Aspire Call Center Technologies

Integration and APIs Guide

Revision 3.1
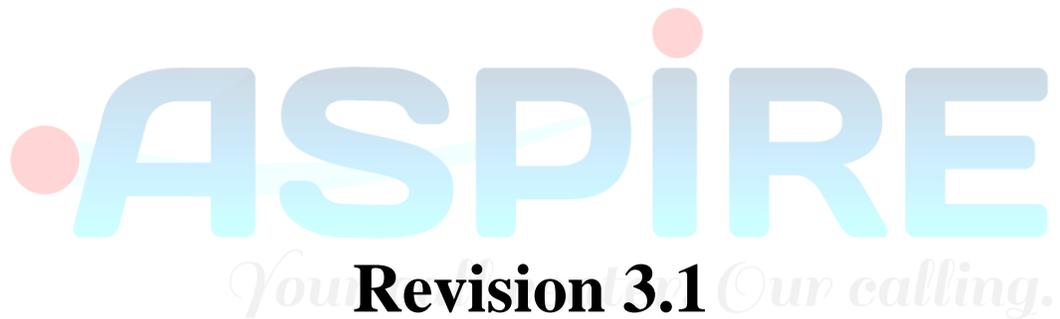
August 2018

Table of Contents

Scope

The purpose of this document is to provide a detailed guide on how to integrate the Optimus Pro Call Center solution with external systems and applications.

This Optimus Integration Guide refers to Optimus version 5.X and no other versions. For other versions please refer to the correct Integration Guide.

Clarification: All male-oriented references in this document are intended for both male and female readers and users.

Concepts and Abbreviations

- **Optimus Server**: The PC server where Optimus Call Center is installed, using Microsoft Windows Server OS software.

- **TAPI**: Telephony API (Application Programming Interface): telephony protocol supplied by Microsoft as part of the Windows OS and used by PBX vendors for application access.

- **TSP / TAPI Server**: TAPI Service Provider: The software developed by the PBX vendor using the TAPI protocol, to enable application access to the PBX. For Optimus, the vendor TSP is installed on the Optimus Server.

- **SIP:** A communication protocol, part of the Voice Over IP method of using a network to transfer telephone calls.

- **Optimus Main Device (OMD)**: Optimus uses a smart extension as part of the PBX interface. This extension should be a 30-key telephone device, and DSS units are added as needed for call center with more than one PRI (or E1) interfaces, with about one DSS needed per PRI interface.

- **IP Extension:** An extension of the PBX supporting a smart phone set connected using the TCP/IP protocol, via either a proprietary interface or SIP. An IP extension has many features available through the PBX, which a regular extension does not.

- **Smart Extension**: An extension of the PBX supporting a smart, digital phone set. A smart extension has many features available through the PBX, which a regular extension does not.

- **Regular Extension**: An extension of the PBX supporting a regular, analog phone set. A regular extension usually supports only simple telephony functions, not unlike a traditional home telephone.

- **TN**: A unique, physical identification of an extension in the PBX.

- **CLAN**: Customer LAN. The main local network of the customer.

- **PLAN**: PBX LAN. A network segment where the PBX and the Optimus Server are located. See Optimus Architecture diagram (Appendix A) for more details.

- **PBX**: Private Branch Exchange.

- **IP Address**: TCP/IP network protocol assigns a unique IP (Internet Protocol) address for each computer or device (such as the PBX or IP Extension).

- **IP Port Number**: TCP/IP network protocol uses different port numbers for different types of communications. For example, the HTTP protocol used for web browsing uses TCP/IP port 80.

- **DNIS**: Dialed Number Identification Service provided for each call and designates the number that was dialed in order to reach the call's destination.

- **CLID / ANI**: Caller ID provided for each call and identifies the caller's number (unless the number is blocked purposefully).

- **ATTACHDATA**: Any additional data (besides DNIS and CLID) attached to the call. For example: Customer ID as entered by the caller using the phone's buttons.

- **ODBC**: Open Data Base Connectivity – a Windows standard for database access and connections. Supported by most commercial database software.

- **WAVE GSM**: the audio file format supported by Optimus for music on hold and waiting announcements.

Call Center Concepts

- **Offered Calls:** Calls that were presented in a queue – and either answered, abandoned or overflowed.
- **Returned Calls (from agent):** Calls that were returned to the queue due to an agent not answering.
- **Answered Calls:** Calls that were answered by an agent.
- **Abandoned Calls:** Calls that were disconnected by the caller while waiting for an agent to answer.
- **Overflowed Calls:** Calls can be overflowed between queues according to several types of rules, and see the Queue Overflow Rules section for more information.
- **Backflowed (Returned) Calls:** Calls that were overflowed to a queue but kept on waiting in the original queue (an option in Optimus) and then were answered by an agent in the original queue.
- **Inflowed Calls:** A call that was overflowed from queue X to queue Y will be presented as an *inflowed call* in queue Y.
- **Search Expansion:** A call overflowing from queue X to queue Y may continue waiting simultaneously on queue X as well.
- **Calls Ended in IVR:** Calls that followed the Call Me Back script through, and so didn't have to be returned to the queue.
- **Calls Routed to Default Extension:** Calls that were routed to the system's default extension because no matching routing rule was found. This is not a proper state. Please refer to section 1.3.2 for more information.
- **Queue:** A queue is where waiting calls wait for an agent to become available and handle them. During waiting time in the queue, callers hear the waiting music from the PBX or, if available, dynamic IVR messages. Calls can overflow from one queue to another queue.
- **Skill:** a skill represents the agent's ability to handle a specific type of calls.

- **Skill Level:** The agent's skill level may have an impact on the agent's priority in receiving and handling calls.

- **Agent Group:** An Agent Group is a collection of agents, which for all purposes function as a single group. By using agent groups, supervisors can assign a whole group of agents to handle calls from a queue with a single action. An agent can be a member of an unlimited number of groups, and an unlimited number of agent groups are possible in Optimus.

- **ACD Call:** a call that was presented to an agent through a queue, as a result of Optimus routing.

- **Internal Incoming Call:** a call that was presented to an agent through a direct dial from within the organization, and not as a result of Optimus routing.

- **External Incoming Call:** a call that was presented to an agent through a direct dial from outside the organization, and not as a result of Optimus routing.

- **Data type**: **Cumulative** - The statistics are accumulated over a specified period of time (for example, the number of calls answered during an interval). This value determines the length of the period of time.

- **Data type**: **State** - The value depends on the instantaneous state of the system (for example, the state of an agent at a given time or the number of waiting calls *right now*).

- **IVR:** A module used to communicate with the caller. For example, to have the caller type in an ID number, or announce information such as queue position of the call.

- **Dialer:** A module used to automatically call numbers. The dialer saves precious time for agents, because only answered calls are transferred to the agent.

- **Call Back:** When this option is enabled, waiting callers are presented with the possibility of hanging up the call and the system will call them back automatically.

1. Integration Components

Integration with Optimus Pro Call Center is comprised of several components, each has its own functionality and purpose. It is not necessary to implement all of these components and the integrator can choose what functionality is required and therefore which components should be used.

Integration components for Optimus Pro Call Center include:

- **Telephony OCX** – is a Microsoft Visual Basic 6.0 OCX used to implement telephony functionality inside CRM or other software. Use this OCX if your software is not web-based or if you require full telephony functionality accessible from your software.

- **Web Screen Pop-Up** – is a configurable parameter within Optimus Pro where you can choose which URL to pop up when a call is incoming to the agent, as well as configure parameters to be transferred. Use this method if your software is web-based and you require only screen pop up functionality.

- **External Database Integration** – is available via an ODBC connection to any database which supports such a connection and SQL queries. Use this integration to implement call routing rules based on external information.

- **Optimus Database Integration** – is available via an ODBC connection to the Optimus Pro database, which is a Microsoft SQL Server Express Edition database 2005 / 2008 / 2012 (depending on your Optimus version). Database integration enables access to aggregate statistical tables and stored functions and is used for integration with BI systems.

- **Dialer Integration** – is comprised of several database stored procedures and web services enabling record adding and deleting from the dialer. There is also a RESTful interface available for dialer target manipulation. Use this API to control some of the Optimus Pro Call Center's dialer functionality from within your CRM software.

- **Recording Module Integration** – is comprised of database stored procedures providing information regarding recorded files. Use this API to allow direct access to recorded files from your software and not through the Optimus Pro Call Center management GUI.

- **Single Sign On Integration –** is a web service allowing external applications to change the status of an agent – login, logout and change the activity code of an agent

2. Telephony OCX

2.1 General Information

- The ActiveX is developed with Microsoft Visual Basic 6.0.
- The connection method between the ActiveX and Optimus server is TCP/IP.
- The ActiveX can only be used to receive information from Optimus about one agent. As a result, you need to implement the OCX in such a way that it is part of your software and runs on all workstations, providing functionality to each workstation and its corresponding agent.
- The information received can be one of the following:
  - o Incoming call notification
  - o Real time statistics information about the queues the agent is assigned to
  - o Real time statistics information about the agent performance

2.2 Methods

1. **Name**

   InitializeActiveX

   Purpose

   This method is used to initialize the ActiveX and register it with Optimus. This method must be called in order to receive information for the client.

   Parameters

   serverIP: The IP address of the Optimus server.

   serverPort: The port number which is used by the ActiveX to interact (receive agent state information and real time data) with the Optimus server.

   The port is configured in Optimus web Management, under the Configuration section in Agent toolbar Configuration (Agent Toolbar IP Port Number).

   userName: The username that the ActiveX receives information for. This user must be logged in for Optimus in order for the ActiveX to receive information.

2.3 Events

1. **Name**

   OnScreenPopUpData

   Purpose

   This event is raised for each incoming call to the agent. The event is raised when the call is offered to the agent and before the agent answers the call.

   Parameters

   clid: The Caller ID of the caller, Provided for each call and identifies the caller's number (unless the number is blocked purposefully).

   dnis: Dialed Number Identification Service, provided for each call and designates the number that was dialed in order to reach the call's destination.

   attachData: Any additional data (besides DNIS and CLID) attached to call. For example: Customer ID as entered by the caller using the phone's buttons.

2. **Name**

   OnQueuesInfoData

   Purpose

   This event is provides either real time statistics information on the queues that the agent is assigned to or real time performance information about the agent, based on a configuration in Optimus Web Management under the Users & Agents section in Advanced screen (Toolbar).

   The information is received every 5 seconds and is based on a configuration in Optimus Web Management under the Configuration section in Agent toolbar Configuration (Agent Toolbar Update Interval (seconds)).

   Parameters

   data: Contains the real time statistics.

If the toolbar type is configured to Queues then the data contains information about the queues that the agent is assigned to, separated with | (pipe) character. The data starts with the characters "3|". The information provided is:

Queue Code: The code of the queue the information applies to

Queue name: The name of the queue the information applies to

Average Waiting Time: The average time a call has waited in the queue before being answered.

Maximum Waiting Time: The maximal time a waiting call has waited in the queue before being answered.

Waiting Calls: The number of calls currently waiting in the queue.

Offered Calls: The number of calls offered to the queue, i.e. the calls that were routed to the queue as a result of a routing rule. Please note that calls that were overflowed into this queue are not counted in this field.

Answered Calls: The number of calls answered in the queue.

Returned Calls: The total number of calls returned from the agents, plus the total number of calls returned from the main system default number, plus the total number of calls returned from a route number, plus the total number of calls that were in-flowed to this queue from other queues.

Abandoned Calls: The number of calls that were abandoned by the caller while waiting in the queue.

Transferred to Main System Default Number: The number of calls that were automatically transferred by Optimus to the systems' default number.

For example, if the agent is assigned to 2 queues the data structure is:

3|Queue Code| Queue name| Average Waiting Time| Maximum Waiting Time| Waiting Calls| Offered Calls| Answered Calls| Returned Calls| Abandoned Calls| Transferred to Main System Default Number|

If the toolbar type is configured to Agent then the data contains performance information about the agent, separated with | (pipe) character. The data starts with the characters "4|". The information provided is:

Agent Code: The code of the agent.

Offered Calls: The number of calls offered to the agent.

Returned Calls: The total number of calls returned from the agent.

Answered Calls: The number of ACD calls answered by the agent.

Direct Incoming Calls: The number of direct incoming calls answered by the agent.

External Outgoing Calls: The number of external outgoing calls dialed by the agent.

Internal Outgoing Calls: The number of internal outgoing calls dialed by the agent.

Waiting Calls: The number of calls currently waiting in all the queues that the agent is assigned to.

Not Ready Time: The total time the agent spent in Not Ready Status.

3. **Name**

    OnCall

Purpose

This event is raised for each incoming ACD call to the agent or for each outbound call initiated by the agent. The event is raised for the first time when the call is answered by the agent (for an ACD call) or by the called party (for an outbound call), and for the second time when the call ends.

Parameters

CallId (string): The Call ID of the call, a unique identifier across the Optimus platform.

isAnswer (int): 1 – First raising of the event (call starts). 0 – Second raising (call ends)

isRecorded (int): Is the call being recorded.

4. Web Screen Pop-Up

Optimus uses URL pop ups to display call information to the agent this can include Caller ID, Dialed Number user defined attach data and other information. This is done via a user chosen URL. Combining this method to use MS-Windows and browser features, Optimus can pop up web-based software, open miscellaneous files (text files, office documents and so forth) and even execute other applications.

Optimus supports up to three (3) simultaneous URLs in order to pop-up different systems. Also, different URLs can be defined for different queues. The settings here are the system defaults.

By changing this value Optimus can integrate with any other application and can pop up not only the Optimus built-in screen but any other screen. The URL can contain any of the available System Parameters:

- CLID – Caller Identification.
- DNIS – Dialed Number Identification Service.
- ATTACHDATA – Added information from external sources
- QUEUE – The queue the call came in from,
- WAITING TIME – The amount of time, in seconds, that the caller waited before the call was answered.
- GREETING – The greeting defined in Optimus for this DNIS.
- EXTENSION – The physical phone extension number where the agent is located
- CALLID – Internal Call ID of Optimus
- QUEUECODE – The code of the queue from which the call was routed to the agent.

See section 1.5 in the Optimus manual for more information.

5. External Database Integration

External database integration is used to influence the way Optimus Pro routes calls by information existing in external systems, such as CRM systems.

One of the most common uses is to configure VIP customers. A simple query to the CRM software can return a special priority for some calling customers. When this happens, Optimus will move these calls forward in the queue, according to their priority, and these VIP callers will be answered more quickly.

For a detailed explanation about the functionality of routing rules, and specifically database driven routing rules, please see section 4.3 in the Optimus Pro Call Center manual.

4.1 General Information

Integration is possible with any database that supports the following:
1. A standard ODBC connection
2. SQL queries

A partial list of supported database includes MS-SQL, Oracle, MS-ACCESS, MySQL, Sybase, DB/2, DB/400 and more.

Throughout this section the external software is referred to as the "CRM software". However, it is of course possible to connect to any database, for ERP, Billing, HRM and other software.

The queries themselves are in the SQL standard and should be provided to you by the CRM implementer for the customer.

4.2 Pre-Requisites

1. Optimus must be running (Optimus Engine service) in order to add a database connection.

2. The stored procedure or SQL statement you're about to use should first be checked and be successfully executed in the database.

4.3 Database Connectivity

Optimus supports a connection any SQL database. The connection is performed via a standard connection string. There are two ways to receive information from the customer database:

1. Executing a stored procedure which must be of the following structure:

   - The stored procedure name must be OPTIMUS_GET_ROUTING_INFO.

   - The stored procedure has 5 parameters:

     i. P_DNIS, Input parameter of type Varchar(20): The Dialed Number Identification Service of the call.

     ii. P_CLID, Input parameter of type Varchar(20): The Caller Identification Number of the call.

     iii. P_QUEUE_CODE, Output parameter of type Int: The code of the queue that Optimus should route the call to. The code is displayed in the Optimus Management System in the Queues screen under the Queues Management node.

     iv. P_CALL_PRIORITY, Output parameter of type Int: The priority to assign to the call. Available priorities are 1 to 9 (where 1 is the highest and 9 is the lowest).

     v. P_ATTACH_DATA, Output parameter of type Varchar(200): This data is available to the agent and displayed in the pop up screen in the Message label.

Note: Below is a sample code which is written for Microsoft SQL Server but can easily be converted to any other database:

```
CREATE PROCEDURE dbo.OPTIMUS_GET_ROUTING_INFO
@P_DNIS varchar(20),
@P_CLID varchar(20),
@P_ATTACH_DATA_IN varchar(200),
@P_QUEUE_CODE int OUT,
@P_CALL_PRIORITY int OUT AS,
@P_ATTACH_DATA varchar(200) OUT
```

- The parameters name and type should be exactly as in the code sample.

2. Executing an SQL statement of the following structure:
```
select QUEUE_CODE, PRIORITY, ATTACHDATA
from <ANY_TABLE>
where <FIELD>=$CLID
```

For example:
```
select QUEUE_CODE, PRIORITY, ATTACHDATA
from Customers
where phone_number=$CLID
```

There are 3 variables which can be used in the SQL statement: $CLID, $DNIS and $ATTACHDATA.

1. $CLID is a variable which Optimus replaces with the Caller Identification Number of the call.
2. $DNIS is a variable which Optimus replaces with the Dialed Number Identification Service of the call.

3. $ATTACHDATA is a variable which Optimus replaces with the data of the call (Available only when using Optimus IVR module).

5. Optimus Database Integration

Database integration allows access to aggregate tables and various stored procedures in the Optimus database. These are usually used to present Optimus statistics and information in external reporting systems, such as BI systems.

All aggregative tables provide data in quarter-hour intervals.

Important Note: running complex and resource-consuming queries on the Optimus database is forbidden, as such activity may affect the call center system stability. If you require running long and resource consuming queries you need to export the database to your data warehouse and run these queries there.

5.1 Configuration

1. The Optimus database (MS SQL Server Express) must be up and running
2. An ODBC connection to that database has to be configured on the machine accessing the Optimus database.
3. Optimus database name is **Optimus**. The username is **OPTEXTRNL** and the password is **OPTEXTRNL.**

5.2 General

The Optimus Stat aggregate tables provide aggregative information regarding DNISs, queues, agents, calls, router nodes and other information covering the call center's activity.

In order to conserve database space, only timestamps with relevant information exist in the tables. So, if in a certain date and time no agents were logged in and no activity was recorded, there will be no record in the tables for that date and time.

These tables are the tables used by the SmartCC reports generator.

5.3 Database Tables

5.3.1   Stat.AgentActivityStat

    a.   This table provides aggregate statistics regarding agent usage of activity types (breaks) and times.

    b.   Primary Key: UserCode, ActivityCode, TheDate

    c.   Table fields:

Field Name	Field Type	Field Contents
CalcTS	DateTime	Date and time when this row was calculated
TheDate	DateTime	Date and time of the record data
TheMonth	Nvarchar(7)	Month of the date and time of the record data
TheWeek	Nvarchar(11)	Week of the date and time of the record data
TheDay	Nvarchar(10)	Day of the date and time of the record data
TheHour	Nvarchar(5)	Hour of the date and time of the record data
UserCode	Smallint	User code in the Optimus system. Note that this is the internal code and **not** the agent login username.
ActivityCode	Smallint	Activity Type Code
ActivityTime	Int	The amount of time spent by this user on that activity in that timeframe, in seconds.

5.3.2 Stat.AgentByDeviceStat

    a. This table provides aggregate statistics regarding agent's activity in specific queues and DNISs.

    b. Primary Key: UserCode, DeviceCode, DeviceType, TheDate

    c. Device Type 2 – DNIS, Device Type 3 – Queue.

    d. Table fields:

Field Name	Field Type	Field Contents
CalcTS	DateTime	Date and time when this row was calculated
TheDate	DateTime	Date and time of the record data
TheMonth	Nvarchar(7)	Month of the date and time of the record data
TheWeek	Nvarchar(11)	Week of the date and time of the record data
TheDay	Nvarchar(10)	Day of the date and time of the record data
TheHour	Nvarchar(5)	Hour of the date and time of the record data
UserCode	Smallint	User code in the Optimus system. Note that this is the internal code and **not** the agent login username.
DeviceCode	Smallint	Device Code in the Optimus System. Please note that this field should be used together with DeviceType
DeviceType	Tinyint	DeviceType. 2: DNIS

		3: Queue
ACDOffered	Smallint	Number of offered calls to that agent from that device (queue or DNIS)
ACDAnswered	Smallint	Number of answered calls by that agent from that device
ACDRecalled	Smallint	Number of calls recalled from the agent back to the queue, originating from that device
ACDAbdRinging	Smallint	Number of abandoned calls, whilte ringing at the agent's extension, originating from that device
MinRingTime	Smallint	Minimum ring time (in seconds)
SumRingTime	Smallint	Total ring time (in seconds)
MaxRingTime	Smallint	Maximum ring time (in seconds)
MinACDTalkTime	Smallint	Minimum ACD Talk Time (in seconds)
SumACDTalkTime	Smallint	Total ACD Talk Time (in seconds)
MaxACDTalkTime	Smallint	Maximum ACD Talk Time (in seconds)
HoldCount	Smallint	The number of times calls were placed on hold by that agent, originating from that device

MinHoldTime	Smallint	Minimum Hold time (in seconds)
SumHoldTime	Smallint	Total Hold time (in seconds)
MaxHoldTime	Smallint	Maximum Hold time (in seconds)
MinWrapUpTime	Smallint	Minimum WrapUp time (in seconds)
SumWrapUpTime	Smallint	Total WrapUp time (in seconds)
MaxWrapUpTime	Smallint	Maximum WrapUp time (in seconds)

5.3.3   Stat.AgentStat

    a.   This table provides aggregate statistics regarding all agents in the system.

    b.   Primary Key: UserCode, TheDate

    c.   Usually, Agents with codes -2, -1, 0 and 1 are internal management agents and should be disregarded, unless purposefully used by the customer.

    d.   Table fields:

Field Name	Field Type	Field Contents
CalcTS	DateTime	Date and time when this row was calculated
TheDate	DateTime	Date and time of the record data
TheMonth	Nvarchar(7)	Month of the date and time of the record data
TheWeek	Nvarchar(11)	Week of the date and time of the record data
TheDay	Nvarchar(10)	Day of the date and time of the record data
TheHour	Nvarchar(5)	Hour of the date and time of the record data
UserCode	Smallint	User code in the Optimus system. Note that this is the internal code and **not** the agent login username.
LoginTime	Smallint	Total logged in time for the agent
ReadyTime	Smallint	Total ready time for the agent. Ready time is the time spent

		in the ready (idle) mode or on call center calls.
NotReadyTime	Smallint	Total not ready time for the agent. Not ready time is the time spent in the not ready mode, including all activity types, and also spent on not ready mode without an activity type
IdleTime	Smallint	Total Idle time for the agent. Idle time is the time spent on ready mode but not on call center calls (basically, doing nothing ☺)
ACDOffered	Smallint	The number of call center calls offered to the agent
ACDAnswered	Smallint	The number of call center calls handled by the agent
ACDRecalled	Smallint	The number of call center calls recalled from the agent due to not answering
ACDAbdRinging	Smallint	The number of call center calls abandoned while ringing at the agent's extension
IntDialed	Smallint	The number of internal calls dialed by the agent

IntAnsweredTo	Smallint	The number of internal calls dialed by the agent and answered to
ExtDialed	Smallint	The number of external calls dialed by the agent
ExtAnsweredTo	Smallint	The number of external calls dialed by the agent and answered to
InOffered	Smallint	The number of internal incoming calls offered to the agent
InAnswered	Smallint	The number of internal incoming calls handled by the agent
InAbandoned	Smallint	The number of internal incoming calls abandoned while ringing at the agent's extension
ExOffered	SmallExt	The number of external Incoming calls offered to the agent
ExAnswered	SmallExt	The number of external Incoming calls handled by the agent
ExAbandoned	SmallExt	The number of external Incoming calls abandoned while ringing at the agent's extension

HoldCount	Smallint	The number of times the agent put a call on hold.
MinHoldTime	Smallint	Minimum Hold time (in seconds)
SumHoldTime	Smallint	Total Hold time (in seconds)
MaxHoldTime	Smallint	Maximum Hold time (in seconds)
MinACDCallsTalkTime	Smallint	Minimum ACD Talk Time (in seconds)
SumACDCallsTalkTime	Smallint	Total ACD Talk Time (in seconds)
MaxACDCallsTalkTime	Smallint	Maximum ACD Talk Time (in seconds)
MinInIntCallsTalkTime	Smallint	Minimum incoming internal calls Talk Time (in seconds)
SumInIntCallsTalkTime	Smallint	Total incoming internal calls Talk Time (in seconds)
MaxInIntCallsTalkTime	Smallint	Maximum incoming internal calls Talk Time (in seconds)
MinInExtCallsTalkTime	Smallint	Minimum incoming external calls Talk Time (in seconds)
SumInExtCallsTalkTime	Smallint	Total incoming external calls Talk Time (in seconds)
MaxInExtCallsTalkTime	Smallint	Maximum incoming external calls Talk Time (in seconds)
MinOutIntCallsTalkTime	Smallint	Minimum outgoing internal calls Talk Time (in seconds)

SumOutIntCallsTalkTime	Smallint	Total outgoing internal calls Talk Time (in seconds)
MaxOutIntCallsTalkTime	Smallint	Maximum outgoing internal calls Talk Time (in seconds)
MinOutExtCallsTalkTime	Smallint	Minimum outgoing external calls Talk Time (in seconds)
SumOutExtCallsTalkTime	Smallint	Total outgoing external calls Talk Time (in seconds)
MaxOutExtCallsTalkTime	Smallint	Maximum outgoing external calls Talk Time (in seconds)
MinRingTime	Smallint	Minimum ring time (in seconds)
SumRingTime	Smallint	Total ring time (in seconds)
MaxRingTime	Smallint	Maximum ring time (in seconds)
MinACWTime	Smallint	Minimum WrapUp time (in seconds)
SumACWTime	Smallint	Total WrapUp time (in seconds)
MaxACWTime	Smallint	Maximum WrapUp time (in seconds)
MinOutIntGoodDialTime	Smallint	Minimum time spent dialing to an internal destination for calls that were answered (in seconds)
SumOutIntGoodDialTime	Smallint	Total time spent dialing to an internal destination for calls

		that were answered (in seconds)
MaxOutIntGoodDialTime	Smallint	Maximum time spent dialing to an internal destination for calls that were answered (in seconds)
MinOutIntBadDialTime	Smallint	Minimum time spent dialing to an internal destination for calls that were not answered (in seconds)
SumOutIntBadDialTime	Smallint	Total time spent dialing to an internal destination for calls that were not answered (in seconds)
MaxOutIntBadDialTime	Smallint	Maximum time spent dialing to an internal destination for calls that were not answered (in seconds)
MinOutExtGoodDialTime	Smallint	Minimum time spent dialing to an external destination for calls that were answered (in seconds)
SumOutExtGoodDialTime	Smallint	Total time spent dialing to an external destination for calls that were answered (in seconds)
MaxOutExtGoodDialTime	Smallint	Maximum time spent dialing to an external destination for

		calls that were answered (in seconds)
MinOutExtBadDialTime	Smallint	Minimum time spent dialing to an external destination for calls that were not answered (in seconds)
SumOutExtBadDialTime	Smallint	Total time spent dialing to an external destination for calls that were not answered (in seconds)
MaxOutExtBadDialTime	Smallint	Maximum time spent dialing to an external destination for calls that were not answered (in seconds)
MinIdleOffHookTime	Smallint	Minimum time spent in the "off hook" situation without dialing (in seconds)
SumIdleOffHookTime	Smallint	Total time spent in the "off hook" situation without dialing (in seconds)
MaxIdleOffHookTime	Smallint	Maximum time spent in the "off hook" situation without dialing (in seconds)

Microsoft Partner
Gold Application Development

5.3.4 Stat.CallsStat

    a. This table provides full information regarding all calls in the system. A single row is created for each call.

    b. Primary Key: CallID

    c. Table includes a row for unanswered call attempts (outbound dials) as well.

    d. Table fields:

Field Name	Field Type	Field Contents
CallID	Int	A unique, internal call ID
CallType	tinyint	1: Call Center call 2: Outgoing external call 3: Outgoing internal call 4: Incoming external call 5: Incoming internal call
DNIS	Smallint	For call center calls, the incoming number that was dialed. This number is represented in the DNIS table
CLID	Nvarchar(100)	The number of the caller
OfferedTS	Datetime	The first timestamp when the call entered the system or was initiated by the agent
FirstQueuedTS	Datetime	The first timestamp when the call was queued to a queue by the system. Relevant only to ACD calls
AnsweredTS	Datetime	The exact timestamp when the call was answered. If the call wasn't answered this value will be NULL

AbandonedTS	Datetime	The exact timestamp when a call center call was abandoned. If the call wasn't abandoned this value will be NULL
Transferred2ExtTS	Datetime	The exact timestamp when a call center call was transferred from the system to an external destination (usually due to overflow). If the call wasn't transferred to an external destination this value will be NULL
EndedTS	Datetime	The exact timestamp when a call ends after talking to an agent. If the call wasn't answered (and thus, cannot end) this value will be NULL
InitialQueue	Smallint	The initial queue that a call center call was assigned to. In case of a call overflowing between multiple queues, this field will show the first queue
QueueHops	Tinyint	The amount of times a call "hopped" between queues while waiting, due to overflow rules
Agent	Smallint	The code of the agent handling this call. Represented in the Users table in the database

AgentExtension	Smallint	The extension where the call was handled. Represented in the extensions table in the database
HandledOnQueue	Smallint	The final queue that a call center call was assigned to. In case of a call overflowing between multiple queues, this field will show the final queue
TransferDestination	Nvarchar(100)	In the case of a call transferred to an external destination, the number the call was transferred to
EndType	Tinyint	The manner in which the call ended: 101: Talked and Ended 102: Abandoned 103: Abandoned While Ringing 104: Transferred to External 107: Outbound Call Unanswered 108: Ended In IVR (Callback) 109: Off / On Hook With No Action
AgentHops	Tinyint	The amount of times a call "hopped" between agents while waiting, due to agents not answering the call
Adata1	Nvarchar(100)	The attached data of the call, mostly used for CRM integration. This is information received from

		external systems for screen pop up purposes, such as customer name.
Adata2	Nvarchar(100)	The attached data of the call, mostly used for CRM integration. This is information received from external systems for screen pop up purposes, such as customer name.
Router	Smallint	If the smart router module is active and the call went through a router, this is the ID of the router the call went through.

5.3.5   Stat.DeviceStat

   a.   This table provides aggregate statistics regarding all agents in the system.

   b.   Primary Key: DeviceCode, DeviceType, TheDate

   c.   Table fields:

Field Name	Field Type	Field Contents
CalcTS	DateTime	Date and time when this row was calculated
TheDate	DateTime	Date and time of the record data
TheMonth	Nvarchar(7)	Month of the date and time of the record data
TheWeek	Nvarchar(11)	Week of the date and time of the record data
TheDay	Nvarchar(10)	Day of the date and time of the record data
TheHour	Nvarchar(5)	Hour of the date and time of the record data
DeviceCode	Smallint	Device Code in the Optimus System. Please note that this field should be used together with DeviceType
DeviceType	Tinyint	Device Type 2: DNIS 3: Queue
Offered	Smallint	The number of call center calls offered to that queue or DNIS

Answered	Smallint	The number of call center calls answered by agents, originating from that queue or DNIS
Abandoned	Smallint	The number of call center calls abandoned while waiting in queue, originating from that queue or DNIS
ABDRinging	Smallint	The number of call center calls abandoned while ringing at the agent's extension
OverflowedQueue	Smallint	Number of calls overflowed due to the queue position property
OverflowedTime	Smallint	Number of calls overflowed due to the waiting time property
OverflowedLogin	Smallint	Number of calls overflowed due to the no agents logged in property
Backflowed	Smallint	Number of calls backflowed to the queue
Inflowed	Smallint	Number of calls inflowed to the queue
Tran2Def	Smallint	Number of calls transferred to the Optimus default extension
RetFromAgent	Smallint	Number of calls returned from agent

OfferedCB	Smallint	Number of times the callback option was offered to calls in the queue or DNIS. Note that a call can receive multiple callback offers.
EndedInIVR	Smallint	Number of calls ended in the IVR (requested callback)
MinAnswerTime	Smallint	The minimum time (in seconds) that a call waited before being answered
SumAnswerTime	Smallint	Total waiting time (in seconds) for calls that were answered
MaxAnswerTime	Smallint	The maximum time (in seconds) that a call waited before being answered
MinAbdTime	Smallint	The minimum time (in seconds) that a call waited before being abandoned
SumAbdTime	Smallint	Total waiting time (in seconds) for calls that were abandoned
MaxAbdTime	Smallint	The maximum time (in seconds) that a call waited before being abandoned
MinRingTime	Smallint	Minimum ring time (in seconds)
SumRingTime	Smallint	Total ring time (in seconds)

MaxRingTime	Smallint	Maximum ring time (in seconds)
MinTalkTime	Smallint	Minimum talk time (in seconds)
SumTalkTime	Smallint	Total talk time (in seconds)
MaxTalkTime	Smallint	Maximum talk time (in seconds)
HoldCount	Smallint	The number of times calls were placed on hold. Multiple holds in the same calls are counted as multiple.
MinHoldTime	Smallint	Minimum Hold time (in seconds)
SumHoldTime	Smallint	Total Hold time (in seconds)
MaxHoldTime	Smallint	Maximum Hold time (in seconds)
MinACWTime	Smallint	Minimum WrapUp time (in seconds)
SumACWTime	Smallint	Total WrapUp time (in seconds)
MaxACWTime	Smallint	Maximum WrapUp time (in seconds)
LoginCount	Smallint	The number of times agents assigned to that queue performed the login action
SumLoginTime	Smallint	The total time agents assigned to that queue were logged into the system in. Note that an

		agent can be logged into several queues, and the login time will be counted for each queue independently.
SumReadyTime	Smallint	The total time agents assigned to that queue were in the "Ready" state. Note that an agent can be logged into several queues, and the "Ready" time will be counted for each queue independently.
SumNotReadyTime	Smallint	The total time agents assigned to that queue were in the "Not Ready" state. Note that an agent can be logged into several queues, and the "Not Ready" time will be counted for each queue independently.

5.3.6   Stat.DeviceTimes

a. This table provides aggregate statistics regarding different time types for devices in the system. For example, how many calls were answered within 10 seconds in a specific queue, or how many calls lasted 16 seconds for a certain agent.

b. Primary Key: DeviceCode, DeviceType, TimeType, TimeFrame, TheDate

c. Table fields:

Field Name	Field Type	Field Contents
CalcTS	DateTime	Date and time when this row was calculated
TheDate	DateTime	Date and time of the record data
TheMonth	Nvarchar(7)	Month of the date and time of the record data
TheWeek	Nvarchar(11)	Week of the date and time of the record data
TheDay	Nvarchar(10)	Day of the date and time of the record data
TheHour	Nvarchar(5)	Hour of the date and time of the record data
DeviceCode	Smallint	Device Code in the Optimus System. Please note that this field should be used together with DeviceType
DeviceType	Tinyint	DeviceType. 2: DNIS 3: Queue
TimeType	Tinyint	1: Agent ACD Calls Talk Time

		2: Agent Inbound Internal Talk Time 3: Agent Inbound External Talk Time 4: DNIS Talk Time 5: Queue Talk Time 6: Agent Outbound Internal Talk Time 7: Agent Outbound External Talk Time 10: DNIS Answered Waiting Time 11: DNIS Abandoned Waiting Time 12: Queue Answered Waiting Time 13: Queue Abandoned Waiting Time 20: Extension Inbound Internal Talk Time 21: Extension Inbound External Talk Time 22: Extension Outbound Internal Talk Time 23: Extension Outbound External Talk Time
TimeFrame	Smallint	The specific length of time, in seconds. Timeframes are in two-seconds intervals. For example, a value of 4 represents 3 and 4.
Amount	Smallint	The amount of occurences

5.3.7   Stat.ExtensionStat

    a. This table provides aggregate statistics regarding all extensions in the system, including extensions not manned by agents, with regard to ClickServer. The table does not include statistics regarding call center calls.

    b. Primary Key: Extension, TheDate

    c. Table fields:

Field Name	Field Type	Field Contents
CalcTS	DateTime	Date and time when this row was calculated
TheDate	DateTime	Date and time of the record data
TheMonth	Nvarchar(7)	Month of the date and time of the record data
TheWeek	Nvarchar(11)	Week of the date and time of the record data
TheDay	Nvarchar(10)	Day of the date and time of the record data
TheHour	Nvarchar(5)	Hour of the date and time of the record data
Extension	Smallint	Extension code in the Optimus system. Note that this is the internal code and **not** the extension number itself.
InOffered	Smallint	The number of internal incoming calls offered to the agent

InAnswered	Smallint	The number of internal incoming calls handled by the agent
InAbandoned	Smallint	The number of internal incoming calls abandoned while ringing at the agent's extension
ExOffered	SmallExt	The number of external Incoming calls offered to the agent
ExAnswered	SmallExt	The number of external Incoming calls handled by the agent
ExAbandoned	SmallExt	The number of external Incoming calls abandoned while ringing at the agent's extension
IntDialed	Smallint	The number of internal calls dialed by the agent
IntAnsweredTo	Smallint	The number of internal calls dialed by the agent and answered to
ExtDialed	Smallint	The number of external calls dialed by the agent
ExtAnsweredTo	Smallint	The number of external calls dialed by the agent and answered to

HoldCount	Smallint	The number of times the agent put a call on hold.
MinHoldTime	Smallint	Minimum Hold time (in seconds)
SumHoldTime	Smallint	Total Hold time (in seconds)
MaxHoldTime	Smallint	Maximum Hold time (in seconds)
MinInIntCallsTalkTime	Smallint	Minimum incoming internal calls Talk Time (in seconds)
SumInIntCallsTalkTime	Smallint	Total incoming internal calls Talk Time (in seconds)
MaxInIntCallsTalkTime	Smallint	Maximum incoming internal calls Talk Time (in seconds)
MinInExtCallsTalkTime	Smallint	Minimum incoming external calls Talk Time (in seconds)
SumInExtCallsTalkTime	Smallint	Total incoming external calls Talk Time (in seconds)
MaxInExtCallsTalkTime	Smallint	Maximum incoming external calls Talk Time (in seconds)
MinOutIntCallsTalkTime	Smallint	Minimum outgoing internal calls Talk Time (in seconds)
SumOutIntCallsTalkTime	Smallint	Total outgoing internal calls Talk Time (in seconds)
MaxOutIntCallsTalkTime	Smallint	Maximum outgoing internal calls Talk Time (in seconds)
MinOutExtCallsTalkTime	Smallint	Minimum outgoing external calls Talk Time (in seconds)

SumOutExtCallsTalkTime	Smallint	Total outgoing external calls Talk Time (in seconds)
MaxOutExtCallsTalkTime	Smallint	Maximum outgoing external calls Talk Time (in seconds)
MinOutIntGoodDialTime	Smallint	Minimum time spent dialing to an internal destination for calls that were answered (in seconds)
SumOutIntGoodDialTime	Smallint	Total time spent dialing to an internal destination for calls that were answered (in seconds)
MaxOutIntGoodDialTime	Smallint	Maximum time spent dialing to an internal destination for calls that were answered (in seconds)
MinOutIntBadDialTime	Smallint	Minimum time spent dialing to an internal destination for calls that were not answered (in seconds)
SumOutIntBadDialTime	Smallint	Total time spent dialing to an internal destination for calls that were not answered (in seconds)
MaxOutIntBadDialTime	Smallint	Maximum time spent dialing to an internal destination for calls that were not answered (in seconds)

MinOutExtGoodDialTime	Smallint	Minimum time spent dialing to an external destination for calls that were answered (in seconds)
SumOutExtGoodDialTime	Smallint	Total time spent dialing to an external destination for calls that were answered (in seconds)
MaxOutExtGoodDialTime	Smallint	Maximum time spent dialing to an external destination for calls that were answered (in seconds)
MinOutExtBadDialTime	Smallint	Minimum time spent dialing to an external destination for calls that were not answered (in seconds)
SumOutExtBadDialTime	Smallint	Total time spent dialing to an external destination for calls that were not answered (in seconds)
MaxOutExtBadDialTime	Smallint	Maximum time spent dialing to an external destination for calls that were not answered (in seconds)
MinInIntCallsRingTime	Smallint	Minimum incoming internal calls Ring Time (in seconds)
SumInIntCallsRingTime	Smallint	Total incoming internal calls Ring Time (in seconds)

MaxInIntCallsRingTime	Smallint	Maximum incoming internal calls Ring Time (in seconds)
MinInExtCallsRingTime	Smallint	Minimum incoming external calls Ring Time (in seconds)
SumInExtCallsRingTime	Smallint	Total incoming external calls Ring Time (in seconds)
MaxInExtCallsRingTime	Smallint	Maximum incoming external calls Ring Time (in seconds)
MinIdleOffHookTime	Smallint	Minimum time spent in the "off hook" situation without dialing (in seconds)
SumIdleOffHookTime	Smallint	Total time spent in the "off hook" situation without dialing (in seconds)
MaxIdleOffHookTime	Smallint	Maximum time spent in the "off hook" situation without dialing (in seconds)

5.3.8 Stat.RouterStat

    a. This table provides aggregate statistics regarding smart routers, if the module is active.

    b. Primary Key: Extension, DeviceType, DeviceCode, BlockID

    c. The table includes cumulative data for an entire router (using device type 11 and block ID = -1) and for each block in the router (using device type 12 and the relevant block ID in the router)

    d. Table fields:

Field Name	Field Type	Field Contents
CalcTS	DateTime	Date and time when this row was calculated
TheDate	DateTime	Date and time of the record data
TheMonth	Nvarchar(7)	Month of the date and time of the record data
TheWeek	Nvarchar(11)	Week of the date and time of the record data
TheDay	Nvarchar(10)	Day of the date and time of the record data
TheHour	Nvarchar(5)	Hour of the date and time of the record data
DeviceType	Tinyint	Device Type: 11 – Router 12 – Router Block
DeviceCode	Smallint	The Device Code
BlockType	Tinyint	See Router.BlockTypes

Offered	Smallint	The number of calls that entered this specific router or block
Answered	Smallint	The number of calls that received treatment in this router or block
Abandoned	Smallint	The number of calls that were abandoned in this router or block
MinAnswerTime	Smallint	The minimum time (in seconds) that a call spent in this router or block, for calls that received treatment.
SumAnswerTime	Smallint	Total waiting time (in seconds) for calls that received treatment
MaxAnswerTime	Smallint	The maximum time (in seconds) for calls that received treatment.
MinAbdTime	Smallint	The minimum time (in seconds) that a call spent in this router or block before being abandoned
SumAbdTime	Smallint	Total waiting time (in seconds) for calls that were abandoned in this router or block

MaxAbdTime	Smallint	The maximum time (in seconds) that a call spent in this router or block before being abandoned

5.4 Stored Procedures

Note: support for BI functions was discontinued in version 5.0. Please contact Aspire support.

5.4.1 BI.AgentData

    a.   A Table-valued function providing information regarding agents

    b.   Header:

```sql
FUNCTION [BI].[AgentData]
(
    @FromDate datetime,
    @TillDate datetime
)

RETURNS @Ret table
(
            QueueCode Int,
            QueueDesc nvarchar(50),
            UserCode Int,
            FirstName nvarchar(20),
            LastName nvarchar(20),
            TheDay nvarchar(15),
            Offered Int,
            Answered Int,
            AnsweredPercentage Float,
            TalkTime Int,
            WaitingTime Int,
            OutboundCalls Int,
            LoginTime int,
            NotReadyTime int,
            ReadyTime int,
            ForceNotReady Int,
            RetFromAgent Int
)
```

5.4.2 BI.QueueData

    a.   A Table-valued function providing information regarding queues

    b.   Header:

```sql
FUNCTION [BI].[QueueData]
(
    @FromDate datetime,
    @TillDate datetime
)

RETURNS @Ret table
(
            QueueCode Int,
            QueueDesc nvarchar(50),
            TheDay nvarchar(15),
```

```
                        Offered Int,
                        Answered Int,
                        AnsweredPercentage Float,
                        Abandoned Int,
                        AbandonedPercentage Float,
                        TalkTime Int,
                        WaitingTime Int
    )
```

5.4.3 BI.GetAllQueues

a. A Table-valued function providing a list of all queues defined in Optimus

b. Header:

```
FUNCTION BI.GetAllQueues()
RETURNS table
(   Code Int,
    Description nvarchar(50))
```

5.4.4 BI.GetAllQueuesAgents

a. A Table-valued function providing a list of all queues and agents affiliation as defined in Optimus

b. Header:

```
FUNCTION BI.GetAllQueuesAgents()
RETURNS table
(   QueueCode Int,
    UserCode Int)
```

5.4.5 BI.DialerTargetsData

a. A Table-valued function providing complete information about records in the dialer table and the exact results of the activity for each record.
You can retrieve information about multiple targets by passing the Target_ID identifier separated by commas.

b. Header:

```
FUNCTION [BI].[DialerTargetsData](@Targets nvarchar(max))

RETURNS @Ret table
(TargetID int,
FirstDialTS datetime,
NextDialTS datetime,
LastDialTS datetime,
CampaignCode int,
```

```
CampaignName nvarchar(200),
QueueName nvarchar(50),
AgentName nvarchar(40),
DialLength decimal,
CallTalkLength decimal,
RecordingPath nvarchar(max),
TotalDials decimal,
DaysCount decimal,
CurrentStatus nvarchar(50),
CurrestStatusCode int,
TimeSinceLastCall decimal,
PhoneI nvarchar(30),
PhoneII nvarchar(30),
PhoneIII nvarchar(30),
DataI nvarchar(50),
DataII nvarchar(50),
DataIII nvarchar(50),
FirstName nvarchar(50),
LastName nvarchar(50))
```

6. Dialer Integration

Dialer integration enables interaction with the Optimus Pro dialer campaigns from within your software. The Optimus Pro dialer is a powerful outbound calls tool, enabling both predictive and progressive dialing work methods.

Integration uses a web service and SQL stored procedures or RESTful interface to enable the following functionality:

- Adding a record to a dialer campaign
- Deleting a record from a dialer campaign

6.1 WSDL Interface

Location: http://optimus_server/OptimusIntegration/WebServices/Dialer.asmx

Dialer.wsdl

Functions

6.1.1   Add record to campaign

6.1.1.1 Signature

> public bool AddTargetToCampaign(string userName, string password, int campaignId, string fName, string lName, string phoneI, string phoneII, string phoneIII, string dataI, string dataII, string dataIII, int priority)

6.1.1.2 Fields Additional Information

username =!@#UN   (string, not null)
password= !@#Password (string, not null)
priority – 1 – High, 2 – Medium (system default), 3 - Low

6.1.1.3 Output

The function returns **true** for success and **false** for failure

6.1.2 Delete record from campaign

6.1.2.1 Signature

public bool RemoveTargetFromCampaign (string userName, string password, int campaignId, string phone)

6.1.2.2 Fields Additional Information

username =!@#UN   (string, not null)
password= !@#Password (string, not null)

6.1.2.3 Output

The function returns **true** for success and **false** for failure

6.1.3 Find the status of a certain target (person to call)

6.1.3.1 Signature

public TargetStatus GetTargetStatus(string userName, string password,
        int campaignId, string phoneI,
        string phoneII, string phoneIII, string dataI,
        string dataII, string dataIII

6.1.3.2 Fields Additional Information

username =!@#UN   (string, not null)
password= !@#Password (string, not null)

6.1.3.3 Output

The function returns the following object:

```
public class TargetStatus
        {
            public int StatusCode
            public int Sum
        }
```

6.1.4 Set a campaign to be active or not active

6.1.4.1 Signature

```
public bool SetCampaignData(string userName, string password, int campaignId,
bool isActive)
```

6.1.4.2 Fields Additional Information

username =!@#UN   (string, not null)
password= !@#Password (string, not null)

6.1.4.3 Output

The function returns **true** for success and **false** for failure

6.1.5 Add a new campaign

6.1.5.1 Signature

```
public int AddNewCampaign(string userName, string password, string campaignName,
int campaignPriority, int campaignType, int retryNumber, int noAnswerDuration, bool
chase, int queueCode, int callPriority, bool isActive)
```

6.1.5.2 Fields Additional Information

username =!@#UN   (string, not null)
password= !@#Password (string, not null)

6.1.5.3 Output

The function returns the new campaign's Campaign ID, type int.

6.1.6 Update an existing campaign

6.1.6.1 Signature

```
 public bool UpdateCampaign(string userName, string password, int campaignId, int
campaignPriority, int campaignType,int retryNumber, int noAnswerDuration, bool
chase, int queueCode, int callPriority, bool isActive)
```

6.1.6.2 Fields Additional Information

username =!@#UN   (string, not null)

password= !@#Password (string, not null)

6.1.6.3 Output

The function returns **true** for success and **false** for failure

6.1.7 Delete an existing campaign

please note that the campaign will not be deleted from the database but will no longer appear in the GUI.

6.1.7.1 Signature

```
public bool DeleteCampaign(string userName, string password, int campaignId)
```

6.1.7.2 Fields Additional Information

username =!@#UN   (string, not null)
password= !@#Password (string, not null)

6.1.7.3 Output

The function returns **true** for success and **false** for failure

6.1.8 Add a new schedule

6.1.8.1 Signature

```
public int AddNewSchedule(string userName, string password, string scheduleName,
DateTime fromDateTime, DateTime toDateTime, bool isSpecialDays, bool isSunday, bool
isMonday, bool isTuesday,bool isWednesday, bool isThursday , bool isFriday, bool
isSaturday )
```

6.1.8.2 Fields Additional Information

username =!@#UN   (string, not null)
password= !@#Password (string, not null)

6.1.8.3 Output

The function returns the new schedule's Schedule ID, type int.

6.1.9   Update an existing schedule

6.1.9.1 Signature

```
public int UpdateSchedule(string userName, string password, int scheduleId,
DateTime fromDateTime, DateTime toDateTime, bool isSpecialDays, bool isSunday, bool
isMonday, bool isTuesday,bool isWednesday, bool isThursday, bool isFriday, bool
isSaturday)
```

6.1.9.2 Fields Additional Information

username =!@#UN   (string, not null)
password= !@#Password (string, not null)

6.1.9.3 Output

The function returns **true** for success and **false** for failure

6.1.10 Delete an existing schedule

6.1.10.1   Signature

```
public bool DeleteSchedule(string userName, string password, int scheduleId)
```

6.1.10.2   Fields Additional Information

username =!@#UN   (string, not null)
password= !@#Password (string, not null)

6.1.10.3   Output

The function returns **true** for success and **false** for failure

6.2 SQL Interface

Accessible through the Optimus Pro database (see "Optimus Database Integration" in this guide for connection details)

Stored Procedures and Functions

6.2.1   Add record to campaign

6.2.1.1 Header

```
PROCEDURE [Dialer].[InsertTargetToCampaign]
(
        @CampaignCode int,
        @Name nvarchar(100),
        @LastName nvarchar(100),
        @PhoneI nvarchar(12),
        @PhoneII nvarchar(12),
        @PhoneIII nvarchar(12),
        @DataI nvarchar(100),
        @DataII nvarchar(100),
        @DataIII nvarchar(100),
        @FileNumber int,
        @PRIORITY int,
        @EXTERNAL_TARGET_ID int
)
```

6.2.2   Delete record from campaign

6.2.2.1 Header

```
PROCEDURE [Dialer].[DeleteTarget]
(
        @Code bigint
)
```

6.2.3   Retrieve a target's status in the dialer

6.2.3.1 Header

```
FUNCTION [Dialer].[GetTargetsLastStatus]
(
        @Campaign int,
        @phoneNumber nvarchar(12)
)

RETURNS int
```

6.2.4 Retrieve the call status description

6.2.4.1 Header

```
FUNCTION [Dialer].[GetCallStatusDescription]
(
        @StatusCode int,
        @languageId int
)

RETURNS nvarchar(500)
```

6.2.5 Retrieve a list of all targets according to campaign, timestamp and status

6.2.5.1 Header

```
function bi.GetDialerTargetsByStatus(
        @CampaignID int,
        @FromTS datetime,
        @TillTS datetime,
        @Status int
)
returns table

(Target_ID int,
Last_Call_TS datetime,
Next_Call_TS datetime,
Next_Phone tinyint,
Call_Me tinyint,
Phone_I nvarchar(20),
Phone_II nvarchar(20),
Phone_III nvarchar(20),
Data_I nvarchar(50),
Data_II nvarchar(50),
Data_III nvarchar(50),
First_Name nvarchar(50),
Last_Name nvarchar(50),
Phone_I_Counter int,
Phone_II_Counter int,
Phone_III_Counter int)
```

6.3 RESTful Interface

Please note that currently on a GET method is supported.

```
INTEGRATION_METHOD_USER_NAME = "aspireun";
INTEGRATION_METHOD_PASSWORD = "aspirepass";
```

Methods

6.3.1   Add record to campaign

6.3.1.1 Signature

AddTargetToCampaign:
```
public bool AddTargetToCampaign(string userName, string password, campaignID
int,fName string,lName string,PhoneI string,PhoneII string,dataI string,dataII
string,dataIII string, priority int)
```

6.3.1.2 Example

http://OptimusServer/OptimusIntegration/Rest/DialerWebMethods.aspx/**AddTargetToCampaign**?username=aspireun&password=aspirepass&campaignId=1&phoneI=0536608810&dataI=Tami

6.3.2   Add record to campaign with targetID

6.3.2.1 Signature

AddTargetToCampaignWithTargetID:
```
public int AddTargetToCampaignWithTargetID(string userName, string password,
campaignID int,fName string,lName string,PhoneI string,PhoneII string,dataI
string,dataII string,dataIII string, priority int)
```

6.3.2.2 Example

URL Example :
http://OptimusServer/OptimusIntegration/Rest/DialerWebMethods.aspx/**AddTargetToCampaignWithTargetID**?username=aspireun&password=aspirepass&campaignId=1&phoneI=0536608810&dataI=Tami

Return : targetID

6.3.3 Get Target Status

6.3.3.1 Signature

GetTargetStatus:

```
public TargetStatus GetTargetStatus(string userName, string password, int
campaignId, string phoneI, string phoneII, string phoneIII, string dataI, string
dataII, string dataIII)
```

6.3.3.2 Example

URL Example :
http://OptimusServer/OptimusIntegration/Rest/DialerWebMethods.aspx/**GetTargetStatus**?username=aspireun&password=aspirepass&campaignId=1&phoneI=0536608810&dataI=tami

```
Return -   StatusCode,Sum
```

6.3.4 Delete Target

6.3.4.1 Signature

RemoveTargetFromCampaign:

```
public bool RemoveTargetFromCampaign(string userName, string password, int
campaignId, string phone)
```

6.3.4.2 Example

URL Example :
http://OptimusServer/OptimusIntegration/Rest/DialerWebMethods.aspx/**RemoveTargetFromCampaign**?username=aspireun&password=aspirepass&campaignId=1&phoneI=0536608810

Return – state (TRUE or FALSE)

7. Recording Module Integration

Recording Module integration enables interaction with the Optimus Pro recording module from within your software. The main feature is receiving the recorded call file location and being able to play it from within your software.

Integration uses an SQL stored functions to receive the following information:

- The full file path of a single call according to the Call ID
- The full file path and Call ID of all calls that match a certain CLID

7.1 SQL Interface

Accessible through the Optimus Pro database (see "Optimus Database Integration" in this guide for connection details)

Stored Procedures and Functions

7.1.1 Recording file path of a single call

7.1.1.1 Header

```
FUNCTION [Configuration].[GetCallRecordingFile]
(
        @CallID int
)

RETURNS nvarchar(max)
```

7.1.2 All recorded calls for a certain CLID

7.1.2.1 Header

```
FUNCTION [Stat].[GetAllRecordings]
(
        @FromTS datetime,
        @TillTS datetime,
        @CLID nvarchar(20)
)

RETURNS @ret table(
CallID int,
Recfile nvarchar(200))
```

8. Single Sign On Integration

Single Sign On integration enables interaction with the Optimus Pro from within your software. It allows changing the status of an agent in real time. You can log an agent into Optimus, log them out, and change their status to a specific activity code.

Integration uses a WSDL interface, as follows:

8.1 WSDL Interface

AgentServices.asmx

8.1.1 LOGIN

- webServiceUserName – Web service authentication. Set value to: !@#UN
- password – Web service authentication. Set value to: !@#Password
- Username – Optimus Usercode of the user to log in.
- ExtensionNumber – the extension where the user is located and logs into.

8.1.2 LOGOUT

- webServiceUserName – Web service authentication. Set value to: !@#UN
- password – Web service authentication. Set value to: !@#Password
- Username – Optimus Usercode of the user to log out.

8.1.3 NOT READY

- webServiceUserName – Web service authentication. Set value to: !@#UN
- password – Web service authentication. Set value to: !@#Password
- Username – Optimus Usercode of the user to change the status of.
- activityCode – The activity code to which the agent status will be changed.

8.1.4 CheckExtensionState

- webServiceUserName – Web service authentication. Set value to: !@#UN

- password – Web service authentication. Set value to: !@#Password

- Username – Optimus Usercode of the user to check status of