

# CRM API

User Guide



Copyright © 2021, 8x8, Inc. All rights reserved.

This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

8x8® is a registered trademark of 8x8, Inc.

All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owner/s. All other brands and/or product names are the trademarks (or registered trademarks) and property of their respective owner/s.

## Contents

---

<b>Overview: 8x8 CRM API</b> .....	<b>7</b>
Customer .....	7
Case .....	7
Follow-Up .....	8
Task .....	8
Objects .....	9
Actions .....	9
<b>Use CRM API</b> .....	<b>11</b>
Configure CRM API Access .....	11
Incoming Requests .....	13
<i>List of Parameters</i> .....	13
Use Special Characters in CRM API Requests .....	14
Responses to Queries .....	15
<i>Examples of Queries and Responses</i> .....	16
<b>Manipulate Customer Objects</b> .....	<b>19</b>
Add Customer .....	19
<i>Mandatory and Optional Parameters</i> .....	20
Reply from Local CRM to Add Customer .....	22
Modify Customer .....	23
<i>List of Mandatory and Optional Parameters</i> .....	23
Reply from Local CRM to Modify Customer .....	26
Get Customer Details .....	26
<i>Tags to Pair for Range Queries</i> .....	26
<i>Examples of Get Customer Operations</i> .....	27
<i>List of Parameters</i> .....	31
Reply from Local CRM to Get Customer Details .....	32
Delete Customer .....	33
<i>Example of Delete Customer Operation</i> .....	33
<i>List of Mandatory and Optional Parameters</i> .....	33
Reply from Local CRM to Delete Customer .....	34
<b>Manipulate Case Objects</b> .....	<b>35</b>
Add Case .....	35
<i>List of Mandatory and Optional Parameters</i> .....	36
Reply from Local CRM to Add Case .....	38

Modify Case .....	38
<i>List of Mandatory and Optional Parameters</i> .....	39
Reply from Local CRM to Modify Case .....	41
Get Case .....	41
<i>Tags that Accept Substring Values</i> .....	41
<i>Tags that Accept Multiple Parameters</i> .....	42
<i>Tags that Can be Paired for Range Queries</i> .....	42
<i>Tags that Can be paired for Range Queries and Creation Dates</i> .....	42
<i>Examples of Get Case Operations</i> .....	42
<i>List of Parameters</i> .....	50
Reply from Local CRM to Get Case .....	52
Delete Case .....	53
<i>Example of Delete Case Operation</i> .....	53
<i>List of Mandatory and Optional Parameters</i> .....	53
Reply from Local CRM to Delete Case .....	54
<b>Manipulate Follow-Up Objects</b> .....	<b>55</b>
Add Follow-Up .....	55
<i>List of Mandatory and Optional Parameters</i> .....	56
Reply from Local CRM to Add Follow-Up .....	58
Get Follow-Up .....	58
<i>Tags to Pair for Range Queries</i> .....	58
<i>Examples of Get Follow-up Operations</i> .....	59
<i>List of Parameters</i> .....	59
Reply from Local CRM to Get Follow-Up .....	60
Delete Follow-Up .....	61
<i>Example of Delete Follow-Up Operation</i> .....	61
<i>List of Mandatory and Optional Parameters</i> .....	61
Reply from Local CRM to Delete Follow-Up .....	61
<b>Manipulate Task Objects</b> .....	<b>62</b>
Add Task .....	62
<i>List of Mandatory and Optional Parameters</i> .....	63
Reply from Local CRM to Add a Task .....	65
Modify Task .....	65
<i>List of Mandatory and Optional Parameters</i> .....	66
Reply from Local CRM to Modify Task .....	67
Get Task .....	67
<i>Tags that Accept Substring Values</i> .....	68
<i>Tags that Accept Predefined Values</i> .....	68
<i>Tags to Pair for Range Queries</i> .....	68
<i>Examples of Get Queries</i> .....	69
<i>List of Parameters</i> .....	75

Reply from Local CRM to Get Task .....	77
Delete Task .....	79
<i>Example of Delete Task Operation</i> .....	79
<i>List of Mandatory and Optional Parameters</i> .....	79
Reply from Local CRM to Delete Task .....	79
<b>Manipulate FAQ Category Objects .....</b>	<b>80</b>
Add FAQ Category .....	80
<i>Mandatory and Optional Parameters</i> .....	81
Reply from Local CRM to Add FAQ Category .....	81
Modify FAQ Category .....	82
<i>Summary of Mandatory and Optional Parameters</i> .....	82
Reply from Local CRM to Modify FAQ Category .....	83
Get FAQ Category .....	83
Reply from Local CRM to Get FAQ Category .....	83
Delete FAQ Category .....	84
<i>Mandatory and Optional Parameters</i> .....	84
Reply from Local CRM to Delete FAQ Category .....	84
<b>Manipulate FAQ Objects .....</b>	<b>86</b>
Add FAQ .....	86
<i>Summary of Mandatory and Optional Parameters</i> .....	87
Reply from Local CRM to Add FAQ .....	88
Modify FAQ .....	89
<i>Summary of Mandatory and Optional Parameters</i> .....	89
Reply from Local CRM to Modify FAQ .....	90
Get FAQ .....	90
<i>Tags to Pair for Range Queries</i> .....	91
<i>Examples of Get FAQ Operations</i> .....	91
<i>List of Parameters</i> .....	92
Reply from Local CRM to Get FAQ .....	92
Delete FAQ .....	93
<i>Summary of Parameters</i> .....	93
Reply from Local CRM to Delete FAQ .....	93
<b>Manipulate Custom Field Objects and Values .....</b>	<b>95</b>
Get Custom Field .....	95
<i>List of Parameters</i> .....	96
Reply from Local CRM to Get Custom Field .....	96
Modify Custom Field Values .....	96
<i>List of Mandatory and Optional Parameters</i> .....	97

---

Reply from the Local CRM to Modify Custom Field Values .....	98
<b>List Attribute Values .....</b>	<b>99</b>
List COUNTRY Attributes .....	99
Reply from Local CRM to List COUNTRY .....	99
<b>List Tenant Configuration Attributes .....</b>	<b>101</b>
List Agent GROUPS .....	101
<i>Reply to List Agent GROUPS</i> .....	101
List AGENTS .....	102
<i>Reply to List AGENTS</i> .....	102
List FAQ Categories .....	103
<i>Reply to list FAQ Category</i> .....	103
<b>Historical Statistics .....</b>	<b>104</b>
<b>Error Cases .....</b>	<b>105</b>

## Overview: 8x8 CRM API

The 8x8 Virtual Contact Center Local CRM provides a model for managing your contact center's interactions with customers. The Local CRM stores your contact center's customer, case, and follow-up data. To manage customer relationships, the Local CRM provides the following hierarchy of standard objects:

- Customer
- Case
- Follow-up
- Task

### Customer

A customer object allows you to create customer instances with unique account numbers. A customer record stores information such as name, address, and phone number, and offers the capability to customize the customer object with custom fields.

### Case

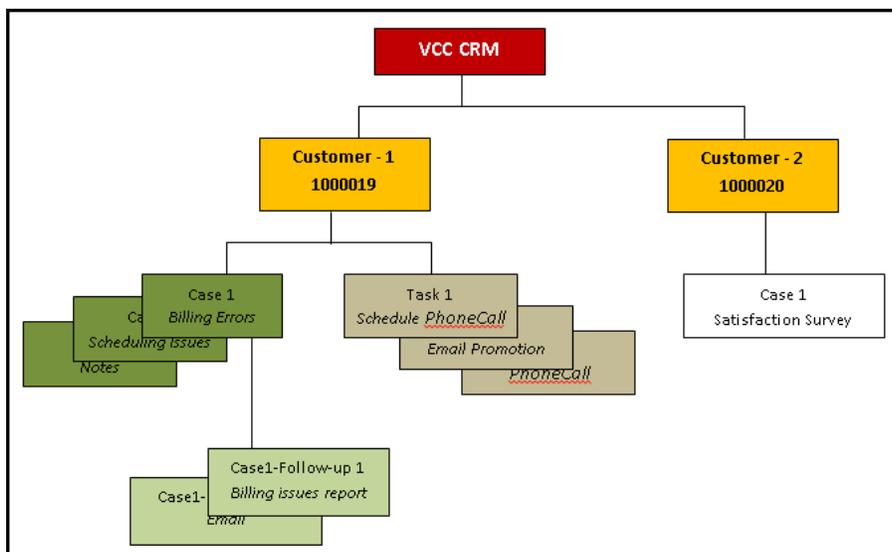
A case object describes customer's feedback, queries, or issues. You can create cases to systematically track and solve customer issues. You can quickly create, update, and view cases. You may indicate the status of a case as open or closed. The standard definition of a case allows you add additional attributes to a case by stating the status, visibility, severity, and priority.

For example, a support agent at AcmeJets creates a case when a customer calls in reporting issues with scheduling private jets. The support agent captures and records the customer-reported issue by creating a case.

## Follow-Up

A case may have multiple follow-ups before it is resolved. Each communication regarding the case from its creation until it is closed may be recorded as individual follow-up instances. This may include notes as well as email communications regarding the case. The follow-up object allows you to track how a case is resolved by creating multiple follow-up records.

For example, when an AcmeJets customer reports scheduling issues, the support agent communicates with the scheduling supervisor to resolve the problem. All these communications by phone or email are transcribed as follow-up records.



## Task

A task refers to a call, email, meeting, chat, or any other type of contact made with a customer. A task organizes and helps you track all interactions associated with a customer. The task object serves to create and manage all tasks you plan to perform or have performed, such as making calls or sending mail. For example, you can create a task to schedule a phone call to a customer.

This document describes how to use the CRM API component of the 8x8, Inc. Integration Suite (formerly referred to as the 8x8, Inc. WAPI). The CRM API component of the Integration Suite enables you to develop external programs that access the following data in Virtual Contact Center:

- Local CRM Customer, Case, Follow-up, and Task objects
- Local CRM custom field data for Customer, Case, Follow-up, and Task objects
- FAQ category and FAQ answer data

The CRM API uses HTTP or HTTPS to transmit requests and return results formatted as Extensible Mark-up Language (XML).

## Objects

- Customer
- Case
- Follow-up
- Task
- FAQ Category
- Custom Fields
- FAQ

## Actions

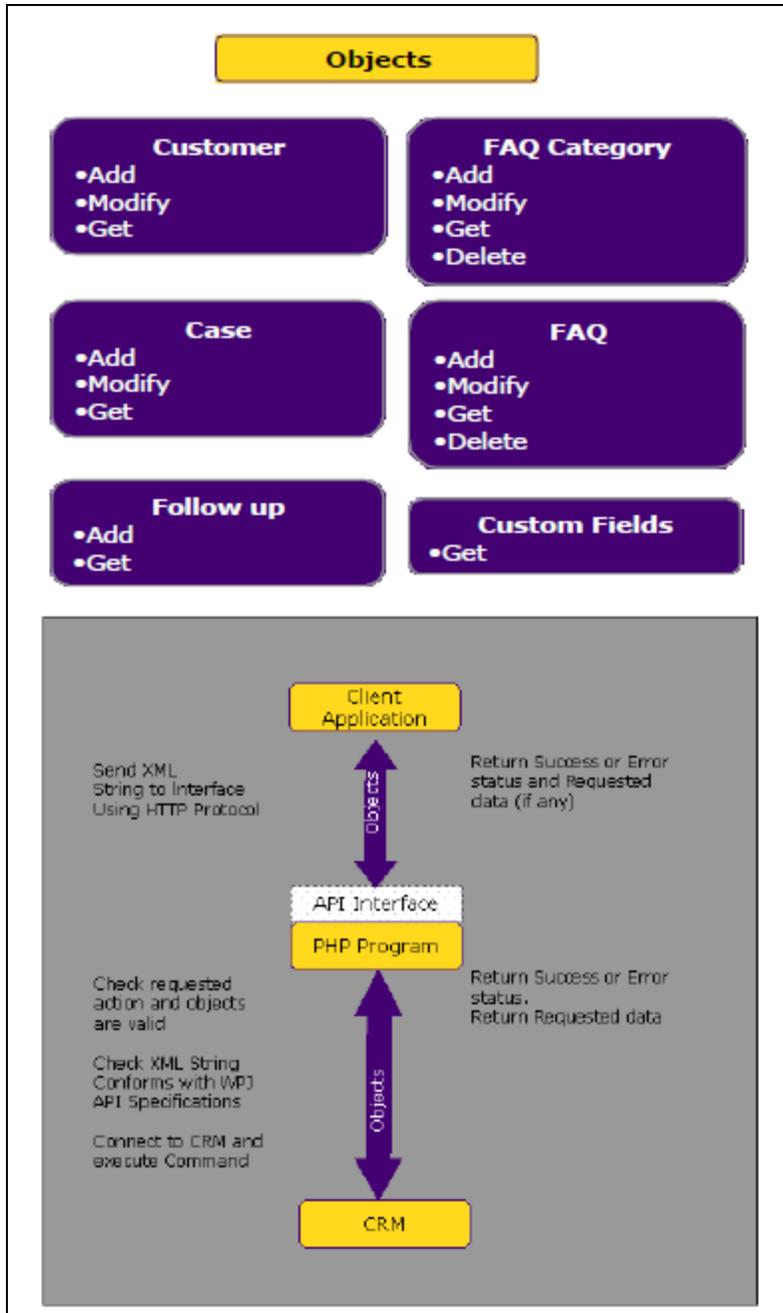
- Add
- Modify
- Get
- Delete
- List

All actions executed through CRM API update the Local CRM database through data insertion, modification, and retrieval.

The CRM API:

- Packages requests, and receives responses formatted as XML.
- Transmits and receives XML requests and responses via HTTP or HTTPS.  
XML-based CRM API actions must be made by an authenticated user login and password. For information about configuring CRM API access to the Local CRM, see the [Virtual Contact Center Configuration Manager user guide](#).

The following diagram outlines the object and CRM API application flow:



## Use CRM API

The 8x8 CRM API Interface accepts XML packets from external applications. The requests are sent via HTTP protocol. The interface expects a variable titled `xml_query` which contains the XML packet.

Once the API Interface receives a request, it processes the query as follows:

1. The API Interface checks for an `xml_query` variable.
2. The XML string is extracted from the variable.
3. The XML string is validated and parsed.
4. The data and command in the XML string are validated.
5. The action is executed.
6. An error or success status is returned.
7. Requested data, if any, is returned.

The access point to the interface varies based on the platform your tenant is hosted on:

- For p5/p6: <https://vcc-na1.8x8.com/WAPI/wapi.php>
- For p7/p8: <https://vcc-na2.8x8.com/WAPI/wapi.php>
- For p9/p10: <https://vcc-na3.8x8.com/WAPI/wapi.php>
- For p11/p12: <https://vcc-na4.8x8.com/WAPI/wapi.php>

For a full list of platform URLs, refer to our [Virtual Contact Center platform URL guide](#).

## Configure CRM API Access

Before using 8x8 Local CRM, you must enable and set it up in Configuration Manager to define the Integration Suite user and configure the Integration Suite's permissions.

### To configure CRM API access:

1. Log in to Configuration Manager.
2. Go to **Integration > CRM API**.

Configuration Manager [Acme]

8x8, Inc.

Integration » CRM API Info

CRM API CRM Triggers Screen Pop API Token

Enable CRM API Access

\* CRM API username: wapi

\* Password: ●●●●●●

\* Retype Password: ●●●●●●

Used Identity: John Smith

Access Rights	Customers	Cases	Followups	FAQ
Add	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Modify	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Get	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Save Cancel

3. Check the **Enable CRM API Access**.
4. Enter a **CRM API username** to access the API.



**Note:** Username length cannot exceed 10 characters.

5. Enter a **Password** to authenticate user access.



**Note:** Length of the password has to be between 3 and 15 characters.

6. Confirm your password by retyping.
7. For **Used Identity**, select an agent whose identity is used when accessing data through the CRM API.
8. For **Access Rights**, specify data access rights to individual components allowed through the use of CRM API.

## Incoming Requests

XML requests to the 8x8 CRM API must be submitted as a standard HTTP request on Port 80, and as HTTPS on Port 443.

The following example illustrates an HTML containing HTTP request to Virtual Contact Center XML API. In this example, `WAPI_COMMAND` specifies an XML request of the format `<COMMAND OBJECT="object_type" ACTION="action_type"> <TAG>param_value</TAG>`

```
<html>
  <body onload="document.xml.submit();">
    <form action=https://vcc-na4.8x8.com/WAPI/wapi.php method="post" name="xml">
<input type= "hidden" name="xml_query" value='WAPI_COMMNAD'>
    </form>
  </body>
</html>
```

The following example illustrates HTML containing HTTP request to Virtual Contact Center XML API.

```
<WAPI>
  <TENANT>tenant_name</TENANT>
  <USERNAME>wapi_username</USERNAME>
  <PASSWORD>wapi_password</PASSWORD>
  <COMMAND OBJECT="object_type" ACTION="action_type">
    <TAG>param_value</TAG>
  </COMMAND>
</WAPI>
```

All parameters included in the `WAPI_` command must contain a value.

## List of Parameters

The following table summarizes the parameters used in the example above.

Parameter	Summary
tenant_name	Name of the tenant as defined in the console management.
wapi_username	The login for the agent as defined in Configuration Manager.

Parameter	Summary
wapi_password	The password associated to the wapi_username.
object_type	Type of the object manipulated. The valid values include CUSTOMER, CASE, FOLLOWUP, FAQ, FAQCATEGORY, and CUSTOMFIELD.
action_type	The action to perform on manipulated object. This could be any of the following values, depending on the type of the object: ADD, MODIFY, GET, LIST, or DELETE.
param_value	Value given to the parameter defined by TAG.



**Note:** If a request contains one or more quote (") characters, they must be encoded as &rsquot;.

## Use Special Characters in CRM API Requests

To send a request to the 8x8 CRM API, you must use HTTP (or HTTPS) to transmit the request formatted in XML. The XML-formatted request may include strings and characters that have special meanings to HTTP.

When sending a request to the CRM API that contains special characters, you must:

- Use the XML CDATA construct to contain the reserved characters.
- Use both HTTP and XML escape encoding as a substitute for the reserved character.
- Use the reserved characters to perform their special functions in CRM API searches.

The following table lists common HTTP control characters and describes how to use those characters in CRM API requests.

Character	Replace With	CRM API Request Encoding
"	Blank	No special encoding required.
'	Blank	Do not include the apostrophe character in CRM API requests. The apostrophe character is the Local CRM database's string delimiter character.
~	Blank	No special encoding required.

Character	Replace With	CRM API Request Encoding
!	Blank	No special encoding required.
#	No	No special encoding required.
\$	Blank	No special encoding required.
%	Percent	The percent character is the CRM API wildcard character. For information about using Wildcard in CRM API requests, see <a href="#">tags that accept substring values</a> , and <a href="#">examples of Get Case operations</a> .
^	Blank	No special encoding required.
&	and	&amp;amp;
*	Asterisk	No special encoding required.
<	Blank	&amp;lt;
>	Blank	No special encoding required.

## Responses to Queries

The following example illustrates a response from the 8x8 Local CRM.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<WAPI>
<REPLY STATUS= "error_code" ERROR_STR= "error_string">
<TAG> result_value</TAG>
</REPLY>
</WAPI>
```

In the above example:

- `error_string` contains a message that describes the reason the query was unsuccessful.
- `result_value` contains the value returned for the parameter defined by TAG.

If the request was unsuccessful:

- If the request succeeds, `error_code` = 0 (zero) and `error_string` = "".
  - If the request fails, `error_code`="-1" and `error string`="".
- See [Error Cases](#) for a list of errors in a unsuccessful reply.

If the reply contains a list of ITEMS:

- Each item in the returned list is contained within the XML beginning and ending tags: <ITEM> and </ITEM>.
- Items for which no data is available are indicated by empty tags: <TAG></TAG> or <TAG/>.

## Examples of Queries and Responses

The following example illustrates HTML that queries the Local CRM for all customer records in which the first name is Bill.

```
<html>
<body onload="document.xml.submit();">
<form action=https://vcc-na4.8x8.com/WAPI/wapi.php
method="post" name="xml">
<input type="hidden" name="xml_query"
value='<wapi>
<tenant>AcmeJets</tenant>
<username>johnd</username>
<password>john123</password>
<command object="customer" action="get"> <firstname>Bill</firstname>
</command>
</wapi>'>
</form>
</body>
</html>
```

The following example illustrates Java code that queries the Local CRM for all customer records in which the first name is Bill.

```
package WapiAPI;
import java.net.* ;
import java.io.*;
/**
 * Title: Using Virtual Contact Center CRM API
 * Description:
 * Copyright: Copyright (c) 2001
 * Company: 8x8.com*
 * @version 1.0
 */
public class usePhp {
```

```

public usePhp() {
}
public static void main(String[] args) {
int c;
try {

/*connect to wapi.php*/
URL source = new URL("https://vcc-na4.8x8.com/WAPI/wapi.php") ;
URLConnection conn;
conn = source.openConnection() ;
conn.setDoOutput(true);
/*create xml_query string*/
String xmlString = "xml_query=";
xmlString += "<wapi>";
xmlString += "<tenant>mandrake</tenant><username>wapi</username>";
xmlString += "<password>wapi101</password><command object=\"customer\"
action=\"get\">";
xmlString += "<firstname>bill</firstname></command>";
xmlString += "</wapi>" ;

/*write data to connection*/
PrintWriter out = new PrintWriter(conn.getOutputStream());
out.println(xmlString);

```

The following example illustrates the response to the example queries for the Local CRM records in which the first name is Bill.

```

<?xml version="1. 0" encoding="ISO-8859-1"?>
<WAPI>
<REPLY STATUS=0 ERROR_STR="">
<ITEM>
<FIRSTNAME>Bill</FIRSTNAME>
<LASTNAME>Gates</LASTNAME>
<EMAIL>bill.gates@microsoft.com</EMAIL>
<CUSTOMERTYPE>NT User</CUSTOMERTYPE>
<COMPANY>Microsoft</COMPANY> <VOICE>123 456 7890</VOICE> <ALTERNATIVE>789 123
4560</ALTERNATIVE>
<FAX>456 123 7890</FAX>

```

```
<COMMENTS>Very important customer</COMMENTS>
<PASSWORD>bill123</PASSWORD>
<AUTOPASSWD>False</AUTOPASSWD>
<ADDR1STR1>1234 John Fitzgerald</ADDR1STR1>
<ADDR1STR2>Kennedy Boulevard</ADDR1STR2>
<ADDR1CITY>Los Angeles</ADDR1CITY>
<ADDR1STATE>CA</ADDR1STATE> <ADDR1ZIP>12345</ADDR1ZIP>
<ADDR1COUNTRY>United States</ADDR1COUNTRY>
<ADDR2STR1>456 Martin Luther</ADDR2STR1>
<ADDR2STR2>King Boulevard</ADDR2STR2>
<ADDR2CITY>San Francisco</ADDR2CITY>
<ADDR2 STATE>CA< /ADDR2 STATE> <ADDR2ZIP>67890</ADDR2ZIP>
<ADDR2COUNTRY>United States</ADDR2COUNTRY>
<ACCOUNTNUM>123456< /ACCOUNTNUM>
</ITEM>
<ITEM>
...
</ITEM>
</REPLY>
</WAPI>
```

## Manipulate Customer Objects

You can use the 8x8 CRM API to add, modify, and get customer data. This section outlines the actions that can be performed on a Customer object, and the parameters that are used in both the request and the reply.

You can perform the following actions on the Customer object:

- [Add Customer](#)
- [Modify Customer](#)
- [Get Customer Details](#)

### Add Customer

The `Add Customer` action allows a tenant to add a new customer in the 8x8 Local CRM. This corresponds to the `Create Contact` action in Agent Console.

- The `Customer` keyword is not case-sensitive.
- The `Add` keyword is not case-sensitive.

The following example illustrates an `Add Customer` request that includes custom fields.

```
'<WAPI>
<tenant>AcmeJets</tenant>
<username>wapi</username>
<password>wapi1</password>
<COMMAND OBJECT="Customer" ACTION="Add">
<FIRSTNAME>Bill</FIRSTNAME>
<LASTNAME>Gates</LASTNAME>
<EMAIL>b.gates@microsoft.com</EMAIL>
<COMPANY>Microsoft</COMPANY>
```

```

<VOICE>123 456 7890</VOICE>
<ALTERNATIVE>789 123 4560</ALTERNATIVE>
<FAX>456 123 7890</FAX>
<COMMENTS>Very important customer</COMMENTS>
<AUTOPASSWD>FALSE</AUTOPASSWD>
<PASSWORD>bill123</PASSWORD>
<ADDR1STR1>1234 John Fitzgerald</ADDR1STR1>
<ADDR1STR2>Kennedy Boulevard</ADDR1STR2>
<ADDR1CITY>Los Angeles</ADDR1CITY>
<ADDR1STATE>CA</ADDR1STATE>
<ADDR1ZIP>12345</ADDR1ZIP>
<ADDR1COUNTRY>United States</ADDR1COUNTRY>
<ADDR2STR1>456 Martin Luther</ADDR2STR1>
<ADDR2STR2>King Boulevard</ADDR2STR2>
<ADDR2CITY>San Francisco</ADDR2CITY>
<ADDR2STATE>CA</ADDR2STATE>
<ADDR2ZIP>67890</ADDR2ZIP>
<ADDR2COUNTRY>United States</ADDR2COUNTRY>
<CUSTOMERTYPE>Default</CUSTOMERTYPE>
<CF01_PICKLIST.NAME>value1</CF01_PICKLIST.NAME> <CF02.NAME>value2</CF02.NAME>
</COMMAND>
</WAPI>'

```

In the example results, `CF01_PICKLIST.NAME` and `CF02.NAME` identify previously-defined custom Local CRM fields.

Adding multiple customers or cases requires writing a program to enable the same. If you need to add multiple customers, modify the script to include multiple customer records.

## Mandatory and Optional Parameters

The following table summarizes the mandatory and optional Add parameters for the `Customer` object.

Parameter	Mandatory/Optional	Format
FIRSTNAME	M	Maximum characters: 30
LASTNAME	M	Maximum characters: 30
EMAIL	O	<ul style="list-style-type: none"> <li>■ Maximum characters: 55</li> <li>■ A-Z a-z - 0-9 _ @</li> </ul>

Parameter	Mandatory/Optional	Format
		 <b>Note:</b> Email address has to be unique for each customer instance.
COMPANY	O	<ul style="list-style-type: none"> <li>■ Maximum characters: 50</li> <li>■ Invalid: ~, &lt;, &gt;, whitespace-only</li> </ul>
VOICE	O	<ul style="list-style-type: none"> <li>■ Maximum characters: 38</li> <li>■ Valid: + ( ) / . - _ 0-9 space</li> <li>■ Invalid: whitespace-only, non-numeric characters</li> </ul>
ALTERNATIVE	O	<ul style="list-style-type: none"> <li>■ Maximum characters: 38</li> <li>■ Valid: + ( ) / . - _ 0-9 space</li> <li>■ Invalid: whitespace-only, non-numeric characters</li> </ul>
FAX	O	<ul style="list-style-type: none"> <li>■ Maximum characters: 38</li> <li>■ Valid: + ( ) / . - _ 0-9 space</li> <li>■ Invalid: whitespace-only, non-numeric characters</li> </ul>
COMMENTS	O	Maximum characters: 1024
PASSWORD	O	<ul style="list-style-type: none"> <li>■ Minimum characters: 3</li> <li>■ Maximum characters: 15</li> <li>■ A-Z, a-z, _, 0-9</li> </ul>
AUTOPASSWD	M	<ul style="list-style-type: none"> <li>■ Possible values: <code>False</code> or <code>True</code> (case-insensitive).</li> <li>■ If it is <code>False</code> the <code>PASSWORD</code> parameter must be provided.</li> <li>■ If it is <code>True</code> the <code>PASSWORD</code> parameter is ignored.</li> </ul>
ADDR1 STR1	O	Any character, maximum 50
ADDR1 STR2	O	Any character, maximum 50
ADDR1CITY	O	Any character, maximum 40
ADDR1 STATE	O	Any character, maximum 30
ADDR1ZIP	O	Any character, maximum 15

Parameter	Mandatory/Optional	Format
ADDR1COUNTRY	O	See the <a href="#">List of Attribute Values</a> to determine the list of countries. If no country is provided, United States is used.
ADDR2STR1	O	Any character, maximum 50
ADDR2STR2	O	Any character, maximum 50
ADDR2CITY	O	Any character, maximum 40
ADDR2 STATE	O	Any character, maximum 30
ADDR2ZIP	O	Any character, maximum 15
ADDR2COUNTRY	O	See the <a href="#">List of Attribute Values</a> to determine the list of countries. If no country is provided, the United States is used.
CUSTOMERTYPE	O	The contact type must be one valid type (not deactivated) as defined in Configuration Manager. The list of possible values is defined by the administrator of the tenant.
CF01_ PICKLIST.NAME	O	Custom fields and their parameter names are added by the administrator of the tenant. The values must be one valid type (not deactivated) as defined in Configuration Manager. The list of possible values is defined by the administrator of the tenant.
CF02.NAME	O	Custom fields and their parameter names are added by the administrator of the tenant.  The values must be one valid type (not deactivated) as defined in Configuration Manager. The values passed must be in a text or number format.

## Reply from Local CRM to Add Customer

The following example illustrates a reply to an `Add Customer` request in the 8x8 Local CRM. If a customer record is added successfully, a response indicating the successful addition of a new customer along with the assigned account number is shown.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<WAPI>
<REPLY ERROR_CODE="0" ERROR_STR="" STATUS="0">
```

```
<ITEM>
<ACCOUNTNUM>1453</ACCOUNTNUM>
</ITEM>
</REPLY>
</WAPI>
```

See [Error Cases](#) for a list of errors in an unsuccessful reply.

## Modify Customer

The `Modify Customer` action modifies an existing 8x8 Local CRM customer record. This operation corresponds to the `Edit Contact` action in Agent Console.

- The `Modify` keyword is not case-sensitive.
- Please note that the user may only send mandatory and modified parameters.
- Any parameter not supplied in a `Modify` request retains its previous value.
- Account Number is a mandatory field for modifying a customer record.

The following example illustrates a `Modify` customer operation that includes custom fields. The request passes an updated email address.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME> <PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Customer" ACTION="Modify">
<ACCOUNTNUM>1453</ACCOUNTNUM>
<FIRSTNAME>Bill</FIRSTNAME> <LASTNAME>Gates</LASTNAME>
<EMAIL>MSGuru@microsoft.com</EMAIL>
</COMMAND>
</WAPI>'
```

## List of Mandatory and Optional Parameters

The following table summarizes the mandatory and optional `Modify` parameters for the `Customer` object.

Parameter	Mandatory/Optional	Format
ACCOUNTNUM	M	Customer account number to be modified as returned by <code>Get</code> request.
FIRSTNAME	O	Maximum characters: 30

Parameter	Mandatory/Optional	Format
LASTNAME	O	Maximum characters: 30
EMAIL	O	<ul style="list-style-type: none"> <li>■ Maximum characters: 55</li> <li>■ (A-Z a-z - 0-9 _ @)</li> </ul>
COMPANY	O	<ul style="list-style-type: none"> <li>■ Maximum characters: 50</li> <li>■ Invalid: ~, &lt;, &gt;, whitespace-only</li> </ul>
VOICE	O	<ul style="list-style-type: none"> <li>■ Maximum characters: 38</li> <li>■ + ( ) / . - _ 0-9 space</li> <li>■ Invalid: whitespace-only, non-numeric characters</li> </ul>
ALTERNATIVE	O	<ul style="list-style-type: none"> <li>■ Maximum characters: 38</li> <li>■ + ( ) / . - _ 0-9 space</li> <li>■ Invalid: whitespace-only, non-numeric characters</li> </ul>
FAX	O	Valid- + ( ) / . - _ 0-9 space
ACCOUNTNUM	M	Customer account number to be modified as returned by GET request.
FIRSTNAME	O	Maximum characters: 30
LASTNAME	O	Maximum characters: 30
EMAIL	O	<ul style="list-style-type: none"> <li>■ Maximum characters: 55</li> <li>■ (A-Z a-z - 0-9 _ @)</li> </ul>
COMPANY	O	<ul style="list-style-type: none"> <li>■ Maximum characters: 50</li> <li>■ Invalid: ~, &lt;, &gt;, whitespace-only</li> </ul>
VOICE	O	<ul style="list-style-type: none"> <li>■ Maximum characters: 38</li> <li>■ + ( ) / . - _ 0-9 space</li> <li>■ Invalid: whitespace-only, non-numeric characters</li> </ul>
ALTERNATIVE	O	<ul style="list-style-type: none"> <li>■ Maximum characters: 38</li> <li>■ + ( ) / . - _ 0-9 space</li> <li>■ Invalid: whitespace-only, non-numeric characters</li> </ul>
FAX	O	<ul style="list-style-type: none"> <li>■ Maximum characters: 38</li> <li>■ Valid- + ( ) / . - _ 0-9 space</li> </ul>

Parameter	Mandatory/Optional	Format
		<ul style="list-style-type: none"> <li>Invalid: whitespace-only, non-numeric characters</li> </ul>
COMMENTS	O	Maximum characters: 1024
PASSWORD	O	<ul style="list-style-type: none"> <li>Minimum characters: 3</li> <li>Maximum characters: 15</li> <li>A-Z, a-z, _, 0-9</li> </ul>
AUTOPASSWD	O	<ul style="list-style-type: none"> <li>Possible values: <code>False</code> or <code>True</code> (case insensitive).</li> <li>If the value is <code>false</code>, the <code>PASSWORD</code> parameter must be provided.</li> <li>If the value is <code>true</code>, the <code>PASSWORD</code> parameter is ignored.</li> </ul>
ADDR1 STR1	O	Any character, maximum: 50
ADDR1 STR2	O	Any character, maximum: 50
ADDR1CITY	O	Any character, maximum: 40
ADDR1 STATE	O	Any character, maximum: 30
ADDR1ZIP	O	Any character, maximum: 15
ADDR1COUNTRY	O	See the <a href="#">List of Attribute Values</a> to determine the list of countries. If no country is provided, the United States is used.
ADDR2STR1	O	Any character, maximum: 50
ADDR2STR2	O	Any character, maximum: 50
ADDR2CITY	O	Any character, maximum: 40
ADDR2 STATE	O	Any character, maximum: 30
ADDR2ZIP	O	Any character, maximum: 15
ADDR2COUNTRY	O	See the <a href="#">List of Attribute Values</a> to determine the list of countries. If no country is provided, the United States is used.
CUSTOMERTYPE	O	The contact type must be one valid type (not deactivated) as defined in Configuration Manager. The list of possible values is defined by the administrator of the tenant.
CF01_	O	Custom fields and their parameter names are added by the administrator

Parameter	Mandatory/Optional	Format
PICKLIST.NAME		of the tenant. The values must be one valid type as defined in Configuration Manager. The list of possible values is defined by the administrator of the tenant.
CF02.NAME	O	Custom fields and their parameter names are added by the administrator of the tenant. The values must be one valid type as defined in Configuration Manager. The values passed should be in a text or number format.

## Reply from Local CRM to Modify Customer

The following example illustrates a reply to a `Modify Customer` request in the 8x8 Local CRM.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<WAPI>
<REPLY ERROR_CODESTATUS="0" ERROR_STR="">
<ACCOUNTNUM>1453</ACCOUNTNUM>
</REPLY>
</WAPI>
```

See [Error Cases](#) for a list of errors to an unsuccessful reply.

## Get Customer Details

The `Get Customer Details` action enables you to retrieve one or more customer details from the Local CRM. This corresponds to the `Search` and `View` links in Agent Console.

- The `Get` keyword is not case-sensitive.
- Command object only needs to be passed once, even when searching for multiple parameters.
- The parameter values being searched are case-sensitive.
- A `Get` request can supply any combination of parameters to retrieve data.

## Tags to Pair for Range Queries

- `FromCreateDate` / `ToCreateDate`
- `FromLastModDate` / `ToLastModDate`

## Examples of Get Customer Operations

The following example illustrates a request to get customer details such as company name.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME> <PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Customer" ACTION="GET">
<COMPANY>microsoft</COMPANY>
</COMMAND>
</WAPI>'
```

To retrieve customer records where company name includes a particular letter, use % to specify a single wildcard character.

The following example illustrates a customer details search. It uses a single wildcard character to return all customer records where the company name begins with m.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Customer" ACTION="GET"> <COMPANY>m%</COMPANY>
</COMMAND>
</WAPI>'
```

The following example illustrates a customer details search for a particular last name.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Customer" ACTION="GET">
<LASTNAME>gates</LASTNAME>
</COMMAND>
</WAPI>'
```

The following example illustrates a customer details search. It uses a single wildcard character to return all customer records where the last name begins with g.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
```

```
<COMMAND OBJECT="Customer" ACTION="GET"> <LASTNAME>g%</LASTNAME>
</COMMAND>
</WAPI> '
```

The following example illustrates a customer details search for a specific email address.

```
' <WAPI>
<TENANT>CompanyX</TENANT> <USERNAME>johnd</USERNAME> <PASSWORD>john12
3</PASSWORD>
<COMMAND OBJECT="Customer" ACTION="GET"> <EMAIL>bill@microsoft.com</EMAIL>
</COMMAND>
</WAPI> '
```

The following example illustrates a customer details search. It uses a single wildcard character to return all customer records where the email begins with b.

```
' <WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Customer" ACTION="GET"> <EMAIL>b%</EMAIL>
</COMMAND>
</WAPI> '
```

The following example illustrates a customer details search for a specific contact type. The contact type must be one valid (not deactivated) type defined in Configuration Manager.

```
' <WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Customer" ACTION="GET">
<CUSTOMERTYPE>NT User</CUSTOMERTYPE>
</COMMAND>
</WAPI> '
```

The following example illustrates a customer detail search for a specific account number.

```
' <WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Customer" ACTION="GET"> <ACCOUNTNUM>1453</ACCOUNTNUM>
</COMMAND>
```

```
</WAPI>'
```

The following example illustrates a customer detail search. It uses a wildcard construct to return all customer records that contain 45.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Customer" ACTION="GET"> <ACCOUNTNUM>%45%</ACCOUNTNUM>
</COMMAND>
</WAPI>'
```

A generic search of customer records returns:

- A range of available account numbers.
- The total number of available records.

The following example illustrates the general form of the XML used to perform customer record searches.

```
...
<FIRSTID>ID of first record available</FIRSTID>
<LASTID>ID of last record available</LASTID>
<TOTAL>total number of records</TOTAL>
...
```

The following example illustrates a search for all customer records in a tenant. The response includes information on the account number, range of available records, and the total number of customer records.

```
'<WAPI>
TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Customer" ACTION="GET">
</COMMAND>
</WAPI>'
```

The following example illustrates the Local CRM response to the customer detail search.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<WAPI>
<REPLY ERROR_CODE="0" ERROR_STR="" STATUS="0" >
<ITEM>
<FIRSTID>10000000</FIRSTID>
<LASTID>10000083</LASTID>
```

```
<TOTAL>81</TOTAL>
</ITEM>
</REPLY>
</WAPI>
```

See **Error Cases** for a list of errors to a unsuccessful reply.

In the Local CRM, the contact center administrator uses Configuration Manager to define custom fields and parameters.

The following example illustrates a customer details search for the custom picklist name CF01.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Customer" ACTION="GET">
<CF01_PICKLIST.NAME>Value</CF01_PICKLIST.NAME>
</COMMAND>
</WAPI>'
```

The following example illustrates a customer details search for the custom field name with a specific value of CF02.

```
'<WAPI>
<TENANT>AcmeJets</TENANT> <USERNAME>johnd</USERNAME> <PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Customer" ACTION="GET">
<CF02.NAME>Value</CF02.NAME>
</COMMAND>
</WAPI>'
```

The following example illustrates a search for customers created on a specific day.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="CUSTOMER" ACTION="GET">
<FROMCREATEDATE>06202012</FROMCREATEDATE>
<TOCREATEDATE>06252012</TOCREATEDATE>
</COMMAND>
</WAPI>'
```

The following example illustrates a search for customers last modified on a specific day.

```
'<WAPI>
```

```

<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123<PASSWORD>
<COMMAND OBJECT="CUSTOMER" ACTION="GET">
<FROMLASTMODDATE>06202012</FROMLASTMODDATE>
<TOLASTMODDATE>06252012</TOLASTMODDATE>
</COMMAND>
</WAPI>'

```

## List of Parameters

The following table summarizes the `Get` parameters for the `Customer` object.

Parameter	Format
FIRSTNAME	Maximum characters: 30
LASTNAME	Maximum characters: 30
EMAIL	<ul style="list-style-type: none"> <li>■ Maximum characters: 55</li> <li>■ (A-Z a-z - 0-9 _ @)</li> </ul>
VOICE	<ul style="list-style-type: none"> <li>■ Maximum characters: 38</li> <li>■ + ( ) / . - _ 0-9 space</li> <li>■ Invalid: whitespace-only, non-numeric characters</li> </ul>
COMPANY	<ul style="list-style-type: none"> <li>■ Maximum characters: 50</li> <li>■ Characters</li> <li>■ Invalid: ~, &lt;, &gt;, whitespace-only</li> </ul>
ACCOUNTNUM	Account number of the customer who is related to the case. This account number is returned after a <code>Get</code> action on a <code>Customer</code> object.
CUSTOMERTYPE	The contact type must be one valid type (not deactivated) as defined in Configuration Manager. The list of possible values is defined by the administrator of the tenant.
CF01_ PICKLIST.NAME	<p>Custom fields and their parameter names are added by the administrator of the tenant.</p> <p>The values must be one valid type (not deactivated) as defined in Configuration Manager. The list of possible values is defined by the administrator of the tenant.</p>
CF02.NAME	<p>Custom fields and their parameter names are added by the administrator of the tenant.</p> <p>The values must be one valid type (not deactivated) as defined in Configuration Manager.</p>

Parameter	Format
	The list of possible values is defined by the administrator of the tenant.
FROMCREATEDATE	The format for this attribute is MMDDYYYY, where MM is the month, DD the day, and YYYY the year.
TOCREATEDATE	The format for this attribute is MMDDYYYY, where MM is the month, DD the day, and YYYY the year.
FROMLASTMODDATE	The format for this attribute is MMDDYYYY, where MM is the month, DD the day, and YYYY the year.
TOLASTMODDATE	The format for this attribute is MMDDYYYY, where MM is the month, DD the day, and YYYY the year.

See [Error Cases](#) for a list of errors to a unsuccessful reply.

## Reply from Local CRM to Get Customer Details

The following example illustrates a reply to a `Get Customer Details` request in the 8x8 Local CRM. In this example, `CF01_PICKLIST.NAME` and `CF02.NAME` are custom fields.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<WAPI>
<REPLY STATUS=0 ERROR_STR=""> <ITEM>
<FIRSTNAME>Bill</FIRSTNAME> <LASTNAME>Gates</LASTNAME>
<EMAIL>bill.gates@microsoft.com</EMAIL>
<COMPANY>Microsoft</COMPANY> <VOICE>123 456 7890</VOICE> <ALTERNATIVE>789 123
4560</ALTERNATIVE>
<FAX>456 123 7890</FAX>
<COMMENTS>Very important customer</COMMENTS> <PASSWORD>bill123</PASSWORD>
<ADDR1STR1>1234 John Fitzgerald</ADDR1STR1> <ADDR1STR2>Kennedy
Boulevard</ADDR1STR2>
<ADDR1CITY>Los Angeles</ADDR1CITY>
<ADDR1 STATE>CA</ADDR1 STATE> <ADDR1ZIP>12345</ADDR1ZIP> <ADDR1COUNTRY>United
States</ADDR1COUNTRY>
<ADDR2STR1>4 56 Martin Luther</ADDR2STR1>
<ADDR2STR2>King Boulevard</ADDR2STR2>
<ADDR2CITY>San Francisco</ADDR2CITY>
<ADDR2 STATE>CA< /ADDR2 STATE> <ADDR2ZIP>678 90</ADDR2ZIP> <ADDR2COUNTRY>United
```

```

States</ADDR2COUNTRY>
<ACCOUNTNUM> 123456< /ACCOUNTNUM> <CUSTOMERTYPE>NT user< /CUSTOMERTYPE>
<CF01_PICKLIST.NAME>value1< /CF01_PICKLIST.NAME>
<CF02.NAME>value2</CF02.NAME> ...
</ITEM>
<ITEM>
...
</ITEM>
</REPLY>
</WAPI>

```

See [Error Cases](#) for a list of errors to an unsuccessful reply.

## Delete Customer

The `Delete Customer` action deletes an existing Local CRM customer record. This operation corresponds to the `Delete Contact` action in Agent Console.

- The `Delete` keyword is not case-sensitive.
- Deleting a customer causes a cascade effect; all cases, tasks, and follow-ups related to a given customer are deleted as well.

## Example of Delete Customer Operation

The following example illustrates a request to delete a customer.

```

`<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123<PASSWORD>
<COMMAND OBJECT="CUSTOMER" ACTION="DELETE">
<ACCOUNTNUM>10000019<ACCOUNTNUM>
</WAPI>`

```

## List of Mandatory and Optional Parameters

The following table summarizes the mandatory and optional `Delete` parameters for the `Customer` object.

Parameter	Mandatory/Optional	Format
ACCOUNTNUM	M	The account number of the customer

## Reply from Local CRM to Delete Customer

The following example illustrates a reply to a Delete Customer request in the 8x8 Local CRM.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<WAPI>
<REPLY STATUS="0" ERROR_STR="" ERROR_CODE="0"/>
</WAPI>
```

See [Error Cases](#) for a list of errors to an unsuccessful reply.

## Manipulate Case Objects

You can use the 8x8 CRM API to add, modify, and get case data. The section outlines all the possible actions that can be performed on a Case object, and the parameters that are used in both the request and the reply.

You can perform the following actions on the Case object:

- **Add Case**
- **Modify Case**
- **Get Case**

### Add Case

The `Add Case` action enables you to add a new case to the 8x8 Local CRM. This action corresponds to the `Create Case` action in Agent Console.

- The `Case` keyword is not case-sensitive.
- The `Add` keyword is not case-sensitive.
- `ACCOUNTNUM`, `SUBJECT`, and `DESCRIPTION` are mandatory fields to add a case.

The following example illustrates an `Add Case` operation. In this example, `CF01_PICKLIST.NAME` and `CF02.NAME` are custom fields.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Case" ACTION="Add">
<ACCOUNTNUM>10000019</ACCOUNTNUM>
<SUBJECT>My printer doesn't work</SUBJECT>
```

```

<DESCRIPTION>This customer downloaded the new driver
XXXX v5.4 of the driver, installed it and since then the
printer doesn't work.</DESCRIPTION>
<STATUS>Unassigned</STATUS>
<PRIORITY>Medium</PRIORITY>
<SEVERITY>Information</SEVERITY>
<PROJECT>default</PROJECT>
<CATEGORY>Default</CATEGORY>
<VISIBILITY>Internal</VISIBILITY>
<MEDIATYPE>Phone</MEDIATYPE>
<CF01_PICKLIST.NAME>value1</CF01_PICKLIST.NAME>
<CF02.NAME>value2</CF02.NAME>
</COMMAND>
</WAPI>'

```

## List of Mandatory and Optional Parameters

The following table summarizes the mandatory and optional Add parameters for the Case object.

Parameter	Mandatory/Optional	Format
ACCOUNTNUM	M	Account number of the customer who is related to the case. This account number is returned after a <code>Get</code> action on a <code>Customer</code> object.
SUBJECT	M	<ul style="list-style-type: none"> <li>■ Any character</li> <li>■ Maximum characters: 255</li> <li>■ Invalid: whitespace-only</li> </ul>
DESCRIPTION	M	<ul style="list-style-type: none"> <li>■ Any character</li> <li>■ Maximum characters: 2 MB</li> <li>■ Invalid: whitespace-only</li> </ul>
STATUS	O	<p>The status must be one valid status (not deactivated) as defined in Configuration Manager.</p> <p>The <code>Get</code> action on a custom field object can be used to determine the list of possible statuses.</p> <p>If no status is provided, the default value is unassigned. If status is used, the <code>ASSIGNEDTO</code> parameter must also be defined.</p>

Parameter	Mandatory/Optional	Format
PRIORITY	<input type="radio"/>	<p>The priority must be one valid priority (not deactivated) as defined in Configuration Manager. The list of possible values is defined by the administrator of the tenant.</p> <p>The <code>Get</code> action on a custom field object can be used to determine the list of possible priorities. If no priority is provided, the default is used.</p>
SEVERITY	<input type="radio"/>	<p>The severity must be one valid severity (not deactivated) as defined in Configuration Manager. The list of possible values is defined by the administrator of the tenant.</p> <p>The <code>Get</code> action on a custom field object can be used to determine the list of possible severities.</p> <p>If no severity is provided, the default is used.</p>
PROJECT	<input type="radio"/>	<p>The project must be a valid project (not deactivated) as defined in Configuration Manager. The list of possible values is defined by the administrator of the tenant.</p> <p>The <code>Get</code> action on a custom field object can be used to determine the list of possible projects.</p> <p>If no project is provided, the default is used.</p>
CATEGORY	<input type="radio"/>	<p>The category must be one valid category (not deactivated) as defined in Configuration Manager. The list of possible values is defined by the administrator of the tenant.</p> <p>The <code>Get</code> action on a custom field object can be used to determine the list of possible categories.</p> <p>If no category is provided, the default is used.</p>
VISIBILITY	<input type="radio"/>	<p>The visibility must be one valid category (not deactivated) as defined in Configuration Manager.</p> <p>Possible values are <code>Internal</code> or <code>External</code>. If no visibility is provided, the internal value is used.</p>
MEDIATYPE	<input type="radio"/>	<p>The media type must be valid (not deactivated) as defined in Configuration Manager.</p> <p>Possible values are: <code>Phone</code>, <code>Chat</code>, <code>Email</code>, or <code>None</code>.</p>

Parameter	Mandatory/Optional	Format
ASSIGNEDTO	O	The ASSIGNEDTO must be equal to a valid agent or group user ID. If ASSIGNEDTO is used, the STATUS parameter must also be defined.  To search for cases assigned to the deleted agents, deleted groups, or unassigned agents, the values are: Deleted Agents, Deleted Groups, or None.
CF01_PICKLIST.NAME	O	Custom fields and their parameter names are added by the administrator of the tenant. The values must be valid (not deactivated) as defined in Configuration Manager. The list of possible values is defined by the administrator of the tenant.
CF02.NAME	O	Custom fields and their parameter names are added by the administrator of the tenant. The values must be valid (not deactivated) as defined in Configuration Manager. The values passed must be in a text or number format.

See [Error Cases](#) for a list of errors to a unsuccessful reply.

## Reply from Local CRM to Add Case

The following example illustrates a reply to an Add Case request in the 8x8 Local CRM.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<WAPI>
<REPLY STATUS="0" ERROR _STR="">
<CASENUM>67890</CASENUM>
</REPLY>
</WAPI>
```

See [Error Cases](#) for a list of errors to an unsuccessful reply.

## Modify Case

The `Modify Case` action modifies an existing 8x8 Local CRM case record. This corresponds to the `Edit Case` action in Agent Console.

- The `Modify` keyword is not case-sensitive.
- The user may only send mandatory and modified parameters.

- Any parameter not supplied in the Modify request retains the previous value.
- Any optional parameter not supplied in Modify request is set to null in the Local CRM.
- The Case Number is mandatory to modify a case.

The Modify Case operation generates an automated follow-up containing a description of the modification consistent with an Edit Case action taken through Agent Console.

The following example illustrates a `Modify` case request that includes custom fields. In this example, `CF01_PICKLIST.NAME` and `CF02.NAME` identify previously identified custom Local CRM fields.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Case" ACTION="Modify">
<CASENUM>67890</CASENUM>
<STATUS>Open</STATUS>
<PRIORITY>High</PRIORITY>
<ASSIGNEDTO>jsmith</ASSIGNEDTO>
</COMMAND>
<WAPI>'
```

## List of Mandatory and Optional Parameters

The following table summarizes the mandatory and optional `Modify` parameters for the `Case` object.

Parameter	Mandatory/Optional	Format
CASENUM	M	This case number is returned after a <code>Get</code> action on a <code>Case</code> object.
SUBJECT	O	<ul style="list-style-type: none"> <li>■ Any character</li> <li>■ Maximum characters: 255</li> <li>■ Invalid: whitespace-only</li> </ul>
DESCRIPTION	O	<ul style="list-style-type: none"> <li>■ Any character</li> <li>■ Maximum characters: 2 MB</li> <li>■ Invalid: whitespace-only</li> </ul>
STATUS	O	<p>The status must be valid (not deactivated) as defined in Configuration Manager.</p> <p>The <code>Get</code> action on a custom field object can be used to determine the list of</p>

Parameter	Mandatory/Optional	Format
		<p>possible statuses.</p> <p>If no status is provided, the default value is <code>unassigned</code>. If a status is assigned, the <code>ASSIGNEDTO</code> parameter must also be defined.</p>
PRIORITY	O	<p>The priority must be valid (not deactivated) as defined in Configuration Manager. The list of possible values is defined by the administrator of the tenant.</p> <p>The <code>Get</code> action on a custom field object can be used to determine the list of possible priorities. If no priority is provided, the default is used.</p>
SEVERITY	O	<p>The severity must be valid (not deactivated) as defined in Configuration Manager. The list of possible values is defined by the administrator of the tenant.</p> <p>The <code>Get</code> action on a custom field object can be used to determine the list of possible severities. If no severity is provided, the default is used.</p>
CATEGORY	O	<p>The category must be valid (not deactivated) as defined in Configuration Manager. The list of possible values is defined by the administrator of the tenant.</p> <p>The <code>Get</code> action on a custom field object can be used to determine the list of possible categories. If no category is provided, the default is used.</p>
VISIBILITY	O	<p>The visibility must be valid (not deactivated) as defined in Configuration Manager.</p> <p>Possible values are <code>Internal</code> or <code>External</code>. If no visibility is provided, the <code>Internal</code> value is used.</p>
MEDIATYPE	O	<p>The media type must be valid (not deactivated) as defined in Configuration Manager.</p> <p>Possible values are: <code>Phone</code>, <code>Chat</code>, <code>Email</code>, or <code>None</code>.</p>
ASSIGNEDTO	O	<p>The <code>ASSIGNEDTO</code> must be equal to an agent or group user ID. If <code>ASSIGNEDTO</code> is used, the <code>STATUS</code> parameter must also be defined.</p>

See [Error Cases](#) for a list of errors to a unsuccessful reply.

## Reply from Local CRM to Modify Case

The following example illustrates a reply to a `Modify Case` request in the 8x8 Local CRM.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<WAPI>
<REPLY ERROR_CODE="0" ERROR_STR="" STATUS="0">
<CASENUM>67890</CASENUM>
</REPLY>
</WAPI>
```

See [Error Cases](#) for a list of errors to an unsuccessful reply.

## Get Case

The `Get Case` action enables a tenant to get a list of existing cases from the 8x8 Local CRM that matches the supplied parameter values. This action corresponds to the Search, View, and List actions in Agent Console.

- The `Get` keyword is not case-sensitive.
- Command object only needs to be passed once, even when searching for multiple parameters.
- Parameter values being searched are case-sensitive.
- This request returns entire descriptions for each case that matches the supplied parameter values including the customer account number.

The query parameters on a `Get` request can vary in the following ways:

- Query on a substring
- Pass multiple parameters to generate an `OR` query
- Pass multiple parameters to generate a `range` query

## Tags that Accept Substring Values

- Last Name
- Subject
- Company

The parameters values being searched as a substring require the use of `%`. For example, to search for the word printer, the search string `%printer%`.

## Tags that Accept Multiple Parameters

- CreatedBy
- AssignedTo
- Status
- Category
- Priority
- Severity
- Project

## Tags that Can be Paired for Range Queries

- FromCreateDate / ToCreateDate
- FromLastModDate / ToLastModDate

## Tags that Can be paired for Range Queries and Creation Dates

- FromDate
- ToDate

To retrieve cases created on a particular date, `FromDate` must correspond to the first date desired and `TODATE` must correspond to the day after.

For example, a `FROMDATE` of 12072001 to a `TODATE` of 12082001 retrieves cases created December 7<sup>th</sup>, 2001.

## Examples of Get Case Operations

The following example searches for a case given a case number.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Case" ACTION="GET"> <CASENUM>67890</CASENUM>
</COMMAND>
</WAPI>'
```

The following example searches for a case where `STATUS` is set to `Unassigned`.

The case status must match the values defined in Configuration Manager. The `GET` action on a custom field object can be used to determine the list of possible statuses.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Case" ACTION="GET"> <STATUS>Unassigned</STATUS>
<STATUS>Open</STATUS>
</COMMAND>
</WAPI>'
```

The following example illustrates a case search by creation date.

Only one occurrence of the `FROMDATE` and `TODATE` parameters can be used in one `GET` request.

To retrieve cases created on a particular date, `FROMDATE` must correspond to the first date desired and the `TODATE` corresponds to the day after. For example, a `FROMDATE` of 12072001 to a `TODATE` of 12082001 retrieves cases created on December 7<sup>th</sup>, 2001.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Case" ACTION="GET">
<FROMDATE>01112010</FROMDATE>
<TODATE>02112001</TODATE>
</COMMAND>
</WAPI>'
```

The following example illustrates a case search by category.

You can retrieve cases that match multiple category types. The categories must be active and match the values defined in Configuration Manager. The `GET` action on a custom field object can be used to determine the list of possible categories.

```
'<WAPI>
<TENANT>AcmeJets</TENANT> <COMMAND OBJECT="Case" ACTION="GET">
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Case" ACTION="GET">
<CATEGORY>Gold</CATEGORY> <CATEGORY>Platinum</CATEGORY>
</COMMAND>
</WAPI>'
```

The following example fetches cases assigned to specific agents.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Case" ACTION="GET">
<ASSIGNEDTO>jsmith</ASSIGNEDTO>
<ASSIGNEDTO>rshasta</ASSIGNEDTO>
<ASSIGNEDTO>sbaker</ASSIGNEDTO>
</COMMAND>
</WAPI>'
```

The following example fetches cases created by specific agents.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Case" ACTION="GET">
<CREATEDBY>jsmith</CREATEDBY>
<CREATEDBY>jdoe</CREATEDBY>
</COMMAND>
</WAPI>'
```

To retrieve cases where the subject includes a particular character string, enclose the string with %.

Only one occurrence of this tag can be used in one GET request. The parameters values being searched as a substring require the use of %. For example, to search for the word printer, the value must be set as: %printer%.

The following example illustrates a search for cases where the SUBJECT line contains the substring printer.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Case" ACTION="GET">
<SUBJECT>%printer%</SUBJECT>
</COMMAND>
</WAPI>'
```

The following example illustrates a GET search for cases where SUBJECT begins with p, using a wildcard character.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
```

```
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Case" ACTION="GET">
<SUBJECT>p %</SUBJECT>
</COMMAND>
</WAPI>'
```

The following example illustrates a GET search for cases attributed to a company. The search using wildcard characters returns all records where COMPANY contains the substring Microsoft.

Only one occurrence of this tag can be used in one GET request. The parameters values being searched as a substring require the use of %. For example, to search for the word Microsoft, the value must be set as: %Microsoft%.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Case" ACTION="GET">
<COMPANY>%Microsoft%</COMPANY>
</COMMAND>
</WAPI>'
```

The following example illustrates a GET case search using wildcard characters to return all records where COMPANY begins with m.

Only one occurrence of this tag can be used in one GET request. The parameters values being searched as a substring require the use of %. For example, to search for a portion of a parameter value such as Microsoft in Microsoft Corporation, the value must be set as: %Microsoft%. You must not use the wildcard search if you provide the whole parameter value.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Case" ACTION="GET">
<COMPANY>%m%</COMPANY>
</COMMAND>
</WAPI>'
```

The following example illustrates a GET search for cases using wildcard characters where LASTNAME contains the substring gate.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
```

```
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Case" ACTION="GET"> <LASTNAME>%gate%</LASTNAME>
</COMMAND>
</WAPI>'
```

The following example illustrates a GET search for cases, using wildcard characters, to fetch all records where LASTNAME begins with g.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Case" ACTION="GET"> <LASTNAME>g%</LASTNAME>
</COMMAND>
</WAPI>'
```

To fetch information on the total number of cases in a tenant, you can send a GET request without specifying any parameters.

The following example illustrates a GET search that fetches a brief summary of cases in the tenant.

```
<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Case" ACTION="GET">
</COMMAND>
</WAPI>'
```

The search yields a brief summary of cases in the tenant including the Case ID of first case, Case ID of the last case, and the total number of cases.

The following example illustrates the general form of the information returned by the CRM API when searching a case without providing any parameters.

```
<FIRSTID>ID of first case record available</FIRSTID>
<LASTID>ID of last case record available</LASTID>
<TOTAL>total number of case records</TOTAL>
```

The following example illustrates the results of a GET case search performed without any parameters. In this example, the first case records equals 1, the last case record equals 100, and the total number of active cases equals 90.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<WAPI>
```

```
<REPLY STATUS="0" ERROR _STR="">
<ITEM>
<FIRST ID>1001</FIRSTID> <LASTID>70030</LASTID>
<TOTAL>90</TOTAL>
</ITEM>
</REPLY>
</WAPI>
```

The following example illustrates a GET search for cases that match a value specified by a custom field of Picklist datatype. In the Local CRM, the contact center administrator uses Configuration Manager to define custom fields and parameters.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Case" ACTION="GET">
<CF01_PICKLIST.NAME>Value</CF01_PICKLIST.NAME>
</COMMAND>
</WAPI>'
```

The following example illustrates a GET search for cases created by Customers.

Customers is the value specified by the CRM API user to fetch all the cases created by customers. This value is not case-sensitive.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Case" ACTION="GET"> <CREATEDBY>Customers</CREATEDBY>
</COMMAND>
</WAPI>'
```

To retrieve cases assigned to a Group, you must get the group ID assigned to the group by the Local CRM. To access the group ID of a given group, run a request to list the groups. See [List Agent GROUPS](#).

The following example illustrates a GET search for cases assigned to Sales group. The Group ID for sales returned by the Local CRM is 101. Group names are defined by the administrator of the tenant in Configuration Manager.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
```

```
<COMMAND OBJECT="Case" ACTION="GET"> <ASSIGNEDTO>101</ASSIGNEDTO>
</COMMAND>
</WAPI> '
```

The following example illustrates a GET search for cases assigned to Deleted Groups.

Deleted Groups is the value specified by the CRM API user to fetch all the cases assigned to deleted groups. This value is not case-sensitive. Only one space character must separate Deleted and Groups.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Case" ACTION="GET">
<ASSIGNEDTO>Deleted Groups</ASSIGNEDTO>
</COMMAND>
</WAPI> '
```

The following example illustrates a GET search for cases created by deleted agents.

Deleted Agents is the value specified by the CRM API user to fetch cases created by deleted agents. This value is not case-sensitive. Only one space character must separate Deleted and Agents. This is useful since your search for all cases does not automatically yield cases created by deleted agents.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Case" ACTION="GET">
<CREATEDBY>Deleted Agents</CREATEDBY>
</COMMAND>
</WAPI> '
```

The following example illustrates a GET search for cases assigned to deleted agents.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Case" ACTION="GET">
<ASSIGNEDTO>Deleted Agents</ASSIGNEDTO>
</COMMAND>
</WAPI> '
```

The following example illustrates a GET search for unassigned cases.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Case" ACTION="GET"> <ASSIGNEDTO>None</ASSIGNEDTO>
</COMMAND>
</WAPI>'
```

The following example illustrates a GET search for cases by customer account number.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123<PASSWORD>
<COMMAND OBJECT="CASE" ACTION="GET">
<ACCOUNTNUM>10000019<ACCOUNTNUM>
</WAPI>'
```

The following example illustrates a search for cases created on a specific day.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123<PASSWORD>
<COMMAND OBJECT="CASE" ACTION="GET">
<FROMCREATEDATE>06202012</FROMCREATEDATE>
<TOCREATEDATE>06252012</TOCREATEDATE>
</WAPI>'
```

The following example illustrates a search for cases last modified on a specific day.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123<PASSWORD>
<COMMAND OBJECT="CASE" ACTION="GET">
<FROMLASTMODDATE>06202012</FROMLASTMODDATE>
<TOLASTMODDATE>06252012</TOLASTMODDATE>
</WAPI>'
```

## List of Parameters

The following table summarizes the `Get` parameters for the `Case` object. At least one parameter must be used in a `Get` request.

Parameter	Format
CASENUM	This case number is returned after a <code>Get</code> action on a <code>Case</code> object.
STATUS	The status must be one valid status (not deactivated) as defined in Configuration Manager. The <code>Get</code> action on a custom field object can be used to determine the list of possible statuses.
FROMDATE	The format for this attribute is MMDDYYYY where MM: Month, DD: Day, YYYY: Year.
TODATE	The format for this attribute is MMDDYYYY where MM: Month, DD: Day, YYYY: Year.
CATEGORY	The category must be valid (not deactivated) as defined in Configuration Manager. The list of possible values is defined by the administrator of the tenant. The <code>Get</code> action on a custom field object can be used to determine the list of possible categories.
PRIORITY	The priority must be valid (not deactivated) as defined in Configuration Manager. The list of possible values is defined by the administrator of the tenant. The <code>Get</code> action on a custom field object can be used to determine the list of possible priorities.
SEVERITY	The severity must be valid (not deactivated) as defined in Configuration Manager. The list of possible values is defined by the administrator of the tenant. The <code>Get</code> action on a custom field object can be used to determine the list of possible severities.
PROJECT	The project must be valid (not deactivated) as defined in Configuration Manager. The list of possible values is defined by the administrator of the tenant. The <code>Get</code> action on a custom field object can be used to determine the list of possible categories.
ASSIGNEDTO	The assigned to must be equal to a valid agents or groups user ID. To search for cases assigned to deleted agents, deleted groups, or unassigned agents the values are:

Parameter	Format
	Deleted Agents, Deleted Groups, or None.
CREATEDBY	<p>The agent must be a valid agent or group user ID. See the <a href="#">List Tenant Configuration Attributes</a> section to get a list of agents and groups.</p> <p>To search a list of all customers, the generic value <code>Customers</code> can be used.</p> <p>To search cases created by deleted agents, the generic value <code>Deleted Agents</code> can be used.</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;">  <p><b>Note:</b> When cases are created by customers, the <code>CREATEDBY</code> tag is returned empty.</p> </div>
SUBJECT	<ul style="list-style-type: none"> <li>■ Any character</li> <li>■ Maximum characters: 255</li> <li>■ Invalid: whitespace-only</li> </ul>
COMPANY	<ul style="list-style-type: none"> <li>■ Any character</li> <li>■ Maximum characters: 50</li> <li>■ Invalid: ~ , &lt;&gt; , whitespace-only</li> </ul>
LASTNAME	<ul style="list-style-type: none"> <li>■ Any character</li> <li>■ Maximum characters: 30</li> <li>■ Invalid: ~ , &lt;&gt; , whitespace-only</li> </ul>
ATTACHMENTS	Possible values: yes, or no.
CFO _ PICKLIST.NAME	<p>Custom fields and their parameter names are added by the administrator of the tenant.</p> <p>The values must be one valid type (not deactivated) as defined in Configuration Manager.</p> <p>The list of possible values is defined by the administrator of the tenant.</p>
CFO2.NAME	<p>Custom fields and their parameter names are added by the administrator of the tenant.</p> <p>The values must be one valid type (not deactivated) as defined in Configuration Manager.</p> <p>The values passed must be in a text or number format.</p>
ACCOUNTNUM	Account number of the customer related to a set of cases. This retrieves all cases related to a given customer.

Parameter	Format
FROMCREATEDATE	The format for this attribute is MMDDYYYY, where MM is the month, DD the day, and YYYY the year.
TOCREATEDATE	The format for this attribute is MMDDYYYY, where MM is the month, DD the day, and YYYY the year.
FROMLASTMODDATE	The format for this attribute is MMDDYYYY, where MM is the month, DD the day, and YYYY the year.
TOLASTMODDATE	The format for this attribute is MMDDYYYY, where MM is the month, DD the day, and YYYY the year.

## Reply from Local CRM to Get Case

The following example illustrates a reply to a Get Case request in the 8x8 Local CRM. In this example, CF01\_PICKLIST.NAME and CF02.NAME are custom fields.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<WAPI>
<REPLY ERROR_CODE="0" ERROR_STR="" STATUS="0">
<ITEM>
<ACCOUNTNUM>123456</ACCOUNTNUM>
<CASENUM>67890</CASENUM>
<SUBJECT>My printer doesn't work</SUBJECT>
<DESCRIPTION>This customer downloaded the new driver XXXX v5.4 of the driver,
installed it and since then the printer doesn't work.</DESCRIPTION>
<STATUS>Closed</STATUS>
<PRIORITY>High</PRIORITY>
<SEVERITY>Information</SEVERITY>
<CATEGORY>Default</CATEGORY>
<PROJECT>Default</PROJECT>
<VISIBILITY>Private</VISIBILITY>
<MEDIATYPE>Phone</MEDIATYPE>
<ASSIGNEDTO>jsmith</ASSIGNEDTO>
<ASSIGNEDDATE>01122010</ASSIGNEDDATE>
<CREATEDBY>bpower</CREATEDBY>
<CREATEDDATE>0 215200 0</CREATEDDATE>
<CLOSEDBY>bpower</CLOSEDBY>
```

```

<CLOSEDDATE>01232000< /CLOSEDDATE>
<LASTACTDATE>01232000</LASTACTDATE>
<CF01_PICKLIST.NAME>value1< /CF01_PICKLIST.NAME>
<CF02.NAME>value2</CF02.NAME>
</ITEM>
<ITEM>
...
</ITEM>
</REPLY>
</WAPI>

```

See [Error Cases](#) for a list of errors to an unsuccessful reply.

## Delete Case

The `Delete Case` action deletes an existing Local CRM case record. This action corresponds to the `Delete Case` action in Agent Console.

- The `Delete` keyword is not case-sensitive.
- Deleting a customer causes a cascade effect; all follow-ups related to a given case are deleted as well.

## Example of Delete Case Operation

The following example illustrates a request to delete a case.

```

\<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123<PASSWORD>
<COMMAND OBJECT="CASE" ACTION="DELETE">
<CASENUM>67890<CASENUM>
</WAPI>'

```

## List of Mandatory and Optional Parameters

The following table summarizes the mandatory and optional `Delete` parameters for the `Case` object.

Parameter	Mandatory/Optional	Format
CASENUM	M	The case number of the case

## Reply from Local CRM to Delete Case

The following example illustrates a reply to a Delete Case request in the 8x8 Local CRM.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<WAPI>
<REPLY STATUS="0" ERROR_STR="" ERROR_CODE="0"/>
</WAPI>
```

See [Error Cases](#) for a list of errors to an unsuccessful reply.

## Manipulate Follow-Up Objects

The section describes the actions that can be performed on an 8x8 Local CRM Follow-up object, including the parameters used in both the request and the reply.

For Follow-up objects, the following actions can be performed:

- [Add Follow-up](#)
- [Get Follow-up](#)

### Add Follow-Up

The `Add Followup` action enables you to add a follow-up to a case in the 8x8 Local CRM. This action corresponds to the `Create follow-up` action in Agent Console.

- The `Followup` keyword is not case-sensitive.
- The `Add` keyword is not case-sensitive.

The following example illustrates an `Add Followup` operation.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>wapi</USERNAME>
<PASSWORD>wapi1</PASSWORD>
<COMMAND OBJECT="Followup" ACTION="Add">
<CASENUM>35</CASENUM>
<SUBJECT>New Order placed</SUBJECT>
<Description>This customer downloaded the new driver XXXX v5.4 of the driver,
installed it and since then the printer doesn't work.</Description>
<STATUS>Open</STATUS>
```

```

<PRIORITY>Medium</PRIORITY>
<SEVERITY>Information</SEVERITY>
<Project>Default</Project>
<CATEGORY>Default</CATEGORY>
<VISIBILITY>Internal</VISIBILITY>
<MEDIATYPE>Voice</MEDIATYPE>
<ASSIGNEDTO>jsmith</ASSIGNEDTO>
</COMMAND>
</WAPI> '

```

## List of Mandatory and Optional Parameters

The following table summarizes the mandatory and optional Add parameters for the Followup object.

Parameter	Mandatory/Optional	Format
CASENUM	M	Case number related to the follow-up. This case number is returned after a Get action on a Case object.
SUBJECT	M	<ul style="list-style-type: none"> <li>■ Any character</li> <li>■ Maximum characters: 255</li> <li>■ Invalid – whitespace-only</li> </ul>
DESCRIPTION	M	<ul style="list-style-type: none"> <li>■ Any character</li> <li>■ Maximum characters: 2 MB</li> <li>■ Invalid: whitespace-only</li> </ul>
STATUS	O	<p>The status must be a valid value (not deleted) as defined in Configuration Manager.</p> <p>The Get action on a custom field object can be used to determine the list of possible statuses.</p>
PRIORITY	O	<p>The priority must be a valid value (not deleted) as defined in Configuration Manager.</p> <p>The list of possible values is defined by the administrator of the tenant.</p> <p>The Get action on a custom field object can be used to determine the list of possible priorities.</p>
SEVERITY	O	The severity must be a valid value (not deleted) as defined in Configuration Manager.

Parameter	Mandatory/Optional	Format
		<p>The list of possible values is defined by the administrator of the tenant.</p> <p>The <code>Get</code> action on a custom field object can be used to determine the list of possible severities.</p>
PROJECT	O	<p>The project must be a valid project (not deleted) as defined in Configuration Manager.</p> <p>The list of possible values is defined by the administrator of the tenant.</p> <p>The <code>Get</code> action on a custom field object can be used to determine the list of possible projects.</p>
CATEGORY	O	<p>The category must be one valid category (not deleted) as defined in Configuration Manager.</p> <p>The list of possible values is defined by the administrator of the tenant.</p> <p>The <code>Get</code> action on a custom field object can be used to determine the list of possible categories.</p>
VISIBILITY	O	<p>The visibility must be one valid value (not deleted) as defined in Configuration Manager.</p> <p>Possible values are <code>Internal</code> or <code>External</code>.</p> <p>If no visibility is provided, the default value is used.</p>
MEDIATYPE	O	<p>The <code>mediatype</code> must be one valid media type (not deleted) as defined in Configuration Manager.</p> <p>Possible values are: <code>Phone</code>, <code>Chat</code>, <code>Email</code>, or <code>Other</code>.</p> <p>If no <code>mediatype</code> is provided, the default value is used.</p>
ASSIGNEDTO	O	<p>The <code>ASSIGNEDTO</code> must be equal to a valid agents or groups user id.</p> <p>If <code>ASSIGNEDTO</code> is used, the <code>STATUS</code> parameter must be consistent.</p> <p>To search for cases assigned to deleted agents, deleted groups, or unassigned cases the values are: <code>Deleted Agents</code>, <code>Deleted Groups</code>, or <code>None</code>.</p>
CF01_ PICKLIST.NAME	O	<p>Custom fields and their parameter names are added by the administrator of the tenant. The values must be valid (not deactivated) and match the values defined in Configuration Manager. The list of possible values is</p>

Parameter	Mandatory/Optional	Format
		defined by the administrator of the tenant.
CF02.NAME	O	Custom fields and their parameter names are added by the administrator of the tenant. The values must be valid (not deactivated) and match the values defined in Configuration Manager. The values passed must be in a text or number format.

Case modification generates automated follow-ups containing a description of the modification consistent with an edit action taken through Agent Console.

## Reply from Local CRM to Add Follow-Up

The following example illustrates a reply to an `Add Followup` request in the 8x8 Local CRM.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<WAPI>
REPLY ERROR_CODE="0" ERROR_STR="" STATUS="0">
<FOLLOWUPNUM>11</FOLLOWUPNUM>
</REPLY>
</WAPI>
```

See [Error Cases](#) for a list of errors to an unsuccessful reply.

## Get Follow-Up

The `Get Followup` action enables you to get a list of existing follow-ups in the Local CRM that matches the supplied parameter values. This action corresponds to the `Search`, `View`, and `List` actions in Agent Console.

- The `Get` keyword is not case-sensitive.
- Command object only needs to be passed once, even when searching for multiple parameters.
- The Case Number is a mandatory field to fetch follow-up data.
- Follow-up records cannot be referenced by any field other than the case number.
- When searching follow-ups, only one case number can be provided in the `Get` request.
- Wildcard search is limited to case numbers.

## Tags to Pair for Range Queries

- `FromCreateDate` / `ToCreateDate`

## Examples of Get Follow-up Operations

The following example illustrates searching follow-ups for a specific case.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Followup" ACTION="GET">
<CASENUM>67890</CASENUM>
</COMMAND>
</WAPI>'
```

The following example illustrates a wildcard search for follow-ups to all CASENUM containing the substring 78.

```
<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Followup" ACTION="GET">
<CASENUM>%78%</CASENUM>
</COMMAND>
</WAPI>
```

The following example illustrates a search for retrieving follow-up created on a specific day.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123<PASSWORD>
<COMMAND OBJECT="FOLLOWUP" ACTION="GET">
<CASENUM>67890</CASENUM>
<FROMCREATEDATE>06202012</FROMCREATEDATE>
<TOCREATEDATE>06252012</TOCREATEDATE>
</WAPI>'
```

## List of Parameters

The CRM API allows referencing follow-up records only by case numbers.

The following table summarizes the Get parameters for the Followup object.

Parameter	Format
CASENUM	Case number related to the follow-ups. This case number is returned after a <code>Get</code> action on a Case object.
FROMCREATEDATE	The format for this attribute is MMDDYYYY, where MM is the month, DD the day, and YYYY the year.
TOCREATEDATE	The format for this attribute is MMDDYYYY, where MM is the month, DD the day, and YYYY the year.

## Reply from Local CRM to Get Follow-Up

The following example illustrates a reply to a `Get Followup` request in the 8x8 Local CRM.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
-<WAPI>
-<REPLY ERROR_CODE="0" ERROR_STR="" STATUS="0">
-<ITEM><CASENUM>35</CASENUM>
<FOLLOWUPNUM>3</FOLLOWUPNUM>
<SUBJECT>Order Received - Issue Resolved</SUBJECT>
<CREATEDDATE>0503201212:26:59</CREATEDDATE>
<ATTACHMENTS>0</ATTACHMENTS>
<FOLLOWUPID>5</FOLLOWUPID>
<CREATEDBY>jsmith</CREATEDBY>
<status>Open</status>
<priority>Medium</priority>
<severity>Information</severity>
<category>Default</category>
<project>Default</project>
<mediatype>Voice</mediatype>
<visibility>Internal</visibility>
<DESCRIPTION>This customer downloaded the new driver XXXX v5.4 of the driver,
installed it and since then the printer doesn't work.</DESCRIPTION>
</ITEM>
<ITEM>
...
</ITEM>
</REPLY>
</WAPI>
```

See [Error Cases](#) for a list of errors to an unsuccessful reply.

## Delete Follow-Up

The `Get Followup` action enables you to delete an existing follow-up record in the 8x8 Local CRM.

- The `Delete` keyword is not case-sensitive.
- You must specify the case from which the follow-up should be deleted.

### Example of Delete Follow-Up Operation

The following example illustrates a request to delete a follow-up.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123<PASSWORD>
<COMMAND OBJECT="FOLLOWUP" ACTION="DELETE">
<CASENUM>67890<CASENUM>
<FOLLOWUPNUM>2</FOLLOWUPNUM>
</WAPI>'
```

### List of Mandatory and Optional Parameters

The following table summarizes the mandatory and optional `Delete` parameters for the `Followup` object.

Parameter	Mandatory/Optional	Format
FOLLOWUPNUM	M	The follow-up number of the follow-up
CASENUM	M	The case number of the case related to a given follow-up

### Reply from Local CRM to Delete Follow-Up

The following example illustrates a reply to a `Delete Followup` request in the 8x8 Local CRM.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<WAPI>
<REPLY STATUS="0" ERROR_STR="" ERROR_CODE="0"/>
</WAPI>
```

See [Error Cases](#) for a list of errors to an unsuccessful reply.

## Manipulate Task Objects

You can use the CRM API to add, modify, and get task data. This section outlines all the possible actions that can be performed on a Task object, and all the parameters that are used in both the request and the reply.

You can perform the following actions on the Task object:

- **Add Task**
- **Modify Task**
- **Get Task**

### Add Task

The `Add Task` action enables you to add a new task to the 8x8 Local CRM. This action corresponds to the `Create Task` action in Agent Console.

- The `Task` keyword is not case-sensitive.
- The `Add` keyword is not case-sensitive.
- The `ACCOUNTNUM` is a mandatory parameter to pass.

The following example illustrates an `Add Task` operation. In this example, `CF01_PICKLIST_NAME` and `CF02.NAME` are custom fields.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Task" ACTION="Add">
<ACCOUNTNUM>10000019</ACCOUNTNUM>
<CREATEDBY>jsmith</CREATEDBY>
```

```

<SUBJECT>Phone Appointment</SUBJECT>
<DESCRIPTION>Customer has asked to discuss his questions about the Promotional
campaign.</DESCRIPTION>
<TASK_STATUS>Pending</TASK_STATUS>
<ASSIGNEDTO>jsmith</ASSIGNEDTO>
<CALL_TYPE>DAA</CALL_TYPE>
<MEDIA_TYPE>Phone</MEDIA_TYPE>
<CAMPAIGNNAME>Promotional Campaign</CAMPAIGNNAME>
<CALLDURATION>360</CALLDURATION>
<TRANSID>1111111111</TRANSID>
<EMAIL>jsmith@AcmeJets.com</EMAIL>
<MEDIATYPE>Phone</MEDIATYPE>
<CF01_PICKLIST.NAME>value1</CF01_PICKLIST.NAME>
<CF02.NAME>value2</CF02.NAME>
</COMMAND>
</WAPI> '

```

## List of Mandatory and Optional Parameters

The following table summarizes the mandatory and optional Add parameters for the Task object.

Parameter	Mandatory/Optional	Format
ACCOUNTNUM	M	The account number of the customer who is associated to the case. This account number is returned after a Get action on a Customer object.
SUBJECT	O	<ul style="list-style-type: none"> <li>■ Any character</li> <li>■ Maximum characters: 255</li> <li>■ Invalid: whitespace-only</li> </ul>
DESCRIPTION	O	<ul style="list-style-type: none"> <li>■ Any character</li> <li>■ Maximum characters: 2 MB</li> <li>■ Invalid: whitespace-only</li> </ul>
ASSIGNEDTO	O	<p>The ASSIGNEDTO must be equal to a valid agents or groups user id. If ASSIGNEDTO is used, the STATUS parameter must also be defined. If the value is not passed, the value of USERNAME is used.</p> <p>To search for tasks assigned to deleted agents, deleted groups, or unassigned agents the values are:</p>

Parameter	Mandatory/Optional	Format
		Deleted Agents, Deleted Groups, or None.
TASK_STATUS	O	The status must be one valid status (not deactivated) as defined in Configuration Manager. If no status is provided, the default value is <i>New</i> .
MEDIATYPE	O	The mediatype must be one valid media type (not deactivated) as defined in Configuration Manager. Possible values are: Phone, Chat, Email, or None.
CALL_TYPE	O	Call Type must be valid.
PHONE	O	Maximum 128 (digits and formatting characters must be used)
EMAIL	O	Valid email address, maximum 255 characters.
DUEDATE	O	Due date is the date and time by which the task must be completed. It is represented as MMDDYYYYHH24MI (12 characters) such as 062120121315 for June 21st 2012 1:15PM.
REMINDER	O	<ul style="list-style-type: none"> <li>■ Y or N</li> <li>■ REMINDER needs to be used with DUEDATE</li> </ul>
CAMPAIGNNAME	O	<ul style="list-style-type: none"> <li>■ Maximum characters: 255</li> <li>■ Any character</li> <li>■ Invalid: whitespace-only</li> </ul>
CALLDURATION	O	<ul style="list-style-type: none"> <li>■ Call duration in seconds</li> <li>■ Digits only</li> <li>■ Maximum digits: 13</li> </ul>
TRANSID	O	<ul style="list-style-type: none"> <li>■ Transaction ID</li> <li>■ Digits only</li> <li>■ Maximum digits: 10</li> </ul>
CF01_ PICKLIST.NAME	O	Custom fields and their parameter names are added by the administrator of the tenant. The values must be valid (not deactivated) as defined in Configuration Manager. The list of possible values is defined by the administrator of the tenant.
CF02.NAME	O	Custom fields and their parameter names are added by the administrator

Parameter	Mandatory/Optional	Format
		of the tenant. The values must be valid (not deactivated) as defined in Configuration Manager. The values passed must be in a text or number format.

## Reply from Local CRM to Add a Task

The following example illustrates a reply to an `Add Task` request in the 8x8 Local CRM.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<WAPI>
<REPLY STATUS="0" ERROR _STR="">
<TASKNUM>100</TASKNUM>
</REPLY>
</WAPI>
```

See [Error Cases](#) for a list of errors to an unsuccessful reply.

## Modify Task

The `Modify Task` action modifies an existing Local CRM task record. This action corresponds to the `Edit Case` action in Agent Console.

- The `Modify` keyword is not case-sensitive.
- The user may only send mandatory and modified parameters.
- Any parameter not supplied in a `Modify` request retains the previous value.
- If no value is modified, a reply status of 0 is returned but the task is left unchanged
- Any change in the task updates its last modification date attribute and also causes the creation of a task activity that reports on the updated fields.

The following example illustrates a `MODIFY` request for a task. In this example, `CF01_PICKLIST.NAME` and `CF02.NAME` identify previously identified custom Local CRM fields.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Case" ACTION="Modify">
<TASKNUM>100</TASKNUM>
```

```

<SUBJECT>Phone call completed</SUBJECT>
<DESCRIPTION>Walk through on the promotional campaign options completed with the
customer.</DESCRIPTION>
<TASK_STATUS>Closed</TASK_STATUS>
<ASSIGNEDTO>jsmith</ASSIGNEDTO>
<CF01_PICKLIST.NAME>value3</CF01_PICKLIST.NAME>
<CF02.NAME>value4</CF02.NAME>
</COMMAND>
<WAPI>'

```

## List of Mandatory and Optional Parameters

The following table summarizes the mandatory and optional `Modify` parameters for the `Task` object.

Parameter	Mandatory/Optional	Format
TASKNUM	M	Task number that needs to be modified. This task number is returned after a <code>Get</code> action on a <code>Task</code> object.
SUBJECT	O	<ul style="list-style-type: none"> <li>■ Any character</li> <li>■ Maximum characters: 255</li> <li>■ Invalid: whitespace-only</li> </ul>
DESCRIPTION	O	<ul style="list-style-type: none"> <li>■ Any character</li> <li>■ Maximum characters: 2MB</li> <li>■ Invalid: whitespace-only</li> </ul>
ASSIGNEDTO	O	<p>The <code>ASSIGNEDTO</code> must be equal to a valid agent or group user ID. If <code>ASSIGNEDTO</code> is used, the <code>STATUS</code> parameter must also be defined. If the value is not passed, the value of <code>USERNAME</code> is used.</p> <p>To search for tasks assigned to deleted agents, deleted groups, or unassigned agents, the values are <code>Deleted Agent</code>, <code>Deleted Groups</code>, or <code>None</code>.</p>
TASK_STATUS	O	The status must be one valid status (not deactivated) as defined in Configuration Manager.
MEDIATYPE	O	<p>The mediatype must be valid (not deactivated) as defined in Configuration Manager.</p> <p>Possible values are: <code>Phone</code>, <code>Chat</code>, <code>Email</code>, or <code>None</code>.</p>

Parameter	Mandatory/Optional	Format
CALL_TYPE	O	Call Type must be valid.
PHONE	O	Maximum 128 digits and formatting characters must be used.
EMAIL	O	Valid email address, 255 characters maximum.
DUEDATE	O	Due Date is the date and the time by which the task must be completed. It is represented as MMDDYYYYHH24MI (12 characters), such as 062120121315 for June 21st 2012 1:15PM
REMINDER	O	<ul style="list-style-type: none"> <li>■ Y or N.</li> <li>■ REMINDER must be used with DUEDATE.</li> </ul>
CF01_PICKLIST.NAME	O	Custom fields and their parameter names are added by the administrator of the tenant. The values must be one valid type (not deactivated) as defined in Configuration Manager. The list of possible values is defined by the administrator of the tenant.
CF02.NAME	O	Custom fields and their parameter names are added by the administrator of the tenant. The values must be one valid type (not deactivated) as defined in Configuration Manager. The values passed should be in a text or number format.

## Reply from Local CRM to Modify Task

The following example illustrates a reply to a `Modify Task` request in the 8x8 Local CRM.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<WAPI>
<REPLY STATUS="0" ERROR _STR="">
<TASKNUM>100</TASKNUM>
</REPLY>
</WAPI>
```

See [Error Cases](#) for a list of errors to an unsuccessful reply.

## Get Task

The `Get Task` action enables you to get a list of existing tasks from the 8x8 Local CRM that matches the supplied parameter values. This action corresponds to the `Search`, `View`, and `List` actions in Agent Console.

- The `Get` keyword is not case-sensitive.
- Command object only needs to be passed once, even when searching for multiple parameters.
- Parameter values being searched are case-sensitive.
- The request returns entire descriptions for each task that matches the supplied parameter values, including the customer account number.

The query parameters on a `Get` request can vary in the following ways:

- Query on a substring
- Pass multiple parameters to generate an `OR` query
- Pass multiple parameters to generate a `range` query

### Tags that Accept Substring Values

- Subject
- Phone
- Email
- CampaignName
- TransID



**Note:** The parameters values being searched as a substring require the use of "%". For example, to search for the word `call`, the value must be set as: `%call%`.

### Tags that Accept Predefined Values

- CreatedBy
- AssignedTo
- Task\_Status
- Call\_Type
- Media\_Type

### Tags to Pair for Range Queries

- FromCreateDate / ToCreateDate
- FromDueDate / ToDueDate

- FromLastModDate / ToLastModDate



**Note:** To retrieve tasks created on a particular date, `FromCreateDate` must correspond to the first desired date, and `ToCreateDate` must correspond to the day after. For example, a `FromCreateDate` of 02292012 to a `ToCreateDate?` of 03012012 retrieves tasks created on the February 29th, 2012.

## Examples of Get Queries

The following example illustrates a `Get Case` search given a case number.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Task" ACTION="GET"> <TASKNUM>100</TASKNUM>
</COMMAND>
</WAPI>'
```

The following example illustrates a `GET` query to retrieve tasks whose status is closed. The task status must match the values defined in Configuration Manager. The `GET` action on a custom field object can be used to determine the list of possible statuses.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Task" ACTION="GET">
<TASK_STATUS>Closed</TASK_STATUS>
</COMMAND>
</WAPI>'
```

The following example illustrates a search for retrieving tasks with a specified due date range.

Only one occurrence of the `FromDueDate` and `ToDueDate` parameters can be used in one `Get` request.

To retrieve tasks due on a particular day, `FromDueDate` must correspond to the first date desired and the `ToDueDate` must correspond to the day after. For example, a `FromDueDate` of 12072001 to a `ToDueDate` of 12082001 retrieves cases created on December 7<sup>th</sup>, 2001.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
```

```
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Task" ACTION="GET">
<FROMDUEDATE>060120120000</FROMDUEDATE>
<TODUEDATE>061520120000</TODUEDATE>
</COMMAND>
</WAPI>'
```

The following example illustrates a search for retrieving tasks assigned to a specific agent. You can pass multiple parameters to look for tasks assigned to multiple agents.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Task" ACTION="GET">
<ASSIGNEDTO>jdoe</ASSIGNEDTO>
</COMMAND>
</WAPI>'
```

The following example illustrates a search for retrieving tasks created by a specific agent. You can pass multiple parameters to look for tasks created by multiple agents.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Task" ACTION="GET">
<CREATEDBY>jdoe</CREATEDBY>
</COMMAND>
</WAPI>'
```

The following example illustrates a search for retrieving tasks with a specific keyword in the task subject.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Task" ACTION="GET">
<SUBJECT>%promotional%</SUBJECT>
</COMMAND>
</WAPI>'
```



**Note:** Values being searched as a substring require the use of "%". For instance, to search for all promotional campaigns, the value must be set as: %promotional%. If you don't use the '%' wildcard characters, the CRM API performs an exact match instead, as shown in the following example.

The following example illustrates a search for retrieving tasks with a specific keyword in the task subject.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Task" ACTION="GET">
<SUBJECT>promotional campaign 10</SUBJECT>
</COMMAND>
</WAPI>'
```

The following example illustrates a search for retrieving tasks last modified on a specific day.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Task" ACTION="GET">
<FROMLASTMODDATE>06202012</FROMLASTMODDATE>
<TOLASTMODDATE>06252012</TOLASTMODDATE>
</COMMAND>
</WAPI>'
```



**Note:** FROMLASTMODDATE and TOLASTMODDATE don't necessarily have to be used together. If only FROMLASTMODDATE is used, the CRM API engine looks for tasks modified after the passed date. Conversely, if only TOLASTMODATE is used, the CRM API engine looks for tasks modified before the passed date. The same concept applies for FROMCREATEDATE/TOCREATEDATE and for FROMDUEDATE/TODUEDATE.

The following example illustrates a search for retrieving tasks related to a campaign.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
```

```
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Task" ACTION="GET">
<CAMPAIGNNAME>%Promotional campaign for Holidays%</CAMPAIGNNAME>
</COMMAND>
</WAPI>'
```

The following example illustrates a search for retrieving tasks related to a specific phone number.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Task" ACTION="GET">
<PHONE>1-555-292-8608</PHONE>
</COMMAND>
</WAPI>'
```

The following example illustrates a search for retrieving tasks related to a specific email.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Task" ACTION="GET">
<EMAIL>sirben.kingsley@stars.com</EMAIL>
</COMMAND>
</WAPI>'
```

The following example illustrates a search for retrieving tasks related to a specific Transaction ID.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Task" ACTION="GET">
<TRANSID>12345</TRANSID>
</COMMAND>
</WAPI>'
```

The following example illustrates a search for retrieving tasks related to call type.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
```

```
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Task" ACTION="GET">
<CALL_TYPE>Outbound</CALL_TYPE>
</COMMAND>
</WAPI>'
```

The following example illustrates a search for retrieving tasks related to media type.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Task" ACTION="GET">
<MEDIA_TYPE>Outbound</MEDIA_TYPE>
</COMMAND>
</WAPI>'
```

The next example illustrates a `Get Task` query without passing any parameters. The CRM API returns:

- **FIRSTID:** ID of first task record available
- **LASTID:** ID of last task record available
- **TOTAL:** Total number of task records

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Task" ACTION="GET">
</COMMAND>
</WAPI>'
```

The following example illustrates the reply received from Local CRM after searching for a task without passing parameters.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<WAPI>
<REPLY STATUS="0" ERROR_STR="0">
<ITEM> <FIRSTID>170</FIRSTID>
<LASTID>263</LASTID>
<TOTAL>79</TOTAL>
</ITEM>
```

The following example retrieves tasks matching a specific value of a custom picklist field.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Task" ACTION="GET">
<TASK_PICKLIST>two</TASK_PICKLIST>
</COMMAND>
</WAPI>'
```



**Note:** TASK\_PICKLIST is the Field Name attribute for the picklist custom field defined in the Local CRM. You have to pass a valid value.

The following example receives tasks with a specific value in a custom task\_text field.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Task" ACTION="GET">
<TASK_TEXT>random text</TASK_TEXT>
</COMMAND>
</WAPI>'
```



**Note:** TASK\_TEXT is the Field Name attribute for the text custom field defined in the Local CRM. You have to pass a valid value.

The following example retrieves tasks matching the date specified in the TASK\_DATE custom field.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Task" ACTION="GET">
<TASK_DATE>06222012</TASK_DATE>
</COMMAND>'
```

```
</WAPI>'
```



**Note:** TASK\_DATE is the Field Name attribute for the date custom field defined in the Local CRM.

The following example retrieves tasks created by and assigned to deleted agents.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Task" ACTION="GET">
<CREATEDBY>Deleted Agents</CREATEDBY>
<ASSIGNEDTO>Deleted Agents</ASSIGNEDTO>
</COMMAND>
</WAPI>'
```



**Note:** Deleted Agents is the value the CRM API user has to specify to fetch tasks created by deleted agents. This value is not case-sensitive. Only one space must separate Deleted and Agents.

The following example illustrates a search for tasks by customer account number.

```
`<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123<PASSWORD>
<COMMAND OBJECT="TASK" ACTION="GET">
<ACCOUNTNUM>10000019<ACCOUNTNUM>
</WAPI>'
```

## List of Parameters

The following table summarizes the Get parameters for the Task object.

Parameter	Format
TASKNUM	Task number that needs to be modified. This task number is returned after a Get action on a Task object.

Parameter	Format
SUBJECT	<ul style="list-style-type: none"> <li>■ Any character</li> <li>■ Max: 255</li> <li>■ Invalid: whitespace-only</li> </ul>
DESCRIPTION	<ul style="list-style-type: none"> <li>■ Any character</li> <li>■ Max: 2MB</li> <li>■ Invalid: whitespace-only</li> </ul>
ASSIGNEDTO	<p>The ASSIGNEDTO value must be equal to a valid agent or group user ID. If ASSIGNEDTO is used, the <code>status</code> parameter must also be defined. If the value is not passed, the value of <code>USERNAME</code> is used.</p> <p>To search for tasks assigned to deleted agents, deleted groups, or unassigned agents, the values are <code>Deleted Agents</code>, <code>Deleted Groups</code>, or <code>None</code>.</p>
CREATEDBY	It must be a valid agent, or Deleted Agents to search for tasks created by deleted agents.
TASK_STATUS	The status must be one valid status (not deactivated) as defined in Configuration Manager.
FROMDUEDATE	The format for this attribute is: MMDDYYYY where MM is the month, DD the day and YYYY the year.
TODUEDATE	The format for this attribute is: MMDDYYYY where MM is the month, DD the day and YYYY the year.
FROMLASTMODDATE	The format for this attribute is: MMDDYYYY where MM is the month, DD the day and YYYY the year.
TOLASTMODDATE	The format for this attribute is: MMDDYYYY where MM is the month, DD the day and YYYY the year.
MEDIATYPE	<p>The mediatype must be one valid media type (not deactivated) as defined in Configuration Manager.</p> <p>Possible values are <code>Phone</code>, <code>Chat</code>, <code>Email</code>, or <code>None</code>.</p>
CALL_TYPE	Call Type must be valid.
PHONE	Maximum 128 digits and formatting characters must be used
EMAIL	The valid email address is 255 characters maximum
CAMPAGINNAME	<ul style="list-style-type: none"> <li>■ Any character</li> </ul>

Parameter	Format
	<ul style="list-style-type: none"> <li>■ Maximum characters: 255</li> <li>■ Invalid: whitespace-only</li> </ul>
TRANSID	<ul style="list-style-type: none"> <li>■ Transaction ID</li> <li>■ Digits only</li> <li>■ Maximum digits: 10</li> </ul>
REMINDER	Y or N. REMINDER must be used with DUEDATE.
CF01_PICKLIST.NAME	Custom fields and their parameter names are added by the administrator of the tenant. The values must be one valid type (not deactivated) as defined in Configuration Manager. The list of possible values is defined by the administrator of the tenant.
CF02.NAME	Custom fields and their parameter names are added by the administrator of the tenant. The values must be one valid type (not deactivated) as defined in Configuration Manager. The values passed should be in a text or number format.
ACCOUNTNUM	Account number of the customer related to a set of tasks. This retrieves all tasks related to a given customer.

## Reply from Local CRM to Get Task

The following example illustrates a reply to a `Get Task` request in the 8x8 Local CRM.

In the preceding example, `task_text`, `task_text_unique`, `task_url`, `task_number`, `task_number_unique`, `task_date`, and `task_picklist` are custom fields with the obvious type.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<WAPI>
<REPLY ERROR_CODE="0" ERROR_STR="" STATUS="0">
<ITEM>
<TASKNUM>183</TASKNUM>
<SUBJECT>Follow-up with customer on pricing</SUBJECT>
<DESCRIPTION>Description for task</DESCRIPTION>
<CREATEDDATE>04062012 18:33:49</CREATEDDATE>
<DUEDATE>06092012 03:00:00</DUEDATE>
<REMINDER>N</REMINDER>
<PHONE/>
<LASTMODDATE>04092012 18:04:02</LASTMODDATE>
```

```

<EMAIL>angel@8x8.com</EMAIL>
<TRANSID/>
<CALLDURATION>45</CALLDURATION>
<CAMPAIGNNAME/>
<task_text>some text</task_text>
<task_text_unique>some unique string 4724732827489324893274</task_text_unique>
<task_url>http://www.8x8.com</task_url>
<task_number>111444</task_number>
<task_number_unique>613293374</task_number_unique>
<task_date>04062012</task_date>
<task_picklist>three</task_picklist>
<task_status>New</task_status>
<media_type>Email</media_type>
<call_type/>
<ACCOUNTNUM>10000001</ACCOUNTNUM>
<ASSIGNEDTO>pma</ASSIGNEDTO>
<CREATEDBY>pma</CREATEDBY>
</ITEM>
<ITEM>
...
</ITEM>
</REPLY>
</WAPI>

```

See [Error Cases](#) for a list of errors to an unsuccessful reply.



**Notes:**

- When ASSIGNEDTO is not passed in a GETTASK query, only tasks assigned to valid (not deleted) agents are returned.
- When CREATEDBY is not passed in a GETTASK query, only tasks created by valid (not deleted) agents are returned.
- So, if you want to retrieve a task created AND assigned to deleted agents (not necessarily the same as long as they are both deleted), you must use both tags in the XML query with the same "Deleted Agents" value.

## Delete Task

The `Delete Task` action deletes an existing task record in the 8x8 Local CRM. This action corresponds to the `Delete Task` action in Agent Console.

- The `Delete` keyword is not case-sensitive.

### Example of Delete Task Operation

The following example illustrates a request to delete a task.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123<PASSWORD>
<COMMAND OBJECT="TASK" ACTION="DELETE">
<TASK>183<TASK>
</WAPI>'
```

### List of Mandatory and Optional Parameters

The following table summarizes the mandatory and optional `Delete` parameters for the `Task` object.

Parameter	Mandatory/Optional	Format
TASKNUM	M	The task number of the task

### Reply from Local CRM to Delete Task

The following example illustrates a reply to a `Delete Task` request in the 8x8 Local CRM.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<WAPI>
<REPLY STATUS="0" ERROR_STR="" ERROR_CODE="0"/>
</WAPI>
```

See [Error Cases](#) for a list of errors to an unsuccessful reply.

## Manipulate FAQ Category Objects

The 8x8 Local CRM allows a tenant administrator or supervisor to create an FAQ library for its agents to quickly find answers to frequently asked questions. The FAQ library contains FAQ categories and organizes FAQs and answers under each category.

For example, AcmeJets defines FAQ categories for Support, Sales, and Safety. Questions and answers are grouped under the related category.

This section outlines the actions that can be performed on an FAQ Category object in the 8x8 Local CRM, and the parameters that are used in both the request and the reply.

For the FAQ Category object, the following actions can be performed:

- Add FAQ Category
- Modify FAQ Category
- Get FAQ Category
- Delete FAQ Category

### Add FAQ Category

The `Add FAQCategory` action enables a tenant to add a new FAQ category to the Local CRM. This action corresponds to the `Create` action in Supervisor Console.

- The `FAQCategory` keyword is not case-sensitive.
- The `Add` keyword is not case-sensitive.

The following example illustrates an `Add` operation for `FAQCategory`.

```
'<WAPI>  
<TENANT>AcmeJets</TENANT>
```

```

<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="FAQCategory" ACTION="Add">
<FAQCATEGORY>Safety</FAQCATEGORY>
<PRIVATE>Y</PRIVATE>
</COMMAND>
</WAPI> '

```

## Mandatory and Optional Parameters

The following table summarizes the mandatory and optional Add parameters for the `FAQCategory` object.

Parameter	Mandatory/Optional	Format
FAQCATEGORY	M	<ul style="list-style-type: none"> <li>■ Any character that is not defined is invalid .</li> <li>■ The defined characters are alphabet characters, numbers, and special characters such as ~, !, @, #, \$, %, ^, :, ;, ', ,, ., ? /</li> <li>■ &amp;, *, (, ), -, =, _ , {, }, [, ]   are not accepted.</li> <li>■ Maximum characters: 50</li> <li>■ Invalid: whitespace-only</li> </ul>
PRIVATE	O	<ul style="list-style-type: none"> <li>■ Possible values are Y or N.</li> <li>■ The default value is N.</li> <li>■ Values are case-sensitive.</li> <li>■ Selecting Y declares the FAQ category private and vice versa.</li> </ul>

## Reply from Local CRM to Add FAQ Category

The following example illustrates a reply to an Add `FAQCategory` request in the 8x8 Local CRM.

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<WAPI>
<REPLY ERROR_CODE="0" ERROR_STR="" STATUS="0">
<FAQCATID>1</FAQCATID>
</REPLY>
</WAPI>

```

See [Error Cases](#) for a list of errors to an unsuccessful reply.

## Modify FAQ Category

The `Modify FAQCategory` action enables you to modify an existing FAQ category in the 8x8 Local CRM. This action corresponds to the `Edit FAQ Category` action in Supervisor Console.

- The `Modify` keyword is not case-sensitive.
- Any parameter not supplied in a `Modify` request retains its previous value.
- The parameter value passed to declare a category private or public is case-sensitive.

The example below illustrates a request to modify an existing FAQ category. The category is referenced by its category ID, which can be accessed by one of the following:

- Passing a request to `Get` FAQ categories.
- Passing a request to `List` all FAQ categories defined for the tenant.

```
<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="FAQCategory" ACTION="Modify"> <FAQCATID>1</FAQCATID>
<FAQCATEGORY>Legalities</FAQCATEGORY>
<PRIVATE>Y</PRIVATE>
</COMMAND>
</WAPI>
```

## Summary of Mandatory and Optional Parameters

The following table summarizes the mandatory and optional `Modify` parameters for the `FAQCategory` object.

Parameter	Mandatory/Optional	Format
FAQCATID	M	Internal ID of the FAQ category. This ID is returned after an <code>Add</code> or <code>Get</code> action on an <code>FAQCategory</code> object.
FAQCATEGORY	M	<ul style="list-style-type: none"> <li>■ Any character</li> <li>■ Maximum characters: 50</li> <li>■ Invalid: whitespace-only</li> </ul>
PRIVATE	O	<ul style="list-style-type: none"> <li>■ Possible values: Y or N.</li> <li>■ Values are case-sensitive.</li> </ul>

Parameter	Mandatory/Optional	Format
		<ul style="list-style-type: none"> <li>The default value is N.</li> </ul>

See [Error Cases](#) for a list of errors to a unsuccessful reply.

## Reply from Local CRM to Modify FAQ Category

The following example illustrates a reply to a `Modify FAQCategory` request in the 8x8 Local CRM.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<WAPI>
<REPLY ERROR_CODE="0" ERROR_STR="" STATUS="0">
<FAQCATID>1</FAQCATID>
</REPLY>
</WAPI>
```

See [Error Cases](#) for a list of errors to an unsuccessful reply.

## Get FAQ Category

The `Get FAQCategory` action retrieves all FAQ categories from the Local CRM.

- The `Get` keyword is not case-sensitive.
- Command object only needs to be passed once, even when searching for multiple parameters.
- The parameter values are case-sensitive.

The following example fetches all FAQ categories defined in the tenant.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="FAQCategory" ACTION="Get">
</COMMAND>
</WAPI>'
```

## Reply from Local CRM to Get FAQ Category

The following example illustrates a reply to a `Get FAQCategory` request in the 8x8 Local CRM.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<WAPI>-<REPLY ERROR_CODE="0" ERROR_STR="" STATUS="0">-
<ITEM><FAQCATID>2</FAQCATID><FAQCATEGORY>Safety</FAQCATEGORY></ITEM>
<ITEM><FAQCATID>3</FAQCATID><FAQCATEGORY>legalities</FAQCATEGORY></ITEM>
<ITEM><FAQCATID>1</FAQCATID><FAQCATEGORY>Services</FAQCATEGORY></ITEM>
</REPLY>
</WAPI>
```

See [Error Cases](#) for a list of errors to an unsuccessful reply.

## Delete FAQ Category

The `Delete FAQCategory` action enables you to delete an existing FAQ Category in the 8x8 Local CRM. The `Delete` keyword is not case-sensitive.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="FAQCategory" ACTION="Delete"> <FAQCATID>4</FAQCATID>
</COMMAND>
</WAPI>'
```

## Mandatory and Optional Parameters

The following table summarizes the mandatory and optional `Delete` parameters for the `FAQCategory` object.

Parameter	Mandatory/Optional	Format
FAQCATID	M	Internal ID of the FAQ category. This ID is returned after an <code>Add</code> or <code>Get</code> action on an <code>FAQCategory</code> object.

## Reply from Local CRM to Delete FAQ Category

The following example illustrates a reply to a `Delete FAQCategory` request in the 8x8 Local CRM.

- You cannot delete a category that contains FAQ items.
- You must delete all FAQ items in a category before attempting to delete an `FAQCategory`.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<WAPI>
<REPLY ERROR_CODE="0" STATUS="0" ERROR_STR="">
```

```
</REPLY>  
</WAPI>
```

See **Error Cases** for a list of errors to an unsuccessful reply.

## Manipulate FAQ Objects

This section outlines the possible actions that can be performed on an FAQ object in the 8x8 Local CRM, and the parameters that are used in both the request and the reply.

You can perform the following actions on the FAQ object:

- **Add FAQ**
- **Modify FAQ**
- **Get FAQ**
- **Delete FAQ**

### Add FAQ

The `Add FAQ` action enables the tenant to add an FAQ to an FAQ category in the 8x8 Local CRM. This action corresponds to the `Add FAQ` action in Agent Console.

- The `FAQ` keyword is not case-sensitive.
- FAQs are limited to non-HTML format only.
- `FAQCATID` identifies the FAQ category to which the FAQ question and answer are added.
- `FAQCATID`, `QUESTION`, and `ANSWER` are mandatory data to add an FAQ item.
- `SHORTCUT` uniquely identifies an FAQ item and allows agents to reference and retrieve the item quickly without having to search the FAQ library.
- `PRIVATE` Declares an FAQ item private or public. The possible values are `Y` or `N`. The values are case-sensitive.

```
'<WAPI>  
<TENANT>AcmeJets</TENANT>  
<USERNAME>wapi</USERNAME>
```

```

<PASSWORD>wapi1</PASSWORD>
<COMMAND OBJECT="FAQ" ACTION="Add">
<FAQCATID>1</FAQCATID>
<QUESTION>What types of private jets are available for leasing?</QUESTION>
<ANSWER> Private jets are typically categorized into three different size classes
- light jets, mid-sized jets and heavy jets. Light Jets can seat up to 4
passengers, mid-sized jets seat up to 8, while the heavy jets seat up to 12
passengers.</ANSWER>
<SHORTCUT>Jettype1</SHORTCUT>
<PRIVATE>Y</PRIVATE>
</COMMAND>
</WAPI> '

```

## Summary of Mandatory and Optional Parameters

The following table summarizes the mandatory and optional Add parameters for the FAQ object.

Parameter	Mandatory/Optional	Format
FAQCATID	M	Internal ID of the FAQ category. This ID is returned after an Add or Get action on an FAQCategory object.
QUESTION	M	<ul style="list-style-type: none"> <li>■ Any character</li> <li>■ Maximum: 250</li> <li>■ Invalid: whitespace-only</li> </ul>
ANSWER	M	<ul style="list-style-type: none"> <li>■ Any character</li> <li>■ Maximum: 4000</li> <li>■ Invalid: whitespace-only</li> </ul>
SHORTCUT	O	<ul style="list-style-type: none"> <li>■ Any character</li> <li>■ Maximum: 20</li> <li>■ Invalid: whitespace-only</li> </ul>
PRIVATE	O	<ul style="list-style-type: none"> <li>■ Possible values are Y or N.</li> <li>■ The default value is N.</li> <li>■ Values are case-sensitive.</li> </ul>
LANGUAGE	O	Possible values are as follows. The default value is en.

Parameter	Mandatory/Optional	Format
		<ul style="list-style-type: none"> <li>■ en</li> <li>■ ru</li> <li>■ de</li> <li>■ ja</li> <li>■ es</li> <li>■ fr</li> <li>■ pt</li> <li>■ it</li> <li>■ pl</li> <li>■ hr</li> <li>■ hi</li> <li>■ nl</li> <li>■ ar</li> <li>■ da</li> <li>■ ko</li> <li>■ no</li> <li>■ sv</li> <li>■ vi</li> <li>■ cy</li> <li>■ th</li> <li>■ zh-CN</li> <li>■ zh-TW</li> </ul>

## Reply from Local CRM to Add FAQ

The following example illustrates a reply to an Add FAQ request in the 8x8 Local CRM.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<WAPI>
<REPLY STATUS_CODE="0" ERROR_STR="" STATUS="0">
<FAQID>1</FAQID>
```

```
</REPLY>
</WAPI>
```

See [Error Cases](#) for a list of errors to an unsuccessful reply.

## Modify FAQ

The `Modify FAQ` action allows you to update an existing FAQ item in an FAQ category in the 8x8 Local CRM.

- The `Modify` keyword is not case-sensitive.
- Any parameter not supplied in a `Modify` request retains the previous value.
- `FAQID` uniquely identifies the FAQ item to be updated and hence is mandatory.

The following example illustrates a `Modify` request for an FAQ.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="FAQ" ACTION="Modify">
<FAQID>1</FAQID>
<QUESTION>What types of private jets are available for leasing?</QUESTION>
<ANSWER>Private jets are typically categorized into three different size classes
light jets, mid-sized jets and heavy jets. Light Jets can seat up to 4 passengers,
mid-sized jets seat up to 8, while the heavy jets seat up to 12
passengers.</ANSWER>
</COMMAND>
</WAPI>'
```

## Summary of Mandatory and Optional Parameters

The following table summarizes the mandatory and optional `Modify` parameters for the `FAQ` object.

Parameter	Mandatory/Optional	Format
FAQID	M	Internal ID of the FAQ. This ID is returned after an <code>Add</code> or <code>Get</code> action on an <code>FAQ</code> object.
QUESTION	O	<ul style="list-style-type: none"> <li>■ Any character</li> <li>■ Maximum characters: 250</li> <li>■ Invalid: whitespace-only</li> </ul>

Parameter	Mandatory/Optional	Format
		 <b>Note:</b> If the field is not specified, the database field is created empty.
ANSWER	O	<ul style="list-style-type: none"> <li>■ Any character</li> <li>■ Maximum characters: 4000</li> <li>■ Invalid: whitespace-only</li> </ul>  <b>Note:</b> If the field is not specified, the database field is created empty.
SHORTCUT	O	<ul style="list-style-type: none"> <li>■ Any character</li> <li>■ Maximum characters: 20</li> <li>■ Invalid: whitespace-only</li> </ul>
PRIVATE	O	<ul style="list-style-type: none"> <li>■ Possible values: Y or N. The default value is N.</li> <li>■ Values are not case-sensitive.</li> </ul>

## Reply from Local CRM to Modify FAQ

The following example illustrates a reply to a `Modify FAQ` request in the 8x8 Local CRM.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<WAPI>
<REPLY ERROR_CODE="0" ERROR_STR="" STATUS="0">
<FAQID>1</FAQID>
</REPLY>
</WAPI>
```

See [Error Cases](#) for a list of errors to an unsuccessful reply.

## Get FAQ

The `Get FAQ` action lists all FAQs in a specified FAQ category in the 8x8 Local CRM. It also enables you to search for all FAQs containing a specified substring within the `QUESTION` or `ANSWER` parameters.

- The `Get` keyword is not case-sensitive.
- Command object only needs to be passed once, even when searching for multiple parameters.
- Parameter values being searched are case-sensitive.

Get options:

- List all FAQs in an FAQ category
- List all FAQs containing a specified substring

## Tags to Pair for Range Queries

- `FromCreateDate / ToCreateDate`

## Examples of Get FAQ Operations

The following example lists all FAQs in FAQ category 1.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="FAQ" ACTION="Get">
<FAQCATID>1</FAQCATID>
</COMMAND>
</WAPI>'
```

The following example lists FAQs that use wildcard characters to return FAQs including the `light jets` substring.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="FAQ" ACTION="get">
<STRING>%light jets%</STRING>
</COMMAND>
</WAPI>'
```



**Note:** Use `<STRING>%<STRING>` to fetch all FAQs in the FAQ library.

The following example illustrates a search for FAQs created on a specific day.

```
'<WAPI>
```

```
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123<PASSWORD>
<COMMAND OBJECT="FAQ" ACTION="GET">
<FAQCATID>67890</FAQCATID>
<FROMCREATEDATE>06202012</FROMCREATEDATE>
<TOCREATEDATE>06252012</TOCREATEDATE>
</WAPI>
```

## List of Parameters

The following table summarizes the `Get` parameters for the `FAQ` object.

You can search for `FAQ` based on `FAQCATID` and `STRING` parameters only. The `STRING` parameter looks for a specified string in the `QUESTION` or `ANSWER` fields.

Parameter	Format
FAQCATID	Internal ID of the FAQ. This ID is returned after an <code>Add</code> or <code>GET</code> action on an FAQ object.
STRING	Any character.
FROMCREATEDATE	The format for this attribute is <code>MMDDYYYY</code> , where <code>MM</code> is the month, <code>DD</code> the day, and <code>YYYY</code> the year.
TOCREATEDATE	The format for this attribute is <code>MMDDYYYY</code> , where <code>MM</code> is the month, <code>DD</code> the day, and <code>YYYY</code> the year.

## Reply from Local CRM to Get FAQ

The following example illustrates a reply to a `Get FAQ` request in the 8x8 Local CRM.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
-<WAPI>
-<REPLY ERROR_CODE="0" ERROR_STR="" STATUS="0">
-<ITEM>
<FAQCATID>1</FAQCATID>
<FAQID>3</FAQID>
<QUESTION>What types of private jets are available for leasing?</QUESTION>
<ANSWER>Private jets are typically categorized into three different size classes
i.e. light jets, mid-sized jets and heavy jets. Light Jets can seat up to 4
passengers, mid-sized jets seat up to 8, while the heavy jets seat up to 12
```

```
passengers.</ANSWER>
<SHORTCUT>Jettypel</SHORTCUT>
<PRIVATE>Y</PRIVATE>
</ITEM>
<ITEM>
. . . .
</ITEM></REPLY></WAPI>
```

See [Error Cases](#) for a list of errors to an unsuccessful reply.

## Delete FAQ

The `Delete FAQ` action allows you to delete an FAQ item from an FAQ category in the 8x8 Local CRM.

- the `Delete` keyword is not case-sensitive.
- `FAQID` is mandatory to refer to an FAQ item to be deleted.

The following example illustrates a `Delete` request for an FAQ.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="FAQ" ACTION="Delete">
<FAQID>45</FAQID>
</COMMAND>
</WAPI>'
```

## Summary of Parameters

The following table summarizes the mandatory and optional `Delete` parameters for the `FAQ` object.

Parameter	Mandatory/Optional	Format
FAQID	M	Internal ID of the FAQ. This ID is returned after an <code>Add</code> or a <code>Get</code> action on an FAQ object.

## Reply from Local CRM to Delete FAQ

The following example illustrates a reply to a `Delete FAQ` request in the 8x8 Local CRM.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<WAPI>  
<REPLY ERROR_CODE="0" ERROR_STR="" STATUS="0"/>  
</WAPI>
```

See [Error Cases](#) for a list of errors to an unsuccessful reply.

## Manipulate Custom Field Objects and Values

This section outlines the actions that can be performed on a Custom Field object and custom field values in the 8x8 Local CRM, and all the parameters that are used in both the request and the reply.

- For custom field objects, the CRM API supports the `Get` operation.
- For custom field values, the CRM API supports the `Get` and `Modify` operations.

You can perform the following actions on custom field objects and values:

- [Get Custom Fields](#)
- [Modify Custom Field Values](#)

### Get Custom Field

The `Get Customfield` action allows you to get the different values of an existing picklist custom field in the 8x8 Local CRM.

- The `Get` keyword is not case-sensitive.
- Command object only needs to be passed once, even when searching for multiple parameters.
- Parameter values being searched are case-sensitive.
- The contact center administrator uses Configuration Manager to define custom fields and values.

The following example illustrates a `Get` request for a custom field. In this example, the picklist values of the custom field called `PrivacyLevel` are returned.

```
'<WAPI>  
<TENANT>AcmeJets</TENANT>  
<USERNAME>johnd</USERNAME>  
<PASSWORD>john123</PASSWORD>
```

```
<COMMAND OBJECT="Customfield" ACTION="GET"> <NAME>PrivacyLevel</NAME>
</COMMAND>
</WAPI> '
```

## List of Parameters

The following table summarizes the `Get` parameters for the `Customfield` object.

Parameter	Format
Custom Field Name	<ul style="list-style-type: none"> <li>Name of the picklist custom field.</li> <li>The contact center administrator uses Configuration Manager to define custom fields and values.</li> </ul>

## Reply from Local CRM to Get Custom Field

The following example illustrates a reply to a `Get Customfield` request in the 8x8 Local CRM.

If the parameter given in the `Get` request does not correspond to a picklist, the Local CRM returns an empty response.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
-<WAPI>-<REPLY ERROR_CODE="0" ERROR_STR="" STATUS="0">
-<ITEM><VALUE>No Solicitation</VALUE></ITEM>
-<ITEM><VALUE>Unknown</VALUE></ITEM>
-<ITEM><VALUE>Neutral</VALUE></ITEM>
</REPLY></WAPI>
```

See [Error Cases](#) for a list of errors to an unsuccessful reply.

## Modify Custom Field Values

This action allows a tenant to modify values for the custom fields in the 8x8 Local CRM added by the tenant administrator in Configuration Manager. Custom fields can be added to the Customer, Case, and Follow-up objects in Agent Console.

- Custom fields can be added to the Customer, Case, and Follow-up objects of CRM API.
- Only custom field values can be modified, not custom field objects.
- The `Modify` keyword is not case-sensitive.
- Any parameter not supplied in a `Modify` request retains the previous value.
- Custom fields can vary for each tenant.

- When submitting a `Modify` request for a Customer or Case object, you must ensure the mandatory parameters for this type of object are supplied.

A list of the mandatory and optional parameters for a Customer or Case object can be found in [Manipulate Customer Objects](#) and [Manipulate Case Objects](#).

The following example modifies the custom field values `CF01_PICKLIST.NAME` and `CF02.NAME`.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="Customer" ACTION="Modify">
<ACCOUNTNUM>1000019</ACCOUNTNUM>
<FIRSTNAME>Bill</FIRSTNAME>
<LASTNAME>Gates</LASTNAME>
<CF01_PICKLIST.NAME>value1</CF01_PICKLIST.NAME>
<CF02.NAME>value2</CF02.NAME>
</COMMAND>
</WAPI>'
```

## List of Mandatory and Optional Parameters

The following table summarizes the mandatory and optional `Modify` parameters for custom field values. Any parameter not supplied in `Modify` retains its previous value.

Parameter	Mandatory/Optional	Format
ACCOUNTNUM	M	Customer account number to be modified as returned by a <code>Get</code> request.
FIRSTNAME	O	Maximum characters: 30
LASTNAME	O	Maximum characters: 30
CF01_PICKLIST.NAME	O	Custom fields and their parameter names are added by the administrator of the tenant. The values must be valid (not deleted) and match the values as defined in Configuration Manager. The tenant administrator defines the custom field values.
CF02.NAME	O	Custom fields and their parameter names are added by the administrator of the tenant. The values must be valid (not deleted) as defined in Configuration Manager. The values passed must be in a text or number format.

## Reply from the Local CRM to Modify Custom Field Values

The following example illustrates a reply to a `Modify` request for custom field values in the 8x8 Local CRM.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<WAPI>
<REPLY STATUS="0" ERROR_STR="">
<ITEM>
<ACCOUNTNUM>100019</ACCOUNTNUM>
</ITEM>
</REPLY>
</WAPI>
```

See [Error Cases](#) for a list of errors to an unsuccessful reply.

## List Attribute Values

This section provides examples on how to list attributes, such as Country, in the 8x8 Local CRM.

### List COUNTRY Attributes

The following example illustrates a request to list the country attribute.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="COUNTRY" ACTION="LIST">
</COMMAND>
</WAPI>'
```

### Reply from Local CRM to List COUNTRY

The following example illustrates a reply to a request to list the country attribute.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<WAPI>
<REPLY ERROR_CODE="0" ERROR_STR="" STATUS="0">
<ITEM>
<COUNTRY>Afghanistan</COUNTRY>
</ITEM> <ITEM>
<COUNTRY>Albania</COUNTRY>
</ITEM> <ITEM>
<COUNTRY>Algeria</COUNTRY>
```

```
</ITEM> <ITEM> ...  
</ITEM> </REPLY> </WAPI>
```

See **Error Cases** for a list of errors to an unsuccessful reply.

## List Tenant Configuration Attributes

This section describes how to retrieve tenant configuration attributes, such as groups of agents and agent names, in the 8x8 Local CRM.

You can perform the following actions on the List Tenant Configuration Attributes:

- List Agent GROUPS
- List AGENTS
- List FAQ Categories

### List Agent GROUPS

The following example illustrates a request to list the current agent groups of the tenant in the 8x8 Local CRM.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="GROUPS" ACTION="LIST">
</COMMAND>
</WAPI>'
```

### Reply to List Agent GROUPS

The following example illustrates a request to list agent groups.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<WAPI>
<REPLY STATUS="0" ERROR_STR="">
```

```

<ITEM> <AGENTID>bobp</AGENT ID>
<AGENT>bob power< /AGENT>
</ITEM>
<ITEM>
<AGENTID>johnd</AGENTID>
<AGENT>john doe</AGENT>
</ITEM>
<ITEM> ...
</ITEM>
</REPLY>
</WAPI>

```

See [Error Cases](#) for a list of errors to an unsuccessful reply.

## List AGENTS

The following example illustrates a request to list agents in the 8x8 Local CRM.

```

'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="AGENTS" ACTION="LIST">
</COMMAND>
</WAPI>'

```

## Reply to List AGENTS

The following example illustrates the reply to the requested list of agents.

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<WAPI>
<REPLY ERROR_CODE="0" ERROR_STR="" STATUS="0">
<ITEM>
<AGENTID>bobp</AGENT ID>
<AGENT>bob power</AGENT>
</ITEM>
<ITEM>
<AGENTID>johnd</AGENTID>
<AGENT>john doe</AGENT>
</ITEM>

```

```
<ITEM>
...
</ITEM>
</REPLY>
</WAPI>
```

See [Error Cases](#) for a list of errors to an unsuccessful reply.

## List FAQ Categories

The following example illustrates a list of FAQ categories defined in the tenant in the 8x8 Local CRM.

```
'<WAPI>
<TENANT>AcmeJets</TENANT>
<USERNAME>johnd</USERNAME>
<PASSWORD>john123</PASSWORD>
<COMMAND OBJECT="FAQCATEGORY" ACTION="LIST">
</COMMAND>
</WAPI>'
```

## Reply to list FAQ Category

The following example illustrates the reply to a list of FAQ categories.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
-<WAPI>-<REPLY ERROR_CODE="0" ERROR_STR="" STATUS="0">
-<ITEM><FAQCATID>2</FAQCATID>
<FAQCATEGORY>Safety</FAQCATEGORY>
</ITEM>
-<ITEM><FAQCATID>4</FAQCATID>
<FAQCATEGORY>legal</FAQCATEGORY></ITEM>
-<ITEM>
FAQCATID>1</FAQCATID>
<FAQCATEGORY>Services</FAQCATEGORY>
</ITEM><
/REPLY>
</WAPI>
```

See [Error Cases](#) for a list of errors to an unsuccessful reply.

## Historical Statistics

Starting with the 3.0.0 release, the 8x8 CRM API no longer supports retrieval of Historical Statistics.

To export historical statistics to Microsoft Excel, go to **Supervisor Console > Reporting > Historical Reports**.

## Error Cases

The following table lists the error messages returned in response to improperly-formed queries to the 8x8 Local CRM.

Error Message	Error Definition
Action 'Action Name' Not Valid for the Object 'Object Name'	The action value for the given object is not valid.
Action Value not passed	The action value is not passed in the XML string.
Action Value Not Valid	The action value in the XML string is not valid.
CASENUM 'case number' is an EMAIL Case. The Description of an Email Case cannot be modified	A case is created via an email to tenant. The description of email cases cannot be modified.
Duplicate Tag 'Alias Name'	Duplicate tags are passed for the <code>ADD</code> , <code>MODIFY</code> , or <code>DELETE</code> actions.
Duplicate value 'Value passed' of 'Alias Name'	The value inserted must be unique and the value passed already exists in the database.
FAQCATID not passed	An FAQ category is deleted and the <code>FAQCATID</code> tag is not passed.
FAQ Category cannot be deleted when FAQ exists for the given category	An FAQ category cannot be deleted when an FAQ exists in the given category.
Inconsistent tags. AUTOPASSWORD is TRUE and PASSWORD is also provided	A password is automatically generated, and therefore, there is a conflict when the password is also provided.

Error Message	Error Definition
Input Error: The tag 'tag name' not passed	An expected tag is not passed.
Internal Error	Internal error; please contact customer support.
No tags are found	No tags are passed in the ADD, MODIFY, or DELETE action.
None of the possible mandatory tags or its values is found. Please check the documentation and pass at least one mandatory tag	A mandatory tag is not passed.
Not a valid XML string	The XML string is not valid.
Not a valid XML string. Error after 'Tag Name'	The XML string is not valid; error after TagName.
OBJECT name 'Object Name' not valid	The object name is not valid.
OBJECT 'Object Name' is not a valid object	The value of the object in the XML string is not valid.
OBJECT value not passed	In the XML string passed, there is no object value.
Regular expression not allowed for the alias 'Alias Name'	Some tag values can be passed with % for the GET action. Other tag values cannot be passed.
TAGS Incompatible. ASSIGNEDTO passed but no STATUS is passed. By default STATUS value is label with USABILITY ='UNA'	Must pass the ASSIGNEDTO and STATUS parameters. By default, the status is set to Unassigned.
TAGS Incompatible. ASSIGNEDTO passed but STATUS is UNASSIGNED	The ASSIGNEDTO parameter passed, but the status is set to Unassigned.
TAGS Incompatible. ASSIGNEDTO passed with STATUS = Unassigned	A case cannot be assigned to an agent when the status of the case is Unassigned.
TAGS Incompatible. EMPTY ASSIGNEDTO passed with STATUS = OPEN	A case must be assigned to an agent if the status is Open.
TAGS Incompatible. The usability of STATUS cannot be OPN, when the Case is not assigned to some agent	The status of a case cannot be Open when the case is not assigned to an agent. In addition, it

Error Message	Error Definition
	is possible that the ASSIGNEDTO tag was not passed.
TAGS Incompatible. The usability of STATUS cannot be UNA, when the Case is assigned to some agent	The status of the case is changed to Unassigned without passing ASSIGNEDTO tag.
TAGS Incompatible. The usability of STATUS is OPN, but no ASSIGNEDTO passed	The ASSIGNEDTO parameter needs to be passed in XML string.
TAGS Incompatible. The value of STATUS cannot be Unassigned, when the Case is assigned to some agent	The status cannot be Unassigned when the case is assigned to an agent.
The ALIAS 'AUTOPASSWD' is not given	The AUTOPASSWD tag is not passed in the ADD customer request.
The ALIAS 'AUTOPASSWD' is FALSE, but ALIAS 'PASSWORD' is not provided	If AUTOPASSWD is false, the alias PASSWORD must be provided.
The length of the value for the alias 'Alias Name' doesn't confirm to the criterion. The length of the value passed is 'length of the value', min length allowed is 'min length allowed' and max length allowed is 'max length allowed'	Length mismatch; consult documentation.
The length of the value of the tag 'Alias Name' is 'Passed Length' which is greater than the allowed value 'Allowed Max Value'	The custom field length exceeds the maximum value.
The mandatory tag 'Alias Name' is not passed or the value for this tag is not passed	The case number is not valid.
The passed caseid 'Case Number' is not valid	The value passed for the given alias does not meet the specified criteria.
The passed value 'Alias Value' of the alias 'Alias Name' doesn't meet the specified criterion	The tag is not a valid action for the object type selected.
The tag 'Alias Name' is not a valid tag for the 'Action' on the 'object'	The custom field status is deleted.
The STATUS of the tag 'Alias Name' is 'DEL'. No action allowed on this tag	A case is assigned to a deleted agent and group.
The STATUS of the value 'Value of the ASSIGNEDTO' of the	The value passed for ASSIGNEDTO is

Error Message	Error Definition
ASSIGNEDTO tag is deleted	deleted.
The STATUS of the value 'Value Passed' of the ASSIGNEDTO tag is deleted	The custom field of the Date data type is not in the correct format.
The value of the passed alias 'Alias Name' is not in the correct format (mmddyyyy)	The case number is not valid.
The value 'Passed Value' of the alias 'Alias Name' is not valid	The value passed for TagName is not valid.
The value 'value' for the tag 'tag name' is not valid	The value for TagName is not valid.
The value = 'Value passed' of the tag 'Tag Name' is not valid	The XML string is not correct.
XML string not correct. Error at line 'line number': 'Error Message'	The value for TagName is not valid.