Outlook Data Model

for use with Invantive SQL

23.0

Copyright

Important Safety and Usage Information

Intended Use and Limitations: This software, developed by Invantive, is designed to support a variety of business and information technology data processing functions, such as accounting, financial reporting an sales reporting. It is important to note that this software is not designed, tested, or approved for use in environments where malfunction or failure could lead to life-threatening situations or severe physical or environmental damage. This includes, but is not limited to:

- Nuclear facilities: The software should not be used for operations or functions related to the control, maintenance, or operation of nuclear facilities.
- Defense and Military Applications: This software is not suitable for use in defense-related applications, including but not limited to weaponry control, military strategy planning, or any other aspects of national defense.
- Aviation: The software is not intended for use in the operation, navigation, or communication systems of any aircraft or air traffic control environments.
- Healthcare and Medicine Production: This software should not be utilized for medical device operation, patient data analysis for critical health decisions, pharmaceutical production, or medical research where its failure or malfunction could impact patient health.
- Chemical and Hazardous Material Handling: This software is not intended for the management, control, or operational aspects of chemical plants or hazardous material handling facilities. Any malfunction in software used in these settings could result in dangerous chemical spills, explosions, or environmental disasters.
- Transportation and Traffic Control Systems: The software should not be used for the control, operation, or management of transportation systems, including railway signal controls, subway systems, or traffic light management. Malfunctions in such critical systems could lead to severe accidents and endanger public safety.
- Energy Grid and Utility Control Systems: This software is not designed for the control or operation of energy grid systems, including electrical substations, renewable energy control systems, or water utility control systems. The failure of software in these areas could lead to significant power outages, water supply disruptions, or other public utility failures, potentially endangering communities and causing extensive damage.
- Other High-Risk Environments: Any other critical infrastructure and environments where a failure of the software could result in significant harm to individuals or the environment.

User Responsibility: Users must ensure that they understand the intended use of the software and refrain from deploying it in any setting that falls outside of its designed purpose. It is the responsibility of the user to assess the suitability of the software for their intended application, especially in any scenarios that might pose a risk to life, health, or the environment.
Disclaimer of Liability: Invantive disclaims any responsibility for damage, injury, or legal consequences resulting from the use or misuse of this software in prohibited or unintended applications.

Contents

1 SQL Driver for Outlook API

Use the "Search" option in the left menu to search for a specific term such as the table or column description. When you already know the term, please use the "Index" option. When you can't find the information needed, please click on the Chat button at the bottom or place your question in the user community. Invantive Support or other users will try to help you.

Microsoft Outlook data such as contacts and calendar items.

The Outlook driver covers 6 tables and 271 columns.

Outlook API Clients

Invantive UniversalSQL is available on many user interfaces ("clients" in traditional server-client paradigma). All Invantive UniversalSQL statements can be exchanged with a close to 100% compatibility across all clients and operating systems (Windows, MacOS, Linux, iOS, Android).

The clients include Microsoft Excel, Microsoft Power BI, Microsoft Power Query, Microsoft Word and Microsoft Outlook. Web-based clients include Invantive Cloud, Invantive Bridge Online as OData proxy, Invantive App Online for interactive apps, Online SQL Editor for query execution and Invantive Data Access Point as extended proxy.

For technical users there are command-line editions of Invantive Data Hub running on iOS, Android, Windows, MacOS and Linux. Invantive Data Hub is also often used for enterprise server applications such as ETL.

Specifications

The SQL driver for Outlook supports partitioning: data from multiple companies are all listed together in one table. The value of the column indicates which company the data belongs to.

An introduction into the concepts of Invantive UniversalSQL such as databases, data containers and partitioning can be found in the Invantive UniversalSQL grammar.

The configuration can be changed using various attributes from the database definition, on log on and during use. A full list of configuration options is listed in the driver attributes 1.

The catalog name is used to compose the full qualified name of an object like a table or view. The schema name is used to compose the full qualified name of an object like a table or view. On Outlook the comparison of two texts is case sensitive by default.

Changes and bug fixes on the Outlook SQL driver can be found in the release notes. Get access to the community through the Outlook section of the Invantive forums.

Driver code for use in settings.xml: `Outlook`

Alias: `outlook`

Recommended alias: `olk`

The Outlook instance currently running can be accessed on Microsoft Windows through COM.

Updated 30-05-2024 10:38 using Invantive UniversalSQL version 24.1.3-BETA+4689.

2 SQL Driver Attributes for Outlook API

The SQL driver for Outlook has many attributes that can be finetuned to improve handling in scenarios with unreliable network connections to the API server of Outlook or high volumes

of data. Also, many drivers have driver-specific attributes to finetune actual behaviour or handle data not matching specifications.

The Outlook driver attributes are assigned a default value which seldom requires change. However, changes can be applied when needed on four levels, which are reflected in the table below by separate checkmarks:

- Connection string: the connection string from the settings*.xml file and applied during log on.
- Set SQL statement: a set SQL-statement to be executed once connection has been established.
- Log on: value to be specified interactively by user during log on in a user interface.

The connection string for Outlook can be found in the settings*.xml file used for the database. The reference manuals contain instructions how to relocate the settings*.xml files. Settings*.xml files are typically located in the `%USERPROFILE%\invantive` folder in most deployment scenarios. Each data container of a database in the connection string can have a `connectionString` element specifying the name and values of attributes. Both name and value must be properly escaped according to XML-semantics. Actual application of the value is solely done during log on. A new connection must be established to change the value of a driver attribute using a connection string.

The set SQL statement can be executed after log on. The syntax is: `set NAME VALUE`, or for a distributed database: `set NAME@ALIAS VALUE`. In some scenarios you may need to enclose the driver attribute name in square brackets to escape it from parsing, for instance when a reserved SQL keyword is part of the name. The new value takes effect straight after execution of the set-statement. The set-statement can be executed as often as needed during a session.

Driver attributes that can be interactively set to a value are typically presented in the log on window. Depending on the platform and design decisions of the user interface designer, some or all of the available driver attributes can have been made available.

The Outlook driver can be configured using the following attributes:

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
ignore-com-exception	Whether to ignore COM exceptions by default.	Outlook	False	✓	✓	✓	

3 Schema: Calendars

3.1 Tables

3.1.1 APPOINTMENTS: Outlook Appointments

Catalog: Outlook

Schema: Calendars

Primary Keys: EntryID

Label: Appointments

Documentation:

Outlook appointments.

Can retrieve data and change data using insert, update and delete.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function APPOINTMENTS. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
folderEntryId	string	☐		Unique ID of the folder as a reference to the table 'FOLDERS'.
storeEntryId	string	☐		Unique ID of a store as a reference to the table 'FOLDERS' when queried with include external stores.

Columns of Table Function

The columns of the table function APPOINTMENTS are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
AllDayEvent	boolean		☐	
BillingInformation	string		☐	
Body	string			
BusyStatus	int32		☐	
Categories	string	Categories		
ConversationIndex	string		☐	
ConversationTopic	string		☐	
CreationTime	datetime		☐	
Duration	int32		☐	
End	datetime		☐	
EntryID	string(1024)			
FolderID	string(1024)			

Name	Data Type	Label	Required	Documentation
Importance	int32		☐	
IsOnlineMeeting	boolean		☐	
IsRecurring	boolean		☐	
LastModificationTime	datetime		☐	
Location	string		☐	
MeetingStatus	int32		☐	
MeetingWorkspaceURL	string		☐	
Mileage	string		☐	
NetMeetingAutoStart	boolean		☐	
NetMeetingDocPathName	string		☐	
NetMeetingOrganizerAlias	string		☐	
NetMeetingServer	string		☐	
NetMeetingType	int32		☐	
NetShow URL	string		☐	
NoAging	boolean		☐	
OptionalAttendees	string		☐	
Organizer	string		☐	
ReminderMinutesBeforeStart	int32		☐	
ReminderOverrideDefault	boolean		☐	
ReminderPlaySound	boolean		☐	
ReminderSet	boolean		☐	
ReminderSoundFile	string		☐	
ReplyTime	datetime		☐	
RequiredAttendees	string		☐	
Resources	string		☐	
ResponseRequested	boolean		☐	
ResponseStatus	int32		☐	
Sensitivity	int32		☐	
Size	int32		☐	
Start	datetime		☐	
StoreID	string(1024)			
Subject	string		☐	
UnRead	boolean		☐	

4 Schema: Contacts

4.1 Tables

4.1.1 CONTACTS: Outlook Contacts

Catalog: Outlook

Schema: Contacts

Primary Keys: EntryID

Label: Contacts

Documentation:

Outlook contacts.

Can retrieve data and change data using insert, update and delete.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CONTACTS. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
folderEntryId	string	☐		Unique ID of the folder as a reference to the table 'FOLDERS'.
storeEntryId	string	☐		Unique ID of a store as a reference to the table 'FOLDERS' w hen queried w ith include external stores.

Columns of Table Function

The columns of the table function CONTACTS are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
Account	string		☐	
Anniversary	datetime		☐	
AssistantName	string		☐	
AssistantTelephoneNumber	string		☐	
BillingInformation	string			
Birthday	datetime		☐	
Body	string			
Business2TelephoneNumber	string		☐	
BusinessAddress	string		☐	

Name	Data Type	Label	Required	Documentation
BusinessAddressCity	string		☐	
BusinessAddressCountry	string		☐	
BusinessAddressPostalCode	string		☐	
BusinessAddressPostOfficeBox	string		☐	
BusinessAddressState	string		☐	
BusinessAddressStreet	string		☐	
BusinessFaxNumber	string		☐	
BusinessHomePage	string		☐	
BusinessTelephoneNumber	string		☐	
CallbackTelephoneNumber	string		☐	
CarTelephoneNumber	string		☐	
Categories	string	Categories		
CompanyAndFullName	string		☐	
CompanyMainTelephoneNumber	string		☐	
CompanyName	string		☐	
ComputerNetworkName	string		☐	
ConversationIndex	string		☐	
ConversationTopic	string		☐	
CreationTime	datetime		☐	
CustomerID	string		☐	
Department	string		☐	
Email1Address	string		☐	
Email1AddressType	string		☐	
Email1DisplayName	string		☐	
Email2Address	string		☐	
Email2AddressType	string		☐	
Email2DisplayName	string		☐	
Email3Address	string		☐	
Email3AddressType	string		☐	
Email3DisplayName	string		☐	
EntryID	string(1024)			
FileAs	string		☐	
FirstName	string		☐	
FolderID	string(1024)			
FTPSite	string		☐	
FullName	string		☐	
FullNameAndCompany	string		☐	
Gender	int32		☐	
GovernmentIDNumber	string		☐	
Hobby	string		☐	
Home2TelephoneNumber	string		☐	

Name	Data Type	Label	Required	Documentation
HomeAddress	string		☐	
HomeAddressCity	string		☐	
HomeAddressCountry	string		☐	
HomeAddressPostalCode	string		☐	
HomeAddressPostOfficeBox	string		☐	
HomeAddressState	string		☐	
HomeAddressStreet	string		☐	
HomeFaxNumber	string		☐	
HomeTelephoneNumber	string		☐	
IMAddress	string		☐	
Importance	int32		☐	
Initials	string		☐	
InternetFreeBusyAddress	string		☐	
ISDNNumber	string		☐	
JobTitle	string		☐	
Journal	boolean		☐	
Language	string		☐	
LastModificationTime	datetime		☐	
LastName	string		☐	
LastNameAndFirstName	string		☐	
MailingAddress	string		☐	
MailingAddressCity	string		☐	
MailingAddressCountry	string		☐	
MailingAddressPostalCode	string		☐	
MailingAddressPostOfficeBox	string		☐	
MailingAddressState	string		☐	
MailingAddressStreet	string		☐	
ManagerName	string		☐	
MiddleName	string		☐	
Mileage	string		☐	
MobileTelephoneNumber	string		☐	
NickName	string		☐	
NoAging	boolean		☐	
OfficeLocation	string		☐	
OrganizationalIDNumber	string		☐	
OtherAddress	string		☐	
OtherAddressCity	string		☐	
OtherAddressCountry	string		☐	
OtherAddressPostalCode	string		☐	
OtherAddressPostOfficeBox	string		☐	
OtherAddressState	string		☐	

Name	Data Type	Label	Required	Documentation
OtherAddressStreet	string		☐	
OtherFaxNumber	string		☐	
OtherTelephoneNumber	string		☐	
PagerNumber	string		☐	
PersonalHomePage	string		☐	
PrimaryTelephoneNumber	string		☐	
Profession	string		☐	
RadioTelephoneNumber	string		☐	
ReferredBy	string		☐	
ReminderOverrideDefault	boolean		☐	
ReminderPlaySound	boolean		☐	
ReminderSet	boolean		☐	
ReminderSoundFile	string		☐	
ReminderTime	datetime		☐	
SelectedMailingAddress	int32		☐	
Sensitivity	int32		☐	
Size	int32		☐	
Spouse	string		☐	
StoreID	string(1024)			
Subject	string		☐	
Suffix	string		☐	
TaskCompletedDate	datetime		☐	
TaskDueDate	datetime		☐	
TaskStartDate	datetime		☐	
TelexNumber	string		☐	
Title	string		☐	
ToDoTaskOrdinal	datetime		☐	
TTYTDDTelephoneNumber	string		☐	
UnRead	boolean		☐	
User1	string		☐	
User2	string		☐	
User3	string		☐	
User4	string		☐	
UserCertificate	string		☐	
WebPage	string		☐	
YomiCompanyName	string		☐	
YomiFirstName	string		☐	
YomiLastName	string		☐	

5 Schema: General
5.1 Tables
5.1.1 FOLDERS: Outlook Folders

Catalog: Outlook

Schema: General

Primary Keys: EntryID

Label: Folders

Documentation:

Outlook folders.

This is a read-only table function. The Outlook API may not support changing the data or the Invantive UniversalSQL driver for Outlook does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Outlook API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function FOLDERS. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
ignoreComException	boolean	☐	False	Whether to ignore COM exception during retrieval of data.
includeDefaultFolders	boolean	☐	True	Whether to include default folders of the current user.
includeExternalStores	boolean	☐	False	Whether to include external stores available to the session besides the current user.
includeSharedFolders	boolean	☐	False	Whether to include shared folders available to the session besides the current user.

Columns of Table Function

The columns of the table function FOLDERS are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
AddressBookName	string(240)		☐	
Class	string		☑	
DefaultMessageClass	string		☑	
Description	string		☐	
DisplayName	string(240)		☐	
EntryID	string(1024)		☑	
FolderPath	string(240)		☑	
GroupName	string(240)		☐	
GroupPosition	int32		☐	
IsSharepointFolder	boolean		☑	
Label	string(240)		☑	
Name	string		☑	
Position	int32		☐	
SourceName	string		☑	
SourceType	string(30)		☑	
StoreID	string(1024)		☑	
UnReadItemCount	int32		☑	
WebView URL	string(240)		☐	

5.1.2 SESSIONS: Outlook Sessions

Catalog: Outlook

Schema: General

Label: Sessions

Documentation:

Outlook sessions.

This is a read-only table function. The Outlook API may not support changing the data or the Invantive UniversalSQL driver for Outlook does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Outlook API.

Columns of Table Function

The columns of the table function SESSIONS are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
className	string(240)		☑	
sessionAutoDiscoverConnectionMode	string(240)		☐	
sessionAutoDiscoverXml	string		☐	
sessionCurrentProfileName	string(240)		☑	
sessionCurrentUserAddress	string(240)		☐	

Name	Data Type	Label	Required	Documentation
sessionCurrentUserName	string(240)		☐	
sessionDefaultStoreId	string(1024)		☐	
sessionExchangeConnectionMode	string(240)		☐	
sessionExchangeMailboxServerName	string(240)		☐	
sessionExchangeMailboxServerVersion	string(240)		☐	
sessionOffline	boolean		☑	
sessionType	string(240)		☐	

5.1.3 STORES: Outlook Stores

Catalog: Outlook

Schema: General

Primary Keys: StoreID

Label: Stores

Documentation:

Outlook stores.

This is a read-only table function. The Outlook API may not support changing the data or the Invantive UniversalSQL driver for Outlook does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Outlook API.

Columns of Table Function

The columns of the table function STORES are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ClassName	string(240)		☑	
DefaultMapiFolderCalendarEntryId	string(1024)		☐	
DefaultMapiFolderCalendarFullPath	string(240)		☐	
DisplayName	string(240)		☑	
FilePath	string(240)		☐	
IsCachedExchange	boolean		☑	
IsConversationEnabled	boolean		☑	
IsDataFileStore	boolean		☑	
IsDefaultStore	boolean		☑	
IsInstantSearchEnabled	boolean		☑	
IsOpen	boolean		☑	
RootMapiFolderEntryId	string(1024)		☐	
RootMapiFolderFullFolderPath	string(240)		☐	
StoreID	string(1024)		☑	
TypeCode	string(240)		☑	

6 Schema: Mails

6.1 Tables

6.1.1 MAILS: Outlook Mails

Catalog: Outlook

Schema: Mails

Primary Keys: EntryID

Label: Mails

Documentation:

Outlook mails.

Can retrieve data and change data using insert, update and delete.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function MAILS. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
folderEntryId	string	☐		Unique ID of the folder as a reference to the table 'FOLDERS'.
storeEntryId	string	☐		Unique ID of a store as a reference to the table 'FOLDERS' when queried with include external stores.

Columns of Table Function

The columns of the table function MAILS are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
AlternateRecipientAllowed	boolean		☐	
AutoForwarded	boolean		☐	

Name	Data Type	Label	Required	Documentation
BCC	string		☐	
BillingInformation	string		☐	
Body	string			
Categories	string	Categories		
CC	string		☐	
ConversationIndex	string		☐	
ConversationTopic	string		☐	
CreationTime	datetime		☐	
DeferredDeliveryTime	datetime		☐	
DeleteAfterSubmit	boolean		☐	
EntryID	string(1024)			
ExpiryTime	datetime		☐	
FlagDueBy	datetime		☐	
FlagRequest	string		☐	
FlagStatus	int32		☐	
FolderID	string(1024)			
Importance	int32		☐	
LastModificationTime	datetime		☐	
MarkForDownload	int32		☐	
Mileage	string		☐	
NoAging	boolean		☐	
OriginatorDeliveryReportRequested	boolean		☐	
ReadReceiptRequested	boolean		☐	
ReceivedByName	string		☐	
ReceivedOnBehalfOfName	string		☐	
ReceivedTime	datetime		☐	
RecipientReassignmentProhibited	boolean		☐	
ReminderOverrideDefault	boolean		☐	
ReminderPlaySound	boolean		☐	
ReminderSet	boolean		☐	
ReminderSoundFile	string		☐	
ReminderTime	datetime		☐	
RemoteStatus	string		☐	
ReplyRecipientNames	string		☐	
SenderEmailAddress	string		☐	
SenderEmailType	string		☐	
SenderName	string		☐	
Sensitivity	int32		☐	
SentOn	datetime		☐	
SentOnBehalfOfName	string		☐	
Size	int32		☐	

Name	Data Type	Label	Required	Documentation
StoreID	string(1024)			
Subject	string		☐	
TaskCompletedDate	datetime		☐	
TaskDueDate	datetime		☐	
TaskStartDate	datetime		☐	
To	string		☐	
ToDoTaskOrdinal	datetime		☐	
UnRead	boolean		☐	
VotingResponse	string		☐	

Index

- A -

- B -

- C -

- D -

- E -

- F -

- G -

- H -

- I -

- J -

- L -

- M -

- N -

- O -

- P -

- R -

- S -

- T -

- U -

- V -

- W -

- Y -

Invantive B.V.
Biesteweg 11
3849 RD Hierden
the Netherlands

Tel: +31 88 00 26 500
Fax: +31 84 22 58 178
info@invantive.com
invantive.com

IBAN NL25 BUNQ 2098 2586 07
Chamber of Industry and Commerce
13031406
VAT NL812602377B01
RSIN 8122602377
Managing Director: Guido Leenders
Registered office: Roermond

23.0.92
15/01/2025, 20:20
ISBN 978-94-6302-000-8: