolution	4-14
DeleteRouting	4-15
DeleteSpecialRequest	4-16
DeleteTravelDoc	4-17
DeleteWorkOrder	4-17
DeleteWorkOrderAssignment	4-18
DeleteWorkOrderImage	4-18
DeleteWorkOrderItem	4-19
EvtBooking	4-19
ExcBooking	4-22
ExcBookingPackage	4-25
FCUIFolio	4-29
FCUIGuestInquiry	4-30
FCUIPayment	4-41
FCUIPosting	4-44
Sample JSON Http Post	4-52
FCUIUDFUpdate	4-53
FCUIUpdatePin	4-53
GangwayGuestSearch	4-54
GetAccBalance	4-55
GetAdvanceLoyaltyDiscount	4-56
GetCheckImage	4-57
GetCreditCardInfo	4-58
GetEvent	4-59
GetEvtFolio	4-64
GetExcFolio	4-67
GetExcPackage	4-70
GetExcursion	4-75
GetFreeCabin	4-80
GetHouseKeepingSections	4-82
GetILOViolations	4-83
GetLuggageInfo	4-84
GetMicrosAdvanceDiscount	4-86
GetPackageInfo	4-86
GetPeopleCount	4-88

GetPreviousClockInOutActivities	4-89
GetRFIDEncodingData	4-90
GetSeatingZoneAvailability	4-92
GetTreatmentGroupsForLocation	4-93
GuestSearch	4-94
GetPackageTreatmentAvailability	4-97
GetTreatmentAvailability	4-99
Login	4-100
Logout	4-102
MarkAsNoShow	4-102
MarkAssignmentStatus	4-102
MarkSingleAssignmentStatus	4-103
MessageDelete	4-104
MessageRead	4-104
MessageSend	4-106
MessageSetRead	4-107
OHCCreditCardAddRouting	4-108
OHCCreditCardGetAmount	4-109
OHCCreditCardGuestSearch	4-110
OHCCreditCardTmlTweet	4-111
OHCCreditCardUpdateCardInfo	4-112
PayCrewInvoice	4-114
PurchasePackage	4-115
QueryCrewInvoice	4-116
RegisterRFIDBoardCard	4-117
RegisterRFIDBoardCardQuery	4-118
RetrievePage	4-119
SearchAmenity	4-119
SearchCabin	4-123
SearchCntrAssgn	4-126
SearchComment	4-128
SearchCrashCabin	4-131
SearchGangwaylog	4-131
SearchHKForecast	4-134
SearchHouseKeeping	4-136
SearchPassenger	4-143
SearchPassengerByID	4-144
SearchPayrolls	4-146
SearchSpecialRequest	4-147
SearchWorkOrder	4-150
SpaBooking	4-154

SpaBookingPackageRetail	4-156
SpecialRequestCompleted	4-158
UndoAmenityCompleted	4-158
UndoSpecialRequestCompleted	4-159
UpdateAmenity	4-159
UpdateCabCleanStatus	4-160
UpdateCabState	4-161
UpdateComment	4-162
UpdateCompleteStatus	4-164
UpdateCreditCardInfo	4-164
UpdateCreditLimit	4-167
UpdateCrewPinCode	4-168
UpdateDayNeeded	4-168
UpdateEmail	4-169
UpdateGuestInfo	4-169
UpdateGuestMessage	4-176
UpdateInvoiceStatus	4-176
UpdateLuggageLink	4-177
UpdateMessageStatus	4-178
UpdatePostingStatus	4-179
UpdatePromoterStatus	4-180
UpdateRFIDEncoded	4-180
UpdateSpecialRequest	4-181
UpdateTravelDoc	4-181
UpdateTravelDocAdvance	4-183
UpdateWorkOrder	4-185
UpdateWorkOrderAssignment	4-187
UpdateWorkOrderImage	4-187
UpdateWorkOrderItem	4-188
ValidateCrewPinCode	4-189
Version	4-189
WSFunction	4-190

# Preface

This document provides an explanation and description of each of the exposed functions in the Oracle Hospitality Cruise Shipboard Property Management System (SPMS) Web Service.

## Audience

This document is intended for:

- Third-party vendors who wish to integrate with SPMS.
- This document should be used as a technical reference for external parties to develop applications which requires interfacing to SPMS.

## Customer Support

To contact Oracle Customer Support, access the Customer Support Portal at the following URL:

<https://iccp.custhelp.com>

When contacting Customer Support, please provide the following:

- Product version and program/module name
- Functional and technical description of the problem (include business impact)
- Detailed step-by-step instructions to re-create
- Exact error message received
- Screenshots of each step you take

## Documentation

Oracle Hospitality product documentation is available on the Oracle Help Center at <http://docs.oracle.com/en/industries/hospitality/cruise.html>.

## Revision History

**Table 1 Revision History**

Version	Date	Description of Change
23.1	October 2023	Initial publication.
23.1	November 2023	Updated PostingTotal Class for parameter gsCallNoDialed to String Type.

# 1

## General Overview

The document provides an explanation and description of each of the exposed functions available in OHCWebServices and is used as the Technical Reference by third-party vendors who wish to develop and integrate their applications with the Oracle Hospitality Cruise Shipboard Property Management System (SPMS).

All of the web messages used here are based on:

- Extensible Markup Language (XML) format,
- JavaScript Object Notation (JSON) or
- JSON with Padding (JSONP).

The examples provided here are based on Microsoft Visual Studio 2008. Accessing the web service through other programming language is possible and at your disposition. Oracle Hospitality Cruise will not provide any assistance on this.

## Handling for JSON Format

### Date Type Handling

- All JSON date type must be passed in as JSON string in “YYYYMMDDHHMMSS” format. For example, 2011-01-14 3:14PM should be passed in as “20110114151400”.

### Object Type Handling

- All JSON object must be passed in as serialized JSON string.

### Byte Array Handling

- All JSON ByteArray must be passed in as base64 serialized JSON string.

## Passing Date Variable using XML format

By default, the XML Serialized Date is a format that includes time zone information. For example, if the server is in Florida (GMT-5) and the caller is in Malaysia (GMT+8), the value passed to web service is shown as:

```
2011-12-11T04:30:38.0946974+08:00
```

The server then de-serializes the date to “2011-12-10 3:30pm” as the server in Florida is 13 hours behind the caller in Malaysia.

If you want to preserve the date and time, you must make sure the XML Serialized Date format excludes the time zone info as shown below:

```
2011-12-11T04:30:38.0946974
```

### Method to Handle Date Passing between different Time Zones



For a dataset, the method shown below ensures that the date/time remains the same between different time zones.

```

For Each oTable In oResponse.oDataSet.Tables
  For nX = 0 To oTable.Columns.Count - 1
    If oTable.Columns(nX).DataType.ToString = "System.DateTime" Then
      oTable.Columns(nX).DateTimeMode =DataSetDateTime.Unspecified
    End If
  Next
Next

```

Shown below is the code used to ensure the date field will be serialized without a time zone. This function *does not* support properties with an additional parameter, it also *does not* support multi-dimension array.

```

Private Sub RemoveTimeZone(ByVal poObject As Object)
  Dim oType As System.Reflection.PropertyInfo
  Dim oField As System.Reflection.FieldInfo

  For Each oType In poObject.GetType.GetProperties
    Try
      If oType.PropertyType.IsArray Then
        Call RemoveArrayTimeZone(oType.GetValue(poObject, New Object() {}))
      End If
    Catch ex As Exception
      End Try
  Next

  For Each oField In poObject.GetType.GetFields
    Try
      If oField.FieldType.IsArray Then
        Call RemoveArrayTimeZone(oField.GetValue(poObject))
      ElseIf oField.FieldType.Name = "DateTime" Then
        oField.SetValue(poObject, _
          DateTime.SpecifyKind(oField.GetValue(poObject), _
            New Object() {}), _
            DateTimeKind.Unspecified), _
          New Object() {})
      End If
    Catch ex As Exception
      End Try
  Next
End Sub

Private Sub RemoveArrayTimeZone(ByVal poObject As Object)
  Dim nX As Integer
  If poObject(0).GetType.ToString = "DateTime" Then
    For nX = 0 To poObject.Length - 1
      poObject(nX) = DateTime.SpecifyKind(poObject(nX), _
        DateTimeKind.Unspecified)
    Next
  End If
End Sub

```

```
Next
ElseIf poObject(0).GetType.IsClass = True Then
  For nX = 0 To poObject.Length - 1
    Call RemoveTimeZone(poObject(nX))
  Next
End If
End Sub
```

Shown below is the code used for normal date variable.

```
Dim gdCheckOpenDateTime As Date
gdCheckOpenDateTime = DateTime.SpecifyKind(Now(), _
                                           DateTimeKind.Unspecified)
```

## Connecting to OHC Web Services using Microsoft .NET

The Microsoft IIS is used to host the SOAP Web Service. Thus this document assumes that the reader is familiar with accessing the SOAP Web Service function.

### How to use OHCWebServices with Microsoft .NET

1. Add a **Web Reference** for Web Service in your application. For example, named web reference as **FidelioSPMSWS**.
2. Add a variable reference to the Web Service reference like *Public goWs As New FidelioSPMSWS FidelioSPMSWSSoapClient*
3. With the above in place, you can then call the available web methods. For example:
  - a. goWs.FidelioSPMSWSJsonGet().
  - b. goWs.FidelioSPMSWSJsonPost().
  - c. goWs.FidelioSPMSWSXML().

## Recommended EndPoint Setting

This section describes the recommended EndPoint setting for the application.

### Configure the EndPoint in App.Config

Use Notepad to open the **App.Config** file and edit the address to the correct IP Address of the web server.

```
<client>
  <endpoint address=http://localhost:50844/OHCWebServices/OHCWebServices.asmx
            binding="basicHttpBinding"
            bindingConfiguration="FidelioSPMSWSSoap"
            contract="FidelioSPMSWS.FidelioSPMSWSSoap"
            name="FidelioSPMSWSSoap" />
</client>
```

Below is the recommended setting

```
<bindings>
  <basicHttpBinding>
    <binding name="FidelioSPMSWSSoap"
      closeTimeout="00:01:00"
      openTimeout="00:01:00"
      receiveTimeout="00:10:00"
      sendTimeout="00:01:00"
      allowCookies="false"
      bypassProxyOnLocal="false"
      hostNameComparisonMode="StrongWildcard"
      maxBufferSize="10000000"
      maxBufferPoolSize="524288"
      maxReceivedMessageSize="10000000"
      messageEncoding="Text"
      textEncoding="utf-8"
      transferMode="Buffered"
      useDefaultWebProxy="true">
      <readerQuotas maxDepth="32"
        maxStringLength="1000000"
        maxArrayLength="16384"
        maxBytesPerRead="4096"
        maxNameTableCharCount="16384" />
    </binding>
  </basicHttpBinding>
</bindings>
```

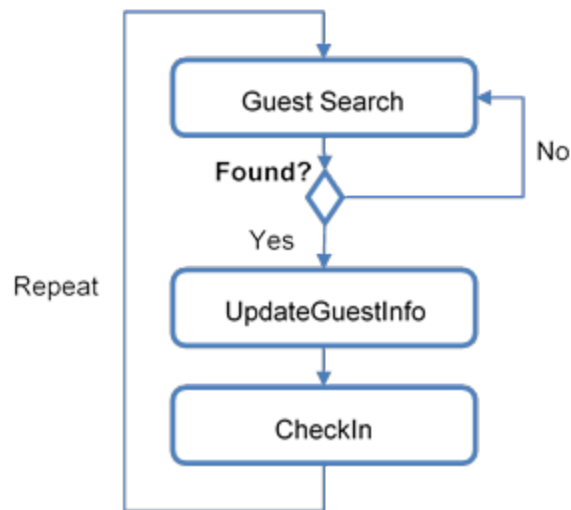
# 2

## General Process Flow

### Check-In Guest

Shown in the diagram below is the general process flow for Guest Check-In.

**Figure 2-1 General Process Flow — Check-In**

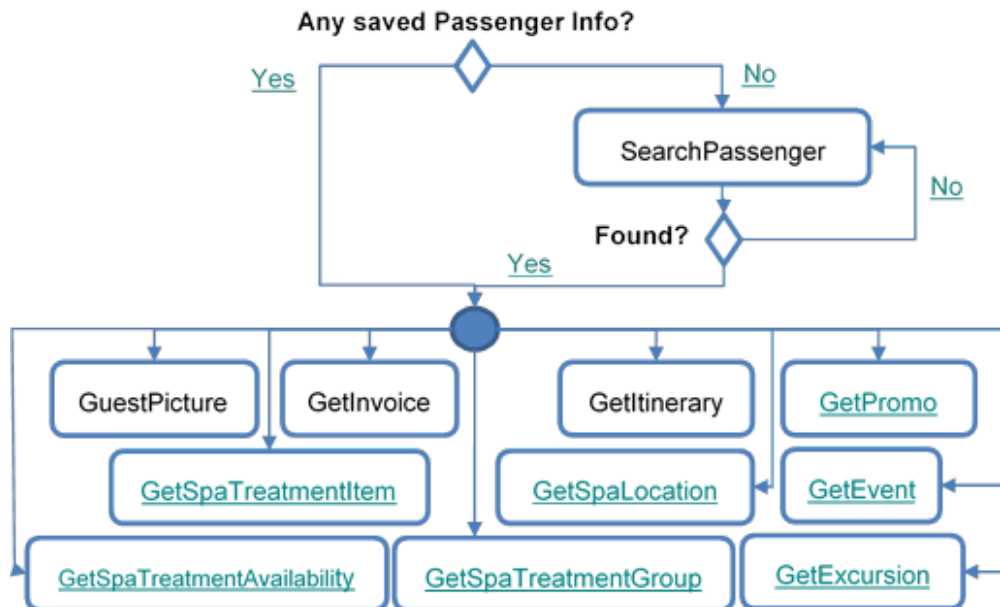


1. First, call the **GuestSearch** function to return a list of guests.
2. Subsequently, call the **UpdateGuestInfo** and **CheckIn** functions for the Check-In of each guest. You must call the **UpdateGuestInfo** function before the **CheckIn** function. Once the **CheckIn** is called, you must call the **GuestSearch** again before calling **UpdateGuestInfo**.

### Get Passenger Information

Shown in the diagram below is the general process flow used to obtain Passenger Information.

Figure 2-2 General Process Flow – Get Passenger Information



1. First, call the **SearchPassenger** function to return one guest.
2. This saves the unique Guest ID into the local device as a cookie for the web so that it does not need to perform the **SearchPassenger** function again.
3. However, it needs to check the expiry date of the saved information against the disembarkation date.
4. You can then use the unique Guest ID to perform the **GuestPicture**, **GetInvoice**, **GetItinerary**, **GetPromo**, **GetSpaTreatmentItem**, **GetSpaLocation**, **GetEvent**, **GetSpaTreatmentAvailability**, **GetSpaTreatmentGroup**, **GetExcursion**, and other functions.

# 3

## Web Methods

### FidelioSPMSWSJsonGet

This function supports JSON GET method and JSONP GET method . It *does not* support JSON POST method.

#### Input Parameters

**Table 3-1 FidelioSPMSWSJsonGet Input Parameters**

Parameter	Type	Description	Required
psFunction	String	Function to call.	Yes
psSessionID	String	Session ID.	Yes
psParam	Array of String	The field value will depend on the Function. Please refer to function detail for each available parameters.	Optional

The following section describes the parameters.

#### psFunction

The 'psFunction' is the function name to be called.

#### psSessionID

The SessionID is a unique ID returned by OHCWebServices after a successful login function. This field is required in all function calls *except* login function.

The session has an idle timeout of between *5 to 15 minutes* from the last function call and value is configurable from the Database Parameter. The default value is 5 minutes.

```
PAR("WebService", "Session Idle Timeout")
```

When a function call and returns a session expiry error, please re-verify by performing another function call. If it again fail for the same error, you should then re-login and retry.

```
Const SESSIONEXPIRY = "Session Expired"  
Const INVALIDSESSIONID = "Invalid Session ID or Session Expiry"
```

When a function call returns similar error to the one shown below, you *must* change the password and retry.

```
Const PLEASECHANGEPASSWORD = "Please change your password first"
```

#### poParam

The field value depends on the Function called. Please refer to the function details for each available parameters.

### Return Values

Return the following Class value. The variable **bSuccess** is used to determine if the function call is successful or unsuccessful. If unsuccessful, **sErrMsg** will populate with the error message. All other fields will depend on the function call.

```
Public Class CResponseJson
    Public bSuccess As Boolean
    Public sErrMsg As String
    Public sTables As String
    Public nTotalPage As Integer
    Public sPackageInfo As String
    Public sCrewInvoice As String
    Public sObj As String
End Class
```

#### Note:

- oDataSet, oPassenger, oGuestInfo, oGuest and oAdvanceLoyalty will be serialized to **sTables**.
- oPackageInfo will be serialized to **sPackageInfo**.
- oCrewInvoice will be serialized to **sCrewInvoice**.
- oObj will be serialized to **sObj** and **sTables**.

If oDataSet, oPassenger, oGu0estInfo, oGuest and oAdvanceLoyalty is 'Nothing', you may ignore them as this is returned for backward compatibility.

### Sample JSONP

The requirement is that you must pass in the *callback* and *format=json* parameter.

```
GET /OHCWebServices/OHCWebServices.asmx/FidelioSPMSWSJsonGet?
callback="Testing"&psFunction='WSFunction'&psSessionID='SessionID'&psPa
ram=['GetCabin','05002']&format=json HTTP/1.0
```

Host: localhost

Or

```
https://localhost/OHCWebServices/OHCWebServices.asmx/
FidelioSPMSWSJsonGet?
callback="Testing"&psFunction='WSFunction'&psSessionID='SessionID'&psPa
ram=['GetCabin','05002']&format=json
```

Below are sample responses that return two tables.

**Note:** The Testing() is a JSONP response.

```
Testing(
  {
    "bSuccess":true,
    "sErrMsg":"","
    "sTables":{"
      "Table1": {
        "CAB_ID": "05002",
        "CAB_DECK": "D05",
        "DECK_DESC": "DECK 05",
        "CAB_DESIGN": "P",
        "CAB_TYPE": "CD4",
        "TYPE_DESC": "CD4 - O/C Porthole (D4, D5, D6)",
        "CAB_STATION": "B",
        "MST_NAME": "ASSEMBLY STATION B",
        "CAB_MODDATE": "2011-10-27T10:26:43"
      },
      "Table2": [
        {
          "TYP_ART": "D00",
          "TYP_COMMENT": "SEAT 1"
        },
        {
          "TYP_ART": "D02",
          "TYP_COMMENT": "DECK 02"
        },
        {
          "TYP_ART": "D03",
          "TYP_COMMENT": "DECK 03"
        },
        {
          "TYP_ART": "D04",
          "TYP_COMMENT": "DECK 04"
        },
        {
          "TYP_ART": "D05",
          "TYP_COMMENT": "DECK 05"
        },
        {
          "TYP_ART": "D06",
          "TYP_COMMENT": "DECK 06"
        },
        {
          "TYP_ART": "D07",
          "TYP_COMMENT": "DECK 07"
        },
        {
          "TYP_ART": "D08",
          "TYP_COMMENT": "DECK 08"
        },
        {
          "TYP_ART": "D09",
          "TYP_COMMENT": "DECK 09"
        }
      ]
    }
  }
)
```



```

        {
            "TYP_ART": "D10  ",
            "TYP_COMMENT": "DECK 10"
        },
        {
            "TYP_ART": "D11  ",
            "TYP_COMMENT": "DECK 11"
        },
        {
            "TYP_ART": "D12  ",
            "TYP_COMMENT": "DECK 12"
        },
        {
            "TYP_ART": "D88  ",
            "TYP_COMMENT": "PAX TEMP CABIN"
        },
        {
            "TYP_ART": "D99  ",
            "TYP_COMMENT": "CREW TEMP CABIN"
        }
    ]
}
);

```

### Sample JSON

The requirement is you must pass in the *format=json* parameter

```

GET /OHCWebServices/OHCWebServices.asmx/FidelioSPMSWSJsonGet?
psFunction='WSFunction'&psSessionID='SessionID'&psParam=['GetCabin','05
002']&format=json HTTP/1.0
Host: localhost
Or
http://localhost/OHCWebServices/OHCWebServices.asmx/
FidelioSPMSWSJsonGet?
psFunction='WSFunction'&psSessionID='SessionID'&psParam=['GetCabin','05
002']&format=json

```

Below are the sample response that returns two tables, and the text in red represents the second table.

```

{
    "bSuccess":true,
    "sErrMsg":"","
    "sTables":{"Table1": {
        "CAB_ID": "05002",
        "CAB_DECK": "D05  ",
        "DECK_DESC": "DECK 05",
        "CAB_DESIGN": "P",
        "CAB_TYPE": "CD4  ",
        "TYPE_DESC": "CD4 - O/C Porthole (D4, D5,
D6)",
        "CAB_STATION": "B  ",

```

```
        "MST_NAME": "ASSEMBLY STATION B",
        "CAB_MODDATE": "2011-10-27T10:26:43"
    },
    "Table2": [
        {
            "TYP_ART": "D00  ",
            "TYP_COMMENT": "SEAT 1"
        },
        {
            "TYP_ART": "D02  ",
            "TYP_COMMENT": "DECK 02"
        },
        {
            "TYP_ART": "D03  ",
            "TYP_COMMENT": "DECK 03"
        },
        {
            "TYP_ART": "D04  ",
            "TYP_COMMENT": "DECK 04"
        },
        {
            "TYP_ART": "D05  ",
            "TYP_COMMENT": "DECK 05"
        },
        {
            "TYP_ART": "D06  ",
            "TYP_COMMENT": "DECK 06"
        },
        {
            "TYP_ART": "D07  ",
            "TYP_COMMENT": "DECK 07"
        },
        {
            "TYP_ART": "D08  ",
            "TYP_COMMENT": "DECK 08"
        },
        {
            "TYP_ART": "D09  ",
            "TYP_COMMENT": "DECK 09"
        },
        {
            "TYP_ART": "D10  ",
            "TYP_COMMENT": "DECK 10"
        },
        {
            "TYP_ART": "D11  ",
            "TYP_COMMENT": "DECK 11"
        },
        {
            "TYP_ART": "D12  ",
            "TYP_COMMENT": "DECK 12"
        },
        {
            "TYP_ART": "D88  ",
            "TYP_COMMENT": "PAX TEMP CABIN"
        }
    ]
}
```

```

    },
    {
      "TYP_ART": "D99  ",
      "TYP_COMMENT": "CREW TEMP CABIN"
    }
  ]
}"
}

```

## FidelioSPMSWSJsonPost

This function supports JSON POST method only. It does not support JSON GET method and JSONP GET method.

**Table 3-2 FidelioSPMSWSJsonPost Input Parameters**

Parameter	Type	Description	Required
psFunction	String	Function to call.	Yes
psSessionID	String	Session ID.	Yes
psParam	Array of String	The field value will depend on the Function. Please refer to function detail for each available parameters.	Optional

Below are the descriptions of the parameters.

### psFunction

The 'psFunction' is the function name to be called.

### psSessionID

The SessionID is a unique ID returned by SPMS OHCWebServices after a successful login function. This field is required for all function calls *except* login function.

The session has an idle timeout of between 5 to 15 *minutes* from the last function call and value is configurable from the Database Parameter. The default value is 5 minutes.

```
PAR("WebService", "Session Idle Timeout")
```

When a function call returns a session expiry error, then please re-verify by performing another function call. If it again fails for the same error, you should then re-login and retry.

```
Const SESSIONEXPIRY = "Session Expired"
```

```
Const INVALIDSESSIONID = "Invalid Session ID or Session Expiry"
```

When you call a function and it returns the error below, you *must* change the password and retry.

```
Const PLEASECHANGEPASSWORD = "Please change your password first"
```

## poParam

The field value depends on the function called. Please refer to the function details for each available parameters.

## Return Values

Return the following Class value. The variable **bSuccess** is used to determine if the function call is successful or unsuccessful. If unsuccessful, **sErrMsg** will populate with the error message. All other fields will depend on the function call.

```

Public Class CResponseJson
    Public bSuccess As Boolean
    Public sErrMsg As String
    Public sTables As String
    Public nTotalPage As Integer
    Public sPackageInfo As String
    Public sCrewInvoice As String
    Public sObj As String
End Class

```



### Note:

- oDataSet, oPassenger, oGuestInfo, oGuest and oAdvanceLoyalty will be serialized to **sTables**.
- oPackageInfo will be serialized to **sPackageInfo**.
- oCrewInvoice will be serialized to **sCrewInvoice**.
- oObj will be serialized to **sObj** and **sTables**.

## Sample JSON

The requirement is that you must pass in the

```

Content-Type: application/json; charset=utf-8
POST /OHCWebServices/OHCWebServices.asmx/FidelioSPMSWSJsonPost
HTTP/1.0 Host: localhost
Content-Type: application/json; charset=utf-8
Content-Length: 69
{psFunction:'WSFunction',psSessionID:'SessionID',psParam:
['GetCabin','05002']}

```

Below is the sample response that return two tables.

```

{
  "bSuccess":true,
  "sErrMsg":"",
  "sTables":{"Table1": {
    "CAB_ID": "05002",
    "CAB_DECK": "D05"
  },

```

```
    "DECK_DESC": "DECK 05",
    "CAB_DESIGN": "P",
    "CAB_TYPE": "CD4  ",
    "TYPE_DESC": "CD4 - O/C Porthole (D4, D5,
D6)",
    "CAB_STATION": "B  ",
    "MST_NAME": "ASSEMBLY STATION B",
    "CAB_MODDATE": "2011-10-27T10:26:43"
  },
  "Table2": [
    {
      "TYP_ART": "D00  ",
      "TYP_COMMENT": "SEAT 1"
    },
    {
      "TYP_ART": "D02  ",
      "TYP_COMMENT": "DECK 02"
    },
    {
      "TYP_ART": "D03  ",
      "TYP_COMMENT": "DECK 03"
    },
    {
      "TYP_ART": "D04  ",
      "TYP_COMMENT": "DECK 04"
    },
    {
      "TYP_ART": "D05  ",
      "TYP_COMMENT": "DECK 05"
    },
    {
      "TYP_ART": "D06  ",
      "TYP_COMMENT": "DECK 06"
    },
    {
      "TYP_ART": "D07  ",
      "TYP_COMMENT": "DECK 07"
    },
    {
      "TYP_ART": "D08  ",
      "TYP_COMMENT": "DECK 08"
    },
    {
      "TYP_ART": "D09  ",
      "TYP_COMMENT": "DECK 09"
    },
    {
      "TYP_ART": "D10  ",
      "TYP_COMMENT": "DECK 10"
    },
    {
      "TYP_ART": "D11  ",
      "TYP_COMMENT": "DECK 11"
    },
    {
```

```

        "TYP_ART": "D12  ",
        "TYP_COMMENT": "DECK 12"
    },
    {
        "TYP_ART": "D88  ",
        "TYP_COMMENT": "PAX TEMP CABIN"
    },
    {
        "TYP_ART": "D99  ",
        "TYP_COMMENT": "CREW TEMP CABIN"
    }
]
}"
}

```

## FidelioSPMSWSXML

This function only supports Simple Object Access Protocol (SOAP).

**Table 3-3 FidelioSPMSWSXML Input Parameters**

Parameter	Type	Description	Required
psFunction	String	Function to call.	Yes
psSessionID	String	Session ID.	Yes
poParam	Array of Object	The field value will depend on the Function. Please refer to function detail for each available parameters.	Optional

### psFunction

The 'psFunction' is the function name to be called.

### psSessionID

The SessionID is a unique ID returned by SPMS OHCWebServices after a successful login function. This field is required in all function calls *except* login function.

The session has an idle timeout of between 5 to 15 *minutes* from the last function call and value is configurable from the Database Parameter. The default value is 5 minutes..

```
PAR("WebService", "Session Idle Timeout")
```

When a function call returns a session expiry error, then please re-verify by performing another function call. If it again fails for the same error, you should then re-login and retry.

```
Const SESSIONEXPIRY = "Session Expired"
```

```
Const INVALIDSESSIONID = "Invalid Session ID or Session Expiry"
```

When a function call returns similar error to the one shown below, you *must* change the password and retry.

```
Const PLEASECHANGEPASSWORD = "Please change your password first"
```

Do make sure you change the *app.config* in order for this to work.

```
<readerQuotas maxDepth="32" maxStringContentLength="100000"  
maxArrayLength="16384"
```

### poParam

The field value depends on the function called. Please refer to the function details for each available parameters.

### Return Values

Return the below Class value. The variable **bSuccess** is used to determine if the function call is successful or fails. If fail, **sErrMsg** will be populated with the error message. All other fields depend on the function call.

```
Public Class CResponseXml  
    Public bSuccess As Boolean  
    Public sErrMsg As String  
    Public oTables() As Data Table  
    Public oPassenger As CPassenger  
    Public oGuestInfo() As CiPos.iGuestInfo  
    Public oObj As Object  
    Public oPeopleCount As CFCSecurity.PeopleCount  
    Public oAdvanceLoyaltyDiscount As CiPos.CAdvanceLoyaltyDiscount  
    Public nTotalPage As Integer  
    Public oCreditCard As COHCCreditCard  
    Public oTerminal As TerminalNotification  
End Class
```

### Sample XML

```
Public goWS As New FidelioSPMSWS.FidelioSPMSWSSoapClient  
  
Dim oResponse As FidelioSPMSWS.CResponseXml  
  
oResponse = goWS.FidelioSPMSWSXML("WSFunction", _  
    "Session", _  
    New Object() {"GetCabin", _  
        "05002"})
```

# 4

## Functions

The following sections describe the OHCWebservices functions with their required parameters and return values.

### ActivateCreditCard

This function activates the credit card of the guest.

#### Input Parameters

psFunction="DeactivateCreditCard." – below is the list of psParam or poParam.

**Table 4-1 psParam**

Parameter	Type	Description	Required
pnCrdID	Integer	Unique Credit Card ID.	Yes
pbAssignInvoice1	Boolean	Assign the Credit Card to Invoice 1.	Yes
pbAssignInvoice2	Boolean	Assign the Credit Card to Invoice 2.	Yes
pbAssignInvoice3	Boolean	Assign the Credit Card to Invoice 3.	Yes
pbAssignInvoice4	Boolean	Assign the Credit Card to Invoice 4.	Yes

#### Return Values

CRResponseXml.bSuccess = False when Update failed.

CRResponseXml.bSuccess = True when Update successful.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("ActivateCreditCard", "SessionID", New  
Object() {12345, True, True, True, True})
```

### AddHKTask

This function adds a new housekeeping task created externally into the SPMS database.

**Note:** Not supported when it is connected to an offline QCI Sync database.

#### Input Parameters

psFunction="AddHKTask" — below is the list of psParam or poParam



**Table 4-2 psParam**

Parameter	Type	Description	Required
pdTaskDate	Date	Housekeeping Task Date.	Yes
psCabin	String	Cabin Number.	Yes
psLocation	String	Location	Yes
pnStatus	Integer	0-Pending, 1-Started, 2-Completed.	Yes
pnCrewID	Integer	Task assigned to this Crew ID.	Yes
pnTaskID	Integer	Housekeeping Task ID.	Yes
pnPortID	Integer	Port ID.	Yes
psComment	String	Task Comments.	

**Return Values**

CResponseXml.bSuccess = False when AddHKTask has failed.

CResponseXml.bSuccess = True when AddHKTask is successful.

For Example:

```
oResponse = goWS.FidelioSPMSWSXML("AddHKTask", "Session ID", New Object()
{DateTime.SpecifyKind(DEFAULT_DATE, DateTimeKind.Unspecified), <Cabin>,
<Location>, 0, 123, 2, 123, <Comments>})
```

## AddRouting

This function allows you to add routing.

Please take note that we support both: all department route from passenger A to passenger B, and specific department route from passenger A to passenger C.

The individual department routing always take priority

For JSON, the array of string will be a string separated by comman (","), Each string must start and end with a double quotation mark ("").

For example:

"A","B" = 2 Element

Element 1 = A,"

Element 2 = B,

**Input Parameters**

psFunction="AddRouting" — below is the list of psParam or poParam.

**Table 4-3 psParam**

Parameter	Type	Description	Required
pnBuyer	Integer	Unique Buyer Account ID.	Yes
pnPayer	Integer	Unique Payer Account ID.	Yes
pnWindow	Integer	Invoice Window (0 to 3) only.	Yes
psNote	String(40)	Reason of the routing.	Yes
psDept	Array of String(10)	Array of department.	Yes

Below are the descriptions of the parameters.

**pnBuyer**

Unique Buyer Account ID.

**pnPayer**

Unique Payer Account ID.

**pnWindow**

Invoice Window. Valid value 0 to 3 and the default is 0. If the buyer and payer are the same, then the valid value is 1 to 3.

**psNote**

Reason of the routing. The field can be empty.

**psDept**

An array of department. If the routing is all department, please pass in nothing or an empty array. Please note that if the department passed in does not exist, it will skip and returns no error.

**Return Values**

CResponseXml.bSuccess = False when Add Routing failed.

CResponseXml.bSuccess = True when Add Routing successful.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("AddRouting", "SessionID", New Object()
{12345,12346,0,"Testing",New String() {"9011","9012"}})
```

## AdvanceGuestSearch

This function searches for a guest with various criteria from the SPMS system

**Input Parameters**

psFunction="AdvanceGuestSearch"— below is the list of psParam or poParam.

**Table 4-4 psParam**

Parameter	Type	Description	Required
psCabCategory	String	Cabin Category.	Yes
psNationalty	String	Nationality.	Yes
psVipStatus	String	Guest Classification.	Yes
psPosDiscountLevel	String	Loyalty Discount Level.	Yes
pnGrpID	Integer	Group ID.	Yes
peGuestType	GuestType	Guest Type Enumerator.	Yes
peReservationType	ReservationType	Reservation Type Enumerator.	Yes
pdDate	Date	Embarkation Date.	Yes
Pnscdid	Integer	Current port ID.	Optional
Peonboardstatus	OnboardStatus	0 – all. 1 – onboard. 2 – shoreside.	Optional

Below are the descriptions of the parameters.

**psCabCategory**

Search guests with specified cabin category. Enter a blank to skip this criteria.

**psNationality**

Search guests with specified nationality. Enter a blank to skip this criteria..

**psVipStatus**

Search guests with specified classification. Enter a blank to skip this criteria.

**psPosDiscountLevel**

Search guests with specified POS Discount Level. Enter a blank to skip this criteria.

**pnGrpID**

Search guests with specified group. Enter 0 to skip this criteria.

**peGuestType**

(Enumerator)

Guest	0	=>	Search guests matching search criteria
Crew	1	=>	Search crews matching search criteria
Visitor	2	=>	Search visitors matching search criteria
All	3	=>	Search guests, crews, visitors matching search criteria

**peReservationType**

(Enumerator) Reserve = Search only passenger with reservation status. **Note:** Visitors remain at reservation status only.

Reserve	0	=>	Reservation passenger
CheckIn	1	=>	Search only passenger that had check-in
All	2	=>	Search both check-in and reserve passenger
CheckOut	3	=>	Search check-out passenger only
ReserveFCManagement	4	=>	Reservation passenger, for shore viewer
CheckInFCManagement	5	=>	Search only passenger that had check-in, for shore viewer
CheckOutFCManagement	6	=>	Search check-out passenger only, for shore viewer
Cancelled	7	=>	Canceled reservation passenger.

### pdDate

Embarkation date, if you want to search for all guest regardless of the embarkation date, pass in #12:00:00AM#

### Return Values

CResponseXml.bSuccess = False when input values are invalid or no record found.

CResponseXml.bSuccess = True and CResponseXML.oTables(0) = ADO.NET data table when input values are valid. The data table will contain all fields in uxp and res table, see SPMS Data Dictionary for details of the data fields.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("AdvanceGuestSearch", "SessionID", New
Object() {"D04","","",",,0,3,0, #01-01-2011#})
```

## AmenityCompleted

This function updates the amenity status as “Completed.”

**Note:** Not supported when it is connected to an offline QCI Sync database.

### Input Parameters

psFunction=“AmenityCompleted” — below is the list of psParam or poParam.

**Table 4-5 psParam**

Parameter	Type	Description	Required
pnAmeID	Integer	Amenity Unique ID.	Yes

### Return Values

CResponseXml.bSuccess = False when Update failed.

CResponseXml.bSuccess = True when Update successful.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("AmenityCompleted", "SessionID", New
Object() {12334})
```

## ApproveCrewWorkedHoursDaily

This function approves the Crew Worked hours for a specified date.

### Input Parameters

psFunction="ApproveCrewWorkedHoursDaily" — below is the list of psParam or poParam.

**Table 4-6 psParam**

Parameter	Type	Description	Required
pnAcclID	Integer	Unique Crew Account ID.	Yes
pdDayToApprove	Date	Date of the worked hours to be approved.	Yes

### Return Values

CResponseXml.bSuccess = False when Update failed.

CResponseXml.bSuccess = True when Update successful.

### Example:

```
oResponse = goWS.FidelioSPMSWSXML("ApproveCrewWorkedHoursDaily",
"SessionID", New Object() {12345, DateTime.SpecifyKind(DEFAULT_DATE,
DateTimeKind.Unspecified)})
```

## ChangeBoardStatus

This function toggles the onboard/offboard status of the passenger.

### Input Parameters

psFunction="ChangeBoardStatus" — below is the list of psParam or poParam.

**Table 4-7 psParam**

Parameter	Type	Description	Required
pnAcclID	Integer	Unique Account ID.	Yes
pbSwipe	Boolean	True, the information is captured using swipe card, False, the information is manually entered.	Yes
pbResponseYes	Boolean	Default response to confirmation message.	Yes

Below are the descriptions of the parameters.

**pnAccID**

Unique Account ID.

**pbSwipe**

True, the information is captured through a card swipe.

False, the information is manually entered.

**pbResponseYes**

The default response to the confirmation message, including the Defined Security Action message prompt. For example,

- Warning! Resident is NOT expected today. Are you sure you want to check in?
- Do you want to Check-In <Guest Name>?
- This person is on duty and ashore denied, do you want to override the onboard status?
- This person is ashore denied, do you want to override the onboard status?

**Return Values**

CResponseXml.bSuccess = False when Status change failed.

CResponseXml.bSuccess = True when Status change successful.

In both instances, CResponse.oObj returns the guest onboard status 0 – Onboard, 1- Shore Side.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("ChangeBoardStatus", "SessionID", New Object()
{123456, True, True})
```

## ChangePassword

This function changes the user password.

**Input Parameters**

psFunction="ChangePassword" — below is the list of psParam or poParam.

**Table 4-8 psParam**

Parameter	Type	Description	Required
psOldPassword	String	Plain Text Password.	Yes
psNewPassword	String	New Password in plain text.	Yes

**psOldPassword**

Plain text password

**psNewPassword**

Plain text password

### Return Values

CResponseXml.bSuccess = True when change password is successful.

CResponseXml.bSuccess = False when change password has failed.

## CheckCabinAvailable

This function checks the cabin availability for the specific booking date.

### Input Parameters

psFunction="CheckCabinAvailable" — below is the list of psParam or poParam.

**Table 4-9 psParam**

Parameter	Type	Description	Required
psCabin	String	Cabin Number.	Yes
pdEmbDate	Date	Embarkation Date.	Yes
pdDisDate	Date	Disembarkation Date.	Yes
pnAcclD	Array of Integer	Array of Unique Account ID.	Yes

### Return Values

CResponseXml.bSuccess = False when Cabin is not available, the reason will be returned as an error message.

CResponseXml.bSuccess = True when Cabin is available.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("CheckCabinAvailable", "", New Object()
{"05002", DateTime.SpecifyKind(#1/1/2011#,
DateTimeKind.Unspecified), _DateTime.SpecifyKind(#2/1/2011#,
DateTimeKind.Unspecified), 12334})
```

## CheckIn

This function checks in the expected guest into the SPMS system.

### Input Parameters

psFunction="CheckIn" — below is the list of psParam or poParam.

**Table 4-10 psParam**

Parameter	Type	Description	Required
pnAcclD	Integer	Unique Guest ID.	Yes

Below are the descriptions of the parameters.

### pnAccID

This is the SPMS Guest Unique ID; this value can be retrieved using GuestSearch function. The field name is UXP\_A\_ID.

### Return Values

CResponseXml.bSuccess = True when check-in is successful.

CResponseXml.bSuccess = False when check-in has failed.

Returns *True* when check-in is successful.

Returns *False* when check-in has failed.



### Note:

The **UpdateGuestInfo** must be called before the **CheckIn** function. This is because the **CheckIn** function will also update the guest record and may overwrite any previous updates. If a **CheckIn** is called before **UpdateGuestInfo**, another call to **GuestSearch** must be made again before an update is allowed.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("CheckIn", "SessionID", New Object() {12345})
```

## CheckSag

This function allows the client to check for custom Gangway alert information.

### Input Parameters

psFunction="CheckSag" — below is the list of psParam or poParam.

**Table 4-11 psParam**

Parameter	Type	Description	Required
pnAccID	Integer	Unique Account ID.	Yes
pbShoreToShip	Boolean	True-From Shore to Ship, False from Ship to Shore.	Yes

### Return Values

CResponseXml.bSuccess = Always return True

### CResponseXML.oObj(0)

0-Allow Pass

1-Allow Overwrite

2-Denied Pass



**CResponseXML.oObj(1)**

Message to show the guest.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("CheckSag", "SessionID", New Object()
{12345, True})
```

## ClockInMobile

This function allows a clock-in transaction.

### Input Parameters

psFunction="ClockInMobile" — below is the list of psParam or poParam.

**Table 4-12 psParam**

Parameter	Type	Description	Required
pnAccID	Integer	Unique Crew Account ID.	Yes
pnWorkCodeId	Integer	Task Work Code ID. TARS_ID	Yes

### Return Values

CResponseXml.bSuccess = False when Update failed.

CResponseXml.bSuccess = True when Update successful

Example:

```
oResponse = goWS.FidelioSPMSWSXML("ApproveCrewWorkedHours", "SessionID",
New Object() {12345, 2})
```

## ClockInOutGetStatus

This function retrieves the last clock-in/out transaction of the crew.

### Input Parameters

psFunction="ClockInOutGetStatus" — below is the list of psParam or poParam.

**Table 4-13 psParam**

Parameter	Type	Description	Required
psBoardCardNo	String	Crew boardcard number (RES_BOARDCC)	Either psBoardCardNo or pnAccID is required.
pnAccID	Integer	Unique Crew Account ID.	

**Return Values**

CResponseXml.bSuccess = False when Update failed.

CResponseXml.bSuccess = True when Update successful, CResponse.oDataset will have 1 Data Table.

**Table 4-14 ADO.NET Data Table field structure for data table (“CLOCKINOUT\_STATUS”)**

Code	Type	Description
TARS_ID	Integer	Unique task work code ID.
TARS_CODE	String	Task work code.
TARS_NAME	String	Task work name.
TCLK_DATETIMESTAMP	Date	Clock-in/out datetime stamp.
TCLK_IN_OUT	String	'I' for clock-in, 'O' for clock-out.
CRW_NAME	String	Last Name.
CRW_FSTN	String	First Name.
CRW_DIV	String	Department.
CRW_STAFF_POS	String	Position.
RES_CAB	String	Cabin Number.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("ClockInOutGetStatus", "SessionID", New
Object() {"12345", 0})
```

## ClockOutMobile

This function allows a clock-out transaction.

**Input Parameters**

psFunction="ClockOutMobile" — below is the list of psParam or poParam.

**Table 4-15 psParam**

Parameter	Type	Description	Required
pnAcclID	Integer	Unique Crew Account ID.	Yes
pnWorkCodeId	Integer	Task Work Code ID. TARS_ID Optional, if 0 then will use the last clock-in task work code ID.	

**Return Values**

CResponseXml.bSuccess = False when Update failed.

CResponseXml.bSuccess = True when Update successful.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("ApproveCrewWorkedHours", "SessionID",
New Object() {12345, 0})
```

## DeactivateCreditCard

This function deactivates the credit card for the guest.

### Input Parameters

psFunction="DeactivateCreditCard" — below is the list of psParam or poParam.

**Table 4-16** psParam

Parameter	Type	Description	Required
pnCrdID	Integer	Unique Credit Card ID.	Optional
pnAccID	Integer	Unique Account ID.	Optional

Either pnCrdID or pnAccID must be pass in a valid value.

pnCrdID > 0 mean deactivate single credit card record.

pnAccID > 0 mean deactivate all credit card belong to the guest.

If both pnCrdID > 0 and pnAccID > 0, we will deactivate a single credit card record base on pnCrdID.

### Return Values

CResponseXml.bSuccess = False when Update failed.

CResponseXml.bSuccess = True when Update successful.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("DeactivateCreditCard", "SessionID", New
Object() {12345,0})
```

## DeleteAmenity

This function deletes the guest amenity.

**Note:** Not supported when it is connected to an offline QCI Sync database.

### Input Parameters

psFunction= "DeleteAmenity" — below is the list of psParam or poParam.

**Table 4-17** psParam

Parameter	Type	Description	Required
pnAmeID	Integer	Unique Amenity ID.	Yes

**Return Values**

CResponseXml.bSuccess = False when Delete failed.

CResponseXml.bSuccess = True when Delete successful.

**Example:**

```
oResponse = goWS.FidelioSPMSWSXML("DeleteAmenity", "SessionID", New Object()
{12345})
```

## DeleteComment

This function deletes the guest comment.

**Note:** Not supported when it is connected to an offline QCI Sync database.

**Input Parameters**

psFunction= "DeleteComment" — below is the list of psParam or poParam.

**Table 4-18 psParam**

Parameter	Type	Description	Required
pnCplID	Integer	Unique Guest Comment ID.	Optional
pnCplLinkID	Integer	Guest Comment Link ID.	Optional

You must pass in either pnCplID or pnCplLinkID.

If pnCplLinkID > 0 mean it will delete all comments including the resolution for the same link ID.

If pnCplID > 0 then it will only delete the single comment (only the comment will be deleted)

**Return Values**

CResponseXml.bSuccess = False when Delete failed.

CResponseXml.bSuccess = True when Delete successful.

**Example:**

```
oResponse = FidelioWS("DeleteComment", New Object() {1234,0})
```

## DeleteGuestMessage

This function deletes the guest message.

**Note:** Not supported when it is connected to an offline QCI Sync database.

**Input Parameters**

psFunction= "DeleteGuestMessage" — below is the list of psParam or poParam.

**Table 4-19 psParam**

Parameter	Type	Description	Required
pnMsgID	Integer	Unique Message ID.	Yes

**Return Values**

CResponseXml.bSuccess = False when Delete failed.

CResponseXml.bSuccess = True when Delete successful.

**Example:**

```
oResponse = goWS.FidelioSPMSWSXML("DeleteGuestMessage", "SessionID", New
Object() {12345})
```

## DeleteLuggageLink

This function removes the luggage link for the guest.

**Input Parameters**

psFunction= "UpdateRFIDEncoded" — below is the list of psParam or poParam.

**Table 4-20 psParam**

Parameter	Type	Description	Required
psRFIDUID	String(16)	Convert the value to HEX.	Yes

**Return Values**

CResponseXml.bSuccess = False when Update failed.

CResponseXml.bSuccess = True when Update successful.

**Example:**

```
oResponse = goWS.FidelioSPMSWSXML("DeleteLuggageLink", "SessionID", New
Object() {"E01581A07C123B"})
```

## DeleteResolution

This function deletes the guest comment.

**Note:** Not supported when it is connected to an offline QCI Sync database.

**Input Parameters**

psFunction= "DeleteResolution" — below is the list of psParam or poParam.

**Table 4-21 psParam**

Parameter	Type	Description	Required
pnCplLinkID	Integer	Guest Comment Link ID.	Yes

**Return Values**

CResponseXml.bSuccess = False when Delete failed.

CResponseXml.bSuccess = True when Delete successful.

Example:

```
oResponse = FidelioWS("DeleteResolution", New Object() {1243})
```

## DeleteRouting

This function deletes the routing, you can either delete all routing or delete individual department routing. Deleting all routing also deletes all individual department routing.

For JSON, the array of string will be a string separated by “,”, and each string must start and end with a double quotation mark (“”).

For example,

“A,””,”B,” = 2 Element

Element 1 = A,”

Element 2 = B,”

**Input Parameters**

psFunction= “DeleteRouting” — below is the list of psParam or poParam.

**Table 4-22 psParam**

Parameter	Type	Description	Required
psSearchStr	String	Unique Buyer Account ID or Board Card or PGID or Pax ID depends on the peSearchType.	Yes
psDept	Array of String(10)	Array of department.	Yes
peSearchType	Enum	InternalID = 0 'res_acc PGID = 1 'res_pgid BoardCard = 2 'res_boardcc PaxID = 3 'res_referenz	Optional

Below are the descriptions of the parameters.

**psSearchString**

Unique Buyer Account ID or  
Board Card or  
PGID or  
Pax ID depends on the peSearchType

**psDept**

An array of department, if the routing is all departments, please pass in nothing or an empty array. Please note that if the department pass in does not exist, it will just skip; no error will be returned.

**peSearchType**

InternalID = 0 'res\_acc'  
PGID = 1 'res\_pgid'  
BoardCard = 2 'res\_boardcc'  
PaxID = 3 'res\_referenz'

**Return Values**

CResponseXml.bSuccess = False when Delete Routing failed.

CResponseXml.bSuccess = True when Delete Routing successful.

**Example:**

```
oResponse = goWS.FidelioSPMSWSXML("DeleteRouting", "SessionID", New
Object() {12345,2,New String() {"9011","9012"}})
```

## DeleteSpecialRequest

This function deletes the guest amenity.

**Note:** Not supported when it is connected to an offline QCI Sync database.

**Input Parameters**

psFunction= "DeleteSpecialRequest" — below is the list of psParam or poParam.

**Table 4-23 psParam**

Parameter	Type	Description	Required
pnSrtID	Integer	Unique Special Request ID.	Yes

**Return Values**

CResponseXml.bSuccess = False when Delete failed.

CResponseXml.bSuccess = True when Delete successful.

**Example:**

```
oResponse = goWS.FidelioSPMSWSXML("DeleteSpecialRequest", "SessionID", New
Object() {12345})
```

## DeleteTravelDoc

This function deletes the travel document record.

### Input Parameters

psFunction= "DeleteTravelDoc" — below is the list of psParam or poParam.

**Table 4-24 psParam**

Parameter	Type	Description	Required
pnVisID	Integer	Unique Travel Document ID.	Yes

### Return Values

CResponseXml.bSuccess = False when Delete failed.

CResponseXml.bSuccess = True when Delete successful.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("DeleteTravelDoc", "SessionID", New Object()
{12345})
```

## DeleteWorkOrder

This function deletes the maintenance work order.

**Note:** Not supported when it is connected to an offline QCI Sync database.

### Input Parameters

psFunction= "DeleteWorkOrder" — below is the list of psParam or poParam.

**Table 4-25 psParam**

Parameter	Type	Description	Required
pnMwrID	Integer	Unique Work Order ID.	Yes
psReason	String(1000)	Reason to delete.	Yes

### Return Values

CResponseXml.bSuccess = False when Delete failed.

CResponseXml.bSuccess = True when Delete successful.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("DeleteWorkOrder", "SessionID", New Object()
{12345, "Testing"})
```



## DeleteWorkOrderAssignment

This function removes a person/group from the maintenance work order.

**Note:** Not supported when it is connected to an offline QCI Sync database.

### Input Parameters

psFunction= "DeleteWorkOrderAssignment" — below is the list of psParam or poParam.

**Table 4-26 psParam**

Parameter	Type	Description	Required
pnMwaID	Integer	Unique Work Order Assignment ID.	Yes

### Return Values

CResponseXml.bSuccess = False when Delete failed.

CResponseXml.bSuccess = True when Delete successful.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("DeleteWorkOrderAssignment",
"SessionID", New Object() {12345})
```

## DeleteWorkOrderImage

This function deletes the maintenance work order image.

**Note:** Not supported when it is connected to an offline QCI Sync database.

### Input Parameters

psFunction= "DeleteWorkOrderImage" — below is the list of psParam or poParam.

**Table 4-27 psParam**

Parameter	Type	Description	Required
pnMwpID	Integer	Unique Work Order Image ID.	Yes

### Return Values

CResponseXml.bSuccess = False when Delete failed.

CResponseXml.bSuccess = True when Delete successful.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("DeleteWorkOrderImage", "SessionID", New
Object() {12345})
```

## DeleteWorkOrderItem

This function deletes the maintenance work order item.

**Note:** Not supported when it is connected to an offline QCI Sync database.

### Input Parameters

psFunction= "DeleteWorkOrderItem" — below is the list of psParam or poParam.

**Table 4-28 psParam**

Parameter	Type	Description	Required
pnMwiID	Integer	Unique Work Order Item ID.	Yes

### Return Values

CResponseXml.bSuccess = False when Delete failed.

CResponseXml.bSuccess = True when Delete successful.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("DeleteWorkOrderItem", "SessionID", New
Object() {12345})
```

## EvtBooking

This function creates an event booking in SPMS.

**Note:** Not supported when it is connected to an offline QCI Sync database.

### Input Parameters

psFunction= "EvtBooking" — below is the list of psParam or poParam.

**Table 4-29 psParam**

Parameter	Type	Description	Required
pnAcclID	Integer	Unique Guest ID.	Yes
psUniqID	String(30)	Unique Event Booking ID.	Yes
pnEvtID	Integer	Event ID.	Yes
psPriceCategory	String(6)	Price Category Code.	Yes
psPersonGroup	String(6)	Person Group Code.	Yes
pnTicket	Integer	Number of Tickets.	Yes
psNotes	String(50)	Notes (Default Empty).	Optional
psPin	String(4)	Pin Code (Default Empty).	Optional
psSource	String(100)	Booking Source (Default Empty).	Optional

**Table 4-29 (Cont.) psParam**

Parameter	Type	Description	Required
pbForceBooking	Boolean	True = Allow posting even the guest does not have credit limit or posting is disabled, False – Normal posting.	Yes
pblsFixedSeating	Boolean	True-Fixed Seating assignment booking, False – regular free seating booking.	Optional
pnSeatID	Integer	Seat ID.	Optional
pnPriceID	Integer	Price ID.	Optional

**pnAccID**

Unique Guest ID, return from SearchPassenger.

**psUniqID**

The field identifies the booking unique ID within the entire external system. The field is used as a cross-reference to obtain detailed information about all the transactions recorded on a booking.

**pnEvtID**

The field identifies the unique event ID. This ID can be obtained through GetEvent WebMethod. The field is an internal, unique event number.

**psPriceCategory**

The field identifies the Price Category. This Price Category can be obtained through GetEvent WebMethod message. This field is an internal, unique price category code.

**psPersonGroup**

The field identifies the Person Group. This Person Group can be obtained through GetEvent WebMethod message. This field is an internal, unique person group code.

**pnTicket**

The field identifies the number of tickets to book.

**psNotes**

The field identifies the additional information send by third-party.

**psPin**

The field identifies the telephone PIN code of the reservation account. The valid character is 0,1,2,3,4,5,6,7,8,9. Leading zero support depends on SPMS Parameter setting.

**psSource**

The field identifies the source of the booking. For example, Workstation ID, Kiosk ID, etc. or any other details describing the location where the booking was made.

### **pbForceBooking**

True = Allow posting even the guest do not have a credit limit or posting is disabled. This is applicable to check-in guest only.

False = Normal posting

### **pbIsFixedSeating**

True-Fixed Seating assignment booking, False – regular free seating booking.

### **pnSeatID**

The field identifies the unique seat identification number. This ID can be obtained through GetSeatingZoneAvailability WebMethod. The field is an internal, unique seat number.

### **pnPriceID**

The field identifies the unique pricing setup identification number. This ID can be obtained through GetEvent WebMethod. The field is an internal, unique pricing setup number.

### **Return Values**

CResponseXml.bSuccess = False when input values are invalid or no record found.

CResponseXml.bSuccess = True and CResponseXML.oTables(0) = ADO.NET data table when input values are valid.

**Table 4-30 ADO.NET Data Table field structure for data table xxx**

<b>Code</b>	<b>Type</b>	<b>Description</b>
Code	Type	Description.
AccID	Integer	Account Posted.
Balance	Double	Account Balance.
CruiseID	Integer	Posted Cruise ID.

### **AccID**

The account posted can be different from the pnAccID pass in if the pbForceBooking is set to True and the said account does not exist and not allow to do posting the return value will be internal system error account ID.

### **Balance**

The field identifies the balance after the posting is applied. This field can return positive (account is to be invoiced) or negative (account prepaid) values.

### **CruiseID**

The field identifies the posted cruise ID.

**Example:**

```
oResponse = goWS.FidelioSPMSWSXML("EvtBooking", "SESSIONID", New Object()
{12345, 37464, 245457, "123", "A", 1, "", "", "", False})
```

**For fixed seating assignment bookings:**

```
oResponse = goWS.FidelioSPMSWSXML("EvtBooking", "SESSIONID", New Object()
{12345, 37464, 245457, "123", "A", 1, "", "", "", False, True, 802, 762})
```

## ExcBooking

This function creates an excursion booking in SPMS.

**Note:** Not supported when it is connected to an offline QCI Sync database.

### Input Parameters

psFunction= "ExcBooking" — below is the list of psParam or poParam.

**Table 4-31 psParam**

Parameter	Type	Description	Required
pnAcclID	Integer	Unique Guest ID.	Yes
psUniqID	String(30)	Unique Excursion Booking ID.	Yes
pnExclID	Integer	Excursion ID.	Yes
pnAdultTicket	Integer	Number of Adult Tickets.	Yes
pnChildTicket	Integer	Number of Child Tickets.	Yes
pnAdultPromoID	Integer	Adult Promotion ID (Default = 0).	Optional
pnChildPromotID	Integer	Child Promotion ID (Default = 0).	Optional
psLanguage	String(2)	ISO Language Code (Default Empty).	Optional
psNotes	String(200)	Notes (Default Empty).	Optional
psPin	String(4)	Pin Code (Default Empty).	Optional
psSource	String(100)	Booking Source (Default Empty).	Optional
pbForceBooking	Boolean	True = Allow posting even the guest who does not have credit limit or posting is disabled. False – Normal posting.	Yes
pbKiosk	Boolean	Kiosk Flag.	Optional
pbySigImage	Array of Byte	Signature Byte Array.	Optional
pbPercentageDiscount	Boolean	True = Percentage Discount, False = Value Discount	Optional
pnDiscount	Decimal	Discount Percentage Rate or Discount Value depends on pbPercentageDiscount.	Optional
pnOvewriteSalesPrice	Decimal	User define price.	Optional
pnWeight	Decimal	Weight, 0 if overweight is not used, > 0 it will charge overweight depend on the setup.	Optional
pnHeight	Decimal	Height, for information only.	Optional

**Table 4-31 (Cont.) psParam**

Parameter	Type	Description	Required
pbAllowWaitList	Boolean	True = Allow Wait List when shore excursion is fully booked, False = Return error when shore excursion is fully booked.	Optional
pnParticipant	Integer	Unique Guest ID for the person who will take part in the excursion.	Optional

**pnAccID**

Unique Guest ID, return from SearchPassenger.

**psUniqID**

The field identifies the booking unique ID within the entire external system. The field is used as a cross-reference to obtain detailed information about all the transactions recorded on a booking.

**pnExclID**

The field identifies the unique excursion identification number. This ID can be obtained through GetExcursion WebMethod. The field is an internal, unique excursion number.

**pnAdultTicket**

The field identifies the number of adult priced tickets to book.

**pnChildTicket**

The field identifies the number of child priced tickets to book.

**pnAdultPromoID**

The field identifies the adult promotion ID. If this field is ignored, the guest is still entitled to the promotion price if available, or else, the normal price is applied. If the promotion ID is sent and the promotion price no longer available, an error message is returned.

For example, if a promotion price is valid for only one seat, and the guest tries to book two seats:

- With pnAdultPromoID send.  
An error message will return indicating only one seat is available.
- Without pnAdultPromoID send.  
Booking is made with one seat at the promotion price and one seat at the normal price.

**pnChildPromoID**

The field identifies the child promotion ID. If this field is ignored, the guest is still entitled to the promotion price if available, or else, the normal price is applied. If the promotion ID is sent and the promotion price no longer available, an error message returned.

**psLanguage**

The field identifies the language ID of the tour.

**psNotes**

The field identifies the additional information send by third-party.

**psPin**

The field identifies the telephone PIN code of the reservation account. The valid characters are 0,1,2,3,4,5,6,7,8,9. Leading zero support depends on SPMS Parameter setting.

**psSource**

The field identifies the source of the booking. For example, Workstation ID, Kiosk ID, and so on, or any other details describing the location where the booking was made.

**pbForceBooking**

True = Allow posting even the guest who does not have a credit limit or posting is disabled. This is only applicable to checked-in guest only.

False = Normal posting

**pbKiosk**

For future use, currently no effect.

**pbySignature**

This is the signature image in byte array format.

**pbPercentageDiscount**

True = Percentage discount: False = dollar value discount.

**pnDiscount**

If pbPercentageDiscount = True, this is the discount percentage, or else this is the discount value.

**pnOvewriteSalesPrice**

If this is > 0, this price is used to post instead of the price defined in the DB. Use it with care.

**pnWeight**

Pass is 0 if overweight is not used. If more than 0, it will charge overweight depending on the setup.

**pnHeight**

For information only.

**pbAllowWaitList**

True = Allow Wait List when shore excursion is fully booked.

False = Return error when shore excursion is fully booked. (Default value).

If the booking is posted to the waitlist, CResponseXml.oObj(0) returns the Wait List Number. This only happens when pbAllowWaitList = True and the booking is fully booked.

**pnParticipant**

0-pnAccID is the participant.

>0 - Must be a valid unique guest ID, the guest who will participate in the excursion, no validation on the ID.

**Return Values**

CResponseXml.bSuccess = False when input values are invalid or no record found.

CResponseXml.bSuccess = True and CResponseXML.oTables(0) = ADO.NET data table when input values are valid. CResponse.Xml.oObj(0) will be > 0 if the booking is posted to a wait list.

**Table 4-32 ADO.NET Data Table field structure. It will always return one record when the input is valid**

Code	Type	Description
Code	Type	Description.
AccID	Integer	Account Posted.
Balance	Double	Account Balance.
CruiseID	Integer	Posted Cruise ID.

**AccID**

The account posted can be different from the pnAccID pass in if the pbForceBooking is set to True and the said account does not exist and is not allowed to do posting. The return value will be internal system error account ID.

**Balance**

The field identifies the balance after the posting is applied. This field can return positive (account is to be invoiced) or negative (account prepaid) value.

**CruiseID**

The field identifies the posted cruise ID.

**Example:**

```
oResponse = goWS.FidelioSPMSWSXML("ExcBooking", "SESSIONID", New Object()
{12345, 37464, 245457, 1, 0, 0, 0, "", "", "", "", False})
```

## ExcBookingPackage

This function creates an excursion package booking in SPMS.



**Note:** Not supported when it is connected to an offline QCI Sync database.

### Input Parameters

psFunction="ExcBookingPackage" — below is the list of psParam or poParam.

**Table 4-33 psParam**

Parameter	Type	Description	Required
pnAccID	Integer	Unique Guest ID.	Yes
psUniqID	String(30)	Unique Excursion Package Booking ID.	Yes
pnExcPkgID	Integer	Excursion Package ID.	Yes
pnAdultTicket	Integer	Number of Adult Tickets.	Yes
pnChildTicket	Integer	Number of Child Tickets.	Yes
pnAdultPromoID	Integer	Adult Promotion ID (Default = 0).	Optional
pnChildPromotID	Integer	Child Promotion ID (Default = 0).	Optional
psLanguage	String(2)	ISO Language Code (Default Empty)	Optional
psNotes	String(200)	Notes (Default Empty).	Optional
psPin	String(4)	Pin Code (Default Empty).	Optional
psSource	String(100)	Booking Source (Default Empty).	Optional
pbForceBooking	Boolean	True = Allow posting even the guest who does not have credit limit or posting is disabled. False – Normal posting.	Yes
pbKiosk	Boolean	Kiosk Flag.	Optional
pbySigImage	Array of Byte	Signature Byte Array.	Optional
pnWeight	Decimal	Weight, 0 if overweight is not used, > 0 it will charge overweight depend on the setup.	Optional
pnHeight	Decimal	Height, for information only.	Optional
pbAllowWaitList	Boolean	True = Allow Wait List when shore excursion is fully booked, False = Return error when shore excursion is fully booked.	Optional
pnParticipant	Integer	A Unique Guest ID for the person who takes part in the excursion.	Optional

#### pnAccID

Unique Guest ID, return from SearchPassenger.

#### psUniqID

The field identifies the booking unique ID within the entire external system. The field is used as a cross-reference to obtain detailed information about all of the transactions recorded on a booking.

#### pnExcPkgID

The field identifies the unique excursion package ID. This ID can be obtained through GetExcPackage WebMethod. The field is an internal, unique excursion number.

**pnAdultTicket**

The field identifies the number of adult priced tickets to book.

**pnChildTicket**

The field identifies the number of child priced tickets to book.

**pnAdultPromoID**

The field identifies the adult promotion ID. If this field is ignored, the guest is still entitled to the promotion price, if available, or else the normal price is applied. If the promotion ID is sent and the promotion price is no longer available, an error message is returned.

For example, the promotion price is valid for one seat, and the guest tries to book two seats.

- With pnAdultPromoID send  
An error message is returned indicating only one seat is available.
- Without pnAdultPromoID send:  
Booking is made with one seat at the promotion price and one seat at the normal price.

**pnChildPromoID**

The field identifies the child promotion ID. If this field is ignored, the guest is entitled to promotion price, if available, or else the normal price is applied. If promotion ID is sent and promotion price no longer available, an error message is returned.

**psLanguage**

The field identifies the language ID of the tour.

**psNotes**

The field identifies the additional information sent by a third-party.

**psPin**

The field identifies the telephone PIN code of the reservation account. The valid characters are 0,1,2,3,4,5,6,7,8,9. Leading zero support depends on SPMS parameter setting.

**psSource**

The field identifies the source of the booking. For example, Workstation ID, Kiosk ID, and so on or any other details describing the location where the booking was made.

**pbForceBooking**

True = Allow posting even if the guest does not have a credit limit or the posting is disabled. This only applies to checked-in guests.

False = Normal posting.

**pbKiosk**

For future use, current no effect.

**pbySignature**

This is the signature image in byte array format.

**pbPercentageDiscount**

True = Percentage discount.

False = Dollar value discount.

**pnDiscount**

If pbPercentageDiscount = True, this is the discount percentage or else this is the discount value.

**pnOvewriteSalesPrice**

If this is > 0, we will use this price to post instead of the price defined in the database. Use with care.

**pnWeight**

Pass in 0 if overweight is not used. If > 0, it will charge overweight depending on the setup.

**pnHeight**

For information only.

**pbAllowWaitList**

True = Allow Wait List when shore excursion is fully booked.

False = Return error when shore excursion is fully booked. (Default value).

If the booking is posted to the waitlist, CResponseXml.oObj(0) will return the Wait List Number. This only happens when pbAllowWaitList = True and the booking is fully booked.

**pnParticipant**

When the pnParticipant is set to 0 - the guest pnAccID is the participant of this excursion.

When the pnParticipant is not 0 - the pnAccID book the excursion for a participant.

For example, if there are two people joining the excursion, there will be two requests sent to the server and the pnAccID will be the person who pays the bill.

**Return Values**

CResponseXml.bSuccess = False when input values are invalid or no record found.

CResponseXml.bSuccess = True and CResponseXML.oTables(0) = ADO.NET data table when input values are valid. CResponse.Xml.oObj(0) will be > 0 if the booking is posted to wait list.

**Table 4-34 ADO.NET Data Table field structure. Return one record when input is valid**

Code	Type	Description
Balance	Double	Account Balance.
CruiseID	Integer	Posted Cruise ID.

#### Balance

The field identifies the balance after the posting is applied. This field can return positive (account is to be invoiced) or negative (account prepaid) values.

#### CruiseID

The field identifies the posted cruise ID.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("ExcBookingPackage", "SESSIONID", New Object()
{12345, 37464, 245457, 1, 0, 0, 0, "", "", "", "", False})
```

## FCUIFolio

This function returns the guest posting list.

**Note:** Not supported when it is connected to an offline QCI Sync database.

#### Input Parameters

psFunction= "FCUIFolio" — below is the list of psParam or poParam.

**Table 4-35 psParam**

Parameter	Type	Description	Required
pnAcclID	Integer	Unique Account ID.	Yes
psPin	String	Pin Code (Can be blank).	Yes
pbIncludePackage	Boolean	Include Package.	Yes

#### Return Values

CResponseXml.bSuccess = False when failed.

CResponseXml.bSuccess = True when success, CResponse.oDataset will have 1 Data Table.

**Table 4-36 ADO.NET Data Table field structure for data table**

Code	Type	Description
DEP_MAINID	String	Main Department ID.
DEM_COMMENT	String	Main Department Description.
POSDATE	Date	Posting System Date.
POSDDEMFLAG	String	P-Package,C-Credit,D-Debit.
POS_PPP_ID	Integer	Unique Package ID.
POSDEBIT	Decimal	Debit Total.
POSCREDIT	Decimal	Credit Total.
POSCLOSED	String	Always = No.
POSNOPRINT	String	Yes Or No.
POS_WIN	Integer	Invoice Windows.
POS_POSNOTE_I	String	Posting Remark
POS_MIB_BILL_NO	String	Check No.
POS_TRANS_ID	Internal	Internal Transaction ID.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("FCUIFolio", "SessionID", New Object()
{12345, "", True})
```

## FCUIGuestInquiry

This function queries the list of guest from SPMS based on the input parameter. The function is compatible with the FCUI Inquire message.

**Note:** Not supported when it is connected to an offline QCI Sync database.

### Input Parameters

psFunction=" FCUIGuestInquiry" — below is the list of psParam or poParam.

**Table 4-37 psParam**

Parameter	Type	Description	Required
psSearchStr	String	Search String.	Yes
peSearchType	Integer	Search String type.	Yes
peAccountStatus	Integer	Account Status.	Yes
peAccountType	Integer	Account Type.	Yes
pbIncludePicture	Boolean	Include Guest Picture in Return Result.	Yes

Below are the descriptions of the parameters.

### psSearchStr

Entered by the external system operator (manually or through a magnetic card reader) and generally contains any information necessary to identify the account. The field can be an account number, card number, cabin number or person name.

### peSearchType

The field identifies the type of entry in the Account ID field. The following Account ID Entry Types are defined:

0	→	Card identification number (RES_BOARDCC)
1	→	Cabin number or System account (RES_CAB/UXP_A_SYS_ACC)
2	→	Name of a person or company account(UXP_A_NAME_N)
3	→	Internal identification number (UXP_A_ID)
4	→	Group ID number (RES_GRP)
5	→	External identification number (RES_BOOKNR)
6	→	Phone trunk number (CAB_TEL or CAB_TEL2)
7	→	e-mail ID (UXP_H_EMAIL or RES_EMAIL_SHIP)
8	→	Internal ID number(UXP_C_INTERNALID)
9	→	Guest Category(GCT_CODE)
10	→	RFID UID (RFID_ID)
11	→	Manning Number/Safety Number
12	→	Loyalty Number (UXP_A_FRQ_CARDNO)
13	→	Pre-generated Random Number (RES_PGID)
14	→	External ID (UXP_C_EXTERNALID)

### peAccountStatus

The field identifies the account current reservation status. The following status values are defined:

0	→	Reservations
1	→	Checked-In
2	→	Active (Reservations and Checked-in)
3	→	Disembarked
4	→	Disembarked Residents (Owners – Valid only for ResidenSea)
5	→	Checked-In and all Reservation for the current cruise

### peAccountType

The field identifies the account type. The following account types are defined:

If this field is ignored or blank then it will return all types.

0	→	All
1	→	Passenger (Guest)
2	→	Resident (Guest)
3	→	Crew member

4	→	Group
5	→	Visitor
6	→	System account
7	→	Staff
8	→	Gift Card

**pblIncludeIncludePicture**

To include a guest picture in the return result, true will cause the query to take a longer time to return, you can expect a large amount of data to return.

**Return Values**

Returns True when a search is successful.

Returns False when nothing is found.

**Return of GuestInfo When success****CResponse.oGuest**

This is the search result.

When single record is returned, the result will in JSON object format {}.

When there are multiple record returned, it is an Array of GuestInfo class.

Please refer details in GuestInfo Class Details.

**Table 4-38 GuestInfo Class**

Code	Type	Description	Required	SPMS Data Field Reference
gnAccID	Integer	Account ID.	Yes	RES_ACC
geAccountType	Integer	Account Type.	Yes	RES_TAG
gbAllowPosting	Boolean	Account Enabled.	Yes	UXP_A_POSTING
gsName	String	Salutation, Forename and Surname	Yes	UXP_A_SALUT, UXP_A_FSTN, UXP_A_NAME
gsCabin	String	Cabin Number.	Optional	RES_CAB
gdEmbDate	Date	Embarkation Date.	Yes	RES_EMB_E for expected accounts, RES_EMB_A for checked in accounts.
gdDisDate	Date	Disembarkation Date.	Yes	RES_DIS_E for expected & checked in accounts, RES_DIS_A for checked out accounts.
gdBirthDate	Date	Date of birth.	Optional	UXP_H_BIRTHD
gsEMail	String	E-mail address.	Optional	UXP_H_EMAIL
gnBalance	Decimal	Balance	Yes	(UXP_A_POS_DEBIT – UXP_A_POS_CREDIT)

Table 4-38 (Cont.) GuestInfo Class

Code	Type	Description	Required	SPMS Data Field Reference
gsFirstName	String	First Name.	Yes	UXP_A_FSTN
gsLastName	String	Last Name.	Yes	UXP_A_NAME
gsSalutation	String	Salutations	Yes	UXP_A_SALUT
gsTitle	String	Title	Yes	UXP_A_TITEL
gsGender	String	Gender	Yes	UXP_H_SEX
geAgeGroup	Integer	Minor	Yes	Parameter "Adult Min Age" compared to Birthdate.
gsLanguage	String	Language.	Yes	UXP_H_LANG
gsResStatus	String	Reservation Status.	Yes	RES_STATUS translated to E/C/D.
gsGroup	String	Group ID & Description [ID-Description]	Optional	RES_GRP + UXP_A_NAME
gsGroupShoreID	String	Group ID.	Optional	RES_SHOREID (refer to Group)
gsCategory	String	Category ID & Description [ID-Description; ID-Description].	Optional	GCT_CODE + TYP_GCT.TYP_COMMENT
gsAddress	String	Address.	Optional	UXP_A_STREET
gsStreet	String	Street.	Optional	UXP_A_STREET2
gsCity	String	City.	Optional	UXP_A_CITY
gsZip	String	ZIP Code.	Optional	UXP_A_ZIP
gsState	String	State.	Optional	UXP_A_STATE
gsCountry	String	Country Code.	Optional	UXP_A_COUNTRY
gsTel	String	Telephone Number.	Optional	UXP_H_TELNO
gnCreditLimit	Decimal	Credit Limit.	Optional	Credit limit as per assigned credit department code, unless individual credit limit is assigned in UXP_A_CRELIM
gsShipEMail	String	Ship's e-Mail ID.	Optional	RES_EMAIL_SHIP
gsPin	String	PIN Code	Optional	RES_PIN_CODE
gsAuthKey	String	Authorization Key.	Optional	Refer to PAR Value for PAR-Interfaces-"AuthKey Criteria".
gsBookNo	String	Booking ID.	Optional	RES_BOOKNR
gsShoreID	String	Shore ID.	Optional	RES_SHOREID
gsVGuestID	String	Unique Guest ID.	Optional	RES_V_GUESTID
gsAward	String	Award Level.	Optional	RES_AWARD_LEVEL
gsBoardCard	String	Onboard Cabin Card.	Optional	RES_BOARDCC



Table 4-38 (Cont.) GuestInfo Class

Code	Type	Description	Required	SPMS Data Field Reference
gnRoutedAcc	Integer	Routed Account ID.	Optional	RES_QROUTE_ACC
gbRoutedEnable	Boolean	Routed Account Enable.	Optional	UXP_A_POSTING(refer to the payer)
gnRouteCreditLimit	Decimal	Routed Account Credit Limit.	Optional	Routed Account Credit limit per assigned credit department code, unless individual credit limit is assigned in UXP_A_CRELIM
gsPassportCountry	String	Passport Nationality.	Optional	UXP_H_PASSPI_COUNTRY
gbyPicture	Byte Array	Guest Picture (Unlimited Length).	Optional	SEC_IMAGE
gsCabinStewardName	String	Cabin Steward Name.	Optional	RES_ACC (linked to CAB_KEEPER)
gsHandicap	String	Handicap Description Separate by ASCII 28.	Yes	RES_HANDICAPCODE
gsHandicapRemark	String	Handicap Remarks.	Yes	RES_HANDIREMARKS
gsMusterStation	String	Muster Station Description.	Yes	MST_NAME
gbOnboard	Boolean	Onboard/ Shoreside status.	Yes	RES_OFFBOARD
gsSafetyNo	String	Safety Number.	Optional	UXP_C_POS
goSafetyMusterStation	Class	Muster Station Class.	Optional	ORG_ASSEMBLY(linked to MST_ID), MST_NAME
goEmergencyMusterStation	Array of Class	Muster Station Class 0 to 5.	Optional	ORG_E_MST1 to ORG_E_MST6(linked to MST_ID), MST_NA<E
gbNoSafetyDrill	Boolean	Excuse From Safety Drill (Yes/ No).	Optional	UXP_C_NODRILL
gePicture	Integer	Picture Source.	Yes	SEC_SOURCE
gdLastPictureDate	Date	Picture Last Modified Date Time.	Optional	SEC_MODDATE
gsFreqCardNo	String	Frequent Traveller Card Number.	Yes	UXP_A_FRQ_CARDNO
gsPriceCategory	String	Price Category.	Yes	RES_PCATEGORY
gsCabinType	String	Cabin Type.	Yes	CAB_TYPE
gnNoneRefundableCredit	Decimal	None Refundable Credit Balance.	Optional	UXP_A_NR_BAL

**Table 4-38 (Cont.) GuestInfo Class**

Code	Type	Description	Required	SPMS Data Field Reference
gnCurrentCruiseID	Integer	Current Cruise ID.	Optional	SCR_ID
gsPassportNo	String	Passport Number.	Optional	UXP_H_PASSNO
gsPaymentDep	String	Default Payment Type.	Optional	RES_DEP_PAY
gsClassification	String	Reservation Classification/Vip Flag.	Optional	RES_CLASSIFICATION
gdSysdate	Date	SPMS System Date.	Yes	sys_date
gdCurrentCruiseStartDate	Date	Current Cruise Start Date.	Yes	scr_a_date of current cruise
gnPGID	Integer	Pregenerated Random ID.	Yes	RES_PGID
gsCruiseItnID	String	Cruise external system identifier.	Yes	SCR_ITN_ID
gsResReference	String	External reservation ID.	Yes	RES_REFERENZ
gsDeck	String	Cabin deck code.	Optional	CAB_DECK
gsExternalID	String	External Crew ID.	Yes	UXP_C_EXTERNALID

**Table 4-39 MusterStation Class**

Code	Type	Description	Required	SPMS Data Field Reference
gsCode	String	Code	Yes	MST_ID
gsName	String	Name	Yes	MST_NAME

**gnAccID**

The field identifies the unique account (person) ID. The field is an internal, unique account (person) number (I type).

**geAccountType**

The field identifies the account type. The following account types are defined:

- |   |   |                   |
|---|---|-------------------|
| 1 | → | Passenger (Guest) |
| 2 | → | Resident (Guest)  |
| 3 | → | Crew member       |
| 4 | → | Group             |
| 5 | → | Visitor           |
| 6 | → | System account    |
| 7 | → | Staff             |

---

8 → Gift Card

**gbAllowPosting**

The field identifies the information if an account is enabled (True) or disabled (False).

**gsName**

The field identifies the salutation, forename, and surname fields (separated with space) of an account (person).

**gsCabin**

The field identifies the cabin number that the account (person) occupies. If the account number is returned (not a person), the cabin number field is omitted.

**gdEmbDate**

The field identifies the date and the time the account (person) is expected or embarked.

**gdDisDate**

The field identifies the date and the time the account (person) is expected to disembark.

**gdBirthDate**

The field identifies the person date of birth. If the account number is returned (not a person) or date of birth is not present, the field is omitted.

**gsEMail**

The field identifies the person's e-mail address. If the account number is returned (not a person) or e-mail is not present, the field is omitted.

**gnBalance**

The field identifies the account (person) current balance. This field can return positive (account is to be invoiced) or negative (account prepaid) value.

**gsFirstName**

The field identifies the forename of an account (person).

**gsLastName**

The field identifies the surname of an account (person).

**gsSalutation**

The field identifies the salutation of an account (person).

**gsTitle**

The field identifies the title of an account (person).

**gsGender**

The field identifies the gender of an account (person).

**geAgeGroup**

0 - Adult

1 - YoungAdult

2 - Minor

**gsLanguage**

The field returns the language of an account (person).

**gsResStatus**

The field identifies the reservation status [E–Expected, C–Checked-In, D-Departed] of an account (person).

**gsGroup**

The field identifies the group ID & the name of the group if a passenger is assigned to a group.

**gsGroupShoreID**

The field identifies the group shore ID of the group if a passenger is assigned to a group.

**gsCategory**

The field identifies the category IDs and the names of the categories if a passenger is assigned any categories.

**gsAddress**

The field identifies the address (Line 1).

**gsStreet**

The field identifies the street (Line 2).

**gsCity**

The field identifies the city.

**gsZip**

The field identifies the zip code.

**gsState**

The field identifies the state.

**gsCountry**

The field identifies the country code.

**gsTel**

The field identifies the telephone number.

**gnCreditLimit**

The field identifies the credit limit set for the person(s).

**gsShipEMail**

The field identifies the Ship's e-Mail ID for the person(s).

**gsPin**

The field identifies the telephone PIN code for the person(s). The valid characters are 0,1,2,3,4,5,6,7,8,9. Leading zero support depends on SPMS Parameter setting.

**gsAuthKey**

The field identifies the authorization key for the person(s).

**gsBookNo**

The field identifies the booking ID.

**gsShoreID**

The field identifies the shore ID.

**gsVGuestID**

The field identifies the unique guest ID.

**gsAward**

The field identifies the guest award level.

**gsBoardCard**

It contains Track 2 field 1 value of the onboard cabin card excluding start sentinel and end sentinel.

For example, The value encode in track 2 is ;9910000657580701=20060810?, we will only send 9910000657580701

**gnRoutedAcc**

The field identifies the routed account ID (payer account ID).

**gbRoutedEnable**

The field identifies the information if the routed account is enabled (1) or disabled (0).

**gnRouteCreditLimit**

The field identifies the credit limit set for the routed person(s) (payer account).

**gsPassportCountry**

The field identifies the guest passport nationality

**gbyPicture**

The field identifies the guest pictures.

**gsCabinStewardName**

The field identifies the housekeeper assign to this cabin

**gsHandicap**

If the guest has handicap, it will be shown. For a non-handicapped guest it will be blank. For example, Wheel Chair Require<FS>Deaf .

**gsHandicapRemark**

This is special remark for the handicapped guest.

**gsMusterStation**

The master station associate with the guest.

**gbOnboard**

True-Onboard, False = Shoreside.

**gsSafetyNo**

Safety Number (only apply to crew).

**goSafetyMusterStation**

Return MusterStation Class with Safety Muster Station Code and Description (only apply to crew).

**goEmergencyMusterStation(5)**

Return MusterStation Class with Emergency Muster Station Code and Description (The emergency type is user define) (only apply to crew).

**gbNoSafetyDrill**

Excuse From Safety Drill (only apply to crew).

**gePicture**

Picture Source

-1 = No Picture

0 = Unknown

1 = Web Cam

2 = RFID Passport

3 = Scan Passport Image

4 = Guest History

### **gdLastPictureDate**

Picture Last Modified Date, due to the size of the picture, it is not efficient to return the picture in FCUIGuestInquiry message, you must compare the last modified date of the picture store in your DB and if the picture had been updated, send WSFunction with GuestPicture and gnAccID as parameter message to get the picture in a separate call.

### **gsFreqCardNo**

Frequent Traveller Card Number.

### **gsPriceCategory**

Booking Price Category (res\_pcategory).

### **gsCabinType**

Cabin Type.

### **gnNoneRefundableCredit**

None Refundable Credit, there is an option set in Web.Config to indicate do not return None Refundable credit if the guest search is x number of more. This is for performance purpose. Default is five.

### **gnCurrentCruiseID**

Current Cruise ID.

### **gsPassportNo**

Passport Number.

### **gsPaymentDep**

Default payment department ID.

### **gsClaasification**

This field identified the reservation classification/VIP Flag.

### **gdSysdate**

Current SPMS System Date.

#### **gdCurrentCruiseStartDate**

Current cruise start date.

#### **gnPGID**

Pregenerated Random Number.

#### **gsCruiseItnID**

Cruise external system identifier.

#### **gsResReference**

External reservation ID.

#### **gsDeck**

Cabin Deck code.

#### **gsExternalID**

External Crew ID.

Example:

This is to search all checked-in guest/crew stay in cabin 05002 and include the picture in the response message.

```
oResponse = goWS.FidelioSPMSWSXML("FCUIGuestInquiry", "SESSIONID", New Object()
{ "05002", 1, 1, 0, True})
```

## FCUIPayment

This function posts the payment to the SPMS account based on the input parameter. This is mainly for generic usage and is compatible with FCUI Payment.

**Note:** Not supported when it is connected to an offline QCI Sync database.

#### **Input Parameters**

psFunction= "FCUIPayment" — below is the list of psParam or poParam.

**Table 4-40 psParam**

Parameter	Type	Description	Required
psSearchStr	String	Search String.	Yes
peSearchType	Integer	Search String type.	Yes
peAccountStatus	Integer	Account Status.	Yes
peAccountType	Integer	Account Type.	Yes
pblIncludePicture	Boolean	Include Guest Picture in Return Result.	Yes



**Table 4-40 (Cont.) psParam**

Parameter	Type	Description	Required
psUniquePostingID	String(20)	Unique Posting ID.	Yes
psOutletID	String(13)	Outlet ID.	Yes
pnValue	Decimal	Posting Total.	Yes
psNote	String	Notes.	Yes
pnInvoiceWin	Integer	Invoice Windows (0 – 3).	Yes

Below are the descriptions of the parameters.

### psSearchStr

Entered by the external system operator manually or through a magnetic card reader and generally contains any information necessary to identify the account. The field can be an account number, card number, cabin number or person name.

### peSearchType

The field identifies the type of entry in the Account ID field. The following Account ID Entry Types are defined:

0	→	Card identification number (RES_BOARDCC)
1	→	Cabin number or System account (RES_CAB/UXP_A_SYS_ACC)
2	→	Name of a person or company account(UXP_A_NAME_N)
3	→	Internal identification number (UXP_A_ID)
4	→	Group ID number (RES_GRP)
5	→	External identification number (RES_BOOKNR)
6	→	Phone trunk number (CAB_TEL or CAB_TEL2)
7	→	e-mail ID (UXP_H_EMAIL or RES_EMAIL_SHIP)
8	→	Internal ID number(UXP_C_INTERNALID)
9	→	Guest Category(GCT_CODE)
10	→	RFID UID (RFID_ID)
11	→	Loyalty Number (UXP_A_FRQ_CARDNO)
12	→	Manning Number/Safety Number
13	→	Pre-generated Random Number (RES_PGID)

### peAccountStatus

The field identifies the current account reservation status. The following status values were defined:

0	→	Reservations
1	→	Checked-In
2	→	Active (Reservations and Checked-in)
3	→	Disembarked

- 4 → Disembarked Residents (Owners – Valid only for ResidenSea)
- 5 → Checked-In and all Reservation for the current cruise

### **peAccountType**

The field identifies the account type. The following account types are defined:

If this field is ignored or blank then it will return all type.

- 0 → All
- 1 → Passenger (Guest)
- 2 → Resident (Guest)
- 3 → Crew member
- 4 → Group
- 5 → Visitor
- 6 → System account
- 7 → Staff
- 8 → Gift Card

### **pbIncludeIncludePicture**

To include a guest picture in the return result, true will cause the query to take a longer time to return and expect a huge amount of data to return.

### **psUniquePostingID**

The field identifies the guest check uniquely within the whole external system. The field is used as a cross-reference to obtain detailed information about all the transactions recorded on a check.

### **psOutletID**

The field identifies the revenue center of the outlet from which the charge originated. The SPMS description equivalent to the Outlet ID will be displayed or printed on a customer invoice.

### **pnValue**

The field identifies the payment amount, for forex this is the forex amount receive.

### **psNote**

The field identifies the additional information send by a third-party system.

### **pnInvoiceWin**

The invoice windows, a valid value is from 0 to 3, any invalid value will default to 0.

### **Return Values**

CResponseXml.oObj(0) Returns 0 when update is successful

CResponseXml.oObj(0) Returns 1 when update has failed

CResponseXml.oObj(0) Returns 2 when the search string return more than 1 persons, it is recommend to use Unique Account ID to make sure it is unique.

## FCUIPosting

This function will post to an SPMS account based on the input parameter, this is mainly for generic usage and is compatible with FCUI Posting.

**Note:** Not supported when it is connected to an offline QCI Sync database.

### Input Parameters

psFunction= "FCUIPosting" — below is the list of psParam or poParam.

**Table 4-41 psParam**

Parameter	Type	Description	Required
psSearchStr	String	Search String.	Yes
peSearchType	Integer	Search String type.	Yes
peAccountStatus	Integer	Account Status.	Yes
peAccountType	Integer	Account Type.	Yes
pblIncludePicture	Boolean	Include Guest Picture in Return Result.	Yes
poPosting	PostingDetails Class	The is the Posting Data.	Yes

Below are the descriptions of the parameters.

### psSearchStr

Entered by the external system operator (manually or through a magnetic card reader) and generally contains any information necessary to identify the account. The field can be an account number, card number, cabin number or person name.

### peSearchType

The field identifies the type of entry in the Account ID field. The following Account ID Entry Types are defined:

- 0 → Card identification number (RES\_BOARDCC)
- 1 → Cabin number or System account (RES\_CAB/UXP\_A\_SYS\_ACC)
- 2 → Name of a person or company account(UXP\_A\_NAME\_N)
- 3 → Internal identification number (UXP\_A\_ID)
- 4 → Group ID number (RES\_GRP)
- 5 → External identification number (RES\_BOOKNR)
- 6 → Phone trunk number (CAB\_TEL or CAB\_TEL2)
- 7 → e-mail ID (UXP\_H\_EMAIL or RES\_EMAIL\_SHIP)
- 8 → Internal ID number(UXP\_C\_INTERNALID)

9	→	Guest Category(GCT_CODE)
10	→	RFID UID (RFID_ID)
11	→	Loyalty Number (UXP_A_FRQ_CARDNO)
12	→	Manning Number/Safety Number
13	→	Pre-generated Random Number (RES_PGID)

### **peAccountStatus**

The field identifies the account current reservation status. The following status values were defined:

0	→	Reservations
1	→	Checked-In
2	→	Active (Reservations and Checked-in)
3	→	Disembarked
4	→	Disembarked Residents (Owners – Valid only for ResidenSea)
5	→	Checked-In and all Reservation for the current cruise

### **peAccountType**

The field identifies the account type. The following account types are defined:

If this field is ignored or blank then it will return all type.

0	→	All
1	→	Passenger (Guest)
2	→	Resident (Guest)
3	→	Crew member
4	→	Group
5	→	Visitor
6	→	System account
7	→	Staff
8	→	Gift Card

### **pbIncludeIncludePicture**

To include a guest picture in the return result, "True" will cause the query to take a longer time to return and expect a huge amount of data to return.

### **poPosting**

Information of the transactions to be posted.

Please refer to PostingDetails Class for more detail.

### **Return Values**

CResponseXml.oObj(0) Returns 0 when update is successful.

CResponseXml.oObj(0) Returns 1 when update has failed.

CResponseXml.oObj(0) Returns 2 when the search string return more than 1 person, it is recommend to use Unique Account ID to make sure it is unique.

### **CResponse.oGuest**

This is the search result; it is an Array of GuestInfo class.

It will have 1 record if return result is 0.

It will have 0 or 1 record if return result is 1 depend on fail reason.

It will more than 1 record if return result is 2.

Please refer detail in GuestInfo Class Details.

CResponse.oObj(1) = Pos Transactions ID, only valid when CResponse.oObj(0) = 0

### **Posting Class**

**Table 4-42 PostingDetails Class**

<b>Code</b>	<b>Type</b>	<b>Description</b>	<b>Required</b>
gbCallPosting	Boolean	Indicate this is call posting.	Yes
gsPin	String	PIN Code.	Yes
gsUniquePostingID	String(20)	Unique Posting ID.	Yes
gsCheckImage	String	Receipt image.	Yes
gbyCheckImage	Byte Array	Receipt image including signature in JPG format.	Yes
gsPosInfo	String	Notes.	Yes
gsItemDetail	String(x)	Refer Item Detail for.	Yes
gbForcePosting	Boolean	True = Allow posting even when the guest has reached his credit limit or posting is disabled. False = Normal posting.	Yes
goPosting	Array of PostingTotal Class	Refer to PostingTotal Class.	Yes

#### **gbCallPosting**

If this is a call posting the behavior will be the same as gbForcePosting with the exception that for a call posting when the guest is not found it will then post to system error account. It will use a different data in the goPosting.

For normal posting, it will return 2 when more than 1 guest found but for call posting, it will post to the first guest.

#### **gsPin**

The field identifies the telephone code of the reservation account. The valid characters are 0,1,2,3,4,5,6,7,8,9. Leading zero support depends on SPMS Parameter setting.

Pass in an empty string if not in use.

**gsUniquePostingID**

The field identifies the bill (check) uniquely within the whole external system. The field is used as a cross-reference to obtain detailed information about all the transactions recorded on a check.

**gsCheckImage**

The field identifies the receipt image that shows the same data as printed on the customer receipt. This should be plain text with CR & LF characters separating each line of the text. It should be formatted to be displayed using non-proportional fonts. Pass in an empty string if not in use.

**gbyCheckImage**

This is the customer receipt which include a digital signature.

You can use the below VB .NET sample code to merge the text base check image with JPG digital signature. Pass in nothing if not in use

```
oImage = MergeCheckAndSignatureImage(TextToImage(gsCheckImage), oSignature)

Private Function MergeCheckAndSignatureImage(ByVal poCheckImage As Image,
ByVal poSignatureImage As System.Drawing.Image) As Image
    MergeCheckAndSignatureImage = Nothing
    Try
        'check Width of check and signature image
        'width
        If Not poCheckImage.Width = poSignatureImage.Width Then
            Dim oNewImage As Image = ResizeImage(poCheckImage, poSignatureImage)
            poSignatureImage.Dispose()
            poSignatureImage = oNewImage
        End If

        Dim oImage As New Bitmap(Math.Max(poCheckImage.Width,
poSignatureImage.Width), poCheckImage.Height + poSignatureImage.Height)

        Dim gGraphic As Graphics = Graphics.FromImage(oImage)

        gGraphic.DrawImage(poCheckImage, New Point(0, 0))
        gGraphic.DrawImage(poSignatureImage, New Point(0, poCheckImage.Height))

        gGraphic.Dispose()
        gGraphic = Nothing
        poCheckImage.Dispose()
        Return oImage

    Catch ex As Exception

    End Try
End Function

'Resize image if check width is larger than signature width ,if the
signature image not same width as check image, the picture seems a bit out
of place (please see, signature.jpg)
```

```

Private Function ResizeImage(ByVal poReferenceImage As Image, ByRef
poTargetImage As Image) As Image
    Dim oCurrentImage As New Bitmap(poTargetImage)
    Dim oNewImage As New Bitmap(poReferenceImage.Width,
poTargetImage.Height)
    Dim oGraphic As Graphics = Graphics.FromImage(oNewImage)

    oGraphic.InterpolationMode =
Drawing2D.InterpolationMode.HighQualityBicubic
    oGraphic.DrawImage(oCurrentImage, New Rectangle(0, 0,
poReferenceImage.Width, poTargetImage.Height), _
        New Rectangle(0, 0, oCurrentImage.Width,
oNewImage.Height), GraphicsUnit.Pixel)
    oGraphic.Dispose()
    oCurrentImage.Dispose()
    Return oNewImage
End Function

Private Function TextToImage(ByVal psText As String) As Image
    Dim oGra As Graphics
    Dim sStr() As String
    Dim oImage As Image

    sStr = Split(psText, vbNewLine)
    oImage = New Bitmap(32 * 7, sStr.Length * 12.2 + 18,
PixelFormat.Format32bppArgb)
    oGra = Graphics.FromImage(oImage)
    oGra.FillRectangle(Brushes.White, 0, 0, CSng(32 * 7),
CSng(sStr.Length * 12.2 + 18))
    oGra.DrawString(psText, New Font("Courier New", 8), Brushes.Black,
0, 30)
    oGra.DrawImage(oImage, New PointF(0, 0))
    TextToImage = oImage
End Function

```

**gsPosInfo**

The field identifies the additional information send by a third-party system.

**gsItemDetail**

In order to maintain compatibility with FCUI, the class object will not be used.

This field identify the items order on the external system. The field contains many sub-field, with each field separated with <FS>. If the external systems do not have the particular field, just pass in the default value indicated in bracket.

**Table 4-43 Item Identifier**

Field	Type	Description
1	Integer	Employee Number (Default 0).
2	String(30)	Employee Name (Default Blank).
3	Integer	Guest Count (Default 0).

**Table 4-43 (Cont.) Item Identifier**

Field	Type	Description
4	Integer	Table Number (Default 0).
5	String(60)	Outlet Name.
6	String	Variable Length field. Each record will have 6 fields and separate by pipe " ", each field will be separated by comma "," Field 1: Item Number (9). Field 2: Item Name X(50). Field 3: Quantity (9). Field 4: Sales Total (9.9). Field 5: 1-Voided, 0-Normal. Field 6: 1-Refunded, 0-Normal.
7	String (12)	Posting Server Information.

For example, Guest A order 1 T-Shirt for 10.00 and 2 Key Chain for 2.50 each

```
1234<FS>Cashier 1<FS>1<FS>0<FS>Gift Shop<FS>205789412,T-Shift,1,10.00,0,0|
205789123,Key Chain,2,5.00,0,0
```

**gbForcePosting**

True = Allow posting even the guest who has reached his/her credit limit or posting is disabled. This only applies to check-in guest and the posting will fail if the guest not found.

**PostingTotal Class**

True = Allow posting even the guest who has reached his credit limit or posting is disabled. This only applies to check-in guest and the posting will fail if the guest not found.

**Table 4-44 PostingTotal Class**

Code	Type	Description	Required
gnServingPeriod	Integer	Serving Period	Yes
gnPostingTotal	Decimal	Posting Total	Yes
gnPostingDiscount	Decimal	Discount Total	Yes
gnPostingServiceCharge	Decimal	Service Charge Total	Yes
gnPostingTip	Decimal	Tip Total	Yes
gnPostingTax	Decimal	Tax Total	Yes
gnInvoiceWin	Integer	Invoice Windows (0 – 3)	Yes
gsOutletID	String(13)	Outlet ID	Yes
gdCallStart	Date	Call Start Date or Time (For Call Accounting Posting)	Yes
gsCallNoDialed	String	Dialed Number (For Call Accounting Posting)	Yes



**Table 4-44 (Cont.) PostingTotal Class**

<b>Code</b>	<b>Type</b>	<b>Description</b>	<b>Required</b>
gnCallDuration	Integer	Call Duration (For Call Accounting Posting)	Yes
gsCallType	String	Call Type (For Call Accounting Posting)	Yes

**gnServingPeriod**

The field identifies the serving period number that is active when the charge is posted.

Pass in 0 if not in use.

**gnPostingTotal**

The field identifies the actual amount (gross) is posted. This value represents value in full (including discount, service charge and tip). This field can have positive (account is debited) or negative (void) value.

**gnPostingDiscount**

The field identifies the total amount of discounts taken against the transaction total amount. This field can have positive (account is debited) or negative (void) value.

**gnPostingServiceCharge**

The field identifies the total amount of any automatic percentage add-on amounts to the transaction total. This field can have positive (account is debited) or negative (void) value.

**gnPostingTip**

The field identifies the total amount of any add-on amounts to the transaction total such as tips, cover charges etc. This value is typically entered manually per customer specification. This field can have positive (account is debited) or negative (void) value.

**gnPostingTax**

The field identifies the total amount of any tax to the transaction total. This field can have positive (account is debited) or negative (void) value.

**gnInvoiceWin**

The invoice windows, a valid value is from 0 to 3, any invalid value will default to 0

**gsOutletID**

The field identifies the revenue center of the outlet from which the charge originated. The FCSPMS description equivalent of the Outlet ID will be displayed or printed on a customer invoice.

**gdCallStart**

The field identifies the date and the time when the phone call started. Full date and time format is required.

### **gsCallNoDialed**

The field identifies the phone number called.

### **gnCallDuration**

The field identifies the phone call duration in seconds.

### **gsCallType**

The field identifies the phone call type. The following types are defined:

VL	→	Line Voice Call
VC	→	Cellular Voice Call
VS	→	Satellite Voice Call
VN	→	Non-chargeable Voice Call
FL	→	Line Fax
FC	→	Cellular Fax
FS	→	Satellite Fax
FN	→	Non-chargeable Fax
DL	→	Line Data Call
DC	→	Cellular Data Call
DS	→	Satellite Data Call
DN	→	Non-chargeable Data Call

### **Example:**

For example, Posting using unique account ID, exclude picture from response message

```
Dim oPosting As New FidelioSPMSWS.PostingDetils
With oPosting
    .gbCallPosting = False
    .gbForcePosting = False
    If File.Exists("c:\jpg.jpg") Then
        .gbyCheckImage = File.ReadAllBytes("c:\jpg.jpg")
    End If
    .gsCheckImage = <Check Image>
    .gsItemDetail = "1234<FS>Cashier <FS>1<FS>0<FS>Gift Shop<FS>2xxxxxxxx|T-
Shift|1|100.00"
    .gsPin = ""
    .gsPosInfo = "Remark"
    .gsUniquePostingID = "<POSTINGID>" 'You have to generate you own unique
number
End With
ReDim oPosting.goPosting(0)

oPosting.goPosting(0) = New FidelioSPMSWS.PostingTotal
With oPosting.goPosting(0)
    .gnInvoiceWin = 0
    .gnPostingDiscount = 0.00 'No Discount
```

```

.gnPostingServiceCharge = 10.00 '10% Service Charge
.gnPostingTax = 5.00 '5% Tax
.gnPostingTip = 0.00
.gnPostingTotal = 115.00
.gnServingPeriod = 0
.gnCallDuration = 0
.gsCallNoDialed = ""
.gsCallType = ""
.gsOutletID = "123" 'This value need to setup in SPMS dep mapping
.gdCallStart = Now()'This is only use for call posting
End With

oResponse = goWS.FidelioSPMSWSXML("FCUIPosting", "SESSIONID", New
Object() { 2358, 3, 1, 0, False, oPosting})

```

## Sample JSON Http Post

For JSON you must serialize the object to string first.

**Figure 4-1 Sample JSON String**

Example using JSON.stringify():

```

<script type="text/language">

// A valid json string
var someObject = {};
someObject.someProperty = "someValue";

// jsonString now contains a JSON string representation of someObject
var jsonString = JSON.stringify(someObject);

// Will display the string '{"someProperty":"someValue"}'
alert(jsonString);

</script>

```

```

POST /OHCWebServices/OHCWebServices.asmx/FidelioSPMSWSJsonPost HTTP/1.0
Host: localhost
Content-Length: 1771

```

```

{callback:"Testing",psSessionID:"ad765829-3fb7-4f1b-9754-587d43a879ab",
psFunction:"FCUIPosting",psParam:
["2440167",3,1,0,true,{"gbCallPosting":false,"gsPin":"","gsUniq
uePostingID":"","1234","gsCheckImage":"INVOICE â€" EXAMPLE COMPANY
\r\n
        RESTAURANT          \r\n\r\n1 CASHIER 1\r\n
\n-----\r\n\r\n1/1          1234   GST 1\r\n
\n
        JUN12'07  6:44PM          \r\n
\n-----\r\n\r\n 1 FRIED RICE      12.34\r\n
\n  DISCOUNT\t\t\t 2.10- \r\n\r\n  SERVICE CHARGE      0.12 \r\n
\n  TIP                1.00 \r\n\r\n  TOTAL PAID          11.36 \
\r\n\r\n  05000\r\n\r\n  XXXN070612184500\r\n\r\n  MANUAL ENTRY\r\n
\n  Cabin/Acc:05000\r\n\r\n  <LAST/FIRST NAME>\r\n\r\n  CABIN
CHARGE          00.00 \r\n\r\n---1168 CLOSED JUN12  6:45PM---\r\n

```

```

\n                                     \r\nSIGN:.....\r\
\n                                     \r\nAll Amounts in US Dollars (USD$)
\n, \gbyCheckImage\":null, \gsPosInfo\": \"\", \gsItemDetail\": \"1234?Cashier
1?1?0?Gift Shop?205789412|T-Shift|1|
12.34\", \gbForcePosting\":false, \goPosting\":
[{\gnServingPeriod\":0, \gnPostingTotal\":10.01, \gnPostingDiscount\":1.01, \
gnPostingServiceCharge\":1.02, \gnPostingTip\":1.03, \gnPostingTax\":1.04, \
gnInvoiceWin\":0, \gsOutletID\": \"264\", \gdCallStart\": \"20130620100105\", \
gsCallNoDialed\": \"12345678\", \gnCallDuration\":159, \gsCallType\": \"Sat\"},
{\gnServingPeriod\":0, \gnPostingTotal\":11.0, \gnPostingDiscount\":1.01, \gn
PostingServiceCharge\":1.02, \gnPostingTip\":1.03, \gnPostingTax\":1.04, \gn
InvoiceWin\":0, \gsOutletID\": \"262\", \gdCallStart\": \"20130620100105\", \gs
CallNoDialed\": \"12345678\", \gnCallDuration\":259, \gsCallType\": \"Sat\"}}]

```

## FCUIUDFUpdate

This function updates the guest custom information.

**Note:** Not supported when it is connected to an offline QCI Sync database.

### Input Parameters

psFunction= "FCUIUDFUpdate" — below is the list of psParam or poParam.

**Table 4-45 psParam**

Parameter	Type	Description	Required
pnAcclID	Integer	Unique Account ID.	Yes
psUDFID	Array of String	Array of UDF_ID.	Yes
psValue	Array of String	Max length 1000 for each value.	Yes

### Return Values

CResponseXml.bSuccess = False when failed.

CResponseXml.bSuccess = True when success.

### Example:

```

oResponse = goWS.FidelioSPMSWSXML("FCUIFolio", "SessionID", New Object() {12345,
New String() {"1","1"}, New String() {"6411", "0123"}})

```

## FCUIUpdatePin

This function updates the guest pin code.

**Note:** Not supported when it is connected to an offline QCI Sync database.

### Input Parameters

psFunction="FCUIUpdatePin" — below is the list of psParam or poParam.

**Table 4-46 psParam**

Parameter	Type	Description	Required
pnAcclID	Integer	Unique Account ID.	Yes
psCabin	String	Cabin Number.	Yes
psAuthKey	String	4 Digits authentication code to prevent unauthorize update.	Yes
psPin	String	Pin Code.	Yes

**Return Values**

CResponseXml.bSuccess = False when failed.

CResponseXml.bSuccess = True when success.

**Example:**

```
oResponse = goWS.FidelioSPMSWSXML("FCUIFolio", "SessionID", New Object()
{12345, "678", "6411", "0123"})
```

## GangwayGuestSearch

This function is used to search the guest by Gangway movement.

**Input Parameters**

psFunction= "GangwayGuestSearch" — below is the list of psParam or poParam.

**Table 4-47 psParam**

Code	Type	Description	Required
pbAshore	Boolean	True=Guest at shoreside, False=Guest Onboard.	Yes
peGuestType	Integer	0-All. 1-Guest Only. 2-Resident Only. 3-Crew Only. 4-Visitor Only. 5-Check-Out but onboard.	Yes
pbShowCurrentPortOnly	Boolean	For ferry operator only, true=only return guest which disembark at the current port only.	Yes
pnExcludeOvernightTour	Boolean	True=Exclude guest stay at shoreside for overnight tour.	Yes

**Return Values**

CResponseXml.bSuccess = False when input values are invalid or no record found.

CResponseXml.bSuccess = True and CResponseXML.oTables(0) = ADO.NET data table when input values are valid.

**Table 4-48 ADO.NET Data Table field structure**

Code	Type	Description
Res_status	String	Reservation Status.
Tname	String	Guest Name.
Res_cab	String	Cabin Number.
Res_owner	Integer	0-Normal Guest, 1-Owner.
Nationality	String	Nationality Code
Res_id	Integer	Unique Reservation ID.
Res_emb_a	Date	Actual Embarkation Date.
Res_signonport	String	Port where the guest embark.
Res_dis_e	Date	Expected Disembarkation Date.
Res_signoffport	String	Port where the guest disembark.
Uxp_a_name	String	Last Name.
Uxp_a_fstn	String	First Name.
Uxp_a_as_crw	Integer	Guest show in crew manifest.
Uxp_c_as_pax	Integer	Crew show in pax manifest.
Res_acc	Integer	Unique Guest ID.
SwitchAcc	Integer	1-Switch Manifest.
ManingNumber	String	Crew Safety Number.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("GangwayGuestSearch", "SESSIONID", New
Object() {False, 0, False, False})
```

## GetAccBalance

This function gets the account outstanding balance.

**Note:** Not supported when it is connected to an offline QCI Sync database.

### Input Parameters

psFunction= "GetAccBalance" — below is the list of psParam or poParam.

**Table 4-49 psParam**

Parameter	Type	Description	Required
pnAccID	Integer	Unique Guest ID.	Yes
pbPreBill	Boolean	True – Get the Pre-Cruise Balance, False – Get the account balance.	Yes

Below are the descriptions of the parameters.

**pnAccID**

Unique Guest ID, return from GuestSearch.

**pbPreBill**

True – Get the Pre-Cruise Balance, False – Get the account balance

**Return Values**

CResponseXml.bSuccess = True when GetAccBalance is successful, CResponse.oObj returns an Array with 2 value, the first value is the Pre-Cruise Balance or Account Balance depending on pbPreBill and the second value is Non-Refundable Credit Pre-Cruise Balance or Non-Refundable Credit Balance.

CResponseXml.bSuccess = False when GetAccBalance has failed

**Example:**

```
oResponse = goWS.FidelioSPMSWSXML("GetAccBalance", "", New Object()
{12345, False})
```

## GetAdvanceLoyaltyDiscount

This function gets the guest loyalty discount eligibility from SPMS.

**Note:** Not supported when it is connected to an offline QCI Sync database.

**Input Parameters**

psFunction=" GetAdvanceLoyaltyDiscount" — below is the list of psParam or poParam.

**Table 4-50 psParam**

Parameter	Type	Description	Required
pnAccID	Integer	Unique Account ID.	Yes
pnRvc	Integer	MICROS POS Revenue Center Number.	Yes

**Return Values**

CResponseXml.bSuccess = False when GetAdvanceLoyaltyDiscount failed

CResponseXml.bSuccess = True when GetAdvanceLoyaltyDiscount is successful and return Class CAdvanceLoyaltyDiscount

**Table 4-51 CAdvanceLoyaltyDiscount Class**

Code	Type	Description	Required
gnDiscNo	Integer	MICROS POS Discount Number.	Yes
gnDiscLimit	Decimal	Maximum Discount Allow Per Check.	Yes

**Table 4-51 (Cont.) CAdvanceLoyaltyDiscount Class**

Code	Type	Description	Required
goEligibility	Array of CDiscRate	Discount eligibility.	Yes

**Table 4-52 CAdvanceLoyaltyDiscount Class**

Code	Type	Description	Required
gsType	String	M-Menu Item ,G-Major Group, F-Family Group.	Yes
gnNumber	Integer	Object Number.	Yes
gnRate	Decimal	Discount Percentage.	Yes

## GetCheckImage

This function gets the check image from SPMS.

**Note:** Not supported when it is connected to an offline QCI Sync database.

### Input Parameters

psFunction= "GetCheckImage" — below is the list of psParam or poParam.

**Table 4-53 psParam**

Parameter	Type	Description	Required
pnMibID	Integer	SPMS Unique Check Image ID.	Yes

Below are the descriptions of the parameters.

### pnMibID

SPMS Unique Check Image ID, return from GetInvoiceTotal.

### Return Values

CResponseXml.bSuccess = True when GetCheckImage is successful

### CResponseXML.oObj(0)

This is the check image from MICROS POS with signature. This is the preferred image to display if both pbyImage and psCheckImage exists.

### CResponseXML.oObj(1)

This is the text image of a guest check.

CResponseXml.bSuccess = False when GetCheckImage has failed or no record found

Example:



```
oResponse = goWS.FidelioSPMSWSXML("GetCheckImage", "SESSIONID", New
Object() {12345678})
```

## GetCreditCardInfo

This function gets the list of register credit card assign to the passenger.

### Input Parameters

psFunction= "GetCreditCardInfo" — below is the list of psParam or poParam.

**Table 4-54 psParam**

Parameter	Type	Description	Required
pnAcclID	Integer	Unique Guest ID.	Yes

Below are the descriptions of the parameters.

### pnAcclID

Unique Guest ID, return from GuestSearch.

### Return Values

CResponseXml.bSuccess = False when GetCreditCardInfo has failed

CResponseXml.bSuccess = True when GetCreditCardInfo is successful,  
CResponse.oDataset will have 2 Data Tables.

**Table 4-55 ADO.NET Data Table field structure for FIRST DATA Credit Card Information**

Code	Type	Description
CRD_NO	String	Credit Card No, the card number will always mask to XX*****XXXX
CRD_EXP	String	Expiry Date in MMY format.
CRD_NAME	String	Credit Card owner name.
CRD_ACTIVE	String	Yes or No.
CRD_LIMITS	String	'No Limit' or the credit limit amount.
CRD_ID	Integer	Unique ID.
CRD_INVOICE	String	'All' or the individual invoice separated by ','
CRD_DCC	String	Y or N for Dynamic Currency Exchange.
CRD_PARCELADO	String	Y or N special Parcelado flag, currently not in use.
CRD_SIG_ID	String	Yes or No, indicate the signature image exists.
DCC_EXRATE	Double	Exchange Rate.
CRD_CUR	String	Currency code.
SIGN_ID	Integer	Signature Unique ID.

**Table 4-56 ADO.NET Data Table field structure for second data table**

Code	Type	Description
CRD_NO	String	Credit Card No, the card number will always mask to XX*****XXXX
CRD_EXP	String	Expiry Date in MMY format.
CRD_NAME	String	Credit Card owner name.
CRD_ID	Integer	Unique ID.
CRD_ACTIVE	String	Yes or No.
CRD_REWARDS	Integer	1 - Reward Card, 2 - Gift Card.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("GetCreditCardInfo", "", New Object() {12345})
```

## GetEvent

This function search and list the available ticket events in SPMS.

**Note:** Not supported when it is connected to an offline QCI Sync database.

### Input Parameters

psFunction="GetEvent" — below is the list of psParam or poParam.

**Table 4-57 psParam**

Parameter	Type	Description	Required
pnAccID	Integer	Unique Guest ID.	Yes
pdSDate	Date	Event start date.	Yes
pdEDate	Date	Event end date.	Yes
peCodeType	EvtCodeType	Code Type.	Yes
psCode	String	Code.	Optional

Below are the descriptions of the parameters.

### pnAccID

Unique Guest ID, return from SearchPassenger.

### pdSDate

The field identifies the minimum date the event can start from.

### pdEDate

The field identifies the maximum date the event must be completed.

### peCodeType

(Enumerator)

The field identifies the type of the code being passed for querying events.

None	0	→	No Code will be used
LocationID	1	→	Event Location ID, you can get the value from WsFunction, GetLocationInfo, LOC_ID.
EventNo	2	→	Event No (EvtNo)
UniqueID	3	→	Event Unique ID (EvtID)

### psCode

The code for the Events to run the query on. It can be either the location code, The Event No (EvtNo), the Unique ID (EvtID).

### Return Values

CResponseXml.bSuccess = False when input values are invalid or no record found.

CResponseXml.bSuccess = True and CResponseXML.oTables(0) = *ADO.NET data table* when input values are valid.

**Table 4-58 ADO.NET Data Table field structure**

Code	Type	Description
EvtID	Integer	Event ID.
EvtNo	String(20)	Event Number.
EvtName	String(50)	Event Name.
EvtBegDate	Date	Starting Date and Time.
EvtEngDate	Date	Ending Date and Time .
EvtLastCall	Date	Last Call Time.
EvtStatus	String(1)	Booking Status.
EvtPriceCategory	String(6)	Price Category Code.
EvtPriceDesc	String(50)	Price Category Description.
EvtPersonCode	String(6)	Person Group Code.
EvtPersonDesc	String(50)	Person Group Description.
EvtIsFixedSeatingEvent	Boolean	True-Fixed Seating Event, False = regular Free Seating Event.
EvtIsLevelHybrid	Boolean	True-Level Hybrid mode.
EvtHybridLevelOpenSeatingCapacity	Integer	Open Seating max capacity for Hybrid Levels.
EvtLayoutLevelCode	String(20)	Layout Level Code.
EvtLayoutLevelDesc	String(50)	Layout Level Description.
EvtSeatingZoneCode	String(20)	Seating Zone Code.
EvtSeatingZoneDesc	String(50)	Seating Zone Description.
EvtIsFixedSeatingPrice	Boolean	True- Fixed Seating Price Setup, False = regular Free Seating Price.
EvtPrice	Double	Price.

**Table 4-58 (Cont.) ADO.NET Data Table field structure**

<b>Code</b>	<b>Type</b>	<b>Description</b>
EvtDesc	String(200)	Event Description.
EvtShowInKiosk	Boolean	Show in Kiosk [External Booking System].
EvtBookInKiosk	Boolean	Book in Kiosk [External Booking System].
EvtMaxSeat	Integer	Maximum Seats.
EvtBooked	Integer	Seats Currently Booked.
EvtLocation	String(50)	Location Name.
EvtShowInITV	Boolean	Show in Internet TV (ITV) - [External Booking System].
EvtBookInITV	Boolean	Book in ITV [External Booking System].
EvtLongDesc	X(1000)	Event Long Description.
EvtPriceID	Integer	Price ID.
EvtLayoutLevelID	Integer	Layout Level ID.
EvtSeatingZoneID	Integer	Seating Zone ID.

**EvtID**

The field identifies the unique event identifier. This ID can be used for event booking. The field is an internal, unique event number.

**EvtNo**

The field identifies the custom, users entered, a number that identifies the event.

**EvtName**

The field identifies the event name.

**EvtBegDate**

The field identifies the event starting date and time.

**EvtEndDate**

The field identifies the event ending date and time.

**EvtLastCall**

The field identifies the last call time, please ignore the date part.

**EvtStatus**

The field identifies the event booking status. The following status identifiers values are defined:

A	→	Available
F	→	Fully booked

O → Overbooked

For Fixed Seating Events, the Booking status applies to the Seating Zone.

### **EvtPriceCategory**

The field identifies the event price category code, this field is required during booking.

For example, Gold Class, Silver Class

### **EvtPriceDesc**

The field identifies the event price category description.

### **EvtPersonCode**

The field identifies the person group code, this field is required during booking. For example, Adult, Child or Senior

### **EvtPersonDesc**

The field identifies the person group description

### **EvtIsFixedSeatingEvent**

True = Fixed Seating Event, False = regular Free Seating Event

### **EvtIsLevelHybrid**

True = Level Hybrid mode (Fixed Seating plus Open Seating Capacity), False = Level is only Fixed Seating

The field identifies the layout level description for fixed seating assignment bookings.

**EvtHybridLevelOpenSeatingCapacity** Open Seating max attendees capacity for Hybrid Levels

**EvtLayoutLevelCode** The field identifies the layout level code for fixed seating assignment bookings.

### **EvtLayoutLevelDesc**

### **EvtSeatingZoneCode**

The field identifies the seating zone code for fixed seating assignment bookings.

### **EvtSeatingZoneDesc**

The field identifies the seating zone description for fixed seating assignment bookings.

### **EvtIsFixedSeatingPrice**

True = Fixed Seating Price Setup, False = regular Free Seating Price

**EvtPrice** The field identifies the event selling price.

**EvtDesc** The field identifies the additional information about the event.

#### **EvtShowInKiosk**

The flag to denote if the event can be shown on the external system.

#### **EvtBookInKiosk**

The flag to denote if the event can be booked on the external system.

#### **EvtMaxSeat**

Total seats available on the event.

**Note:** For the same event ID the same seat could be sold at a different price depending on the price category and person group. The total seats appear on different price category and person group is representation of the same seats as shown in below example.

Below table means the event ID=1 only have 50 seats available, for the same EID will always have the same MAX value.

EID	PCT	PGT	MAX
1	Gold	Adult	50
1	Silver	Adult	50
1	Gold	Child	50
1	Silver	Child	50

For Fixed Seating Events, the total seat count applies to the Seating Zone.

#### **EvtBooked**

The field indicates the total event seat booked.

For Fixed Seating Events, the total seats booked count applies to the Seating Zone.

#### **EvtLocation**

The field identifies the location name of the event.

#### **EvtShowInITV**

The flag to denote if the event can be shown on ITV the external system.

#### **EvtBookInITV**

The flag to denote if the event can be booked on the ITV external system, the field returns 'Yes' or 'No' only.

#### **EvtLongDesc**

The field indicates the Event Long Description intended for ITV usage.

**EvtPriceID**

The field identifies the unique pricing setup identification number. This ID must be used for event fixed seating booking. The field is an internal, unique pricing setup number.

**EvtLayoutLevelID**

The field identifies the unique Layout Level identification number. This ID must be used for event fixed seating booking. The field is an internal, unique layout level number.

**EvtSeatingZoneID**

The field identifies the unique Seating Zone identification number. This ID must be used for event fixed seating booking. The field is an internal, unique seating zone number.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("GetEvent", "SESSIONID", New Object()
{12345, #1/1/2011#, #1/12/2011#, 0, ""})
```

## GetEvtFolio

This function is used to request a list of event booked on a specific account in SPMS.

**Note:** Not supported when it is connected to an offline QCI Sync database.

**Input Parameters**

psFunction= "GetEvtFolio" — below is the list of psParam or poParam.

**Table 4-59 psParam**

Parameter	Type	Description	Required
pnAccID	Integer	Unique Guest ID.	Yes

Below are the descriptions of the parameters.

**pnAccID**

Unique Guest ID, return from SearchPassenger.

**Return Values**

CResponseXml.bSuccess = False when input values are invalid or no record found.

CResponseXml.bSuccess = True and CResponseXML.oTables(0) = *ADO.NET data table* when input values are valid.

**Table 4-60 ADO.NET Data Table field structure**

Code	Type	Description
EvtID	Integer	Event ID.

**Table 4-60 (Cont.) ADO.NET Data Table field structure**

<b>Code</b>	<b>Type</b>	<b>Description</b>
EvtNo	String(20)	Event Number.
EvtName	String(50)	Event Name.
EvtBegDate	Date	Starting Date and Time.
EvtEngDate	Date	Ending Date and Time.
EvtStatus	String(1)	Booking Status.
EvtBookDate	Date	Booking Date and Time.
EvtPriceCategory	String(6)	Price Category Code.
EvtPriceDesc	String(50)	Price Category Description.
EvtPersonCode	String(6)	Person Group Code.
EvtPersonDesc	String(50)	Person Group Description.
EvtLayoutLevelCode	String(20)	Layout Level Code.
EvtLayoutLevelDesc	String(50)	Layout Level Description.
EvtSeatingZoneCode	String(20)	Seating Zone Code.
EvtSeatingZoneDesc	String(50)	Seating Zone Description.
EvtSeatCode	String(20)	Seat Code.
EvtLocation	String(50)	Location Name.
EvtQty	Integer	Number of Tickets Booked.
EvtTotal	Double	Booking Value.
EvtNotes	String(200)	Notes.
EvtPrint	Integer	Ticket Printed Count.
EvtDiscount	Double	Discount Amount.
EvtControl	Integer	Internal Booking Number.
EvtPrepaid	Boolean	Prepaid.

**EvtID**

The field identifies the unique event identifier. This ID can be used for event booking. The field is an internal, unique event number.

**EvtNo**

The field identifies the custom, users entered, a number that identifies the event.

**EvtName**

The field identifies the event name description.

**EvtBegDate**

The field identifies the event starting date and time.

**EvtEndDate**



The field identifies the event ending date and time.

### **EvtStatus**

The field identifies the event booking status. The following status identifiers values are defined:

A	→	Available
F	→	Fully booked
O	→	Overbooked

### **EvtBookDate**

The field identifies the excursion booking date and time (real date and time).

### **EvtPriceCategory**

The field identifies the event price category code; this field is required during booking. For example, Gold Class or Silver Class

### **EvtPriceDesc**

The field identifies the event price category description.

### **EvtPersonCode**

The field identifies the person group code, this field is required during booking. For example, Adult, Child or Senior

### **EvtPersonDesc**

The field identifies the person group description.

### **EvtLayoutLevelCode**

The field identifies the layout level code for fixed seating assignment bookings.

### **EvtLayoutLevelDesc**

The field identifies the layout level description for fixed seating assignment bookings.

### **EvtSeatingZoneCode**

The field identifies the seating zone code for fixed seating assignment bookings.

### **EvtSeatingZoneDesc**

The field identifies the seating zone description for fixed seating assignment bookings.

### **EvtSeatCode**

The field identifies the seat ID, For example, B15 for fixed seating assignment bookings.

**EvtLocation**

The field identifies the location name of the event.

**EvtQty**

The field identifies the number of tickets booked.

**EvtTotal**

The field identifies the total value of the bookings for the returned excursions.

**EvtNotes**

The field identifies the additional information about the event.

**EvtPrint**

The field identifies how many times the tickets had been printed.

**EvtDiscount**

Discount Amount.

**EvtControl**

Internal Booking Number.

**EvtPrepaid**

True – Prepaid at shore side, False – Book onboard.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("GetEvtFolio", "SESSIONID", New Object()
{12345})
```

## GetExcFolio

This function is used to request a list of excursions booked on a specific account in SPMS.

**Note:** Not supported when it is connected to an offline QCI Sync database.

**Input Parameters**

psFunction= "GetExcFolio" — below is the list of psParam or poParam.

**Table 4-61 psParam**

Parameter	Type	Description	Required
pnAccID	Integer	Unique Guest ID.	Yes

Below are the descriptions of the parameters.

**pnAccID**

Unique Guest Id, return from SearchPassenger.

### Return Values

CResponseXml.bSuccess = False when input values are invalid or no record found.

CResponseXml.bSuccess = True and CResponseXML.oTables(0) = *ADO.NET data table* when input values are valid.

**Table 4-62 ADO.NET Data Table field structure**

Code	Type	Description
ExcID	Integer	Excursion ID.
ExcNo	String(10)	Excursion Number.
ExcName	String(50)	Excursion Name.
ExcBegDate	Date	Starting Date and Time.
ExcEndDate	Date	Ending Date and Time.
ExcStatus	String(1)	Excursion Status.
ExcBookDate	Date	Booking Date and Time.
ExcQty	Integer	Number of Tickets Booked.
ExcTotal	Double	Booking Value.
ExcMeetDate	Date	Meeting Date and Time.
ExcMeetPlace	String(100)	Meeting Point.
ExcClothes	String(100)	Recommended Clothes.
ExcNotes	String(200)	Notes.
ExcTag	String(1)	TAG.
ExcPrint	Integer	Ticket Printed Count.
ExcDiscount	Double	Discount Amount.
ExcControl	Integer	Internal Booking Number.
ExcLanguage	String(250)	Language.
ExcPrepaid	Boolean	Prepaid.
ExcCategory	String(500)	Category ID & Description [ID–Description; ID–Description].

### ExcID

The field identifies the unique excursion identifier. This ID can be use to link the excursion with items retrieved from Excursions message. The field is an internal, unique excursion number.

### ExcNo

The field identifies the custom, users entered, a number that identifies excursion.

### ExcName

The field identifies the excursion name (description).

### ExcBegDate

The field identifies the excursion starting date and time.

**ExcEndDate**

The field identifies the excursion ending date and time.

**ExcStatus**

The field identifies the excursion booking status. The following status identifiers values are defined:

A	→	Available
F	→	Fully Booked
O	→	Overbooked

**ExcBookDate**

The field identifies the excursion booking date and time (real date and time).

**ExcQty**

The field identifies the number of tickets booked.

**ExcTotal**

The field identifies the total value of the bookings for the returned excursions.

**ExcMeetDate**

The field identifies the excursion participants meeting date and time.

**ExcMeetPlace**

The field identifies the excursion participants meeting place.

**ExcClothes**

The field identifies the recommended clothes for excursions participants.

**ExcNotes**

The field identifies the additional information about the excursion.

**ExcTag**

The field identifies the adult or children indicator.

A	→	Adult
C	→	Children

**ExcPrint**

The field identifies how many times the tickets had been printed.

**ExcDiscount**

Discount Amount.

**ExcControl**

Internal Booking Number.

**ExcLanguage**

Language use. For example, English.

**ExcPrepaid**

True – Prepaid at shore side, False – Book onboard.

**ExcCategory**

The field identifies the categories for the excursion.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("GetExcFolio", "SESSIONID", New Object()
{12345})
```

## GetExcPackage

This function searches the list of excursion package available in SPMS.

The message is usually used to display a list of excursion packages that fall within the start and end date criteria. The following is the description of the fields in the message data block. A package usually contains more than one excursion; all tours within the package must fall within the start date and end date. If either date is ignored, it will return all the package belongs to current cruise.

**Note:** Not supported when it is connected to an offline QCI Sync database.

**Input Parameters**

psFunction= "GetExcPackage" — below is the list of psParam or poParam.

**Table 4-63 psParam**

Parameter	Type	Description	Required
pnAccID	Integer	Unique Guest ID.	Yes
pdSDate	Date	Excursion start date.	Yes
pdEDate	Date	Excursion end date.	Yes

Below are the descriptions of the parameters.

**pnAccID**

Unique Guest IdD, return from SearchPassenger.

**pdSDate**

The field identifies the minimum date the excursions package can start from. Enter 30-Dec-1899 to return all package belong to current cruise.

**pdEDate**

The field identifies the maximum date the excursions package must be completed. Enter 30-Dec-1899 to return all packages belong to current cruise.

**psErrMsg**

It will return an error message, For example, DB connection is down.

**Return Values**

CResponseXml.bSuccess = False when input values are invalid or no record found.

CResponseXml.bSuccess = True and CResponseXML.oTables(0) = *ADO.NET data table* when input values are valid.

**Table 4-64 ADO.NET Data Table field structure for data table xxx. Return one record when input is valid**

Code	Type	Description
ExcPkgID	Integer	Package ID.
ExcPkgNo	String(10)	Package Number.
ExcPkgName	String(50)	Package Name.
ExcAdultPrice	Double	Package Adult Price.
ExcChildPrice	Double	Package Child Price.
ExcShowInKiosk	Boolean	Show in Kiosk [External Booking System].
ExcBookInKiosk	Boolean	Book in Kiosk [External Booking System].
ExcMaxSeat	Integer	Maximum Package or Maximum Adult Package
ExcBooked	Integer	Package or Adult Package Currently Booked
ExcAdultStatus	String(1)	Booking or Adult Booking Status.
ExcAdultAv	Integer	Package or Adult package available.
ExcMaxChildSeat	Integer	Maximum Childs Package.
ExcChildBooked	Integer	Child Package Booked.
ExcChildStatus	String(1)	Child Booking Status.
ExcChildAv	Integer	Child package available.
ExcProAdultPrice	Integer	Promotion Adult Price.
ExcProAdultAv	Integer	Promotion Adult Availability.
ExcProAdultID	Integer	Adult Promotion ID.
ExcProChildPrice	Integer	Promotion Child Price.
ExcProChildAv	Integer	Promotion Child Availability.
ExcProChildID	Integer	Child Promotion ID.
ExcID	Integer	Excursion ID.
ExcNo	String(10)	Excursion Number.

**Table 4-64 (Cont.) ADO.NET Data Table field structure for data table xxx. Return one record when input is valid**

Code	Type	Description
ExcName	String(50)	Excursion Name.
ExcBegDate	Date	Starting Date and Time.
ExcEndDate	Date	Ending Date and Time.
ExcMeetDate	Date	Meeting Date and Time.
ExcMeetPlace	String (100)	Meeting Point.
ExcClothes	String (100)	Recommended Clothes.
ExcNotes	String (200)	Notes.
ExcPort	String (50)	Port Name.
ExcCategory	String(500)	Category ID & Description [ID–Description; ID–Description].
ExcLang	String(500)	Language ID + “-“ + Language Description With “,” as field separator.
ExcLongDesc	String(1000)	Tour Long Description.

**ExcPkgID**

The field identifies the unique excursion identifier. This ID can be used for excursion package booking. The field is an internal, unique excursion number.

**ExcPkgNo**

The field identifies the custom, users entered, a number that identifies excursion.

**ExcPkgName**

The field identifies the excursion name (description).

**ExcAdultPrice**

The field identifies the excursion adult price.

**ExcChildPrice**

The field identifies the excursion child price.

**ExcShowInKiosk**

The flag to denote if the excursion can be shown on the external system.

**ExcBookInKiosk**

The flag to denote if the excursion can be booked on the external system.

**ExcMaxSeat**

If PAR 'Use Adult/Child Max Counts' is Enabled, this is the maximum adult packages available, if the PAR is Disabled this is the total packages available.

### **ExcBooked**

If PAR 'Use Adult/Child Max Counts' is Enabled, this is the adult packages currently booked on the excursion, if the PAR is Disabled this is the total packages booked.

### **ExcAdultStatus**

The field identifies the excursion booking status. If PAR 'Use Adult/Child Max Counts' is Enabled, this is the adult booking status, if the PAR is Disabled this is the general booking status.

The following status identifiers values are defined:

A	→	Available
F	→	Fully booked
O	→	Overbooked

### **ExcAdultAv**

If PAR 'Use Adult/Child Max Counts' is Enabled, this is the current adult package available on the excursion, if the PAR is Disabled this is the total package available.

### **ExcMaxChildSeat**

The field identifies the maximum child packages available. This field only valid when PAR 'Use Adult/Child Max Counts' is Enabled.

### **ExcChildBooked**

The field identifies the child packages currently booked. This field only valid when PAR 'Use Adult/Child Max Counts' is Enabled.

### **ExcChildStatus**

The field identifies the excursion booking status. If PAR 'Use Adult/Child Max Counts' is Enabled. This field only valid when PAR 'Use Adult/Child Max Counts' is Enabled.

The following status identifiers values are defined:

A	→	Available
F	→	Fully booked
O	→	Overbooked

### **ExcChildAv**

If PAR 'Use Adult/Child Max Counts' is Enabled, this is the current child package available. This field only valid when PAR 'Use Adult/Child Max Counts' is Enabled.

### **ExcProAdultPrice**

This is the adult promotion price.



**ExcProAdultAv**

This is the number of packages available for adult promotion price.

**ExcProAdultID**

This is the adult promotion ID, it can be used during 'ExcPackageBooking'. Please refer to pnAdultPromoID in ExcPackageBooking message type for more details.

**ExcProChildPrice**

This is the child promotion price.

**ExcProChildAv**

This is the number of packages available for child promotion price.

**ExcProChildID**

This is the child promotion ID, it can be used during Booking. Please refer to pnChildPromoID in ExPackageBooking message type for more details

**ExcID**

The field identifies the unique excursion identifier. This ID can be used for excursion booking. The field is an internal, unique excursion number.

**ExcNo**

The field identifies the custom, users entered, a number that identifies excursion.

**ExcName**

The field identifies the excursion name description.

**ExcBegDate**

The field identifies the excursion starting date and time.

**ExcEndDate**

The field identifies the excursion ending date and time.

**ExcMeetDate**

The field identifies the excursion participants meeting date and time.

**ExcMeetPlace**

The field identifies the excursion participants meeting place.

**ExcClothes**

The field identifies the recommended clothes for excursions participants.

**ExcNotes**

The field identifies the additional information about the excursion.

**ExcPort**

The field identifies the port for the excursion.

**ExcCategory**

The field identifies the categories for the excursion.

**ExcLang**

The field identifies the Language ID (2 digits) + “-“ + Language Description, + “,” as the field separator. In one tour there could be more than 1 language available

For example,

EN – English

DE – German

The field will be “EN-English, DE-German”

**ExcLongDesc**

Tour long Description intended for ITV usage.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("GetExcPackage", "SESSIONID", New Object()
{12345, #1/1/2011#, #1/12/2011#})
```

## GetExcursion

This function searches and list the available excursion in SPMS.

**Note:** Not supported when it is connected to an offline QCI Sync database.

**Input Parameters**

psFunction= “GetExcursion” — below is the list of psParam or poParam.

**Table 4-65 psParam**

Parameter	Type	Description	Required
pnAcclID	Integer	Unique Guest ID.	Yes
pdSDate	Date	Excursion start date.	Yes
pdEDate	Date	Excursion end date.	Yes
peCodeType	ExcCodeType	Code Type.	Yes
psCode	String	Code.	Optional

Below are the descriptions of the parameters.

**pnAccID**

Unique Guest ID, return from SearchPassenger.

**pdSDate**

The field identifies the minimum date the excursions can start from.

**pdEDate**

The field identifies the maximum date the excursions must be completed.

**peCodeType**

(Enumerator)

The field identifies the type of the code being passed for querying excursions.

None	0	→	No Code will be used
PortName	1	→	Excursion Port Name
ExcursionNo	2	→	Excursion No (ExcNo)
UniqueID	3	→	Excursion Unique ID (ExcID)
Category	4	→	Excursion Category Code

**psCode**

The code used to query the excursions. It can be either the Port name, The Excursion No (ExcNo), the Unique ID (ExcID) or the Excursion Category Code.

**Return Values**

CResponseXml.bSuccess = False when input values are invalid or no record found.

CResponseXml.bSuccess = True and CResponseXML.oTables(0) = *ADO.NET data table* when input values are valid.

**Table 4-66 ADO.NET Data Table field structure**

Code	Type	Description
ExcID	Integer	Excursion ID.
ExcNo	String (10)	Excursion Number.
ExcName	String (50)	Excursion Name.
ExcBegDate	Date	Starting Date and Time.
ExcEndDate	Date	Ending Date and Time.
ExcAdultStatus	String (1)	Booking or Adult Booking Status.
ExcAdultPrice	Double	Adult Price.
ExcChildPrice	Double	Child Price.
ExcMeetDate	Date	Meeting Date and Time.
ExcMeetPlace	String (100)	Meeting Point.
ExcClothes	String (100)	Recommended Clothes.

**Table 4-66 (Cont.) ADO.NET Data Table field structure**

<b>Code</b>	<b>Type</b>	<b>Description</b>
ExcNotes	String (200)	Notes.
ExcShowInKiosk	Boolean	Show in Kiosk [External Booking System]
ExcBookInKiosk	Boolean	Book in Kiosk [External Booking System].
ExcMaxSeat	Integer	Maximum Seats or Maximum Adult Seats.
ExcBooked	Integer	Seats or Adult Seats Currently Booked.
ExcPort	String (50)	Port Name.
ExcCategory	String (500)	Category ID & Description [ID–Description; ID–Description].
ExcLang	String (500)	Language ID + “-“ + Language Description With “,” as field separator
ExcShowInITV	Boolean	Show in ITV [External Booking System].
ExcBookInITV	Boolean	Book in ITV [External Booking System].
ExcBookChildInITV	Boolean	Book Child Ticket in ITV [External Booking System].
ExcMaxChildSeat	Integer	Maximum Childs Seats.
ExcChildBooked	Integer	Child Seats Booked.
ExcChildStatus	String(1)	Child Booking Status.
ExcProAdultPrice	Integer	Promotion Adult Price.
ExcProAdultAv	Integer	Promotion Adult Availability.
ExcProAdultID	Integer	Adult Promotion ID.
ExcProChildPrice	Integer	Promotion Child Price.
ExcProChildAv	Integer	Promotion Child Availability.
ExcProChildID	Integer	Child Promotion ID.
ExcLongDesc	String(1000)	Tour Long Description.
ExcNo2	String	Excursion Brochure Number.
ExcMinSeat	Integer	Excursion minimum capacity.
ExcWeight	Integer	Total weight allow for the excursion.
ExcWeightPerGuest	Boolean	True the weight is per guest, False the weight is per tour.
ExcExternalID	String	External ID.

**ExcID**

The field identifies the unique excursion identifier. This ID can be used for excursion booking and is an internal, unique excursion number.

**ExcNo**

The field identifies the custom, users entered, a number that identifies excursion.

**ExcName**

The field identifies the excursion name (description).

**ExcBegDate**

The field identifies the excursion starting date and time.

**ExcEndDate**

The field identifies the excursion ending date and time.

**ExcAdultStatus**

The field identifies the excursion booking status. If PAR 'Use Adult/Child Max Counts' is 'Enabled', this is the adult booking status. If the PAR is 'Disabled' this is the general booking status.

The following status identifiers values are defined:

A	→	Available
F	→	Fully booked
O	→	Overbooked

**ExcAdultPrice**

The field identifies the excursion adult price.

**ExcChildPrice**

The field identifies the excursion child price.

**ExcMeetDate**

The field identifies the excursion participants meeting date and time.

**ExcMeetPlace**

The field identifies the excursion participants meeting place.

**ExcClothes**

The field identifies the recommended clothes for excursions participants.

**ExcNotes**

The field identifies the additional information about the excursion.

**ExcShowInKiosk**

The flag to denote if the excursion can be shown on the external system.

**ExcBookInKiosk**

The flag to denote if the excursion can be booked on the external system.

### **ExcMaxSeat**

If PAR 'Use Adult/Child Max Counts' is Enabled, this is the maximum adult seats available on the excursion, if the PAR is Disabled this is the total seats available on the excursion.

### **ExcBooked**

If PAR 'Use Adult/Child Max Counts' is Enabled, this is the adult seats currently booked on the excursion, if the PAR is Disabled this is the total seats booked on the excursion.

### **ExcPort**

The field identifies the port for the excursion.

### **ExcCategory**

The field identifies the categories for the excursion.

### **ExcLang**

The field identifies the Language ID (2 digits) + "-" + Language Description, + "," as the field separator. In one tour there could be more than 1 language available

For example,

EN – English

DE – German

The field then becomes "EN-English, DE-German"

### **ExcShowInITV**

The flag to denote if the excursion can be shown on ITV the external system.

### **ExcBookInITV**

The flag to denote if the excursion can be booked on the ITV external system.

### **ExcBookChildInITV**

The flag to denote if the excursion child ticket can be booked on the ITV external system.

### **ExcMaxChildSeat**

The field identifies the maximum child seats available on the excursion. This field only exists when PAR 'Use Adult/Child Max Counts' is Enabled.

### **ExcChildBooked**

The field identifies the child seats currently booked on the excursion. This field only exists when PAR 'Use Adult/Child Max Counts' is Enabled.

### **ExcChildStatus**

The field identifies the excursion booking status. If PAR 'Use Adult/Child Max Counts' is Enabled, this is the child booking status. This field does not exist if the PAR is Disabled.

The following status identifiers values are defined:

A	→	Available
F	→	Fully booked
O	→	Overbooked

**ExcProAdultPrice**

This is the adult promotion price.

**ExcProAdultAv**

This is the number of seats available for adult promotion price.

**ExcProAdultID**

This is the adult promotion ID, it can be used during Booking. Please refer to pnAdultPromoID in ExcBooking message type for more details.

**ExcProChildPrice**

This is the child promotion price.

**ExcProChildAv**

This is the number of seats available for child promotion price.

**ExcProChildID**

This is the child promotion ID, it can be used during Booking. Please refer to pnChildPromoID in ExcBooking message type for more details.

**ExcLongDesc**

Tour long description intended for ITV usage.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("GetExcursion", "SESSIONID", New  
Object() {12345, #1/1/2011#, #1/12/2011#, 0, ""})
```

## GetFreeCabin

This function gets the available cabin for booking.

**Input Parameters**

psFunction= "GetFreeCabin" — below is the list of psParam or poParam.

**Table 4-67 psParam**

Parameter	Type	Description	Required
psCabinType	String	Cabin Type, "P" – Passenger, "C" - Crew.	Yes
pdEmbDate	Date	Embarkation Date.	Yes
pdDisDate	Date	Disembarkation Date.	Yes
pdFromDoneDate	Date	Work Order Completed From Date.	Optional
pdToDoneDate	Date	Work Order Completed To Date.	Optional
psWorkOrderDone	String	Completed Work Order Task Code, link to TYP_MWT.TYP_ART, default value XXX.	Optional
psWorkOrderNotDone	String	Unfinish Work Order Task Code, link to TYP_MWT.TYP_ART, default value XXX.	Optional
psCabinCategory	String	Cabin Category.	Optional
psFeature	String	Cabin Feature	Optional
psPartialCabin	String	First few digit of partial cabin number.	Optional
psDeck	String	Cabin Deck.	Optional
psSpecialNeed	Enum	0-All. 1-With Special Need. 2-Without Special Need.	Optional

**Return Values**

CResponseXml.bSuccess = False when GetFreeCabin has failed

CResponseXml.bSuccess = True when GetFreeCabin is successful, CResponse.oDataset will have 1 Data Tables.

**Table 4-68 ADO.NET Data Table field structure for data table**

Code	Type	Description
CAB_ID	String	Cabin Number.
TYP_COMMENT	String	Cabin Category Description.
CAB_DECK	String	Cabin Deck Code.
CAB_STATUS	String	Cabin Status.
CAB_BEDS	Integer	Number of bed in this cabin.
CAB_TYPE	String	Cabin Category Code.
CAB_DECK	String	Cabin Deck.
CAB_CONNECTED	String	Connected Cabin Number.
CAB_FEATURE	String	List of Cabin Feature separated by ",".
CAB_AVAILABLE	Integer	Available Berth.
CAB_SORT	Integer	Sort Priority.
CAB_SPECIALNEED	Integer	0-Normal Cabin, 1-cabin with special needs feature.
CAB_DECK_NAME	String	Cabin Deck Description.



**Example:**

```
oResponse = goWS.FidelioSPMSWSXML("GetFreeCabin", "", New Object() {"P",
DateTime.SpecifyKind(#1/1/2011#, DateTimeKind.Unspecified), _
DateTime.SpecifyKind(#2/1/2011#, DateTimeKind.Unspecified)})
```

## GetHouseKeepingSections

This function returns a list of ALL housekeeping section codes and names based on the criteria below.

**Input Parameters**

psFunction="GetHouseKeepingSections" — below is the list of poParam.

For Array field, pass in an empty array or "Nothing" to skip this checking.

For JSON, the array of string will be a string separated by ",", each string must start and end with a double quotation mark ("").

For example,

"A","B" = 2 Element

Element 1 = A,

Element 2 = B,

**Note:** Not supported when it is connected to an offline QCI Sync database.

**Table 4-69 psParam**

Parameter	Type	Description	Required
psCrewIds	Array of String	List of unique crew Account IDs. If this is not passed in, then all HouseKeeping sections are returned.	Yes
psCabin	Array of String	List of Cabins.	Yes
psLocations	Array of String	List of Locations.	Yes
psDeck	Array of String	List of Deck.	Yes
psCabinCategory	Array of String	List of Cabin Categories.	Yes
psCabinType	Array of String	List of Cabin Types.	Yes
psCabinVerticalZone	Array of String	List of Cabin Vertical Zones.	Yes

**Return Values**

CRResponseXml.bSuccess = False when SearchHouseKeepingSections has failed

CRResponseXml.bSuccess = True when SearchHouseKeepingSections is successful,  
CRResponse.oDataset will have 1 Data Table.

**Table 4-70 ADO.NET Data Table field structure**

Code	Type	Description
HSC_NAME	String	Housekeeping Section Code.
HSC_COMMENT	String	Housekeeping Section Name.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("GetHouseKeepingSections", "Session ID", New
Object() { Nothing , Nothing , Nothing , Nothing , Nothing , Nothing ,
Nothing })
```

## GetILOViolations

This function retrieves all the ILO violations on a specified date for a crew.

### Input Parameters

psFunction= "GetILOViolations" — below is the list of psParam or poParam.

**Table 4-71 psParam**

Parameter	Type	Description	Required
pnAccID	Integer	Unique Crew Account ID. Either psBoardCardNo or pnAccID is required.	Yes
pdDayRequested	Date	Date of the ILO Violations.	Yes

### Return Values

CResponseXml.bSuccess = False when Update failed.

CResponseXml.bSuccess = True when Update successful, CResponse.oDataset will have 1 Data Table

**Table 4-72 ADO.NET Data Table field structure for data table TALG**

Code	Type	Description
TALG_ID	Integer	Unique ID of the record.
TALG_TCLK_ID	Integer	Unique ID of the TCLK record. Linked to TCLK.TCLK_ID.
TALG_MODDATE	Date	Last modified date.
TALG_ILO_NOTES	String	ILO Violation Notes.
TALG_MGR_NOTES	String	Manager approval notes.
TALG_ILO_7DAYS	Integer	1=violation for ILO for 7 days (exceeded max worked hours for the last 7 days).
TALG_ILO_24HRS	Integer	1=violation for ILO 24hrs (exceeded max worked hours for the last 24hrs).

**Table 4-72 (Cont.) ADO.NET Data Table field structure for data table TALG**

Code	Type	Description
TALG_ILO_MINRESTBLOC K	Integer	1=violation for ILO minimum rest hr block.
TALG_ILO_MINREST_7DA YS	Integer	1=violation for ILO minimum rest hours in 7 days.
TALG_ILO_MINREST_24H RS	Integer	1=violation for ILO minimum rest hours for the last 24hrs.

## GetLuggageInfo

This function retrieves the luggage movement of the guest.

**Note:** Not supported when it is connected to an offline QCI Sync database.

### Input Parameters

psFunction= "GetLuggageInfo" — below is the list of psParam or poParam.

**Table 4-73 psParam**

Parameter	Type	Description	Required
peSearchType	Enum	InternalID = 0 'res_acc. PGID = 1 'res_pgid. BoardCard = 2 'res_boardcc. PaxID = 3 'res_referenz. Cabin = 4 'res_cab.	Yes
psSearchStr	String	Unique Payer Account ID or Board Card or PGID or Pax ID depends on the peSearchType.	Yes
peSearchLevel	Enum(Byte)	0 – By Passenger. 1 – By Cabin. 2 – By Booking No.	Yes
peSearchStatus	Enum(Byte)	0 – All Location (All Status). 1 – Latest Location (Current Status).	Yes

### Return Values

CResponseXml.bSuccess = False when nothing is found or invalid input parameters.

CResponseXml.bSuccess = True when successful. CResponse.oDataset will have 1 Data Tables.

The Data Tables will be different depending on peSearchStatus input parameter.

**Table 4-74 ADO.NET Data Table field structure for data table for peSearchStatus = 0**

Code	Type	Description
LUG_ACC_ID	Integer	Unique Account ID.
UXP_A_FSTN	String(40)	Guest First Name.
UXP_A_NAME	String(40)	Guest Last Name.
LUM_RFID	String(20)	RFID Tag.
LUM_ZONE	String(20)	Zone ID.
LUM_TYPE	String(20)	RFID Type – as send from Strategic.
PAIRING_TIME	Date	The time the tag is associated with the guest
LUM_MOVEMENT	Date	The scan time of the tag.
TIME_REMAINING	Integer	Number of minutes before delivery to room/pier (From the last movement time).
LUZ_LOCATION	String(50)	Actual Location.
DIRECTION	String(8)	Inbound or Outbound.
GUEST_LOCATION	String(50)	Guest Friendly Location Description.
DELIVERED	Integer	1-Delivery to guest room, 0-Not yet delivery.
QUARANTINE	Integer	Y-In quarantine zone, N-In normal zone.
PERCENT_COMPLETE	Double	Percentage complete for delivery.

**Table 4-75 ADO.NET Data Table field structure for data table for peSearchStatus = 1**

Code	Type	Description
LUG_ACC_ID	Integer	Unique Account ID.
UXP_A_FSTN	String(40)	Guest First Name.
UXP_A_NAME	String(40)	Guest Last Name.
LUM_RFID	String(20)	RFID Tag.
LUM_ZONE	String(20)	Zone ID.
LUM_TYPE	String(20)	RFID Type – as send from Strategic.
PAIRING_TIME	Date	The time the tag is associated with the guest
FIRSTSCAN_TIME	Date	The time the tag first scan, For example, on X-Ray machine
LUM_MOVEMENT	Date	Last movement time of the bag.
DELIVERY_TIME	Date	Expected delivery time.
TIME_REMAINING	Integer	Number of minutes before delivery to room/pier from current time.
LUZ_LOCATION	String(50)	Actual Location.
DIRECTION	String(8)	Inbound or Outbound.
GUEST_LOCATION	String(50)	Guest Friendly Location Description.
DELIVERED	Integer	1-Delivery to guest room, 0-Not yet delivery.
QUARANTINE	Integer	Y-In quarantine zone, N-In normal zone.
PERCENT_COMPLETE	Double	Percentage complete for delivery.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("GetLuggageInfo", "SessionID", New
Object() {0,12345,0,1})
```

## GetMicrosAdvanceDiscount

This function gets the list of eligible advance MICROS discount list.

**Note:** Not supported when it is connected to an offline QCI Sync database.

### Input Parameters

psFunction= "GetMicrosAdvanceDiscount" — below is the list of psParam or poParam.

**Table 4-76 psParam**

Parameter	Type	Description	Required
pnAccID	Integer	Unique Guest ID.	Yes

Below are the descriptions of the parameters.

### pnAccID

Unique Guest ID, return from GuestSearch.

### Return Values

CResponseXml.bSuccess = True when GetMicrosAdvanceDiscount is successful, CResponse.oObj returns list of discount in a string separated by ASCII 10, it mainly for display purpose only.

CResponseXml.bSuccess = False when GetMicrosAdvanceDiscount has failed.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("GetMicrosAdvanceDiscount", "", New
Object() {12345})
```

## GetPackageInfo

This function gets the list of Package Plan assigned to the passenger.

**Note:** Not supported when it is connected to an offline QCI Sync database.

### Input Parameters

psFunction="GetPackageInfo" — below is the list of psParam or poParam.

**Table 4-77 psParam**

Parameter	Type	Description	Required
pnAccID	Integer	Unique Guest ID.	Yes

**Table 4-77 (Cont.) psParam**

Parameter	Type	Description	Required
pbIncludeDisablePackage	Boolean	True=Include Disable Package. False=Just return active package.	Optional

Below are the descriptions of the parameters.

**pnAccID**

Unique Guest ID, return from GuestSearch.

**Return Values**

CResponseXml.bSuccess = False when GetPackageInfo has failed

CResponseXml.bSuccess = True when GetPackageInfo is successful, CResponse.oDataset will have 2 Data Tables.

**Table 4-78 ADO.NET Data Table field structure for first data table**

Code	Type	Description
PPN_NAME	String	Name of the Package Plan.
PPN_DAY_CREDIT	Double	Package Total.
PPP_DEBIT	Double	Package Used.
PPP_BOOKMARK	Integer	A point to the record position of the second data table.
PPP_SELECTED	Integer	Record filtering, when 0 should hide the record from the second data table. However, the implementation is up to the client.
PPN_USE_DEP_TOTAL	Integer	1 - Different Department Total has it own limit
PPP_ID	Integer	Unique Package ID.
PPN_FLEX_PACKAGE	Integer	Flexible Package Flag, the guest will need to choose the available department from the setup.
AvailableBalance	Double	Available Balance = PPN_DAY_CREDIT – PPP_DEBIT

**Table 4-79 ADO.NET Data Table field structure for second data table**

Code	Type	Description
DEP_PPP_ID	Integer	Unique Package ID = PPP_ID of the first data table.
DEP_ID	String	Department ID or Main Department ID.
DEP_NAME	String	Department Name or Main Department Name.
DEP_TOTAL	Double	This will be the individual department total when PPN_USE_DEP_TOTAL = 1.
DEP_MAIN	Boolean	Indicate this record is the Department or Main Department.

**Table 4-79 (Cont.) ADO.NET Data Table field structure for second data table**

Code	Type	Description
DEP_DISC	Double	Discount Percentage that the posting amount will go to package plan.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("GetPackageInfo", "", New Object()
{12345})
```

## GetPeopleCount

This function gets the onboard/shoreside count per the WPF Security module.

**Note:** Not supported when it is connected to an offline QCI Sync database.

### Input Parameters

psFunction= "GetPeopleCount", no parameter.

### psParam

None.

### Return Values

CResponseXml.bSuccess = False when not connection to the DB.

CResponseXml.bSuccess = True and CResponseXML.PeopleCount

**Table 4-80 PeopleCount field structure**

Parameter	Type	Description
nPaxTotal	Integer	Total Passenger.
nPaxShoreside	Integer	Total Passenger shore side.
nPaxOnboard	Integer	Total Passenger on board.
nPaxOvernight	Integer	Total Passenger shore side with overnight tour.
nPaxExpected	Integer	Total expected passenger to check-in today.
nPaxDisembaraked	Integer	Total passenger disembark today.
nResTotal	Integer	Total Resident.
nResShoreside	Integer	Total Resident shore side.
nResOnboard	Integer	Total Resident on board.
nResOvernight	Integer	Total Resident shore side with overnight tour.
nResExpected	Integer	Total expected resident to check-in today.
nResDisembaraked	Integer	Total resident disembark today.
nCrwTotal	Integer	Total Crew.
nCrwShoreside	Integer	Total Crew shore side.

**Table 4-80 (Cont.) PeopleCount field structure**

Parameter	Type	Description
nCrwOvernight	Integer	Total Crew on board.
nCrwOnboard	Integer	Total Crew shore side with overnight tour.
nCrwExpected	Integer	Total expected crew to check-in today.
nCrwDisembarked	Integer	Total crew disembark today.
nTotal	Integer	Total.
nTotalOnboard	Integer	Total shore side.
nTotalShoreside	Integer	Total on board.
nTotalOvernight	Integer	Total shore side with overnight tour.
nTotalExpected	Integer	Total expected to check-in today.
nTotalDisembarked	Integer	Total disembark today.
nVisTotal	Integer	Total visitor on board.
nPaxNotPassImmigration	Integer	Total passenger not yet pass immigration gate.
nCrewNotPassImmigration	Integer	Total crew not yet pass immigration gate.
nResNotPassImmigration	Integer	Total resident not yet pass immigration gate.
nPaxExcludeImmigration	Integer	Total passenger no need to pass immigration gate.
nCrewExcludeImmigration	Integer	Total crew no need to pass immigration gate.
nResExcludeImmigration	Integer	Total resident no need to pass immigration gate.
sPortName	String	Port Description.
oVisitor	Array or CVisitorDetailCount	This only applies to HRG only.

**Table 4-81 CVisitorDetailCount**

Parameter	Type	Description
Code	String	Visitor Category Code.
Description	String	Visitor Category Description.
Count	Integer	Total Visitor on board.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("GetPeopleCount", "SESSIONID", New Object()
{})
```

## GetPreviousClockInOutActivities

This function retrieves the previous days' clock-in/out transactions.



**Input Parameters**

psFunction= "GetPreviousClockInOutActivities" — below is the list of psParam or poParam.

**Table 4-82 psParam**

Parameter	Type	Description	Required
pnAcclID	Integer	Unique Crew Account ID.	Yes
pnDays	Integer	Optional. The number of previous days to get. If 0 then will default to days defined in parameter 'Display Previous X Days Clock-In/Out at terminal.'	

**Return Values**

CResponseXml.bSuccess = False when Update failed.

CResponseXml.bSuccess = True when Update successful, CResponse.oDataset will have 1 Data Table.

**Table 4-83 ADO.NET Data Table field structure for data table TCLK**

Code	Type	Description
TCLK_DATETIMESTAMP	Date	Date time stamp of the clock-in/out transaction.
TCLK_IN_OUT	String	'I' for clock-in, 'O' for clock-out.
TCLK_ENTRYTYPE	String	'A' for auto, 'M' for manual.
TARS_CODE	String	Task work code.
TARS_NAME	String	Task work name.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("GetPreviousClockInOutActivities",
"SessionID", New Object() {12345, 0})
```

## GetRFIDEncodingData

This function retrieves the Ving door lock information and charge card information from SPMS.

**Note:** Not supported when it is connected to an offline QCI Sync database.

**Input Parameters**

psFunction= "GetRFIDEncodingData" — below is the list of psParam or poParam.

**Table 4-84 psParam**

Parameter	Type	Description	Required
peSearchType	Enum	0 = Internal ID. 1 = PGID. 2 = BoardCard. 3 = Pax ID.	Yes
psSearchStr	String	Unique Payer Account ID or Board Card or PGID or Pax ID depends on the peSearchType.	Yes
psRFIDUID	String(16)	Convert the value to HEX.	Yes

**Return Values**

CResponseXml.bSuccess = False when failed.

CResponseXml.bSuccess = True when successful. CResponse.oDataset will have 1 Data Tables.

**Table 4-85 ADO.NET Data Table field structure for data table**

Code	Type	Description
TRACK1	String(76)	Track 1 – Custom Information.
TRACK2	String(37)	Track 2 – SPMS Charge Card information.
TRACK2_BIN	Byte(7)	Numeric Track 2 In Binary Format (Excluding Expiry).
TRACK3	String(200)	Track 3 - Ving encoding string format.
ENCODED	Boolean	True=Previously Encoded, False=New RFID.
ACCID	Integer	Unique Guest ID.
TRACK1_CUSTOM	STRING(168)	84-byte Hex String representing ASCII track 1 - including 00s at the end For example, 32363534323735315E4C5E4F415E43335E3230 31343035303335E4A6F686E5E536D6974685E41 6E74686F6E79000000000000000000000000 0000000000000000000000000000000000 00000000000000.
TRACK2_CUSTOM	STRING(24)	Custom Track 2 - 8-byte Hex String representing numerical RES_BOARDCC concatenated with 4 byte- Hex String representing numerical DEBARK_DATE) For example, 8126E86056F62200E5513301.
TRACK3_CUSTOM	STRING(96)	48-byte HexString representation of the VING String including any 00s at the end. For example, B824935021D7B0C2A81573C63456F85AB3B15 D2DFA7E91F000000000000000000000000 000000000000000000000000.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("GetRFIDEncodingData", "SessionID", New
Object() {0,12345,"E01581A07C123B"})
```

## GetSeatingZoneAvailability

This function requests for all available Seats of a given Event, Layout Level and Seating Zone from the SPMS.

**Note:** Not supported when it is connected to an offline QCI Sync database.

### Input Parameters

psFunction= "GetSeatingZoneAvailability" — below is the list of psParam or poParam.

**Table 4-86 psParam**

Parameter	Type	Description	Required
pnEvtID	Integer	Event ID.	Yes
EvtLayoutLevelID	Integer	Layout Level ID.	Yes
EvtSeatingZoneID	Integer	Seating Zone ID.	Yes

Below are the descriptions of the parameters.

#### pnEvtID

The field identifies the unique event ID. This ID can be obtained through GetEvent WebMethod. The field is an internal, unique event number.

#### EvtLayoutLevelID

The field identifies the unique Layout Level identification number. This ID can be obtained through GetEvent WebMethod. The field is an internal, unique layout level number.

#### EvtSeatingZoneID

The field identifies the unique Seating Zone identification number. This ID can be obtained through GetEvent WebMethod. The field is an internal, unique seating zone number.

### Return Values

CResponseXml.bSuccess = False when input values are invalid or no record found.

CResponseXml.bSuccess = True and CResponseXML.oTables(0) = *ADO.NET data table* when input values are valid.

**Table 4-87 ADO.NET Data Table field structure for data table**

Code	Type	Description
TKSD_ID	Integer	Unique ID of the record.
TKSD_CODE	String(20)	Seat ID.

**TKSD\_ID**

The field returns the unique ID for each Seat.

**TKSD\_CODE**

The field returns the Seat ID.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("GetSeatingZoneAvailability", "SESSIONID", New
Object() {62, 82, 24})
```

## GetTreatmentGroupsForLocation

This function is used to request for all Treatment Groups assigned to each Location from the SPMS.

**Note:** Not supported when it is connected to an offline QCI Sync database.

**Input Parameters**

psFunction= "GetTreatmentGroupsForLocation" — below is the list of psParam or poParam.

**Table 4-88 psParam**

Parameter	Type	Description	Required
psSPATreatmentGroups	String	All Default Treatment Groups assigned to a Location, separated by commas.	Yes

**psSPATreatmentGroups**

All Default Treatment Groups assigned to a Location, separated by commas as returned by WSF function, GetSPALocation in field spl\_treatment\_groups.

**Return Values**

CResponseXml.bSuccess = False when input values are invalid or no record found.

CResponseXml.bSuccess = True and CResponseXML.oTables(0) = *ADO.NET data table* when input values are valid.

**Table 4-89 ADO.NET Data Table field structure**

Code	Type	Description
typ_art	String	Treatment Group Code.
typ_comment	String	Treatment Group Description.

**TYP\_ART**

The field returns the Treatment Group Code.

**TYP\_COMMENT**

The field returns the Treatment Group Description.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("GetTreatmentGroupsForLocation",
"SESSIONID", New Object() {"BODY,FACE,"})
```

## GuestSearch

This function searches for guests with various criteria from the SPMS.

### Input Parameters

psFunction= "GuestSearch" — below is the list of psParam or poParam.

**Table 4-90 psParam**

Parameter	Type	Description	Required
psSearchStr	String	Search Criteria.	Yes
pdDate	Date	Embarkation Date.	Yes
peGuestType	GuestType	Guest Type Enumerator.	Yes
peReservationType	ReservationType	Reservation Type Enumerator.	Yes
pbUseSearchType	Boolean	True-Use specific search type False = Use only Search Criteria.	Yes
peSearchType	SearchType	SearchType Enumerator.	Yes
pnScdlID	Integer	Current Port ID.	Optional
peOnboardStatus	OnboardStatus	– All. – Onboard. – Shoreside.	Optional

Below are the descriptions of the parameters.

### psSearchStr

This is the search criteria, supported values as shown below:

- Cabin number
- Guest Name
- Board Card Track 2 and must include “;” and “?” in the string. For example, “;9841000241243900?”
- Credit Card Track Data and must include the start and end sentinel in the search string. For example “;” and “?”)
- Cabin List, separate by “;”. For example, “05002,05000”.
- Passport Number
- Name List (Last Name separate by “;”)
- Booking Number

**pdDate**

Embarkation date, if you want to search for all guest regardless of the embarkation date, pass in #12:00:00AM#

**peGuestType**

(Enumerator)

Guest	0	→	Search guest matching search criteria
Crew	1	→	Search crew matching search criteria
Visitor	2	→	Search visitor matching search criteria
All	3	→	Search guest,crew,visitor matching search criteria

**peReservationType**

(Enumerator)

Reserve = Search only passenger with reservation status only (Note visitor always have reservation status only)

Reserve	0	→	Reservation passenger
CheckIn	1	→	Searches only passenger that had check-in
All	2	→	Search both check-in and reserve passenger
CheckOut	3	→	Search check-out passenger only
ReserveFCManagement	4	→	Reservation passenger, for shore viewer
CheckInFCManagement	5	→	Searches only passenger that had check-in, for shore viewer
CheckOutFCManagement	6	→	Search check-out passenger only, for shore viewer
Cancelled	7	→	Canceled reservation passenger.
AQCIWizardCheckIn	8	→	Use by AQCI Wizard Only
AQCIWizardRegister	9	→	Use by AQCI Wizard Only
ReserveExcludeNoShow	10	→	Reservation passenger excluding No Show
NoShow	11	→	No Show Passenger only

**pbUseSearchType**

**True** = Use specific search type. The search type is normally used to return the same group of guest. For example,

- Search using passport number; function will return list of guests staying in the same cabin
- Search using credit card number; function will return a list of guests staying in the same cabin.
- Want to return a specific guest using UXP\_A\_ID (SPMS unique key) value.

Below is an *incorrect* use of search type:-

Search by name;

In general, there could be more than one guest with the same name, if this happens then the function will search for the first guest in SPMS database that matches the name criteria and

returns the list of guests staying in that same cabin. Since this is based on the first found record and returns the list of names based on that, this may not be accurate.

**False** = Use only Search Criteria

### **peSearchType**

(Enumerator)

#### **SamePartialName = 0**

1. This only works when search criteria psSearchStr passed in is the last name, board card track 2 or credit card track data.
2. The psSearchStr is the partial last name, it will search for guest with matching the partial last name.
3. The psSearchStr is board card track 2 or credit card track data, it will search using the supplied information. When there are records found, it will take the first record last name and search again for all guest with the same partial last name.
4. If the search criteria are other than three of the above, it will only return the first record matching the search criteria.

#### **SameSurname = 1**

1. It will search using the supplied information. When there are records found, it will take the first record last name and search again for all guest with the same last name.
2. However, if the first record last name is blank, it will only return the first record.

#### **NameList = 2**

1. If the search criteria passed in is a list of last names separate by “;”, it will search for guests matching the partial last names in the list.
2. If the psSearchStr is a partial last name, it will search for guest matching the partial last name
3. If the psSearchStr is board card track 2 or credit card track data, it will search using the supplied information. When there are records found, it will take the first record last name and search again for all guests with the same partial last name.
4. For other search criteria, it will search using the supplied information.

#### **SameCabin = 3**

1. It will search using the supplied information. When there are records found, it will take the first record cabin number and search again for all guests with the same cabin number.
2. However, if the first record cabin number is blank, it will only return the first record.

#### **SameCabinList = 4**

1. If the search criteria passed in is a list of cabin numbers separate by “;”, it will search for all guests matching the cabin number in the list.
2. The psSearchStr is cabin number, it will search for all guests matching the cabin number
3. For other search criteria, it will search using the supplied information.

#### **SameBookingNo = 5**

1. It will search using the supplied information. When there are records found, it will take the first record booking number and search again for all guests with the same booking number.
2. However, if the first record booking number is blank, it will only return the first record.

#### SameGroupNo = 6

1. It will search using the supplied information. When there are records found, it takes the first record group number and searches again for all guests with the same group number.
2. However, if the first record group number is blank, it only returns the first record.

#### SameRouting = 7

1. It will search using the supplied information. When the first record is found, it will check if this account has routings to another account. If yes, then the list of records returned would include the first record found and all other accounts that share the same routing as the first account.
2. However, if the first record found does not have routings to another account, then it will search for all accounts that have routings to the first record found.

#### AccountID = 8

This is an SPMS internal account ID, so the parameter *psSearchStr* passed in should be a valid ID (UXP\_A\_ID). All other parameters (pdDate, peGuestType, peReservationType) passed in will be ignored.

#### Return Values

CResponseXml.bSuccess = False when input values are invalid or no record found.

- CResponseXml.bSuccess = True and CResponseXML.oTables(0) = *ADO.NET data table* when input values are valid. The data table contains
  - all fields in uxp table, all fields in res table,
  - uxp\_a\_name field or uxp\_a\_othname field,
  - uxp\_a\_fstn field or uxp\_a\_othfstn field, and
  - org\_assembly field or res\_station field or cab\_station field or par\_value from par table.

See SPMS Data Dictionary for details of the data fields.

#### Note

In all the above statements when we mention “first record found” and the function returns more than one record and the list of records returned does not have any specific sort order, we cannot predict which record will be the first to display.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("GuestSearch", "SessionID", New Object()
{"05002", #01-01-2011#, 3, 0, True, 3})
```

## GetPackageTreatmentAvailability

This function requests the SPA Package Treatment Availability by time slot from the SPMS.

**Note:** Not supported when it is connected to an offline QCI Sync database.



## Input Parameters

psFunction= "GetPackageTreatmentAvailability" — below is the list of psParam or poParam.

**Table 4-91 psParam**

Parameter	Type	Description	Required
pnAccID	Integer	Account ID.	Yes
pnSPATreatmentItem	Integer	Treatment Item ID.	Yes
pnSPAPackageID	Integer	Package ID.	Yes
pdDateFrom	Date	Starting Date period.	Optional
pdDateTo	Date	Ending Date period.	Optional

Below are the descriptions of the parameters.

### pnAccID

Unique Guest ID, return from SearchPassenger.

### pnSPATreatmentItem

The field identifies the unique Package treatment item identification. The field is an internal, unique code. This field is retrieve from GetSPAPackageItems or GetSPAPackageItemsForPeriod WebMethods.

### pnSPAPackageID

The field identifies the unique Package ID. The field is an internal, unique code. This value is retrieved from GetSPAPackages WebMethod.

### pdDateFrom

The fields identify the optional Starting and Ending Dates period for which the system will return availability time slots when **pnAccID** is set to 0.

### pdDateTo

The fields identify the optional Starting and Ending Dates period for which the system will return availability time slots when **pnAccID** is set to 0.

## Return Values

CResponseXml.bSuccess = False when input values are invalid or no record found.

CResponseXml.bSuccess = True and CResponseXML.oTables(0) = *ADO.NET data table* when input values are valid.

**Table 4-92 ADO.NET Data Table field structure**

Code	Type	Description
spa_date	Date	Date and Time.

**Table 4-92 (Cont.) ADO.NET Data Table field structure**

Code	Type	Description
spa_price	Double	Treatment Price.

**SPA\_DATE**

The field returns the date which the treatment is available.

**SPA\_PRICE**

The field identifies the Package treatment item price.

**Example:**

```
oResponse = goWS.FidelioSPMSWSXML("GetPackageTreatmentAvailability",
"SESSIONID", New Object() {12345, 12, 15})
```

```
oResponse = goWS.FidelioSPMSWSXML("GetPackageTreatmentAvailability",
"SESSIONID", New Object() {0, 12, 15, #1/1/2011#, #1/12/2011#})
```

## GetTreatmentAvailability

This function requests the SPA Treatment Availability by time slot from the SPMS.

**Note:** Not supported when it is connected to an offline QCI Sync database.

**Input Parameters**

psFunction= "GetTreatmentAvailability", — below is the list of psParam or poParam.

**Table 4-93 psParam**

Parameter	Type	Description	Required
pnAccID	Integer	Account ID.	Yes
pnSPALocation	Integer	Location ID.	Yes
pnSPATreatmentItem	Integer	Treatment Item ID.	Yes
pdDateFrom	Date	Starting Date period.	Optional
pdDateTo	Date	Ending Date period.	Optional

Below are the descriptions of the parameters.

**pnAccID**

Unique Guest ID, return from SearchPassenger.

**pnSPALocation**

The field identifies the unique location ID. The field is an internal, unique code. This value is retrieved from GetSPALocation WebMethod.

**pnSPATreatmentItem**

The field identifies the unique treatment item identification. The field is an internal, unique code. This field is retrieve from GetSPATreatmentItem or GetSPATreatmentItemForPeriod WebMethods.

**pdDateFrom**

The fields identify the optional Starting and Ending Dates period for which the system will return availability time slots when **pnAccid** is set to 0.

**pdDateTo**

The fields identify the optional Starting and Ending Dates period for which the system will return availability time slots when **pnAccid** is set to 0.

**Return Values**

CResponseXml.bSuccess = False when input values are invalid or no record found.

CResponseXml.bSuccess = True and CResponseXML.oTables(0) = *ADO.NET data table* when input values are valid.

**Table 4-94 ADO.NET Data Table field structure**

Code	Type	Description
spa_date	Date	Date and Time.
spa_price	Double	Treatment Price.

**SPA\_DATE**

The field returns the date which the treatment is available.

**SPA\_PRICE**

The field identifies the treatment item price.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("GetTreatmentAvailability", "SESSIONID",
New Object() {12345, 123, 12})
```

```
oResponse = goWS.FidelioSPMSWSXML("GetTreatmentAvailability", "SESSIONID",
New Object() {0, 123, 12, #1/1/2011#, #1/12/2011#})
```

## Login

This function logged in the user into the system and return a session ID. You must use this session ID to subsequent function calls.

**Note:**

The Login function can only be called with JSON POST or WS XML. JSON GET is not supported for Login.

**Input Parameters**

psFunction= "Login" — below is the list of psParam or poParam.

**Table 4-95 psParam**

Parameter	Type	Description	Required
psLoginName	String	User Login Name.	Yes
psPassword	String	Plain Text Password.	Yes
pnRowsPerPage	Integer	Default 0 = Unlimited, Number of Rows Return per Page.	Optional

**psPassword**

Plain text password

**Return Values**

CResponseXml.bSuccess = True when Login is successful

CResponseXml.bSuccess = False when Login has failed

When login is successful

CResponseXML.oObj return array of string

**Table 4-96 Return Values**

Parameter	Type	Description	Required
oObj(0)	String	Session ID, you must use this ID when making a subsequence call.	Yes
oObj(1)	String(99000)	User Security Access, a string of "0" and "1".	Yes
oObj(2)	Boolean	True = Is Administrator.	Yes
oObj(3)	String	User Full Name.	Yes
oObj(4)	Integer	User Crew ID if link.	Yes
oObj(5)	Boolean	Must change the password before proceeding. If you do not change the password, you are not allowed to call any other function.	Yes
oObj(6)	Boolean	Allow to change the password.	Yes

If oObj(5) and oObj(6) both is True mean you should not allow the user to login.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("Login", "", New Object() {"admin",
"23f9d46d769b154c511f80a016784059"})
```

## Logout

This function logs out the user and the session ID will become invalid, you must call the “Login” function again in order to use the system.

### Input Parameters

psFunction= “Logout” do not have any parameter

### Return Values

CResponseXml.bSuccess always = True

### Example:

```
oResponse = goWS.FidelioSPMSWSXML("Logout", "SessionID", Nothing)
```

## MarkAsNoShow

This function marks the expected guest as no show.

### Input Parameters

psFunction= “MarkAsNoShow” — below is the list of psParam or poParam.

**Table 4-97 psParam**

Parameter	Type	Description	Required
pnAcclID	Integer	Unique Account ID.	Yes

### Return Values

CResponseXml.bSuccess = False when Update failed.

CResponseXml.bSuccess = True when Update successful.

### Example:

```
oResponse = goWS.FidelioSPMSWSXML("MarkAsNoShow", "SessionID", New
Object() {12345})
```

## MarkAssignmentStatus

This function updates all housekeeping assignment task status for a particular cabin or location for a selected date and port.

**Note:** Not supported when it is connected to an offline QCI Sync database.

### Input Parameters

psFunction=“MarkAssignmentStatus” — below is the list of psParam or poParam.

**Table 4-98 psParam**

Parameter	Type	Description	Required
pnHASstatus	Integer	Housekeeping Assignment Task Status (0-Pending, 1-Started, 2-Completed)	Yes
pblsCabin	Boolean	True – Cabin, False – Public Location.	Yes
psLocation	String	Cabin Number or Public Location Code.	Yes
pdAssignedDate	Date	Date of the Housekeeping Assignment Task.	Yes
pnPortID	Integer	Port ID. Linked to SCD.SCD_ID. A task that was created and linked to this port.	Yes
psHASComment	String	Assignment Comment.	Yes

**Return Values**

CRResponseXml.bSuccess = True when Update is successful

CRResponseXml.bSuccess = False when Update has failed

**Example:**

```
oResponse = goWS.FidelioSPMSWSXML("MarkAssignmentStatus", "SessionID", New
Object() {2,True,1024,<AssignedDate>,123, "Comment"})
```

## MarkSingleAssignmentStatus

This function updates a single housekeeping assignment task status for a particular housekeeping assignment record.

**Note:** Not supported when it is connected to an offline QCI Sync database.

**Input Parameters**

psFunction="MarkSingleAssignmentStatus" — below is the list of psParam or poParam

**Table 4-99 psParam**

Parameter	Type	Description	Required
pnHASstatus	Integer	Housekeeping Assignment Task Status (0-Pending, 1-Started, 2-Completed).	Yes
pblsCabin	Boolean	True – Cabin, False – Public Location.	Yes
psLocation	String	Cabin Number or Public Location Code.	Yes
pdAssignedDate	Date	Date of the Housekeeping Assignment Task.	Yes
pnPortID	Integer	Port ID. Linked to SCD.SCD_ID. A task that was created and linked to this port.	Yes
psHASComment	String	Assignment Comment.	Yes

**Return Values**

CResponseXml.bSuccess = True when Update is successful

CResponseXml.bSuccess = False when Update has failed

**Example:**

```
oResponse = goWS.FidelioSPMSWSXML("MarkAssignmentStatus", "SessionID", New
Object() {2,True,1024,<AssignedDate>,123, "Comment"})
```

## MessageDelete

This function flag a message as deleted in SPMS database.

**Note:** Not supported when it is connected to an offline QCI Sync database.

**Input Parameters**

psFunction= "MessageDelete" — below is the list of poParam.

**Table 4-100 psParam**

Code	Type	Description	Required
pnMsgID	Integer	Unique Message ID.	Yes

**Return Values**

CResponseXml.bSuccess = False when MessageDelete has failed

CResponseXml.bSuccess = True when MessageDelete is successful

**Example:**

```
oResponse = goWS.FidelioSPMSWSXML("MessageDelete", "Session ID", New
Object() {123})
```

## MessageRead

This function returns a list of messages from SPMS database.

**Note:** Not supported when it is connected to an offline QCI Sync database.

**Input Parameters**

psFunction= "MessageRead" — below is the list of poParam.

For Date field pass in 30-Dec-1899 if not required.

**Table 4-101 psParam**

Code	Type	Description	Required
psMsgDirection	String	S - All sent messages from the user that logged in. R – All received messages for the user that logged in.	Yes
pdMsgDateFrom	Date	Retrieve Messages from this date. Pass in DEFAULT_DATE if this is not required.	Yes
pdMsgDateTo	Date	Retrieve Messages until this date. Pass in DEFAULT_DATE if this is not required.	Yes
pnMsgProfileType	Integer	1=message to/from a guest. 2=message to/from another crew member (private messages). 3=Official ship messages including broadcasted alert messages.	Yes
pnMsgDeleted	Integer	Message deleted? (1-deleted, 0-not deleted yet).	Yes
psMsgModule	String	Messaging for which module (TA=TimeAttendance, HK=HouseKeeping, OTHER=for others).	Yes

**Return Values**

CResponseXml.bSuccess = False when MessageRead has failed

CResponseXml.bSuccess = True when MessageRead is successful, CResponse.oDataset will have 1 Data Table.

**Table 4-102 ADO.NET Data Table field structure for data table ordered by field TAMS\_MODDATE in descending order**

Code	Type	Description
TAMS_ID	Integer	Unique Message ID.
TAMS_CRW_ID	Integer	Message for this crew. Unique Crew identifier. Linked to RES.RES_ACC.
TAMS_FROM	Integer	Message from this crew. Unique user login identifier. Linked to USR.USR_ID
TAMS_SUBJECT	String	Message subject.
TAMS_MSG	String	Message Content.
TAMS_MSG_CREATED	Date	Message created date.
TAMS_READ	String	Message read? (1-read, 0-not read yet).
TAMS_READ_DATE	Date	Message read date.
TAMS_DELETED	String	Message deleted? (1-deleted, 0-not deleted yet).
TAMS_MSG_TYPE	Number	0-Individual message, 1-Broadcasted message.



**Table 4-102 (Cont.) ADO.NET Data Table field structure for data table ordered by field TAMS\_MODDATE in descending order**

Code	Type	Description
TAMS_MODULE	String	Messaging for which component (TA=TimeAttendance, HK=HouseKeeping, CPL=Complaints, CHKIN=Upon Checkin, CABCHG=Cabin Change).
TAMS_PROFILE	Number	Type of Message (1-Passenger, 2-Crew, 3-Ship).
TAMS_LEVEL	Number	Message Level (1-Normal, 2-High).
TAMS_TO_USR_ID	Number	Message for this person. Linked to USR.USR_ID.
TAMS_FROM_ACCID	Number	Message from this person. Linked to RES.RES_ACC.
TAMS_MODDATE	Date	Last modified date.
CRWNAME	String	Crew Member's Name.
CRW_ID	Number	Crew ID.
OPO_ID	String	Position ID.
OPO_POSITION	String	Position Name.
DEPTID	String	Department ID.
DEPTNAME	String	Department Name.
TAMS_EXTERNALID	Number	Linked to external record ID. (TAMS_MODULE='CPL',>>CPL_ID), (TAMS_MODULE='CHKIN',>>RES_ACC), (TAMS_MODULE='CABCHG',>>RES_ACC).
XDOC_FILE	Blob	Binary file contained inside the table that stored the document (Compressed).
XDOC_FILENAME	String	Name of the file.
XDOC_SIZE	Number	Size of file before compression.

**Example:**

```
oResponse = goWS.FidelioSPMSWSXML("MessageRead", "Session ID", New
Object() {'R', DateTime.SpecifyKind(DEFAULT_DATE,
DateTimeKind.Unspecified), DateTime.SpecifyKind(DEFAULT_DATE,
DateTimeKind.Unspecified), 2, 0, 'OTHER' })
```

## MessageSend

This function adds new messages created externally into SPMS database.

**Note:** Not supported when it is connected to an offline QCI Sync database.

**Input Parameters**

psFunction= "MessageSend", — below is the list of poParam.

For Date field pass in 30-Dec-1899 if not required.

**Table 4-103 psParam**

Code	Type	Description	Required
pnMsgID	Integer	Unique Message ID (TAMS_ID). Put 0 for adding new record.	Yes
pnFromUsrLogin	Long	Unique User ID of the crew that is sending the message. In SPMS, this is the USR.USR_ID value.	Yes
pnToUsrLogin	Long	Unique User ID of the crew that is receiving the message. In SPMS, this is the USR.USR_ID value.	Yes
psMsgSubject	String	Message subject.	Yes
psMsgContent	String	Message content.	Yes
pnProfileType	Integer	Message Profile Type. 1-Guest, 2-Crew, 3-Ship.	Yes
pnMsgLevel	Integer	Message Level. 1-Normal, 2-High.	Yes
pnMsgStatus	Integer	Message Status. 1-Sent, 2-Read, 3-Deleted.	Yes
pdMsgCreated	Date	Message created date & time.	Yes
psMsgModule	String	Messaging for which module (TA=TimeAttendance, HK=HouseKeeping, OTHER=for others).	Yes

**Return Values**

CResponseXml.bSuccess = False when MessageSend has failed

CResponseXml.bSuccess = True when MessageSend is successful

CResponsesXml.oObj = TAMS\_ID when bSuccess = True

**Example:**

```
oResponse = goWS.FidelioSPMSWSXML("MessageSend", "Session ID", New Object()
{123, 345, <Subject>, <MessageContent>, 2, 1, 2,
DateTime.SpecifyKind(DEFAULT_DATE, DateTimeKind.Unspecified) , 'OTHER'})
```

## MessageSetRead

This function flag a message as read in SPMS database.

**Note:** Not supported when it is connected to an offline QCI Sync database.

**Input Parameters**

psFunction="MessageSetRead" — below is the list of poParam.

**Table 4-104 psParam**

Code	Type	Description	Required
pnMsgID	Integer	Unique Message ID.	Yes

**Return Values**

CResponseXml.bSuccess = False when MessageSetRead has failed

CResponseXml.bSuccess = True when MessageSetRead is successful

**Example:**

```
oResponse = goWS.FidelioSPMSWSXML("MessageSetRead", "Session ID", New
Object() {123})
```

## OHCCreditCardAddRouting

This function identifies a list of guests to be routed to the payer with a registered credit card.

**Note:** Not supported when it is connected to an offline QCI Sync database.

**Input Parameters**

psFunction= "OHCCreditCardAddRouting" — below is the list of psParam or poParam.

**Table 4-105 psParam**

Parameter	Type	Description	Required
TerminalId	Integer	POS Terminal Identifier.	Yes
paBuyer	Array of String	Unique Buyer Account Numbers.	Yes
pnPayer	Integer	Unique Payer Internal Account Number.	Yes

**Return Values**

CResponseXml.bSuccess = False when input values are invalid or no record found.

CResponseXml.bSuccess = True and CResponseXML. CCreditCard when input values are valid.

**Table 4-106 CCreditCard field structure**

Code	Type	Description
TerminalId	Integer	POS Terminal Identifier which sent the request.

**Example:**

```
oResponse = goWS.FidelioSPMSWSXML("OHCCreditCardAddRouting", "SESSIONID", New
Object() {12345678, New String() {"1026604","1026605","1026606"} , 1026601})
```

## OHCCreditCardGetAmount

This function obtains the guest card type (debit/credit) and the pre-authorization/payment amount.

**Note:** Not supported when it is connected to an offline QCI Sync database.

### Input Parameters

psFunction= "OHCCreditCardGetAmount" — below is the list of psParam or poParam.

**Table 4-107 psParam**

Parameter	Type	Description	Required
TerminalId	Integer	POS Terminal Identifier.	Yes
pnPayer	Integer	Unique Payer Internal Account Number.	Yes
pnCardBin	Integer	First 6 digit (BIN) of the Payment Card Number (PAN).	Yes
TerminalMode	String	POS Mode: "UNATTENDED" or "ATTENDED".	Yes

### Return Values

CResponseXml.bSuccess = False when input values are invalid or no record found.

CResponseXml.bSuccess = True and CResponseXML. CCreditCard when input values are valid.

**Table 4-108 CCreditCard field structure**

Code	Type	Description
TerminalId	Integer	POS Terminal Identifier which sent the request.
gnTransID	Integer	Unique Transaction Reference ID.
gsCardType	String	"DEBIT" or "CREDIT" or "COMBO".
gsCurrency	String	Currency used onboard. Only "EUR" or "USD". If the currency is different from the one used by the POS, the POS will abort the current transaction. In that case, after the periodic function (TmlTweet) an automatic currency alignment procedure is performed by the POS (no need of any other FC action).
gnAmount1	Integer	PreAuthorization (only for Credit Card during registration) or Payment Amount. The amount is expressed In cents. For example, 125000 = 1.250,00.
gnAmount2	Integer	Second Payment Amount (only for Debit/Combo card) (in cents).
gnAmount3	Integer	Third Payment Amount (only for Debit/Combo card) (in cents).

**Table 4-108 (Cont.) CCreditCard field structure**

Code	Type	Description
gnAmount4	Integer	Fourth Payment Amount (only for Debit/Combo card) (in cents).
gnAmount5	Integer	Fifth Payment Amount (only for Combo card) (in cents).
gbDebitRefund	Boolean	Indicate if there is refund amount on debit card True – charge amount >= invoice balance. False – charge amount < invoice balance. POS in unattended mode: if <gbLastday> askfor a payment only if <gbDebitRefund> is false. POS in unattended mode: if not <gbLastDay> doesn't care about <gbDebitRefund>.
gsCruiseId	String	Cruise Identifier.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("OHCCreditCardGetAmount", "SESSIONID",
New Object() {12345678, 1026601, 411111, "UNATTENDED"})
```

## OHCCreditCardGuestSearch

This function searches the guest account by cruise card.

**Note:** Not supported when it is connected to an offline QCI Sync database.

### Input Parameters

psFunction= "OHCCreditCardGuestSearch" — below is the list of psParam or poParam.

**Table 4-109 psParam**

Code	Type	Description	Required
TerminalId	Integer	POS Terminal Identifier.	Yes
psSearchStr	String	Swiped MSC Cruise Card PAN within ';' and '?'. Example: 9841000099628100?	Yes

### Return Values

CResponseXml.bSuccess = False when input values are invalid or no record found.

CResponseXml.bSuccess = True and CResponseXML.CCreditCard when input values are valid.

**Table 4-110 CCreditCard field structure**

Code	Type	Description
TerminalId	Integer	POS Terminal Identifier which sent the request.

**Table 4-110 (Cont.) CCreditCard field structure**

Code	Type	Description
gnPayer	Integer	Unique Payer Account Number. It matches with one record (usually the first) in goAccount array (see below).
gsCabin	String	Guest Cabin Number.
gdEmbDate	Date	Guest Embarkation Date (DD/MM/YYYY).
gdDebDate	Date	Guest Debarkation Date (DD/MM/YYYY).
gbCardRegistration	Boolean	Identify if card registration is allowed.
gsCruiseId	String	Cruise Identifier.
gbRouting	Boolean	Identify if routing can be performed. True – routing allowed. False – routing not allowed (routing already done).
gbLastDay	Boolean	Indicate if today it's the last day cruise for the guest.
goAccount	Array of CGuest	Array of CGuest.

**Table 4-111 CGuest field structure**

Code	Type	Description
gnAccID	Integer	SPMS Internal Account Number.
gsFirstName	String	First Name.
gsLastName	String	Last Name.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("OHCCreditCardGuestSearch", "SESSIONID", New
Object() {12345678, ";9841000099628100"})
```

## OHCCreditCardTmlTweet

This function is used to monitor the POS mode and provide POS information.

**Note:** Not supported when it is connected to an offline QCI Sync database.

### Input Parameters

psFunction= "OHCCreditCardTmlTweet" — below is the list of psParam or poParam.

**Table 4-112 psParam**

Code	Type	Description	Required
TerminalId	Integer	POS Terminal Identifier.	Yes
IpAddress	String	Ip address assigned to POS.	Yes
ReleaseSW	String	SW version loaded into the POS.	Yes

**Table 4-112 (Cont.) psParam**

Code	Type	Description	Required
TerminalMode	String	POS Mode: "ATTENDED" or "UNATTENDED".	Yes
Currency	String	Only "USD" or "EUR".	Yes
PendingNumber	Integer	A number of pending transactions to be performed once the connection with Payment Server (GT) will be automatically established.	

**Return Values**

CResponseXml.bSuccess = False when input values are invalid or no record found.

CResponseXml.bSuccess = True and CResponseXML.TerminalNotification when input values are valid.

**Table 4-113 TerminalNotification field structure**

Code	Type	Description
TerminalId	Integer	POS Terminal Identifier which sent the request.
Settings	TAG Settings	TAG Settings (see below).

**Table 4-114 TAG Settings field structure**

Code	Type	Description
TerminalId	Integer	POS Terminal Identifier which sent the request.
Settings	TAG Settings	TAG Settings (see below).

**Example:**

```
oResponse = goWS.FidelioSPMSWSXML("OHCCreditCardTmlTweet", "SESSIONID",
New Object() {12345678, "[IP ADDRESS]", "01.10", "UNATTENDED", "EUR", 0})
```

## OHCCreditCardUpdateCardInfo

This function sends the results of pre-authorization (credit card registration) or payment transaction to SPMS.

**Note:** Not supported when it is connected to an offline QCI Sync database.

**Input Parameters**

psFunction= "OHCCreditCardUpdateCardInfo", — below is the list of psParam or poParam.

Table 4-115 psParam

Parameter	Type	Description	Required
TerminalId	Integer	POS Terminal Identifier.	Yes
pnAcclID	Integer	Unique Payer Internal Account Number (Equal to gnPayer of GuestSearchResponse).	Yes
pnTransID	Integer	Unique Transaction Reference ID (Equal to gnTransID of GetAmountResponse).	Yes
psCardExpMMYY	String	Card Expiry Date in MMYY.	
pnCDebitIndicator	Integer	0=Credit Card; 1=Debit Card.	Yes
psCaptureMethod	String	'I'= ICC (Chip); 'A'=Swiped (ICCFallback); 'M'=MoTo (pending executed).	Yes
psToken	String	Credit Card Token First 6 and last 4 digits are the same as the credit card number (PAN) If "pending transaction" is generated (psResult = "PENDING") Token is temporary (pseudoToken): For example, "411111999999999999999999999999991111" "	Yes
psSignature	String	Hex ASCII signature or Hex (jpg,bmp,png) format.	
pnAmount	Integer	Amount being authorized/settled The amount is expressed In cents. For example, 30000 = 300,00	Yes
psResult	String	Update Card Info Status: "PENDING" = only for a credit card if, during a registration, the online connection to the Payment Server (GT) is not possible. psToken is temporary. "APPROVE" = transaction (debit or credit, auth or direct sale) approved. "DECLINE" = transaction (debit or credit, auth or direct sale) not approved/aborted.	Yes
psTransType	String	'AUTH'=Pre-authorization (only credit card). 'PAYMENT'=Direct sales for credit/debit card.	Yes
psAuthCode	String	Authorization Code.	
pnSTAN	Integer	STAN of the transaction. Sequential Transaction Number in Payment Gateway (GT).	
psReceipt	String	Transaction Receipt.	
pnCardBin	Integer	First 6 digits (BIN) of Payment Card Number (PAN).	Yes



**Table 4-115 (Cont.) psParam**

Parameter	Type	Description	Required
psCurrency	String	The currency used for the transaction. Only 'USD' or 'EUR'	
pnActionCode	Integer	Action Code.	
psAcquirerName	String	Acquirer/Card name.	
psDateTime	String	Transaction Date and Time in the following format: YYYY-MM-DDThh:mm:ssZ. For example, "2014-11-28T15:17:46Z" Note that the Date and Time are of the POS/GT, not the current ones (on the vessel).	

**Return Values**

CResponseXml.bSuccess = False when input values are invalid or no record found.

CResponseXml.bSuccess = True and CResponseXML. CCreditCard when input values are valid.

**Table 4-116 CCreditCard field structure**

Code	Type	Description
TerminalId	Integer	POS Terminal Identifier which sent the request.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("OHCCreditCardUpdateCardInfo",
"SESSIONID", New Object() {12345678, 1026604, 171859, "", 0, "I",
"4111111EF6BBE22235E0AA0D7E43F1111", "", 25000, "APPROVE", "AUTH",
"x12345", 000002, "", 411111})
```

## PayCrewInvoice

This function searches for folio deduction for payrolls using various criteria from the SPMS system.

**Input Parameters**

psFunction= "PayCrewInvoice" — below is the ordered list of input parameters and its descriptions

**Table 4-117 psParam**

Parameter	Type	Description	Required
UXP_A_ID	Integer	Required input. Unique identifier of UXP record - UXP_A_ID.	Yes

Table 4-117 (Cont.) psParam

Parameter	Type	Description	Required
BTD	String	Required input. Balance-to-date specifier. If TRUE, folio deduction will be based on up-to-date payroll balance. If FALSE, folio deduction will be based on total payroll end-balance supposedly available at the end of the month.	Yes
AmountPayable	Decimal	Required input. The amount of folio deduction.	Yes
Currency	String	Required input. Currency sign. Can use the currency sign return from previous "QueryCrewInvoice" function call.	Yes

**Return Values**

CResponseXml.bSuccess = False when WS function did not execute successfully. For example, due to invalid input parameters.

CResponseXml.bSuccess = True when WS function executed successfully, and

CResponseXML.oDataSet = *ADO.NET dataset*. The dataset will contain table SSH and SPP of the payroll used in folio deduction. \*

Remark:

See SPMS Data Dictionary for details of the fields.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("PayCrewInvoice", "SessionID", New Object()
{12345, True, 250.00, "USD"})
```

## PurchasePackage

This function is used for Package Plan purchase. Please refer to FCWSFunction Technical Specification, GetPackagePurchase function on how to obtain a list of package preset for user selection.

**Note:** Not supported when it is connected to an offline QCI Sync database.

**Input Parameters**

psFunction= "PurchasePackage" — below is the list of psParam or poParam.

Table 4-118 psParam

Parameter	Type	Description	Required
pnAccID	Integer	Unique Account ID.	Yes
pnPpnID	Integer	Unique Package Plan ID.	Yes

**Table 4-118 (Cont.) psParam**

Parameter	Type	Description	Required
pnRvc	Integer	MICROS POS Revenue Center Number.	Yes
psRvcName	String(30)	MICROS POS Revenue Center Name.	Yes
pnEmpl	Long	MICROS POS Employee Number.	Yes
pnWSID	Long	MICROS POS Workstation ID.	Yes
pbyImage	Byte Array	Signature Image.	Optional
psTeam1	String(12)	Posting additional information - Server number 1	Optional
pnChkEmplNum	Long	Employee number on initial order	Optional

**Return Values**

CResponseXml.bSuccess = False when Purchase failed.

CResponseXml.bSuccess = True when Purchase successful. CResponseXml.oObj return the check image for printing

**Example:**

```
oResponse = goWS.FidelioSPMSWSXML("PurchasePackage", "SessionID", New
Object() {12345,123457,1,"Restaurant",1234,1,<Signature Image byte array>})
```

Use WSFunction->GetPackagePurchase to get a list of available package to purchase

## QueryCrewInvoice

This function returns crew member's invoice information along with information about his/her available payroll credit for folio deduction.

**Input Parameters**

psFunction= "QueryCrewInvoice" — below is the ordered list of input parameters and its descriptions.

**Table 4-119 psParam**

Parameter	Type	Description	Required
UXP_A_ID	Integer	Required input. Unique identifier of UXP record - UXP_A_ID.	Yes
BTD	String	Required input. Balance-to-date specifier. If TRUE, will return up-to-date payroll balance available for folio deduction. If FALSE, will return total payroll end-balance supposedly available at the end of the month.	Yes

**Return Values**

CResponseXml.bSuccess = False when WS function did not execute successfully. For example, due to invalid input parameters.

CResponseXml.bSuccess = True when WS function executed successfully, and

CResponseXML.oDataSet = ADO.NET dataset. The dataset will contain table SSH and SPP of the payroll to be used for folio deduction. \*

CResponseXML.oCrewInvoice(0) = information about available payroll credit for folio deduction.

- Amount =  $uxp\_a\_pos\_debit - uxp\_a\_pos\_credit$ , total invoice amount in onboard currency.
- AmountPayable = invoice amount, in wage currency, that can be settled via folio deduction.
- Currency = onboard currency.
- AvailPay = available payroll amount or credit.
- WageCurrency =  $CTR\_WAGE\_CUR\_TYP * CurrencyConvRate$  = onboard to wage currency conversion rate.

Remark:

See SPMS Data Dictionary for details of the fields.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("QueryCrewInvoice", "SessionID", New Object()
{12345, True})
```

## RegisterRFIDBoardCard

This function associates a RFID board card to a guest for the purpose of door key encoding at door lock with Visionline System. This feature is applicable to cards encoded externally before embarkation day. The door key encoding will happen at the door lock only when the guest has checked in and the card is used at the lock.

The guest to be registered must have an expected reservation. Guests that are checked-in, checked-out or with a canceled reservation will be rejected. Also, guests already issued with a door-key encoded board card will be rejected as well.

In the event that the card belongs to another person, the function will also reject the registration. The only time such card can be re-registered under the new person is when the card has not been encoded with door key and the person is currently not checked-in.

When registration is successful, it will override the previously registered board card for the guest.

**Note:** The above validation is not applicable when connected to offline QCI Sync DB.

**Input Parameters**

psFunction= "RegisterRfidBoardCard" — below is the list of psParam or poParam.

**Table 4-120 psParam**

Parameter	Type	Description	Required
pnAcclID	Long	Unique Payer Account ID.	Yes
psUID	String	UID of The RFID Board Card.	Yes

**Return Values**

CResponseXml.bSuccess = False when Update failed.

CResponseXml.bSuccess = True when Update successful.

This function supports enhanced error messaging:

CResponseXml.sErrorMsg: detailed long description of the error.

CResponseXml.sErrWord: short brief description of error title.

CResponseXml.sErrCode: 6-CHAR or less error code.

**Example:**

```
oResponse = goWS.FidelioSPMSWSXML("RegisterRfidBoardCard", "SessionID",
New Object() {12345,"CD125E84"})
```

## RegisterRFIDBoardCardQuery

This function queries SPMS on the passed in UID before "RegisterRfidBoardCard" is called to make sure "RegisterRfidBoardCard" will be successful.

**Input Parameters**

psFunction= "RegisterRfidBoardCardQuery" — below is the list of psParam or poParam.

**Table 4-121 psParam**

Parameter	Type	Description	Required
pnAcclID	Long	Unique Payer Account ID.	Yes
psUID	String	UID of The RFID Board Card.	Yes

**Return Values**

CResponseXml.bSuccess = False when Update failed.

CResponseXml.bSuccess = True when Update successful.

This function supports enhanced error messaging:

CResponseXml.sErrorMsg: detailed long description of the error.

CResponseXml.sErrWord: short brief description of error title.

CResponseXml.sErrCode: 6-CHAR error code. (can be less than 6-char).

Example:

```
oResponse = goWS.FidelioSPMSWSXML("RegisterRfidBoardCardQuery", "SessionID", New
Object() {12345,"CD125E84"})
```

## RetrievePage

This function searches for a guest using various criteria from the SPMS system.

### Input Parameters

psFunction= "GuestSearch" — below is the list of psParam or poParam.

**Table 4-122 psParam**

Parameter	Type	Description	Required
pnPage	Integer	Page Number.	Yes
psFunction	String	Last call function.	Yes
psWSFunction	String	Need to pass in when psFunction = 'WSFunction.'	Optional

### Return Values

CResponseXml.bSuccess = False when RetrievePage has failed.

CResponseXml.bSuccess = True when RetrievePage is successful.

Other value depend on the psFunction and psWSFunction, refer to the psFunction for detail.

## SearchAmenity

This function is an advanced amenity search.

**Note:** Not supported when it is connected to an offline QCI Sync database.

### Input Parameters

psFunction= "SearchAmenity" — below is the list of poParam

Please take note some parameters are optional. However, the more parameters passed in, the fewer records it will return.

For Date field pass in 30-Dec-1899 if not required.

For Array field, pass in an empty array or "Nothing" to skip this checking.

For JSON, the array of string will be a string separated by ",", each string must start and end with a double quotation (").

For example,

"A","" ,"B," = 2 Element

Element 1 = A,

Element 2 = B,

**Table 4-123 psParam**

Parameter	Type	Description	Required
peReservationType	Integer	ReserveAndCheckIn, 1 – Reserve, 2- CheckIn, 3- CheckOut.	
pdExpDeliveryDateFrom	Date	Amenity Delivery Date (AME_DDATE)If nothing is passed in, then it will skip this criteria.	
pdExpDeliveryDateTo	Date	Amenity Delivery Date (AME_DDATE)If nothing is passed in, then it will skip this criteria.	
pdDeliveredDateFrom	Date	Amenity Service Done Date (AME_DONE_DATE)If nothing is passed in, then it will skip this criteria.	
pdDeliveredDateTo	Date	Amenity Service Done Date (AME_DONE_DATE)If nothing is passed in, then it will skip this criteria.	
psGuestLastName	String	Guest Last Name (UXP_A_NAME)If nothing is passed in, then it will skip this criteria.	
psGroupID	Array of String	Travel Group ID (UXP_A_ID)If nothing is passed in, then it will skip this criteria.	
psCabinCategory	Array of String	Cabin Category (TYP_CAT.TYP_ART)If nothing is passed in, then it will skip this criteria.	
psDeliveryLocation	Array of String	Amenity Delivery Location (AME_LOCATION)If nothing is passed in, then it will skip this criteria.	
psDeck	Array of String	Cabin Deck (TYP_DEK.TYP_ART)If nothing is passed in, then it will skip this criteria.	
psDeliveredBy	Array of String	Amenity Delivery By (USR.USR_ID)If nothing is passed in, then it will skip this criteria.	
psCabin	Array of String	Unique Cabin IDs (CAB.CAB_ID)If nothing is passed in, then it will skip this criteria.	
psAmenityCode	Array of String	Amenity Code (TYP_AMT.TYP_ART)If nothing is passed in, then it will skip this criteria.	
psDepartment	Array of String	Amenity Delivery Division (AME_ODI_ID)If nothing is passed in, then it will skip this criteria.	
psHKSection	Array of String	Housekeeping Section Name (HSC_NAME)If nothing is passed in, then it will skip this criteria.	

**Return Values**

CResponseXml.bSuccess = False when SearchAmenity has failed

CResponseXml.bSuccess = True when SearchAmenity is successful, CResponse.oDataset will have 1 Data Table.

**Table 4-124 ADO.NET Data Table field structure**

Code	Type	Description
AME_ID	Integer	Amenity ID (Unique ID).
AMECODE	String	Amenity Code. Linked to TYP_AMT.TYP_ART.
AMENAME	String	Port Name. Linked to TYP_AMT.TYP_COMMENT.
ACCID	Integer	Account ID.
NAME	String	Guest Name.
RESSTATUS	String	Reservation Status.
VIPCODE	String	VIP ID. Linked to TYP_VIP.TYP_ART.
VIPNAME	String	VIP Name. Linked to TYP_VIP.TYP_COMMENT.
PORTID	String	Embarkation Port ID.
PORTNAME	String	Embarkation Port Name. Linked to TYP_HRB.TYP_COMMENT.
GROUPID	Integer	Travel Group Reference ID.
GROUPNAME	String	Travel Group Name.
CABINID	String	Unique Cabin ID.
HSC_NAME	String	Housekeeping Section Code.
HSC_COMMENT	String	Housekeeping Section Name.
AME_LOCATION	String	Amenity Delivery Location.
AME_ODI_ID	String	Amenity Delivery Division ID. Linked to TYP_ODI.TYP_ART.
TYP_COMMENT	String	Amenity Delivery Division Name. Linked to TYP_ODI.TYP_COMMENT.
CAB_VZONE	String	Cabin vertical zone. Linked to TYP_VTZ.TYP_ART.
CAB_VZONE_NAME	String	Cabin vertical zone description. Linked to TYP_VTZ.TYP_COMMENT.
CAB_TYPE	String	Cabin Type ID. Linked to TYP_CAT.TYP_ART.
TYP_COMMENT1	String	Cabin Type Name. Linked to TYP_CAT.TYP_COMMENT.
CAB_DECK	String	Cabin Deck ID. Linked to TYP_DEK.TYP_ART.
TYP_COMMENT2	String	Cabin Deck Name. Linked to TYP_DEK.TYP_COMMENT.
AME_DDATE	Date	Amenity Delivery Date.
AME_DONE_DATE	Date	Amenity Service Done Date.
AME_DONE_USERID	Integer	Amenity Service done by the waiter. Linked to USR.USR_ID.
USR_LOGIN	String	User Login Name.



**Table 4-124 (Cont.) ADO.NET Data Table field structure**

Code	Type	Description
CAB_KEEPER	Integer	Housekeeper ID. Linked to RES.RES_ACC.
CAB_KEEPER_NAME	String	Housekeeper Name.
CAB_A_KEEPER	Integer	Assistant Housekeeper ID. Linked to RES.RES_ACC.
CAB_A_KEEPER_NAME	String	Assistant Housekeeper Name.
CAB_G_WAITER	Integer	Waiter ID. Linked to RES.RES_ACC.
CAB_G_WAITER_NAME	String	Waiter Name.
CAB_A_WAITER	Integer	Assistant Waiter. Linked to RES.RES_ACC.
CAB_A_WAITER_NAME	String	Assistant Waiter Name.
AME_AMOUNT	Double	Amenity Price.
AME_COMMENTS	String	Amenity Comments.
AME_COST	Integer	Amenity Cost.
AME_CREATED_BY	Integer	User Created Amenity Request.
AME_CREATED_DATE	Date	Amenity Created Date.
AME_CURR_ID	String	Amenity price currency. Linked to TYP_CUR.TYP_ART.
AME_EXT_CODE	String	External Brochure Code.
AME_EXT_ID	String	External ID.
AME_FLG	String	Displayed as a flag in the Amenities section in guest billing screen. Not been referenced anywhere else.
AME_MESSAGE	String	Message From Sender.
AME_MODDATE	Date	Last Modified Date.
AME_NOTE	String	Amenity Description.
AME_OCCURRENCE	String	Amenity Deliver Occurrence.
AME_QUANTITY	Integer	Amenity Quantity.
AME_RESBOOKNR	String	External Reservation Booking Number. Linked to RES.RES_BOOKNR.
AME_RESID	Integer	Reservation account ID. Linked to RES.RES_ID.
AME_SENDER1	String	Amenity Sender.
AME_SENDER2	String	Amenity Sender 2 (Not been referenced).
AME_V_GUESTID	Integer	Reservation System account ID. Linked to RES.RES_V_GUESTID.
AME_VID	String	Unique ID.

**Example:**

```
oResponse = goWS.FidelioSPMSWSXML("SearchAmenity", New Object()
{<ReservationType>,<ExpDeliveryDateFrom>,<ExpDeliveryDateTo>,<DeliveryDate
From>,<DeliveryDateTo>,<LastName>,<GroupID>,<CabinCategory>,<DeliveryLocat
ion>,<Deck>,<DeliveryBy>,<CabinID>,<AmeCode>,<Department>,<HKSection>})
```

## SearchCabin

This function performs an advanced search on cabins in SPMS. If a date is passed in, then only the list of crash cabins found for that date will be returned.

**Note:** Not supported when it is connected to an offline QCI Sync database.

### Input Parameters

psFunction= "SearchCabin — below is the list of poParam.

Please note that all the parameter is "Optional" and the more parameters pass in, the fewer records it will return.

For Array field, pass in an empty array or "Nothing" to skip this checking.

For Date field pass in 30-Dec-1899 if it is not required.

For JSON, the array of string will be a string separated by ",", each string must start and end with a double quotation mark ("").

For example,

"A,","B," = 2 Element

Element 1 = A,

Element 2 = B,

**Table 4-125 psParam**

Parameter	Type	Description	Required
psCabin	Array of String	Unique Cabin IDs.	Yes
psCabinHKStatus	Array of String	Cabin Housekeeping Status (OD, OC, VD, VC).	Yes
psCabinAvlStatus	Array of String	Cabin Availability (OO, OV, BL, AV).	Yes
psHKSection	Array of String	Housekeeping section codes.	Yes
psDeck	Array of String	Cabin Deck (TYP_DEK.TYP_ART).	Yes
psCabinCategory	Array of String	Cabin Category (TYP_CAT.TYP_ART).	Yes
psCabinType	Array of String	'P' for Pax, 'C' for Crew.	Yes
psCabinVerticalZone	Array of String	Cabin Vertical Zone (TYP_VTZ.TYP_ART).	Yes
pdCrashCabinDate	Date	Date to check for crash cabins. If do not require to check for crash cabins please pass in 30-Dec-1899.	Yes
psCabinSpecialNeeds	Array of String	0-Normal Cabin,1-Cabin with special need feature.	Yes
psCabinFeatures	Array of String	Features available in the cabin. Linked to TYP_CAF.TYP_ART.	Yes

### Return Values

CResponseXml.bSuccess = False when SearchCabin has failed.

CResponseXml.bSuccess = True when SearchCabin is successful,  
CResponse.oDataset will have 2 Data Table.

**Table 4-126 ADO.NET Data Table field structure for data table (“CAB”). This lists the cabins and its details.**

Code	Type	Description
CAB_ID	String	Unique Cabin ID.
CAB_DESIGN	String	'P' for Pax, 'C' for Crew.
CAB_STATE	String	Cabin Availability Status (OO=Out of Order, OS=Out of Service, BL=Blocked, AV=Available).
CAB_BEDS	Number	No of beds/berths in the cabin.
CAB_BEDS_SETUP	String	Beds setup for each cabin. Format saved A,B,C,D,E or 1,2,3,4,5.
CAB_TEL	String	Cabin Telephone Number.
CAB_TEL2	String	Cabin Telephone Number 2.
CAB_BEG_BLOCK_D	Date	The cabin is pending repairs, this field stores the date for when this cabin is blocked from. This is the 'Block From' field in FC Admin Cabin setup.
CAB_REPAIR_D	Date	The cabin is pending repairs, this field stores the date for when this cabin is blocked until. This is the 'Block Until' field in FC Admin Cabin setup.
CAB_REPAIR_T	String	The cabin is pending repairs, this field stores the unavailability reason when CAB_STATE=OO or OS or BL. This is the 'Unavailability Reason' field in FC Admin Cabin setup.
CAB_LIFEBOAT	String	Lifeboat this cabin is assigned to. TYP_BOT.TYP_ART.
CAB_LIFEBOAT_NAME	String	Lifeboat this cabin is assigned to. TYP_BOT.TYP_COMMENT.
CAB_STATION	String	Muster Station this cabin is assigned to. Linked to MST.MST_ID.
CAB_STATION_NAME	String	Muster Station this cabin is assigned to. Linked to MST.MST_NAME.
CAB_LOCATION	Integer	Null-Undefine,1-Forward,2-Mid Ship,3-Aft
CAB_STARBOARD	Integer	0-Portside,1-Starboard.
CAB_CAT	String	Cabin Type. Linked to TYP_CAT.TYP_ART.
CAB_CAT_NAME	String	Cabin Type Description. Linked to TYP_CAT.TYP_COMMENT.
CAB_DECK	String	Cabin Deck. Linked to TYP_DEK.TYP_ART.
CAB_DECK_NAME	String	Cabin Deck Description. Linked to TYP_DEK.TYP_COMMENT.
CAB_HKSTATUS	String	Cabin Housekeeping status (OD, VD, OC, VC). Linked to TYP_CCS.TYP_ART.
CAB_HKSTATUS_NAME	String	Cabin Housekeeping status description. Linked to TYP_CCS.TYP_COMMENT.
CAB_VZONE	String	Cabin vertical zone. Linked to TYP_VTZ.TYP_ART.

**Table 4-126 (Cont.) ADO.NET Data Table field structure for data table (“CAB”). This lists the cabins and its details.**

Code	Type	Description
CAB_VZONE_NAME	String	Cabin vertical zone description. Linked to TYP_VTZ.TYP_COMMENT.
CAB_CRW_POSITION	String	Assign default crew position to a cabin.
CAB_KEEPER	Integer	Unique Crew ID. Housekeeper assigned to this cabin for gratuity distribution. Linked to RES.RES_ACC. This is the 'Keeper' field in the FC Admin Cabin setup.
CAB_KEEPER_NAME	String	Keeper's Crew Name.
CAB_A_KEEPER	Integer	Unique Crew ID. The assistant housekeeper that is assigned to this cabin for gratuity distribution. Linked to RES.RES_ACC. This is the 'Assistant Keeper' field in the FC Admin Cabin setup.
CAB_A_KEEPER_NAME	String	Assistant Keeper's Crew Name.
CAB_H_WAITER	Integer	Unique Crew ID. Head Waiter that is assigned to this cabin for gratuity distribution. Linked to RES.RES_ACC. This is the Head Waiter' field in the FC Admin Cabin setup.
CAB_H_WAITER_NAME	String	Head Waiter's Crew Name.
CAB_A_WAITER	Integer	Unique Crew ID. The assistant waiter that is assigned to this cabin for gratuity distribution. Linked to RES.RES_ACC. This is the 'Assistant Waiter' field in the FC Admin Cabin setup.
CAB_A_WAITER_NAME	String	Assistant Waiter's Crew Name.
CAB_G_WAITER	Integer	Unique Crew ID. The waiter that is assigned to this cabin for gratuity distribution. Linked to RES.RES_ACC. This is the Waiter' field in the FC Admin Cabin setup.
CAB_G_WAITER_NAME	String	Waiter's Crew Name.
HKSECTION_NAME	Integer	Housekeeping Section Name. Linked to HSC.HSC_NAME.
HKSECTION_COMMENT	String	Housekeeping Section Comment. Linked to HSC.HSC_COMMENT.
HSC_CRW	String	Housekeeping Crew ID. List of Housekeeping crew ID that can be assigned housekeeping tasks.
CAB_SPECIALNEED	Integer	0-Normal Cabin,1-Cabin with special need feature.
CAB_FEATURE	String	Features available in the cabin. Linked to TYP_CAF.TYP_ART.

# SearchCntrAssgn

## Input Parameters

psFunction= "SearchCntrAssgn" — below is the ordered list of input parameters and its descriptions.

**Table 4-127 psParam**

Parameter	Type	Description	Required
MemoryLifeSpan	Integer	Optional in seconds, default as zero. If set to non-zero, resultset will be kept in memory for reuse when a subsequent call is issued within the specified time interval with MemoryReuse = TRUE.	Optional
MemoryReuse	Boolean	Optional, default as FALSE. When set to TRUE, the search will not query DB but instead just return records from the previous result set.	Optional
RecordPerPage	Integer	Optional, default as zero. Specify the number of unique person records for which their contracts will be returned. If it is zero, it will return all records in the result set.	Optional
PageNum	Integer	Optional, default as zero for page 1. Specify the page to return in a multi-page result set.	Optional
UXP_A_IDs	String	Optional. Comma delimited UXP_A_ID.	Optional
SearchStr	String	Optional. If specified, will be used to search based on following fields. The order of following also determines the sequence in which the search is carried out until a match is received. RES_BOARDCC (exact match), RES_CC_NUMBER (exact match), RES_CAB (exact match), UXP_A_NAME_N (partial match), and UXP_C_EXTERNALID or UXP_C_INTERNALID (exact match) depending on PAR "Display UXP_C_EXTERNALID As Crew ID".	Optional
ORDER_BY	String	Optional. Only supports UXP, RES and CTR column ordering.	Optional
RES_STATUSES	String	Optional. Comma delimited RES_STATUS. *	Optional
EMB_DATE_01	Date	Optional. EMB_DATE_01 and EMB_DATE_02 specify the date range of NVL(res_emb_a,res_emb_e). Or CTR_BEG_DATE if NOB.	Optional

Table 4-127 (Cont.) psParam

Parameter	Type	Description	Required
EMB_DATE_02	Date	Optional. EMB_DATE_01 and/or EMB_DATE_02 specify the date range of NVL(res_emb_a,res_emb_e). Or CTR_BEG_DATE if NOB.	Optional
DEB_DATE_01	Date	Optional. DEB_DATE_01 and/or DEB_DATE_02 specify the date range of NVL(res_dis_a,res_dis_e). Or CTR_END_DATE if NOB.	Optional
DEB_DATE_02	Date	Optional. DEB_DATE_01 and/or DEB_DATE_02 specify the date range of NVL(res_dis_a,res_dis_e). Or CTR_END_DATE if NOB.	Optional
CTR_STATUSES	String	Optional. Comma delimited CTR_STATUS. *	Optional
SGR_IDs	String	Optional. Comma delimited SGR_ID.	Optional
CTR_TYP_CTRs	String	Optional. Comma delimited CTR_TYP_CTR.	Optional
CTR_NOB_FLAG	String	Optional. "1" for NOB assignments. "0" for normal assignments.	Optional
CTR_PCY_IDs	String	Optional. Comma delimited CTR_PCY_ID.	Optional
CTR_XCET_IDs	String	Optional. Comma delimited CTR_XCET_ID.	Optional
UXP_H_NATIONS	String	Optional. Comma delimited UXP_H_NATION.	Optional
ENDING_PYRL	Boolean	Optional. If TRUE, will return assignment with pending final month payroll.	Optional

**Return Values**

CResponseXml.bSuccess = False when WS function did not execute successfully. For example, due to invalid input parameters.

CResponseXml.bSuccess = True when WS function executed successfully, along with

CResponseXML.oDataSet = Nothing, when no record is found. Otherwise,

CResponseXML.oDataSet = *ADO.NET dataset*, which will contain table ACC (UXP/RES), CTR, SPD, COH, PCY, SED, SPO and XCET. \*

CResponseXML.nTotalPage = total number of pages in the resultset, when input parameter *RecordPerPage* is specified.

Remark:

See SPMS Data Dictionary table for details of the fields.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("SearchCntrAssgn", "SessionID", New Object()
{60, True, 10, 1, "", "123", "UXP_C_EXTERNALID", "RR,RE,RX,RN,CC,CL,DD", _
```

```
"", "", "", "", "A,C", "", "", "", "0", "", "", "", False}}
```

## SearchComment

This function searches for the guest comment.

**Note:** Not supported when it is connected to an offline QCI Sync database.

### Input Parameters

psFunction= "SearchComment" — below is the list of poParam.

For Array field, pass in an empty array or "Nothing" to skip this checking.

For JSON, the array of string will be a string separated by ",", each string must start and end with a double quotation mark ("").

For example,

"A","B," = 2 Elements

Element 1 = A,

Element 2 = B,

**Table 4-128** psParam

Code	Type	Description	Required
pdFrom	Date	From date range.	Yes
pdTo	Date	To date range.	Yes
psCabin	Array of String	List of Cabin.	Optional
pnUsrID	Array of Long	List of user who created the comment.	Optional
psComplaintCode	Array of String	List of Comment Description Code (TYP_CPL.TYP_ART).	Optional
psReportType	Array of String	List of Report Type Code (TYP_CPI,TYP_ART).	Optional
psCategory	Array of String	List of Comment Category (TYP_CMC.TYP_ART).	Optional
peDone	Integer	0 - All, 1 - Done, 2 - Not Done.	Optional
peUrgent	Integer	0 – All, 1 – Urgent, 2 – Not Urgent.	Yes
peSensitive	Integer	0 – All, 1 – Sensitive, 2 – Not Sensitive.	Yes
peGuestType	Integer	0 – All, 1 – Crew, 2 – Staff, 3 – Crew and Staff, 4 - Guest.	Yes
psDeptInvolved	Array of String	Department Involved code (TYP_CDI.TYP_ART).	Yes
psOperationPosition	Array of String	Operation Position (OPO_ID), operation position is link to department involved.	Optional

**Table 4-128 (Cont.) psParam**

Code	Type	Description	Required
pbIncludePicture	Boolean	True = Will return the comment picture/ document.  False = Will return nothing, this is for slow device, so it will need to look up the picture using WSF using the value CPL_XDOC_ID, default = True.	Optional
pnAccID	Array of Integer	List of Account ID.	Optional

**Return Values**

CResponseXml.bSuccess = False when SearchComment has failed.

CResponseXml.bSuccess = True when SearchComment is successful, CResponse.oDataset will have 1 Data Table if paging is not use 2 Data Table is paging is use (Login.pnRowsPerPage > 0), please take note all the same cpl\_link\_id will be return with the exception of psCabin and pnAccID.

**Table 4-129 ADO.NET Data Table field structure for data table for table 1 when paging is in use**

Code	Type	Description
CPL_LINK_ID	Integer	Link ID between the parent comment and child comment and resolution and also link to multiple guest comment.
CPL_ACC_ID	Integer	Link ID between the parent comment and child comment and resolution and also link to multiple guest comment.
CPL_RES_DONE	Integer	0 – Not Resolved, 1 - Resolved.

**Table 4-130 ADO.NET Data Table field structure for table 1 - paging not in use, table 2 - paging in use.**

Code	Type	Description
CPL_ID	Integer	Unique ID.
CPL_LINK_ID	Integer	Link ID between the parent comment and child comment and resolution and also link to multiple guest comment.
CPL_ACC_ID	Integer	Account ID.
CPL_USR_ID	Long	User who create the comments/resolution.
CPL_TYPE	String	C-Comment, R-Resolution.
CPL_DATE	Date	Date/Time the comment/resolution created.
CPL_EXT_ID	String	External Code.



**Table 4-130 (Cont.) ADO.NET Data Table field structure for table 1 - paging not in use, table 2 - paging in use.**

Code	Type	Description
CPL_DESCR_CODE	String	Comment Description Code (TYP_CPL.TYP_ART) Resolution Description Code (TYP_RES.TYP_ART).
CPL_DESCR	String	Comment/Resolution Description.
CPL_COMMENT	String	Comment/Resolution Comment.
CPL_REQ_RES_DATE	Date	Requested Resolution Date.
CPL_RES_DONE	Integer	0 – Not Resolved, 1 - Resolved.
CPL_RES_BY	String	Person who done the resolution.
CPL_RES_DATE	Date	Resolution Resolved Date.
CPL_RES_COST	Decimal	Comment/Resolution Associate Cost.
CPL_REPORT_TYPE	String	Define Comment Reporting Method.
CPL_URGENT	Integer	0-Default;1-Urgent.
CPL_CRW_ID	Integer	The Crew link to the CPL_USR_ID.
CPL_SENSITIVE	Integer	0-Default, 1-Sensitive.
CPL_TYP_CMC	String	Comment Category Code (TYP_CMC.TYP_ART).
CPL_XDOC_ID	Integer	Scan Document/Picture, Link to xdoc_id.
CPL_VID	String	External Unique ID for use with import.
CPL_VEN_ID	Integer	Vendor ID, link to VEN.VEN_ID
CPL_INTERNAL	Integer	Internal Usage Flag
CPL_DEPARTMENT_INVOLVE D	String	Comment Department Involved Codes. Separated by comma (,) (TYP_CDI.TYP_ART).
CRW_NAME	String	Crew Name who created the comment/resolution base on the crew link in user setup.
UXP_A_TAG	String	P-Pax, C-Crew or Staff.
NAME	String	Guest Name.
RES_CAB	String	Cabin Number.
CATEGORY	String	Comment Category Description.
REPORTTYPE	String	Comment Report Type Description.
DESCRCODE	String	Comment Description Code Description.
XDOC_FILE	Array of Byte	The doc/picture uploaded.
XDOC_FILENAME	String	The filename of the doc/picture, use the file extension to determine if it is a document or picture.
XDOC_ISPICTURE	Integer	1-This is a picture,0-This is a document, need to use the XDOC_FILENAME to determine the document type.

Example:

For example, this is to search all comment for cabin 05000 and 05002

```
oResponse = FidelioWS("SearchComment", New Object() {DateTime.SpecifyKind(<From
Date>, DateTimeKind.Unspecified), _
DateTime.SpecifyKind(<To Date>, DateTimeKind.Unspecified), _
New String() {"05002", "05000"}, Nothing, Nothing, Nothing, _
Nothing, 0, 0, 0, 0, Nothing, Nothing, False, False, Nothing})
```

## SearchCrashCabin

This function searches for 'crash cabins' in SPMS. Crash Cabin is defined as guests who checks in and checks out into the same cabin on the same day and port.

**Note:** Not supported when it is connected to an offline QCI Sync database.

### Input Parameters

psFunction= "SearchCrashCabin" — below is the list of poParam.

For Date field pass in 30-Dec-1899 if not required.

**Table 4-131 psParam**

Code	Type	Description	Required
pdDateSelected	Date	Date of check-in/check-out for searching crash cabins. If 30-Dec-1899 is passed in, then it will search based on SPMS system date.	Yes
psPortID	String	Port ID. Linked to SCD.SCD_PORT_ID. If an empty string is passed in, then it will search for all ports for pdDateSelected.	Yes

### Return Values

CResponseXml.bSuccess = False when SearchCrashCabin has failed

CResponseXml.bSuccess = True when SearchCrashCabin is successful,  
CResponse.oDataset will have 1 Data Table.

**Table 4-132 ADO.NET Data Table field structure for data table.**

Code	Type	Description
CABIN	String	Unique Cabin ID.
SCD_PORT_ID	String	Port Name.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("SearchCrashCabin", "Session ID", New Object()
{DateTime.SpecifyKind(DEFAULT_DATE, DateTimeKind.Unspecified),0})
```

## SearchGangwaylog

This function gets the gangway activities from SPMS.

**Note:** Not supported when it is connected to an offline QCI Sync database.

### Input Parameters

psFunction= "SearchGangwaylog" — below is the list of psParam or poParam.

**Table 4-133 psParam**

Parameter	Type	Description	Required
pdFromDate	Date	From date & time.	Yes
pdToDate	Date	To date & time.	Yes
pnAccID	Integer	Unique Account ID.	Yes
psLocation	String	Gangway location.	Yes
psMovement	String	Movement.	Yes
psAccType	String	Account Type.	Yes
psCab	String	Cabin Number.	Yes
psName	String	Name.	Yes
peSortType	Integer	Sort Type.	Optional

Below are the descriptions of the parameters.

#### pdFromDate

From date and time.

#### pdToDate

To date and time.

#### pnAccID

Unique Account ID, this is optional, if you want to search all guest, pass in 0.

#### psLocation

This is the location where the guest pass through the gangway.

#### psMovement

Pass in blank for all movement. Below is the valid value:

IMMIGRATION UNCHECK

IMMIGRATION CHECK

Going Shore-side

Coming Onboard

#### psAccType

Account Type, pass in blank for all. Below is the valid value:

C

G

V

**psCab**

Cabin number

**psName**

Guest Name, if “,” exists, all text before “,” will be first name, all text after “,” will be the last name. If “,” does not exist we will search the last name only.

**peSortType**

0 – Sort by GLG\_ID ASC

1 – Sort by GLG\_ID DESC

2 – Sort by GLG\_DATE ASC

3 – Sort by GLG\_DATE DESC

Default = 0

**Return Values**

CResponseXml.bSuccess = False when SearchGangwaylog has failed

CResponseXml.bSuccess = True when SearchGangwaylog is successful,  
CResponse.oDataset will have 1 Data Tables.

**Table 4-134 ADO.NET Data Table field structure for data table**

Code	Type	Description
GLG_DATE	Date	Movement Date and Time.
GLG_STATION	String	PC Name where the movement is done.
GLG_LOCATION	String	Location where the movement takes place.
GLG_MOVEMENT	String	Movement direction.
UXP_A_FSTN	Integer	First Name.
UXP_A_NAME	String	Last Name.
UXP_A_TAG	String	Account Type P-Passenger,C-Crew,V-Visitor.
RES_CAB	String	Cabin Number.
GLG_SHORE_DENIED	Integer	Flag overridden shore leave denied passenger.
GLG_SWIPE	String	Yes-Card is swipe, No-Manual Enter.
TYP_COMMENT	String	Port Name (Only exists in Ferry Mode).
RES_ACC	Integer	Unique Account ID.

**Example:**

```
oResponse = goWS.FidelioSPMSWSXML("SearchGangwaylog", "", New Object()
{ DateTime.SpecifyKind(#1/1/2011 1:00:00 pm#, DateTimeKind.Unspecified), _
```

```
DateTime.SpecifyKind(#2/1/2011 6:00:00 pm#, DateTimeKind.Unspecified),0,
"", "", "", "", "")
```

## SearchHKForecast

This function returns a list of itinerary port days and list of guests' reservations per cabin.

**Note:** Not supported when it is connected to an offline QCI Sync database.

### Input Parameters

psFunction= "SearchHKForecast" — below is the list of poParam.

Please take note some parameters are optional. However, the more parameters passed in, the fewer records it will return.

For Date field pass in 30-Dec-1899 if not required.

For Array field, pass in an empty array or "Nothing" to skip this checking.

For JSON, the array of string will be a string separated by ",", each string must start and end with a double quotation mark ("").

For example,

"A,","B," = 2 Element

Element 1 = A,

Element 2 = B,

**Table 4-135** psParam

Parameter	Type	Description	Required
pdItineraryFrom	Date	Port Itinerary From Day.	Yes
pdItineraryTo	Date	Port Itinerary To Day.	Yes
psCabinHKStatus	Array of String	Cabin Housekeeping Status.	Yes
psHKSection	Array of String	Housekeeping Section.	Yes
psDeck	Array of String	Cabin Deck.	Yes
psCabin	Array of String	Cabins.	Yes
psCabinAvlStatus	Array of String	Cabin Availability Status.	Yes
pdCrashCabinDate	Date	Date to search for Crash Cabins.	Yes

### Return Values

CResponseXml.bSuccess = False when SearchHKForecast has failed.

CResponseXml.bSuccess = True when SearchHKForecast is successful,  
CResponse.oDataset will have 4 Data Table.

**Table 4-136 ADO.NET Data Table 1 (“ITIN”) - Master Table 1 lists all the itinerary port days**

Code	Type	Description
SCD_ID	Integer	Unique Port Day ID.
SCD_DATE	Date	Port Date.
PORTNAME	String	Port ID (SCD_PORT_ID).
PORTDESCR	String	Port Name.
SCD_DESCR	String	Port Description.
SCD_ARRIVE_TIME	Date	Port Arrival Time.
SCD_DEPART_TIME	Date	Port Departure Time.

**Table 4-137 ADO.NET Data Table 2 (“CABINS”) – Master Table 2 lists all cabins**

Code	Type	Description
CAB_ID	Integer	Unique Cabin ID.
CAB_STATE	String	Cabin State (OO=Out of Order, OS=Out of Service, BL=Blocked, AV=Available).
CAB_CLEAN_STATE	String	Cabin Housekeeping status.
CAB_BEG_BLOCK_D	Date	Date of cabin blocked from.
CAB_REPAIR_D	Date	Date of cabin blocked until.
CAB_REPAIR_T	String	Unavailability Reason.

**Table 4-138 ADO.NET Data Table 3 (“RESERV”) – linked to Data Table2 by cabin, lists all reservations grouped by booking number**

Code	Type	Description
RES_BOOKNR	String	Reservation Booking Number.
TOTALGUEST	Integer	Number of Guests.
RES_CAB	String	Cabin.
RES_STATUS	String	Reservation status.
EMBARKDATE	Date	Embarkation Date (Arrival date & time) for expected guests.
DEBARKDATE	Date	Debarkation Date (Arrival date & time) for already onboard guests.
EMBARKPORT	String	Embarkation Port for expected guests.
DEBARKPORT	String	Debarkation Port for already onboard guests.

**Table 4-139 ADO.NET Data Table 4 (“CRASH\_CABINS”) – Master Table 4 lists all crash cabins and their reservations**

Code	Type	Description
CAB	String	Cabin.

**Table 4-139 (Cont.) ADO.NET Data Table 4 (“CRASH\_CABINS”) – Master Table 4 lists all crash cabins and their reservations**

Code	Type	Description
EMBARK_RES_BOOKNR	String	Reservation Booking Number for the embarking guest.
EMBARK_STATUS	String	Reservation Status of the embarking guest.
EMBARK_PORT	String	Embarkation Port of the embarking guest.
DEBARK_RES_BOOKNR	String	Reservation Booking Number for the debarking guest.
DEBARK_STATUS	String	Reservation Status of the debarking guest.
DEBARK_PORT	String	Debarkation Port of the debarking guest.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("SearchHKForecast", "Session ID", New
Object() {DateTime.SpecifyKind(DEFAULT_DATE, DateTimeKind.Unspecified),
DateTime.SpecifyKind(DEFAULT_DATE, DateTimeKind.Unspecified),Nothing,
Nothing, Nothing, Nothing, Nothing, DateTime.SpecifyKind(DEFAULT_DATE,
DateTimeKind.Unspecified)})
```

## SearchHouseKeeping

This function performs an advanced search for housekeeping cabins and its tasks.

**Note:** Not supported when it is connected to an offline QCI Sync database.

### Input Parameters

psFunction= “SearchHousekeeping” — below is the list of poParam.

Please take note some parameters are optional. However, the more parameters passed in, the fewer records it will return.

For Date field pass in 30-Dec-1899 if not required.

For Array field, pass in an empty array or “Nothing” to skip this checking.

For JSON, the array of string will be a string separated by “,”, each string must start and end with a double quotation mark (“”).

For example,

“A,”,”B,” = 2 Element

Element 1 = A,

Element 2 = B,

**Table 4-140 psParam**

Parameter	Type	Description	Required
pbIsCabin	Boolean	True – Cabin, False = Location.	Yes

Table 4-140 (Cont.) psParam

Parameter	Type	Description	Required
pdTaskDateFrom	Date	Date From - task that has been created for the cabin/location.	Yes
pdTaskDateTo	Date	Date To - task that has been created for the cabin/location.	Yes
psPort	Array of String	Port ID (SCD.SCD_ID) - for Housekeeping Tasks only.	Yes
psCabin	Array of String	Unique Cabin IDs (CAB.CAB_ID).	Yes
psTaskStatus	Array of String	0 – Pending, 1 – Started, 2 - Completed.	Yes
psCabinHKStatus	Array of String	Cabin Housekeeping Status (OD, OC, VD, VC).	Yes
psCabinAvlStatus	Array of String	Cabin Availability (OO, OV, BL, AV).	Yes
psHKSection	Array of String	Housekeeping section codes.	Yes
psCrewID	Array of String	Crew Account IDs (RES.RES_ACC)	Yes
psTaskID	Array of String	Task IDs (CTK.CTK_ID).	Yes
psDeck	Array of String	Cabin Deck (TYP_DEK.TYP_ART).	Yes
psCabinCategory	Array of String	Cabin Category (TYP_CAT.TYP_ART).	Yes
psCabinType	Array of String	'P' for Pax, 'C' for Crew.	Yes
psCabinVerticalZone	Array of String	Cabin Vertical Zone (TYP_VTZ.TYP_ART).	Yes
psLinenChange	String	1 – Task requires linen change. 0 – Task does not require linen change. If nothing is passed in, then it will skip this criteria.	
pnAMESRTstatus	Integer	0 – Not Delivered. 1 – Delivered. 2 – Both Delivered and Not Delivered. 3 – Do not return any Amenities or Special Requests.	Yes
pnAMESRTportID	Integer	Embarkation Port ID (SCD.SCD_ID) - for Amenities and Special Requests only. This is only for expected guests embarking on this port only.	Yes
pdAMESRTembarkdate	Date	Embarkation Date – for Amenities and Special Requests Only. This is only for expected guests embarking on this date only.	Yes
psAMESRTdepts	Array of String	Amenity/Special Requests Delivery Department.	Yes
pdAMESRTdeliverydate	Date	Amenity/Special Requests Delivery Date.	Yes
pdChangeModDate	Date	Last Changed Date from the last refresh from iCrew.	Yes



**Table 4-140 (Cont.) psParam**

Parameter	Type	Description	Required
pbSortByCabinsOnly	Boolean	TRUE – CABINS and TASKS output table to be sorted by cabin number. FALSE – CABINS and TASKS output table to be sorted by port arrival time and cabin number.	Yes
pbShowAllInCompleteTasks	Boolean	TRUE - show all incomplete tasks regardless of created date. FALSE - show incomplete tasks within date range based on par pdTaskDateFrom and pdTaskDateTo.	Yes
pbShowAllCompletedTasksToday	Boolean	TRUE - show all completed tasks today. FALSE - show all completed tasks within date range based on par pdTaskDateFrom and pdTaskDateTo.	Yes
pbUseOldSorting	Boolean	TRUE - use the current method of sorting in .846 (ignores par pbShowAllInCompleteTasks and pbShowAllCompletedTasksToday). This is for backward compatibility. FALSE - use the new method of sorting (uses par pbShowAllInCompleteTasks and pbShowAllCompletedTasksToday, ignores psTaskStatus).	Yes

**Return Values**

CResponseXml.bSuccess = False when SearchHousekeeping has failed.

CResponseXml.bSuccess = True when SearchHousekeeping is successful, CResponse.oDataset will have 11 Data Tables for cabins/locations.

**Table 4-141 ADO.NET Data Table field structure for data table (“SERVERDATETIME”), current server date/time.**

Code	Type	Description
CURRENTDATETIME	Date	Server Date Time.

**Table 4-142 ADO.NET Data Table field structure for data table (“STATS”), Overall Housekeeping Task Statistics and pending amenities/special requests for Cabins based on parameters passed in.**

Code	Type	Description
HK_PENDING_COUNT	Integer	Number of pending housekeeping tasks.
HK_STARTED_COUNT	Integer	Number of housekeeping tasks that have started.

**Table 4-142 (Cont.) ADO.NET Data Table field structure for data table (“STATS”), Overall Housekeeping Task Statistics and pending amenities/special requests for Cabins based on parameters passed in.**

Code	Type	Description
HK_COMPLETED_COUNT	Integer	Number of completed housekeeping tasks.
AME_PENDING_COUNT	Integer	Number of amenities count based on pnAMESRTstatus passed in.
SRT_PENDING_COUNT	Integer	Number of special requests count based on pnAMESRTstatus passed in.

**Table 4-143 ADO.NET Data Table field structure for data table (“CABINS”) - Cabin Details. It returns a list of cabins with or without housekeeping tasks AND pending amenities/special requests, sorted by SCD\_ARRIVE\_TIME, CAB**

Code	Type	Description
CAB	String	Unique Cabin ID/Location ID.
SCD_ARRIVE_TIME	Date	Port Arrival Time (linked to HAS_PORT_ID).
CAB_NAME	String	Cabin/Location Name.
CAB_DESIGN	String	'P' for Pax, 'C' for Crew. Does not apply to location.
CAB_STATE	String	Cabin Availability Status (OO=Out of Order, OS=Out of Service, BL=Blocked, AV=Available). Does not apply to location.
CAB_BEDS	Number	No of beds/berths in the cabin.
CAB_BEDS_SETUP	String	Beds setup for each cabin. Format saved A,B,C,D,E or 1,2,3,4,5.
CAB_TEL	String	Cabin Telephone Number.
CAB_TEL2	String	Cabin Telephone Number 2.
CAB_BEG_BLOCK_D	Date	The cabin is pending repairs, this field stores the date for when this cabin is blocked from.This is the 'Block From' field in FC Admin Cabin setup.
CAB_REPAIR_D	Date	The cabin is pending repairs, this field stores the date for when this cabin is blocked until.This is the 'Block Until' field in FC Admin Cabin setup.
CAB_REPAIR_T	String	The cabin is pending repairs, this field stores the unavailability reason when CAB_STATE=OO or OS or BL. This is the 'Unavailability Reason' field in FC Admin Cabin setup.
CAB_LIFEBOAT	String	Lifeboat this cabin is assigned to. TYP_BOT.TYP_ART.
CAB_LIFEBOAT_NAME	String	Lifeboat this cabin is assigned to. TYP_BOT.TYP_COMMENT.
CAB_STATION	String	Muster Station this cabin is assigned to. Linked to MST.MST_ID.
CAB_STATION_NAME	String	Muster Station this cabin is assigned to. Linked to MST.MST_NAME.

**Table 4-143 (Cont.) ADO.NET Data Table field structure for data table (“CABINS”) - Cabin Details. It returns a list of cabins with or without housekeeping tasks AND pending amenities/special requests, sorted by SCD\_ARRIVE\_TIME, CAB**

Code	Type	Description
CAB_LOCATION	Integer	Null-Undefined, 1-Forward, 2-Mid Ship, 3-Aft
CAB_STARBOARD	Integer	0-Portside, 1-Starboard.
CAB_CAT	String	Cabin Type. Linked to TYP_CAT.TYP_ART. Does not apply to location.
CAB_CAT_NAME	String	Cabin Type Description. Linked to TYP_CAT.TYP_COMMENT. Does not apply to location.
CAB_DECK	String	Cabin Deck. Linked to TYP_DEK.TYP_ART. Does not apply to location.
CAB_DECK_NAME	String	Cabin Deck Description. Linked to TYP_DEK.TYP_COMMENT. Does not apply to location.
CAB_HKSTATUS	String	Cabin Housekeeping status (OD, VD, OC, VC). Linked to TYP_CCS.TYP_ART. Does not apply to location.
CAB_HKSTATUS_NAME	String	Cabin Housekeeping status description. Linked to TYP_CCS.TYP_COMMENT. Does not apply to location.
CAB_VZONE	String	Cabin vertical zone. Linked to TYP_VTZ.TYP_ART. Does not apply to location.
CAB_VZONE_NAME	String	Cabin vertical zone description. Linked to TYP_VTZ.TYP_COMMENT. Does not apply to location.
HKSECTION_ID	Integer	Housekeeping Section ID (HSC_ID).
HKSECTION_NAME	String	Housekeeping Section Name (HSC_NAME).

**Table 4-144 ADO.NET Data Table field structure for data table (“CRASH\_CABINS”) - Crash Cabins**

Code	Type	Description
CAB	String	Cabin.
PORTNAME	String	Port Name.
PORTID	Integer	Port ID (SCD_ID).
PORTDATE	Date	Port Date (SCD_DATE).

**Table 4-145 ADO.NET Data Table field structure for data table (“TASKS”) sort by SCD\_ARRIVAL\_TIME field - Cabin Task Assignments.**

Code	Type	Description
CTK_ID	Integer	Task ID (CTK.CTK_ID).
CTK_NAME	String	Task Name (CTK.CTK_NAME).

**Table 4-145 (Cont.) ADO.NET Data Table field structure for data table (“TASKS”) sort by SCD\_ARRIVAL\_TIME field - Cabin Task Assignments.**

Code	Type	Description
CTK_DESCRIPTIONS	String	Task Description .
CTK_DURATION	Integer	Duration of Task.
CTK_LINENRULE	Integer	1 – requires linen change, 0 – does not require linen change.
HAS_ID	Integer	Unique Housekeeping Task ID.
CAB	String	Unique Cabin/Location ID.
HAS_STATUS	Integer	0 – Pending, 1 – Started, 2 - Completed.
HAS_ASSIGNDATE	Date	Task Created date.
HAS_COMMENT	String	Task Comments.
HAS_PORT_ID	Integer	Port ID. Linked to SCD.SCD_ID.
SCD_PORT_ID	Integer	Port Name.
HAS_MODDATE	Date	Last modified date for the task.
HAS_COMPLETEDDATE	Date	Task Completed Date.
CRW_ID	Integer	Unique Crew ID.
CRW_NAME	String	Crew that has been assigned the task to.
SCD_ARRIVE_TIME	Date	Port Arrival Time.

**Table 4-146 ADO.NET Data Table field structure for table “STATS\_CABINS”, Housekeeping Task Statistics per Cabin based on parameters passed in.**

Code	Type	Description
CAB	String	Unique Cabin ID/Location.
HK_PENDING_COUNT	Integer	Number of pending tasks per cabin/location.
HK_STARTED_COUNT	Integer	Number of tasks that have started per cabin/ location.
HK_COMPLETED_COUNT	Integer	Number of completed tasks per cabin/location.

**Table 4-147 ADO.NET Data Table field structure for table “STATS\_AME”, Pending Amenities per Cabin.**

Code	Type	Description
CAB	String	Unique Cabin ID/Location.
AME_PENDING_COUNT	Integer	Number of pending amenities per cabin/location.

**Table 4-148 ADO.NET Data Table field structure for table “STATS\_SRT”, Pending Special Requests per Cabin.**

Code	Type	Description
CAB	String	Unique Cabin ID/Location.

**Table 4-148 (Cont.) ADO.NET Data Table field structure for table “STATS\_SRT”, Pending Special Requests per Cabin.**

Code	Type	Description
SRT_PENDING_COUNT	Integer	Number of pending special requests per cabin/location.

**Table 4-149 ADO.NET Data Table field structure for table “EMB\_PORTS”, Reservation Embark/Disembark Ports per Cabin**

Code	Type	Description
CAB	String	Unique Cabin ID/Location.
EMBARKPORT	String	Embarkation port for the checked-in guest for this cabin.
DEBARKPORT	String	Disembarkation port for the checked-in guest for this cabin.

**Table 4-150 ADO.NET Data Table field structure for table “RUSH\_CABINS”, Rush Cabins**

Code	Type	Description
CAB	String	Unique Cabin ID/Location.
EMBARKPORT	String	Embarkation port for the reservation that is embarking on pdTaskDate.
DEBARKPORT	String	Disembarkation port for the reservation that is disembarking on pdTaskDate.
EMBARKSTATUS	String	Embarkation status for the reservation that is embarking on pdTaskDate.
EMBARK_ARRIVEDATE	Date	Date and Time of the reservation that is embarking on pdTaskDate.

**Table 4-151 ADO.NET Data Table field structure for table “COUT\_CABINS”, Cabins with check-out tasks**

Code	Type	Description
CAB	String	Unique Cabin ID/Location.
HAS_PORT_ID	Integer	Port ID where the cabin has check-out tasks.
PORTNAME	String	Port name where the cabin has check-out tasks.
SCD_DATE	Date	Port Arrival Date.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("SearchHouseKeeping", "Session ID", New
Object() {True, DateTime.SpecifyKind(DEFAULT_DATE,
DateTimeKind.Unspecified), Nothing, Nothing, Nothing, Nothing, Nothing,
Nothing, Nothing, Nothing, Nothing, Nothing, Nothing, Nothing, 1, 0,
```

```
12345, DateTime.SpecifyKind(DEFAULT_DATE, DateTimeKind.Unspecified), Nothing,
DateTime.SpecifyKind(DEFAULT_DATE, DateTimeKind.Unspecified),
DateTime.SpecifyKind(DEFAULT_DATE, DateTimeKind.Unspecified))
```

## SearchPassenger

This function searches for a guest using various criteria available in the SPMS system.

### Input Parameters

psFunction= "SearchPassenger" — below is the list of psParam or poParam.

**Table 4-152 psParam**

Parameter	Type	Description	Required
psFirstName	String	First Name.	Optional
psLastName	String	Embarkation Date.	Optional
psPassportNum	String	Guest Type Enumerator.	Optional
pdBirthDate	Date	Guest Birth date.	Yes

Below are the descriptions of the parameters.

#### psFirstName

Guest First Name, this field is optional, can pass in an empty string.

#### psLastName

Guest Last Name, this field is optional, can pass in an empty string.

#### psPassportNum

Guest Passport Number, this field is optional, can pass in an empty string.

**Note:** For psFirstName, psLastName and psPassportNum, *must* supply at least one information.

#### pdBirthdate

Guest Birth date

### Return Values

CResponseXml.bSuccess = False when input values are invalid or no record found.

CResponseXml.bSuccess = True and CResponseXML.CPassenger when input values are valid.

**Table 4-153 CPassenger field structure**

Code	Type	Description
gnID	Integer	Unique Guest ID, this ID is used to perform in other search.

**Table 4-153 (Cont.) CPassenger field structure**

Code	Type	Description
gsFirstName	String	First Name.
gsLastName	String	Last Name.
gsMiddleName	String	Middle Name.
gsTitle	String	Title.
gsSalutation	String	Salutation.
gdEmbDate	Date	Embarkation Date.
gdDisDate	Date	Disembarkation Date.
gsCabin	String	Cabin Number.
gsCabCategory	String	Cabin Category.
gsFreqCruiseNumber	String	Frequent Cruise Number.
gsFreqCruiseStatus	String	Frequent Cruise Status.
gsDeck	String	Cabin Deck.
gsMusterStation	String	Muster Station Name.
gsAward	String	Award Level.
gsLoyaltyCode	String	Loyalty Discount Code.
gsLoyaltyDesc	String	Loyalty Discount Description.
gsAccTag	String	Account TAG,P – Pax, C - Crew.
gsLanguage1	String	Language 1.
gsLanguage2	String	Language 2.
gsLanguage3	String	Language 3.
gnGrdID	Integer	Group ID
gsGrdName	String	Group Name
gsBookingNo	String	Reservation Booking No.
gsDiscount	String	Current not in use.

**Example:**

```
oResponse = goWS.FidelioSPMSWSXML("SearchPassenger", "SESSIONID", New
Object() {<FIRST NAME>, <Last Name>, <Passport Number>, #1/1/1970#})
```

## SearchPassengerByID

This function searches for a guest with a unique Guest ID from the SPMS system.

**Input Parameters**

psFunction= "SearchPassengerByID" — below is the list of psParam or poParam.

**Table 4-154 psParam**

Parameter	Type	Description	Required
pnAccID	Integer	SPMS Unique Guest ID.	Yes

Below are the descriptions of the parameters.

**pnAccID**

Unique Guest ID.

**Return Values**

CResponseXml.bSuccess = False when input values are invalid or no record found.

CResponseXml.bSuccess = True and CResponseXML.CPassenger when input values are valid.

**Table 4-155 CPassenger field structure**

Parameter	Type	Description
gnID	Integer	Unique Guest ID, this ID is used to perform in other search.
gsFirstName	String	First Name.
gsLastName	String	Last Name.
gsMiddleName	String	Middle Name.
gsTitle	String	Title.
gsSalutation	String	Salutation.
gdEmbDate	Date	Embarkation Date.
gdDisDate	Date	Disembarkation Date.
gsCabin	String	Cabin Number.
gsCabCategory	String	Cabin Category.
gsFreqCruiseNumber	String	Frequent Cruise Number.
gsFreqCruiseStatus	String	Frequent Cruise Status.
gsDeck	String	Cabin Deck.
gsMusterStation	String	Muster Station Name.
gsAward	String	Award Level.
gsLoyaltyCode	String	Loyalty Discount Code.
gsLoyaltyDesc	String	Loyalty Discount Description.
gsAccTag	String	Account TAG,P – Pax, C - Crew.
gsLanguage1	String	Language 1.
gsLanguage2	String	Language 2.
gsLanguage3	String	Language 3.
gnGrdID	Integer	Group ID.
gsGrdName	String	Group Name.
gsBookingNo	String	Reservation Booking No.



**Table 4-155 (Cont.) CPassenger field structure**

Parameter	Type	Description
gsDiscount	String	Current not in use.

## SearchPayrolls

This function returns payroll data based on various search criteria from the SPMS system.

### Input Parameters

psFunction="SearchPayrolls" — below is the ordered list of input parameters and its descriptions.

**Table 4-156 psParam**

Parameter	Type	Description	Required
MemoryLifeSpan	Integer	Optional in seconds, default as zero. If set to non-zero, result set will be kept in memory for reuse when a subsequent call is issued within the specified time interval with MemoryReuse = TRUE.	Optional
MemoryReuse	Boolean	Optional, default as FALSE. When set to TRUE, the search will not query DB but instead just return records from the previous result set.	Optional
RecordPerPage	Integer	Optional, default as zero. Specify the number of CTR records for which their payroll data will be returned. If zero, will return all CTR records in the result set.	Optional
PageNum	Integer	Optional, default as zero for page 1. Specify the page to return in a multi-page result set.	Optional
CTR_IDs	String	Optional. Comma delimited CTR_ID.	Optional
SPP_STATUSES	String	Optional. Comma delimited SPP_STATUS. *	Optional
SSH_STATUSES	String	Optional. Comma delimited SSH_STATUS. *	Optional

### Return Values

CResponseXml.bSuccess = False when WS function did not execute successfully. For example, due to invalid input parameters.

CResponseXml.bSuccess = True when WS function executed successfully, along with

CResponseXML.oDataSet = Nothing, when no record is found. Otherwise,

CResponseXML.oDataSet = *ADO.NET dataset*, which will contain table SSH, PAY, SPP, SED, BEN and USR. \*

CResponseXML.nTotalPage = total number of pages in the resultset, when input parameter *RecordPerPage* is specified.

Remark:

See SPMS Data Dictionary for details of the fields.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("SearchPayrolls", "SessionID", New Object()
{60, True, 10, 1, 46598,46599,46600, "O,A,C,F", "A,P"})
```

## SearchSpecialRequest

This function is an advanced search for special request.

### Input Parameters

psFunction= "SearchSpecialRequest"— below is the list of poParam.

Please take note some parameters are optional. However, the more parameters passed in, the fewer records it will return.

For Date field pass in 30-Dec-1899 if not required.

For Array field, pass in an empty array or "Nothing" to skip this checking.

For JSON, the array of string will be a string separated by ",", each string must start and end with a double quotation mark ("").

For example,

"A","B" = 2 Element

Element 1 = A,

Element 2 = B,

**Table 4-157 psParam**

Parameter	Type	Description	Required
peReservationType	Integer	Reserve and CheckIn. 1 – Reserve 2 - CheckIn 3 - CheckOut	
pdExpDeliveryDateFrom	Date	Special Request Delivery Date (SRT_DDATE) If nothing is passed in, then it will skip this criteria.	
pdExpDeliveryDateTo	Date	Special Request Delivery Date (SRT_DDATE)If nothing is passed in, then it will skip this criteria.	
pdDeliveredDateFrom	Date	Special Request Delivery Date (SRT_DONE_DATE)If nothing is passed in, then it will skip this criteria.	

**Table 4-157 (Cont.) psParam**

Parameter	Type	Description	Required
pdDeliveredDateTo	Date	Special Request Delivery Date (SRT_DONE_DATE)If nothing is passed in, then it will skip this criteria.	
psGuestLastName	String	Guest Last Name (UXP_A_NAME)If nothing is passed in, then it will skip this criteria.	
psCode	Array of String	Special Request Code (TYP_SRT.TYP_ART)If nothing is passed in, then it will skip this criteria.	
psExtCode	Array of String	Special Request Identifier Code.	
psGroupID	Array of String	Travel Group ID (UXP_A_ID)If nothing is passed in, then it will skip this criteria.	
psCabinCategory	Array of String	Cabin Category (TYP_CAT.TYP_ART)If nothing is passed in, then it will skip this criteria.	
psDeliveryLocation	Array of String	Delivery Location (SRT_LOCATION)If nothing is passed in, then it will skip this criteria.	
psDeck	Array of String	Cabin Deck (TYP_DEK.TYP_ART)If nothing is passed in, then it will skip this criteria.	
psDeliveryBy	Array of String	Special Request Delivery By. Linked to USR.USR_ID.If nothing is passed in, then it will skip this criteria.	
psCabin	Array of String	Unique Cabin IDs (CAB.CAB_ID)If nothing is passed in, then it will skip this criteria.	
psDepartment	Array of String	Amenity Delivery Division (SRT_ODI_ID)If nothing is passed in, then it will skip this criteria.	
psHKSection	Array of String	Housekeeping Section Name (HSC_NAME)If nothing is passed in, then it will skip this criteria.	

**Return Values**

CRresponseXml.bSuccess = False when SearchSpecialRequest has failed.

CRresponseXml.bSuccess = True when SearchSpecialRequest is successful,  
CRresponse.oDataset will have 1 Data Table.

**Table 4-158 ADO.NET Data Table field structure for data table**

Code	Type	Description
SRT_ID	Integer	Special Request ID (Unique ID).
SRTCODE	String	Special Request Code. Linked to TYP_SRT.TYP_ART.

**Table 4-158 (Cont.) ADO.NET Data Table field structure for data table**

<b>Code</b>	<b>Type</b>	<b>Description</b>
SRTNAME	String	Port Name. Linked to TYP_SRT.TYP_COMMENT.
SRT_EXT_CODE	String	Special Request Identifier Code.
ACCID	String	Account ID.
Name	String	Guest Name.
RESSTATUS	String	Reservation Status.
VIPCODE	String	VIP ID. Linked to TYP_VIP.TYP_ART.
VIPNAME	String	VIP Name. Linked to TYP_VIP.TYP_COMMENT.
PORTID	String	Embarkation Port ID.
PORTNAME	Integer	Embarkation Port Name. Linked to TYP_HRB.TYP_COMMENT.
GROUPID	String	Travel Group Reference ID.
GROUPNAME	String	Travel Group Name.
CABINID	String	Unique Cabin ID.
HSC_NAME	String	Housekeeping Section Code.
HSC_COMMENT	String	Housekeeping Section Name.
SRT_LOCATION	String	Delivery Location.
SRT_ODI_ID	String	Amenity Delivery Division ID. Linked to TYP_ODI.TYP_ART.
TYP_COMMENT	String	Amenity Delivery Division Name. Linked to TYP_ODI.TYP_COMMENT.
CAB_VZONE	String	Cabin vertical zone. Linked to TYP_VTZ.TYP_ART.
CAB_VZONE_NAME	String	Cabin vertical zone description. Linked to TYP_VTZ.TYP_COMMENT.
CAB_TYPE	String	Cabin Type ID. Linked to TYP_CAT.TYP_ART.
TYP_COMMENT1	string	Cabin Type Name. Linked to TYP_CAT.TYP_COMMENT.
CAB_DECK	String	Cabin Deck ID. Linked to TYP_DEK.TYP_ART.
TYP_COMMENT2	String	Cabin Deck Name. Linked to TYP_DEK.TYP_COMMENT.
SRT_DDATE	Date	Special Request Delivery Date.
SRT_DONE_DATE	Date	Special Request Service Done Date.
SRT_DONE_BY	Integer	Special Request done by the waiter. Linked to USR.USR_ID.
USR_LOGIN	String	User Login Name.
CAB_KEEPER	Integer	Housekeeper ID. Linked to RES.RES_ACC.
CAB_KEEPER_NAME	String	Housekeeper Name.
CAB_A_KEEPER	Integer	Assistant Housekeeper ID. Linked to RES.RES_ACC.

**Table 4-158 (Cont.) ADO.NET Data Table field structure for data table**

Code	Type	Description
CAB_A_KEEPER_NAME	String	Assistant Housekeeper Name.
CAB_G_WAITER	Integer	Waiter ID. Linked to RES.RES_ACC.
CAB_G_WAITER_NAME	String	Waiter Name
CAB_A_WAITER	Integer	Assistant Waiter. Linked to RES.RES_ACC.
CAB_A_WAITER_NAME	String	Assistant Waiter Name.
SRT_CREATED_BY	Number	The user that created this special request.
SRT_DATE_CREATED	Date	Special request actual creation date.
SRT_DESCR	String	Special Request Description.
SRT_EVENTDATE	Date	Event Date & Time.
SRT_EXT_ID	Number	Interface System Special Request ID.
SRT_MODDATE	Date	Last Modification Date.
SRT_QUANTITY	Number	Special Request Quantity.
SRT_RESID	Integer	Guest Account ID. Linked to RES.RES_ID.
SRT_VID	String	Unique ID (for import duplicate functionality).

**Example:**

```
oResponse = goWS.FidelioSPMSWSXML("SearchSpecialRequest", New Object()
{<ReservationType>, <ExpDeliveryDateFrom>, <ExpDeliveryDateTo>, <DeliveryDate
From>, <DeliveryDateTo>, <LastName>, <SRTCode>, <ExternalCode>, <GroupID>, <Cabi
nCategory>, <DeliveryLocation>, <Deck>, <DeliveryBy>, <Department>, <HKSection>
})
```

## SearchWorkOrder

This function gets the maintenance work order from SPMS.

**Note:** Not supported when it is connected to an offline QCI Sync database.

### Input Parameters

psFunction= "SearchWorkOrder" — below is the list of psParam or poParam.

Please take note all the parameter are optional except pdFromDate and pdToDate, the more parameters you pass in, the fewer records it will return.

For Array field, pass in an empty array or "Nothing" to skip this checking.

For Date field pass in 30-Dec-1899 if not required.

For JSON, the array of string will be a string separated by ",", each string must start and end with a double quotation mark (").

For example,

"A","B" = 2 Element

Element 1 = A,"

Element 2 = B,

**Table 4-159 psParam**

Parameter	Type	Description	Required
pdFromDate	Date	From date & time.	Yes
pdToDate	Date	To date & time.	Yes
pdCompleteFromDate	Date	Completion date & time from.	Yes
pdCompleteToDate	Date	Completion date & time to.	Yes
psWorkOrderTyp	Array of String	Work Order Type.	Yes
psCabin	Array of String	Cabin Location.	Yes
psPublicLocation	Array of String	Public Location.	Yes
psLocation	Array of String	Custom Location.	Yes
psTask	Array of String	Work Order Task.	Yes
psWorkGroup	Array of String	Work Order Assign Worker Group.	Yes
pnUsrCreate	Array of String	User ID which created the Work Order. Although the data type is numeric, you must pass in as Array of String due to .NET limitation.	Yes
pnUsrAssign	Array of String	Work Order Assign to the Crew. Although the data type is numeric, you must pass in as Array of String due to .NET limitation.	Yes
psPriority	Array of String	Priority.	Yes
pnMwrID	Array of String	Work Order Unique ID. Although the data type is numeric, you must pass in as Array of String due to .NET limitation.	Yes
psMwrNo	Array of String	Work Order Number.	Yes
pnCompleteStatus	Array of String	Completion Status, Range 0,25,50,75,100. Although the data type is numeric, you must pass in as Array of String due to .NET limitation.	Yes
pbIncludePicture	Boolean	True = Will return the work order picture. False = Will return nothing, this is for slow device, so it will need to look up the picture using WSF using the value WSF_XDOC_ID, default = True.	Optional
pbSortModDateAsc	Boolean	True it will sort the records by last modify date ASC. False it will sort the records by last modify date DESC. Default = True.	Optional

### Return Values

CResponseXml.bSuccess = False when SearchWorkOrder has failed.

CResponseXml.bSuccess = True when SearchWorkOrder is successful,  
CResponse.oDataset will have 4 Data Tables.

**Table 4-160 ADO.NET Data Table field structure for first data table, Work Order**

Code	Type	Description
MWR_CSTATUS	Integer	Completion Status Percentage 0,25,50,75 or 100.
MWR_DATE	Date	Date Time the order created.
MWR_LOCATION	String	Custom Location of the maintenance.
MWR_LOCATIONCABIN	String	Cabin Number.
MWR_LOCATIONPUBLIC	String	Public Location.
PRIORITY	String	Priority Description.
MWR_TASK	String	Task Code.
TSK_COMMENT	String	Task Name.
MWR_TASKDESCR	String	Task Description.
USR_LOGIN	String	User Login Name.
MWR_TYPE	String	Work Order Type.
WOT_NAME	String	Word Order Type Description.
MWR_ID	Integer	Unique Work Order ID.
MWR_WONUMBER	String	Work Order Number.
ARE_CATEGORY	String	Custom Location Category.
CAT_NAME	String	Custom Location Category Name.
STATUS	String	Useable Status.
XDOC_FILE	Byte Array	Picture of the work order.
MWR_TIME_NEEDED	Double	Time Needed in Days.
MWR_PICTUREFILE	String	File Name of the picture.
MWR_COMMENT	String	Comment.
MWR_CAB_ID	String	Report By Cabin Number.
ARE_NAME	String	Custom Location Name.
PUBLIC_NAME	String	Public Location Name.
MWR_COMPLETEDATE	Date	Completion Date.
MWR_PRINTED	Integer	Indicated how many time it printed.
MWR_REJECTED	Boolean	True – Rejected.
MWR_XDOC_ID	Long	The work order picture ID = xdoc_id.

**Table 4-161 ADO.NET Data Table field structure for second data table, Work Order Assignment**

Code	Type	Description
MWA_ID	Integer	Unique Work Order Assignment ID.
MWA_MWRID	Integer	Work Order ID.

**Table 4-161 (Cont.) ADO.NET Data Table field structure for second data table, Work Order Assignment**

Code	Type	Description
MWA_ASSIGNEDTOGROUP	String	Assign Group Code.
MWA_ASSIGNEDTOWORKER	Integer	Unique Crew ID.
MWA_ASSIGNEDWHEN	Date	Date and Time of assignment.
MWA_REMARKS	String	Remarks.
CRW_NAME	String	Crew Name.
MWR_GROUP	String	Group Description.
USR_LOGIN	String	User Login Name.

**Table 4-162 ADO.NET Data Table field structure for the third data table, Work Order Assignment**

Code	Type	Description
MWI_ID	Integer	Unique Work Order Item ID.
MWI_MWRID	Integer	Work Order ID.
MWI_ITEMCODE	String	Item Code.
MWI_QUANTITY	Integer	Quantity Needed.
MWI_PRICE	Double	Price of the item.
MWI_REMARKS	String	Remarks.
MIT_DESC	String	Item Description.
MIT_GROUP	String	Item Group.
MIT_STOCK	Integer	Quantity On Hand.
TYP_COMMENT	String	Item Group Description.

**Table 4-163 ADO.NET Data Table field structure for fourth data table, Work Order Additional Image**

Code	Type	Description
MWP_ID	Integer	Unique Work Order Image ID.
MWP_MWRID	Integer	Work Order ID.
MWP_COMMENT	String	User Comment.
MWP_FILENAME	String	Original Picture Filename, Information purpose.
MWP_IMAGE	Byte Array	Additional Picture of the work order.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("SearchWorkOrder", "", New Object()
{DateTime.SpecifyKind(#1/1/2000#, DateTimeKind.Unspecified), _
DateTime.SpecifyKind(#1/1/2012#, DateTimeKind.Unspecified), _
```



```

DateTime.SpecifyKind(DEFAULT_DATE, DateTimeKind.Unspecified), _
DateTime.SpecifyKind(DEFAULT_DATE, DateTimeKind.Unspecified), _ Nothing,
New String() {"05002", "00000"}, Nothing, Nothing, _ Nothing, New String()
{"1"}, Nothing, Nothing, Nothing, _ Nothing, Nothing, Nothing, False,
False))

```

## SpaBooking

This function creates a single or Packaged Treatment Spa booking in SPMS.

**Note:** Not supported when it is connected to an offline QCI Sync database.

### Input Parameters

psFunction= "SpaBooking" — below is the list of psParam or poParam.

**Table 4-164 psParam**

Parameter	Type	Description	Required
pnAccID	Integer	Unique Guest ID.	Yes
pnLocation	Integer	Location ID.	Yes
pnItem	Integer	Treatment Item ID.	Yes
pdDate	Date	Treatment Date and Time.	Yes
psNotes	String(255)	Notes (Default Empty).	Optional
psPin	String(4)	Pin Code (Default Empty).	Optional
psSource	String(10)	Booking Source (Default Empty).	Optional
pbForceBooking	Boolean	True = Allow posting even the guest who does not have credit limit or posting is disabled, False = Normal posting.	Yes
pbIsUnassignedBooking	Boolean	True = Allow booking without Treatment Date and Time in order to be manually confirmed onboard, False = Regular Booking with Treatment Date and Time.	Yes
pnPackageID	Integer	Package ID.	Optional
pnPackageControlNo	Integer	Unique ID Grouping Package Treatments and Retail Items.	Optional

### pnAccID

Unique Guest ID, return from SearchPassenger.

### pnLocation

The field identifies the unique location ID. The field is an internal, unique code. This value is retrieved from the SPALocation message type. For Package Treatments, the value is 0, because originally, packages can contain treatments from multiple/different locations. In this case, the function will automatically find the appropriate location schedule for the requested date and time.

### pnItem

The field identifies the unique treatment item identification. The field is an internal, unique code. This field is retrieved from SPATreatmentItem.

**pdDate**

The field identifies the date and time in which the treatment takes place. This field is retrieved from SPATreatmentAvailability. If pbIsUnassignedBooking is True, then pdDate must be empty.

**psNotes**

The field identifies the additional information send by third-party.

**psPin**

The field identifies the telephone PIN code of the reservation account. The valid characters are 0,1,2,3,4,5,6,7,8,9. Leading zero support depends on SPMS Parameter setting.

**psSource**

The field identifies the source of the booking. For example, Workstation ID, Kiosk ID, and so on or any other details describing the location where the booking was made.

**pbForceBooking**

True = Allow posting even if the guest does not have a credit limit or posting is disabled. This is applicable to check-in guest only.

False = Normal posting.

**pbIsUnassignedBooking**

True = Allow booking without Treatment Date and Time in order to be manually confirmed onboard.

False = Regular Booking with Treatment Date and Time.

**pnPackageID**

The field identifies the unique Package ID. The field is an internal, unique code. This value is retrieved from GetSPAPackages WebMethod.

**pnPackageControlNo**

The field identifies the grouping of Treatments and Retail Items that belong to the same package and are sold together in one transaction. When booking the first Package Treatment, this field must always be "0" in order to return the new Package ControlNo value upon booking of the first package treatment with return value CResponse.oObj(1). For the rest of treatment bookings, this value must contain the same Package ControlNo as returned after the 1st Package Treatment Booking.

**Return Values**

CResponseXml.bSuccess = True when Booking is successful.

CResponse.oObj(0) return the booking ID (spa\_id).

CResponse.oObj(1) return the Package ControlNo (spa\_control).

CResponseXml.bSuccess = False when Booking has failed.

Example:

For Regular single Treatments

```
oResponse = goWS.FidelioSPMSWSXML("SpaBooking", "SESSIONID", New Object()
{12345, 37464, 1, 123, #1/1/2011 3:00 pm#, "", "", "", False, False})
```

For Unassigned single Treatment without Treatment Date & Time assignment

```
oResponse = goWS.FidelioSPMSWSXML("SpaBooking", "SESSIONID", New Object()
{12345, 37464, 1, 123, "", "", "", "", False, True})
```

For 1<sup>st</sup> Regular Package Treatment

```
oResponse = goWS.FidelioSPMSWSXML("SpaBooking", "SESSIONID", New Object()
{12345, 37464, 0, 123, #1/1/2011 3:00 pm#, "", "", "", False,False,4,0})
```

CResponse.oObj(0) return the booking id (spa\_id) = 1342

CResponse.oObj(1) return the Package ControlNo (spa\_control) = 1341

For rest of Regular Package Treatments

```
oResponse = goWS.FidelioSPMSWSXML("SpaBooking", "SESSIONID", New Object()
{12345, 37464, 0, 133, #1/1/2011 3:00 pm#, "", "", "",
False,False,4,1341})
```

For 1<sup>st</sup> Unassigned Package Treatment

```
oResponse = goWS.FidelioSPMSWSXML("SpaBooking", "SESSIONID", New Object()
{12345, 37464, 0, 123, "", "", "", "", False,True,4,0})
```

CResponse.oObj(0) return the booking id (spa\_id) = 1342

CResponse.oObj(1) return the Package ControlNo (spa\_control) = 1341

For rest of Unassigned Package Treatments

```
oResponse = goWS.FidelioSPMSWSXML("SpaBooking", "SESSIONID", New Object()
{12345, 37464, 0, 133, "", "", "", "", False,True,4,1341})
```

## SpaBookingPackageRetail

This function creates a spa package retail booking in SPMS after all Package Treatments Bookings have been completed.

**Note:** Not supported when it is connected to an offline QCI Sync database.

### Input Parameters

psFunction= "SpaBookingPackageRetail" — below is the list of psParam or poParam.

**Table 4-165 psParam**

Parameter	Type	Description	Required
pnAccID	Integer	Unique Guest ID.	Yes
pnRetailItem	Integer	Retail Item ID.	Yes
pnPackageID	Integer	Package ID.	Yes
pnPackageControlNo	Integer	Unique ID Grouping Package Treatments and Retail Items.	Yes
psNotes	String(255)	Notes (Default Empty).	Optional
psPin	String(4)	Pin Code (Default Empty).	Optional
psSource	String(10)	Booking Source (Default Empty).	Optional
pbForceBooking	Boolean	True = Allow posting even the guest who does not have credit limit or posting is disabled, False = Normal posting.	Yes

**pnAccID**

Unique Guest ID, return from SearchPassenger.

**pnRetailItem**

The field identifies the unique retail item identification. The field is an internal, unique code. This field is retrieve from GetSPAPackageItems or GetSPAPackageItems ForPeriod.

**pnPackageID**

The field identifies the unique Package ID. The field is an internal, unique code. This value is retrieved from GetSPAPackages WebMethod.

**pnPackageControlNo**

The field identifies the grouping of Treatments and Retail Items that belong to the same package and are sold together in one transaction. Since all included Package Retail Items can only be booked after all Package Treatments have been booked, the Package Control No is already known because it has already been returned after the 1st Package Treatment Booking.

**psNotes**

The field identifies the additional information sent by third-party.

**psPin**

The field identifies the telephone PIN code of the reservation account. The valid characters are 0,1,2,3,4,5,6,7,8,9. Leading zero support depends on SPMS Parameter setting.

**psSource**

The field identifies the source of the booking. For example, Workstation ID, Kiosk ID and so on or any other details describing the location where the booking was made.

**pbForceBooking**

True = Allow posting even the guest who does not have credit limit or posting is disabled (This only apply to check-in guest only), False = Normal posting.

**Return Values**

CResponseXml.bSuccess = True when Booking is successful.

CResponse.oObj(0) return the booking ID (spa\_id).

CResponseXml.bSuccess = False when Booking has failed.

Example:

For 1st Package Treatment

```
oResponse = goWS.FidelioSPMSWSXML("SPABookingPackageRetail", "SESSIONID",
New Object() {12345, 123, 4, 1341, "", "", "", False})
```

## SpecialRequestCompleted

This function updates the special request status as “Completed.”

**Note:** Not supported when it is connected to an offline QCI Sync database.

**Input Parameters**

psFunction= “SpecialRequestCompleted” — below is the list of psParam or poParam.

**Table 4-166 psParam**

Parameter	Type	Description	Required
pnSrtID	Integer	Special Request Unique ID.	Yes

**Return Values**

CResponseXml.bSuccess = False when Update failed.

CResponseXml.bSuccess = True when Update successful.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("SpecialRequestCompleted", "SessionID",
New Object() {12334})
```

## UndoAmenityCompleted

This function updates the amenity status as “Not complete”.

**Note:** Not supported when it is connected to an offline QCI Sync database.

**Input Parameters**

psFunction= “UndoAmenityCompleted” — below is the list of psParam or poParam.

**Table 4-167 psParam**

Parameter	Type	Description	Required
pnAmeID	Integer	Amenity Unique ID.	Yes

**Return Values**

CResponseXml.bSuccess = False when Update failed.

CResponseXml.bSuccess = True when Update successful.

**Example:**

```
oResponse = goWS.FidelioSPMSWSXML("UndoAmenityCompleted", "SessionID", New
Object() {12334})
```

## UndoSpecialRequestCompleted

This function updates the special status as “Not complete”.

**Note:** Not supported when it is connected to an offline QCI Sync database.

**Input Parameters**

psFunction= “UndoSpecialRequestCompleted” — below is the list of psParam or poParam.

**Table 4-168 psParam**

Parameter	Type	Description	Required
pnSrtID	Integer	Special Request Unique ID.	Yes

**Return Values**

CResponseXml.bSuccess = False when Update failed.

CResponseXml.bSuccess = True when Update successful.

**Example:**

```
oResponse = goWS.FidelioSPMSWSXML("UndoSpecialRequestCompleted", "SessionID",
New Object() {12334})
```

## UpdateAmenity

This function adds or updates the guest amenity.

**Note:** Not supported when it is connected to an offline QCI Sync database.

**Input Parameters**

psFunction= “UpdateAmenity” — below is the list of psParam or poParam.

**Table 4-169 psParam**

Parameter	Type	Description	Required
pnAmeID	Integer	Unique Amenity ID (0 for add new record).	Yes
pnAcclID	Integer	Unique Account ID.	Yes
psCode	String(6)	Amenity Code.	Yes
psNote	String(2500)	Amenity Description.	Yes
pnAmount	Double	Amenity Price each.	Yes
pnQuantity	Integer	Amenity Quantity.	Yes
psSender1	String(255)	Sender 1.	Yes
psSender2	String(30)	Sender 2.	Yes
psMessage	String(255)	Message for sender.	Yes
pdDeliveryDate	Date	Expected Delivery Date and Time.	Yes
psLocation	String(30)	Location.	Yes
psOccurrence	String(10)	Occurrence.	Yes
psExtCode	String(15)	External Code.	Yes
pnCost	Double	Cost.	Yes
psComments	String(2500)	Comments.	Yes
psExtID	String(20)	External ID.	Yes
psVID	String(100)	Unique External ID.	Yes
psOdiID	String(6)	Delivery Division.	Yes
psCurlID	String(6)	Currency Code.	Yes
psFlag	String(1)	Flag	Yes

**Return Values**

CResponseXml.bSuccess = False when Update/Add failed.

CResponseXml.bSuccess = True when Update/Add successful, CResponseXML.oObj = ame\_id (Unique Amenity ID).

**Example:**

```
oResponse = goWS.FidelioSPMSWSXML("UpdateAmenity", "SessionID", New
Object() {12345,1234576,"Code","Note",1.23,1, "Sender1", "Sender2",
"Message", DateTime.SpecifyKind(#01-01-2013 3:00:00 pm#,
DateTimeKind.Unspecified), "Location", "Once", "ExtCode", 0.45,
"Comments", "ExtID", "", "GRP", "US", ""})
```

## UpdateCabCleanStatus

This function updates the cabin clean status.

**Note:** Not supported when it is connected to an offline QCI Sync database.

### Input Parameters

psFunction= "UpdateCabCleanStatus" — below is the list of psParam or poParam.

**Table 4-170 psParam**

Parameter	Type	Description	Required
psCabID	String(20)	Cabin Number.	Yes
psCabCleanStatus	String(2)	Available value is OD,OC,VD,VC,PU,FA,VF.	Yes

### Return Values

CResponseXml.bSuccess = False when Update failed.

CResponseXml.bSuccess = True when Update successful.

#### Example:

```
oResponse = goWS.FidelioSPMSWSXML("UpdateCabCleanStatus", "SessionID", New
Object() {"05002", "OD"})
```

## UpdateCabState

This function updates the cabin status, reason for cabin blocking and blocking period.

**Note:** Not supported when it is connected to an offline QCI Sync database.

### Input Parameters

psFunction= "UpdateCabState" — below is the list of psParam or poParam.

**Table 4-171 psParam**

Parameter	Type	Description	Required
psCabId	String	Unique Cabin ID.	Yes
psCabStat	String	Cabin Status (OO, OV, BL, AV).	Yes
psComment	String	Cabin Blocking Reason.	
pdBlockFromDate	Date	Cabin Block From Date.	
pdBlockToDate	Date	Cabin Block To Date.	

### Return Values

CResponseXml.bSuccess always return True, always show the CResponse.sErrMsg.

#### Example:

```
oResponse = goWS.FidelioSPMSWSXML("UpdateCabState", "SessionID", New Object()
{12345, '00', <Comment>, <BlockFromDate>, <BlockToDate>})
```



# UpdateComment

This function adds or updates the guest comment.

**Note:** Not supported when it is connected to an offline QCI Sync database.

## Input Parameters

psFunction= "UpdateComment" — below is the list of psParam or poParam.

**Table 4-172 psParam**

Parameter	Type	Description	Required
peQuickAssignment	Integer	0-None,1-Same Cabin,2-Same Booking No,3-Same Group, this apply to new comment only, when adding a new comment. If the value is > 0, it will add the same comment to those related guest.	Yes
pnCplID	Integer	Unique Comment ID (0 for add new).	Yes
pnCplLinkID	Integer	This is the comment link ID, use to update the comment create with Quick Assignment only apply to parent comment.	Yes
pnAcctID	Integer	Unique Account ID.	Yes
psComment	String(1500)	Comment Description.	Yes
psCommentCategory	String(6)	Comment Category (TYP_CMC.TYP_ART).	Yes
pnCost	Decimal	Comment/Resolution Cost.	Yes
psDepartmentInvolve	Array of String(6)	List of Department Involve Code (TYP_CDI.TYP_ART).	Yes
psDescCode	String(6)	Comment Description Code (TYP_CPL.TYP_ART). Resolution Description Code (TYP_RES.TYP_ART).	Yes
psDesc	String(1500)	Comment Description.	Yes
psExternalCode	String(20)	External ID.	Yes
psFileName	String(300)	This the original filename of the uploaded document. For picture, it uses the file extension to determine if the document is a doc or a picture, file extension .DOC, .DOCX and .PDF will be considered a document and the rest as a picture file.	Yes
pbInternal	Boolean	Internal Flag.	Yes
psInvolvePerson	String(50)	Person who resolved the issue or Person who reported the complaint.	Yes
pbComment	Boolean	True = Comment, False = Resolution.	Yes
pbSensitive	Boolean	Sensitive Flag.	Yes

**Table 4-172 (Cont.) psParam**

Parameter	Type	Description	Required
pbUrgent	Boolean	Urgent Flag.	Yes
pbyPicture	Array of Byte	This can be a document or a picture, max size 4,000,000.	Yes
psReportType	String(6)	Report Type (TYP_CPI.TYP_ART).	Yes
pdRequestDate	Date	Requested resolution date.	Yes
pbResolved	Boolean	This only applies to resolution.	Yes
pdResolvedDate	Date	Resolved Date	Yes
pnVendorID	Integer	Vendor ID, we either send the Account ID or Vendor ID.	Yes
psVid	String(100)	External Unique ID for use with import.	Yes
pnXDocID	Long	Unique XDOC ID, the table where the picture is stored.  If the value pass in = 0 and picture pbyPicture is passed in, it will create a new record and delete the old record.  If the value pass in > 0 and picture pbyPicture is pass in, it will overwrite the existing record.  If the value pass in = 0 and pbyPicture is nothing then it will delete the existing record.  If the value pass in > 0 and pbyPicture is nothing then it will keep the existing record.	Yes

**Return Values**

CResponseXml.bSuccess = False when Update/Add failed.

CResponseXml.bSuccess = True when Update/Add successful.

CResponseXML.oObj(0) = cpl\_id, comment unique ID.

CResponseXML.oObj(1) = cpl\_link\_id, comment link ID.

**Example:****Add new comment**

```
oResponse = FidelioWS("UpdateComment", New Object() {0, 0, 0, 12345, "Comment", _
"CAT1", 0.00, New String() {"HSE"}, "MISB", "Missing Bag", "", "Bag.jpg", False,
"Person Name", True, False, Yes, File.ReadAllBytes("Bag.Jpg"), "FD",
DateTime.SpecifyKind(Now(), DateTimeKind.Unspecified), False,
DateTime.SpecifyKind(Now(), DateTimeKind.Unspecified), 0, "", 0})
```

**Update existing comment**

```
oResponse = FidelioWS("UpdateComment", New Object() {0, 456, 457, 12345,
"Comment", "CAT1", 0.00, New String() {"HSE"}, "MISB", "Missing Bag", "",
```

```
"Bag.jpg", False, "Person Name", True, False, Yes,
File.ReadAllBytes("Bag.Jpg"), "FD", DateTime.SpecifyKind(Now(),
DateTimeKind.Unspecified), False, DateTime.SpecifyKind(Now(),
DateTimeKind.Unspecified), 0, "", 0})
```

## UpdateCompleteStatus

This function updates the maintenance work order completion status.

**Note:** Not supported when it is connected to an offline QCI Sync database.

### Input Parameters

psFunction= "UpdateCompleteStatus" — below is the list of psParam or poParam.

**Table 4-173 psParam**

Parameter	Type	Description	Required
pnMwrID	Integer	Unique Work Order ID (0 for add new record).	Yes
pnStatus	Integer	Completion Status, must be 0, 25, 50, or 100.	Yes
pdCompleteDate	Date	Completion Date, only use when pnStatus = 100.	Yes

### Return Values

CResponseXml.bSuccess = False when Update failed.

CResponseXml.bSuccess = True when Update successful.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("UpdateCompleteStatus", "SessionID", New
Object() {12345, 100, DateTime.SpecifyKind(#01-01-2013 3:00:00 pm#,
DateTimeKind.Unspecified)})
```

## UpdateCreditCardInfo

This function to updates the guest credit card information.

To deactivate all credit card pass in, set psCardNumber and psCCTrack = Blank String.

### Input Parameters

psFunction= "UpdateCreditCardInfo" — below is the list of psParam or poParam.

**Table 4-174 psParam**

Parameter	Type	Description	Required
pnAcclD	Integer	SPMS Internal Account Number.	Yes

Table 4-174 (Cont.) psParam

Parameter	Type	Description	Required
psCardNumber	String	Credit Card Number.	Yes
psCardExpMMYY	String	Card Expiry Date in MMY format.	Yes
psCardName	String	Card Holder Name.	Yes
psCardCtlNo	String	3 Digits CVV.	Yes
psCCTrack	String	Credit Card Track data.	Yes
pbySignature	Byte Array	Byte Array of the signature image, this only apply when PAR->General->Enable Signature Capture for Credit Card is enable.	Yes
pbyEncTrack1	Byte Array	Papagena Encrypted Track 1.	Optional
pbyEncTrack2	Byte Array	Papagena Encrypted Track 1.	Optional
pbyEncTrack3	Byte Array	Papagena Encrypted Track 1.	Optional
psDeviceSerial	String(32)	Papagena Device Serial Number.	Optional
psKSN	String(32)	Papagena KSN Number.	Optional
psToken	String(120)	Papagena Token.	Optional
pbAssignInvoice1	Boolean	Assign the Credit Card to Invoice 1 (Default = True).	Optional
pbAssignInvoice2	Boolean	Assign the Credit Card to Invoice 2 (Default = True).	Optional
pbAssignInvoice3	Boolean	Assign the Credit Card to Invoice 3 (Default = True).	Optional
pbAssignInvoice4	Boolean	Assign the Credit Card to Invoice 4 (Default = True).	Optional
pnParcelado	Enum	0 – Unknown (Pass in this value, the system will check is the card entitle for Parcelado). 1 – No (Default). 2 - Yes.	Optional
pnDCC	Enum	0 – Unknown (Pass in this value, the system will check is the card entitle for DCC). 1 – No (Default). 2 – Yes.	Optional
pbAllowRegisterSameCard	Boolean	True = Allow to register credit which already register to other guest False = Allow to register credit which already register to other guest depend on PAR.	Optional
psCardCurrency	String(3)	Three characters of ISO currency code.	Optional

Below are the descriptions of the parameters.

**Note:** For the first card, it always assigns to all invoice regardless of the pbAssignInvoice?

**Value**

If pnParcelado = "Unknow" and the card entitle for Parcelado, it will return an error. If the error ends with "?". This means you can ask for a user confirmation. The error message will be per below, and you can check for the word Parcelado to determine if this is the question for Parcelado, or else the question will be for pnDCC.

If the user answer yes, pass in pnPaceldo = 2, if the user answer no, pass in pnParcelado = 1.

"This card is eligible for Parcelado Handling – would you like to enroll this card?".

If pnDCC = "Unknow" and the card entitle for DCC, it will return and error ending with "?" This means you can ask for a user confirmation. If the user answer = yes, pass in pnDCC = 2, If the user answer = no, pass in pnDCC = 1.

### **pnAccID**

You can get this value from the poUxpRes field `uxp_a_id`.

### **psCardNumber**

This is the credit card number. Supply this field only when the credit card was manually entered.

### **psCardExpMMYY**

This is the credit card expiry date (MMYY). Supply this field only when the credit card was manually entered.

### **psCardName**

This is the credit card holder name. Supply this field only when the credit card was manually entered.

### **psCardCtlNo**

This is the CVV number printed on the back of the credit card. This field is optional.

### **psCCTrack**

This is the credit card track data. Supply this field only when the credit card was swiped.

The field must include the track preamble and postamble.

For example:

```
%B4509190000858834^VISA  
<NAME>^2019010142780000000000182000000?;4xxxxxxxxxxxxxxxx=20190101427818200  
000?
```

### **pbySignature**

Byte Array of the signature image, this only applies when PAR. 'General', 'Enable Signature Capture for Credit Card' is enabled. You only pass in when the signature capture device is available.

**Return Values**

CResponseXml.bSuccess = True when update is successful.

CResponseXml.bSuccess = False when update has failed.

**Example:**

```
oResponse = goWS.FidelioSPMSWSXML("UpdateCreditCardInfo", "SESSIONID", New
Object() { 1234, <CardNumber>, <ExpiryDate>, <CardName>, <Card CVV>, <Card All
Track>, <Signature Byte Array>})
```

## UpdateCreditLimit

This function updates the guest credit limit. The credit limit also depends on the payment type. If the credit limit is greater than the payment type credit limit, it will take the payment type credit limit.

**Input Parameters**

psFunction= "UpdateCreditLimit" — below is the list of psParam or poParam.

**Table 4-175 psParam**

Parameter	Type	Description	Required
peSearchType	Enum	0 = Internal ID. 1 = PGID. 2 = BoardCard. 3 = Pax ID.	Yes
psSearchStr	String	Unique Payer Account ID or Board Card or PGID or Pax ID depends on the peSearchType.	Yes
psCreditLimit	Decimal	The credit limit, -1 for unlimited.	Yes
pbDailyLimit	Boolean	This applies to the buyer only. This requires the PAR->General->Check Buyer Limit to be enabled. True – The amount is daily limit. False – The amount is cruise limit.	Yes

**Return Values**

CResponseXml.bSuccess = False when Update failed.

CResponseXml.bSuccess = True when Update successful.

**Example:**

```
oResponse = goWS.FidelioSPMSWSXML("UpdateCreditLimit", "SessionID", New Object()
{0,12345,5000.00,False})
```

## UpdateCrewPinCode

This function updates the crew pin code.

### Input Parameters

psFunction= "UpdateCrewPinCode" — below is the list of psParam or poParam.

**Table 4-176 psParam**

Parameter	Type	Description	Required
pnAcclID	Integer	Unique Crew Account ID.	Yes
pnNewPinCode		New pin code.	Yes

### Return Values

CResponseXml.bSuccess = False when Update failed.

CResponseXml.bSuccess = True when Update successful.

### Example:

```
oResponse = goWS.FidelioSPMSWSXML("UpdateCrewPinCode", "SessionID", New
Object() {12345, 1234})
```

## UpdateDayNeeded

This function updates the maintenance work order day needed to complete the task.

**Note:** Not supported when it is connected to an offline QCI Sync database.

### Input Parameters

psFunction= "UpdateDayNeeded" — below is the list of psParam or poParam.

**Table 4-177 psParam**

Parameter	Type	Description	Required
pnMwrlID	Integer	Unique Work Order ID (0 for add new record).	Yes
pnDays	Double	Number of Days Needed.	Yes

### Return Values

CResponseXml.bSuccess = False when Update failed.

CResponseXml.bSuccess = True when Update successful.

### Example:

```
oResponse = goWS.FidelioSPMSWSXML("UpdateDayNeeded", "SessionID", New
Object() {12345, 1.5})
```

## UpdateEmail

This function allows you to update the e-mail address (RES\_EMAIL\_SHIP) to SPMS.

### Input Parameters

psFunction= "UpdateEmail" — below is the list of psParam or poParam.

**Table 4-178 psParam**

Parameter	Type	Description	Required
pnAcclID	Integer	SPMS Guest Unique ID.	Yes
psOldEMail	String	Old E-Mail address.	Yes
psNewEMail	String	New E-Mail address.	Yes

Below are the descriptions of the parameters.

### pnAcclID

Unique Guest ID, return from SearchPassenger.

### psOldEmail

This is the existing e-mail address for validation purpose. If there is no previous e-mail address pass in an empty string.

### psNewEmail

This is the new e-mail address to update to the DB.

### Return Values

CResponseXml.bSuccess = True when UpdateEmail is successful.

CResponseXml.bSuccess = False when UpdateEmail has failed.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("UpdateEmail", "SESSIONID", New Object()
{12345, "[EMAIL ADD]", "[EMAIL ADD]", 0, ""})
```

## UpdateGuestInfo

This function updates the guest information retrieved from **GuestSearch** function. There could be instances whereby, after the **GuestSearch** function is called, some updates could have taken place to the same account in the SPMS system on the customer side. In cases like this, an error message 'xxxxxx' will be returned by the function. This is to prevent the new information from being overwritten on either end. When this error message occurs, you must make another function call to the **GuestSearch** function or a refresh of the underlying data table and then only make a call to the **UpdateGuestInfo** function, to update the account details.



## Input Parameters

psFunction= "UpdateGuestInfo" — below is the list of psParam or poParam.

**Table 4-179 psParam**

Parameter	Type	Description	Required
poUxpRes	String/ Datatable	This is the data table send from GuestSearch. However, only one record is allowed at one time, JSON – string, XML = Datatable.	Optional when pnAccID is required.
pbyPicture	Byte Array	Byte Array of the picture content.	Yes
psPictureSource	Integer	How picture is capture (This is required when the pbyPicture exists) 0 – Unknown source. 1 – From Web Cam. 2 – From RFID passport. 3 – From Scan passport image.	Yes
pnAccID	Integer	SPMS Internal Account Number (For JSON pass in 0 to add new reservation).	Optional when poUxpRes is required.
psReason	String	Reason for change cabin.	Optional
psNatID	String	Nationality ID.	Optional

Below are the descriptions of the parameters:

For XML pass in poUxpRes.Rows(0)!uxp\_a\_id = 0 to add new reservation.

To update picture only pass in poUxpRes as nothing for XML, empty string for JSON

### poUxpRes

This is the data table returned from the **GuestSearch** function. However, only one record is allowed at any one time. Therefore, if the **GuestSearch** function returns more than one record, you must copy the record that you want to a new data table. Refer to the sample code below on how this is done. Do note that not all fields will be updated. Below is the list of fields currently allowed for an update.

For **FidelioSPMSWSXML** function, the poUxpRes need to serialize first before using the function below.

```
Public Function SerializeObject(ByVal poObject As Object) As String
    Dim oWriter As XmlSerializer
    Dim oMemory As MemoryStream

    oWriter = New XmlSerializer(poObject.GetType)
    oMemory = New MemoryStream
    oWriter.Serialize(oMemory, poObject)
    SerializeObject =
System.Text.UnicodeEncoding.Default.GetString(oMemory.ToArray)
    oMemory.Dispose()
    oMemory = Nothing
```

```

oWriter = Nothing
End Function

```

**Table 4-180 Data Field Codes**

<b>Code</b>	<b>Description</b>	<b>Required</b>
RES_AWARD_LEVEL	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
RES_BOARDCC_EXT	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
RES_BOOKNR	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
RES_CAB	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
RES_CANCEL	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
RES_CLASSIFICATION	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
RES_CRUISE_DAYS	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
RES_DIS_E	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
RES_EMAIL_SHIP	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
RES_EMB_E	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
RES_GRP	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
RES_HANDICAPCODE	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
RES_HANDIREMARKS	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
RES_NOOFCRUISES	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
RES_PAXCLASS	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
RES_PPD_TAG	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
RES_QUEUEID	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
RES_RFID	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
RES_SIGNOFFPORT_ID	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
RES_SIGNONPORT_ID	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
RES_TRAVEL_TYPE	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional

**Table 4-180 (Cont.) Data Field Codes**

<b>Code</b>	<b>Description</b>	<b>Required</b>
RES_V_GUESTID	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
RES_V_PRIMARY	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_A_CITY	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_A_COUNTRY	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_A_FRQ_CARDNO	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_A_FSTN	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_A_HANDICAPED	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_A_LOYALTY	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_A_NAME	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Yes
UXP_A_OTHNAME	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_A_SALUT	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_A_STATE	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_A_STREET	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_A_STREET2	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_A_STREET3	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_A_TITEL	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_A_TRAV_INSURE	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_A_ZIP	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_C_INSURE	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_H_BIRTHD	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_H_BIRTHNAT	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_H_BIRTHP	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional

**Table 4-180 (Cont.) Data Field Codes**

<b>Code</b>	<b>Description</b>	<b>Required</b>
UXP_H_CONT_CITY	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_H_CONT_ZIP	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_H_CONT_COUNTRY	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_H_CONT_EMAIL	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_H_CONT_MOB_PHONE	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_H_CONT_NAME	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_H_CONT_PHONE	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_H_CONT_STATE	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_H_CONT_STREET	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_H_CONT_STREET2	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_C_RELATIONSHIP	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_H_EMAIL	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_H_EMAIL2	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_H_MNAME	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_H_NATION	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_H_PASSDI	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_H_PASSEX	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_H_PASSNO	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_H_PASSPI	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_H_PASSPI_COUNTRY	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_H_REPEATER	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_H_REPEATER_OTHER	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional

**Table 4-180 (Cont.) Data Field Codes**

<b>Code</b>	<b>Description</b>	<b>Required</b>
UXP_H_SEX	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_H_TELNO	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_P_PROFESSION	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_P_VISA_ADDRESS1	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_P_VISA_ADDRESS2	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_P_VISA_CITY	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_P_VISA_EMAIL	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_P_VISA_PHONE	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_P_VISA_STATE	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_P_VISA_STREET	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_P_VISA_ZIP	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
RES_REMARKS2	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_H_LANG	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_H_LANG2	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_H_LANG3	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_H_LANG4	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_C_EXTERNALID	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_H_MARTIAL	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_H_HAIR	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_H_EYES	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_H_WEIGHT	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_H_HEIGHT	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional

**Table 4-180 (Cont.) Data Field Codes**

<b>Code</b>	<b>Description</b>	<b>Required</b>
UXP_H_MOB_TELNO	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_H_STAFF_DIV	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_H_STAFF_POS	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_C_BOOKNO	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_C_BKDATE	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_C_BKPASSEX	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_C_BKPASSPI	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_C_BKPASSPI_COUNTRY	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional
UXP_A_OTHFSTN	If no changes, please skip this field. See SPMS Data Dictionary for further description.	Optional

**pbyPicture**

This is the byte array of the picture file.

**pnAccID**

You can get this value from the poUxpRes field UXP\_A\_ID.

**psReason**

Input is optional when the absent of psNatID.

Input is required when the present of psNatID, pass this value as empty string if NO input reason needed.

**psNatID**

This field is optional.

**Return Values**

CResponseXml.bSuccess = True when update is successful.

CResponseXml.bSuccess = False when update has failed.

When success CResponseXml.oObj(0) = <uxp\_a\_id>

**Example:**

```
oResponse = goWS.FidelioSPMSWSXML("UpdateGuestInfo", "SESSIONID", New Object()
{ SerializeObject(oUxpRes), <Picture Byte Array>, 1})
```

## UpdateGuestMessage

This function adds or updates the guest message.

**Note:** Not supported when it is connected to an offline QCI Sync database.

### Input Parameters

psFunction= "UpdateGuestMessage" — below is the list of psParam or poParam.

**Table 4-181 psParam**

Parameter	Type	Description	Required
pnMsgD	Integer	Unique Message ID (0 for add new record).	Yes
pnAcclD	Integer	Unique Account ID.	Yes
psSender	String(50)	Sender.	Yes
psMessage	String(1000)	Message to the guest.	Yes
peDeliveryMode	Enum	0-NoDelivery. 1-DeliverAtAdvanceQCIWizard. 2-DeliverAtGangway.	Yes
peAcknowledgement	Enum	0-NoAcknowledge. 1-UserAcknowledgeWithLog. 2-ManagerAcknowledgeWithLog. 3-DoNotContinue (8.0 only).	Yes
peType (8.0 only)	Enum	0-Standard. 1-Priority. 2-Silent.	Yes
pdStartDate (8.0 only)	Integer	Nothing=Always Active.	Yes
pdExpiry (8.0 only)	Date	Nothing=Always Active.	Yes

### Return Values

CResponseXml.bSuccess = False when Update/Add failed.

CResponseXml.bSuccess = True when Update/Add successful.

### Example:

```
oResponse = goWS.FidelioSPMSWSXML("UpdateGuestMessage", "SessionID", New
Object() {12345,1234576,"Guest","Guest have parcel in front desk",2,1,
0,DateTime.SpecifyKind(#01-01-2014 3:00:00 pm#, DateTimeKind.Unspecified),
DateTime.SpecifyKind(#06-01-2014 3:00:00 pm#, DateTimeKind.Unspecified)})
```

## UpdateInvoiceStatus

This function updates the Invoice batch print status in SPMS.

**Note:** Not supported when it is connected to an offline QCI Sync database.

**Input Parameters**

psFunction= "UpdateInvoiceStatus" — below is the list of psParam or poParam.

**Table 4-182 psParam**

Parameter	Type	Description	Required
pnAcclD	Integer	Unique Guest ID.	Yes
pbBatchPrint	Boolean	True = Print in Quick Billing Screen. False = Do not print in Quick Billing Screen.	Yes
psEMail	String(72)	Ship Email Address.	Yes

Below are the descriptions of the parameters.

**pnAcclD**

Unique Guest ID, return from SearchPassenger.

**pbBatchPrint**

True – Print in Quick Billing Screen.

False – Do not print in Quick Billing Screen.

**psEMail**

e-Mail address to send the invoice to.

**Return Values**

CResponseXml.bSuccess = True when UpdateInvoiceStatus is successful.

CResponseXml.bSuccess = False when UpdateInvoiceStatus has failed.

**Example:**

```
oResponse = goWS.FidelioSPMSWSXML("UpdateInvoiceStatus", "SESSIONID", New
Object() {12345, True, "[EMAIL ADD]"})
```

## UpdateLuggageLink

This function links the luggage to the guest. If the luggage is already linked to a different guest, it will overwrite it.

**Input Parameters**

psFunction= "UpdateRFIDEncoded" — below is the list of psParam or poParam.



**Table 4-183 psParam**

Parameter	Type	Description	Required
psSearchStr	String	Unique Payer Account ID or Board Card or PGID or Pax ID depends on the peSearchType.	Yes
psRFIDUID	String(16)	Convert the value to HEX.	Yes
peSearchType	Enum	InternalID = 0 'res_acc PGID = 1 'res_pgid BoardCard = 2 'res_boardcc PaxID = 3 'res_referenz Cabin = 4 'res_cab Default value = InternalID	Optional

**Return Values**

CResponseXml.bSuccess = False when Update failed.

CResponseXml.bSuccess = True when Update successful.

**Example:**

```
oResponse = goWS.FidelioSPMSWSXML("UpdateLuggageLink", "SessionID", New
Object() {12345, "E01581A07C123B"})
```

## UpdateMessageStatus

This function updates the message delivered status.

**Note:** Not supported when it is connected to an offline QCI Sync database.

**Input Parameters**

psFunction= "UpdateMessageStatus" — below is the list of psParam or poParam.

**Table 4-184 psParam**

Parameter	Type	Description	Required
pnMsgID	Integer	SPMS Unique Message ID.	Yes
pbDelivered	Boolean	True = Set the message status to delivered. False – Set the message status to not delivered.	Yes

Below are the descriptions of the parameters.

**pnMsgID**

SPMS Unique Message ID, return from GetMessage.

**pbDelivered**

True = Set the message status to delivered.

False – Set the message status to not delivered.

**Return Values**

CResponseXml.bSuccess = True when UpdateMessageStatus is successful.

CResponseXml.bSuccess = False when UpdateMessageStatus has failed.

**Example:**

```
oResponse = goWS.FidelioSPMSWSXML("UpdateMessageStatus", "SESSIONID", New
Object() {12345, #1/1/2011#, #1/12/2011#, 0, ""})
```

## UpdatePostingStatus

This function updates the guest posting status.

**Input Parameters**

psFunction= "UpdatePostingStatus" — below is the list of psParam or poParam.

**Table 4-185 psParam**

Parameter	Type	Description	Required
peSearchType	Enum	0 = Internal ID. 1 = PGID. 2 = BoardCard. 3 = Pax ID.	Yes
psSearchStr	String	Unique Payer Account ID or Board Card or PGID or Pax ID depends on the peSearchType.	Yes
pnStatus	Byte	0 – Posting disable 1 - Posting enable. 2 – Posting disable but manual posting allow.	Yes
psReason	String(900)	The reason to change the status.	Yes

**Return Values**

CResponseXml.bSuccess = False when Update failed.

CResponseXml.bSuccess = True when Update successful.

**Example:**

```
oResponse = goWS.FidelioSPMSWSXML("UpdatePostingStatus", "SessionID", New
Object() {0,12345,0,"Credit Limit Exceeded"})
```

## UpdatePromoterStatus

This function notifies the SPMS that the promotion has been view in the web client.

**Note:** Not supported when it is connected to an offline QCI Sync database.

### Input Parameters

psFunction= "UpdatePromoterStatus" — below is the list of psParam or poParam.

**Table 4-186 psParam**

Parameter	Type	Description	Required
pnPrdID	Integer	Promotion ID.	Yes
pnAcclID	Integer	Unique Account ID.	Yes
psCode	String(20)	Promotion Code. This code is return from field ESE_ID in WFunction, GetPromotWebClient.	Yes

### Return Values

CResponseXml.bSuccess = False when update failed.

CResponseXml.bSuccess = True when update successful.

Example:

```
oResponse = FidelioWS("UpdatePromoterStatus", New Object() {3565, 12345, "73B"})
```

## UpdateRFIDEncoded

This function marks the RFID as encoded successfully. The GETRFIDEncodingData, ENCODED field will return "1" when this function is called and successfully executed.

**Note:** Not supported when it is connected to an offline QCI Sync database.

### Input Parameters

psFunction= "UpdateRFIDEncoded" — below is the list of psParam or poParam.

**Table 4-187 psParam**

Parameter	Type	Description	Required
psRFIDUID	String(16)	Convert the value to HEX.	Yes

### Return Values

CResponseXml.bSuccess = False when Update failed.

CResponseXml.bSuccess = True when Update successful.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("UpdateRFIDEncoded", "SessionID", New Object()
{"E01581A07C123B"})
```

## UpdateSpecialRequest

This function adds or updates the guest special request.

**Note:** Not supported when it is connected to an offline QCI Sync database.

### Input Parameters

psFunction= "UpdateSpecialRequest" — below is the list of psParam or poParam.

**Table 4-188 psParam**

Parameter	Type	Description	Required
pnSrtID	Integer	Unique Special Request ID (0 for add new record).	Yes
pnAcclID	Integer	Unique Account ID.	Yes
psCode	String(6)	Special Request Code.	Yes
psVid	String(100)	Special Request Unique External ID.	Yes
psDescription	String(1000)	Special Request Description.	Yes
pnExtID	Double	Special Request External ID.	Yes
psExtCode	String(15)	Special Request External Code.	Yes
pnQuantity	Integer	Special Request Quantity.	Yes
pdDeliveryDate	Date	Expected Delivery Date and Time.	Yes
psLocation	String(30)	Location.	Yes
psOdiID	String(6)	Delivery Division.	Yes
pdEventDate	Date	Special Request Event Date.	Yes

### Return Values

CResponseXml.bSuccess = False when Update/Add failed.

CResponseXml.bSuccess = True when Update/Add successful. CResponseXML.oObj = srt\_id (Unique Special Request ID).

Example:

```
oResponse = goWS.FidelioSPMSWSXML("UpdateSpecialRequest", "SessionID", New
Object() {12345,1234576,"Code","Vid","Description", 2, "ExtCode", 1,
DateTime.SpecifyKind(#01-01-2013 3:00:00 pm#, DateTimeKind.Unspecified),
"Location", "GRP", DateTime.SpecifyKind(#01-01-2013 3:00:00 pm#,
DateTimeKind.Unspecified)})
```

## UpdateTravelDoc

This function adds or updates the travel document.

**Input Parameters**

psFunction= "UpdateTravelDoc" — below is the list of psParam or poParam.

**Table 4-189 psParam**

Parameter	Type	Description	Required
pnVisID	Integer	Unique Travel Document ID (0 for add new record).	Yes
pnAcclID	Integer	Unique Account ID.	Yes
psCountry	String(6)	Country.	Yes
psNumber	String(20)	Document Number.	Yes
psVisType	String(6)	Internal Document Type.	Yes
pdIssueDate	Date	Issue Date.	Yes
psIssuePlace	Date	Issue Place.	Yes
pdExpiryDate	Date	Expiry Date.	Yes
psVid	String(100)	Unique External ID, for Data Import.	Yes
pdRMUDate	Date	RMU Date.	Yes
psDocType	String(6)	Document Type.	Yes
psFirstName	String(40)	First Name.	Yes
psMiddleName	String(20)	Middle Name.	Yes
psLastName	String(40)	Last Name.	Yes
pdBirthDate	Date	Birth Date.	Yes
psRemoteID	String(30)	Remote ID	Yes
psMrz1	String(100)	MRZ 1.	Yes
psMrz2	String(100)	MRZ 2.	Yes
psMrz3	String(100)	MRZ 3.	Yes
psDocName	String(100)	Document Name.	Yes
psGUID	String(50)	GUID, for CMS.	Yes
pnEntry	Integer	Number of Allow Entry.	Yes
pdValidFrom	Date	Valid From.	Yes
pnDurationOfStay	Integer	Duration of Stay.	Yes
pbyDocImage	Byte Array	Document Image.	Yes
pbyDocPhoto	Byte Array	Document Photo.	Yes
pbyMultiPageDocImage	Byte Array	Multipage Scan document image.	Yes
psFileName	String(100)	Original File Name for file upload.	Yes
pbAssignPhotoToAccount	Boolean	True – Assign the Document Photo to Guest Photo.	Yes

**Return Values**

CResponseXml.bSuccess = False when Update/Add failed.

CResponseXml.bSuccess = True when Update/Add successful, CResponseXML.oObj = vis\_id (Unique Document ID).

**Example:**

```
oResponse = goWS.FidelioSPMSWSXML("UpdateTravelDoc", "SessionID", New Object()
{12345,1234576,"US","Bxxxx","P",DateTime.SpecifyKind(#01-01-2009#,
DateTimeKind.Unspecified), "Florida", DateTime.SpecifyKind(#01-01-2019#,
DateTimeKind.Unspecified), "", DateTime.SpecifyKind(#12:00:00 AM#,
DateTimeKind.Unspecified), "P", "FirstName", "", "LastName",
DateTime.SpecifyKind(#01-01-1960#, DateTimeKind.Unspecified), "", "", "", "",
"Passport", "", 999, DateTime.SpecifyKind(#01-01-2009#,
DateTimeKind.Unspecified), 30, Nothing, Nothing, Nothing, "", False)
```

## UpdateTravelDocAdvance

This function updates the travel document with selected fields. To add a new record you must use function [UpdateTravelDoc](#).

### Input Parameters

psFunction= "UpdateTravelDocAdvance" — below is the list of psParam or poParam.

**Table 4-190 psParam for XML**

Parameter	Type	Description	Required
poVis	String/Datatable	You must create a WSF to return all fields from VIS view and the data table. One record is allowed at one time, JSON – string, XML = Datatable. Contact Oracle Support to obtain the WSF scripts.	Yes
pbAssignPhotoToAccount	Boolean	True – Assign the Document Photo to Guest Photo.	Yes

**Table 4-191 psParam for JSON**

Parameter	Type	Description	Required
poVis	String/Datatable	You must create a WSF to return all fields from VIS view and the data table. One record is allowed at one time, JSON – string, XML = Datatable. Please contact Oracle Support to obtain the WSF scripts.	Yes
pnVisID	Integer	Unique Travel Document ID.	Yes
pbAssignPhotoToAccount	Boolean	True – Assign the Document Photo to Guest Photo.	Yes

Below are the descriptions of the parameters.

### poVis

This is the data table returned from the WSF (SELECT \* FROM vis WHERE vis\_id = <PARAMNUM1>). However, only one record is allowed at any one time. For FidelioSPMSWSXML function the poVis need to serialize first using the below function

```
Public Function SerializeObject(ByVal poObject As Object) As String
    Dim oWriter As XmlSerializer
    Dim oMemory As MemoryStream

    oWriter = New XmlSerializer(poObject.GetType)
    oMemory = New MemoryStream
    oWriter.Serialize(oMemory, poObject)
    SerializeObject =
System.Text.UnicodeEncoding.Default.GetString(oMemory.ToArray)
    oMemory.Dispose()
    oMemory = Nothing
    oWriter = Nothing
End Function
```

**Table 4-192** xxxx

Parameter	Type	Description	Required
pnVisID	Integer	Unique Travel Document ID.	Yes
pnAcclID	Integer	Unique Account ID.	Optional
psCountry	String(6)	Country.	Optional
psNumber	String(20)	Document Number.	Optional
psVisType	String(6)	Internal Document Type.	Optional
pdIssueDate	Date	Issue Date.	Optional
psIssuePlace	Date	Issue Place.	Optional
pdExpiryDate	Date	Expiry Date.	Optional
psVid	String(100)	Unique External ID, for Data Import.	Optional
pdRMUDate	Date	RMU Date.	Optional
psDocType	String(6)	Document Type.	Optional
psFirstName	String(40)	First Name.	Optional
psMiddleName	String(20)	Middle Name.	Optional
psLastName	String(40)	Last Name.	Optional
pdBirthDate	Date	Birth Date.	Optional
psRemoteID	String(30)	Remote ID.	Optional
psMrz1	String(100)	MRZ 1.	Optional
psMrz2	String(100)	MRZ 2.	Optional
psMrz3	String(100)	MRZ 3.	Optional
psDocName	String(100)	Document Name.	Optional
psGUID	String(50)	GUID, for CMS.	Optional
pnEntry	Integer	Number of Allow Entry.	Optional
pdValidFrom	Date	Valid From.	Optional
pnDurationOfStay	Integer	Duration of Stay.	Optional

**Table 4-192 (Cont.) xxxx**

Parameter	Type	Description	Required
pbyDocImage	Byte Array	Document Image.	Optional
pbyDocPhoto	Byte Array	Document Photo.	Optional
pbyMultiPageDocImage	Byte Array	Multipage Scan document image.	Optional
psFileName	String(100)	Original File Name for file upload.	Optional

**pbAssignPhotoToAccount**

True – Assign the Document Photo to Guest Photo.

**pnVisID**

You can get this value from the WSF field vis\_id.

**Return Values**

CResponseXml.bSuccess = True when update is successful.

CResponseXml.bSuccess = False when update has failed.

**Example:**

```
oResponse = goWS.FidelioSPMSWSXML("UpdateTravelDocAdvance", "SESSIONID", New
Object() { SerializeObject(oVis), True, <VIS_ID>})
```

## UpdateWorkOrder

This function adds or updates the maintenance work order.

**Note:** Not supported when it is connected to an offline QCI Sync database.

**Input Parameters**

psFunction= "UpdateWorkOrder" — below is the list of psParam or poParam.

**Table 4-193 psParam**

Parameter	Type	Description	Required
pnMwrlD	Integer	Unique Work Order ID (0 for add new record).	Yes
psWorkOrder	String(20)	Work Order Number.	Yes
psPublicLocation	String(6)	Public Location, link to TYP_MWL.TYP_ART.	Yes
psCabin	String(20)	Cabin Number.	Yes
psTaskDesc	String(1000)	Task Description.	Yes
pnCompleteStatus	Integer	Completion Status. Must be 0, 25, 50, or 100 (This only applies to Insert new entry).	Yes
psLocation	String(10)	Location, link to ARE_CODE.	Yes



**Table 4-193 (Cont.) psParam**

Parameter	Type	Description	Required
psPictureFileName	String(50)	Filename of the picture file uploaded.	Yes
psTask	String(6)	Task Code, link to TYP_MWT.TYP_ART, default value XXX.	Yes
psType	String(6)	Task Type, link to TYP_WOT.TYP_ART, default value WOT01.	Yes
pnXDocID	Long	Unique XDOC ID, the table where the picture is stored.	Yes
psPriority	String(6)	Priority Code, link to TYP_MPI.TYP_ART.	Yes
psLocationStatus	String(6)	Location Status, link to TYP_MLS.TYP_ART.	Yes
pbyPicture	Byte Array	Picture File, only needs to send when the picture is added or update. Send nothing or empty array when no picture to update. Existing picture will not be deleted even when no picture is send.	Yes
pbPrint	Boolean	Indicate to print the work order, PAR->Maintenance->Use Work Order Printer Service must be enabled in order to print.	Yes
psComment	String(2000)	Comment Field.	Yes
psReportByCabID	String(20)	Reported By Cabin Number.	Yes
pbRejected	Boolean	Rejected Status.	Yes
psReportByUser	String(50)	Reported By Person.	Optional
pdReportedDate	Date	Reported Date.	Optional

**Note:** psPublicLocation, psCabin and psLocation, only one of the three variable needs to pass in.

### Return Values

CResponseXml.bSuccess = False when Update/Add failed.

CResponseXml.bSuccess = True when Update/Add successful.

CResponesXml.oObj = mwr\_id when bSuccess = True.

Note:

1. pnXDocID = 0 and picture upload during update/insert means the add new picture.
2. pnXDocID = 0 and no picture upload during update mean sremove picture.
3. pnXDocID = 0 and no picture upload during insert means no picture.
4. pnXDocID > 0 and picture upload during update/insert means overwrite the existing picture.
5. pnXDocID > 0 and no picture upload during update/insert means use the existing picture.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("UpdateWorkOrder", "SessionID", New Object()
{12345,"1234576","LOC1","","Testing", 0, "", "", "XXX", "WOT01", 0, "1" , "1",
Nothing, True)
```

## UpdateWorkOrderAssignment

This function will assign person/group to the maintenance work order.

**Note:** Not supported when it is connected to an offline QCI Sync database.

### Input Parameters

psFunction= "UpdateWorkOrderAssignment" — below is the list of psParam or poParam.

**Table 4-194 psParam**

Parameter	Type	Description	Required
pnMwalD	Integer	Unique Work Order Assignment ID (0 for add new record).	Yes
pnMwrID	Integer	Unique Work Order ID.	Yes
psAssignGroup	String(6)	Assignment Group, link to TYP_MWG.TYP_ART.	Yes
psAssignCrew	Integer	Unique Crew ID.	Yes
psRemark	String(1000)	Remarks.	Yes

### Return Values

CResponseXml.bSuccess = False when Update/Add failed.

CResponseXml.bSuccess = True when Update/Add successful.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("UpdateWorkOrderAssignment", "SessionID", New
Object() {12345,123457, "1", 12345, "Testing")
```

## UpdateWorkOrderImage

This function adds or updates the maintenance work order additional image.

**Note:** Not supported when it is connected to an offline QCI Sync database.

### Input Parameters

psFunction= "UpdateWorkOrderImage" — below is the list of psParam or poParam.

**Table 4-195 psParam**

Parameter	Type	Description	Required
pnMwpID	Integer	Unique Work Order Image ID (0 for add new record).	Yes
pnMwrID	Integer	Unique Work Order ID.	Yes

**Table 4-195 (Cont.) psParam**

Parameter	Type	Description	Required
psFileName	String(50)	Original filename of the image file.	Yes
psComment	String(100)	Comment on the image.	Yes
pbyImage	Byte Array	Image.	Yes

**Return Values**

CResponseXml.bSuccess = False when Update/Add failed.

CResponseXml.bSuccess = True when Update/Add successful.

**Example:**

```
oResponse = goWS.FidelioSPMSWSXML("UpdateWorkOrderImage", "SessionID", New
Object() {12345,123457,"Cabin.jpg","Shower not working",<Image byte
array>})
```

## UpdateWorkOrderItem

This function adds or updates the maintenance work order item required.

**Note:** Not supported when it is connected to an offline QCI Sync database.

**Input Parameters**

psFunction= "UpdateWorkOrderItem" — below is the list of psParam or poParam.

**Table 4-196 psParam**

Parameter	Type	Description	Required
pnMwIID	Integer	Unique Work Order Item ID (0 for add new record).	Yes
pnMwrID	Integer	Unique Work Order ID.	Yes
pnItemCode	Integer	Unique Maintenance Item ID, link to MIT_ID.	Yes
pnQuantity	Integer	Quantity.	Yes
pnPrice	Double	Price.	Yes
psRemark	String(500)	Remarks.	Yes

**Return Values**

CResponseXml.bSuccess = False when Update/Add failed.

CResponseXml.bSuccess = True when Update/Add successful.

**Example:**

```
oResponse = goWS.FidelioSPMSWSXML("UpdateWorkOrderItem", "SessionID", New
Object() {12345,123457,1,1,2.0, "Testing"})
```

## ValidateCrewPinCode

This function validates the crew PIN code to see if it matches with the existing PIN code.

### Input Parameters

psFunction= "ValidateCrewPinCode" — below is the list of psParam or poParam.

**Table 4-197 psParam**

Parameter	Type	Description	Required
pnAcclID	Integer	Unique Crew Account ID.	Yes
pnPinCode	Integer	New pin code.	Yes

### Return Values

CResponseXml.bSuccess = False when Update failed.

CResponseXml.bSuccess = True when Update successful.

### Example:

```
oResponse = goWS.FidelioSPMSWSXML("UpdateCrewPinCode", "SessionID", New Object()
{12345, 1234})
```

## Version

This function returns the version number of the OHCWeb Services in use. This is used to determine a list of available services for each version as more services will be added from time to time. It is advisable to check the version number of the OHCWeb Services to ensure the client application is backward compatible and to determine the list of services available before proceeding.

### Input Parameters

psFunction= "Version". There is no input parameter and sends NULL to psParam or poParam.

### psParam

None.

### Return Values

CResponseXml.oObj = Version number of the web service in use.

### Example:

```
oResponse = goWS.FidelioSPMSWSXML("Version", "SESSIONID", Nothing)
```

## WSFunction

This function retrieves information from SPMS by calling a user defined function. You must read a separate document on the available function as they differ from customer to customer.

### Input Parameters

psFunction= "WsFunction" — below is the list of psParam or poParam.

**Table 4-198 psParam**

Parameter	Type	Description	Required
poParam	Array	Array of parameter, this parameter different from function to function.	Optional

Below are the descriptions of the parameters.

### poParam

The first element is the function name, the rest depends on one function to another.

If the parameter is date field, you must pass in a date string in YYYYMMDDHHMMSS format.

### Return Values

CResponseXml.bSuccess = False when input values are invalid or no record found.

CResponseXml.bSuccess = True and CResponseXML.oTables = Array of ADO.NET data table when input values are valid.

Example:

```
oResponse = goWS.FidelioSPMSWSXML("WSFunction", "SESSIONID", New Object()  
{"GetCabin", "05002"})
```