

Visma Severa API Data Model

for use with Invantive SQL

24.0

invantive
the SQL company

Copyright

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher.

Despite all the care taken in the compilation of this text, neither the author nor the publisher can accept liability for any damage, which might result from any error, which might appear in this publication.

This manual is a reference guide intended to clarify usage. If data in the sample images match data in your system, the similarity is coincidental.

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 Visma Severa API 1

2 {res:itgen_doc_sql_attr_api} 2

3 Catalog: Severa 16

3.1 Schemas 16
3.1.1 Schema: API 16
3.1.2 Schema: Native 392
3.1.3 Schema: Views 394

4 Package: dcr_metadata 402

4.1 Procedures 402
4.1.1 dcr_metadata.get_partitions: Visma Severa Data container metadata package 402

Index 403

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

1 SQL Driver for Visma Severa API

Invantive UniversalSQL is the fastest, easiest and most reliable way to exchange data with the Visma Severa 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.

Visma Severa is cloud-based project management software. Visma Severa is available in the Nordics and the Netherlands.

The Visma Severa driver covers 264 tables and 5315 columns.

Visma Severa 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.

The Visma Severa Power BI connector is based on the Invantive UniversalSQL driver for Visma Severa, completed by a high-performance OData connector which works straight on Power BI without any add-on. The OData protocol is always version 4, independent whether the backing platform uses OData, SOAP or another protocol.

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. High-volume replication of data taken from the Visma Severa API into traditional databases such as SQL Server (on-premises and Azure), MySQL, PostgreSQL and Oracle is possible using Invantive Data Replicator. Invantive Data Replicator automatically creates and maintains Visma Severa datawarehouses, possibly in combination with data from over 75 other (cloud) platforms. Invantive Data Replicator supports data volumes up to over 1 TB and over 5.000 companies. The on-premise edition of Invantive Bridge offers an Visma Severa ADO.net provider.

Finally, online web apps can be build for Visma Severa using App Online of Invantive Cloud.

Monitor API Calls

When a query or DML-statement has been executed on Invantive UniversalSQL a developer can evaluate the actual calls made to the Visma Severa API using a query on sessionios@DataDictionary. As an alternative, extensive request and response logging can be enabled by setting log-native-calls-to-disk to true. In the %USERPROFILE% \Invantive\NativeLog folder Invantive UniversalSQL will create log files per Visma Severa API request and response.

Specifications

The SQL driver for Visma Severa does not support partitioning. Define one data container in a database for each company in Visma Severa to enable parallel access for data from multiple companies.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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 2.

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 Visma Severa the comparison of two texts is case sensitive by default.

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

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

Alias: `severa`

Recommended alias: `sva`

More technical documentation as provided by the supplier of Visma Severa on the native connection used can be found at https://severa.com/nl/.

General documentation on Visma Severa is available at http://severa.visma.com.

2 {res:itgen_doc_sql_attr_api}

The SQL driver for Visma Severa has many attributes that can be finetuned to improve handling in scenarios with unreliable network connections to the API server of Visma Severa or high volumes of data. Also, many drivers have driver-specific attributes to finetune actual behaviour or handle data not matching specifications.

The Visma Severa 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 Visma Severa 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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 Visma Severa 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
analysis-enforce-row-uniqueness	Enforce rows to be unique for software analysis. A fingerprint is calculated from the whole row of data when the primary key column is unknown.	Shared	False	✓	✓	✓	
api-url	URL of web service.	Visma Severa		✓		✓	
bulk-delete-page-size-rows	Number of rows to delete per batch when bulk deleting.	Shared	10000	✓	✓	✓	
bulk-insert-page-size-bytes	Approximate maximum size in bytes of batch when bulk inserting.	Shared	10000000	✓	✓	✓	
bulk-insert-page-size-rows	Number of rows to insert per batch when bulk inserting.	Shared	10000	✓	✓	✓	
download-error-400-bad-request-max-tries	Maximum number of tries when HTTP server reports bad format during retrieval of data.		3	✓	✓	✓	
download-error-400-bad-request-sleep-initial-ms	Initial sleep in milliseconds between retries when HTTP server reports that the API server is unavailable during retrieval of data.		500	✓	✓	✓	
download-error-400-bad-request-sleep-max-ms	Maximum sleep in milliseconds between retries when HTTP server reports that the API server is unavailable during retrieval of data.		5000	✓	✓	✓	
download-error-400-bad-request-sleep-multiplicator	Multiplication factor for sleep between retries HTTP server reports that the API server is unavailable during retrieval of data.		2	✓	✓	✓	
download-error-403-forbidden-max-tries	Maximum number of tries when the website reports a HTTP status 403 (Forbidden).		2	✓	✓	✓	
download-error-403-forbidden-sleep-initial-ms	Initial sleep in milliseconds between retries when the website reports a HTTP status 403 (Forbidden).		3000	✓	✓	✓	
download-error-403-forbidden-sleep-max-ms	Maximum sleep in milliseconds between retries when the website reports a HTTP status 403 (Forbidden).		10000	✓	✓	✓	
download-error-403-forbidden-sleep-multiplicator	Multiplication factor for sleep between retries when the website reports a HTTP status 403 (Forbidden).		2	✓	✓	✓	
download-error-408-request-timeout-max-tries	Maximum number of tries when the website reports a HTTP status 408.		10	✓	✓	✓	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
download-error-408-request-timeout-sleep-initial-ms	Initial sleep in milliseconds between retries when the website reports a HTTP status 408.		10000	✓	✓	✓	
download-error-408-request-timeout-sleep-max-ms	Maximum sleep in milliseconds between retries when the website reports a HTTP status 408.		300000	✓	✓	✓	
download-error-408-request-timeout-sleep-multiplicator	Multiplication factor for sleep between retries when the website reports a HTTP status 408.		2	✓	✓	✓	
download-error-422-bad-request-max-tries	Maximum number of tries when HTTP server reports unprocessable entity during retrieval of data.		30	✓	✓	✓	
download-error-422-bad-request-sleep-initial-ms	Initial sleep in milliseconds between retries when HTTP server reports unprocessable entity during retrieval of data.		10000	✓	✓	✓	
download-error-422-bad-request-sleep-max-ms	Maximum sleep in milliseconds between retries when HTTP server reports unprocessable entity during retrieval of data.		300000	✓	✓	✓	
download-error-422-bad-request-sleep-multiplicator	Multiplication factor for sleep between retries HTTP server reports unprocessable entity during retrieval of data.		2	✓	✓	✓	
download-error-429-too-many-requests-max-tries	Maximum number of tries when the website reports that too many requests have been made during a timeslot of one minute or one day.		10	✓	✓	✓	
download-error-429-too-many-requests-sleep-initial-ms	Initial sleep in milliseconds between retries when the website reports that too many requests have been made during a timeslot of one minute or one day.		10000	✓	✓	✓	
download-error-429-too-many-requests-sleep-max-ms	Maximum sleep in milliseconds between retries when the website reports that too many requests have been made during a timeslot of one minute or one day.		300000	✓	✓	✓	
download-error-429-too-many-requests-sleep-multiplicator	Multiplication factor for sleep between retries when the website reports that too many requests have been made during a timeslot of one minute or one day.		2	✓	✓	✓	
download-error-500-internal-server-error-max-tries	{res:itgen_pae_download_error_500_internal_server_error_max_tries}		10	✓	✓	✓	
download-error-500-internal-server-error-sleep-initial-ms	{res:itgen_pae_download_error_500_internal_server_error_sleep_initial_ms}		10000	✓	✓	✓	
download-error-500-internal-server-	{res:itgen_pae_download_error_500_internal_server_error_sleep_max_		300000	✓	✓	✓	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
error-sleep-max-ms	ms}						
download-error-500-internal-server-error-sleep-multiplicator	{res:itgen_pae_download_error_500_internal_server_error_sleep_multiplicator}		2	✓	✓	✓	
download-error-502-server-unavailable-max-tries	Maximum number of tries when HTTP server reports a bad gateway during retrieval of data.		30	✓	✓	✓	
download-error-502-server-unavailable-sleep-initial-ms	Initial sleep in milliseconds between retries when HTTP server reports a bad gateway during retrieval of data.		10000	✓	✓	✓	
download-error-502-server-unavailable-sleep-max-ms	Maximum sleep in milliseconds between retries when HTTP server reports that a bad gateway during retrieval of data.		300000	✓	✓	✓	
download-error-502-server-unavailable-sleep-multiplicator	Multiplication factor for sleep between retries HTTP server reports a bad gateway during retrieval of data.		2	✓	✓	✓	
download-error-503-server-unavailable-max-tries	Maximum number of tries when HTTP server reports that the API server is unavailable during retrieval of data.		30	✓	✓	✓	
download-error-503-server-unavailable-sleep-initial-ms	Initial sleep in milliseconds between retries when HTTP server reports that the API server is unavailable during retrieval of data.		10000	✓	✓	✓	
download-error-503-server-unavailable-sleep-max-ms	Maximum sleep in milliseconds between retries when HTTP server reports that the API server is unavailable during retrieval of data.		300000	✓	✓	✓	
download-error-503-server-unavailable-sleep-multiplicator	Multiplication factor for sleep between retries HTTP server reports that the API server is unavailable during retrieval of data.		2	✓	✓	✓	
download-error-504-gateway-timeout-max-tries	Maximum number of tries when the website reports a gateway timeout.		10	✓	✓	✓	
download-error-504-gateway-timeout-sleep-initial-ms	Initial sleep in milliseconds between retries when the website reports a gateway timeout.		10000	✓	✓	✓	
download-error-504-gateway-timeout-sleep-max-ms	Maximum sleep in milliseconds between retries when the website reports a gateway timeout.		300000	✓	✓	✓	
download-error-504-gateway-	Multiplication factor for sleep between retries when the website reports a gateway timeout.		2	✓	✓	✓	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
timeout-sleep-multiplicator							
download-error-590-network-connect-timeout-max-tries	Maximum number of tries when the website reports a HTTP status 590.		10	✓	✓	✓	
download-error-590-network-connect-timeout-sleep-initial-ms	Initial sleep in milliseconds between retries when the website reports a HTTP status 590.		10000	✓	✓	✓	
download-error-590-network-connect-timeout-sleep-max-ms	Maximum sleep in milliseconds between retries when the website reports a HTTP status 590.		300000	✓	✓	✓	
download-error-590-network-connect-timeout-sleep-multiplicator	Multiplication factor for sleep between retries when the website reports a HTTP status 590.		2	✓	✓	✓	
download-error-599-network-connect-timeout-max-tries	Maximum number of tries when the website reports a HTTP status 599.		10	✓	✓	✓	
download-error-599-network-connect-timeout-sleep-initial-ms	Initial sleep in milliseconds between retries when the website reports a HTTP status 599.		10000	✓	✓	✓	
download-error-599-network-connect-timeout-sleep-max-ms	Maximum sleep in milliseconds between retries when the website reports a HTTP status 599.		300000	✓	✓	✓	
download-error-599-network-connect-timeout-sleep-multiplicator	Multiplication factor for sleep between retries when the website reports a HTTP status 599.		2	✓	✓	✓	
download-error-argument-exception-max-tries	Maximum number of tries when an argument exception is returned when downloading a blob.		10	✓	✓	✓	
download-error-argument-exception-sleep-initial-ms	Initial sleep in milliseconds between retries when an argument exception is returned when downloading a blob.		10000	✓	✓	✓	
download-error-argument-exception-sleep-max-ms	Maximum sleep in milliseconds between retries when an argument exception is returned when downloading a blob.		300000	✓	✓	✓	
download-error-argument-exception-sleep-multiplicator	Multiplication factor for sleep between retries when an argument exception is returned when downloading a blob.		2	✓	✓	✓	
download-error-internet-down-max-	Maximum number of tries when the Internet connection seems down		10	✓	✓	✓	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
tries	during retrieval of data.						
download-error-internet-down-sleep-initial-ms	Initial sleep in milliseconds between retries when the Internet connection seems down during retrieval of data.		10000	✓	✓	✓	
download-error-internet-down-sleep-max-ms	Maximum sleep in milliseconds between retries when the Internet connection seems down during retrieval of data.		300000	✓	✓	✓	
download-error-internet-down-sleep-multiplicator	Multiplication factor for sleep between retries when the Internet connection seems down during retrieval of data.		2	✓	✓	✓	
download-error-io-exception-max-tries	Maximum number of tries when a network I/O connection failure occurs during retrieval of data.		10	✓	✓	✓	
download-error-io-exception-sleep-initial-ms	Initial sleep in milliseconds between retries when a network I/O connection failure occurs during retrieval of data.		10000	✓	✓	✓	
download-error-io-exception-sleep-max-ms	Maximum sleep in milliseconds between retries when a network I/O connection failure occurs during retrieval of data.		300000	✓	✓	✓	
download-error-io-exception-sleep-multiplicator	Multiplication factor for sleep between retries when a network I/O connection failure occurs during retrieval of data.		2	✓	✓	✓	
download-error-json-exception-max-tries	Maximum number of tries when an invalid JSON body is returned.		3	✓	✓	✓	
download-error-json-exception-sleep-initial-ms	Initial sleep in milliseconds between retries when an invalid JSON body is returned.		1000	✓	✓	✓	
download-error-json-exception-sleep-max-ms	Maximum sleep in milliseconds between retries when an invalid JSON body is returned.		10000	✓	✓	✓	
download-error-json-exception-sleep-multiplicator	Multiplication factor for sleep between retries when an invalid JSON body is returned.		2	✓	✓	✓	
download-error-name-resolution-failure-max-tries	Maximum number of tries when the host name could not be resolved during retrieval of data.		5	✓	✓	✓	
download-error-name-resolution-failure-sleep-initial-ms	Initial sleep in milliseconds between retries when the host name could not be resolved during retrieval of data.		5000	✓	✓	✓	
download-error-name-resolution-failure-sleep-max-ms	Maximum sleep in milliseconds between retries when the host name could not be resolved during retrieval of data.		5000	✓	✓	✓	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
download-error-name-resolution-failure-sleep-multiplicator	{res:itgen_pae_download_error_name_resolution_failure_sleep_multiplicator}		1	✓	✓	✓	
download-error-other-exception-max-tries	Maximum number of tries when an unqualified error occurs during retrieval of data.		3	✓	✓	✓	
download-error-other-exception-sleep-initial-ms	Initial sleep in milliseconds between retries when an unqualified error ocurrs during retrieval of data.		10000	✓	✓	✓	
download-error-other-exception-sleep-max-ms	Maximum sleep in milliseconds between retries when an unqualified error ocurrs during retrieval of data.		300000	✓	✓	✓	
download-error-other-exception-sleep-multiplicator	Multiplication factor for sleep between retries when an unqualified error ocurrs during retrieval of data.		2	✓	✓	✓	
download-error-socket-exception-max-tries	Maximum number of tries when the network connection is forcible dropped during retrieval of data.		10	✓	✓	✓	
download-error-socket-exception-sleep-initial-ms	Initial sleep in milliseconds between retries when the network connection is forcible dropped during retrieval of data.		10000	✓	✓	✓	
download-error-socket-exception-sleep-max-ms	Maximum sleep in milliseconds between retries when the network connection is forcible dropped during retrieval of data.		300000	✓	✓	✓	
download-error-socket-exception-sleep-multiplicator	Multiplication factor for sleep between retries when the network connection is forcible dropped during retrieval of data.		2	✓	✓	✓	
download-error-web-exception-max-tries	Maximum number of tries when a web connection failure occurs during retrieval of data.		10	✓	✓	✓	
download-error-web-exception-sleep-initial-ms	Initial sleep in milliseconds between retries when a web connection failure occurs during retrieval of data.		10000	✓	✓	✓	
download-error-web-exception-sleep-max-ms	Maximum sleep in milliseconds between retries when a web connection failure occurs during retrieval of data.		300000	✓	✓	✓	
download-error-web-exception-sleep-multiplicator	Multiplication factor for sleep between retries when a web connection failure occurs during retrieval of data.		2	✓	✓	✓	
download-error-web-not-found-max-tries	{res:itgen_pae_download_error_web_not_found_max_tries}		1	✓	✓	✓	
download-error-web-not-found-	{res:itgen_pae_download_error_web_not_found_sleep_initial_ms}		10000	✓	✓	✓	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
sleep-initial-ms							
download-error-web-not-found-sleep-max-ms	{res:itgen_pae_download_error_web_not_found_sleep_max_ms}		300000	✓	✓	✓	
download-error-web-not-found-sleep-multiplicator	{res:itgen_pae_download_error_web_not_found_sleep_multiplicator}		2	✓	✓	✓	
download-error-web-not-implemented-max-tries	Maximum number of tries when the connection reports not implemented.		1	✓	✓	✓	
download-error-web-not-implemented-sleep-initial-ms	Initial sleep in milliseconds between retries when the connection reports not implemented.		10000	✓	✓	✓	
download-error-web-not-implemented-sleep-max-ms	Maximum sleep in milliseconds between retries when the connection reports not implemented.		300000	✓	✓	✓	
download-error-web-not-implemented-sleep-multiplicator	Multiplication factor for sleep between retries when the connection reports not implemented.		2	✓	✓	✓	
download-error-web-timeout-max-tries	Maximum number of tries when the connection reports a timeout.		10	✓	✓	✓	
download-error-web-timeout-sleep-initial-ms	Initial sleep in milliseconds between retries when the connection reports a timeout.		1000	✓	✓	✓	
download-error-web-timeout-sleep-max-ms	Maximum sleep in milliseconds between retries when the connection reports a timeout.		30000	✓	✓	✓	
download-error-web-timeout-sleep-multiplicator	Multiplication factor for sleep between retries when the connection reports a timeout.		2	✓	✓	✓	
download-error-web-unauthorized-max-tries	Maximum number of tries when the connection reports an unauthorized error.		1	✓	✓	✓	
download-error-web-unauthorized-sleep-initial-ms	Initial sleep in milliseconds between retries when the connection reports an unauthorized error.		10000	✓	✓	✓	
download-error-web-unauthorized-sleep-max-ms	Maximum sleep in milliseconds between retries when the connection reports an unauthorized error.		300000	✓	✓	✓	
download-error-web-unauthorized-sleep-multiplicator	Multiplication factor for sleep between retries when the connection reports an unauthorized error.		2	✓	✓	✓	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
download-error-web-unknown-max-tries	{res:itgen_pae_download_error_web_unknown_max_tries}		10	✓	✓	✓	
download-error-web-unknown-sleep-initial-ms	{res:itgen_pae_download_error_web_unknown_sleep_initial_ms}		5000	✓	✓	✓	
download-error-web-unknown-sleep-max-ms	{res:itgen_pae_download_error_web_unknown_sleep_max_ms}		30000	✓	✓	✓	
download-error-web-unknown-sleep-multiplicator	{res:itgen_pae_download_error_web_unknown_sleep_multiplicator}		2	✓	✓	✓	
force-case-sensitive-identifiers	Consider identifiers as case-sensitive independent of the platform capabilities.	Shared	False	✓	✓	✓	
forced-casing-identifiers	Forced casing of identifiers. Choose from: Unset, Lower, Upper and Mixed.	Shared		✓	✓	✓	
http-disk-cache-compression-level	Compression level for the HTTP disk cache, ranging from 1 (little) to 9 (intense). Default is 5.		5	✓	✓	✓	
http-disk-cache-directory	Directory where HTTP cache is stored.		C:\Users\guido\Invantive\Cache\http\guido\shared	✓	✓	✓	
http-disk-cache-ignore-write-errors	Whether to ignore write errors to disk cache.		False	✓	✓	✓	
http-disk-cache-max-age-sec	Maximum acceptable age in seconds for use of data in the HTTP disk cache.		2592000	✓	✓	✓	
http-get-timeout-max-ms	HTTP GET maximum timeout on retry (ms).		24000	✓	✓	✓	
http-get-timeout-ms	HTTP GET timeout (ms).		56000	✓	✓	✓	
http-memory-cache-compression-level	Compression level for the HTTP memory cache, ranging from 1 (little) to 9 (intense). Default is 5.		5	✓	✓	✓	
http-memory-cache-max-age-sec	Maximum acceptable age in seconds for use of data in the HTTP memory cache.		14400	✓	✓	✓	
http-post-timeout-max-ms	HTTP POST maximum timeout on retry (ms).		58000	✓	✓	✓	
http-post-timeout-ms	HTTP POST timeout (ms).		57000	✓	✓	✓	
http-public-disk-cache-directory	{res:itgen_pae_http_public_disk_cache_directory}		C:\Users\guido\Invantive\Cache\public	✓	✓	✓	
ignore-http-400-errors	Ignore HTTP 400 errors when exchanging results with the HTTP		False	✓	✓	✓	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
	endpoint.						
ignore-http-401-errors	Ignore HTTP 401 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-402-errors	Ignore HTTP 402 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-403-errors	Ignore HTTP 403 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-404-errors	Ignore HTTP 404 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-422-errors	Ignore HTTP 422 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-429-errors	Ignore HTTP 429 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-500-errors	Ignore HTTP 500 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-502-errors	Ignore HTTP 502 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-503-errors	Ignore HTTP 503 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
invalid-json-on-get-max-tries	Maximum number of tries when the JSON received on GET is invalid.		1	✓	✓	✓	
invalid-json-on-get-sleep-initial-ms	Initial sleep in milliseconds between retries when the JSON received on GET is invalid.		1000	✓	✓	✓	
invalid-json-on-get-sleep-max-ms	Maximum sleep in milliseconds between retries when the JSON received on GET is invalid.		10000	✓	✓	✓	
invalid-json-on-get-sleep-multiplicator	Multiplication factor for sleep between retries when the JSON received on GET is invalid.		2	✓	✓	✓	
invalid-json-on-post-max-tries	Maximum number of tries when the JSON received on POST is invalid.		1	✓	✓	✓	
invalid-json-on-post-sleep-initial-ms	Initial sleep in milliseconds between retries when the JSON received on POST is invalid.		1000	✓	✓	✓	
invalid-json-on-post-sleep-max-ms	Maximum sleep in milliseconds between retries when the JSON received on POST is invalid.		10000	✓	✓	✓	
invalid-json-on-post-sleep-multiplicator	Multiplication factor for sleep between retries when the JSON		2	✓	✓	✓	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
	received on POST is invalid.						
invantive-sql-compress-sparse-arrays	Whether to compress sparse arrays in result sets during compression.	SQL Engine V1	True	✓	✓	✓	
invantive-sql-correct-invalid-date	Whether to correct dates considered invalid since they are before 01-01-1753. When nullable, they are removed. Otherwise they are replaced by 01-01-1753.	SQL Engine V1	False	✓	✓	✓	
invantive-sql-execution-profile-disk-path	{res:itgen_pae_invantive_sql_execution_profile_disk_path}	SQL Engine V1		✓	✓	✓	
invantive-sql-execution-profile-to-disk	{res:itgen_pae_invantive_sql_execution_profile_to_disk}	SQL Engine V1	False	✓	✓	✓	
invantive-sql-forward-filters-to-data-containers	Whether to forward filters to data containers.	SQL Engine V1	True	✓	✓	✓	
invantive-sql-share-byte-arrays	Whether to share the memory used by identical byte arrays in result sets during compression.	SQL Engine V1	True	✓	✓	✓	
invantive-sql-share-strings	Whether to share the memory used by identical strings in result sets during compression.	SQL Engine V1	True	✓	✓	✓	
invantive-sql-shuffle-fetch-results-data-containers	Whether to shuffle results fetched from data containers.	SQL Engine V1	False	✓	✓	✓	
invantive-use-cache	Whether to cache the results of a query.	SQL Engine V1	True	✓	✓	✓	
log-native-calls-to-disk-max-events	Maximum number of call events to register from last activation.	Shared		✓	✓	✓	
log-native-calls-to-disk-max-seconds	Maximum number of seconds to register calls from last activation.	Shared		✓	✓	✓	
log-native-calls-to-disk-on-error	Registers native calls to data container backend as disk files when the call raised an error.	Shared	False	✓	✓	✓	
log-native-calls-to-disk-on-success	Registers native calls to data container backend as disk files when the call raised no error.	Shared	False	✓	✓	✓	
log-native-calls-to-trace	Log native calls to data container backend on the trace.	Shared	False	✓	✓	✓	
maximum-length-identifiers	Non-default maximum length in characters of identifier names.	Shared		✓	✓	✓	
max-url-length-accepted	The maximum accepted URL length before raising an error.		8000	✓	✓	✓	
max-url-length-desired	The maximum desired URL length.		8000	✓	✓	✓	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
minimum-length-text	Extend all text columns to this length to allow processing of XML that uses longer text values than the XSD specifies.	XML		✓			✓
oauth-unauthorized-max-tries	Maximum number of tries when an OAuth exception occurs.		2	✓	✓	✓	
oauth-unauthorized-sleep-initial-ms	Initial sleep in milliseconds between OAuth reauthentication tries when the OAuth authentication fails.		10000	✓	✓	✓	
oauth-unauthorized-sleep-max-ms	Maximum sleep in milliseconds between OAuth reauthentication tries when the OAuth authentication fails.		1000	✓	✓	✓	
oauth-unauthorized-sleep-multiplicator	Multiplication factor for sleep between OAuth reauthentication tries when the OAuth authentication fails.		2	✓	✓	✓	
partition-slot-based-rate-limit-length-ms	Total length in milliseconds across all slots of a partition-based rate limit.	Shared	60000	✓		✓	
partition-slot-based-rate-limit-slots	Number of slots per partition-based rate limit. Null means no slot-based rate limit.	Shared		✓		✓	
pre-request-delay-ms	Pre-request delay in milliseconds per request.	Shared	0	✓	✓	✓	
requested-maximum-usable-partitions	Requested Maximum Usable Partitions	Shared		✓	✓	✓	
requested-page-size	Preferred number of rows to exchange per round trip; only effective on limited platforms such as AFAS Online.	Shared		✓	✓	✓	
requests-parallel-max	Maximum number of parallel data requests from individual partitions on the data container.	Shared	32	✓	✓	✓	
result-set-memory-cache	Action: set to 'empty' to empty.	XML			✓		
simulate-http-400-errors	Simulate HTTP 400 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
simulate-http-400-errors-percentage	Percentage of simulated HTTP 400 errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-401-errors	Simulate HTTP 401 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
simulate-http-401-errors-percentage	Percentage of simulated HTTP 401 errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-403-errors	Simulate HTTP 403 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
simulate-http-403-errors-percentage	Percentage of simulated HTTP 403 errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-408-errors	Simulate HTTP 408 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
simulate-http-408-errors-percentage	Percentage of simulated HTTP 408 errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-429-errors	Simulate HTTP 429 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
simulate-http-429-errors-percentage	Percentage of simulated HTTP 429 errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-500-errors	Simulate HTTP 500 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
simulate-http-500-errors-percentage	Percentage of simulated HTTP 500 errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-502-errors	Simulate HTTP 502 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
simulate-http-502-errors-percentage	Percentage of simulated HTTP 502 errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-503-errors	Simulate HTTP 503 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
simulate-http-503-errors-percentage	Percentage of simulated HTTP 503 errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-504-errors	{res:itgen_pae_simulate_http_504_errors}		False	✓	✓	✓	
simulate-http-504-errors-percentage	{res:itgen_pae_simulate_http_504_errors_percentage}		0	✓	✓	✓	
simulate-http-522-errors	{res:itgen_pae_simulate_http_522_errors}		False	✓	✓	✓	
simulate-http-522-errors-percentage	{res:itgen_pae_simulate_http_522_errors_percentage}		0	✓	✓	✓	
simulate-http-524-errors	{res:itgen_pae_simulate_http_524_errors}		False	✓	✓	✓	
simulate-http-524-errors-percentage	{res:itgen_pae_simulate_http_524_errors_percentage}		0	✓	✓	✓	
simulate-http-protocol-errors	Simulate HTTP protocol errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
simulate-http-protocol-errors-percentage	Percentage of simulated HTTP protocol errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-timeout-errors	Simulate HTTP timeout errors when exchanging results with the HTTP endpoint..		False	✓	✓	✓	
simulate-http-timeout-errors-percentage	Percentage of simulated HTTP timeout errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
slot-based-rate-limit-length-ms	Total length in milliseconds across all slots of a slot-based rate limit.	Shared	60000	✓		✓	
slot-based-rate-limit-slots	Number of slots of a slot-based rate limit. Null means no slot-based rate limit.	Shared		✓		✓	
standardize-identifiers	Rewrite all identifiers to the preferred standards as configured by standardize-identifiers-casing and maximum-length-identifiers.	Shared	True	✓	✓	✓	
standardize-identifiers-casing	Rewrite all identifiers to the recommended standard platform-specific casing when changing a data model on a case-dependent platform.	Shared	True	✓	✓	✓	
use-http-disk-cache	Combination of use-http-disk-cache-read and use-http-disk-cache-write.			✓	✓	✓	
use-http-disk-cache-read	Whether to use HTTP responses from previous queries stored on disk to answer the current query.		False	✓	✓	✓	
use-http-disk-cache-write	Whether to memorize HTTP responses on disk.		False	✓	✓	✓	
use-http-memory-cache	Combination of use-http-memory-cache-read and use-http-memory-cache-write.			✓	✓	✓	
use-http-memory-cache-read	Whether to use HTTP responses from previous queries stored in memory that can answer the current query.		True	✓	✓	✓	
use-http-memory-cache-write	Whether to memorize HTTP responses from previous queries for use by future queries.		True	✓	✓	✓	
use-metadata-memory-cache	Whether to use the metadata in memory calculated previously.	XML	True	✓	✓	✓	
use-result-memory-cache	Whether to use result sets cached in memory from previous queries that can answer the current query.	XML	False	✓	✓	✓	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

3 Catalog: Severa
3.1 Schemas
3.1.1 Schema: API
3.1.1.1 Tables
AccessRightsProfileByGUID: Visma Severa Access Rights Profile by GUID

Catalog: Severa

Schema: API

Label: Access Rights Profile by GUID

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

Filters on API Server: guid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AccessRightsProfileByGUID. 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
guid		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
NAME	varchar2	Name		
RIGHTS_ACCOUNT	varchar2			
RIGHTS_ACCOUNTDELETE	boolean			
RIGHTS_ADMINISTRATOR	varchar2			
RIGHTS_CASE	varchar2			

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
RIGHTS_CASEDELETE	boolean			
RIGHTS_CASEOWNERWORKHOURAPPROVAL	boolean			
RIGHTS_SCHEDULEJOBS	varchar2			
RIGHTS_SHARING	boolean			
RIGHTS_TRAVELREIMBURSEMENT	varchar2			
RIGHTS_USERS	varchar2			
RIGHTS_WORKHOURAPPROVAL	varchar2			

AccessRightsProfiles: Visma Severa Access Rights Profiles

Catalog: Severa

Schema: API

Label: Access Rights Profiles

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

Table Columns

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

Name	Data Type	Label	Required	Documentation
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
NAME	varchar2	Name		
RIGHTS_ACCOUNT	varchar2			
RIGHTS_ACCOUNTDELETE	boolean			
RIGHTS_ADMINISTRATOR	varchar2			
RIGHTS_CASE	varchar2			
RIGHTS_CASEDELETE	boolean			
RIGHTS_CASEOWNERWORKHOURAPPROVAL	boolean			
RIGHTS_SCHEDULEJOBS	varchar2			
RIGHTS_SHARING	boolean			
RIGHTS_TRAVELREIMBURSEMENT	varchar2			
RIGHTS_USERS	varchar2			
RIGHTS_WORKHOURAPPROVAL	varchar2			

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

AccountByGUID: Visma Severa Account by GUID

Catalog: Severa

Schema: API

Label: Account by GUID

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

Filters on API Server: accountGuid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AccountByGUID. 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
accountGuid		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGROUPGUIDS	varchar2			
ACCOUNTOWNERUSERGUID	varchar2	Account Owner User GUID		
ACCOUNTRATING	number	Account Rating		
COMPANYGUID	varchar2	Company GUID		
CURRENCYCODE	varchar2	Currency Code		
CURRENCYGUID	varchar2	Currency GUID		
EINVOICEADDRESS	varchar2	e-invoice Address		
EINVOICEOPERATOR	varchar2	e-invoice Operator		
GUID	varchar2	GUID		
INSERTTS	datetime	Inserted		
INVOICINGVAT	number	Invoicing VAT		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
ISACTIVE	boolean	Is Active		
ISINTERNAL	boolean	Is Internal		
ISPRIVATEPERSON	boolean			
LANGUAGECODE	varchar2	Language Code		
LANGUAGEGUID	varchar2	Language GUID		
NAME	varchar2	Name		
NOTES	varchar2	Notes		
NUMBER	number	Number		
OVERDUEINTEREST	number	Overdue Interest		
PAYMENTTERM	number	Payment Term		
PRICELISTGUID	varchar2	Pricelist GUID		
REVERSECHARGE	boolean	Reverse Charge		
REVERSECHARGEDESCRIPTION	varchar2	Reverse Charge Description		
UPDATEDTS	datetime	Updated		
URLTOACCOUNT	varchar2	URL to Account		

AccountByName: Visma Severa Account by Name

Catalog: Severa

Schema: API

Label: Account by Name

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

Filters on API Server: accountName

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AccountByName. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
accountName		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGROUPGUIDS	varchar2			
ACCOUNTOWNERUSERGUID	varchar2	Account Owner User GUID		
ACCOUNTRATING	number	Account Rating		
COMPANYGUID	varchar2	Company GUID		
CURRENCYCODE	varchar2	Currency Code		
CURRENCYGUID	varchar2	Currency GUID		
EINVOICEADDRESS	varchar2	e-invoice Address		
EINVOICEOPERATOR	varchar2	e-invoice Operator		
GUID	varchar2	GUID		
INSERTTS	datetime	Inserted		
INVOICINGVAT	number	Invoicing VAT		
ISACTIVE	boolean	Is Active		
ISINTERNAL	boolean	Is Internal		
ISPRIVATEPERSON	boolean			
LANGUAGECODE	varchar2	Language Code		
LANGUAGEGUID	varchar2	Language GUID		
NAME	varchar2	Name		
NOTES	varchar2	Notes		
NUMBER	number	Number		
OVERDUEINTEREST	number	Overdue Interest		
PAYMENTTERM	number	Payment Term		
PRICELISTGUID	varchar2	Pricelist GUID		
REVERSECHARGE	boolean	Reverse Charge		
REVERSECHARGEDESCRIPTION	varchar2	Reverse Charge Description		
UPDATEDTS	datetime	Updated		
URLTOACCOUNT	varchar2	URL to Account		

AccountByNumber: Visma Severa Account by Number

Catalog: Severa

Schema: API

Label: Account by Number

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Filters on API Server: accountNumber

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AccountByNumber. 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
accountNumber		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGROUPGUIDS	varchar2			
ACCOUNTOWNERUSERGUID	varchar2	Account Owner User GUID		
ACCOUNTRATING	number	Account Rating		
COMPANYGUID	varchar2	Company GUID		
CURRENCYCODE	varchar2	Currency Code		
CURRENCYGUID	varchar2	Currency GUID		
EINVOICEADDRESS	varchar2	e-invoice Address		
EINVOICEOPERATOR	varchar2	e-invoice Operator		
GUID	varchar2	GUID		
INSERTTS	datetime	Inserted		
INVOICINGVAT	number	Invoicing VAT		
ISACTIVE	boolean	Is Active		
ISINTERNAL	boolean	Is Internal		
ISPRIVATEPERSON	boolean			
LANGUAGECODE	varchar2	Language Code		
LANGUAGEGUID	varchar2	Language GUID		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
NAME	varchar2	Name		
NOTES	varchar2	Notes		
NUMBER	number	Number		
OVERDUEINTEREST	number	Overdue Interest		
PAYMENTTERM	number	Payment Term		
PRICELISTGUID	varchar2	Pricelist GUID		
REVERSECHARGE	boolean	Reverse Charge		
REVERSECHARGEDESCRIPTION	varchar2	Reverse Charge Description		
UPDATEDTS	datetime	Updated		
URLTOACCOUNT	varchar2	URL to Account		

AccountByVatNumber: Visma Severa Account by VAT Number

Catalog: Severa

Schema: API

Label: Account by VAT Number

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

Filters on API Server: vatNumber

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AccountByVatNumber. 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
vatNumber		☐		

Columns of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGROUPGUIDS	varchar2			
ACCOUNTOWNERUSERGUID	varchar2	Account Owner User GUID		
ACCOUNTRATING	number	Account Rating		
COMPANYGUID	varchar2	Company GUID		
CURRENCYCODE	varchar2	Currency Code		
CURRENCYGUID	varchar2	Currency GUID		
EINVOICEADDRESS	varchar2	e-invoice Address		
EINVOICEOPERATOR	varchar2	e-invoice Operator		
GUID	varchar2	GUID		
INSERTTS	datetime	Inserted		
INVOICINGVAT	number	Invoicing VAT		
ISACTIVE	boolean	Is Active		
ISINTERNAL	boolean	Is Internal		
ISPRIVATEPERSON	boolean			
LANGUAGECODE	varchar2	Language Code		
LANGUAGEGUID	varchar2	Language GUID		
NAME	varchar2	Name		
NOTES	varchar2	Notes		
NUMBER	number	Number		
OVERDUEINTEREST	number	Overdue Interest		
PAYMENTTERM	number	Payment Term		
PRICELISTGUID	varchar2	Pricelist GUID		
REVERSECHARGE	boolean	Reverse Charge		
REVERSECHARGEDESCRIPTION	varchar2	Reverse Charge Description		
UPDATEDTS	datetime	Updated		
URLTOACCOUNT	varchar2	URL to Account		

AccountGroups: Visma Severa Account Groups

Catalog: Severa

Schema: API

Label: Account Groups

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

Filters on API Server: options

Parameters of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The following parameters can be used to control the behaviour of the table function AccountGroups. 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
options		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
NAME	varchar2	Name		
PARENTGUID	varchar2	Parent GUID		

AccountGroupsByAccountGUID: Visma Severa Account Group by Account GUID

Catalog: Severa

Schema: API

Label: Account Group by Account GUID

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

Filters on API Server: accountGuid- options

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AccountGroupsByAccountGUID. 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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
accountGuid		☐		
options		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
NAME	varchar2	Name		
PARENTGUID	varchar2	Parent GUID		

Accounts: Visma Severa Accounts

Catalog: Severa

Schema: API

Label: Accounts

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

Filters on API Server: options- startFromIndex- count

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function Accounts. 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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
count		☐		
options		☐		
startFromIndex		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
GUID	varchar2	GUID		
NAME	varchar2	Name		
NUMBER	number	Number		
VATNUMBER	varchar2	VAT Number		

AccountsByCompanyGUID: Visma Severa Accounts by Company GUID

Catalog: Severa

Schema: API

Label: Accounts by Company GUID

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

Filters on API Server: companyGuid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AccountsByCompanyGUID. 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
companyGuid		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGROUPGUIDS	varchar2			
ACCOUNTOWNERUSERGUID	varchar2	Account Owner User GUID		
ACCOUNTRATING	number	Account Rating		
COMPANYGUID	varchar2	Company GUID		
CURRENCYCODE	varchar2	Currency Code		
CURRENCYGUID	varchar2	Currency GUID		
EINVOICEADDRESS	varchar2	e-invoice Address		
EINVOICEOPERATOR	varchar2	e-invoice Operator		
GUID	varchar2	GUID		
INSERTTS	datetime	Inserted		
INVOICINGVAT	number	Invoicing VAT		
ISACTIVE	boolean	Is Active		
ISINTERNAL	boolean	Is Internal		
ISPRIVATEPERSON	boolean			
LANGUAGECODE	varchar2	Language Code		
LANGUAGEGUID	varchar2	Language GUID		
NAME	varchar2	Name		
NOTES	varchar2	Notes		
NUMBER	number	Number		
OVERDUEINTEREST	number	Overdue Interest		
PAYMENTTERM	number	Payment Term		
PRICELISTGUID	varchar2	Pricelist GUID		
REVERSECHARGE	boolean	Reverse Charge		
REVERSECHARGEDESCRIPTION	varchar2	Reverse Charge Description		
UPDATEDTS	datetime	Updated		
URLTOACCOUNT	varchar2	URL to Account		

AccountsByInsertDate: Visma Severa Accounts by Insert Date

Catalog: Severa

Schema: API

Label: Accounts by Insert Date

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Filters on API Server: insertStartDate- insertEndDate

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AccountsByInsertDate. 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
insertEndDate		☐		
insertStartDate		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGROUPGUIDS	varchar2			
ACCOUNTOWNERUSERGUID	varchar2	Account Owner User GUID		
ACCOUNTRATING	number	Account Rating		
COMPANYGUID	varchar2	Company GUID		
CURRENCYCODE	varchar2	Currency Code		
CURRENCYGUID	varchar2	Currency GUID		
EINVOICEADDRESS	varchar2	e-invoice Address		
EINVOICEOPERATOR	varchar2	e-invoice Operator		
GUID	varchar2	GUID		
INSERTTS	datetime	Inserted		
INVOICINGVAT	number	Invoicing VAT		
ISACTIVE	boolean	Is Active		
ISINTERNAL	boolean	Is Internal		
ISPRIVATEPERSON	boolean			
LANGUAGECODE	varchar2	Language Code		
LANGUAGEGUID	varchar2	Language GUID		
NAME	varchar2	Name		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
NOTES	varchar2	Notes		
NUMBER	number	Number		
OVERDUEINTEREST	number	Overdue Interest		
PAYMENTTERM	number	Payment Term		
PRICELISTGUID	varchar2	Pricelist GUID		
REVERSECHARGE	boolean	Reverse Charge		
REVERSECHARGEDESCRIPTION	varchar2	Reverse Charge Description		
UPDATEDTS	datetime	Updated		
URLTOACCOUNT	varchar2	URL to Account		

AccountsChangedSince: Visma Severa Accounts Changed since

Catalog: Severa

Schema: API

Label: Accounts Changed since

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

Filters on API Server: startDateTime- options

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AccountsChangedSince. 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
options		☐		
startDateTime		☐		

Columns of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGROUPGUIDS	varchar2			
ACCOUNTOWNERUSERGUID	varchar2	Account Owner User GUID		
ACCOUNTRATING	number	Account Rating		
COMPANYGUID	varchar2	Company GUID		
CURRENCYCODE	varchar2	Currency Code		
CURRENCYGUID	varchar2	Currency GUID		
EINVOICEADDRESS	varchar2	e-invoice Address		
EINVOICEOPERATOR	varchar2	e-invoice Operator		
GUID	varchar2	GUID		
INSERTTS	datetime	Inserted		
INVOICINGVAT	number	Invoicing VAT		
ISACTIVE	boolean	Is Active		
ISINTERNAL	boolean	Is Internal		
ISPRIVATEPERSON	boolean			
LANGUAGECODE	varchar2	Language Code		
LANGUAGEGUID	varchar2	Language GUID		
NAME	varchar2	Name		
NOTES	varchar2	Notes		
NUMBER	number	Number		
OVERDUEINTEREST	number	Overdue Interest		
PAYMENTTERM	number	Payment Term		
PRICELISTGUID	varchar2	Pricelist GUID		
REVERSECHARGE	boolean	Reverse Charge		
REVERSECHARGEDESCRIPTION	varchar2	Reverse Charge Description		
UPDATEDTS	datetime	Updated		
URLTOACCOUNT	varchar2	URL to Account		

AccountsChangedSinceOpts: Visma Severa Accounts Changed since Opts.

Catalog: Severa

Schema: API

Label: Accounts Changed since Opts.

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

Filters on API Server: startDateTime- options

Parameters of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The following parameters can be used to control the behaviour of the table function AccountsChangedSinceOpts. 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
options		☐		
startDateTime		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGROUPGUIDS	varchar2			
ACCOUNTOWNERUSERGUID	varchar2	Account Owner User GUID		
ACCOUNTRATING	number	Account Rating		
COMPANYGUID	varchar2	Company GUID		
CURRENCYCODE	varchar2	Currency Code		
CURRENCYGUID	varchar2	Currency GUID		
EINVOICEADDRESS	varchar2	e-invoice Address		
EINVOICEOPERATOR	varchar2	e-invoice Operator		
GUID	varchar2	GUID		
INSERTTS	datetime	Inserted		
INVOICINGVAT	number	Invoicing VAT		
ISACTIVE	boolean	Is Active		
ISINTERNAL	boolean	Is Internal		
ISPRIVATEPERSON	boolean			
LANGUAGECODE	varchar2	Language Code		
LANGUAGEGUID	varchar2	Language GUID		
NAME	varchar2	Name		
NOTES	varchar2	Notes		
NUMBER	number	Number		
OVERDUEINTEREST	number	Overdue Interest		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
PAYMENTTERM	number	Payment Term		
PRICELISTGUID	varchar2	Pricelist GUID		
REVERSECHARGE	boolean	Reverse Charge		
REVERSECHARGEDESCRIPTION	varchar2	Reverse Charge Description		
UPDATEDTS	datetime	Updated		
URLTOACCOUNT	varchar2	URL to Account		

ActiveLeadSources: Visma Severa Active Lead Sources

Catalog: Severa

Schema: API

Label: Active Lead Sources

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

Table Columns

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

Name	Data Type	Label	Required	Documentation
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
NAME	varchar2	Name		

ActivitiesByDate: Visma Severa Activities by Date

Catalog: Severa

Schema: API

Label: Activities by Date

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

Filters on API Server: startDate- endDate- activityCategory- businessUnitGuid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ActivitiesByDate. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
activityCategory		☐		
businessUnitGuid		☐		
endDate		☐		
startDate		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	varchar2	Account GUID		
ACTIVITYTYPECODE	varchar2	Activity Type Code		
ACTIVITYTYPEGUID	varchar2	Activity Type GUID		
CASEGUID	varchar2	Project GUID		
ENDDATETIME	datetime	End Date Time		
GUID	varchar2	GUID		
ISALLDAY	boolean	Is All Day		
ISCLOSED	boolean	Is Closed		
ISDURATION	boolean	Is Duration		
ISPRIVATE	boolean	Is Private		
ISREADONLY	boolean	Is Read-only		
ISUNASSIGNED	boolean	Is Unassigned		
LOCATION	varchar2	Location		
NAME	varchar2	Name		
NOTES	varchar2	Notes		
OWNERUSERBUSINESSUNITCODE	varchar2	Owner User Business Unit Code		
OWNERUSERCODE	varchar2	Owner User Code		
OWNERUSERFIRSTNAME	varchar2	Owner User First Name		
OWNERUSERGUID	varchar2	Owner User GUID		
OWNERUSERLASTNAME	varchar2	Owner User Last Name		
STARTDATETIME	datetime	Start Date Time		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
TASKGUID	varchar2	Task GUID		

ActivityByGUID: Visma Severa Activity by GUID

Catalog: Severa

Schema: API

Label: Activity by GUID

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

Filters on API Server: activityGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ActivityByGUID. 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
activityGUID		□		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	varchar2	Account GUID		
ACTIVITYTYPECODE	varchar2	Activity Type Code		
ACTIVITYTYPEGUID	varchar2	Activity Type GUID		
CASEGUID	varchar2	Project GUID		
ENDDATETIME	datetime	End Date Time		
GUID	varchar2	GUID		
ISALLDAY	boolean	Is All Day		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
ISCLOSED	boolean	Is Closed		
ISDURATION	boolean	Is Duration		
ISPRIVATE	boolean	Is Private		
ISREADONLY	boolean	Is Read-only		
ISUNASSIGNED	boolean	Is Unassigned		
LOCATION	varchar2	Location		
NAME	varchar2	Name		
NOTES	varchar2	Notes		
OWNERUSERBUSINESSUNITCODE	varchar2	Owner User Business Unit Code		
OWNERUSERCODE	varchar2	Owner User Code		
OWNERUSERFIRSTNAME	varchar2	Owner User First Name		
OWNERUSERGUID	varchar2	Owner User GUID		
OWNERUSERLASTNAME	varchar2	Owner User Last Name		
STARTDATETIME	datetime	Start Date Time		
TASKGUID	varchar2	Task GUID		

ActivityContactParticipants: Visma Severa Activity Contact Participants

Catalog: Severa

Schema: API

Label: Activity Contact Participants

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

Filters on API Server: activityGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ActivityContactParticipants. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
activityGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CONTACTGUID	varchar2	Contact GUID		
NAME	varchar2	Name		
RESOURCEGUID	varchar2	Resource GUID		
STATUS	number	Status		
USERGUID	varchar2	User GUID		

ActivityInstances: Visma Severa Activity Instances

Catalog: Severa

Schema: API

Label: Activity Instances

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

Filters on API Server: userGUID- startsAfter- startsBefore- endsAfter- endsBefore- activityTypeGUID- accountGUID- caseGUID- firstRow- maxRows

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ActivityInstances. 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
accountGUID		☐		
activityTypeGUID		☐		
caseGUID		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
endsAfter		☐		
endsBefore		☐		
firstRow		☐		
maxRows		☐		
startsAfter		☐		
startsBefore		☐		
userGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	varchar2	Account GUID		
ACTIVITYTYPECODE	varchar2	Activity Type Code		
ACTIVITYTYPEGUID	varchar2	Activity Type GUID		
CASEGUID	varchar2	Project GUID		
ENDDATETIME	datetime	End Date Time		
GUID	varchar2	GUID		
ISALLDAY	boolean	Is All Day		
ISCLOSED	boolean	Is Closed		
ISDURATION	boolean	Is Duration		
ISPRIVATE	boolean	Is Private		
ISREADONLY	boolean	Is Read-only		
ISUNASSIGNED	boolean	Is Unassigned		
LOCATION	varchar2	Location		
NAME	varchar2	Name		
NOTES	varchar2	Notes		
OWNERUSERBUSINESSUNITCODE	varchar2	Owner User Business Unit Code		
OWNERUSERCODE	varchar2	Owner User Code		
OWNERUSERFIRSTNAME	varchar2	Owner User First Name		
OWNERUSERGUID	varchar2	Owner User GUID		
OWNERUSERLASTNAME	varchar2	Owner User Last Name		
STARTDATETIME	datetime	Start Date Time		
TASKGUID	varchar2	Task GUID		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

ActivityParticipants: Visma Severa Activity Participants

Catalog: Severa

Schema: API

Label: Activity Participants

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

Filters on API Server: activityGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ActivityParticipants. 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
activityGUID		□		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CONTACTGUID	varchar2	Contact GUID		
NAME	varchar2	Name		
RESOURCEGUID	varchar2	Resource GUID		
STATUS	number	Status		
USERGUID	varchar2	User GUID		

ActivityResourceParticipants: Visma Severa Activity Resource Participants

Catalog: Severa

Schema: API

Label: Activity Resource Participants

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Filters on API Server: activityGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ActivityResourceParticipants. 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
activityGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CONTACTGUID	varchar2	Contact GUID		
NAME	varchar2	Name		
RESOURCEGUID	varchar2	Resource GUID		
STATUS	number	Status		
USERGUID	varchar2	User GUID		

ActivityTypeByGUID: Visma Severa Activity Type by GUID

Catalog: Severa

Schema: API

Label: Activity Type by GUID

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

Filters on API Server: activityTypeGUID

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ActivityTypeByGUID. 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
activityTypeGUID		□		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CATEGORY	varchar2	Category		
CODE	varchar2	Code		
GUID	varchar2	GUID		
ICON	varchar2	Icon		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
ISPAIDLEAVE	boolean	Is Paid Leave		
NAME	varchar2	Name		

ActivityTypes: Visma Severa Activity Types

Catalog: Severa

Schema: API

Label: Activity Types

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

Table Columns

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Name	Data Type	Label	Required	Documentation
CATEGORY	varchar2	Category		
CODE	varchar2	Code		
GUID	varchar2	GUID		
ICON	varchar2	Icon		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
ISPAIDLEAVE	boolean	Is Paid Leave		
NAME	varchar2	Name		

ActivityUserParticipants: Visma Severa Activity User Participants

Catalog: Severa

Schema: API

Label: Activity User Participants

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

Filters on API Server: activityGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ActivityUserParticipants. 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
activityGUID		☐		

Columns of Table Function

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
CONTACTGUID	varchar2	Contact GUID		
NAME	varchar2	Name		
RESOURCEGUID	varchar2	Resource GUID		
STATUS	number	Status		
USERGUID	varchar2	User GUID		

AddCaseMemberUser: Visma Severa Add Project Member User

Catalog: Severa

Schema: API

Label: Add Project Member User

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

Filters on API Server: caseGuid- userGuid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddCaseMemberUser. 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
caseGuid		☐		
userGuid		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
DUMMY	varchar2	Dummy	☑	Dummy column

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

AddNewAccessRightsProfile: Visma Severa Add New Access Rights Profile

Catalog: Severa

Schema: API

Label: Add New Access Rights Profile

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

Filters on API Server: accessRightsProfile_GUID- accessRightsProfile_IsActive- accessRightsProfile_IsDefault- accessRightsProfile_Name- accessRightsProfile_Rights_Account- accessRightsProfile_Rights_AccountDelete- accessRightsProfile_Rights_Administrator- accessRightsProfile_Rights_Case- accessRightsProfile_Rights_CaseDelete- accessRightsProfile_Rights_CaseOwnerWorkHourApproval- accessRightsProfile_Rights_ScheduleJobs- accessRightsProfile_Rights_Sharing- accessRightsProfile_Rights_TravelReimbursement- accessRightsProfile_Rights_Users- accessRightsProfile_Rights_WorkHourApproval

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewAccessRightsProfile. 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
accessRightsProfile_GUID		☐		
accessRightsProfile_IsActive		☐		
accessRightsProfile_IsDefault		☐		
accessRightsProfile_Name		☐		
accessRightsProfile_Rights_Account		☐		
accessRightsProfile_Rights_AccountDelete		☐		
accessRightsProfile_Rights_Administrator		☐		
accessRightsProfile_Rights_Case		☐		
accessRightsProfile_Rights_CaseDelete		☐		
accessRightsProfile_Rights_CaseOwnerWorkHourApproval		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
accessRightsProfile_Rights_ScheduleJobs		☐		
accessRightsProfile_Rights_Sharing		☐		
accessRightsProfile_Rights_TravelReimbursement		☐		
accessRightsProfile_Rights_Users		☐		
accessRightsProfile_Rights_WorkHourApproval		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
NAME	varchar2	Name		
RIGHTS_ACCOUNT	varchar2			
RIGHTS_ACCOUNTDELETE	boolean			
RIGHTS_ADMINISTRATOR	varchar2			
RIGHTS_CASE	varchar2			
RIGHTS_CASEDELETE	boolean			
RIGHTS_CASEOWNERWORKHOURAPPROVAL	boolean			
RIGHTS_SCHEDULEJOBS	varchar2			
RIGHTS_SHARING	boolean			
RIGHTS_TRAVELREIMBURSEMENT	varchar2			
RIGHTS_USERS	varchar2			
RIGHTS_WORKHOURAPPROVAL	varchar2			

AddNewAccount: Visma Severa Add New Account

Catalog: Severa

Schema: API

Label: Add New Account

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

Filters on API Server: accountInfo_AccountGroupGUIDs- accountInfo_AccountOwnerUserGUID- accountInfo_AccountRating-

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

accountInfo_CompanyGUID- accountInfo_CurrencyCode- accountInfo_CurrencyGUID- accountInfo_GUID- accountInfo_InsertTS- accountInfo_InvoicingVat- accountInfo_IsActive- accountInfo_IsInternal- accountInfo_IsPrivatePerson- accountInfo_LanguageCode- accountInfo_LanguageGUID- accountInfo_Name- accountInfo_Notes- accountInfo_Number- accountInfo_OverdueInterest- accountInfo_PaymentTerm- accountInfo_PricelistGUID- accountInfo_ReverseCharge- accountInfo_ReverseChargeDescription- accountInfo_UpdatedTS- accountInfo_UrlToAccount- accountInfo_eInvoiceAddress- accountInfo_eInvoiceOperator- companyInfo_AnnualRevenue- companyInfo_AnnualRevenue2- companyInfo_Email- companyInfo_Employees- companyInfo_GUID- companyInfo_HeadOfficeAddressGUID- companyInfo_HierarchyGUID- companyInfo_IndustryGUID- companyInfo_KvkNumber- companyInfo_LanguageGUID- companyInfo_Name- companyInfo_TimezoneGUID- companyInfo_VatNumber- companyInfo_Website- addressInfo_Addressline- addressInfo_Addressline2- addressInfo_Addressline3- addressInfo_City- addressInfo_CompanyGUID- addressInfo_CountryCode- addressInfo_CountryGUID- addressInfo_CountryName- addressInfo_CountryRegionGUID- addressInfo_CountryRegionName- addressInfo_Fax- addressInfo_GUID- addressInfo_IsBillingAddress- addressInfo_IsPostalAddress- addressInfo_IsVisitAddress- addressInfo_Phone- addressInfo_PostalCode- contactInfo_AccountGUID- contactInfo_AddressGUID- contactInfo_CommunicationMethods- contactInfo_ContactRoleGUID- contactInfo_DateOfBirth- contactInfo_Description- contactInfo_FirstName- contactInfo_GUID- contactInfo_IsActive- contactInfo_JobTitle- contactInfo_Keywords- contactInfo_LanguageGUID- contactInfo_LastName- contactInfo_Salutation- contactInfo_SatisfactionLevel- contactInfo_TimeZoneGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewAccount. 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
accountInfo_AccountGroupGUIDs		☐		
accountInfo_AccountOwnerUserGUID		☐		
accountInfo_AccountRating		☐		
accountInfo_CompanyGUID		☐		
accountInfo_CurrencyCode		☐		
accountInfo_CurrencyGUID		☐		
accountInfo_eInvoiceAddress		☐		
accountInfo_eInvoiceOperator		☐		
accountInfo_GUID		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
accountInfo_InsertTS		☐		
accountInfo_InvoicingVat		☐		
accountInfo_IsActive		☐		
accountInfo_IsInternal		☐		
accountInfo_IsPrivatePerson		☐		
accountInfo_LanguageCode		☐		
accountInfo_LanguageGUID		☐		
accountInfo_Name		☐		
accountInfo_Notes		☐		
accountInfo_Number		☐		
accountInfo_OverdueInterest		☐		
accountInfo_PaymentTerm		☐		
accountInfo_PricelistGUID		☐		
accountInfo_ReverseCharge		☐		
accountInfo_ReverseChargeDescription		☐		
accountInfo_UpdatedTS		☐		
accountInfo_UrlToAccount		☐		
addressInfo_Addressline		☐		
addressInfo_Addressline2		☐		
addressInfo_Addressline3		☐		
addressInfo_City		☐		
addressInfo_CompanyGUID		☐		
addressInfo_CountryCode		☐		
addressInfo_CountryGUID		☐		
addressInfo_CountryName		☐		
addressInfo_CountryRegionGUID		☐		
addressInfo_CountryRegionName		☐		
addressInfo_Fax		☐		
addressInfo_GUID		☐		
addressInfo_IsBillingAddress		☐		
addressInfo_IsPostalAddress		☐		
addressInfo_IsVisitAddress		☐		
addressInfo_Phone		☐		
addressInfo_PostalCode		☐		
companyInfo_AnnualRevenue		☐		
companyInfo_AnnualRevenue2		☐		
companyInfo_Email		☐		
companyInfo_Employees		☐		
companyInfo_GUID		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
companyInfo_HeadOfficeAddressGUID		☐		
companyInfo_HierarchyGUID		☐		
companyInfo_IndustryGUID		☐		
companyInfo_KvkNumber		☐		
companyInfo_LanguageGUID		☐		
companyInfo_Name		☐		
companyInfo_TimezoneGUID		☐		
companyInfo_VatNumber		☐		
companyInfo_Website		☐		
contactInfo_AccountGUID		☐		
contactInfo_AddressGUID		☐		
contactInfo_CommunicationMethods		☐		
contactInfo_ContactRoleGUID		☐		
contactInfo_DateOfBirth		☐		
contactInfo_Description		☐		
contactInfo_FirstName		☐		
contactInfo_GUID		☐		
contactInfo_IsActive		☐		
contactInfo_JobTitle		☐		
contactInfo_Keywords		☐		
contactInfo_LanguageGUID		☐		
contactInfo_LastName		☐		
contactInfo_Salutation		☐		
contactInfo_SatisfactionLevel		☐		
contactInfo_TimeZoneGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGROUPGUIDS	varchar2			
ACCOUNTOWNERUSERGUID	varchar2	Account Owner User GUID		
ACCOUNTRATING	number	Account Rating		
COMPANYGUID	varchar2	Company GUID		
CURRENCYCODE	varchar2	Currency Code		
CURRENCYGUID	varchar2	Currency GUID		
EINVOICEADDRESS	varchar2	e-invoice Address		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
EINVOICEOPERATOR	varchar2	e-invoice Operator		
GUID	varchar2	GUID		
INSERTTS	datetime	Inserted		
INVOICINGVAT	number	Invoicing VAT		
ISACTIVE	boolean	Is Active		
ISINTERNAL	boolean	Is Internal		
ISPRIVATEPERSON	boolean			
LANGUAGECODE	varchar2	Language Code		
LANGUAGEGUID	varchar2	Language GUID		
NAME	varchar2	Name		
NOTES	varchar2	Notes		
NUMBER	number	Number		
OVERDUEINTEREST	number	Overdue Interest		
PAYMENTTERM	number	Payment Term		
PRICELISTGUID	varchar2	Pricelist GUID		
REVERSECHARGE	boolean	Reverse Charge		
REVERSECHARGEDESCRIPTION	varchar2	Reverse Charge Description		
UPDATEDTS	datetime	Updated		
URLTOACCOUNT	varchar2	URL to Account		

AddNewAccountGroup: Visma Severa Add New Account Group

Catalog: Severa

Schema: API

Label: Add New Account Group

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

Filters on API Server: accountGroupInfo_GUID- accountGroupInfo_IsActive- accountGroupInfo_Name- accountGroupInfo_ParentGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewAccountGroup. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
accountGroupInfo_GUID		☐		
accountGroupInfo_IsActive		☐		
accountGroupInfo_Name		☐		
accountGroupInfo_ParentGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
NAME	varchar2	Name		
PARENTGUID	varchar2	Parent GUID		

AddNewAccountGroupMembers: Visma Severa Add New Account Group Members

Catalog: Severa

Schema: API

Label: Add New Account Group Members

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

Filters on API Server: accountGUID- accountGroupGUIDs

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewAccountGroupMembers. 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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
accountGroupGUIDs		☐		
accountGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
DUMMY	varchar2	Dummy	☑	Dummy column

AddNewActivity: Visma Severa Add New Activity

Catalog: Severa

Schema: API

Label: Add New Activity

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

Filters on API Server: activityInfo_AccountGUID- activityInfo_ActivityTypeCode- activityInfo_ActivityTypeGUID- activityInfo_CaseGUID- activityInfo_EndDateTime- activityInfo_GUID- activityInfo_IsAllDay- activityInfo_IsClosed- activityInfo_IsDuration- activityInfo_IsPrivate- activityInfo_IsReadOnly- activityInfo_IsUnassigned- activityInfo_Location- activityInfo_Name- activityInfo_Notes- activityInfo_OwnerUserBusinessUnitCode- activityInfo_OwnerUserCode- activityInfo_OwnerUserFirstName- activityInfo_OwnerUserGUID- activityInfo_OwnerUserLastName- activityInfo_StartDateTime- activityInfo_TaskGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewActivity. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
activityInfo_AccountGUID		☐		
activityInfo_ActivityTypeCode		☐		
activityInfo_ActivityTypeGUID		☐		
activityInfo_CaseGUID		☐		
activityInfo_EndDateTime		☐		
activityInfo_GUID		☐		
activityInfo_IsAllDay		☐		
activityInfo_IsClosed		☐		
activityInfo_IsDuration		☐		
activityInfo_IsPrivate		☐		
activityInfo_IsReadOnly		☐		
activityInfo_IsUnassigned		☐		
activityInfo_Location		☐		
activityInfo_Name		☐		
activityInfo_Notes		☐		
activityInfo_OwnerUserBusinessUnitCode		☐		
activityInfo_OwnerUserCode		☐		
activityInfo_OwnerUserFirstName		☐		
activityInfo_OwnerUserGUID		☐		
activityInfo_OwnerUserLastName		☐		
activityInfo_StartDateTime		☐		
activityInfo_TaskGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	varchar2	Account GUID		
ACTIVITYTYPECODE	varchar2	Activity Type Code		
ACTIVITYTYPEGUID	varchar2	Activity Type GUID		
CASEGUID	varchar2	Project GUID		
ENDDATETIME	datetime	End Date Time		
GUID	varchar2	GUID		
ISALLDAY	boolean	Is All Day		
ISCLOSED	boolean	Is Closed		
ISDURATION	boolean	Is Duration		
ISPRIVATE	boolean	Is Private		
ISREADONLY	boolean	Is Read-only		
ISUNASSIGNED	boolean	Is Unassigned		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
LOCATION	varchar2	Location		
NAME	varchar2	Name		
NOTES	varchar2	Notes		
OWNERUSERBUSINESSUNITCODE	varchar2	Owner User Business Unit Code		
OWNERUSERCODE	varchar2	Owner User Code		
OWNERUSERFIRSTNAME	varchar2	Owner User First Name		
OWNERUSERGUID	varchar2	Owner User GUID		
OWNERUSERLASTNAME	varchar2	Owner User Last Name		
STARTDATETIME	datetime	Start Date Time		
TASKGUID	varchar2	Task GUID		

AddNewActivityContactParticipant: Visma Severa Add New Activity Contact Participant

Catalog: Severa

Schema: API

Label: Add New Activity Contact Participant

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

Filters on API Server: activityGUID- contactGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewActivityContactParticipant. 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
activityGUID		☐		
contactGUID		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ADDNEWACTIVITYCONTACTPARTICIPANTRESULT	boolean			

AddNewActivityResourceParticipant: Visma Severa Add New Activity Resource Participant

Catalog: Severa

Schema: API

Label: Add New Activity Resource Participant

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

Filters on API Server: activityGUID- resourceGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewActivityResourceParticipant. 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
activityGUID		□		
resourceGUID		□		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ADDNEWACTIVITYRESOURCEPARTICIPANTRESULT	boolean			

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

AddNewActivityType: Visma Severa Add New Activity Type

Catalog: Severa

Schema: API

Label: Add New Activity Type

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

Filters on API Server: activityTypeInfo_Category- activityTypeInfo_Code- activityTypeInfo_GUID- activityTypeInfo_Icon- activityTypeInfo_IsActive- activityTypeInfo_IsDefault- activityTypeInfo_IsPaidLeave- activityTypeInfo_Name

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewActivityType. 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
activityTypeInfo_Category		☐		
activityTypeInfo_Code		☐		
activityTypeInfo_GUID		☐		
activityTypeInfo_Icon		☐		
activityTypeInfo_IsActive		☐		
activityTypeInfo_IsDefault		☐		
activityTypeInfo_IsPaidLeave		☐		
activityTypeInfo_Name		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CATEGORY	varchar2	Category		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
CODE	varchar2	Code		
GUID	varchar2	GUID		
ICON	varchar2	Icon		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
ISPAIDLEAVE	boolean	Is Paid Leave		
NAME	varchar2	Name		

AddNewActivityUserParticipant: Visma Severa Add New Activity User Participant

Catalog: Severa

Schema: API

Label: Add New Activity User Participant

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

Filters on API Server: activityGUID- userGUID- memberStatus

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewActivityUserParticipant. 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
activityGUID		☐		
memberStatus		☐		
userGUID		☐		

Columns of Table Function

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
ADDNEWACTIVITYUSERPARTICIPANTRESULT	boolean			

AddNewAddress: Visma Severa Add New Address

Catalog: Severa

Schema: API

Label: Add New Address

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

Filters on API Server: addressInfo_Addressline- addressInfo_Addressline2- addressInfo_Addressline3- addressInfo_City- addressInfo_CompanyGUID- addressInfo_CountryCode- addressInfo_CountryGUID- addressInfo_CountryName- addressInfo_CountryRegionGUID- addressInfo_CountryRegionName- addressInfo_Fax- addressInfo_GUID- addressInfo_IsBillingAddress- addressInfo_IsPostalAddress- addressInfo_IsVisitAddress- addressInfo_Phone- addressInfo_PostalCode

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewAddress. 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
addressInfo_Addressline		☐		
addressInfo_Addressline2		☐		
addressInfo_Addressline3		☐		
addressInfo_City		☐		
addressInfo_CompanyGUID		☐		
addressInfo_CountryCode		☐		
addressInfo_CountryGUID		☐		
addressInfo_CountryName		☐		
addressInfo_CountryRegionGUID		☐		
addressInfo_CountryRegionName		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
addressInfo_Fax		☐		
addressInfo_GUID		☐		
addressInfo_IsBillingAddress		☐		
addressInfo_IsPostalAddress		☐		
addressInfo_IsVisitAddress		☐		
addressInfo_Phone		☐		
addressInfo_PostalCode		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ADDRESSLINE	varchar2	Addressline		
ADDRESSLINE2	varchar2	Addressline 2		
ADDRESSLINE3	varchar2	Addressline 3		
CITY	varchar2	City		
COMPANYGUID	varchar2	Company GUID		
COUNTRYCODE	varchar2	Country Code		
COUNTRYGUID	varchar2	Country GUID		
COUNTRYNAME	varchar2	Country Name		
COUNTRYREGIONGUID	varchar2	Country Region GUID		
COUNTRYREGIONNAME	varchar2	Country Region Name		
FAX	varchar2	Fax		
GUID	varchar2	GUID		
ISBILLINGADDRESS	boolean	Is Billing Address		
ISPOSTALADDRESS	boolean	Is Postal Address		
ISVISITADDRESS	boolean	Is Visit Address		
PHONE	varchar2	Phone		
POSTALCODE	varchar2	Postal Code		

AddNewBillingAndExpenseForecastToCase: Visma Severa Add New Billing and Expense Forecast to Project

Catalog: Severa

Schema: API

Label: Add New Billing and Expense Forecast to Project

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Filters on API Server: caseGuid- billingAndExpenseForecastInfo_BillingForecast- billingAndExpenseForecastInfo_ExpenseForecast- billingAndExpenseForecastInfo_GUID- billingAndExpenseForecastInfo_InhouseRevenueForecast- billingAndExpenseForecastInfo_Month- billingAndExpenseForecastInfo_Notes- billingAndExpenseForecastInfo_OutsourcingRevenueForecast- billingAndExpenseForecastInfo_Year

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewBillingAndExpenseForecastToCase. 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
billingAndExpenseForecastInfo_BillingForecast		☐		
billingAndExpenseForecastInfo_ExpenseForecast		☐		
billingAndExpenseForecastInfo_GUID		☐		
billingAndExpenseForecastInfo_InhouseRevenueForecast		☐		
billingAndExpenseForecastInfo_Month		☐		
billingAndExpenseForecastInfo_Notes		☐		
billingAndExpenseForecastInfo_OutsourcingRevenueForecast		☐		
billingAndExpenseForecastInfo_Year		☐		
caseGuid		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
BILLINGFORECAST	number	Billing Forecast		
EXPENSEFORECAST	number	Expense Forecast		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
GUID	varchar2	GUID		
INHOUSEREVENUEFORECAST	number	Inhouse Revenue Forecast		
MONTH	number	Month		
NOTES	varchar2	Notes		
OUTSOURCINGREVENUEFORECAST	number	Outsourcing Revenue Forecast		
YEAR	number	Year		

AddNewBusinessUnit: Visma Severa Add New Business Unit

Catalog: Severa

Schema: API

Label: Add New Business Unit

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

Filters on API Server: businessUnitInfo_Code- businessUnitInfo_CostCenterGUID- businessUnitInfo_CurrencyCode- businessUnitInfo_CurrencyGUID- businessUnitInfo_FormattingCulture- businessUnitInfo_GUID- businessUnitInfo_InsertTS- businessUnitInfo_IsActive- businessUnitInfo_LanguageCode- businessUnitInfo_LanguageGUID- businessUnitInfo_Name- businessUnitInfo_NextInvoiceNumber- businessUnitInfo_OverDueInterest- businessUnitInfo_ParentBusinessUnitGUID- businessUnitInfo_PaymentTerm- businessUnitInfo_ShowLogoOnInvoices- businessUnitInfo_ShowLogoOnPdfReports- businessUnitInfo_TimeZoneGUID- businessUnitInfo_UpdatedTS- businessUnitInfo_VatNumber

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewBusinessUnit. 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
businessUnitInfo_Code		☐		
businessUnitInfo_CostCenterGUID		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
businessUnitInfo_CurrencyCode		☐		
businessUnitInfo_CurrencyGUID		☐		
businessUnitInfo_FormattingCulture		☐		
businessUnitInfo_GUID		☐		
businessUnitInfo_InsertTS		☐		
businessUnitInfo_IsActive		☐		
businessUnitInfo_LanguageCode		☐		
businessUnitInfo_LanguageGUID		☐		
businessUnitInfo_Name		☐		
businessUnitInfo_NextInvoiceNumber		☐		
businessUnitInfo_OverDueInterest		☐		
businessUnitInfo_ParentBusinessUnitGUID		☐		
businessUnitInfo_PaymentTerm		☐		
businessUnitInfo_Show LogoOnInvoices		☐		
businessUnitInfo_Show LogoOnPdf Reports		☐		
businessUnitInfo_TimeZoneGUID		☐		
businessUnitInfo_UpdatedTS		☐		
businessUnitInfo_VatNumber		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CODE	varchar2	Code		
COSTCENTERGUID	varchar2	Cost Centre GUID		
CURRENCYCODE	varchar2	Currency Code		
CURRENCYGUID	varchar2	Currency GUID		
FORMATTINGCULTURE	varchar2	Formatting Culture		
GUID	varchar2	GUID		
INSERTTS	datetime	Inserted		
ISACTIVE	boolean	Is Active		
LANGUAGECODE	varchar2	Language Code		
LANGUAGEGUID	varchar2	Language GUID		
NAME	varchar2	Name		
NEXTINVOICENUMBER	number	Next Invoice Number		
OVERDUEINTEREST	number	Overdue Interest		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
PARENTBUSINESSUNITGUID	varchar2	Parent Business Unit GUID		
PAYMENTTERM	number	Payment Term		
SHOWLOGOONINVOICES	boolean	Show Logo on Invoices		
SHOWLOGOONPDFREPORTS	boolean	Show Logo on PDF Reports		
TIMEZONEGUID	varchar2	Timezone GUID		
UPDATEDTS	datetime	Updated		
VATNUMBER	varchar2	VAT Number		

AddNewCase: Visma Severa Add New Project

Catalog: Severa

Schema: API

Label: Add New Project

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

Filters on API Server: caseInfo_AccountGUID- caseInfo_AccountName- caseInfo_AccountNumber- caseInfo_BillingAddress- caseInfo_BillingAddressGUID- caseInfo_BillingInternalNotes- caseInfo_BillingNotesBeforeGrid- caseInfo_BusinessUnitGUID- caseInfo_BusinessUnitName- caseInfo_BusinessUnitNumber- caseInfo_CaseCompanyCurrencyGUID- caseInfo_CaseNumber- caseInfo_CaseOwnerUserCode- caseInfo_CaseOwnerUserGUID- caseInfo_ClosedDate- caseInfo_CompletionEstimate- caseInfo_ContactGUID- caseInfo_CostCenterGUID- caseInfo_CostCenterNumber- caseInfo_CurrencyGUID- caseInfo_CurrencyShortform- caseInfo_CurrencyShortformOfCaseCompanyCurrency- caseInfo_CurrentCaseStatusDescription- caseInfo_CurrentCaseStatusGUID- caseInfo_CurrentCaseStatusInsertTs- caseInfo_DeadlineDate- caseInfo_Description- caseInfo_ExpectedValue- caseInfo_GUID- caseInfo_IncludeExceptions- caseInfo_IncludeOverTime- caseInfo_InsertTs- caseInfo_InternalName- caseInfo_InvoiceTemplateGUID- caseInfo_IsClosed- caseInfo_IsDailyAllowanceBillable- caseInfo_IsInternal- caseInfo_IsMileageBillable- caseInfo_IsOtherTravelExpensesBillable- caseInfo_LatestEstimationDate- caseInfo_LeadRating- caseInfo_LeadSourceGUID- caseInfo_Name- caseInfo_OrderNumber- caseInfo_OurReference- caseInfo_OverdueInterest- caseInfo_PaymentTerm- caseInfo_PricelistGUID- caseInfo_PricingRule- caseInfo_PricingType- caseInfo_Priority- caseInfo_Probability- caseInfo_RootTaskGUID- caseInfo_SalesCloseDate- caseInfo_SalesPersonUserCode- caseInfo_SalesPersonUserGUID- caseInfo_SalesProcessGUID- caseInfo_SalesStatusGUID- caseInfo_SharedToExtranet- caseInfo_StartDate- caseInfo_UpdatedTs- caseInfo_UrlToCase- caseInfo_UseDefaultProductsInWorkTimeEntry- caseInfo_UseDefaultWorkTypes- caseInfo_YourReference

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewCase. A value must be provided at all times for required parameters, but optional

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
caseInfo_AccountGUID		☐		
caseInfo_AccountName		☐		
caseInfo_AccountNumber		☐		
caseInfo_BillingAddress		☐		
caseInfo_BillingAddressGUID		☐		
caseInfo_BillingInternalNotes		☐		
caseInfo_BillingNotesBeforeGrid		☐		
caseInfo_BusinessUnitGUID		☐		
caseInfo_BusinessUnitName		☐		
caseInfo_BusinessUnitNumber		☐		
caseInfo_CaseCompanyCurrency GUID		☐		
caseInfo_CaseNumber		☐		
caseInfo_CaseOw nerUserCode		☐		
caseInfo_CaseOw nerUserGUID		☐		
caseInfo_ClosedDate		☐		
caseInfo_CompletionEstimate		☐		
caseInfo_ContactGUID		☐		
caseInfo_CostCenterGUID		☐		
caseInfo_CostCenterNumber		☐		
caseInfo_CurrencyGUID		☐		
caseInfo_CurrencyShortform		☐		
caseInfo_CurrencyShortformOfCa seCompanyCurrency		☐		
caseInfo_CurrentCaseStatusDescr iption		☐		
caseInfo_CurrentCaseStatusGUID		☐		
caseInfo_CurrentCaseStatusInsert Ts		☐		
caseInfo_DeadlineDate		☐		
caseInfo_Description		☐		
caseInfo_ExpectedValue		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
caseInfo_GUID		☐		
caseInfo_IncludeExceptions		☐		
caseInfo_IncludeOverTime		☐		
caseInfo_InsertTs		☐		
caseInfo_InternalName		☐		
caseInfo_InvoiceTemplateGUID		☐		
caseInfo_IsClosed		☐		
caseInfo_IsDailyAllow anceBillable		☐		
caseInfo_IsInternal		☐		
caseInfo_IsMileageBillable		☐		
caseInfo_IsOtherTravelExpensesBillable		☐		
caseInfo_LatestEstimationDate		☐		
caseInfo_LeadRating		☐		
caseInfo_LeadSourceGUID		☐		
caseInfo_Name		☐		
caseInfo_OrderNumber		☐		
caseInfo_OurReference		☐		
caseInfo_OverdueInterest		☐		
caseInfo_PaymentTerm		☐		
caseInfo_PricelistGUID		☐		
caseInfo_PricingRule		☐		
caseInfo_PricingType		☐		
caseInfo_Priority		☐		
caseInfo_Probability		☐		
caseInfo_RootTaskGUID		☐		
caseInfo_SalesCloseDate		☐		
caseInfo_SalesPersonUserCode		☐		
caseInfo_SalesPersonUserGUID		☐		
caseInfo_SalesProcessGUID		☐		
caseInfo_SalesStatusGUID		☐		
caseInfo_SharedToExtranet		☐		
caseInfo_StartDate		☐		
caseInfo_UpdatedTs		☐		
caseInfo_UrlToCase		☐		
caseInfo_UseDefaultProductsInWorkTimeEntry		☐		
caseInfo_UseDefaultWorkTypes		☐		
caseInfo_YourReference		☐		

Columns of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	varchar2	Account GUID		
ACCOUNTNAME	varchar2	Account Name		
ACCOUNTNUMBER	number	Account Number		
BILLINGADDRESS	varchar2	Billing Address		
BILLINGADDRESSGUID	varchar2	Billing Address GUID		
BILLINGINTERNALNOTES	varchar2	Billing Internal Notes		
BILLINGNOTESBEFOREGRID	varchar2	Billing Notes before Grid		
BUSINESSUNITGUID	varchar2	Business Unit GUID		
BUSINESSUNITNAME	varchar2	Business Unit Name		
BUSINESSUNITNUMBER	varchar2	Business Unit Number		
CASECOMPANYCURRENCYGUID	varchar2	Project Company Currency GUID		
CASENUMBER	number	Project Number		
CASEOWNERUSERCODE	varchar2	Project Owner User Code		
CASEOWNERUSERGUID	varchar2	Project Owner User GUID		
CLOSEDDATE	datetime	Closed Date		
COMPLETIONESTIMATE	number	Completion Estimate		
CONTACTGUID	varchar2	Contact GUID		
COSTCENTERGUID	varchar2	Cost Centre GUID		
COSTCENTERNUMBER	varchar2	Cost Centre Number		
CURRENCYGUID	varchar2	Currency GUID		
CURRENCYSHORTFORM	varchar2	Currency Short Form		
CURRENCYSHORTFORMOFCASECOMPANYCURRENCY	varchar2	Currency Short Form of Project Company Currency		
CURRENTCASESTATUSDESCRIPTION	varchar2	Current Project Status Description		
CURRENTCASESTATUSGUID	varchar2	Current Project Status GUID		
CURRENTCASESTATUSINSERTTS	datetime	Current Project Status Inserted		
DEADLINEDATE	datetime	Deadline Date		
DESCRIPTION	varchar2	Ddescription		
EXPECTEDVALUE	number	Expected Value		
GUID	varchar2	GUID		
INCLUDEEXCEPTIONS	boolean	Include Exceptions		
INCLUDEOVERTIME	boolean	Include Overtime		
INSERTTS	datetime	Inserted		
INTERNALNAME	varchar2	Internal Name		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
INVOICETEMPLATEGUID	varchar2	Invoice Template GUID		
ISCLOSED	boolean	Is Closed		
ISDAILYALLOWANCEBILLABLE	boolean	Is Daily Allowance Billable		
ISINTERNAL	boolean	Is Internal		
ISMILEAGEBILLABLE	boolean	Is Mileage Billable		
ISOTHERTRAVELEXPENSESBILLABLE	boolean	Are Other Travel Expenses Billable		
LATESTESTIMATIONDATE	datetime	Latest Estimation Date		
LEADRATING	number	Lead Rating		
LEADSOURCEGUID	varchar2	Lead Source GUID		
NAME	varchar2	Name		
ORDERNUMBER	varchar2	Order Number		
OURREFERENCE	varchar2	Our Reference		
OVERDUEINTEREST	number	Overdue Interest		
PAYMENTTERM	number	Payment Term		
PRICELISTGUID	varchar2	Pricelist GUID		
PRICINGRULE	varchar2	Pricing Rule		
PRICINGTYPE	varchar2	Pricing Type		
PRIORITY	number	Priority		
PROBABILITY	number	Probability		
ROOTTASKGUID	varchar2	Root Task GUID		
SALESCLOSEDATE	datetime	Sales Close Date		
SALESPERSONUSERCODE	varchar2	Sales Person User Code		
SALESPERSONUSERGUID	varchar2	Sales Person User GUID		
SALESPROCESSGUID	varchar2	Sales Process GUID		
SALESSTATUSGUID	varchar2	Sales Status GUID		
SHAREDTOEXTRANET	boolean	Shared to Extranet		
STARTDATE	datetime	Start Date		
UPDATEDTS	datetime	Updated		
URLTOCASE	varchar2	URL to Project		
USEDEFAULTPRODUCTSINWORKTIMEENTRY	boolean	Use Default Products in Work Time Entry		
USEDEFAULTWORKTYPES	boolean	Use Default Work Types		
YOURREFERENCE	varchar2	Your Reference		

AddNewCaseStatus: Visma Severa Add New Project Status

Catalog: Severa

Schema: API

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Label: Add New Project Status

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

Filters on API Server: caseStatusInfo_GUID- caseStatusInfo_Icon- caseStatusInfo_IsActive- caseStatusInfo_Name- insertPosition

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewCaseStatus. 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
caseStatusInfo_GUID		☐		
caseStatusInfo_Icon		☐		
caseStatusInfo_IsActive		☐		
caseStatusInfo_Name		☐		
insertPosition		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
GUID	varchar2	GUID		
ICON	varchar2	Icon		
ISACTIVE	boolean	Is Active		
NAME	varchar2	Name		

AddNewCaseTag: Visma Severa Add New Project Tag

Catalog: Severa

Schema: API

Label: Add New Project Tag

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Filters on API Server: caseGUID- tagInfo_Context- tagInfo_GUID- tagInfo_IsActive- tagInfo_Keyword- tagInfo_Weight

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewCaseTag. 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
caseGUID		☐		
tagInfo_Context		☐		
tagInfo_GUID		☐		
tagInfo_IsActive		☐		
tagInfo_Keyword		☐		
tagInfo_Weight		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CONTEXT	varchar2	Context		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
KEYWORD	varchar2	Keyword		
WEIGHT	number	Weight		

AddNewContact: Visma Severa Add New Contact

Catalog: Severa

Schema: API

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Label: Add New Contact

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

Filters on API Server: contactInfo_AccountGUID- contactInfo_AddressGUID- contactInfo_CommunicationMethods- contactInfo_ContactRoleGUID- contactInfo_DateOfBirth- contactInfo_Description- contactInfo_FirstName- contactInfo_GUID- contactInfo_IsActive- contactInfo_JobTitle- contactInfo_Keywords- contactInfo_LanguageGUID- contactInfo_LastName- contactInfo_Salutation- contactInfo_SatisfactionLevel- contactInfo_TimeZoneGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewContact. 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
contactInfo_AccountGUID		☐		
contactInfo_AddressGUID		☐		
contactInfo_CommunicationMethods		☐		
contactInfo_ContactRoleGUID		☐		
contactInfo_DateOfBirth		☐		
contactInfo_Description		☐		
contactInfo_FirstName		☐		
contactInfo_GUID		☐		
contactInfo_IsActive		☐		
contactInfo_JobTitle		☐		
contactInfo_Keywords		☐		
contactInfo_LanguageGUID		☐		
contactInfo_LastName		☐		
contactInfo_Salutation		☐		
contactInfo_SatisfactionLevel		☐		
contactInfo_TimeZoneGUID		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	varchar2	Account GUID		
ADDRESSGUID	varchar2	Address GUID		
COMMUNICATIONMETHODS	varchar2			
CONTACTROLEGUID	varchar2	Contact Role GUID		
DATEOFBIRTH	datetime	Date of Birth		
DESCRIPTION	varchar2	Ddescription		
FIRSTNAME	varchar2	First Name		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
JOBTITLE	varchar2	Job Title		
KEYWORDS	varchar2			
LANGUAGEGUID	varchar2	Language GUID		
LASTNAME	varchar2	Last Name		
SALUTATION	varchar2	Salutation		
SATISFACTIONLEVEL	number	Satisfaction Level		
TIMEZONEGUID	varchar2	Timezone GUID		

AddNewContactTag: Visma Severa Add New Contact Tag

Catalog: Severa

Schema: API

Label: Add New Contact Tag

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

Filters on API Server: contactGUID- tagInfo_Context- tagInfo_GUID- tagInfo_IsActive- tagInfo_Keyword- tagInfo_Weight

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewContactTag. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
contactGUID		☐		
tagInfo_Context		☐		
tagInfo_GUID		☐		
tagInfo_IsActive		☐		
tagInfo_Keyword		☐		
tagInfo_Weight		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CONTEXT	varchar2	Context		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
KEYWORD	varchar2	Keyword		
WEIGHT	number	Weight		

AddNewCustomer: Visma Severa Add New Customer

Catalog: Severa

Schema: API

Label: Add New Customer

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

Filters on API Server: customer_AccountGroupGUIDs- customer_AccountOwnerUserGUID- customer_AccountRating- customer_AnnualRevenue- customer_AnnualRevenue2- customer_CompanyGUID- customer_CurrencyCode- customer_CurrencyGUID- customer_Email- customer_Employees- customer_GUID- customer_HeadOfficeAddressGUID- customer_IndustryGUID- customer_InsertTS- customer_InvoicingVat- customer_IsActive- customer_IsInternal- customer_IsPrivatePerson- customer_LanguageCode- customer_LanguageGUID- customer_Name- customer_Notes- customer_Number- customer_OverdueInterest- customer_PaymentTerm- customer_PricelistGUID- customer_ReverseCharge- customer_ReverseChargeDescription- customer_TimezoneGUID- customer_UpdatedTS- customer_UrlToAccount- customer_VatNumber- customer_Website- customer_eInvoiceAddress- customer_eInvoiceOperator

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewCustomer. 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
customer_AccountGroupGUIDs		☐		
customer_AccountOwnerUserGUID		☐		
customer_AccountRating		☐		
customer_AnnualRevenue		☐		
customer_AnnualRevenue2		☐		
customer_CompanyGUID		☐		
customer_CurrencyCode		☐		
customer_CurrencyGUID		☐		
customer_eInvoiceAddress		☐		
customer_eInvoiceOperator		☐		
customer_Email		☐		
customer_Employees		☐		
customer_GUID		☐		
customer_HeadOfficeAddressGUID		☐		
customer_IndustryGUID		☐		
customer_InsertTS		☐		
customer_InvoicingVat		☐		
customer_IsActive		☐		
customer_IsInternal		☐		
customer_IsPrivatePerson		☐		
customer_LanguageCode		☐		
customer_LanguageGUID		☐		
customer_Name		☐		
customer_Notes		☐		
customer_Number		☐		
customer_OverdueInterest		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
customer_PaymentTerm		☐		
customer_PricelistGUID		☐		
customer_ReverseCharge		☐		
customer_ReverseChargeDescription		☐		
customer_TimezoneGUID		☐		
customer_UpdatedTS		☐		
customer_UrlToAccount		☐		
customer_VatNumber		☐		
customer_Website		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGROUPGUIDS	varchar2			
ACCOUNTOWNERUSERGUID	varchar2	Account Owner User GUID		
ACCOUNTRATING	number	Account Rating		
ANNUALREVENUE	number	Annual Revenue		
ANNUALREVENUE2	number			
COMPANYGUID	varchar2	Company GUID		
CURRENCYCODE	varchar2	Currency Code		
CURRENCYGUID	varchar2	Currency GUID		
EINVOICEADDRESS	varchar2	e-invoice Address		
EINVOICEOPERATOR	varchar2	e-invoice Operator		
EMAIL	varchar2	Email		
EMPLOYEES	number	Employees		
GUID	varchar2	GUID		
HEADOFFICEADDRESSGUID	varchar2	Head Office Address GUID		
INDUSTRYGUID	varchar2	Industry GUID		
INSERTTS	datetime	Inserted		
INVOICINGVAT	number	Invoicing VAT		
ISACTIVE	boolean	Is Active		
ISINTERNAL	boolean	Is Internal		
ISPRIVATEPERSON	boolean			
LANGUAGECODE	varchar2	Language Code		
LANGUAGEGUID	varchar2	Language GUID		
NAME	varchar2	Name		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
NOTES	varchar2	Notes		
NUMBER	number	Number		
OVERDUEINTEREST	number	Overdue Interest		
PAYMENTTERM	number	Payment Term		
PRICELISTGUID	varchar2	Pricelist GUID		
REVERSECHARGE	boolean	Reverse Charge		
REVERSECHARGEDESCRIPTION	varchar2	Reverse Charge Description		
TIMEZONEGUID	varchar2	Timezone GUID		
UPDATEDTS	datetime	Updated		
URLTOACCOUNT	varchar2	URL to Account		
VATNUMBER	varchar2	VAT Number		
WEBSITE	varchar2	Website		

AddNewEmployment: Visma Severa Add New Employment

Catalog: Severa

Schema: API

Label: Add New Employment

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

Filters on API Server: userGUID- employmentInfo_CompanyBusinessUnitCode- employmentInfo_CompanyBusinessUnitGUID- employmentInfo_CompanyBusinessUnitName- employmentInfo_DailyHours- employmentInfo_EndDate- employmentInfo_GUID- employmentInfo_HourCost- employmentInfo_IsOvertimeAllowed- employmentInfo_StartDate- employmentInfo_Title

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewEmployment. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
employmentInfo_CompanyBusinessUnitCode		☐		
employmentInfo_CompanyBusinessUnitGUID		☐		
employmentInfo_CompanyBusinessUnitName		☐		
employmentInfo_DailyHours		☐		
employmentInfo_EndDate		☐		
employmentInfo_GUID		☐		
employmentInfo_HourCost		☐		
employmentInfo_IsOvertimeAllowed		☐		
employmentInfo_StartDate		☐		
employmentInfo_Title		☐		
userGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
COMPANYBUSINESSUNITCODE	varchar2	Company Business Unit Code		
COMPANYBUSINESSUNITGUID	varchar2	Company Business Unit GUID		
COMPANYBUSINESSUNITNAME	varchar2	Company Business Unit Name		
DAILYHOURS	number	Daily Hours		
ENDDATE	datetime	End Date		
GUID	varchar2	GUID		
HOURCOST	number	Hour Cost		
ISOVERTIMEALLOWED	boolean	Is Overtime Allowed		
STARTDATE	datetime	Start Date		
TITLE	varchar2	Title		

AddNewExpenseToProductByPoNumber: Visma Severa Add New Expense to Product by PO Number

Catalog: Severa

Schema: API

Label: Add New Expense to Product by PO Number

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Filters on API Server: poNumber- itemInfo_BillingDependencyTaskGUID- itemInfo_BillingSchedule- itemInfo_CaseGUID- itemInfo_CaseNumber- itemInfo_CostCenterGUID- itemInfo_CostCenterIdentifier- itemInfo_CostCurrencyShortform- itemInfo_CostCurrencyShortformOfInvoicingCurrency- itemInfo_CostCurrencyShortformOfOrganizationCurrency- itemInfo_CostCurrencyShortformOfUserCompanyCurrency- itemInfo_Description- itemInfo_DisplayPeriodStart- itemInfo_EventDate- itemInfo_GUID- itemInfo_InvoiceDescription- itemInfo_InvoiceGUID- itemInfo_InvoiceQuantity- itemInfo_InvoiceRowGUID- itemInfo_IsApproved- itemInfo_IsBillable- itemInfo_IsOutSourcingCost- itemInfo_IsReadOnly- itemInfo_IsRealised- itemInfo_MeasurementUnit- itemInfo_Name- itemInfo_OrderDate- itemInfo_PlannedBillingDate- itemInfo_PriceCurrencyShortformOfCaseCompanyCurrency- itemInfo_PriceCurrencyShortformOfInvoicingCurrency- itemInfo_PriceCurrencyShortformOfOrganizationCurrency- itemInfo_PriceCurrencyShortformOfUserCompanyCurrency- itemInfo_ProductCode- itemInfo_ProductGUID- itemInfo_ProductType- itemInfo_PurchaseOrderNumber- itemInfo_PurchaseVAT- itemInfo_Quantity- itemInfo_RecurrenceEndDate- itemInfo_RecurrenceEndType- itemInfo_RecurrenceFrequency- itemInfo_RecurrenceStartDate- itemInfo_RecurrenceTimes- itemInfo_RecurringItemGUID- itemInfo_SalesAccountGUID- itemInfo_SalesStatusGUID- itemInfo_TaskGUID- itemInfo_TravelEndTime- itemInfo_TravelStartTime- itemInfo_UnitCost- itemInfo_UnitCostInInvoicingCurrency- itemInfo_UnitCostInOrganizationCurrency- itemInfo_UnitCostInUserCompanyCurrency- itemInfo_UnitCostVatIncludedAmount- itemInfo_UnitPrice- itemInfo_UnitPriceInBaseCurrency- itemInfo_UnitPriceInCaseCompanyCurrency- itemInfo_UnitPriceInUserCompanyCurrency- itemInfo_UserCode- itemInfo_UserGUID- itemInfo_VAT

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewExpenseToProductByPoNumber. 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
itemInfo_BillingDependencyTaskGUID		☐		
itemInfo_BillingSchedule		☐		
itemInfo_CaseGUID		☐		
itemInfo_CaseNumber		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
itemInfo_CostCenterGUID		☐		
itemInfo_CostCenterIdentifier		☐		
itemInfo_CostCurrencyShortform		☐		
itemInfo_CostCurrencyShortformOfInvoicingCurrency		☐		
itemInfo_CostCurrencyShortformOfOrganizationCurrency		☐		
itemInfo_CostCurrencyShortformOfUserCompanyCurrency		☐		
itemInfo_Description		☐		
itemInfo_DisplayPeriodStart		☐		
itemInfo_EventDate		☐		
itemInfo_GUID		☐		
itemInfo_InvoiceDescription		☐		
itemInfo_InvoiceGUID		☐		
itemInfo_InvoiceQuantity		☐		
itemInfo_InvoiceRow GUID		☐		
itemInfo_IsApproved		☐		
itemInfo_IsBillable		☐		
itemInfo_IsOutSourcingCost		☐		
itemInfo_IsReadOnly		☐		
itemInfo_IsRealised		☐		
itemInfo_MeasurementUnit		☐		
itemInfo_Name		☐		
itemInfo_OrderDate		☐		
itemInfo_PlannedBillingDate		☐		
itemInfo_PriceCurrencyShortformOfCaseCompanyCurrency		☐		
itemInfo_PriceCurrencyShortformOfInvoicingCurrency		☐		
itemInfo_PriceCurrencyShortformOfOrganizationCurrency		☐		
itemInfo_PriceCurrencyShortformOfUserCompanyCurrency		☐		
itemInfo_ProductCode		☐		
itemInfo_ProductGUID		☐		
itemInfo_ProductType		☐		
itemInfo_PurchaseOrderNumber		☐		
itemInfo_PurchaseVAT		☐		
itemInfo_Quantity		☐		
itemInfo_RecurrenceEndDate		☐		
itemInfo_RecurrenceEndType		☐		
itemInfo_RecurrenceFrequency		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
itemInfo_RecurrenceStartDate		☐		
itemInfo_RecurrenceTimes		☐		
itemInfo_RecurringItemGUID		☐		
itemInfo_SalesAccountGUID		☐		
itemInfo_SalesStatusGUID		☐		
itemInfo_TaskGUID		☐		
itemInfo_TravelEndTime		☐		
itemInfo_TravelStartTime		☐		
itemInfo_UnitCost		☐		
itemInfo_UnitCostInInvoicingCurrency		☐		
itemInfo_UnitCostInOrganizationCurrency		☐		
itemInfo_UnitCostInUserCompanyCurrency		☐		
itemInfo_UnitCostVatIncludedAmount		☐		
itemInfo_UnitPrice		☐		
itemInfo_UnitPriceInBaseCurrency		☐		
itemInfo_UnitPriceInCaseCompanyCurrency		☐		
itemInfo_UnitPriceInUserCompanyCurrency		☐		
itemInfo_UserCode		☐		
itemInfo_UserGUID		☐		
itemInfo_VAT		☐		
poNumber		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
BILLINGDEPENDENCYTASKGUID	varchar2	Billing Dependency Task GUID		
BILLINGSCHEDULE	varchar2	Billing Schedule		
CASEGUID	varchar2	Project GUID		
CASENUMBER	number	Project Number		
COSTCENTERGUID	varchar2	Cost Centre GUID		
COSTCENTERIDENTIFIER	varchar2	Cost Centre Identifier		
COSTCURRENCYSHORTFORM	varchar2	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Cost Currency Short Form of Invoicing		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
		Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Cost Currency Short Form of Organisation Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Cost Currency Short Form of User Company Currency		
DESCRIPTION	varchar2	Ddescription		
DISPLAYPERIODSTART	datetime	Display Period Start		
EVENTDATE	datetime	Event Date		
GUID	varchar2	GUID		
INVOICEDESCRIPTION	varchar2	Invoice Description		
INVOICEGUID	varchar2	Invoice GUID		
INVOICEQUANTITY	number	Invoice Quantity		
INVOICEROWGUID	varchar2	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISOUTSOURCINGCOST	boolean	Is Out-sourcing Cost		
ISREADONLY	boolean	Is Read-only		
ISREALISED	boolean	Is Realized		
MEASUREMENTUNIT	varchar2	Measurement Unit		
NAME	varchar2	Name		
ORDERDATE	datetime	Order Date		
PLANNEDBILLINGDATE	datetime	Planned Billing Date		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	varchar2	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Price Currency Short Form of Organisation Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Price Currency Short Form of User Company Currency		
PRODUCTCODE	varchar2	Product Code		
PRODUCTGUID	varchar2	Product GUID		
PRODUCTTYPE	varchar2	Product Type		
PURCHASEORDERNUMBER	varchar2	Purchase Order Number		
PURCHASEVAT	number	Purchase VAT		
QUANTITY	number	Quantity		
RECURRENCEENDDATE	datetime	Recurrence End Date		
RECURRENCEENDTYPE	number	Recurrence End Type		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
RECURRENCEFREQUENCY	number	Recurrence Frequency		
RECURRENCESTARTDATE	datetime	Recurrence Start Date		
RECURRENCETIMES	number	Recurrence Times		
RECURRINGITEMGUID	varchar2	Recurring Item GUID		
SALESACCOUNTGUID	varchar2	Sales Account GUID		
SALESSTATUSGUID	varchar2	Sales Status GUID		
TASKGUID	varchar2	Task GUID		
TRAVELENDTIME	datetime	Travel End Time		
TRAVELSTARTTIME	datetime	Travel Start Time		
UNITCOST	number	Unit Cost		
UNITCOSTININVOICINGCURRENCY	number	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	number	Unit Cost in Organisation Currency		
UNITCOSTINUSERCOMPANYCURRENCY	number	Unit Cost in User Company Currency		
UNITCOSTVATINCLUDEDAMOUNT	number	Unit Cost VAT Included Amount		
UNITPRICE	number	Unit Price		
UNITPRICEINBASECURRENCY	number	Unit Price in Base Currency		
UNITPRICEINCASECOMPANYCURRENCY	number	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	number	Unit Price in User Company Currency		
USERCODE	varchar2	User Code		
USERGUID	varchar2	User GUID		
VAT	number	VAT		

AddNewFileTag: Visma Severa Add New File Tag

Catalog: Severa

Schema: API

Label: Add New File Tag

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

Filters on API Server: fileGUID- tagInfo_Context- tagInfo_GUID- tagInfo_IsActive- tagInfo_Keyword- tagInfo_Weight

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewFileTag. A value must be provided at all times for required parameters, but optional

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
fileGUID		☐		
tagInfo_Context		☐		
tagInfo_GUID		☐		
tagInfo_IsActive		☐		
tagInfo_Keyword		☐		
tagInfo_Weight		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CONTEXT	varchar2	Context		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
KEYWORD	varchar2	Keyword		
WEIGHT	number	Weight		

AddNewHourEntry: Visma Severa Add New Hour Entry

Catalog: Severa

Schema: API

Label: Add New Hour Entry

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

Filters on API Server: hourEntryInfo_AccountGUID- hourEntryInfo_CaseBusinessUnitCode- hourEntryInfo_CaseCostCenterIdentifier- hourEntryInfo_CaseGUID- hourEntryInfo_CaseName- hourEntryInfo_CaseNumber- hourEntryInfo_CostCurrencyShortform- hourEntryInfo_CostCurrencyShortformOfInvoicingCurrency-

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

hourEntryInfo_CostCurrencyShortformOfOrganizationCurrency-
hourEntryInfo_CostCurrencyShortformOfUserCompanyCurrency-
hourEntryInfo_Description- hourEntryInfo_EndTime- hourEntryInfo_EventDate-
hourEntryInfo_GUID- hourEntryInfo_InvoiceDescription- hourEntryInfo_InvoiceGUID-
hourEntryInfo_InvoiceQuantity- hourEntryInfo_InvoiceRowGUID- hourEntryInfo_IsApproved-
hourEntryInfo_IsBillable- hourEntryInfo_IsProductive- hourEntryInfo_IsReadOnly-
hourEntryInfo_OverTimeGUID- hourEntryInfo_OvertimeCode- hourEntryInfo_PhaseGUID-
hourEntryInfo_PriceCurrencyShortformOfCaseCompanyCurrency-
hourEntryInfo_PriceCurrencyShortformOfInvoicingCurrency-
hourEntryInfo_PriceCurrencyShortformOfOrganizationCurrency-
hourEntryInfo_PriceCurrencyShortformOfUserCompanyCurrency- hourEntryInfo_Quantity-
hourEntryInfo_StartTime- hourEntryInfo_TaskGUID- hourEntryInfo_UnitCost-
hourEntryInfo_UnitCostInInvoicingCurrency- hourEntryInfo_UnitCostInOrganizationCurrency-
hourEntryInfo_UnitCostInUserCompanyCurrency- hourEntryInfo_UnitPrice-
hourEntryInfo_UnitPriceInBaseCurrency- hourEntryInfo_UnitPriceInCaseCompanyCurrency-
hourEntryInfo_UnitPriceInUserCompanyCurrency- hourEntryInfo_UseInvoiceQuantity-
hourEntryInfo_UserBusinessUnitCode- hourEntryInfo_UserCode-
hourEntryInfo_UserFirstName- hourEntryInfo_UserGUID- hourEntryInfo_UserLastName-
hourEntryInfo_WorkTypeCode- hourEntryInfo_WorkTypeGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewHourEntry. 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
hourEntryInfo_AccountGUID		☐		
hourEntryInfo_CaseBusinessUnitCode		☐		
hourEntryInfo_CaseCostCenterIdentifier		☐		
hourEntryInfo_CaseGUID		☐		
hourEntryInfo_CaseName		☐		
hourEntryInfo_CaseNumber		☐		
hourEntryInfo_CostCurrencyShortform		☐		
hourEntryInfo_CostCurrencyShortformOfInvoicingCurrency		☐		
hourEntryInfo_CostCurrencyShortformOfOrganizationCurrency		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
hourEntryInfo_CostCurrencyShortformOfUserCompanyCurrency		☐		
hourEntryInfo_Description		☐		
hourEntryInfo_EndTime		☐		
hourEntryInfo_EventDate		☐		
hourEntryInfo_GUID		☐		
hourEntryInfo_InvoiceDescription		☐		
hourEntryInfo_InvoiceGUID		☐		
hourEntryInfo_InvoiceQuantity		☐		
hourEntryInfo_InvoiceRow GUID		☐		
hourEntryInfo_IsApproved		☐		
hourEntryInfo_IsBillable		☐		
hourEntryInfo_IsProductive		☐		
hourEntryInfo_IsReadOnly		☐		
hourEntryInfo_OvertimeCode		☐		
hourEntryInfo_OverTimeGUID		☐		
hourEntryInfo_PhaseGUID		☐		
hourEntryInfo_PriceCurrencyShortformOfCaseCompanyCurrency		☐		
hourEntryInfo_PriceCurrencyShortformOfInvoicingCurrency		☐		
hourEntryInfo_PriceCurrencyShortformOfOrganizationCurrency		☐		
hourEntryInfo_PriceCurrencyShortformOfUserCompanyCurrency		☐		
hourEntryInfo_Quantity		☐		
hourEntryInfo_StartTime		☐		
hourEntryInfo_TaskGUID		☐		
hourEntryInfo_UnitCost		☐		
hourEntryInfo_UnitCostInInvoicingCurrency		☐		
hourEntryInfo_UnitCostInOrganizationCurrency		☐		
hourEntryInfo_UnitCostInUserCompanyCurrency		☐		
hourEntryInfo_UnitPrice		☐		
hourEntryInfo_UnitPriceInBaseCurrency		☐		
hourEntryInfo_UnitPriceInCaseCompanyCurrency		☐		
hourEntryInfo_UnitPriceInUserCompanyCurrency		☐		
hourEntryInfo_UseInvoiceQuantity		☐		
hourEntryInfo_UserBusinessUnitCode		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
hourEntryInfo_UserCode		☐		
hourEntryInfo_UserFirstName		☐		
hourEntryInfo_UserGUID		☐		
hourEntryInfo_UserLastName		☐		
hourEntryInfo_WorkTypeCode		☐		
hourEntryInfo_WorkTypeGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	varchar2	Account GUID		
CASEBUSINESSUNITCODE	varchar2	Project Business Unit Code		
CASECOSTCENTERIDENTIFIER	varchar2	Project Cost Centre Identifier		
CASEGUID	varchar2	Project GUID		
CASENAME	varchar2	Project Name		
CASENUMBER	number	Project Number		
COSTCURRENCYSHORTFORM	varchar2	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Cost Currency Short Form of Organisation Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Cost Currency Short Form of User Company Currency		
DESCRIPTION	varchar2	Ddescription		
ENDTIME	datetime	End Time		
EVENTDATE	datetime	Event Date		
GUID	varchar2	GUID		
INVOICEDESCRIPTION	varchar2	Invoice Description		
INVOICEGUID	varchar2	Invoice GUID		
INVOICEQUANTITY	number	Invoice Quantity		
INVOICEROWGUID	varchar2	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISPRODUCTIVE	boolean	Is Productive		
ISREADONLY	boolean	Is Read-only		
OVERTIMECODE	varchar2	Overtime Code		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
OVERTIMEGUID	varchar2	Overtime GUID		
PHASEGUID	varchar2	Phase GUID		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	varchar2	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Price Currency Short Form of Organisation Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Price Currency Short Form of User Company Currency		
QUANTITY	number	Quantity		
STARTTIME	datetime	Start Time		
TASKGUID	varchar2	Task GUID		
UNITCOST	number	Unit Cost		
UNITCOSTININVOICINGCURRENCY	number	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	number	Unit Cost in Organisation Currency		
UNITCOSTINUSERCOMPANYCURRENCY	number	Unit Cost in User Company Currency		
UNITPRICE	number	Unit Price		
UNITPRICEINBASECURRENCY	number	Unit Price in Base Currency		
UNITPRICEINCASECOMPANYCURRENCY	number	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	number	Unit Price in User Company Currency		
USEINVOICEQUANTITY	boolean			
USERBUSINESSUNITCODE	varchar2	User Business Unit Code		
USERCODE	varchar2	User Code		
USERFIRSTNAME	varchar2	User First Name		
USERGUID	varchar2	User GUID		
USERLASTNAME	varchar2	User Last Name		
WORKTYPECODE	varchar2	Work Type Code		
WORKTYPEGUID	varchar2	Work Type GUID		

AddNewIndustry: Visma Severa Add New Industry

Catalog: Severa

Schema: API

Label: Add New Industry

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Filters on API Server: industryInfo_Code- industryInfo_GUID- industryInfo_IsActive- industryInfo_Name

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewIndustry. 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
industryInfo_Code		☐		
industryInfo_GUID		☐		
industryInfo_IsActive		☐		
industryInfo_Name		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CODE	varchar2	Code		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
NAME	varchar2	Name		

AddNewInvoiceLogEntry: Visma Severa Add New Invoice Log Entry

Catalog: Severa

Schema: API

Label: Add New Invoice Log Entry

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Filters on API Server: invoiceGuid- context- text- source- eventTimeUtc- ignoreDuplicates

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewInvoiceLogEntry. 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
context		☐		
eventTimeUtc		☐		
ignoreDuplicates		☐		
invoiceGuid		☐		
source		☐		
text		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
DUMMY	varchar2	Dummy	☑	Dummy column

AddNewItem: Visma Severa Add New Item

Catalog: Severa

Schema: API

Label: Add New Item

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Filters on API Server: itemInfo_BillingDependencyTaskGUID- itemInfo_BillingSchedule- itemInfo_CaseGUID- itemInfo_CaseNumber- itemInfo_CostCenterGUID- itemInfo_CostCenterIdentifier- itemInfo_CostCurrencyShortform- itemInfo_CostCurrencyShortformOfInvoicingCurrency- itemInfo_CostCurrencyShortformOfOrganizationCurrency- itemInfo_CostCurrencyShortformOfUserCompanyCurrency- itemInfo_Description- itemInfo_DisplayPeriodStart- itemInfo_EventDate- itemInfo_GUID- itemInfo_InvoiceDescription- itemInfo_InvoiceGUID- itemInfo_InvoiceQuantity- itemInfo_InvoiceRowGUID- itemInfo_IsApproved- itemInfo_IsBillable- itemInfo_IsOutSourcingCost- itemInfo_IsReadOnly- itemInfo_IsRealised- itemInfo_MeasurementUnit- itemInfo_Name- itemInfo_OrderDate- itemInfo_PlannedBillingDate- itemInfo_PriceCurrencyShortformOfCaseCompanyCurrency- itemInfo_PriceCurrencyShortformOfInvoicingCurrency- itemInfo_PriceCurrencyShortformOfOrganizationCurrency- itemInfo_PriceCurrencyShortformOfUserCompanyCurrency- itemInfo_ProductCode- itemInfo_ProductGUID- itemInfo_ProductType- itemInfo_PurchaseOrderNumber- itemInfo_PurchaseVAT- itemInfo_Quantity- itemInfo_RecurrenceEndDate- itemInfo_RecurrenceEndType- itemInfo_RecurrenceFrequency- itemInfo_RecurrenceStartDate- itemInfo_RecurrenceTimes- itemInfo_RecurringItemGUID- itemInfo_SalesAccountGUID- itemInfo_SalesStatusGUID- itemInfo_TaskGUID- itemInfo_TravelEndTime- itemInfo_TravelStartTime- itemInfo_UnitCost- itemInfo_UnitCostInInvoicingCurrency- itemInfo_UnitCostInOrganizationCurrency- itemInfo_UnitCostInUserCompanyCurrency- itemInfo_UnitCostVatIncludedAmount- itemInfo_UnitPrice- itemInfo_UnitPriceInBaseCurrency- itemInfo_UnitPriceInCaseCompanyCurrency- itemInfo_UnitPriceInUserCompanyCurrency- itemInfo_UserCode- itemInfo_UserGUID- itemInfo_VAT- currencyCode

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewItem. 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
currencyCode		☐		
itemInfo_BillingDependencyTaskGUID		☐		
itemInfo_BillingSchedule		☐		
itemInfo_CaseGUID		☐		
itemInfo_CaseNumber		☐		
itemInfo_CostCenterGUID		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
itemInfo_CostCenterIdentifier		☐		
itemInfo_CostCurrencyShortform		☐		
itemInfo_CostCurrencyShortformOfInvoicingCurrency		☐		
itemInfo_CostCurrencyShortformOfOrganizationCurrency		☐		
itemInfo_CostCurrencyShortformOfUserCompanyCurrency		☐		
itemInfo_Description		☐		
itemInfo_DisplayPeriodStart		☐		
itemInfo_EventDate		☐		
itemInfo_GUID		☐		
itemInfo_InvoiceDescription		☐		
itemInfo_InvoiceGUID		☐		
itemInfo_InvoiceQuantity		☐		
itemInfo_InvoiceRow GUID		☐		
itemInfo_IsApproved		☐		
itemInfo_IsBillable		☐		
itemInfo_IsOutSourcingCost		☐		
itemInfo_IsReadOnly		☐		
itemInfo_IsRealised		☐		
itemInfo_MeasurementUnit		☐		
itemInfo_Name		☐		
itemInfo_OrderDate		☐		
itemInfo_PlannedBillingDate		☐		
itemInfo_PriceCurrencyShortformOfCaseCompanyCurrency		☐		
itemInfo_PriceCurrencyShortformOfInvoicingCurrency		☐		
itemInfo_PriceCurrencyShortformOfOrganizationCurrency		☐		
itemInfo_PriceCurrencyShortformOfUserCompanyCurrency		☐		
itemInfo_ProductCode		☐		
itemInfo_ProductGUID		☐		
itemInfo_ProductType		☐		
itemInfo_PurchaseOrderNumber		☐		
itemInfo_PurchaseVAT		☐		
itemInfo_Quantity		☐		
itemInfo_RecurrenceEndDate		☐		
itemInfo_RecurrenceEndType		☐		
itemInfo_RecurrenceFrequency		☐		
itemInfo_RecurrenceStartDate		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
itemInfo_RecurrenceTimes		☐		
itemInfo_RecurringItemGUID		☐		
itemInfo_SalesAccountGUID		☐		
itemInfo_SalesStatusGUID		☐		
itemInfo_TaskGUID		☐		
itemInfo_TravelEndTime		☐		
itemInfo_TravelStartTime		☐		
itemInfo_UnitCost		☐		
itemInfo_UnitCostInInvoicingCurrency		☐		
itemInfo_UnitCostInOrganizationCurrency		☐		
itemInfo_UnitCostInUserCompanyCurrency		☐		
itemInfo_UnitCostVatIncludedAmount		☐		
itemInfo_UnitPrice		☐		
itemInfo_UnitPriceInBaseCurrency		☐		
itemInfo_UnitPriceInCaseCompanyCurrency		☐		
itemInfo_UnitPriceInUserCompanyCurrency		☐		
itemInfo_UserCode		☐		
itemInfo_UserGUID		☐		
itemInfo_VAT		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
BILLINGDEPENDENCYTASKGUID	varchar2	Billing Dependency Task GUID		
BILLINGSCHEDULE	varchar2	Billing Schedule		
CASEGUID	varchar2	Project GUID		
CASENUMBER	number	Project Number		
COSTCENTERGUID	varchar2	Cost Centre GUID		
COSTCENTERIDENTIFIER	varchar2	Cost Centre Identifier		
COSTCURRENCYSHORTFORM	varchar2	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Cost Currency Short Form of Invoicing Currency		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Cost Currency Short Form of Organisation Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Cost Currency Short Form of User Company Currency		
DESCRIPTION	varchar2	Ddescription		
DISPLAYPERIODSTART	datetime	Display Period Start		
EVENTDATE	datetime	Event Date		
GUID	varchar2	GUID		
INVOICEDESCRIPTION	varchar2	Invoice Description		
INVOICEGUID	varchar2	Invoice GUID		
INVOICEQUANTITY	number	Invoice Quantity		
INVOICEROWGUID	varchar2	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISOUTSOURCINGCOST	boolean	Is Out-sourcing Cost		
ISREADONLY	boolean	Is Read-only		
ISREALISED	boolean	Is Realized		
MEASUREMENTUNIT	varchar2	Measurement Unit		
NAME	varchar2	Name		
ORDERDATE	datetime	Order Date		
PLANNEDBILLINGDATE	datetime	Planned Billing Date		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	varchar2	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Price Currency Short Form of Organisation Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Price Currency Short Form of User Company Currency		
PRODUCTCODE	varchar2	Product Code		
PRODUCTGUID	varchar2	Product GUID		
PRODUCTTYPE	varchar2	Product Type		
PURCHASEORDERNUMBER	varchar2	Purchase Order Number		
PURCHASEVAT	number	Purchase VAT		
QUANTITY	number	Quantity		
RECURRENCEENDDATE	datetime	Recurrence End Date		
RECURRENCEENDTYPE	number	Recurrence End Type		
RECURRENCEFREQUENCY	number	Recurrence Frequency		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
RECURRENCESTARTDATE	datetime	Recurrence Start Date		
RECURRENCETIMES	number	Recurrence Times		
RECURRINGITEMGUID	varchar2	Recurring Item GUID		
SALESACCOUNTGUID	varchar2	Sales Account GUID		
SALESSTATUSGUID	varchar2	Sales Status GUID		
TASKGUID	varchar2	Task GUID		
TRAVELENDTIME	datetime	Travel End Time		
TRAVELSTARTTIME	datetime	Travel Start Time		
UNITCOST	number	Unit Cost		
UNITCOSTININVOICINGCURRENCY	number	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	number	Unit Cost in Organisation Currency		
UNITCOSTINUSERCOMPANYCURRENCY	number	Unit Cost in User Company Currency		
UNITCOSTVATINCLUDEDAMOUNT	number	Unit Cost VAT Included Amount		
UNITPRICE	number	Unit Price		
UNITPRICEINBASECURRENCY	number	Unit Price in Base Currency		
UNITPRICEINCASECOMPANYCURRENCY	number	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	number	Unit Price in User Company Currency		
USERCODE	varchar2	User Code		
USERGUID	varchar2	User GUID		
VAT	number	VAT		

AddNewItems: Visma Severa Add New Items

Catalog: Severa

Schema: API

Label: Add New Items

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

Filters on API Server: items

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewItems. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
items		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
BILLINGDEPENDENCYTASKGUID	varchar2	Billing Dependency Task GUID		
BILLINGSCHEDULE	varchar2	Billing Schedule		
CASEGUID	varchar2	Project GUID		
CASENUMBER	number	Project Number		
COSTCENTERGUID	varchar2	Cost Centre GUID		
COSTCENTERIDENTIFIER	varchar2	Cost Centre Identifier		
COSTCURRENCYSHORTFORM	varchar2	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Cost Currency Short Form of Organisation Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Cost Currency Short Form of User Company Currency		
DESCRIPTION	varchar2	Ddescription		
DISPLAYPERIODSTART	datetime	Display Period Start		
EVENTDATE	datetime	Event Date		
GUID	varchar2	GUID		
INVOICEDESCRIPTION	varchar2	Invoice Description		
INVOICEGUID	varchar2	Invoice GUID		
INVOICEQUANTITY	number	Invoice Quantity		
INVOICEROWGUID	varchar2	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISOUTSOURCINGCOST	boolean	Is Out-sourcing Cost		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
ISREADONLY	boolean	Is Read-only		
ISREALISED	boolean	Is Realized		
MEASUREMENTUNIT	varchar2	Measurement Unit		
NAME	varchar2	Name		
ORDERDATE	datetime	Order Date		
PLANNEDBILLINGDATE	datetime	Planned Billing Date		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	varchar2	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Price Currency Short Form of Organisation Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Price Currency Short Form of User Company Currency		
PRODUCTCODE	varchar2	Product Code		
PRODUCTGUID	varchar2	Product GUID		
PRODUCTTYPE	varchar2	Product Type		
PURCHASEORDERNUMBER	varchar2	Purchase Order Number		
PURCHASEVAT	number	Purchase VAT		
QUANTITY	number	Quantity		
RECURRENCEENDDATE	datetime	Recurrence End Date		
RECURRENCEENDTYPE	number	Recurrence End Type		
RECURRENCEFREQUENCY	number	Recurrence Frequency		
RECURRENCESTARTDATE	datetime	Recurrence Start Date		
RECURRENCETIMES	number	Recurrence Times		
RECURRINGITEMGUID	varchar2	Recurring Item GUID		
SALESACCOUNTGUID	varchar2	Sales Account GUID		
SALESSTATUSGUID	varchar2	Sales Status GUID		
TASKGUID	varchar2	Task GUID		
TRAVELENDTIME	datetime	Travel End Time		
TRAVELSTARTTIME	datetime	Travel Start Time		
UNITCOST	number	Unit Cost		
UNITCOSTININVOICINGCURRENCY	number	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	number	Unit Cost in Organisation Currency		
UNITCOSTINUSERCOMPANYCURRENCY	number	Unit Cost in User Company Currency		
UNITCOSTVATINCLUDEDAMOUNT	number	Unit Cost VAT Included Amount		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
UNITPRICE	number	Unit Price		
UNITPRICEINBASECURRENCY	number	Unit Price in Base Currency		
UNITPRICEINCASECOMPANYCURRENCY	number	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	number	Unit Price in User Company Currency		
USERCODE	varchar2	User Code		
USERGUID	varchar2	User GUID		
VAT	number	VAT		

AddNewPhase: Visma Severa Add New Phase

Catalog: Severa

Schema: API

Label: Add New Phase

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

Filters on API Server: phaseInfo_CaseGUID- phaseInfo_Code- phaseInfo_Deadline- phaseInfo_Description- phaseInfo_FlatRate_AdditionalHoursUnitPrice- phaseInfo_FlatRate_AreAdditionalHoursBillable- phaseInfo_FlatRate_Hours- phaseInfo_FlatRate_Price- phaseInfo_GUID- phaseInfo_IsCompleted- phaseInfo_IsLocked- phaseInfo_Name- phaseInfo_OriginalDeadline- phaseInfo_OriginalPlannedStartDate- phaseInfo_OriginalWorkEstimate- phaseInfo_OwnerUserGUID- phaseInfo_ParentPhaseGUID- phaseInfo_PlannedStartDate- phaseInfo_PricePerHour- phaseInfo_SortOrder- phaseInfo_WorkEstimate- phaseInfo_WorkTypeGUID- members

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewPhase. 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
members		☐		
phaseInfo_CaseGUID		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
phaseInfo_Code		☐		
phaseInfo_Deadline		☐		
phaseInfo_Description		☐		
phaseInfo_FlatRate_AdditionalHoursUnitPrice		☐		
phaseInfo_FlatRate_AreAdditionalHoursBillable		☐		
phaseInfo_FlatRate_Hours		☐		
phaseInfo_FlatRate_Price		☐		
phaseInfo_GUID		☐		
phaseInfo_IsCompleted		☐		
phaseInfo_IsLocked		☐		
phaseInfo_Name		☐		
phaseInfo_OriginalDeadline		☐		
phaseInfo_OriginalPlannedStartDate		☐		
phaseInfo_OriginalWorkEstimate		☐		
phaseInfo_OwnerUserGUID		☐		
phaseInfo_ParentPhaseGUID		☐		
phaseInfo_PlannedStartDate		☐		
phaseInfo_PricePerHour		☐		
phaseInfo_SortOrder		☐		
phaseInfo_WorkEstimate		☐		
phaseInfo_WorkTypeGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CASEGUID	varchar2	Project GUID		
CODE	varchar2	Code		
DEADLINE	datetime	Deadline		
DESCRIPTION	varchar2	Ddescription		
FLATRATE_ADDITIONALHOURSUNITPRICE	number	Flat Rate Additional Hours Unit Price		
FLATRATE_AREADDITIONALHOURSBILLABLE	boolean	Flat Rate Are additional Hours Billable		
FLATRATE_HOURS	number	Flat Rate Hours		
FLATRATE_PRICE	number	Flat Rate Price		
GUID	varchar2	GUID		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
ISCOMPLETED	boolean	Is Completed		
ISLOCKED	boolean	Is Locked		
NAME	varchar2	Name		
ORIGINALDEADLINE	datetime	Original Deadline		
ORIGINALPLANNEDSTARTDATE	datetime	Original Planned Start Date		
ORIGINALWORKESTIMATE	number	Original Work Estimate		
OWNERUSERGUID	varchar2	Owner User GUID		
PARENTPHASEGUID	varchar2	Parent Phase GUID		
PLANNEDSTARTDATE	datetime	Planned Start Date		
PRICEPERHOUR	number	Price per Hour		
SORTORDER	number	Sort Order		
WORKESTIMATE	number	Work Estimate		
WORKTYPEGUID	varchar2	Work Type GUID		

AddNewProduct: Visma Severa Add New Product

Catalog: Severa

Schema: API

Label: Add New Product

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

Filters on API Server: productInfo_Code- productInfo_Description- productInfo_GUID- productInfo_IsActive- productInfo_MeasurementUnit- productInfo_Name- productInfo_Type- productInfo_UnitCost- productInfo_UnitPrice- productInfo_UseInWorkTimeEntry- productInfo_VAT- productInfo_ProductCategoryGUID- productInfo_SalesAccountGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewProduct. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
productInfo_Code		☐		
productInfo_Description		☐		
productInfo_GUID		☐		
productInfo_IsActive		☐		
productInfo_MeasurementUnit		☐		
productInfo_Name		☐		
productInfo_ProductCategoryGUID		☐		
productInfo_SalesAccountGUID		☐		
productInfo_Type		☐		
productInfo_UnitCost		☐		
productInfo_UnitPrice		☐		
productInfo_UseInWorkTimeEntry		☐		
productInfo_VAT		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CODE	varchar2	Code		
DESCRIPTION	varchar2	Ddescription		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
MEASUREMENTUNIT	varchar2	Measurement Unit		
NAME	varchar2	Name		
PRODUCTCATEGORYGUID	varchar2	Product Category GUID		
SALESACCOUNTGUID	varchar2	Sales Account GUID		
TYPE	varchar2	Type		
UNITCOST	number	Unit Cost		
UNITPRICE	number	Unit Price		
USEINWORKTIMEENTRY	boolean	Use in Work Time Entry		
VAT	number	VAT		

AddNewProductCategory: Visma Severa Add New Product Category

Catalog: Severa

Schema: API

Label: Add New Product Category

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Filters on API Server: productCategoryInfo_Code- productCategoryInfo_GUID- productCategoryInfo_IsActive- productCategoryInfo_IsDefault- productCategoryInfo_Name

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewProductCategory. 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
productCategoryInfo_Code		☐		
productCategoryInfo_GUID		☐		
productCategoryInfo_IsActive		☐		
productCategoryInfo_IsDefault		☐		
productCategoryInfo_Name		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CODE	varchar2	Code		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
NAME	varchar2	Name		

AddNewResource: Visma Severa Add New Resource

Catalog: Severa

Schema: API

Label: Add New Resource

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Filters on API Server: resourceInfo_GUID- resourceInfo_IsActive- resourceInfo_Name

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewResource. 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
resourceInfo_GUID		☐		
resourceInfo_IsActive		☐		
resourceInfo_Name		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
NAME	varchar2	Name		

AddNewResourceAllocation: Visma Severa Add New Resource Allocation

Catalog: Severa

Schema: API

Label: Add New Resource Allocation

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

Filters on API Server: resourceAllocationInfo_AllocatedHours- resourceAllocationInfo_AllocatedPercentage- resourceAllocationInfo_CaseGUID-

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

resourceAllocationInfo_EndDate- resourceAllocationInfo_GUID- resourceAllocationInfo_PhaseGUID- resourceAllocationInfo_StartDate- resourceAllocationInfo_UserGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewResourceAllocation. 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
resourceAllocationInfo_AllocatedHours		☐		
resourceAllocationInfo_AllocatedPercentage		☐		
resourceAllocationInfo_CaseGUID		☐		
resourceAllocationInfo_EndDate		☐		
resourceAllocationInfo_GUID		☐		
resourceAllocationInfo_PhaseGUID		☐		
resourceAllocationInfo_StartDate		☐		
resourceAllocationInfo_UserGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ALLOCATEDHOURS	number	Allocated Hours		
ALLOCATEDPERCENTAGE	number	Allocated Percentage		
CASEGUID	varchar2	Project GUID		
ENDDATE	datetime	End Date		
GUID	varchar2	GUID		
PHASEGUID	varchar2	Phase GUID		
STARTDATE	datetime	Start Date		
USERGUID	varchar2	User GUID		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

AddNewResourceAllocations: Visma Severa Add New Resource Allocations

Catalog: Severa

Schema: API

Label: Add New Resource Allocations

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

Filters on API Server: resourceAllocations

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewResourceAllocations. 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
resourceAllocations		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ALLOCATEDHOURS	number	Allocated Hours		
ALLOCATEDPERCENTAGE	number	Allocated Percentage		
CASEGUID	varchar2	Project GUID		
ENDDATE	datetime	End Date		
GUID	varchar2	GUID		
PHASEGUID	varchar2	Phase GUID		
STARTDATE	datetime	Start Date		
USERGUID	varchar2	User GUID		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

AddNewSalesProcess: Visma Severa Add New Sales Process

Catalog: Severa

Schema: API

Label: Add New Sales Process

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

Filters on API Server: salesProcessInfo_DefaultProbability- salesProcessInfo_GUID- salesProcessInfo_Icon- salesProcessInfo_IsActive- salesProcessInfo_IsAlsoItemStatus- salesProcessInfo_IsInProgress- salesProcessInfo_IsLost- salesProcessInfo_IsOffer- salesProcessInfo_IsWon- salesProcessInfo_Name

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewSalesProcess. 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
salesProcessInfo_DefaultProbability		☐		
salesProcessInfo_GUID		☐		
salesProcessInfo_Icon		☐		
salesProcessInfo_IsActive		☐		
salesProcessInfo_IsAlsoItemStatus		☐		
salesProcessInfo_IsInProgress		☐		
salesProcessInfo_IsLost		☐		
salesProcessInfo_IsOffer		☐		
salesProcessInfo_IsWon		☐		
salesProcessInfo_Name		☐		

Columns of Table Function

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
DEFAULTPROBABILITY	number	Default Probability		
GUID	varchar2	GUID		
ICON	varchar2	Icon		
ISACTIVE	boolean	Is Active		
ISALSOITEMSTATUS	boolean	Is also Item Status		
ISINPROGRESS	boolean	Is in Progress		
ISLOST	boolean	Is Lost		
ISOFFER	boolean	Is Offer		
ISWON	boolean	Is Won		
NAME	varchar2	Name		

AddNewSalesStatus: Visma Severa Add New Sales Status

Catalog: Severa

Schema: API

Label: Add New Sales Status

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

Filters on API Server: salesStatusInfo_CaseGUID- salesStatusInfo_GUID- salesStatusInfo_SalesProcessGUID- salesStatusInfo_TimeStamp

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewSalesStatus. 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
salesStatusInfo_CaseGUID		☐		
salesStatusInfo_GUID		☐		
salesStatusInfo_SalesProcessGUID		☐		
salesStatusInfo_TimeStamp		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CASEGUID	varchar2	Project GUID		
GUID	varchar2	GUID		
SALESPROCESSGUID	varchar2	Sales Process GUID		
TIMESTAMP	datetime	Time Stamp		

AddNewTravelExpense: Visma Severa Add New Travel Expense

Catalog: Severa

Schema: API

Label: Add New Travel Expense

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

Filters on API Server: travelExpenseInfo_Code- travelExpenseInfo_Description- travelExpenseInfo_GUID- travelExpenseInfo_IsActive- travelExpenseInfo_MeasurementUnit- travelExpenseInfo_Name- travelExpenseInfo_Type- travelExpenseInfo_UnitCost- travelExpenseInfo_UnitPrice- travelExpenseInfo_UseInWorkTimeEntry- travelExpenseInfo_VAT- travelExpenseInfo_CostCurrencyGUID- travelExpenseInfo_ExpenseClass

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewTravelExpense. 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
travelExpenseInfo_Code		☐		
travelExpenseInfo_CostCurrencyGUID		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
travelExpenseInfo_Description		☐		
travelExpenseInfo_ExpenseClass		☐		
travelExpenseInfo_GUID		☐		
travelExpenseInfo_IsActive		☐		
travelExpenseInfo_MeasurementUnit		☐		
travelExpenseInfo_Name		☐		
travelExpenseInfo_Type		☐		
travelExpenseInfo_UnitCost		☐		
travelExpenseInfo_UnitPrice		☐		
travelExpenseInfo_UseInWorkTimeEntry		☐		
travelExpenseInfo_VAT		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CODE	varchar2	Code		
COSTCURRENCYGUID	varchar2	Cost Currency GUID		
DESCRIPTION	varchar2	Ddescription		
EXPENSECLASS	varchar2	Expense Class		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
MEASUREMENTUNIT	varchar2	Measurement Unit		
NAME	varchar2	Name		
TYPE	varchar2	Type		
UNITCOST	number	Unit Cost		
UNITPRICE	number	Unit Price		
USEINWORKTIMEENTRY	boolean	Use in Work Time Entry		
VAT	number	VAT		

AddNewUser: Visma Severa Add New User

Catalog: Severa

Schema: API

Label: Add New User

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Filters on API Server: userInfo_Addressline- userInfo_BankAccountNumber- userInfo_BusinessUnitCode- userInfo_BusinessUnitGUID- userInfo_BusinessUnitName- userInfo_Code- userInfo_CountryCode- userInfo_CountryGUID- userInfo_CountryRegionGUID- userInfo_CountryRegionName- userInfo_Culture- userInfo_DefaultActivityTypeGUID- userInfo_Email- userInfo_FirstName- userInfo_GUID- userInfo_IsActive- userInfo_LanguageCode- userInfo_LanguageGUID- userInfo_LastName- userInfo_LicenseType- userInfo_Notes- userInfo_OrganizationGUID- userInfo_Phone- userInfo_PhotoFileGUID- userInfo_PostOffice- userInfo_PostalCode- userInfo_ProfileGUID- userInfo_Salutation- userInfo_SendDailyMail- userInfo_SendWeeklyMail- userInfo_SocialSecurityNumber- userInfo_SuperiorUserGUID- userInfo_TimezoneGUID- userInfo_Title- userInfo_WorktypeGUID- sendWelcomeEmail

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewUser. 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
sendWelcomeEmail		☐		
userInfo_Addressline		☐		
userInfo_BankAccountNumber		☐		
userInfo_BusinessUnitCode		☐		
userInfo_BusinessUnitGUID		☐		
userInfo_BusinessUnitName		☐		
userInfo_Code		☐		
userInfo_CountryCode		☐		
userInfo_CountryGUID		☐		
userInfo_CountryRegionGUID		☐		
userInfo_CountryRegionName		☐		
userInfo_Culture		☐		
userInfo_DefaultActivityTypeGUID		☐		
userInfo_Email		☐		
userInfo_FirstName		☐		
userInfo_GUID		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
userInfo_IsActive		☐		
userInfo_LanguageCode		☐		
userInfo_LanguageGUID		☐		
userInfo_LastName		☐		
userInfo_LicenseType		☐		
userInfo_Notes		☐		
userInfo_OrganizationGUID		☐		
userInfo_Phone		☐		
userInfo_PhotoFileGUID		☐		
userInfo_PostalCode		☐		
userInfo_PostOffice		☐		
userInfo_ProfileGUID		☐		
userInfo_Salutation		☐		
userInfo_SendDailyMail		☐		
userInfo_SendWeeklyMail		☐		
userInfo_SocialSecurityNumber		☐		
userInfo_SuperiorUserGUID		☐		
userInfo_TimezoneGUID		☐		
userInfo_Title		☐		
userInfo_WorktypeGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ADDRESSLINE	varchar2	Addressline		
BANKACCOUNTNUMBER	varchar2	Bank Account Number		
BUSINESSUNITCODE	varchar2	Business Unit Code		
BUSINESSUNITGUID	varchar2	Business Unit GUID		
BUSINESSUNITNAME	varchar2	Business Unit Name		
CODE	varchar2	Code		
COUNTRYCODE	varchar2	Country Code		
COUNTRYGUID	varchar2	Country GUID		
COUNTRYREGIONGUID	varchar2	Country Region GUID		
COUNTRYREGIONNAME	varchar2	Country Region Name		
CULTURE	varchar2	Culture		
DEFAULTACTIVITYTYPEGUID	varchar2	Default Activity Type GUID		
EMAIL	varchar2	Email		
FIRSTNAME	varchar2	First Name		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
LANGUAGECODE	varchar2	Language Code		
LANGUAGEGUID	varchar2	Language GUID		
LASTNAME	varchar2	Last Name		
LICENSETYPE	varchar2			
NOTES	varchar2	Notes		
ORGANIZATIONGUID	varchar2	Organisation GUID		
PHONE	varchar2	Phone		
PHOTOFILEGUID	varchar2	Photo File GUID		
POSTALCODE	varchar2	Postal Code		
POSTOFFICE	varchar2	Post Office		
PROFILEGUID	varchar2	Profile GUID		
SALUTATION	varchar2	Salutation		
SENDDAILYMAIL	boolean	Send Daily Mail		
SENDWEEKLYMAIL	boolean	Send Weekly Mail		
SOCIALSECURITYNUMBER	varchar2	Social Security Number		
SUPERIORUSERGUID	varchar2	Superior User GUID		
TIMEZONEGUID	varchar2	Timezone GUID		
TITLE	varchar2	Title		
WORKTYPEGUID	varchar2	Work Type GUID		

AddNewUserTag: Visma Severa Add New User Tag

Catalog: Severa

Schema: API

Label: Add New User Tag

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

Filters on API Server: userGUID- tagInfo_Context- tagInfo_GUID- tagInfo_IsActive- tagInfo_Keyword- tagInfo_Weight

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewUserTag. 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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
tagInfo_Context		☐		
tagInfo_GUID		☐		
tagInfo_IsActive		☐		
tagInfo_Keyword		☐		
tagInfo_Weight		☐		
userGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CONTEXT	varchar2	Context		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
KEYWORD	varchar2	Keyword		
WEIGHT	number	Weight		

AddNewWorkType: Visma Severa Add New Work Type

Catalog: Severa

Schema: API

Label: Add New Work Type

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

Filters on API Server: workTypeInfo_Code- workTypeInfo_GUID- workTypeInfo_IsActive- workTypeInfo_IsDefault- workTypeInfo_IsProductive- workTypeInfo_Name- workTypeInfo_SalesAccountGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddNewWorkType. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
workTypeInfo_Code		☐		
workTypeInfo_GUID		☐		
workTypeInfo_IsActive		☐		
workTypeInfo_IsDefault		☐		
workTypeInfo_IsProductive		☐		
workTypeInfo_Name		☐		
workTypeInfo_SalesAccountGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CODE	varchar2	Code		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
ISPRODUCTIVE	boolean	Is Productive		
NAME	varchar2	Name		
SALESACCOUNTGUID	varchar2	Sales Account GUID		

AddressByGUID: Visma Severa Address by GUID

Catalog: Severa

Schema: API

Label: Address by GUID

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

Filters on API Server: addressGUID

Parameters of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The following parameters can be used to control the behaviour of the table function AddressByGUID. 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
addressGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ADDRESSLINE	varchar2	Addressline		
ADDRESSLINE2	varchar2	Addressline 2		
ADDRESSLINE3	varchar2	Addressline 3		
CITY	varchar2	City		
COMPANYGUID	varchar2	Company GUID		
COUNTRYCODE	varchar2	Country Code		
COUNTRYGUID	varchar2	Country GUID		
COUNTRYNAME	varchar2	Country Name		
COUNTRYREGIONGUID	varchar2	Country Region GUID		
COUNTRYREGIONNAME	varchar2	Country Region Name		
FAX	varchar2	Fax		
GUID	varchar2	GUID		
ISBILLINGADDRESS	boolean	Is Billing Address		
ISPOSTALADDRESS	boolean	Is Postal Address		
ISVISITADDRESS	boolean	Is Visit Address		
PHONE	varchar2	Phone		
POSTALCODE	varchar2	Postal Code		

AddressesChangedSince: Visma Severa Addresses Changed since

Catalog: Severa

Schema: API

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Label: Addresses Changed since

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

Filters on API Server: accountGUID- startDateTime- options

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AddressesChangedSince. 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
accountGUID		☐		
options		☐		
startDateTime		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ADDRESSLINE	varchar2	Addressline		
ADDRESSLINE2	varchar2	Addressline 2		
ADDRESSLINE3	varchar2	Addressline 3		
CITY	varchar2	City		
COMPANYGUID	varchar2	Company GUID		
COUNTRYCODE	varchar2	Country Code		
COUNTRYGUID	varchar2	Country GUID		
COUNTRYNAME	varchar2	Country Name		
COUNTRYREGIONGUID	varchar2	Country Region GUID		
COUNTRYREGIONNAME	varchar2	Country Region Name		
FAX	varchar2	Fax		
GUID	varchar2	GUID		
ISBILLINGADDRESS	boolean	Is Billing Address		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
ISPOSTALADDRESS	boolean	Is Postal Address		
ISVISITADDRESS	boolean	Is Visit Address		
PHONE	varchar2	Phone		
POSTALCODE	varchar2	Postal Code		

BillingAddress: Visma Severa Billing Address

Catalog: Severa

Schema: API

Label: Billing Address

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

Table Columns

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

Name	Data Type	Label	Required	Documentation
ADDRESSLINE	varchar2	Addressline		
ADDRESSLINE2	varchar2	Addressline 2		
ADDRESSLINE3	varchar2	Addressline 3		
CITY	varchar2	City		
COMPANYGUID	varchar2	Company GUID		
COUNTRYCODE	varchar2	Country Code		
COUNTRYGUID	varchar2	Country GUID		
COUNTRYNAME	varchar2	Country Name		
COUNTRYREGIONGUID	varchar2	Country Region GUID		
COUNTRYREGIONNAME	varchar2	Country Region Name		
FAX	varchar2	Fax		
GUID	varchar2	GUID		
ISBILLINGADDRESS	boolean	Is Billing Address		
ISPOSTALADDRESS	boolean	Is Postal Address		
ISVISITADDRESS	boolean	Is Visit Address		
PHONE	varchar2	Phone		
POSTALCODE	varchar2	Postal Code		

BillingAndExpenseForecastByCaseGUID: Visma Severa Billing and Expense Forecast by Project GUID

Catalog: Severa

Schema: API

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Label: Billing and Expense Forecast by Project GUID

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

Filters on API Server: caseGuid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function BillingAndExpenseForecastByCaseGUID. 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
caseGuid		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
BILLINGFORECAST	number	Billing Forecast		
EXPENSEFORECAST	number	Expense Forecast		
GUID	varchar2	GUID		
INHOUSEREVENUEFORECAST	number	Inhouse Revenue Forecast		
MONTH	number	Month		
NOTES	varchar2	Notes		
OUTSOURCINGREVENUEFORECAST	number	Outsourcing Revenue Forecast		
YEAR	number	Year		

BusinessUnitByCode: Visma Severa Business Unit by Code

Catalog: Severa

Schema: API

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Label: Business Unit by Code

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

Filters on API Server: businessUnitCode

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function BusinessUnitByCode. 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
businessUnitCode		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CODE	varchar2	Code		
COSTCENTERGUID	varchar2	Cost Centre GUID		
CURRENCYCODE	varchar2	Currency Code		
CURRENCYGUID	varchar2	Currency GUID		
FORMATTINGCULTURE	varchar2	Formatting Culture		
GUID	varchar2	GUID		
INSERTTS	datetime	Inserted		
ISACTIVE	boolean	Is Active		
LANGUAGECODE	varchar2	Language Code		
LANGUAGEGUID	varchar2	Language GUID		
NAME	varchar2	Name		
NEXTINVOICENUMBER	number	Next Invoice Number		
OVERDUEINTEREST	number	Overdue Interest		
PARENTBUSINESSUNITGUID	varchar2	Parent Business Unit GUID		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
PAYMENTTERM	number	Payment Term		
SHOWLOGOONINVOICES	boolean	Show Logo on Invoices		
SHOWLOGOONPDFREPORTS	boolean	Show Logo on PDF Reports		
TIMEZONEGUID	varchar2	Timezone GUID		
UPDATEDTS	datetime	Updated		
VATNUMBER	varchar2	VAT Number		

BusinessUnitByGUID: Visma Severa Business Unit by GUID

Catalog: Severa

Schema: API

Label: Business Unit by GUID

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

Filters on API Server: businessUnitGuid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function BusinessUnitByGUID. 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
businessUnitGuid		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CODE	varchar2	Code		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
COSTCENTERGUID	varchar2	Cost Centre GUID		
CURRENCYCODE	varchar2	Currency Code		
CURRENCYGUID	varchar2	Currency GUID		
FORMATTINGCULTURE	varchar2	Formatting Culture		
GUID	varchar2	GUID		
INSERTTS	datetime	Inserted		
ISACTIVE	boolean	Is Active		
LANGUAGECODE	varchar2	Language Code		
LANGUAGEGUID	varchar2	Language GUID		
NAME	varchar2	Name		
NEXTINVOICENUMBER	number	Next Invoice Number		
OVERDUEINTEREST	number	Overdue Interest		
PARENTBUSINESSUNITGUID	varchar2	Parent Business Unit GUID		
PAYMENTTERM	number	Payment Term		
SHOWLOGOONINVOICES	boolean	Show Logo on Invoices		
SHOWLOGOONPDFREPORTS	boolean	Show Logo on PDF Reports		
TIMEZONEGUID	varchar2	Timezone GUID		
UPDATEDTS	datetime	Updated		
VATNUMBER	varchar2	VAT Number		

BusinessUnitHierarchy: Visma Severa Business Unit Hierarchy

Catalog: Severa

Schema: API

Label: Business Unit Hierarchy

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

Filters on API Server: rootBusinessUnitGuid- options

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function BusinessUnitHierarchy. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
options		☐		
rootBusinessUnitGuid		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CODE	varchar2	Code		
COSTCENTERGUID	varchar2	Cost Centre GUID		
CURRENCYCODE	varchar2	Currency Code		
CURRENCYGUID	varchar2	Currency GUID		
FORMATTINGCULTURE	varchar2	Formatting Culture		
GUID	varchar2	GUID		
INSERTTS	datetime	Inserted		
ISACTIVE	boolean	Is Active		
LANGUAGECODE	varchar2	Language Code		
LANGUAGEGUID	varchar2	Language GUID		
NAME	varchar2	Name		
NEXTINVOICENUMBER	number	Next Invoice Number		
OVERDUEINTEREST	number	Overdue Interest		
PARENTBUSINESSUNITGUID	varchar2	Parent Business Unit GUID		
PAYMENTTERM	number	Payment Term		
SHOWLOGOONINVOICES	boolean	Show Logo on Invoices		
SHOWLOGOONPDFREPORTS	boolean	Show Logo on PDF Reports		
TIMEZONEGUID	varchar2	Timezone GUID		
UPDATEDTS	datetime	Updated		
VATNUMBER	varchar2	VAT Number		

BusinessUnitsChangedSince: Visma Severa Business Units Changed since

Catalog: Severa

Schema: API

Label: Business Units Changed since

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Filters on API Server: businessUnitGUID- startDateTime- options

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function BusinessUnitsChangedSince. 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
businessUnitGUID		☐		
options		☐		
startDateTime		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CODE	varchar2	Code		
COSTCENTERGUID	varchar2	Cost Centre GUID		
CURRENCYCODE	varchar2	Currency Code		
CURRENCYGUID	varchar2	Currency GUID		
FORMATTINGCULTURE	varchar2	Formatting Culture		
GUID	varchar2	GUID		
INSERTTS	datetime	Inserted		
ISACTIVE	boolean	Is Active		
LANGUAGECODE	varchar2	Language Code		
LANGUAGEGUID	varchar2	Language GUID		
NAME	varchar2	Name		
NEXTINVOICENUMBER	number	Next Invoice Number		
OVERDUEINTEREST	number	Overdue Interest		
PARENTBUSINESSUNITGUID	varchar2	Parent Business Unit GUID		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
PAYMENTTERM	number	Payment Term		
SHOWLOGOONINVOICES	boolean	Show Logo on Invoices		
SHOWLOGOONPDFREPORTS	boolean	Show Logo on PDF Reports		
TIMEZONEGUID	varchar2	Timezone GUID		
UPDATEDTS	datetime	Updated		
VATNUMBER	varchar2	VAT Number		

CaseByGUID: Visma Severa Project by GUID

Catalog: Severa

Schema: API

Label: Project by GUID

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

Filters on API Server: caseGuid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CaseByGUID. 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
caseGuid		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	varchar2	Account GUID		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
ACCOUNTNAME	varchar2	Account Name		
ACCOUNTNUMBER	number	Account Number		
BILLINGADDRESS	varchar2	Billing Address		
BILLINGADDRESSGUID	varchar2	Billing Address GUID		
BILLINGINTERNALNOTES	varchar2	Billing Internal Notes		
BILLINGNOTESBEFOREGRID	varchar2	Billing Notes before Grid		
BUSINESSUNITGUID	varchar2	Business Unit GUID		
BUSINESSUNITNAME	varchar2	Business Unit Name		
BUSINESSUNITNUMBER	varchar2	Business Unit Number		
CASECOMPANYCURRENCYGUID	varchar2	Project Company Currency GUID		
CASENUMBER	number	Project Number		
CASEOWNERUSERCODE	varchar2	Project Owner User Code		
CASEOWNERUSERGUID	varchar2	Project Owner User GUID		
CLOSEDDATE	datetime	Closed Date		
COMPLETIONESTIMATE	number	Completion Estimate		
CONTACTGUID	varchar2	Contact GUID		
COSTCENTERGUID	varchar2	Cost Centre GUID		
COSTCENTERNUMBER	varchar2	Cost Centre Number		
CURRENCYGUID	varchar2	Currency GUID		
CURRENCYSHORTFORM	varchar2	Currency Short Form		
CURRENCYSHORTFORMOFCASECOMPANYCURRENCY	varchar2	Currency Short Form of Project Company Currency		
CURRENTCASESTATUSDESCRIPTION	varchar2	Current Project Status Description		
CURRENTCASESTATUSGUID	varchar2	Current Project Status GUID		
CURRENTCASESTATUSINSERTTS	datetime	Current Project Status Inserted		
DEADLINEDATE	datetime	Deadline Date		
DESCRIPTION	varchar2	Ddescription		
EXPECTEDVALUE	number	Expected Value		
GUID	varchar2	GUID		
INCLUDEEXCEPTIONS	boolean	Include Exceptions		
INCLUDEOVERTIME	boolean	Include Overtime		
INSERTTS	datetime	Inserted		
INTERNALNAME	varchar2	Internal Name		
INVOICETEMPLATEGUID	varchar2	Invoice Template GUID		
ISCLOSED	boolean	Is Closed		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
ISDAILYALLOWANCEBILLABLE	boolean	Is Daily Allowance Billable		
ISINTERNAL	boolean	Is Internal		
ISMILEAGEBILLABLE	boolean	Is Mileage Billable		
ISOTHERTRAVELEXPENSESBILLABLE	boolean	Are Other Travel Expenses Billable		
LATESTESTIMATIONDATE	datetime	Latest Estimation Date		
LEADRATING	number	Lead Rating		
LEADSOURCEGUID	varchar2	Lead Source GUID		
NAME	varchar2	Name		
ORDERNUMBER	varchar2	Order Number		
OURREFERENCE	varchar2	Our Reference		
OVERDUEINTEREST	number	Overdue Interest		
PAYMENTTERM	number	Payment Term		
PRICELISTGUID	varchar2	Pricelist GUID		
PRICINGRULE	varchar2	Pricing Rule		
PRICINGTYPE	varchar2	Pricing Type		
PRIORITY	number	Priority		
PROBABILITY	number	Probability		
ROOTTASKGUID	varchar2	Root Task GUID		
SALESCLOSEDATE	datetime	Sales Close Date		
SALESPERSONUSERCODE	varchar2	Sales Person User Code		
SALESPERSONUSERGUID	varchar2	Sales Person User GUID		
SALESPROCESSGUID	varchar2	Sales Process GUID		
SALESSTATUSGUID	varchar2	Sales Status GUID		
SHAREDTOEXTRANET	boolean	Shared to Extranet		
STARTDATE	datetime	Start Date		
UPDATEDTS	datetime	Updated		
URLTOCASE	varchar2	URL to Project		
USEDEFAULTPRODUCTSINWORKTIMEENTRY	boolean	Use Default Products in Work Time Entry		
USEDEFAULTWORKTYPES	boolean	Use Default Work Types		
YOURREFERENCE	varchar2	Your Reference		

CaseByNumber: Visma Severa Project by Number

Catalog: Severa

Schema: API

Label: Project by Number

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Filters on API Server: caseNumber

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CaseByNumber. 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
caseNumber		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	varchar2	Account GUID		
ACCOUNTNAME	varchar2	Account Name		
ACCOUNTNUMBER	number	Account Number		
BILLINGADDRESS	varchar2	Billing Address		
BILLINGADDRESSGUID	varchar2	Billing Address GUID		
BILLINGINTERNALNOTES	varchar2	Billing Internal Notes		
BILLINGNOTESBEFOREGRID	varchar2	Billing Notes before Grid		
BUSINESSUNITGUID	varchar2	Business Unit GUID		
BUSINESSUNITNAME	varchar2	Business Unit Name		
BUSINESSUNITNUMBER	varchar2	Business Unit Number		
CASECOMPANYCURRENCYGUID	varchar2	Project Company Currency GUID		
CASENUMBER	number	Project Number		
CASEOWNERUSERCODE	varchar2	Project Owner User Code		
CASEOWNERUSERGUID	varchar2	Project Owner User GUID		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
CLOSEDDATE	datetime	Closed Date		
COMPLETIONESTIMATE	number	Completion Estimate		
CONTACTGUID	varchar2	Contact GUID		
COSTCENTERGUID	varchar2	Cost Centre GUID		
COSTCENTERNUMBER	varchar2	Cost Centre Number		
CURRENCYGUID	varchar2	Currency GUID		
CURRENCYSHORTFORM	varchar2	Currency Short Form		
CURRENCYSHORTFORMOFCASECOMPANYCURRENCY	varchar2	Currency Short Form of Project Company Currency		
CURRENTCASESTATUSDESCRIPTION	varchar2	Current Project Status Description		
CURRENTCASESTATUSGUID	varchar2	Current Project Status GUID		
CURRENTCASESTATUSINSERTTS	datetime	Current Project Status Inserted		
DEADLINEDATE	datetime	Deadline Date		
DESCRIPTION	varchar2	Ddescription		
EXPECTEDVALUE	number	Expected Value		
GUID	varchar2	GUID		
INCLUDEEXCEPTIONS	boolean	Include Exceptions		
INCLUDEOVERTIME	boolean	Include Overtime		
INSERTTS	datetime	Inserted		
INTERNALNAME	varchar2	Internal Name		
INVOICETEMPLATEGUID	varchar2	Invoice Template GUID		
ISCLOSED	boolean	Is Closed		
ISDAILYALLOWANCEBILLABLE	boolean	Is Daily Allowance Billable		
ISINTERNAL	boolean	Is Internal		
ISMILEAGEBILLABLE	boolean	Is Mileage Billable		
ISOTHERTRAVELEXPENSESBILLABLE	boolean	Are Other Travel Expenses Billable		
LATESTESTIMATIONDATE	datetime	Latest Estimation Date		
LEADRATING	number	Lead Rating		
LEADSOURCEGUID	varchar2	Lead Source GUID		
NAME	varchar2	Name		
ORDERNUMBER	varchar2	Order Number		
OURREFERENCE	varchar2	Our Reference		
OVERDUEINTEREST	number	Overdue Interest		
PAYMENTTERM	number	Payment Term		
PRICELISTGUID	varchar2	Pricelist GUID		
PRICINGRULE	varchar2	Pricing Rule		
PRICINGTYPE	varchar2	Pricing Type		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
PRIORITY	number	Priority		
PROBABILITY	number	Probability		
ROOTTASKGUID	varchar2	Root Task GUID		
SALESCLOSEDATE	datetime	Sales Close Date		
SALESPERSONUSERCODE	varchar2	Sales Person User Code		
SALESPERSONUSERGUID	varchar2	Sales Person User GUID		
SALESPROCESSGUID	varchar2	Sales Process GUID		
SALESSTATUSGUID	varchar2	Sales Status GUID		
SHAREDTOEXTRANET	boolean	Shared to Extranet		
STARTDATE	datetime	Start Date		
UPDATEDTS	datetime	Updated		
URLTOCASE	varchar2	URL to Project		
USEDEFAULTPRODUCTSINWORKTIMEENTRY	boolean	Use Default Products in Work Time Entry		
USEDEFAULTWORKTYPES	boolean	Use Default Work Types		
YOURREFERENCE	varchar2	Your Reference		

CaseByTaskGUID: Visma Severa Project by Task GUID

Catalog: Severa

Schema: API

Label: Project by Task GUID

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

Filters on API Server: taskGuid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CaseByTaskGUID. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
taskGuid		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	varchar2	Account GUID		
ACCOUNTNAME	varchar2	Account Name		
ACCOUNTNUMBER	number	Account Number		
BILLINGADDRESS	varchar2	Billing Address		
BILLINGADDRESSGUID	varchar2	Billing Address GUID		
BILLINGINTERNALNOTES	varchar2	Billing Internal Notes		
BILLINGNOTESBEFOREGRID	varchar2	Billing Notes before Grid		
BUSINESSUNITGUID	varchar2	Business Unit GUID		
BUSINESSUNITNAME	varchar2	Business Unit Name		
BUSINESSUNITNUMBER	varchar2	Business Unit Number		
CASECOMPANYCURRENCYGUID	varchar2	Project Company Currency GUID		
CASENUMBER	number	Project Number		
CASEOWNERUSERCODE	varchar2	Project Owner User Code		
CASEOWNERUSERGUID	varchar2	Project Owner User GUID		
CLOSEDDATE	datetime	Closed Date		
COMPLETIONESTIMATE	number	Completion Estimate		
CONTACTGUID	varchar2	Contact GUID		
COSTCENTERGUID	varchar2	Cost Centre GUID		
COSTCENTERNUMBER	varchar2	Cost Centre Number		
CURRENCYGUID	varchar2	Currency GUID		
CURRENCYSHORTFORM	varchar2	Currency Short Form		
CURRENCYSHORTFORMOFCASECOMPANYCURRENCY	varchar2	Currency Short Form of Project Company Currency		
CURRENTCASESTATUSDESCRIPTION	varchar2	Current Project Status Description		
CURRENTCASESTATUSGUID	varchar2	Current Project Status GUID		
CURRENTCASESTATUSINSERTTS	datetime	Current Project Status Inserted		
DEADLINEDATE	datetime	Deadline Date		
DESCRIPTION	varchar2	Ddescription		
EXPECTEDVALUE	number	Expected Value		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
GUID	varchar2	GUID		
INCLUDEEXCEPTIONS	boolean	Include Exceptions		
INCLUDEOVERTIME	boolean	Include Overtime		
INSERTTS	datetime	Inserted		
INTERNALNAME	varchar2	Internal Name		
INVOICETEMPLATEGUID	varchar2	Invoice Template GUID		
ISCLOSED	boolean	Is Closed		
ISDAILYALLOWANCEBILLABLE	boolean	Is Daily Allowance Billable		
ISINTERNAL	boolean	Is Internal		
ISMILEAGEBILLABLE	boolean	Is Mileage Billable		
ISOTHERTRAVELEXPENSESBILLABLE	boolean	Are Other Travel Expenses Billable		
LATESTESTIMATIONDATE	datetime	Latest Estimation Date		
LEADRATING	number	Lead Rating		
LEADSOURCEGUID	varchar2	Lead Source GUID		
NAME	varchar2	Name		
ORDERNUMBER	varchar2	Order Number		
OURREFERENCE	varchar2	Our Reference		
OVERDUEINTEREST	number	Overdue Interest		
PAYMENTTERM	number	Payment Term		
PRICELISTGUID	varchar2	Pricelist GUID		
PRICINGRULE	varchar2	Pricing Rule		
PRICINGTYPE	varchar2	Pricing Type		
PRIORITY	number	Priority		
PROBABILITY	number	Probability		
ROOTTASKGUID	varchar2	Root Task GUID		
SALESCLOSEDATE	datetime	Sales Close Date		
SALESPERSONUSERCODE	varchar2	Sales Person User Code		
SALESPERSONUSERGUID	varchar2	Sales Person User GUID		
SALESPROCESSGUID	varchar2	Sales Process GUID		
SALESSTATUSGUID	varchar2	Sales Status GUID		
SHAREDTOEXTRANET	boolean	Shared to Extranet		
STARTDATE	datetime	Start Date		
UPDATEDTS	datetime	Updated		
URLTOCASE	varchar2	URL to Project		
USEDEFAULTPRODUCTSINWORKTIMEENTRY	boolean	Use Default Products in Work Time Entry		
USEDEFAULTWORKTYPES	boolean	Use Default Work Types		
YOURREFERENCE	varchar2	Your Reference		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

CaseMemberUsersByCaseGUID: Visma Severa Project Member Users by Project GUID

Catalog: Severa

Schema: API

Label: Project Member Users by Project GUID

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

Filters on API Server: caseGuid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CaseMemberUsersByCaseGUID. 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
caseGuid		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ADDRESSLINE	varchar2	Addressline		
BANKACCOUNTNUMBER	varchar2	Bank Account Number		
BUSINESSUNITCODE	varchar2	Business Unit Code		
BUSINESSUNITGUID	varchar2	Business Unit GUID		
BUSINESSUNITNAME	varchar2	Business Unit Name		
CODE	varchar2	Code		
COUNTRYCODE	varchar2	Country Code		
COUNTRYGUID	varchar2	Country GUID		
COUNTRYREGIONGUID	varchar2	Country Region GUID		
COUNTRYREGIONNAME	varchar2	Country Region Name		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
CULTURE	varchar2	Culture		
DEFAULTACTIVITYTYPEGUID	varchar2	Default Activity Type GUID		
EMAIL	varchar2	Email		
FIRSTNAME	varchar2	First Name		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
LANGUAGECODE	varchar2	Language Code		
LANGUAGEGUID	varchar2	Language GUID		
LASTNAME	varchar2	Last Name		
LICENSETYPE	varchar2			
NOTES	varchar2	Notes		
ORGANIZATIONGUID	varchar2	Organisation GUID		
PHONE	varchar2	Phone		
PHOTOFILEGUID	varchar2	Photo File GUID		
POSTALCODE	varchar2	Postal Code		
POSTOFFICE	varchar2	Post Office		
PROFILEGUID	varchar2	Profile GUID		
SALUTATION	varchar2	Salutation		
SENDDAILYMAIL	boolean	Send Daily Mail		
SENDWEEKLYMAIL	boolean	Send Weekly Mail		
SOCIALSECURITYNUMBER	varchar2	Social Security Number		
SUPERIORUSERGUID	varchar2	Superior User GUID		
TIMEZONEGUID	varchar2	Timezone GUID		
TITLE	varchar2	Title		
WORKTYPEGUID	varchar2	Work Type GUID		

Cases: Visma Severa Projects

Catalog: Severa

Schema: API

Label: Projects

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

Filters on API Server: businessUnitGUID- criteria_InternalName- criteria_OnlyActiveCases- criteria_OnlyCustomerCases- criteria_OnlyInactiveCases- criteria_OnlyInternalCases- criteria_SalesProcess- criteria_Tags

Parameters of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The following parameters can be used to control the behaviour of the table function Cases. 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
businessUnitGUID		☐		
criteria_InternalName		☐		
criteria_OnlyActiveCases		☐		
criteria_OnlyCustomerCases		☐		
criteria_OnlyInactiveCases		☐		
criteria_OnlyInternalCases		☐		
criteria_SalesProcess		☐		
criteria_Tags		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	varchar2	Account GUID		
ACCOUNTNAME	varchar2	Account Name		
ACCOUNTNUMBER	number	Account Number		
BILLINGADDRESS	varchar2	Billing Address		
BILLINGADDRESSGUID	varchar2	Billing Address GUID		
BILLINGINTERNALNOTES	varchar2	Billing Internal Notes		
BILLINGNOTESBEFOREGRID	varchar2	Billing Notes before Grid		
BUSINESSUNITGUID	varchar2	Business Unit GUID		
BUSINESSUNITNAME	varchar2	Business Unit Name		
BUSINESSUNITNUMBER	varchar2	Business Unit Number		
CASECOMPANYCURRENCYGUID	varchar2	Project Company Currency GUID		
CASENUMBER	number	Project Number		
CASEOWNERUSERCODE	varchar2	Project Owner User Code		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
CASEOWNERUSERGUID	varchar2	Project Owner User GUID		
CLOSEDDATE	datetime	Closed Date		
COMPLETIONESTIMATE	number	Completion Estimate		
CONTACTGUID	varchar2	Contact GUID		
COSTCENTERGUID	varchar2	Cost Centre GUID		
COSTCENTERNUMBER	varchar2	Cost Centre Number		
CURRENCYGUID	varchar2	Currency GUID		
CURRENCYSHORTFORM	varchar2	Currency Short Form		
CURRENCYSHORTFORMOFCASECOMPANYCURRENCY	varchar2	Currency Short Form of Project Company Currency		
CURRENTCASESTATUSDESCRIPTION	varchar2	Current Project Status Description		
CURRENTCASESTATUSGUID	varchar2	Current Project Status GUID		
CURRENTCASESTATUSINSERTTS	datetime	Current Project Status Inserted		
DEADLINEDATE	datetime	Deadline Date		
DESCRIPTION	varchar2	Ddescription		
EXPECTEDVALUE	number	Expected Value		
GUID	varchar2	GUID		
INCLUDEEXCEPTIONS	boolean	Include Exceptions		
INCLUDEOVERTIME	boolean	Include Overtime		
INSERTTS	datetime	Inserted		
INTERNALNAME	varchar2	Internal Name		
INVOICETEMPLATEGUID	varchar2	Invoice Template GUID		
ISCLOSED	boolean	Is Closed		
ISDAILYALLOWANCEBILLABLE	boolean	Is Daily Allowance Billable		
ISINTERNAL	boolean	Is Internal		
ISMILEAGEBILLABLE	boolean	Is Mileage Billable		
ISOTHERTRAVELEXPENSESBILLABLE	boolean	Are Other Travel Expenses Billable		
LATESTESTIMATIONDATE	datetime	Latest Estimation Date		
LEADRATING	number	Lead Rating		
LEADSOURCEGUID	varchar2	Lead Source GUID		
NAME	varchar2	Name		
ORDERNUMBER	varchar2	Order Number		
OURREFERENCE	varchar2	Our Reference		
OVERDUEINTEREST	number	Overdue Interest		
PAYMENTTERM	number	Payment Term		
PRICELISTGUID	varchar2	Pricelist GUID		
PRICINGRULE	varchar2	Pricing Rule		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
PRICINGTYPE	varchar2	Pricing Type		
PRIORITY	number	Priority		
PROBABILITY	number	Probability		
ROOTTASKGUID	varchar2	Root Task GUID		
SALESCLOSEDATE	datetime	Sales Close Date		
SALESPERSONUSERCODE	varchar2	Sales Person User Code		
SALESPERSONUSERGUID	varchar2	Sales Person User GUID		
SALESPROCESSGUID	varchar2	Sales Process GUID		
SALESSTATUSGUID	varchar2	Sales Status GUID		
SHAREDTOEXTRANET	boolean	Shared to Extranet		
STARTDATE	datetime	Start Date		
UPDATEDTS	datetime	Updated		
URLTOCASE	varchar2	URL to Project		
USEDEFAULTPRODUCTSINWORKTIMEENTRY	boolean	Use Default Products in Work Time Entry		
USEDEFAULTWORKTYPES	boolean	Use Default Work Types		
YOURREFERENCE	varchar2	Your Reference		

CasesByAccountGUID: Visma Severa Projects by Account GUID

Catalog: Severa

Schema: API

Label: Projects by Account GUID

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

Filters on API Server: businessUnitGUID- accountGuid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CasesByAccountGUID. 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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
accountGuid		☐		
businessUnitGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	varchar2	Account GUID		
ACCOUNTNAME	varchar2	Account Name		
ACCOUNTNUMBER	number	Account Number		
BILLINGADDRESS	varchar2	Billing Address		
BILLINGADDRESSGUID	varchar2	Billing Address GUID		
BILLINGINTERNALNOTES	varchar2	Billing Internal Notes		
BILLINGNOTESBEFOREGRID	varchar2	Billing Notes before Grid		
BUSINESSUNITGUID	varchar2	Business Unit GUID		
BUSINESSUNITNAME	varchar2	Business Unit Name		
BUSINESSUNITNUMBER	varchar2	Business Unit Number		
CASECOMPANYCURRENCYGUID	varchar2	Project Company Currency GUID		
CASENUMBER	number	Project Number		
CASEOWNERUSERCODE	varchar2	Project Owner User Code		
CASEOWNERUSERGUID	varchar2	Project Owner User GUID		
CLOSEDDATE	datetime	Closed Date		
COMPLETIONESTIMATE	number	Completion Estimate		
CONTACTGUID	varchar2	Contact GUID		
COSTCENTERGUID	varchar2	Cost Centre GUID		
COSTCENTERNUMBER	varchar2	Cost Centre Number		
CURRENCYGUID	varchar2	Currency GUID		
CURRENCYSHORTFORM	varchar2	Currency Short Form		
CURRENCYSHORTFORMOFCASECOMPANYCURRENCY	varchar2	Currency Short Form of Project Company Currency		
CURRENTCASESTATUSDESCRIPTION	varchar2	Current Project Status Description		
CURRENTCASESTATUSGUID	varchar2	Current Project Status GUID		
CURRENTCASESTATUSINSERTTS	datetime	Current Project Status Inserted		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
DEADLINEDATE	datetime	Deadline Date		
DESCRIPTION	varchar2	Ddescription		
EXPECTEDVALUE	number	Expected Value		
GUID	varchar2	GUID		
INCLUDEEXCEPTIONS	boolean	Include Exceptions		
INCLUDEOVERTIME	boolean	Include Overtime		
INSERTTS	datetime	Inserted		
INTERNALNAME	varchar2	Internal Name		
INVOICETEMPLATEGUID	varchar2	Invoice Template GUID		
ISCLOSED	boolean	Is Closed		
ISDAILYALLOWANCEBILLABLE	boolean	Is Daily Allowance Billable		
ISINTERNAL	boolean	Is Internal		
ISMILEAGEBILLABLE	boolean	Is Mileage Billable		
ISOTHERTRAVELEXPENSESBILLABLE	boolean	Are Other Travel Expenses Billable		
LATESTESTIMATIONDATE	datetime	Latest Estimation Date		
LEADRATING	number	Lead Rating		
LEADSOURCEGUID	varchar2	Lead Source GUID		
NAME	varchar2	Name		
ORDERNUMBER	varchar2	Order Number		
OURREFERENCE	varchar2	Our Reference		
OVERDUEINTEREST	number	Overdue Interest		
PAYMENTTERM	number	Payment Term		
PRICELISTGUID	varchar2	Pricelist GUID		
PRICINGRULE	varchar2	Pricing Rule		
PRICINGTYPE	varchar2	Pricing Type		
PRIORITY	number	Priority		
PROBABILITY	number	Probability		
ROOTTASKGUID	varchar2	Root Task GUID		
SALESCLOSEDATE	datetime	Sales Close Date		
SALESPERSONUSERCODE	varchar2	Sales Person User Code		
SALESPERSONUSERGUID	varchar2	Sales Person User GUID		
SALESPROCESSGUID	varchar2	Sales Process GUID		
SALESSTATUSGUID	varchar2	Sales Status GUID		
SHAREDTOEXTRANET	boolean	Shared to Extranet		
STARTDATE	datetime	Start Date		
UPDATEDTS	datetime	Updated		
URLTOCASE	varchar2	URL to Project		
USEDEFAULTPRODUCTSINWORKTIMEENTRY	boolean	Use Default Products in Work Time Entry		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
USEDEFAULTWORKTYPES	boolean	Use Default Work Types		
YOURREFERENCE	varchar2	Your Reference		

CasesChangedSince: Visma Severa Projects Changed since

Catalog: Severa

Schema: API

Label: Projects Changed since

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

Filters on API Server: businessUnitGUID- startDateTime- options

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CasesChangedSince. 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
businessUnitGUID		☐		
options		☐		
startDateTime		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	varchar2	Account GUID		
ACCOUNTNAME	varchar2	Account Name		
ACCOUNTNUMBER	number	Account Number		
BILLINGADDRESS	varchar2	Billing Address		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
BILLINGADDRESSGUID	varchar2	Billing Address GUID		
BILLINGINTERNALNOTES	varchar2	Billing Internal Notes		
BILLINGNOTESBEFOREGRID	varchar2	Billing Notes before Grid		
BUSINESSUNITGUID	varchar2	Business Unit GUID		
BUSINESSUNITNAME	varchar2	Business Unit Name		
BUSINESSUNITNUMBER	varchar2	Business Unit Number		
CASECOMPANYCURRENCYGUID	varchar2	Project Company Currency GUID		
CASENUMBER	number	Project Number		
CASEOWNERUSERCODE	varchar2	Project Owner User Code		
CASEOWNERUSERGUID	varchar2	Project Owner User GUID		
CLOSEDDATE	datetime	Closed Date		
COMPLETIONESTIMATE	number	Completion Estimate		
CONTACTGUID	varchar2	Contact GUID		
COSTCENTERGUID	varchar2	Cost Centre GUID		
COSTCENTERNUMBER	varchar2	Cost Centre Number		
CURRENCYGUID	varchar2	Currency GUID		
CURRENCYSHORTFORM	varchar2	Currency Short Form		
CURRENCYSHORTFORMOFCASECOMPANYCURRENCY	varchar2	Currency Short Form of Project Company Currency		
CURRENTCASESTATUSDESCRIPTION	varchar2	Current Project Status Description		
CURRENTCASESTATUSGUID	varchar2	Current Project Status GUID		
CURRENTCASESTATUSINSERTTS	datetime	Current Project Status Inserted		
DEADLINEDATE	datetime	Deadline Date		
DESCRIPTION	varchar2	Ddescription		
EXPECTEDVALUE	number	Expected Value		
GUID	varchar2	GUID		
INCLUDEEXCEPTIONS	boolean	Include Exceptions		
INCLUDEOVERTIME	boolean	Include Overtime		
INSERTTS	datetime	Inserted		
INTERNALNAME	varchar2	Internal Name		
INVOICETEMPLATEGUID	varchar2	Invoice Template GUID		
ISCLOSED	boolean	Is Closed		
ISDAILYALLOWANCEBILLABLE	boolean	Is Daily Allowance Billable		
ISINTERNAL	boolean	Is Internal		
ISMILEAGEBILLABLE	boolean	Is Mileage Billable		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
ISOTHERTRAVELEXPENSESBILLABLE	boolean	Are Other Travel Expenses Billable		
LATESTESTIMATIONDATE	datetime	Latest Estimation Date		
LEADRATING	number	Lead Rating		
LEADSOURCEGUID	varchar2	Lead Source GUID		
NAME	varchar2	Name		
ORDERNUMBER	varchar2	Order Number		
OURREFERENCE	varchar2	Our Reference		
OVERDUEINTEREST	number	Overdue Interest		
PAYMENTTERM	number	Payment Term		
PRICELISTGUID	varchar2	Pricelist GUID		
PRICINGRULE	varchar2	Pricing Rule		
PRICINGTYPE	varchar2	Pricing Type		
PRIORITY	number	Priority		
PROBABILITY	number	Probability		
ROOTTASKGUID	varchar2	Root Task GUID		
SALESCLOSEDATE	datetime	Sales Close Date		
SALESPERSONUSERCODE	varchar2	Sales Person User Code		
SALESPERSONUSERGUID	varchar2	Sales Person User GUID		
SALESPROCESSGUID	varchar2	Sales Process GUID		
SALESSTATUSGUID	varchar2	Sales Status GUID		
SHAREDTOEXTRANET	boolean	Shared to Extranet		
STARTDATE	datetime	Start Date		
UPDATEDTS	datetime	Updated		
URLTOCASE	varchar2	URL to Project		
USEDEFAULTPRODUCTSINWORKTIMEENTRY	boolean	Use Default Products in Work Time Entry		
USEDEFAULTWORKTYPES	boolean	Use Default Work Types		
YOURREFERENCE	varchar2	Your Reference		

CaseStatusByGUID: Visma Severa Project Status by GUID

Catalog: Severa

Schema: API

Label: Project Status by GUID

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

Filters on API Server: caseStatusGUID

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CaseStatusByGUID. 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
caseStatusGUID		□		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
GUID	varchar2	GUID		
ICON	varchar2	Icon		
ISACTIVE	boolean	Is Active		
NAME	varchar2	Name		

CaseStatuses: Visma Severa Project Statuses

Catalog: Severa

Schema: API

Label: Project Statuses

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

Table Columns

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

Name	Data Type	Label	Required	Documentation
GUID	varchar2	GUID		
ICON	varchar2	Icon		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
ISACTIVE	boolean	Is Active		
NAME	varchar2	Name		

CaseTags: Visma Severa Project Tags

Catalog: Severa

Schema: API

Label: Project Tags

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

Table Columns

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

Name	Data Type	Label	Required	Documentation
CONTEXT	varchar2	Context		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
KEYWORD	varchar2	Keyword		
WEIGHT	number	Weight		

CompanyByGUID: Visma Severa Company by GUID

Catalog: Severa

Schema: API

Label: Company by GUID

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

Filters on API Server: companyGuid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CompanyByGUID. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
companyGuid		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ANNUALREVENUE	number	Annual Revenue		
ANNUALREVENUE2	number			
EMAIL	varchar2	Email		
EMPLOYEES	number	Employees		
GUID	varchar2	GUID		
HEADOFFICEADDRESSGUID	varchar2	Head Office Address GUID		
HIERARCHYGUID	varchar2	Hierarchy GUID		
INDUSTRYGUID	varchar2	Industry GUID		
KVKNUMBER	varchar2			
LANGUAGEGUID	varchar2	Language GUID		
NAME	varchar2	Name		
TIMEZONEGUID	varchar2	Timezone GUID		
VATNUMBER	varchar2	VAT Number		
WEBSITE	varchar2	Website		

ContactByGUID: Visma Severa Contact by GUID

Catalog: Severa

Schema: API

Label: Contact by GUID

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

Filters on API Server: contactGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ContactByGUID. 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-

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
contactGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	varchar2	Account GUID		
ADDRESSGUID	varchar2	Address GUID		
COMMUNICATIONMETHODS	varchar2			
CONTACTROLEGUID	varchar2	Contact Role GUID		
DATEOFBIRTH	datetime	Date of Birth		
DESCRIPTION	varchar2	Ddescription		
FIRSTNAME	varchar2	First Name		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
JOBTITLE	varchar2	Job Title		
KEYWORDS	varchar2			
LANGUAGEGUID	varchar2	Language GUID		
LASTNAME	varchar2	Last Name		
SALUTATION	varchar2	Salutation		
SATISFACTIONLEVEL	number	Satisfaction Level		
TIMEZONEGUID	varchar2	Timezone GUID		

ContactsByAccountGUID: Visma Severa Contacts by Account GUID

Catalog: Severa

Schema: API

Label: Contacts by Account GUID

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Filters on API Server: accountGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ContactsByAccountGUID. 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
accountGUID		□		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	varchar2	Account GUID		
ADDRESSGUID	varchar2	Address GUID		
COMMUNICATIONMETHODS	varchar2			
CONTACTROLEGUID	varchar2	Contact Role GUID		
DATEOFBIRTH	datetime	Date of Birth		
DESCRIPTION	varchar2	Ddescription		
FIRSTNAME	varchar2	First Name		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
JOBTITLE	varchar2	Job Title		
KEYWORDS	varchar2			
LANGUAGEGUID	varchar2	Language GUID		
LASTNAME	varchar2	Last Name		
SALUTATION	varchar2	Salutation		
SATISFACTIONLEVEL	number	Satisfaction Level		
TIMEZONEGUID	varchar2	Timezone GUID		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

ContactsByCommunicationMethodType: Visma Severa Contacts by Communication Method Type

Catalog: Severa

Schema: API

Label: Contacts by Communication Method Type

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

Filters on API Server: type- value

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ContactsByCommunicationMethodType. 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
type		☐		
value		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	varchar2	Account GUID		
ADDRESSGUID	varchar2	Address GUID		
COMMUNICATIONMETHODS	varchar2			
CONTACTROLEGUID	varchar2	Contact Role GUID		
DATEOFBIRTH	datetime	Date of Birth		
DESCRIPTION	varchar2	Ddescription		
FIRSTNAME	varchar2	First Name		
GUID	varchar2	GUID		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
ISACTIVE	boolean	Is Active		
JOBTITLE	varchar2	Job Title		
KEYWORDS	varchar2			
LANGUAGEGUID	varchar2	Language GUID		
LASTNAME	varchar2	Last Name		
SALUTATION	varchar2	Salutation		
SATISFACTIONLEVEL	number	Satisfaction Level		
TIMEZONEGUID	varchar2	Timezone GUID		

ContactsChangedSince: Visma Severa Contacts Changed since

Catalog: Severa

Schema: API

Label: Contacts Changed since

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

Filters on API Server: accountGUID- startDateTime- options

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ContactsChangedSince. 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
accountGUID		☐		
options		☐		
startDateTime		☐		

Columns of Table Function

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	varchar2	Account GUID		
ADDRESSGUID	varchar2	Address GUID		
COMMUNICATIONMETHODS	varchar2			
CONTACTROLEGUID	varchar2	Contact Role GUID		
DATEOFBIRTH	datetime	Date of Birth		
DESCRIPTION	varchar2	Ddescription		
FIRSTNAME	varchar2	First Name		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
JOBTITLE	varchar2	Job Title		
KEYWORDS	varchar2			
LANGUAGEGUID	varchar2	Language GUID		
LASTNAME	varchar2	Last Name		
SALUTATION	varchar2	Salutation		
SATISFACTIONLEVEL	number	Satisfaction Level		
TIMEZONEGUID	varchar2	Timezone GUID		

CostCenterByGUID: Visma Severa Cost Centre by GUID

Catalog: Severa

Schema: API

Label: Cost Centre by GUID

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

Filters on API Server: costCenterGuid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CostCenterByGUID. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
costCenterGuid		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
GUID	varchar2	GUID		
IDENTIFIER	varchar2	Identifier		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
NAME	varchar2	Name		

CostCenters: Visma Severa Cost Centres

Catalog: Severa

Schema: API

Label: Cost Centres

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

Table Columns

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

Name	Data Type	Label	Required	Documentation
GUID	varchar2	GUID		
IDENTIFIER	varchar2	Identifier		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
NAME	varchar2	Name		

CountryByGUID: Visma Severa Country by GUID

Catalog: Severa

Schema: API

Label: Country by GUID

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Filters on API Server: countryGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CountryByGUID. 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
countryGUID		□		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
DEFAULTLANGUAGECODE	varchar2	Default Language Code		
DEFAULTLANGUAGEGUID	varchar2	Default Language GUID		
GUID	varchar2	GUID		
ISEU	boolean	Is EU		
ISOCODEA2	varchar2	ISO Code A2		
ISOCODEA3	varchar2	ISO Code A3		
NAME	varchar2	Name		
TIMEZONEGUID	varchar2	Timezone GUID		
TRANSLATEDNAME	varchar2	Translated Name		

CountryByISO: Visma Severa Country by ISO

Catalog: Severa

Schema: API

Label: Country by ISO

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Filters on API Server: isoName

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CountryByISO. 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
isoName		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
DEFAULTLANGUAGECODE	varchar2	Default Language Code		
DEFAULTLANGUAGEGUID	varchar2	Default Language GUID		
GUID	varchar2	GUID		
ISEU	boolean	Is EU		
ISOCODEA2	varchar2	ISO Code A2		
ISOCODEA3	varchar2	ISO Code A3		
NAME	varchar2	Name		
TIMEZONEGUID	varchar2	Timezone GUID		
TRANSLATEDNAME	varchar2	Translated Name		

CountryByName: Visma Severa Country by Name

Catalog: Severa

Schema: API

Label: Country by Name

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Filters on API Server: countryName

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CountryByName. 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
countryName		□		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
DEFAULTLANGUAGECODE	varchar2	Default Language Code		
DEFAULTLANGUAGEGUID	varchar2	Default Language GUID		
GUID	varchar2	GUID		
ISEU	boolean	Is EU		
ISOCODEA2	varchar2	ISO Code A2		
ISOCODEA3	varchar2	ISO Code A3		
NAME	varchar2	Name		
TIMEZONEGUID	varchar2	Timezone GUID		
TRANSLATEDNAME	varchar2	Translated Name		

CreateInvoices: Visma Severa Create Invoices

Catalog: Severa

Schema: API

Label: Create Invoices

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Filters on API Server: invoiceInformation_Date- invoiceInformation_DueDate- invoiceInformation_InvoiceStatus- invoiceInformation_ItemGUIDs- invoiceInformation_LanguageGUID- invoiceInformation_Notes- invoiceInformation_OurReference- invoiceInformation_OverDueInterest- invoiceInformation_PaymentTerm- invoiceInformation_ReceiverAddressline- invoiceInformation_ReceiverCity- invoiceInformation_ReceiverContactEmail- invoiceInformation_ReceiverContactName- invoiceInformation_ReceiverCountry- invoiceInformation_ReceiverCustomerName- invoiceInformation_ReceiverPostalCode- invoiceInformation_ReceiverState- invoiceInformation_ReceiverVatNumber- invoiceInformation_SenderAddressline- invoiceInformation_SenderCity- invoiceInformation_SenderContactName- invoiceInformation_SenderCountry- invoiceInformation_SenderName- invoiceInformation_SenderPostalCode- invoiceInformation_SenderState- invoiceInformation_SenderVatNumber- invoiceInformation_YourReference

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CreateInvoices. 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
invoiceInformation_Date		☐		
invoiceInformation_DueDate		☐		
invoiceInformation_InvoiceStatus		☐		
invoiceInformation_ItemGUIDs		☐		
invoiceInformation_LanguageGUID		☐		
invoiceInformation_Notes		☐		
invoiceInformation_OurReference		☐		
invoiceInformation_OverDueInterest		☐		
invoiceInformation_PaymentTerm		☐		
invoiceInformation_ReceiverAddressline		☐		
invoiceInformation_ReceiverCity		☐		
invoiceInformation_ReceiverContactEmail		☐		
invoiceInformation_ReceiverContactName		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
invoiceInformation_ReceiverCountry		☐		
invoiceInformation_ReceiverCustomerName		☐		
invoiceInformation_ReceiverPostalCode		☐		
invoiceInformation_ReceiverState		☐		
invoiceInformation_ReceiverVatNumber		☐		
invoiceInformation_SenderAddressline		☐		
invoiceInformation_SenderCity		☐		
invoiceInformation_SenderContactName		☐		
invoiceInformation_SenderCountry		☐		
invoiceInformation_SenderName		☐		
invoiceInformation_SenderPostalCode		☐		
invoiceInformation_SenderState		☐		
invoiceInformation_SenderVatNumber		☐		
invoiceInformation_YourReference		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	varchar2	Account GUID		
ACCOUNTNAME	varchar2	Account Name		
ACCOUNTNUMBER	number	Account Number		
BUSINESSUNITCODE	varchar2	Business Unit Code		
BUSINESSUNITGUID	varchar2	Business Unit GUID		
BUSINESSUNITNAME	varchar2	Business Unit Name		
BUYERORGANIZATION_BUYERCONTACTPERSONNAME	varchar2			
BUYERORGANIZATION_BUYERELECTRONICINVOICEADDRESS	varchar2			
BUYERORGANIZATION_BUYERELECTRONICINVOICEOPERATOR	varchar2			
BUYERORGANIZATION_BUYEREMAILADDRESSIDENTIFIER	varchar2			
BUYERORGANIZATION_BUYERORGANISATIONNAME	varchar2			

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
BUYERORGANIZATION_BUYERORGANISATIONTAXCODE	varchar2			
BUYERORGANIZATION_BUYERPARTYIDENTIFIER	varchar2			
BUYERORGANIZATION_BUYERPHONENUMBERIDENTIFIER	varchar2			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRY	varchar2			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRYCODE	varchar2			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	varchar2			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_STREETNAME	varchar2			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_TOWNNAME	varchar2			
CULTURE	varchar2	Culture		
CURRENCYGUID	varchar2	Currency GUID		
CURRENCYNAME	varchar2	Currency Name		
CURRENCYRATE	number	Currency Rate		
CURRENCYSHORTFORM	varchar2	Currency Short Form		
DATE	datetime	Date		
DESCRIPTION	varchar2	Ddescription		
DUEDATE	datetime	Due Date		
ENTRYDATE	datetime	Entry Date		
GUID	varchar2	GUID		
INVOICEALLOWEDCASEGUIDS	varchar2			
INVOICECONFIG_BILLEXPENSE	varchar2			
INVOICECONFIG_BILLHOUR	varchar2			
INVOICECONFIG_BILLPRODUCT	varchar2			
INVOICECONFIG_EXPENSEITEMDESCRIPTIONFORMAT	varchar2			
INVOICECONFIG_HOURITEMDESCRIPTIONFORMAT	varchar2			
INVOICECONFIG_HOURPRIMARYGROUPBY	varchar2			
INVOICECONFIG_HOURSECONDARYGROUPBY	varchar2			
INVOICECONFIG_ISTAXFREE	boolean			
INVOICECONFIG_PRODUCTITEMDESCRIPTIONFORMAT	varchar2			
INVOICECONFIG_PRODUCTPRIMARYGROUPBY	varchar2			

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
INVOICECONFIG_PRODUCTSECONDARYGROUPBY	varchar2			
INVOICECONFIG_SHOWATTACHPRICEEXCLUDINGVAT	boolean			
INVOICECONFIG_SHOWATTACHUNIT	boolean			
INVOICECONFIG_SHOWATTACHUNITPRICE	boolean			
INVOICECONFIG_SHOWATTACHVAT	boolean			
INVOICECONFIG_SHOWLOGOONINVOICE	boolean			
INVOICECONFIG_SHOWPRICEEXCLUDINGVAT	boolean			
INVOICECONFIG_SHOWQUANTITY	boolean			
INVOICECONFIG_SHOWUNIT	boolean			
INVOICECONFIG_SHOWUNITPRICE	boolean			
INVOICECONFIG_SHOWVAT	boolean			
INVOICECONFIG_TAXFREEDESCRIPTION	varchar2			
INVOICECONFIG_TRAVELPRIMARYGROUPBY	varchar2			
INVOICECONFIG_TRAVELSECONDARYGROUPBY	varchar2			
INVOICEDETAILS_INVOICEDATE	datetime			
INVOICEDETAILS_INVOICEDUEDATE	datetime			
INVOICEDETAILS_INVOICENUMBER	number			
INVOICEDETAILS_INVOICETOTALVATAMOUNT	number			
INVOICEDETAILS_INVOICETOTALVATEXCLUDEDAMOUNT	number			
INVOICEDETAILS_INVOICETOTALVATINCLUDEDAMOUNT	number			
INVOICEDETAILS_INVOICETYPETEXT	varchar2			
INVOICEDETAILS_ORDERIDENTIFIER	varchar2			
INVOICEDETAILS_ORIGINCODE	varchar2			
INVOICEDETAILS_PAYMENTOVERDUEFINEFREETEXT	varchar2			
INVOICEDETAILS_PAYMENTOVERDUEFINEPERCENT	number			
INVOICEDETAILS_PAYMENTTERMSFREETEXT	varchar2			
INVOICEDETAILS_SELLERREFERENCEIDENTIFIER	varchar2			

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
INVOICEDETAILS_VATBASEAMOUNT	number			
INVOICEDETAILS_VATRATEAMOUNT	number			
INVOICEDETAILS_VATRATEPERCENT	number			
INVOICENUMBERINGBUSINESSUNITGUID	varchar2	Invoice Numbering Business Unit GUID		
INVOICEROWS	varchar2			
INVOICESTATUS	varchar2	Invoice Status		
ISCREDITNOTE	boolean	Is Credit Note		
ISREIMBURSED	boolean	Is Reimbursed		
ISTAXFREE	boolean	Is Tax Free		
LANGUAGECODE	varchar2	Language Code		
LANGUAGEGUID	varchar2	Language GUID		
MAINCASEGUID	varchar2	Main Case GUID		
NOTES	varchar2	Notes		
NOTESAFTERINVOICEROWS	varchar2	Notes after Invoice Lines		
NOTESBEFOREINVOICEROWS	varchar2	Notes before Invoice Lines		
NUMBER	number	Number		
ORDERNUMBER	varchar2	Order Number		
OURREFERENCE	varchar2	Our Reference		
OVERDUEINTEREST	number	Overdue Interest		
PAYMENTDATE	datetime	Payment Date		
PAYMENTSTATUS	varchar2	Payment Status		
PAYMENTTERM	number	Payment Term		
PONUMBER	varchar2	PO Number		
RECEIVERADDRESSLINE	varchar2	Receiver Addressline		
RECEIVERCITY	varchar2	Receiver City		
RECEIVERCONTACTEMAIL	varchar2	Receiver Contact Email		
RECEIVERCONTACTNAME	varchar2	Receiver Contact Name		
RECEIVERCOUNTRY	varchar2	Receiver Country		
RECEIVERCUSTOMERNAME	varchar2	Receiver Customer Name		
RECEIVERPOSTALCODE	varchar2	Receiver Postal Code		
RECEIVERSTATE	varchar2	Receiver State		
RECEIVERVATNUMBER	varchar2	Receiver VAT Number		
REFERENCENUMBER	varchar2	Reference Number		
REIMBURSEINVOICEGUID	varchar2	Reimburse Invoice GUID		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
SELLERCONTACT_SELLERACCOUNTDETAILS	varchar2			
SELLERCONTACT_SELLERCONTACTPERSONNAME	varchar2			
SELLERCONTACT_SELLEREMAILADDRESSIDENTIFIER	varchar2			
SELLERCONTACT_SELLERFAXNUMBER	varchar2			
SELLERCONTACT_SELLERFREETEXT	varchar2			
SELLERCONTACT_SELLERHOMETOWNNAME	varchar2			
SELLERCONTACT_SELLERPHONENUMBER	varchar2			
SELLERCONTACT_SELLERWEBADDRESSIDENTIFIER	varchar2			
SELLERORGANIZATION_SELLEREINVOICEADDRESS	varchar2			
SELLERORGANIZATION_SELLEREINVOICEOPERATOR	varchar2			
SELLERORGANIZATION_SELLERORGANISATIONNAME	varchar2			
SELLERORGANIZATION_SELLERORGANISATIONTAXCODE	varchar2			
SELLERORGANIZATION_SELLERPARTYIDENTIFIER	varchar2			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRY	varchar2			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRYCODE	varchar2			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	varchar2			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_STREETNAME	varchar2			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_TOWNNAME	varchar2			
SENDERADDRESSLINE	varchar2	Sender Addressline		
SENDERCITY	varchar2	Sender City		
SENDERCONTACTNAME	varchar2	Sender Contact Name		
SENDERCOUNTRY	varchar2	Sender Country		
SENDERNAME	varchar2	Sender Name		
SENDERPOSTALCODE	varchar2	Sender Postal Code		
SENDERSTATE	varchar2	Sender State		
SENDERVATNUMBER	varchar2	Sender VAT Number		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
TAXFREEDESCRIPTION	varchar2	Tax Free Description		
TITLE	varchar2	Title		
VIRTUALBANKBARCODE	varchar2	Virtual Bank Barcode		
YOURREFERENCE	varchar2	Your Reference		

Currencies: Visma Severa Currencies

Catalog: Severa

Schema: API

Label: Currencies

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

Table Columns

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

Name	Data Type	Label	Required	Documentation
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
ISOCODE	varchar2	ISO Code		
RATE	number	Rate		
SYMBOL	varchar2	Symbol		

CurrencyByGUID: Visma Severa Currency by GUID

Catalog: Severa

Schema: API

Label: Currency by GUID

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

Filters on API Server: currencyGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CurrencyByGUID. 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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
currencyGUID		□		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
ISOCODE	varchar2	ISO Code		
RATE	number	Rate		
SYMBOL	varchar2	Symbol		

CurrencyByIsoCode: Visma Severa Currency by ISO Code

Catalog: Severa

Schema: API

Label: Currency by ISO Code

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

Filters on API Server: currencyIsoCode

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CurrencyByIsoCode. 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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
currencyIsoCode		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
ISOCODE	varchar2	ISO Code		
RATE	number	Rate		
SYMBOL	varchar2	Symbol		

CustomerByGUID: Visma Severa Customer by GUID

Catalog: Severa

Schema: API

Label: Customer by GUID

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

Filters on API Server: accountGuid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CustomerByGUID. 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
accountGuid		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGROUPGUIDS	varchar2			
ACCOUNTOWNERUSERGUID	varchar2	Account Owner User GUID		
ACCOUNTRATING	number	Account Rating		
ANNUALREVENUE	number	Annual Revenue		
ANNUALREVENUE2	number			
COMPANYGUID	varchar2	Company GUID		
CURRENCYCODE	varchar2	Currency Code		
CURRENCYGUID	varchar2	Currency GUID		
EINVOICEADDRESS	varchar2	e-invoice Address		
EINVOICEOPERATOR	varchar2	e-invoice Operator		
EMAIL	varchar2	Email		
EMPLOYEES	number	Employees		
GUID	varchar2	GUID		
HEADOFFICEADDRESSGUID	varchar2	Head Office Address GUID		
INDUSTRYGUID	varchar2	Industry GUID		
INSERTTS	datetime	Inserted		
INVOICINGVAT	number	Invoicing VAT		
ISACTIVE	boolean	Is Active		
ISINTERNAL	boolean	Is Internal		
ISPRIVATEPERSON	boolean			
LANGUAGECODE	varchar2	Language Code		
LANGUAGEGUID	varchar2	Language GUID		
NAME	varchar2	Name		
NOTES	varchar2	Notes		
NUMBER	number	Number		
OVERDUEINTEREST	number	Overdue Interest		
PAYMENTTERM	number	Payment Term		
PRICELISTGUID	varchar2	Pricelist GUID		
REVERSECHARGE	boolean	Reverse Charge		
REVERSECHARGEDESCRIPTION	varchar2	Reverse Charge Description		
TIMEZONEGUID	varchar2	Timezone GUID		
UPDATEDTS	datetime	Updated		
URLTOACCOUNT	varchar2	URL to Account		
VATNUMBER	varchar2	VAT Number		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
WEBSITE	varchar2	Website		

Customers: Visma Severa Customers

Catalog: Severa

Schema: API

Label: Customers

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

Filters on API Server: options- startFromIndex- count- criteria_AccountGroupGuids

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function Customers. 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
count		☐		
criteria_AccountGroupGuids		☐		
options		☐		
startFromIndex		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGROUPGUIDS	varchar2			
ACCOUNTOWNERUSERGUID	varchar2	Account Owner User GUID		
ACCOUNTRATING	number	Account Rating		
ANNUALREVENUE	number	Annual Revenue		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
ANNUALREVENUE2	number			
COMPANYGUID	varchar2	Company GUID		
CURRENCYCODE	varchar2	Currency Code		
CURRENCYGUID	varchar2	Currency GUID		
EINVOICEADDRESS	varchar2	e-invoice Address		
EINVOICEOPERATOR	varchar2	e-invoice Operator		
EMAIL	varchar2	Email		
EMPLOYEES	number	Employees		
GUID	varchar2	GUID		
HEADOFFICEADDRESSGUID	varchar2	Head Office Address GUID		
INDUSTRYGUID	varchar2	Industry GUID		
INSERTTS	datetime	Inserted		
INVOICINGVAT	number	Invoicing VAT		
ISACTIVE	boolean	Is Active		
ISINTERNAL	boolean	Is Internal		
ISPRIVATEPERSON	boolean			
LANGUAGECODE	varchar2	Language Code		
LANGUAGEGUID	varchar2	Language GUID		
NAME	varchar2	Name		
NOTES	varchar2	Notes		
NUMBER	number	Number		
OVERDUEINTEREST	number	Overdue Interest		
PAYMENTTERM	number	Payment Term		
PRICELISTGUID	varchar2	Pricelist GUID		
REVERSECHARGE	boolean	Reverse Charge		
REVERSECHARGEDESCRIPTION	varchar2	Reverse Charge Description		
TIMEZONEGUID	varchar2	Timezone GUID		
UPDATEDTS	datetime	Updated		
URLTOACCOUNT	varchar2	URL to Account		
VATNUMBER	varchar2	VAT Number		
WEBSITE	varchar2	Website		

CustomersChangedSince: Visma Severa Customers Changed since

Catalog: Severa

Schema: API

Label: Customers Changed since

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Filters on API Server: startDateTime- options

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CustomersChangedSince. 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
options		☐		
startDateTime		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGROUPGUIDS	varchar2			
ACCOUNTOWNERUSERGUID	varchar2	Account Owner User GUID		
ACCOUNTRATING	number	Account Rating		
ANNUALREVENUE	number	Annual Revenue		
ANNUALREVENUE2	number			
COMPANYGUID	varchar2	Company GUID		
CURRENCYCODE	varchar2	Currency Code		
CURRENCYGUID	varchar2	Currency GUID		
EINVOICEADDRESS	varchar2	e-invoice Address		
EINVOICEOPERATOR	varchar2	e-invoice Operator		
EMAIL	varchar2	Email		
EMPLOYEES	number	Employees		
GUID	varchar2	GUID		
HEADOFFICEADDRESSGUID	varchar2	Head Office Address GUID		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
INDUSTRYGUID	varchar2	Industry GUID		
INSERTTS	datetime	Inserted		
INVOICINGVAT	number	Invoicing VAT		
ISACTIVE	boolean	Is Active		
ISINTERNAL	boolean	Is Internal		
ISPRIVATEPERSON	boolean			
LANGUAGECODE	varchar2	Language Code		
LANGUAGEGUID	varchar2	Language GUID		
NAME	varchar2	Name		
NOTES	varchar2	Notes		
NUMBER	number	Number		
OVERDUEINTEREST	number	Overdue Interest		
PAYMENTTERM	number	Payment Term		
PRICELISTGUID	varchar2	Pricelist GUID		
REVERSECHARGE	boolean	Reverse Charge		
REVERSECHARGEDESCRIPTION	varchar2	Reverse Charge Description		
TIMEZONEGUID	varchar2	Timezone GUID		
UPDATEDTS	datetime	Updated		
URLTOACCOUNT	varchar2	URL to Account		
VATNUMBER	varchar2	VAT Number		
WEBSITE	varchar2	Website		

Database: Visma Severa Database

Catalog: Severa

Schema: API

Label: Database

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

Table Columns

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

Name	Data Type	Label	Required	Documentation
DATABASERESULT	varchar2			

DeleteAccessRightsProfileByGUID: Visma Severa Delete Access Rights Profile by GUID

Catalog: Severa

Schema: API

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Label: Delete Access Rights Profile by GUID

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

Filters on API Server: guid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function DeleteAccessRightsProfileByGUID. 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
guid		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
DUMMY	varchar2	Dummy	☑	Dummy column

DeleteActivity: Visma Severa Delete Activity

Catalog: Severa

Schema: API

Label: Delete Activity

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

Filters on API Server: activityGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function DeleteActivity. A value must be provided at all times for required parameters, but optional

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
activityGUID		□		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
DELETEACTIVITYRESULT	boolean			

DeleteActivityType: Visma Severa Delete Activity Type

Catalog: Severa

Schema: API

Label: Delete Activity Type

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

Filters on API Server: activityTypeGUID- replacementActivityTypeGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function DeleteActivityType. 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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
activityTypeGUID		☐		
replacementActivityTypeGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
DELETEACTIVITYTYPERESULT	boolean			

DeleteCase: Visma Severa Delete Project

Catalog: Severa

Schema: API

Label: Delete Project

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

Filters on API Server: caseGuid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function DeleteCase. 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
caseGuid		☐		

Columns of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Name	Data Type	Label	Required	Documentation
DELETECASERESULT	boolean			

DeleteCaseStatus: Visma Severa Delete Project Status

Catalog: Severa

Schema: API

Label: Delete Project Status

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

Filters on API Server: caseStatusGUID- replacementCaseStatusGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function DeleteCaseStatus. 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
caseStatusGUID		☐		
replacementCaseStatusGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
DELETECASESTATUSRESULT	boolean			

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

DeleteContact: Visma Severa Delete Contact

Catalog: Severa

Schema: API

Label: Delete Contact

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

Filters on API Server: contactGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function DeleteContact. 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
contactGUID		□		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
DELETECONTACTRESULT	boolean			

DeleteEmployment: Visma Severa Delete Employment

Catalog: Severa

Schema: API

Label: Delete Employment

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

Filters on API Server: employmentGUID

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function DeleteEmployment. 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
employmentGUID		□		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
DELETEEMPLOYMENTRESULT	boolean			

DeleteHourEntry: Visma Severa Delete Hour Entry

Catalog: Severa

Schema: API

Label: Delete Hour Entry

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

Filters on API Server: hourEntryGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function DeleteHourEntry. 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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
hourEntryGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
DELETEHOURENTRYRESULT	boolean			

DeleteItem: Visma Severa Delete Item

Catalog: Severa

Schema: API

Label: Delete Item

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

Filters on API Server: itemGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function DeleteItem. 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
itemGUID		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
DELETEITEMRESULT	boolean			

DeletePhase: Visma Severa Delete Phase

Catalog: Severa

Schema: API

Label: Delete Phase

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

Filters on API Server: phaseGUID- phaseGUIDToTransferItemsTo

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function DeletePhase. 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
phaseGUID		☐		
phaseGUIDToTransferItemsTo		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
DELETEPHASERESULT	boolean			

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

DeleteProduct: Visma Severa Delete Product

Catalog: Severa

Schema: API

Label: Delete Product

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

Filters on API Server: productGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function DeleteProduct. 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
productGUID		□		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
DELETEPRODUCTRESULT	boolean			

DeleteProductCategory: Visma Severa Delete Product Category

Catalog: Severa

Schema: API

Label: Delete Product Category

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

Filters on API Server: productCategoryGUID

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function DeleteProductCategory. 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
productCategoryGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
DELETEPRODUCTCATEGORYRESULT	boolean			

DeleteResource: Visma Severa Delete Resource

Catalog: Severa

Schema: API

Label: Delete Resource

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

Filters on API Server: resourceGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function DeleteResource. 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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
resourceGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
DELETERESOURCERESULT	boolean			

DeleteResourceAllocation: Visma Severa Delete Resource Allocation

Catalog: Severa

Schema: API

Label: Delete Resource Allocation

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

Filters on API Server: resourceAllocationGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function DeleteResourceAllocation. 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
resourceAllocationGUID		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
DELETERESOURCEALLOCATIONRESULT	boolean			

DeleteTag: Visma Severa Delete Tag

Catalog: Severa

Schema: API

Label: Delete Tag

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

Filters on API Server: tagGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function DeleteTag. 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
tagGUID		□		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
DELETETAGRESULT	boolean			

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

DeleteTravelExpense: Visma Severa Delete Travel Expense

Catalog: Severa

Schema: API

Label: Delete Travel Expense

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

Filters on API Server: travelExpenseGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function DeleteTravelExpense. 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
travelExpenseGUID		□		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
DELETETRAVELEXPENSERESULT	boolean			

DisableUser: Visma Severa Disable User

Catalog: Severa

Schema: API

Label: Disable User

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

Filters on API Server: userGUID

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function DisableUser. 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
userGUID		□		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
DISABLEUSERRESULT	boolean			

EmploymentByGUID: Visma Severa Employment by GUID

Catalog: Severa

Schema: API

Label: Employment by GUID

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

Filters on API Server: employmentGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function EmploymentByGUID. 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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
employmentGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
COMPANYBUSINESSUNITCODE	varchar2	Company Business Unit Code		
COMPANYBUSINESSUNITGUID	varchar2	Company Business Unit GUID		
COMPANYBUSINESSUNITNAME	varchar2	Company Business Unit Name		
DAILYHOURS	number	Daily Hours		
ENDDATE	datetime	End Date		
GUID	varchar2	GUID		
HOURCOST	number	Hour Cost		
ISOVERTIMEALLOWED	boolean	Is Overtime Allowed		
STARTDATE	datetime	Start Date		
TITLE	varchar2	Title		

EmploymentsByUserGUID: Visma Severa Employments by User GUID

Catalog: Severa

Schema: API

Label: Employments by User GUID

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

Filters on API Server: userGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function EmploymentsByUserGUID. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
userGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
COMPANYBUSINESSUNITCODE	varchar2	Company Business Unit Code		
COMPANYBUSINESSUNITGUID	varchar2	Company Business Unit GUID		
COMPANYBUSINESSUNITNAME	varchar2	Company Business Unit Name		
DAILYHOURS	number	Daily Hours		
ENDDATE	datetime	End Date		
GUID	varchar2	GUID		
HOURCOST	number	Hour Cost		
ISOVERTIMEALLOWED	boolean	Is Overtime Allowed		
STARTDATE	datetime	Start Date		
TITLE	varchar2	Title		

EnableUser: Visma Severa Enable User

Catalog: Severa

Schema: API

Label: Enable User

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

Filters on API Server: userGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function EnableUser. 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-

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
userGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ENABLEUSERRESULT	boolean			

EntitySchemaVersion: Visma Severa Entity Schema Version

Catalog: Severa

Schema: API

Label: Entity Schema Version

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

Table Columns

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

Name	Data Type	Label	Required	Documentation
GETENTITYSCHEMAVERSIONRESULT	varchar2			

ExtranetCaseByCaseGUID: Visma Severa Extranet Project by Project GUID

Catalog: Severa

Schema: API

Label: Extranet Project by Project GUID

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Filters on API Server: caseGuid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ExtranetCaseByCaseGUID. 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
caseGuid		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CONTACTS	varchar2			
GUID	varchar2	GUID		
RIGHTS_EDITCASESTATUS	boolean			
RIGHTS_SHOWBILLING	boolean			
RIGHTS_SHOWCASESTATUS	boolean			
RIGHTS_SHOWFILESANDLINKS	boolean			
RIGHTS_SHOWGANTTCHART	boolean			
RIGHTS_SHOWPHASETREE	boolean			
RIGHTS_SHOWREADYTOBILL	boolean			
RIGHTS_SHOWWORKHOURESTIMATES	boolean			
RIGHTS_SHOWWORKHOURS	boolean			
RIGHTS_SHOWWORKHOURSTARTANDENDTIME	boolean			

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

ExtranetContactCaseRightsByContactGUID: Visma Severa Extranet Contact Project Rights by Contact GUID

Catalog: Severa

Schema: API

Label: Extranet Contact Project Rights by Contact GUID

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

Filters on API Server: contactGuid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ExtranetContactCaseRightsByContactGUID. 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
contactGuid		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CASEGUID	varchar2	Project GUID		
CASERIGHTS_EDITCASESTATUS	boolean			
CASERIGHTS_SHOWBILLING	boolean			
CASERIGHTS_SHOWCASESTATUS	boolean			
CASERIGHTS_SHOWFILESANDLINKS	boolean			
CASERIGHTS_SHOWGANTTCHART	boolean			
CASERIGHTS_SHOWPHASETREE	boolean			
CASERIGHTS_SHOWREADYTOBILL	boolean			

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
CASERIGHTS_SHOWWORKHOURESTIMATES	boolean			
CASERIGHTS_SHOWWORKHOURS	boolean			
CASERIGHTS_SHOWWORKHOURSTARTANDENDTIME	boolean			
CONTACTRIGHTS_APPROVAL	boolean			
CONTACTRIGHTS_COMMENTS	boolean			
CONTACTRIGHTS_FILEUPLOAD	boolean			

FileByGUID: Visma Severa File by GUID

Catalog: Severa

Schema: API

Label: File by GUID

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

Filters on API Server: guid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function FileByGUID. 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
guid		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CONTENTTYPE	varchar2	Content Type		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
DESCRIPTION	varchar2	Ddescription		
GUID	varchar2	GUID		
NAME	varchar2	Name		
SIZE	number	Size		

FileBytesByGUID: Visma Severa File Contents by GUID

Catalog: Severa

Schema: API

Label: File Contents by GUID

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

Filters on API Server: guid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function FileBytesByGUID. 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
guid		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
GETFILEBYTESBYGUIDRESULT	blob			

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Files: Visma Severa Files

Catalog: Severa

Schema: API

Label: Files

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

Table Columns

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

Name	Data Type	Label	Required	Documentation
CONTENTTYPE	varchar2	Content Type		
DESCRIPTION	varchar2	Ddescription		
GUID	varchar2	GUID		
NAME	varchar2	Name		
SIZE	number	Size		

FinvoicesByStatus: Visma Severa Financial Invoices by Status

Catalog: Severa

Schema: API

Label: Financial Invoices by Status

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

Filters on API Server: businessUnitGUID- status- options

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function FinvoicesByStatus. 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
businessUnitGUID		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
options		☐		
status		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
FINVOICE	varchar2	Financial Invoice		
GUID	varchar2	GUID		

FinvoicesByStatusAndDate: Visma Severa Financial Invoices by Status and Date

Catalog: Severa

Schema: API

Label: Financial Invoices by Status and Date

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

Filters on API Server: finvoiceVersion- businessUnitGUID- invoiceStatus- invoiceDateStart- invoiceDateEnd- options

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function FinvoicesByStatusAndDate. 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
businessUnitGUID		☐		
finvoiceVersion		☐		
invoiceDateEnd		☐		
invoiceDateStart		☐		
invoiceStatus		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
options		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
FINVOICE	varchar2	Financial Invoice		
GUID	varchar2	GUID		

FirstAccountsBillingAddress: Visma Severa First Account Billing Addresses

Catalog: Severa

Schema: API

Label: First Account Billing Addresses

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

Filters on API Server: accountGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function FirstAccountsBillingAddress. 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
accountGUID		☐		

Columns of Table Function

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
ADDRESSLINE	varchar2	Addressline		
ADDRESSLINE2	varchar2	Addressline 2		
ADDRESSLINE3	varchar2	Addressline 3		
CITY	varchar2	City		
COMPANYGUID	varchar2	Company GUID		
COUNTRYCODE	varchar2	Country Code		
COUNTRYGUID	varchar2	Country GUID		
COUNTRYNAME	varchar2	Country Name		
COUNTRYREGIONGUID	varchar2	Country Region GUID		
COUNTRYREGIONNAME	varchar2	Country Region Name		
FAX	varchar2	Fax		
GUID	varchar2	GUID		
ISBILLINGADDRESS	boolean	Is Billing Address		
ISPOSTALADDRESS	boolean	Is Postal Address		
ISVISITADDRESS	boolean	Is Visit Address		
PHONE	varchar2	Phone		
POSTALCODE	varchar2	Postal Code		

HeadOfficeAddressByAccountGUID: Visma Severa Head Office Address by Account GUID

Catalog: Severa

Schema: API

Label: Head Office Address by Account GUID

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

Filters on API Server: accountGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function HeadOfficeAddressByAccountGUID. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
accountGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ADDRESSLINE	varchar2	Addressline		
ADDRESSLINE2	varchar2	Addressline 2		
ADDRESSLINE3	varchar2	Addressline 3		
CITY	varchar2	City		
COMPANYGUID	varchar2	Company GUID		
COUNTRYCODE	varchar2	Country Code		
COUNTRYGUID	varchar2	Country GUID		
COUNTRYNAME	varchar2	Country Name		
COUNTRYREGIONGUID	varchar2	Country Region GUID		
COUNTRYREGIONNAME	varchar2	Country Region Name		
FAX	varchar2	Fax		
GUID	varchar2	GUID		
ISBILLINGADDRESS	boolean	Is Billing Address		
ISPOSTALADDRESS	boolean	Is Postal Address		
ISVISITADDRESS	boolean	Is Visit Address		
PHONE	varchar2	Phone		
POSTALCODE	varchar2	Postal Code		

HourEntriesByCaseGUID: Visma Severa Hour Entries by Project GUID

Catalog: Severa

Schema: API

Label: Hour Entries by Project GUID

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

Filters on API Server: caseGuid- startDate- endDate

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function HourEntriesByCaseGUID. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
caseGuid		☐		
endDate		☐		
startDate		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	varchar2	Account GUID		
CASEBUSINESSUNITCODE	varchar2	Project Business Unit Code		
CASECOSTCENTERIDENTIFIER	varchar2	Project Cost Centre Identifier		
CASEGUID	varchar2	Project GUID		
CASENAME	varchar2	Project Name		
CASENUMBER	number	Project Number		
COSTCURRENCYSHORTFORM	varchar2	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Cost Currency Short Form of Organisation Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Cost Currency Short Form of User Company Currency		
DESCRIPTION	varchar2	Ddescription		
ENDTIME	datetime	End Time		
EVENTDATE	datetime	Event Date		
GUID	varchar2	GUID		
INVOICEDESCRIPTION	varchar2	Invoice Description		
INVOICEGUID	varchar2	Invoice GUID		
INVOICEQUANTITY	number	Invoice Quantity		
INVOICEROWGUID	varchar2	Invoice Line GUID		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISPRODUCTIVE	boolean	Is Productive		
ISREADONLY	boolean	Is Read-only		
OVERTIMECODE	varchar2	Overtime Code		
OVERTIMEGUID	varchar2	Overtime GUID		
PHASEGUID	varchar2	Phase GUID		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	varchar2	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Price Currency Short Form of Organisation Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Price Currency Short Form of User Company Currency		
QUANTITY	number	Quantity		
STARTTIME	datetime	Start Time		
TASKGUID	varchar2	Task GUID		
UNITCOST	number	Unit Cost		
UNITCOSTININVOICINGCURRENCY	number	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	number	Unit Cost in Organisation Currency		
UNITCOSTINUSERCOMPANYCURRENCY	number	Unit Cost in User Company Currency		
UNITPRICE	number	Unit Price		
UNITPRICEINBASECURRENCY	number	Unit Price in Base Currency		
UNITPRICEINCASECOMPANYCURRENCY	number	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	number	Unit Price in User Company Currency		
USEINVOICEQUANTITY	boolean			
USERBUSINESSUNITCODE	varchar2	User Business Unit Code		
USERCODE	varchar2	User Code		
USERFIRSTNAME	varchar2	User First Name		
USERGUID	varchar2	User GUID		
USERLASTNAME	varchar2	User Last Name		
WORKTYPECODE	varchar2	Work Type Code		
WORKTYPEGUID	varchar2	Work Type GUID		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

HourEntriesByDate: Visma Severa Hour Entries by Date

Catalog: Severa

Schema: API

Label: Hour Entries by Date

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

Filters on API Server: businessUnitGUID- startdate- enddate

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function HourEntriesByDate. 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
businessUnitGUID		☐		
enddate		☐		
startdate		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	varchar2	Account GUID		
CASEBUSINESSUNITCODE	varchar2	Project Business Unit Code		
CASECOSTCENTERIDENTIFIER	varchar2	Project Cost Centre Identifier		
CASEGUID	varchar2	Project GUID		
CASENAME	varchar2	Project Name		
CASENUMBER	number	Project Number		
COSTCURRENCYSHORTFORM	varchar2	Cost Currency Short Form		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Cost Currency Short Form of Organisation Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Cost Currency Short Form of User Company Currency		
DESCRIPTION	varchar2	Ddescription		
ENDTIME	datetime	End Time		
EVENTDATE	datetime	Event Date		
GUID	varchar2	GUID		
INVOICEDESCRIPTION	varchar2	Invoice Description		
INVOICEGUID	varchar2	Invoice GUID		
INVOICEQUANTITY	number	Invoice Quantity		
INVOICEROWGUID	varchar2	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISPRODUCTIVE	boolean	Is Productive		
ISREADONLY	boolean	Is Read-only		
OVERTIMECODE	varchar2	Overtime Code		
OVERTIMEGUID	varchar2	Overtime GUID		
PHASEGUID	varchar2	Phase GUID		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	varchar2	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Price Currency Short Form of Organisation Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Price Currency Short Form of User Company Currency		
QUANTITY	number	Quantity		
STARTTIME	datetime	Start Time		
TASKGUID	varchar2	Task GUID		
UNITCOST	number	Unit Cost		
UNITCOSTININVOICINGCURRENCY	number	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	number	Unit Cost in Organisation Currency		
UNITCOSTINUSERCOMPANYCURRENCY	number	Unit Cost in User Company Currency		
UNITPRICE	number	Unit Price		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
UNITPRICEINBASECURRENCY	number	Unit Price in Base Currency		
UNITPRICEINCASECOMPANYCURRENCY	number	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	number	Unit Price in User Company Currency		
USEINVOICEQUANTITY	boolean			
USERBUSINESSUNITCODE	varchar2	User Business Unit Code		
USERCODE	varchar2	User Code		
USERFIRSTNAME	varchar2	User First Name		
USERGUID	varchar2	User GUID		
USERLASTNAME	varchar2	User Last Name		
WORKTYPECODE	varchar2	Work Type Code		
WORKTYPEGUID	varchar2	Work Type GUID		

HourEntriesByDateAndUser: Visma Severa Hour Entries by Date and User

Catalog: Severa

Schema: API

Label: Hour Entries by Date and User

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

Filters on API Server: businessUnitGUID- startdate- enddate- userGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function HourEntriesByDateAndUser. 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
businessUnitGUID		☐		
enddate		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
startdate		☐		
userGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	varchar2	Account GUID		
CASEBUSINESSUNITCODE	varchar2	Project Business Unit Code		
CASECOSTCENTERIDENTIFIER	varchar2	Project Cost Centre Identifier		
CASEGUID	varchar2	Project GUID		
CASENAME	varchar2	Project Name		
CASENUMBER	number	Project Number		
COSTCURRENCYSHORTFORM	varchar2	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Cost Currency Short Form of Organisation Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Cost Currency Short Form of User Company Currency		
DESCRIPTION	varchar2	Ddescription		
ENDTIME	datetime	End Time		
EVENTDATE	datetime	Event Date		
GUID	varchar2	GUID		
INVOICEDESCRIPTION	varchar2	Invoice Description		
INVOICEGUID	varchar2	Invoice GUID		
INVOICEQUANTITY	number	Invoice Quantity		
INVOICEROWGUID	varchar2	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISPRODUCTIVE	boolean	Is Productive		
ISREADONLY	boolean	Is Read-only		
OVERTIMECODE	varchar2	Overtime Code		
OVERTIMEGUID	varchar2	Overtime GUID		
PHASEGUID	varchar2	Phase GUID		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	varchar2	Price Currency Short Form of Project		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
		Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Price Currency Short Form of Organisation Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Price Currency Short Form of User Company Currency		
QUANTITY	number	Quantity		
STARTTIME	datetime	Start Time		
TASKGUID	varchar2	Task GUID		
UNITCOST	number	Unit Cost		
UNITCOSTININVOICINGCURRENCY	number	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	number	Unit Cost in Organisation Currency		
UNITCOSTINUSERCOMPANYCURRENCY	number	Unit Cost in User Company Currency		
UNITPRICE	number	Unit Price		
UNITPRICEINBASECURRENCY	number	Unit Price in Base Currency		
UNITPRICEINCASECOMPANYCURRENCY	number	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	number	Unit Price in User Company Currency		
USEINVOICEQUANTITY	boolean			
USERBUSINESSUNITCODE	varchar2	User Business Unit Code		
USERCODE	varchar2	User Code		
USERFIRSTNAME	varchar2	User First Name		
USERGUID	varchar2	User GUID		
USERLASTNAME	varchar2	User Last Name		
WORKTYPECODE	varchar2	Work Type Code		
WORKTYPEGUID	varchar2	Work Type GUID		

HourEntriesByDateAndUserGUID: Visma Severa Hour Entry by Date and User GUID

Catalog: Severa

Schema: API

Label: Hour Entry by Date and User GUID

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Filters on API Server: businessUnitGUID- startdate- enddate- userGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function HourEntriesByDateAndUserGUID. 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
businessUnitGUID		☐		
enddate		☐		
startdate		☐		
userGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	varchar2	Account GUID		
CASEBUSINESSUNITCODE	varchar2	Project Business Unit Code		
CASECOSTCENTERIDENTIFIER	varchar2	Project Cost Centre Identifier		
CASEGUID	varchar2	Project GUID		
CASENAME	varchar2	Project Name		
CASENUMBER	number	Project Number		
COSTCURRENCYSHORTFORM	varchar2	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Cost Currency Short Form of Organisation Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Cost Currency Short Form of User Company Currency		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
DESCRIPTION	varchar2	Ddescription		
ENDTIME	datetime	End Time		
EVENTDATE	datetime	Event Date		
GUID	varchar2	GUID		
INVOICEDESCRIPTION	varchar2	Invoice Description		
INVOICEGUID	varchar2	Invoice GUID		
INVOICEQUANTITY	number	Invoice Quantity		
INVOICEROWGUID	varchar2	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISPRODUCTIVE	boolean	Is Productive		
ISREADONLY	boolean	Is Read-only		
OVERTIMECODE	varchar2	Overtime Code		
OVERTIMEGUID	varchar2	Overtime GUID		
PHASEGUID	varchar2	Phase GUID		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	varchar2	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Price Currency Short Form of Organisation Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Price Currency Short Form of User Company Currency		
QUANTITY	number	Quantity		
STARTTIME	datetime	Start Time		
TASKGUID	varchar2	Task GUID		
UNITCOST	number	Unit Cost		
UNITCOSTININVOICINGCURRENCY	number	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	number	Unit Cost in Organisation Currency		
UNITCOSTINUSERCOMPANYCURRENCY	number	Unit Cost in User Company Currency		
UNITPRICE	number	Unit Price		
UNITPRICEINBASECURRENCY	number	Unit Price in Base Currency		
UNITPRICEINCASECOMPANYCURRENCY	number	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	number	Unit Price in User Company Currency		
USEINVOICEQUANTITY	boolean			

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
USERBUSINESSUNITCODE	varchar2	User Business Unit Code		
USERCODE	varchar2	User Code		
USERFIRSTNAME	varchar2	User First Name		
USERGUID	varchar2	User GUID		
USERLASTNAME	varchar2	User Last Name		
WORKTYPECODE	varchar2	Work Type Code		
WORKTYPEGUID	varchar2	Work Type GUID		

HourEntriesChangedSince: Visma Severa Hour Entries Changed since

Catalog: Severa

Schema: API

Label: Hour Entries Changed since

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

Filters on API Server: startDate

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function HourEntriesChangedSince. 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
startDate		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	varchar2	Account GUID		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
CASEBUSINESSUNITCODE	varchar2	Project Business Unit Code		
CASECOSTCENTERIDENTIFIER	varchar2	Project Cost Centre Identifier		
CASEGUID	varchar2	Project GUID		
CASENAME	varchar2	Project Name		
CASENUMBER	number	Project Number		
COSTCURRENCYSHORTFORM	varchar2	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Cost Currency Short Form of Organisation Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Cost Currency Short Form of User Company Currency		
DESCRIPTION	varchar2	Ddescription		
ENDTIME	datetime	End Time		
EVENTDATE	datetime	Event Date		
GUID	varchar2	GUID		
INVOICEDESCRIPTION	varchar2	Invoice Description		
INVOICEGUID	varchar2	Invoice GUID		
INVOICEQUANTITY	number	Invoice Quantity		
INVOICEROWGUID	varchar2	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISPRODUCTIVE	boolean	Is Productive		
ISREADONLY	boolean	Is Read-only		
OVERTIMECODE	varchar2	Overtime Code		
OVERTIMEGUID	varchar2	Overtime GUID		
PHASEGUID	varchar2	Phase GUID		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	varchar2	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Price Currency Short Form of Organisation Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Price Currency Short Form of User Company Currency		
QUANTITY	number	Quantity		
STARTTIME	datetime	Start Time		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
TASKGUID	varchar2	Task GUID		
UNITCOST	number	Unit Cost		
UNITCOSTININVOICINGCURRENCY	number	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	number	Unit Cost in Organisation Currency		
UNITCOSTINUSERCOMPANYCURRENCY	number	Unit Cost in User Company Currency		
UNITPRICE	number	Unit Price		
UNITPRICEINBASECURRENCY	number	Unit Price in Base Currency		
UNITPRICEINCASECOMPANYCURRENCY	number	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	number	Unit Price in User Company Currency		
USEINVOICEQUANTITY	boolean			
USERBUSINESSUNITCODE	varchar2	User Business Unit Code		
USERCODE	varchar2	User Code		
USERFIRSTNAME	varchar2	User First Name		
USERGUID	varchar2	User GUID		
USERLASTNAME	varchar2	User Last Name		
WORKTYPECODE	varchar2	Work Type Code		
WORKTYPEGUID	varchar2	Work Type GUID		

HourEntryByGUID: Visma Severa Hour Entry by GUID

Catalog: Severa

Schema: API

Label: Hour Entry by GUID

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

Filters on API Server: hourEntryGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function HourEntryByGUID. 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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
hourEntryGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	varchar2	Account GUID		
CASEBUSINESSUNITCODE	varchar2	Project Business Unit Code		
CASECOSTCENTERIDENTIFIER	varchar2	Project Cost Centre Identifier		
CASEGUID	varchar2	Project GUID		
CASENAME	varchar2	Project Name		
CASENUMBER	number	Project Number		
COSTCURRENCYSHORTFORM	varchar2	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Cost Currency Short Form of Organisation Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Cost Currency Short Form of User Company Currency		
DESCRIPTION	varchar2	Ddescription		
ENDTIME	datetime	End Time		
EVENTDATE	datetime	Event Date		
GUID	varchar2	GUID		
INVOICEDESCRIPTION	varchar2	Invoice Description		
INVOICEGUID	varchar2	Invoice GUID		
INVOICEQUANTITY	number	Invoice Quantity		
INVOICEROWGUID	varchar2	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISPRODUCTIVE	boolean	Is Productive		
ISREADONLY	boolean	Is Read-only		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
OVERTIMECODE	varchar2	Overtime Code		
OVERTIMEGUID	varchar2	Overtime GUID		
PHASEGUID	varchar2	Phase GUID		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	varchar2	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Price Currency Short Form of Organisation Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Price Currency Short Form of User Company Currency		
QUANTITY	number	Quantity		
STARTTIME	datetime	Start Time		
TASKGUID	varchar2	Task GUID		
UNITCOST	number	Unit Cost		
UNITCOSTININVOICINGCURRENCY	number	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	number	Unit Cost in Organisation Currency		
UNITCOSTINUSERCOMPANYCURRENCY	number	Unit Cost in User Company Currency		
UNITPRICE	number	Unit Price		
UNITPRICEINBASECURRENCY	number	Unit Price in Base Currency		
UNITPRICEINCASECOMPANYCURRENCY	number	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	number	Unit Price in User Company Currency		
USEINVOICEQUANTITY	boolean			
USERBUSINESSUNITCODE	varchar2	User Business Unit Code		
USERCODE	varchar2	User Code		
USERFIRSTNAME	varchar2	User First Name		
USERGUID	varchar2	User GUID		
USERLASTNAME	varchar2	User Last Name		
WORKTYPECODE	varchar2	Work Type Code		
WORKTYPEGUID	varchar2	Work Type GUID		

HourEntryPhasesByUserGUID: Visma Severa Hour Entry Phases by User GUID

Catalog: Severa

Schema: API

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Label: Hour Entry Phases by User GUID

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

Filters on API Server: userGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function HourEntryPhasesByUserGUID. 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
userGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	varchar2	Account GUID		
ACCOUNTNAME	varchar2	Account Name		
CASEGUID	varchar2	Project GUID		
CHILDPHASES	varchar2	Child Phases		
GUID	varchar2	GUID		
ISLOCKED	boolean	Is Locked		
NAME	varchar2	Name		
PARENTPHASEGUID	varchar2	Parent Phase GUID		
WORKTYPEGUID	varchar2	Work Type GUID		

Industries: Visma Severa Industries

Catalog: Severa

Schema: API

Label: Industries

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Table Columns

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

Name	Data Type	Label	Required	Documentation
CODE	varchar2	Code		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
NAME	varchar2	Name		

InvoiceAttachmentFile: Visma Severa Invoice Attachment File

Catalog: Severa

Schema: API

Label: Invoice Attachment File

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

Filters on API Server: attachmentGuid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function InvoiceAttachmentFile. 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
attachmentGuid		☐		

Columns of Table Function

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
GETINVOICEATTACHMENTFILERESULT	blob			

InvoiceAttachmentFileGUIDs: Visma Severa Invoice Attachment File GUIDs

Catalog: Severa

Schema: API

Label: Invoice Attachment File GUIDs

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

Filters on API Server: invoiceGuid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function InvoiceAttachmentFileGUIDs. 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
invoiceGuid		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
GUID	varchar2	GUID		
MIMETYPE	varchar2	MIME-type		
NAME	varchar2	Name		
PRINTORDER	number	Print Order		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

InvoiceByGUID: Visma Severa Invoice by GUID

Catalog: Severa

Schema: API

Label: Invoice by GUID

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

Filters on API Server: invoiceGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function InvoiceByGUID. 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
invoiceGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	varchar2	Account GUID		
ACCOUNTNAME	varchar2	Account Name		
ACCOUNTNUMBER	number	Account Number		
BUSINESSUNITCODE	varchar2	Business Unit Code		
BUSINESSUNITGUID	varchar2	Business Unit GUID		
BUSINESSUNITNAME	varchar2	Business Unit Name		
BUYERORGANIZATION_BUYERCONTACTPERSONNAME	varchar2			
BUYERORGANIZATION_BUYERELECTRONICINVOICEADDRESS	varchar2			
BUYERORGANIZATION_BUYERELECTRONICINVOICEOPERATOR	varchar2			

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
BUYERORGANIZATION_BUYEREMAILADDRESSIDENTIFIER	varchar2			
BUYERORGANIZATION_BUYERORGANISATIONNAME	varchar2			
BUYERORGANIZATION_BUYERORGANISATIONTAXCODE	varchar2			
BUYERORGANIZATION_BUYERPARTYIDENTIFIER	varchar2			
BUYERORGANIZATION_BUYERPHONENUMBERIDENTIFIER	varchar2			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRY	varchar2			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRYCODE	varchar2			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	varchar2			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_STREETNAME	varchar2			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_TOWNNAME	varchar2			
CULTURE	varchar2	Culture		
CURRENCYGUID	varchar2	Currency GUID		
CURRENCYNAME	varchar2	Currency Name		
CURRENCYRATE	number	Currency Rate		
CURRENCYSHORTFORM	varchar2	Currency Short Form		
DATE	datetime	Date		
DESCRIPTION	varchar2	Ddescription		
DUEDATE	datetime	Due Date		
ENTRYDATE	datetime	Entry Date		
GUID	varchar2	GUID		
INVOICEALLOWEDCASEGUIDS	varchar2			
INVOICECONFIG_BILLEXPENSE	varchar2			
INVOICECONFIG_BILLHOUR	varchar2			
INVOICECONFIG_BILLPRODUCT	varchar2			
INVOICECONFIG_EXPENSEITEMDESCRIPTIONFORMAT	varchar2			
INVOICECONFIG_HOURITEMDESCRIPTIONFORMAT	varchar2			
INVOICECONFIG_HOURPRIMARYGROUPBY	varchar2			
INVOICECONFIG_HOURSECONDARYGROUPBY	varchar2			
INVOICECONFIG_ISTAXFREE	boolean			

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
INVOICECONFIG_PRODUCTITEMDESCRIPTIONFORMAT	varchar2			
INVOICECONFIG_PRODUCTPRIMARYGROUPBY	varchar2			
INVOICECONFIG_PRODUCTSECONDARYGROUPBY	varchar2			
INVOICECONFIG_SHOWATTACHPRICEEXCLUDINGVAT	boolean			
INVOICECONFIG_SHOWATTACHUNIT	boolean			
INVOICECONFIG_SHOWATTACHUNITPRICE	boolean			
INVOICECONFIG_SHOWATTACHVAT	boolean			
INVOICECONFIG_SHOWLOGOONINVOICE	boolean			
INVOICECONFIG_SHOWPRICEEXCLUDINGVAT	boolean			
INVOICECONFIG_SHOWQUANTITY	boolean			
INVOICECONFIG_SHOWUNIT	boolean			
INVOICECONFIG_SHOWUNITPRICE	boolean			
INVOICECONFIG_SHOWVAT	boolean			
INVOICECONFIG_TAXFREEDESCRIPTION	varchar2			
INVOICECONFIG_TRAVELPRIMARYGROUPBY	varchar2			
INVOICECONFIG_TRAVELSECONDARYGROUPBY	varchar2			
INVOICEDETAILS_INVOICEDATE	datetime			
INVOICEDETAILS_INVOICEDUEDATE	datetime			
INVOICEDETAILS_INVOICENUMBER	number			
INVOICEDETAILS_INVOICETOTALVATAMOUNT	number			
INVOICEDETAILS_INVOICETOTALVATEXCLUDEDAMOUNT	number			
INVOICEDETAILS_INVOICETOTALVATINCLUDEDAMOUNT	number			
INVOICEDETAILS_INVOICETYPETEXT	varchar2			
INVOICEDETAILS_ORDERIDENTIFIER	varchar2			
INVOICEDETAILS_ORIGINCODE	varchar2			
INVOICEDETAILS_PAYMENTOVERDUEFINEFREETEXT	varchar2			
INVOICEDETAILS_PAYMENTOVERDUEFINEPERCENT	number			

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
INVOICEDETAILS_PAYMENTTERMSFREETEXT	varchar2			
INVOICEDETAILS_SELLERREFERENCEIDENTIFIER	varchar2			
INVOICEDETAILS_VATBASEAMOUNT	number			
INVOICEDETAILS_VATRATEAMOUNT	number			
INVOICEDETAILS_VATRATEPERCENT	number			
INVOICENUMBERINGBUSINESSUNITGUID	varchar2	Invoice Numbering Business Unit GUID		
INVOICEROWS	varchar2			
INVOICESTATUS	varchar2	Invoice Status		
ISCREDITNOTE	boolean	Is Credit Note		
ISREIMBURSED	boolean	Is Reimbursed		
ISTAXFREE	boolean	Is Tax Free		
LANGUAGECODE	varchar2	Language Code		
LANGUAGEGUID	varchar2	Language GUID		
MAINCASEGUID	varchar2	Main Case GUID		
NOTES	varchar2	Notes		
NOTESAFTERINVOICEROWS	varchar2	Notes after Invoice Lines		
NOTESBEFOREINVOICEROWS	varchar2	Notes before Invoice Lines		
NUMBER	number	Number		
ORDERNUMBER	varchar2	Order Number		
OURREFERENCE	varchar2	Our Reference		
OVERDUEINTEREST	number	Overdue Interest		
PAYMENTDATE	datetime	Payment Date		
PAYMENTSTATUS	varchar2	Payment Status		
PAYMENTTERM	number	Payment Term		
PONUMBER	varchar2	PO Number		
RECEIVERADDRESSLINE	varchar2	Receiver Addressline		
RECEIVERCITY	varchar2	Receiver City		
RECEIVERCONTACTEMAIL	varchar2	Receiver Contact Email		
RECEIVERCONTACTNAME	varchar2	Receiver Contact Name		
RECEIVERCOUNTRY	varchar2	Receiver Country		
RECEIVERCUSTOMERNAME	varchar2	Receiver Customer Name		
RECEIVERPOSTALCODE	varchar2	Receiver Postal Code		
RECEIVERSTATE	varchar2	Receiver State		
RECEIVERVATNUMBER	varchar2	Receiver VAT Number		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
REFERENCENUMBER	varchar2	Reference Number		
REIMBURSEINVOICEGUID	varchar2	Reimburse Invoice GUID		
SELLERCONTACT_SELLERACCOUNTDETAILS	varchar2			
SELLERCONTACT_SELLERCONTACTPERSONNAME	varchar2			
SELLERCONTACT_SELLEREMAILADDRESSIDENTIFIER	varchar2			
SELLERCONTACT_SELLERFAXNUMBER	varchar2			
SELLERCONTACT_SELLERFREETEXT	varchar2			
SELLERCONTACT_SELLERHOMETOWNNAME	varchar2			
SELLERCONTACT_SELLERPHONENUMBER	varchar2			
SELLERCONTACT_SELLERWEBADDRESSIDENTIFIER	varchar2			
SELLERORGANIZATION_SELLEREINVOICEADDRESS	varchar2			
SELLERORGANIZATION_SELLEREINVOICEOPERATOR	varchar2			
SELLERORGANIZATION_SELLERORGANISATIONNAME	varchar2			
SELLERORGANIZATION_SELLERORGANISATIONTAXCODE	varchar2			
SELLERORGANIZATION_SELLERPARTYIDENTIFIER	varchar2			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRY	varchar2			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRYCODE	varchar2			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	varchar2			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_STREETNAME	varchar2			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_TOWNNAME	varchar2			
SENDERADDRESSLINE	varchar2	Sender Addressline		
SENDERCITY	varchar2	Sender City		
SENDERCONTACTNAME	varchar2	Sender Contact Name		
SENDERCOUNTRY	varchar2	Sender Country		
SENDERNAME	varchar2	Sender Name		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
SENDERPOSTALCODE	varchar2	Sender Postal Code		
SENDERSTATE	varchar2	Sender State		
SENDERVATNUMBER	varchar2	Sender VAT Number		
TAXFREEDESCRIPTION	varchar2	Tax Free Description		
TITLE	varchar2	Title		
VIRTUALBANKBARCODE	varchar2	Virtual Bank Barcode		
YOURREFERENCE	varchar2	Your Reference		

InvoiceByNumber: Visma Severa Invoice by Number

Catalog: Severa

Schema: API

Label: Invoice by Number

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

Filters on API Server: invoiceNumber- businessUnitGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function InvoiceByNumber. 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
businessUnitGUID		☐		
invoiceNumber		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	varchar2	Account GUID		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
ACCOUNTNAME	varchar2	Account Name		
ACCOUNTNUMBER	number	Account Number		
BUSINESSUNITCODE	varchar2	Business Unit Code		
BUSINESSUNITGUID	varchar2	Business Unit GUID		
BUSINESSUNITNAME	varchar2	Business Unit Name		
BUYERORGANIZATION_BUYERCONTACTPERSONNAME	varchar2			
BUYERORGANIZATION_BUYERELECTRONICINVOICEADDRESS	varchar2			
BUYERORGANIZATION_BUYERELECTRONICINVOICEOPERATOR	varchar2			
BUYERORGANIZATION_BUYEREMAILADDRESSIDENTIFIER	varchar2			
BUYERORGANIZATION_BUYERORGANISATIONNAME	varchar2			
BUYERORGANIZATION_BUYERORGANISATIONTAXCODE	varchar2			
BUYERORGANIZATION_BUYERPARTYIDENTIFIER	varchar2			
BUYERORGANIZATION_BUYERPHONENUMBERIDENTIFIER	varchar2			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRY	varchar2			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRYCODE	varchar2			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	varchar2			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_STREETNAME	varchar2			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_TOWNNAME	varchar2			
CULTURE	varchar2	Culture		
CURRENCYGUID	varchar2	Currency GUID		
CURRENCYNAME	varchar2	Currency Name		
CURRENCYRATE	number	Currency Rate		
CURRENCYSHORTFORM	varchar2	Currency Short Form		
DATE	datetime	Date		
DESCRIPTION	varchar2	Ddescription		
DUEDATE	datetime	Due Date		
ENTRYDATE	datetime	Entry Date		
GUID	varchar2	GUID		
INVOICEALLOWEDCASEGUIDS	varchar2			

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
INVOICECONFIG_BILLEXPENSE	varchar2			
INVOICECONFIG_BILLHOUR	varchar2			
INVOICECONFIG_BILLPRODUCT	varchar2			
INVOICECONFIG_EXPENSEITEMDESCRIPTIONFORMAT	varchar2			
INVOICECONFIG_HOURITEMDESCRIPTIONFORMAT	varchar2			
INVOICECONFIG_HOURPRIMARYGROUPBY	varchar2			
INVOICECONFIG_HOURSECONDARYGROUPBY	varchar2			
INVOICECONFIG_ISTAXFREE	boolean			
INVOICECONFIG_PRODUCTITEMDESCRIPTIONFORMAT	varchar2			
INVOICECONFIG_PRODUCTPRIMARYGROUPBY	varchar2			
INVOICECONFIG_PRODUCTSECONDARYGROUPBY	varchar2			
INVOICECONFIG_SHOWATTACHPRICEEXCLUDINGVAT	boolean			
INVOICECONFIG_SHOWATTACHUNIT	boolean			
INVOICECONFIG_SHOWATTACHUNITPRICE	boolean			
INVOICECONFIG_SHOWATTACHVAT	boolean			
INVOICECONFIG_SHOWLOGOONINVOICE	boolean			
INVOICECONFIG_SHOWPRICEEXCLUDINGVAT	boolean			
INVOICECONFIG_SHOWQUANTITY	boolean			
INVOICECONFIG_SHOWUNIT	boolean			
INVOICECONFIG_SHOWUNITPRICE	boolean			
INVOICECONFIG_SHOWVAT	boolean			
INVOICECONFIG_TAXFREEDESCRIPTION	varchar2			
INVOICECONFIG_TRAVELPRIMARYGROUPBY	varchar2			
INVOICECONFIG_TRAVELSECONDARYGROUPBY	varchar2			
INVOICEDETAILS_INVOICEDATE	datetime			
INVOICEDETAILS_INVOICEDUEDATE	datetime			
INVOICEDETAILS_INVOICENUMBER	number			
INVOICEDETAILS_INVOICETOTALVATAMOUNT	number			
INVOICEDETAILS_INVOICETOTALVATEXCLUDEDAMOUNT	number			

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
INVOICEDETAILS_INVOICETOTALVATINCLUDEDAMOUNT	number			
INVOICEDETAILS_INVOICETYPETEXT	varchar2			
INVOICEDETAILS_ORDERIDENTIFIER	varchar2			
INVOICEDETAILS_ORIGINCODE	varchar2			
INVOICEDETAILS_PAYMENTOVERDUEFINEFREETEXT	varchar2			
INVOICEDETAILS_PAYMENTOVERDUEFINEPERCENT	number			
INVOICEDETAILS_PAYMENTTERMSFREETEXT	varchar2			
INVOICEDETAILS_SELLERREFERENCEIDENTIFIER	varchar2			
INVOICEDETAILS_VATBASEAMOUNT	number			
INVOICEDETAILS_VATRATEAMOUNT	number			
INVOICEDETAILS_VATRATEPERCENT	number			
INVOICENUMBERINGBUSINESSUNITGUID	varchar2	Invoice Numbering Business Unit GUID		
INVOICEROWS	varchar2			
INVOICESTATUS	varchar2	Invoice Status		
ISCREDITNOTE	boolean	Is Credit Note		
ISREIMBURSED	boolean	Is Reimbursed		
ISTAXFREE	boolean	Is Tax Free		
LANGUAGECODE	varchar2	Language Code		
LANGUAGEGUID	varchar2	Language GUID		
MAINCASEGUID	varchar2	Main Case GUID		
NOTES	varchar2	Notes		
NOTESAFTERINVOICEROWS	varchar2	Notes after Invoice Lines		
NOTESBEFOREINVOICEROWS	varchar2	Notes before Invoice Lines		
NUMBER	number	Number		
ORDERNUMBER	varchar2	Order Number		
OURREFERENCE	varchar2	Our Reference		
OVERDUEINTEREST	number	Overdue Interest		
PAYMENTDATE	datetime	Payment Date		
PAYMENTSTATUS	varchar2	Payment Status		
PAYMENTTERM	number	Payment Term		
PONUMBER	varchar2	PO Number		
RECEIVERADDRESSLINE	varchar2	Receiver Addressline		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
RECEIVERCITY	varchar2	Receiver City		
RECEIVERCONTACTEMAIL	varchar2	Receiver Contact Email		
RECEIVERCONTACTNAME	varchar2	Receiver Contact Name		
RECEIVERCOUNTRY	varchar2	Receiver Country		
RECEIVERCUSTOMERNAME	varchar2	Receiver Customer Name		
RECEIVERPOSTALCODE	varchar2	Receiver Postal Code		
RECEIVERSTATE	varchar2	Receiver State		
RECEIVERVATNUMBER	varchar2	Receiver VAT Number		
REFERENCENUMBER	varchar2	Reference Number		
REIMBURSEINVOICEGUID	varchar2	Reimburse Invoice GUID		
SELLERCONTACT_SELLERACCOUNTDETAILS	varchar2			
SELLERCONTACT_SELLERCONTACTPERSONNAME	varchar2			
SELLERCONTACT_SELLEREMAILADDRESSIDENTIFIER	varchar2			
SELLERCONTACT_SELLERFAXNUMBER	varchar2			
SELLERCONTACT_SELLERFREETEXT	varchar2			
SELLERCONTACT_SELLERHOMETOWNNAME	varchar2			
SELLERCONTACT_SELLERPHONENUMBER	varchar2			
SELLERCONTACT_SELLERWEBADDRESSIDENTIFIER	varchar2			
SELLERORGANIZATION_SELLEREINVOICEADDRESS	varchar2			
SELLERORGANIZATION_SELLEREINVOICEOPERATOR	varchar2			
SELLERORGANIZATION_SELLERORGANISATIONNAME	varchar2			
SELLERORGANIZATION_SELLERORGANISATIONTAXCODE	varchar2			
SELLERORGANIZATION_SELLERPARTYIDENTIFIER	varchar2			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRY	varchar2			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRYCODE	varchar2			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	varchar2			

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_STREETNAME	varchar2			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_TOWNNAME	varchar2			
SENDERADDRESSLINE	varchar2	Sender Addressline		
SENDERCITY	varchar2	Sender City		
SENDERCONTACTNAME	varchar2	Sender Contact Name		
SENDERCOUNTRY	varchar2	Sender Country		
SENDERNAME	varchar2	Sender Name		
SENDERPOSTALCODE	varchar2	Sender Postal Code		
SENDERSTATE	varchar2	Sender State		
SENDERVATNUMBER	varchar2	Sender VAT Number		
TAXFREEDESCRIPTION	varchar2	Tax Free Description		
TITLE	varchar2	Title		
VIRTUALBANKBARCODE	varchar2	Virtual Bank Barcode		
YOURREFERENCE	varchar2	Your Reference		

InvoicedHourEntriesByCaseGUID: Visma Severa Invoiced Hour Entries by Project GUID

Catalog: Severa

Schema: API

Label: Invoiced Hour Entries by Project GUID

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

Filters on API Server: caseGuid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function InvoicedHourEntriesByCaseGUID. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
caseGuid		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	varchar2	Account GUID		
CASEBUSINESSUNITCODE	varchar2	Project Business Unit Code		
CASECOSTCENTERIDENTIFIER	varchar2	Project Cost Centre Identifier		
CASEGUID	varchar2	Project GUID		
CASENAME	varchar2	Project Name		
CASENUMBER	number	Project Number		
COSTCURRENCYSHORTFORM	varchar2	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Cost Currency Short Form of Organisation Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Cost Currency Short Form of User Company Currency		
DESCRIPTION	varchar2	Ddescription		
ENDTIME	datetime	End Time		
EVENTDATE	datetime	Event Date		
GUID	varchar2	GUID		
INVOICEDESCRIPTION	varchar2	Invoice Description		
INVOICEGUID	varchar2	Invoice GUID		
INVOICEQUANTITY	number	Invoice Quantity		
INVOICEROWGUID	varchar2	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISPRODUCTIVE	boolean	Is Productive		
ISREADONLY	boolean	Is Read-only		
OVERTIMECODE	varchar2	Overtime Code		
OVERTIMEGUID	varchar2	Overtime GUID		
PHASEGUID	varchar2	Phase GUID		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	varchar2	Price Currency Short Form of Project Company Currency		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Price Currency Short Form of Organisation Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Price Currency Short Form of User Company Currency		
QUANTITY	number	Quantity		
STARTTIME	datetime	Start Time		
TASKGUID	varchar2	Task GUID		
UNITCOST	number	Unit Cost		
UNITCOSTININVOICINGCURRENCY	number	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	number	Unit Cost in Organisation Currency		
UNITCOSTINUSERCOMPANYCURRENCY	number	Unit Cost in User Company Currency		
UNITPRICE	number	Unit Price		
UNITPRICEINBASECURRENCY	number	Unit Price in Base Currency		
UNITPRICEINCASECOMPANYCURRENCY	number	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	number	Unit Price in User Company Currency		
USEINVOICEQUANTITY	boolean			
USERBUSINESSUNITCODE	varchar2	User Business Unit Code		
USERCODE	varchar2	User Code		
USERFIRSTNAME	varchar2	User First Name		
USERGUID	varchar2	User GUID		
USERLASTNAME	varchar2	User Last Name		
WORKTYPECODE	varchar2	Work Type Code		
WORKTYPEGUID	varchar2	Work Type GUID		

InvoiceOriginDetailsByInvoiceGUID: Visma Severa Invoice Origin Details by Invoice GUID

Catalog: Severa

Schema: API

Label: Invoice Origin Details by Invoice GUID

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

Filters on API Server: invoiceGUID

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function InvoiceOriginDetailsByInvoiceGUID. 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
invoiceGUID		□		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CASES	varchar2	Projects		
INVOICEROWORIGINS	varchar2			

InvoicePdf: Visma Severa Invoice PDF

Catalog: Severa

Schema: API

Label: Invoice PDF

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

Filters on API Server: invoiceNumber- invoiceNumberingBusinessUnitGUID- getAttachmentOnly

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function InvoicePdf. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
getAttachmentOnly		☐		
invoiceNumber		☐		
invoiceNumberingBusinessUnitGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
GETINVOICEPDFRESULT	blob			

InvoicePdfByInvoiceGUID: Visma Severa Invoice PDF by Invoice GUID

Catalog: Severa

Schema: API

Label: Invoice PDF by Invoice GUID

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

Filters on API Server: invoiceGuid- getAttachmentOnly

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function InvoicePdfByInvoiceGUID. 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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
getAttachmentOnly		☐		
invoiceGuid		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
GETINVOICEPDFBYINVOICEGUIDRESULT	blob			

InvoicesByDate: Visma Severa Invoices by Date

Catalog: Severa

Schema: API

Label: Invoices by Date

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

Filters on API Server: businessUnitGUID- startDate- endDate- options

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function InvoicesByDate. 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
businessUnitGUID		☐		
endDate		☐		
options		☐		
startDate		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	varchar2	Account GUID		
ACCOUNTNAME	varchar2	Account Name		
ACCOUNTNUMBER	number	Account Number		
BUSINESSUNITCODE	varchar2	Business Unit Code		
BUSINESSUNITGUID	varchar2	Business Unit GUID		
BUSINESSUNITNAME	varchar2	Business Unit Name		
BUYERORGANIZATION_BUYERCONTACTPERSONNAME	varchar2			
BUYERORGANIZATION_BUYERELECTRONICINVOICEADDRESS	varchar2			
BUYERORGANIZATION_BUYERELECTRONICINVOICEOPERATOR	varchar2			
BUYERORGANIZATION_BUYEREMAILADDRESSIDENTIFIER	varchar2			
BUYERORGANIZATION_BUYERORGANISATIONNAME	varchar2			
BUYERORGANIZATION_BUYERORGANISATIONTAXCODE	varchar2			
BUYERORGANIZATION_BUYERPARTYIDENTIFIER	varchar2			
BUYERORGANIZATION_BUYERPHONENUMBERIDENTIFIER	varchar2			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRY	varchar2			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRYCODE	varchar2			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	varchar2			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_STREETNAME	varchar2			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_TOWNNAME	varchar2			
CULTURE	varchar2	Culture		
CURRENCYGUID	varchar2	Currency GUID		
CURRENCYNAME	varchar2	Currency Name		
CURRENCYRATE	number	Currency Rate		
CURRENCYSHORTFORM	varchar2	Currency Short Form		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
DATE	datetime	Date		
DESCRIPTION	varchar2	Ddescription		
DUEDATE	datetime	Due Date		
ENTRYDATE	datetime	Entry Date		
GUID	varchar2	GUID		
INVOICEALLOWEDCASEGUIDS	varchar2			
INVOICECONFIG_BILLEXPENSE	varchar2			
INVOICECONFIG_BILLHOUR	varchar2			
INVOICECONFIG_BILLPRODUCT	varchar2			
INVOICECONFIG_EXPENSEITEMDESCRIPTIONFORMAT	varchar2			
INVOICECONFIG_HOURITEMDESCRIPTIONFORMAT	varchar2			
INVOICECONFIG_HOURPRIMARYGROUPBY	varchar2			
INVOICECONFIG_HOURSECONDARYGROUPBY	varchar2			
INVOICECONFIG_ISTAXFREE	boolean			
INVOICECONFIG_PRODUCTITEMDESCRIPTIONFORMAT	varchar2			
INVOICECONFIG_PRODUCTPRIMARYGROUPBY	varchar2			
INVOICECONFIG_PRODUCTSECONDARYGROUPBY	varchar2			
INVOICECONFIG_SHOWATTACHPRICEEXCLUDINGVAT	boolean			
INVOICECONFIG_SHOWATTACHUNIT	boolean			
INVOICECONFIG_SHOWATTACHUNITPRICE	boolean			
INVOICECONFIG_SHOWATTACHVAT	boolean			
INVOICECONFIG_SHOWLOGOONINVOICE	boolean			
INVOICECONFIG_SHOWPRICEEXCLUDINGVAT	boolean			
INVOICECONFIG_SHOWQUANTITY	boolean			
INVOICECONFIG_SHOWUNIT	boolean			
INVOICECONFIG_SHOWUNITPRICE	boolean			
INVOICECONFIG_SHOWVAT	boolean			
INVOICECONFIG_TAXFREEDESCRIPTION	varchar2			
INVOICECONFIG_TRAVELPRIMARYGROUPBY	varchar2			
INVOICECONFIG_TRAVELSECONDARYGROUPBY	varchar2			
INVOICEDETAILS_INVOICEDATE	datetime			

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
INVOICEDETAILS_INVOICEDUEDATE	datetime			
INVOICEDETAILS_INVOICENUMBER	number			
INVOICEDETAILS_INVOICETOTALVATAMOUNT	number			
INVOICEDETAILS_INVOICETOTALVATEXCLUDEDAMOUNT	number			
INVOICEDETAILS_INVOICETOTALVATINCLUDEDAMOUNT	number			
INVOICEDETAILS_INVOICETYPETEXT	varchar2			
INVOICEDETAILS_ORDERIDENTIFIER	varchar2			
INVOICEDETAILS_ORIGINCODE	varchar2			
INVOICEDETAILS_PAYMENTOVERDUEFINEFREETEXT	varchar2			
INVOICEDETAILS_PAYMENTOVERDUEFINEPERCENT	number			
INVOICEDETAILS_PAYMENTTERMSFREETEXT	varchar2			
INVOICEDETAILS_SELLERREFERENCEIDENTIFIER	varchar2			
INVOICEDETAILS_VATBASEAMOUNT	number			
INVOICEDETAILS_VATRATEAMOUNT	number			
INVOICEDETAILS_VATRATEPERCENT	number			
INVOICENUMBERINGBUSINESSUNITGUID	varchar2	Invoice Numbering Business Unit GUID		
INVOICEROWS	varchar2			
INVOICESTATUS	varchar2	Invoice Status		
ISCREDITNOTE	boolean	Is Credit Note		
ISREIMBURSED	boolean	Is Reimbursed		
ISTAXFREE	boolean	Is Tax Free		
LANGUAGECODE	varchar2	Language Code		
LANGUAGEGUID	varchar2	Language GUID		
MAINCASEGUID	varchar2	Main Case GUID		
NOTES	varchar2	Notes		
NOTESAFTERINVOICEROWS	varchar2	Notes after Invoice Lines		
NOTESBEFOREINVOICEROWS	varchar2	Notes before Invoice Lines		
NUMBER	number	Number		
ORDERNUMBER	varchar2	Order Number		
OURREFERENCE	varchar2	Our Reference		
OVERDUEINTEREST	number	Overdue Interest		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
PAYMENTDATE	datetime	Payment Date		
PAYMENTSTATUS	varchar2	Payment Status		
PAYMENTTERM	number	Payment Term		
PONUMBER	varchar2	PO Number		
RECEIVERADDRESSLINE	varchar2	Receiver Addressline		
RECEIVERCITY	varchar2	Receiver City		
RECEIVERCONTACTEMAIL	varchar2	Receiver Contact Email		
RECEIVERCONTACTNAME	varchar2	Receiver Contact Name		
RECEIVERCOUNTRY	varchar2	Receiver Country		
RECEIVERCUSTOMERNAME	varchar2	Receiver Customer Name		
RECEIVERPOSTALCODE	varchar2	Receiver Postal Code		
RECEIVERSTATE	varchar2	Receiver State		
RECEIVERVATNUMBER	varchar2	Receiver VAT Number		
REFERENCENUMBER	varchar2	Reference Number		
REIMBURSEINVOICEGUID	varchar2	Reimburse Invoice GUID		
SELLERCONTACT_SELLERACCOUNTDETAILS	varchar2			
SELLERCONTACT_SELLERCONTACTPERSONNAME	varchar2			
SELLERCONTACT_SELLEREMAILADDRESSIDENTIFIER	varchar2			
SELLERCONTACT_SELLERFAXNUMBER	varchar2			
SELLERCONTACT_SELLERFREETEXT	varchar2			
SELLERCONTACT_SELLERHOMETOWNNAME	varchar2			
SELLERCONTACT_SELLERPHONENUMBER	varchar2			
SELLERCONTACT_SELLERWEBADDRESSIDENTIFIER	varchar2			
SELLERORGANIZATION_SELLEREINVOICEADDRESS	varchar2			
SELLERORGANIZATION_SELLEREINVOICEOPERATOR	varchar2			
SELLERORGANIZATION_SELLERORGANISATIONNAME	varchar2			
SELLERORGANIZATION_SELLERORGANISATIONTAXCODE	varchar2			
SELLERORGANIZATION_SELLERPARTYIDENTIFIER	varchar2			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRY	varchar2			

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRYCODE	varchar2			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	varchar2			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_STREETNAME	varchar2			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_TOWNNAME	varchar2			
SENDERADDRESSLINE	varchar2	Sender Addressline		
SENDERCITY	varchar2	Sender City		
SENDERCONTACTNAME	varchar2	Sender Contact Name		
SENDERCOUNTRY	varchar2	Sender Country		
SENDERNAME	varchar2	Sender Name		
SENDERPOSTALCODE	varchar2	Sender Postal Code		
SENDERSTATE	varchar2	Sender State		
SENDERVATNUMBER	varchar2	Sender VAT Number		
TAXFREEDESCRIPTION	varchar2	Tax Free Description		
TITLE	varchar2	Title		
VIRTUALBANKBARCODE	varchar2	Virtual Bank Barcode		
YOURREFERENCE	varchar2	Your Reference		

InvoicesByDateAndStatus: Visma Severa Invoices by Date and Status

Catalog: Severa

Schema: API

Label: Invoices by Date and Status

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

Filters on API Server: businessUnitGUID- startDate- endDate- status- options

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function InvoicesByDateAndStatus. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
businessUnitGUID		☐		
endDate		☐		
options		☐		
startDate		☐		
status		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	varchar2	Account GUID		
ACCOUNTNAME	varchar2	Account Name		
ACCOUNTNUMBER	number	Account Number		
BUSINESSUNITCODE	varchar2	Business Unit Code		
BUSINESSUNITGUID	varchar2	Business Unit GUID		
BUSINESSUNITNAME	varchar2	Business Unit Name		
BUYERORGANIZATION_BUYERCONTACTPERSONNAME	varchar2			
BUYERORGANIZATION_BUYERELECTRONICINVOICEADDRESS	varchar2			
BUYERORGANIZATION_BUYERELECTRONICINVOICEOPERATOR	varchar2			
BUYERORGANIZATION_BUYEREMAILADDRESSIDENTIFIER	varchar2			
BUYERORGANIZATION_BUYERORGANISATIONNAME	varchar2			
BUYERORGANIZATION_BUYERORGANISATIONTAXCODE	varchar2			
BUYERORGANIZATION_BUYERPARTYIDENTIFIER	varchar2			
BUYERORGANIZATION_BUYERPHONENUMBERIDENTIFIER	varchar2			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRY	varchar2			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRYCODE	varchar2			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	varchar2			

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_STREETNAME	varchar2			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_TOWNNAME	varchar2			
CULTURE	varchar2	Culture		
CURRENCYGUID	varchar2	Currency GUID		
CURRENCYNAME	varchar2	Currency Name		
CURRENCYRATE	number	Currency Rate		
CURRENCYSHORTFORM	varchar2	Currency Short Form		
DATE	datetime	Date		
DESCRIPTION	varchar2	Ddescription		
DUEDATE	datetime	Due Date		
ENTRYDATE	datetime	Entry Date		
GUID	varchar2	GUID		
INVOICEALLOWEDCASEGUIDS	varchar2			
INVOICECONFIG_BILLEXPENSE	varchar2			
INVOICECONFIG_BILLHOUR	varchar2			
INVOICECONFIG_BILLPRODUCT	varchar2			
INVOICECONFIG_EXPENSEITEMDESCRIPTIONFORMAT	varchar2			
INVOICECONFIG_HOURITEMDESCRIPTIONFORMAT	varchar2			
INVOICECONFIG_HOURPRIMARYGROUPBY	varchar2			
INVOICECONFIG_HOURSECONDARYGROUPBY	varchar2			
INVOICECONFIG_ISTAXFREE	boolean			
INVOICECONFIG_PRODUCTITEMDESCRIPTIONFORMAT	varchar2			
INVOICECONFIG_PRODUCTPRIMARYGROUPBY	varchar2			
INVOICECONFIG_PRODUCTSECONDARYGROUPBY	varchar2			
INVOICECONFIG_SHOWATTACHPRICEEXCLUDINGVAT	boolean			
INVOICECONFIG_SHOWATTACHUNIT	boolean			
INVOICECONFIG_SHOWATTACHUNITPRICE	boolean			
INVOICECONFIG_SHOWATTACHVAT	boolean			
INVOICECONFIG_SHOWLOGOONINVOICE	boolean			
INVOICECONFIG_SHOWPRICEEXCLUDINGVAT	boolean			

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
INVOICECONFIG_SHOWQUANTITY	boolean			
INVOICECONFIG_SHOWUNIT	boolean			
INVOICECONFIG_SHOWUNITPRICE	boolean			
INVOICECONFIG_SHOWVAT	boolean			
INVOICECONFIG_TAXFREEDESCRIPTION	varchar2			
INVOICECONFIG_TRAVELPRIMARYGROUPBY	varchar2			
INVOICECONFIG_TRAVELSECONDARYGROUPBY	varchar2			
INVOICEDETAILS_INVOICEDATE	datetime			
INVOICEDETAILS_INVOICEDUEDATE	datetime			
INVOICEDETAILS_INVOICENUMBER	number			
INVOICEDETAILS_INVOICETOTALVATAMOUNT	number			
INVOICEDETAILS_INVOICETOTALVATEXCLUDEDAMOUNT	number			
INVOICEDETAILS_INVOICETOTALVATINCLUDEDAMOUNT	number			
INVOICEDETAILS_INVOICETYPETEXT	varchar2			
INVOICEDETAILS_ORDERIDENTIFIER	varchar2			
INVOICEDETAILS_ORIGINCODE	varchar2			
INVOICEDETAILS_PAYMENTOVERDUEFINEFREETEXT	varchar2			
INVOICEDETAILS_PAYMENTOVERDUEFINEPERCENT	number			
INVOICEDETAILS_PAYMENTTERMSFREETEXT	varchar2			
INVOICEDETAILS_SELLERREFERENCEIDENTIFIER	varchar2			
INVOICEDETAILS_VATBASEAMOUNT	number			
INVOICEDETAILS_VATRATEAMOUNT	number			
INVOICEDETAILS_VATRATEPERCENT	number			
INVOICENUMBERINGBUSINESSUNITGUID	varchar2	Invoice Numbering Business Unit GUID		
INVOICEROWS	varchar2			
INVOICESTATUS	varchar2	Invoice Status		
ISCREDITNOTE	boolean	Is Credit Note		
ISREIMBURSED	boolean	Is Reimbursed		
ISTAXFREE	boolean	Is Tax Free		
LANGUAGECODE	varchar2	Language Code		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
LANGUAGEGUID	varchar2	Language GUID		
MAINCASEGUID	varchar2	Main Case GUID		
NOTES	varchar2	Notes		
NOTESAFTERINVOICEROWS	varchar2	Notes after Invoice Lines		
NOTESBEFOREINVOICEROWS	varchar2	Notes before Invoice Lines		
NUMBER	number	Number		
ORDERNUMBER	varchar2	Order Number		
OURREFERENCE	varchar2	Our Reference		
OVERDUEINTEREST	number	Overdue Interest		
PAYMENTDATE	datetime	Payment Date		
PAYMENTSTATUS	varchar2	Payment Status		
PAYMENTTERM	number	Payment Term		
PONUMBER	varchar2	PO Number		
RECEIVERADDRESSLINE	varchar2	Receiver Addressline		
RECEIVERCITY	varchar2	Receiver City		
RECEIVERCONTACTEMAIL	varchar2	Receiver Contact Email		
RECEIVERCONTACTNAME	varchar2	Receiver Contact Name		
RECEIVERCOUNTRY	varchar2	Receiver Country		
RECEIVERCUSTOMERNAME	varchar2	Receiver Customer Name		
RECEIVERPOSTALCODE	varchar2	Receiver Postal Code		
RECEIVERSTATE	varchar2	Receiver State		
RECEIVERVATNUMBER	varchar2	Receiver VAT Number		
REFERENCENUMBER	varchar2	Reference Number		
REIMBURSEINVOICEGUID	varchar2	Reimburse Invoice GUID		
SELLERCONTACT_SELLERACCOUNTDETAILS	varchar2			
SELLERCONTACT_SELLERCONTACTPERSONNAME	varchar2			
SELLERCONTACT_SELLEREMAILADDRESSIDENTIFIER	varchar2			
SELLERCONTACT_SELLERFAXNUMBER	varchar2			
SELLERCONTACT_SELLERFREETEXT	varchar2			
SELLERCONTACT_SELLERHOMETOWNNAME	varchar2			
SELLERCONTACT_SELLERPHONENUMBER	varchar2			
SELLERCONTACT_SELLERWEBADDRESSIDENTIFIER	varchar2			

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
SELLERORGANIZATION_SELLEREINVOICEADDRESS	varchar2			
SELLERORGANIZATION_SELLEREINVOICEOPERATOR	varchar2			
SELLERORGANIZATION_SELLERORGANISATIONNAME	varchar2			
SELLERORGANIZATION_SELLERORGANISATIONTAXCODE	varchar2			
SELLERORGANIZATION_SELLERPARTYIDENTIFIER	varchar2			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRY	varchar2			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRYCODE	varchar2			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	varchar2			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_STREETNAME	varchar2			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_TOWNNAME	varchar2			
SENDERADDRESSLINE	varchar2	Sender Addressline		
SENDERCITY	varchar2	Sender City		
SENDERCONTACTNAME	varchar2	Sender Contact Name		
SENDERCOUNTRY	varchar2	Sender Country		
SENDERNAME	varchar2	Sender Name		
SENDERPOSTALCODE	varchar2	Sender Postal Code		
SENDERSTATE	varchar2	Sender State		
SENDERVATNUMBER	varchar2	Sender VAT Number		
TAXFREEDESCRIPTION	varchar2	Tax Free Description		
TITLE	varchar2	Title		
VIRTUALBANKBARCODE	varchar2	Virtual Bank Barcode		
YOURREFERENCE	varchar2	Your Reference		

InvoicesByStatus: Visma Severa Invoices by Status

Catalog: Severa

Schema: API

Label: Invoices by Status

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Filters on API Server: businessUnitGUID- status- options- maxInvoices

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function InvoicesByStatus. 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
businessUnitGUID		☐		
maxInvoices		☐		
options		☐		
status		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	varchar2	Account GUID		
ACCOUNTNAME	varchar2	Account Name		
ACCOUNTNUMBER	number	Account Number		
BUSINESSUNITCODE	varchar2	Business Unit Code		
BUSINESSUNITGUID	varchar2	Business Unit GUID		
BUSINESSUNITNAME	varchar2	Business Unit Name		
BUYERORGANIZATION_BUYERCONTACTPERSONNAME	varchar2			
BUYERORGANIZATION_BUYERELECTRONICINVOICEADDRESS	varchar2			
BUYERORGANIZATION_BUYERELECTRONICINVOICEOPERATOR	varchar2			
BUYERORGANIZATION_BUYEREMAILADDRESSIDENTIFIER	varchar2			
BUYERORGANIZATION_BUYERORGANISATIONNAME	varchar2			
BUYERORGANIZATION_BUYERORGANISATIONTAXCODE	varchar2			

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
BUYERORGANIZATION_BUYERPARTYIDENTIFIER	varchar2			
BUYERORGANIZATION_BUYERPHONENUMBERIDENTIFIER	varchar2			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRY	varchar2			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRYCODE	varchar2			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	varchar2			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_STREETNAME	varchar2			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_TOWNNAME	varchar2			
CULTURE	varchar2	Culture		
CURRENCYGUID	varchar2	Currency GUID		
CURRENCYNAME	varchar2	Currency Name		
CURRENCYRATE	number	Currency Rate		
CURRENCYSHORTFORM	varchar2	Currency Short Form		
DATE	datetime	Date		
DESCRIPTION	varchar2	Ddescription		
DUEDATE	datetime	Due Date		
ENTRYDATE	datetime	Entry Date		
GUID	varchar2	GUID		
INVOICEALLOWEDCASEGUIDS	varchar2			
INVOICECONFIG_BILLEXPENSE	varchar2			
INVOICECONFIG_BILLHOUR	varchar2			
INVOICECONFIG_BILLPRODUCT	varchar2			
INVOICECONFIG_EXPENSEITEMDESCRIPTIONFORMAT	varchar2			
INVOICECONFIG_HOURITEMDESCRIPTIONFORMAT	varchar2			
INVOICECONFIG_HOURPRIMARYGROUPBY	varchar2			
INVOICECONFIG_HOURSECONDARYGROUPBY	varchar2			
INVOICECONFIG_ISTAXFREE	boolean			
INVOICECONFIG_PRODUCTITEMDESCRIPTIONFORMAT	varchar2			
INVOICECONFIG_PRODUCTPRIMARYGROUPBY	varchar2			
INVOICECONFIG_PRODUCTSECONDARYGROUPBY	varchar2			

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
INVOICECONFIG_SHOWATTACHPRICEEXCLUDINGVAT	boolean			
INVOICECONFIG_SHOWATTACHUNIT	boolean			
INVOICECONFIG_SHOWATTACHUNITPRICE	boolean			
INVOICECONFIG_SHOWATTACHVAT	boolean			
INVOICECONFIG_SHOWLOGOONINVOICE	boolean			
INVOICECONFIG_SHOWPRICEEXCLUDINGVAT	boolean			
INVOICECONFIG_SHOWQUANTITY	boolean			
INVOICECONFIG_SHOWUNIT	boolean			
INVOICECONFIG_SHOWUNITPRICE	boolean			
INVOICECONFIG_SHOWVAT	boolean			
INVOICECONFIG_TAXFREEDESCRIPTION	varchar2			
INVOICECONFIG_TRAVELPRIMARYGROUPBY	varchar2			
INVOICECONFIG_TRAVELSECONDARYGROUPBY	varchar2			
INVOICEDETAILS_INVOICEDATE	datetime			
INVOICEDETAILS_INVOICEDUEDATE	datetime			
INVOICEDETAILS_INVOICENUMBER	number			
INVOICEDETAILS_INVOICETOTALVATAMOUNT	number			
INVOICEDETAILS_INVOICETOTALVATEXCLUDEDAMOUNT	number			
INVOICEDETAILS_INVOICETOTALVATINCLUDEDAMOUNT	number			
INVOICEDETAILS_INVOICETYPETEXT	varchar2			
INVOICEDETAILS_ORDERIDENTIFIER	varchar2			
INVOICEDETAILS_ORIGINCODE	varchar2			
INVOICEDETAILS_PAYMENTOVERDUEFINEFREETEXT	varchar2			
INVOICEDETAILS_PAYMENTOVERDUEFINEPERCENT	number			
INVOICEDETAILS_PAYMENTTERMSFREETEXT	varchar2			
INVOICEDETAILS_SELLERREFERENCEIDENTIFIER	varchar2			
INVOICEDETAILS_VATBASEAMOUNT	number			

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
INVOICEDETAILS_VATRATEAMOUNT	number			
INVOICEDETAILS_VATRATEPERCENT	number			
INVOICENUMBERINGBUSINESSUNITGUID	varchar2	Invoice Numbering Business Unit GUID		
INVOICEROWS	varchar2			
INVOICESTATUS	varchar2	Invoice Status		
ISCREDITNOTE	boolean	Is Credit Note		
ISREIMBURSED	boolean	Is Reimbursed		
ISTAXFREE	boolean	Is Tax Free		
LANGUAGECODE	varchar2	Language Code		
LANGUAGEGUID	varchar2	Language GUID		
MAINCASEGUID	varchar2	Main Case GUID		
NOTES	varchar2	Notes		
NOTESAFTERINVOICEROWS	varchar2	Notes after Invoice Lines		
NOTESBEFOREINVOICEROWS	varchar2	Notes before Invoice Lines		
NUMBER	number	Number		
ORDERNUMBER	varchar2	Order Number		
OURREFERENCE	varchar2	Our Reference		
OVERDUEINTEREST	number	Overdue Interest		
PAYMENTDATE	datetime	Payment Date		
PAYMENTSTATUS	varchar2	Payment Status		
PAYMENTTERM	number	Payment Term		
PONUMBER	varchar2	PO Number		
RECEIVERADDRESSLINE	varchar2	Receiver Addressline		
RECEIVERCITY	varchar2	Receiver City		
RECEIVERCONTACTEMAIL	varchar2	Receiver Contact Email		
RECEIVERCONTACTNAME	varchar2	Receiver Contact Name		
RECEIVERCOUNTRY	varchar2	Receiver Country		
RECEIVERCUSTOMERNAME	varchar2	Receiver Customer Name		
RECEIVERPOSTALCODE	varchar2	Receiver Postal Code		
RECEIVERSTATE	varchar2	Receiver State		
RECEIVERVATNUMBER	varchar2	Receiver VAT Number		
REFERENCENUMBER	varchar2	Reference Number		
REIMBURSEINVOICEGUID	varchar2	Reimburse Invoice GUID		
SELLERCONTACT_SELLERACCOUNTDETAILS	varchar2			

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
SELLERCONTACT_SELLERCONTACTPERSONNAME	varchar2			
SELLERCONTACT_SELLEREMAILADDRESSIDENTIFIER	varchar2			
SELLERCONTACT_SELLERFAXNUMBER	varchar2			
SELLERCONTACT_SELLERFREETEXT	varchar2			
SELLERCONTACT_SELLERHOMETOWNNAME	varchar2			
SELLERCONTACT_SELLERPHONENUMBER	varchar2			
SELLERCONTACT_SELLERWEBADDRESSIDENTIFIER	varchar2			
SELLERORGANIZATION_SELLEREINVOICEADDRESS	varchar2			
SELLERORGANIZATION_SELLEREINVOICEOPERATOR	varchar2			
SELLERORGANIZATION_SELLERORGANISATIONNAME	varchar2			
SELLERORGANIZATION_SELLERORGANISATIONTAXCODE	varchar2			
SELLERORGANIZATION_SELLERPARTYIDENTIFIER	varchar2			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRY	varchar2			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRYCODE	varchar2			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	varchar2			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_STREETNAME	varchar2			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_TOWNNAME	varchar2			
SENDERADDRESSLINE	varchar2	Sender Addressline		
SENDERCITY	varchar2	Sender City		
SENDERCONTACTNAME	varchar2	Sender Contact Name		
SENDERCOUNTRY	varchar2	Sender Country		
SENDERNAME	varchar2	Sender Name		
SENDERPOSTALCODE	varchar2	Sender Postal Code		
SENDERSTATE	varchar2	Sender State		
SENDERVATNUMBER	varchar2	Sender VAT Number		
TAXFREEDESCRIPTION	varchar2	Tax Free Description		
TITLE	varchar2	Title		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
VIRTUALBANKBARCODE	varchar2	Virtual Bank Barcode		
YOURREFERENCE	varchar2	Your Reference		

InvoicesByStatusAndBusinessUnit: Visma Severa Invoices by Status and Business Unit

Catalog: Severa

Schema: API

Label: Invoices by Status and Business Unit

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

Filters on API Server: status- businessUnitGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function InvoicesByStatusAndBusinessUnit. 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
businessUnitGUID		☐		
status		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	varchar2	Account GUID		
ACCOUNTNAME	varchar2	Account Name		
ACCOUNTNUMBER	number	Account Number		
BUSINESSUNITCODE	varchar2	Business Unit Code		
BUSINESSUNITGUID	varchar2	Business Unit GUID		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
BUSINESSUNITNAME	varchar2	Business Unit Name		
BUYERORGANIZATION_BUYERCONTACTPERSONNAME	varchar2			
BUYERORGANIZATION_BUYERELECTRONICINVOICEADDRESS	varchar2			
BUYERORGANIZATION_BUYERELECTRONICINVOICEOPERATOR	varchar2			
BUYERORGANIZATION_BUYEREMAILADDRESSIDENTIFIER	varchar2			
BUYERORGANIZATION_BUYERORGANISATIONNAME	varchar2			
BUYERORGANIZATION_BUYERORGANISATIONTAXCODE	varchar2			
BUYERORGANIZATION_BUYERPARTYIDENTIFIER	varchar2			
BUYERORGANIZATION_BUYERPHONENUMBERIDENTIFIER	varchar2			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRY	varchar2			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRYCODE	varchar2			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	varchar2			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_STREETNAME	varchar2			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_TOWNNAME	varchar2			
CULTURE	varchar2	Culture		
CURRENCYGUID	varchar2	Currency GUID		
CURRENCYNAME	varchar2	Currency Name		
CURRENCYRATE	number	Currency Rate		
CURRENCYSHORTFORM	varchar2	Currency Short Form		
DATE	datetime	Date		
DESCRIPTION	varchar2	Ddescription		
DUEDATE	datetime	Due Date		
ENTRYDATE	datetime	Entry Date		
GUID	varchar2	GUID		
INVOICEALLOWEDCASEGUIDS	varchar2			
INVOICECONFIG_BILLEXPENSE	varchar2			
INVOICECONFIG_BILLHOUR	varchar2			
INVOICECONFIG_BILLPRODUCT	varchar2			
INVOICECONFIG_EXPENSEITEMDESCRIPTIONFORMAT	varchar2			

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
INVOICECONFIG_HOURITEMDESCRIPTIONFORMAT	varchar2			
INVOICECONFIG_HOURPRIMARYGROUPBY	varchar2			
INVOICECONFIG_HOURSECONDARYGROUPBY	varchar2			
INVOICECONFIG_ISTAXFREE	boolean			
INVOICECONFIG_PRODUCTITEMDESCRIPTIONFORMAT	varchar2			
INVOICECONFIG_PRODUCTPRIMARYGROUPBY	varchar2			
INVOICECONFIG_PRODUCTSECONDARYGROUPBY	varchar2			
INVOICECONFIG_SHOWATTACHPRICEEXCLUDINGVAT	boolean			
INVOICECONFIG_SHOWATTACHUNIT	boolean			
INVOICECONFIG_SHOWATTACHUNITPRICE	boolean			
INVOICECONFIG_SHOWATTACHVAT	boolean			
INVOICECONFIG_SHOWLOGOONINVOICE	boolean			
INVOICECONFIG_SHOWPRICEEXCLUDINGVAT	boolean			
INVOICECONFIG_SHOWQUANTITY	boolean			
INVOICECONFIG_SHOWUNIT	boolean			
INVOICECONFIG_SHOWUNITPRICE	boolean			
INVOICECONFIG_SHOWVAT	boolean			
INVOICECONFIG_TAXFREEDESCRIPTION	varchar2			
INVOICECONFIG_TRAVELPRIMARYGROUPBY	varchar2			
INVOICECONFIG_TRAVELSECONDARYGROUPBY	varchar2			
INVOICEDETAILS_INVOICEDATE	datetime			
INVOICEDETAILS_INVOICEDUEDATE	datetime			
INVOICEDETAILS_INVOICENUMBER	number			
INVOICEDETAILS_INVOICETOTALVATAMOUNT	number			
INVOICEDETAILS_INVOICETOTALVATEXCLUDEDAMOUNT	number			
INVOICEDETAILS_INVOICETOTALVATINCLUDEDAMOUNT	number			
INVOICEDETAILS_INVOICETYPETEXT	varchar2			

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
INVOICEDETAILS_ORDERIDENTIFIER	varchar2			
INVOICEDETAILS_ORIGINCODE	varchar2			
INVOICEDETAILS_PAYMENTOVERDUEFINEFREETEXT	varchar2			
INVOICEDETAILS_PAYMENTOVERDUEFINEPERCENT	number			
INVOICEDETAILS_PAYMENTTERMSFREETEXT	varchar2			
INVOICEDETAILS_SELLERREFERENCEIDENTIFIER	varchar2			
INVOICEDETAILS_VATBASEAMOUNT	number			
INVOICEDETAILS_VATRATEAMOUNT	number			
INVOICEDETAILS_VATRATEPERCENT	number			
INVOICENUMBERINGBUSINESSUNITGUID	varchar2	Invoice Numbering Business Unit GUID		
INVOICEROWS	varchar2			
INVOICESTATUS	varchar2	Invoice Status		
ISCREDITNOTE	boolean	Is Credit Note		
ISREIMBURSED	boolean	Is Reimbursed		
ISTAXFREE	boolean	Is Tax Free		
LANGUAGECODE	varchar2	Language Code		
LANGUAGEGUID	varchar2	Language GUID		
MAINCASEGUID	varchar2	Main Case GUID		
NOTES	varchar2	Notes		
NOTESAFTERINVOICEROWS	varchar2	Notes after Invoice Lines		
NOTESBEFOREINVOICEROWS	varchar2	Notes before Invoice Lines		
NUMBER	number	Number		
ORDERNUMBER	varchar2	Order Number		
OURREFERENCE	varchar2	Our Reference		
OVERDUEINTEREST	number	Overdue Interest		
PAYMENTDATE	datetime	Payment Date		
PAYMENTSTATUS	varchar2	Payment Status		
PAYMENTTERM	number	Payment Term		
PONUMBER	varchar2	PO Number		
RECEIVERADDRESSLINE	varchar2	Receiver Addressline		
RECEIVERCITY	varchar2	Receiver City		
RECEIVERCONTACTEMAIL	varchar2	Receiver Contact Email		
RECEIVERCONTACTNAME	varchar2	Receiver Contact Name		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
RECEIVERCOUNTRY	varchar2	Receiver Country		
RECEIVERCUSTOMERNAME	varchar2	Receiver Customer Name		
RECEIVERPOSTALCODE	varchar2	Receiver Postal Code		
RECEIVERSTATE	varchar2	Receiver State		
RECEIVERVATNUMBER	varchar2	Receiver VAT Number		
REFERENCENUMBER	varchar2	Reference Number		
REIMBURSEINVOICEGUID	varchar2	Reimburse Invoice GUID		
SELLERCONTACT_SELLERACCOUNTDETAILS	varchar2			
SELLERCONTACT_SELLERCONTACTPERSONNAME	varchar2			
SELLERCONTACT_SELLEREMAILADDRESSIDENTIFIER	varchar2			
SELLERCONTACT_SELLERFAXNUMBER	varchar2			
SELLERCONTACT_SELLERFREETEXT	varchar2			
SELLERCONTACT_SELLERHOMETOWNNAME	varchar2			
SELLERCONTACT_SELLERPHONENUMBER	varchar2			
SELLERCONTACT_SELLERWEBADDRESSIDENTIFIER	varchar2			
SELLERORGANIZATION_SELLEREINVOICEADDRESS	varchar2			
SELLERORGANIZATION_SELLEREINVOICEOPERATOR	varchar2			
SELLERORGANIZATION_SELLERORGANISATIONNAME	varchar2			
SELLERORGANIZATION_SELLERORGANISATIONTAXCODE	varchar2			
SELLERORGANIZATION_SELLERPARTYIDENTIFIER	varchar2			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRY	varchar2			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRYCODE	varchar2			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	varchar2			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_STREETNAME	varchar2			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_TOWNNAME	varchar2			

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
SENDERADDRESSLINE	varchar2	Sender Addressline		
SENDERCITY	varchar2	Sender City		
SENDERCONTACTNAME	varchar2	Sender Contact Name		
SENDERCOUNTRY	varchar2	Sender Country		
SENDERNAME	varchar2	Sender Name		
SENDERPOSTALCODE	varchar2	Sender Postal Code		
SENDERSTATE	varchar2	Sender State		
SENDERVATNUMBER	varchar2	Sender VAT Number		
TAXFREEDESCRIPTION	varchar2	Tax Free Description		
TITLE	varchar2	Title		
VIRTUALBANKBARCODE	varchar2	Virtual Bank Barcode		
YOURREFERENCE	varchar2	Your Reference		

InvoiceStatuses: Visma Severa Invoice Statuses

Catalog: Severa

Schema: API

Label: Invoice Statuses

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

Table Columns

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

Name	Data Type	Label	Required	Documentation
DESCRIPTION	varchar2	Ddescription		
HASINVOICENUMBER	boolean	Has Invoice Number		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
ISPAID	boolean	Is Paid		
ISREADONLY	boolean	Is Read-only		
ISSENT	boolean	Is Sent		
ISWAITINGPAYMENT	boolean	Is Waiting Payment		
NAME	varchar2	Name		
SORTORDER	number	Sort Order		

ItemByGUID: Visma Severa Item by GUID

Catalog: Severa

Schema: API

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Label: Item by GUID

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

Filters on API Server: itemGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ItemByGUID. 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
itemGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
BILLINGDEPENDENCYTASKGUID	varchar2	Billing Dependency Task GUID		
BILLINGSCHEDULE	varchar2	Billing Schedule		
CASEGUID	varchar2	Project GUID		
CASENUMBER	number	Project Number		
COSTCENTERGUID	varchar2	Cost Centre GUID		
COSTCENTERIDENTIFIER	varchar2	Cost Centre Identifier		
COSTCURRENCYSHORTFORM	varchar2	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Cost Currency Short Form of Organisation Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Cost Currency Short Form of User Company Currency		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
DESCRIPTION	varchar2	Ddescription		
DISPLAYPERIODSTART	datetime	Display Period Start		
EVENTDATE	datetime	Event Date		
GUID	varchar2	GUID		
INVOICEDESCRIPTION	varchar2	Invoice Description		
INVOICEGUID	varchar2	Invoice GUID		
INVOICEQUANTITY	number	Invoice Quantity		
INVOICEROWGUID	varchar2	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISOUTSOURCINGCOST	boolean	Is Out-sourcing Cost		
ISREADONLY	boolean	Is Read-only		
ISREALISED	boolean	Is Realized		
MEASUREMENTUNIT	varchar2	Measurement Unit		
NAME	varchar2	Name		
ORDERDATE	datetime	Order Date		
PLANNEDBILLINGDATE	datetime	Planned Billing Date		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	varchar2	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Price Currency Short Form of Organisation Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Price Currency Short Form of User Company Currency		
PRODUCTCODE	varchar2	Product Code		
PRODUCTGUID	varchar2	Product GUID		
PRODUCTTYPE	varchar2	Product Type		
PURCHASEORDERNUMBER	varchar2	Purchase Order Number		
PURCHASEVAT	number	Purchase VAT		
QUANTITY	number	Quantity		
RECURRENCEENDDATE	datetime	Recurrence End Date		
RECURRENCEENDTYPE	number	Recurrence End Type		
RECURRENCEFREQUENCY	number	Recurrence Frequency		
RECURRENCESTARTDATE	datetime	Recurrence Start Date		
RECURRENCETIMES	number	Recurrence Times		
RECURRINGITEMGUID	varchar2	Recurring Item GUID		
SALESACCOUNTGUID	varchar2	Sales Account GUID		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
SALESSTATUSGUID	varchar2	Sales Status GUID		
TASKGUID	varchar2	Task GUID		
TRAVELENDTIME	datetime	Travel End Time		
TRAVELSTARTTIME	datetime	Travel Start Time		
UNITCOST	number	Unit Cost		
UNITCOSTININVOICINGCURRENCY	number	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	number	Unit Cost in Organisation Currency		
UNITCOSTINUSERCOMPANYCURRENCY	number	Unit Cost in User Company Currency		
UNITCOSTVATINCLUDEDAMOUNT	number	Unit Cost VAT Included Amount		
UNITPRICE	number	Unit Price		
UNITPRICEINBASECURRENCY	number	Unit Price in Base Currency		
UNITPRICEINCASECOMPANYCURRENCY	number	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	number	Unit Price in User Company Currency		
USERCODE	varchar2	User Code		
USERGUID	varchar2	User GUID		
VAT	number	VAT		

ItemByPurchaseOrderNumber: Visma Severa Item by Purchase Order Number

Catalog: Severa

Schema: API

Label: Item by Purchase Order Number

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

Filters on API Server: poNumber

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ItemByPurchaseOrderNumber. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
poNumber		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
BILLINGDEPENDENCYTASKGUID	varchar2	Billing Dependency Task GUID		
BILLINGSCHEDULE	varchar2	Billing Schedule		
CASEGUID	varchar2	Project GUID		
CASENUMBER	number	Project Number		
COSTCENTERGUID	varchar2	Cost Centre GUID		
COSTCENTERIDENTIFIER	varchar2	Cost Centre Identifier		
COSTCURRENCYSHORTFORM	varchar2	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Cost Currency Short Form of Organisation Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Cost Currency Short Form of User Company Currency		
DESCRIPTION	varchar2	Ddescription		
DISPLAYPERIODSTART	datetime	Display Period Start		
EVENTDATE	datetime	Event Date		
GUID	varchar2	GUID		
INVOICEDESCRIPTION	varchar2	Invoice Description		
INVOICEGUID	varchar2	Invoice GUID		
INVOICEQUANTITY	number	Invoice Quantity		
INVOICEROWGUID	varchar2	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISOUTSOURCINGCOST	boolean	Is Out-sourcing Cost		
ISREADONLY	boolean	Is Read-only		
ISREALISED	boolean	Is Realized		
MEASUREMENTUNIT	varchar2	Measurement Unit		
NAME	varchar2	Name		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
ORDERDATE	datetime	Order Date		
PLANNEDBILLINGDATE	datetime	Planned Billing Date		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	varchar2	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Price Currency Short Form of Organisation Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Price Currency Short Form of User Company Currency		
PRODUCTCODE	varchar2	Product Code		
PRODUCTGUID	varchar2	Product GUID		
PRODUCTTYPE	varchar2	Product Type		
PURCHASEORDERNUMBER	varchar2	Purchase Order Number		
PURCHASEVAT	number	Purchase VAT		
QUANTITY	number	Quantity		
RECURRENCEENDDATE	datetime	Recurrence End Date		
RECURRENCEENDTYPE	number	Recurrence End Type		
RECURRENCEFREQUENCY	number	Recurrence Frequency		
RECURRENCESTARTDATE	datetime	Recurrence Start Date		
RECURRENCETIMES	number	Recurrence Times		
RECURRINGITEMGUID	varchar2	Recurring Item GUID		
SALESACCOUNTGUID	varchar2	Sales Account GUID		
SALESSTATUSGUID	varchar2	Sales Status GUID		
TASKGUID	varchar2	Task GUID		
TRAVELENDTIME	datetime	Travel End Time		
TRAVELSTARTTIME	datetime	Travel Start Time		
UNITCOST	number	Unit Cost		
UNITCOSTININVOICINGCURRENCY	number	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	number	Unit Cost in Organisation Currency		
UNITCOSTINUSERCOMPANYCURRENCY	number	Unit Cost in User Company Currency		
UNITCOSTVATINCLUDEDAMOUNT	number	Unit Cost VAT Included Amount		
UNITPRICE	number	Unit Price		
UNITPRICEINBASECURRENCY	number	Unit Price in Base Currency		
UNITPRICEINCASECOMPANYCURRENCY	number	Unit Price in Project Company Currency		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
UNITPRICEINUSERCOMPANYCURRENCY	number	Unit Price in User Company Currency		
USERCODE	varchar2	User Code		
USERGUID	varchar2	User GUID		
VAT	number	VAT		

ItemsByCaseGUID: Visma Severa Items by Project GUID

Catalog: Severa

Schema: API

Label: Items by Project GUID

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

Filters on API Server: caseGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ItemsByCaseGUID. 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
caseGUID		□		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
BILLINGDEPENDENCYTASKGUID	varchar2	Billing Dependency Task GUID		
BILLINGSCHEDULE	varchar2	Billing Schedule		
CASEGUID	varchar2	Project GUID		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
CASENUMBER	number	Project Number		
COSTCENTERGUID	varchar2	Cost Centre GUID		
COSTCENTERIDENTIFIER	varchar2	Cost Centre Identifier		
COSTCURRENCYSHORTFORM	varchar2	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Cost Currency Short Form of Organisation Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Cost Currency Short Form of User Company Currency		
DESCRIPTION	varchar2	Ddescription		
DISPLAYPERIODSTART	datetime	Display Period Start		
EVENTDATE	datetime	Event Date		
GUID	varchar2	GUID		
INVOICEDESCRIPTION	varchar2	Invoice Description		
INVOICEGUID	varchar2	Invoice GUID		
INVOICEQUANTITY	number	Invoice Quantity		
INVOICEROWGUID	varchar2	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISOUTSOURCINGCOST	boolean	Is Out-sourcing Cost		
ISREADONLY	boolean	Is Read-only		
ISREALISED	boolean	Is Realized		
MEASUREMENTUNIT	varchar2	Measurement Unit		
NAME	varchar2	Name		
ORDERDATE	datetime	Order Date		
PLANNEDBILLINGDATE	datetime	Planned Billing Date		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	varchar2	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Price Currency Short Form of Organisation Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Price Currency Short Form of User Company Currency		
PRODUCTCODE	varchar2	Product Code		
PRODUCTGUID	varchar2	Product GUID		
PRODUCTTYPE	varchar2	Product Type		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
PURCHASEORDERNUMBER	varchar2	Purchase Order Number		
PURCHASEVAT	number	Purchase VAT		
QUANTITY	number	Quantity		
RECURRENCEENDDATE	datetime	Recurrence End Date		
RECURRENCEENDTYPE	number	Recurrence End Type		
RECURRENCEFREQUENCY	number	Recurrence Frequency		
RECURRENCESTARTDATE	datetime	Recurrence Start Date		
RECURRENCETIMES	number	Recurrence Times		
RECURRINGITEMGUID	varchar2	Recurring Item GUID		
SALESACCOUNTGUID	varchar2	Sales Account GUID		
SALESSTATUSGUID	varchar2	Sales Status GUID		
TASKGUID	varchar2	Task GUID		
TRAVELENDTIME	datetime	Travel End Time		
TRAVELSTARTTIME	datetime	Travel Start Time		
UNITCOST	number	Unit Cost		
UNITCOSTININVOICINGCURRENCY	number	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	number	Unit Cost in Organisation Currency		
UNITCOSTINUSERCOMPANYCURRENCY	number	Unit Cost in User Company Currency		
UNITCOSTVATINCLUDEDAMOUNT	number	Unit Cost VAT Included Amount		
UNITPRICE	number	Unit Price		
UNITPRICEINBASECURRENCY	number	Unit Price in Base Currency		
UNITPRICEINCASECOMPANYCURRENCY	number	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	number	Unit Price in User Company Currency		
USERCODE	varchar2	User Code		
USERGUID	varchar2	User GUID		
VAT	number	VAT		

ItemsByInsertDate: Visma Severa Items by Insert Date

Catalog: Severa

Schema: API

Label: Items by Insert Date

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Filters on API Server: insertStartDate- insertEndDate

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ItemsByInsertDate. 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
insertEndDate		☐		
insertStartDate		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
BILLINGDEPENDENCYTASKGUID	varchar2	Billing Dependency Task GUID		
BILLINGSCHEDULE	varchar2	Billing Schedule		
CASEGUID	varchar2	Project GUID		
CASENUMBER	number	Project Number		
COSTCENTERGUID	varchar2	Cost Centre GUID		
COSTCENTERIDENTIFIER	varchar2	Cost Centre Identifier		
COSTCURRENCYSHORTFORM	varchar2	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Cost Currency Short Form of Organisation Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Cost Currency Short Form of User Company Currency		
DESCRIPTION	varchar2	Ddescription		
DISPLAYPERIODSTART	datetime	Display Period Start		
EVENTDATE	datetime	Event Date		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
GUID	varchar2	GUID		
INVOICEDESCRIPTION	varchar2	Invoice Description		
INVOICEGUID	varchar2	Invoice GUID		
INVOICEQUANTITY	number	Invoice Quantity		
INVOICEROWGUID	varchar2	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISOUTSOURCINGCOST	boolean	Is Out-sourcing Cost		
ISREADONLY	boolean	Is Read-only		
ISREALISED	boolean	Is Realized		
MEASUREMENTUNIT	varchar2	Measurement Unit		
NAME	varchar2	Name		
ORDERDATE	datetime	Order Date		
PLANNEDBILLINGDATE	datetime	Planned Billing Date		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	varchar2	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Price Currency Short Form of Organisation Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Price Currency Short Form of User Company Currency		
PRODUCTCODE	varchar2	Product Code		
PRODUCTGUID	varchar2	Product GUID		
PRODUCTTYPE	varchar2	Product Type		
PURCHASEORDERNUMBER	varchar2	Purchase Order Number		
PURCHASEVAT	number	Purchase VAT		
QUANTITY	number	Quantity		
RECURRENCEENDDATE	datetime	Recurrence End Date		
RECURRENCEENDTYPE	number	Recurrence End Type		
RECURRENCEFREQUENCY	number	Recurrence Frequency		
RECURRENCESTARTDATE	datetime	Recurrence Start Date		
RECURRENCETIMES	number	Recurrence Times		
RECURRINGITEMGUID	varchar2	Recurring Item GUID		
SALESACCOUNTGUID	varchar2	Sales Account GUID		
SALESSTATUSGUID	varchar2	Sales Status GUID		
TASKGUID	varchar2	Task GUID		
TRAVELENDTIME	datetime	Travel End Time		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
TRAVELSTARTTIME	datetime	Travel Start Time		
UNITCOST	number	Unit Cost		
UNITCOSTININVOICINGCURRENCY	number	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	number	Unit Cost in Organisation Currency		
UNITCOSTINUSERCOMPANYCURRENCY	number	Unit Cost in User Company Currency		
UNITCOSTVATINCLUDEDAMOUNT	number	Unit Cost VAT Included Amount		
UNITPRICE	number	Unit Price		
UNITPRICEINBASECURRENCY	number	Unit Price in Base Currency		
UNITPRICEINCASECOMPANYCURRENCY	number	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	number	Unit Price in User Company Currency		
USERCODE	varchar2	User Code		
USERGUID	varchar2	User GUID		
VAT	number	VAT		

ItemsChangedSince: Visma Severa Items Changed since

Catalog: Severa

Schema: API

Label: Items Changed since

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

Filters on API Server: startDate

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ItemsChangedSince. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
startDate		□		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
BILLINGDEPENDENCYTASKGUID	varchar2	Billing Dependency Task GUID		
BILLINGSCHEDULE	varchar2	Billing Schedule		
CASEGUID	varchar2	Project GUID		
CASENUMBER	number	Project Number		
COSTCENTERGUID	varchar2	Cost Centre GUID		
COSTCENTERIDENTIFIER	varchar2	Cost Centre Identifier		
COSTCURRENCYSHORTFORM	varchar2	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Cost Currency Short Form of Organisation Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Cost Currency Short Form of User Company Currency		
DESCRIPTION	varchar2	Ddescription		
DISPLAYPERIODSTART	datetime	Display Period Start		
EVENTDATE	datetime	Event Date		
GUID	varchar2	GUID		
INVOICEDESCRIPTION	varchar2	Invoice Description		
INVOICEGUID	varchar2	Invoice GUID		
INVOICEQUANTITY	number	Invoice Quantity		
INVOICEROWGUID	varchar2	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISOUTSOURCINGCOST	boolean	Is Out-sourcing Cost		
ISREADONLY	boolean	Is Read-only		
ISREALISED	boolean	Is Realized		
MEASUREMENTUNIT	varchar2	Measurement Unit		
NAME	varchar2	Name		
ORDERDATE	datetime	Order Date		
PLANNEDBILLINGDATE	datetime	Planned Billing Date		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	varchar2	Price Currency Short Form of Project		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
		Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Price Currency Short Form of Organisation Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Price Currency Short Form of User Company Currency		
PRODUCTCODE	varchar2	Product Code		
PRODUCTGUID	varchar2	Product GUID		
PRODUCTTYPE	varchar2	Product Type		
PURCHASEORDERNUMBER	varchar2	Purchase Order Number		
PURCHASEVAT	number	Purchase VAT		
QUANTITY	number	Quantity		
RECURRENCEENDDATE	datetime	Recurrence End Date		
RECURRENCEENDTYPE	number	Recurrence End Type		
RECURRENCEFREQUENCY	number	Recurrence Frequency		
RECURRENCESTARTDATE	datetime	Recurrence Start Date		
RECURRENCETIMES	number	Recurrence Times		
RECURRINGITEMGUID	varchar2	Recurring Item GUID		
SALESACCOUNTGUID	varchar2	Sales Account GUID		
SALESSTATUSGUID	varchar2	Sales Status GUID		
TASKGUID	varchar2	Task GUID		
TRAVELENDTIME	datetime	Travel End Time		
TRAVELSTARTTIME	datetime	Travel Start Time		
UNITCOST	number	Unit Cost		
UNITCOSTININVOICINGCURRENCY	number	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	number	Unit Cost in Organisation Currency		
UNITCOSTINUSERCOMPANYCURRENCY	number	Unit Cost in User Company Currency		
UNITCOSTVATINCLUDEDAMOUNT	number	Unit Cost VAT Included Amount		
UNITPRICE	number	Unit Price		
UNITPRICEINBASECURRENCY	number	Unit Price in Base Currency		
UNITPRICEINCASECOMPANYCURRENCY	number	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	number	Unit Price in User Company Currency		
USERCODE	varchar2	User Code		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
USERGUID	varchar2	User GUID		
VAT	number	VAT		

LanguageByGUID: Visma Severa Language by GUID

Catalog: Severa

Schema: API

Label: Language by GUID

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

Filters on API Server: languageGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function LanguageByGUID. 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
languageGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
GUID	varchar2	GUID		
IETFLANGUAGETAG	varchar2	IETF Language Tag		
NAME	varchar2	Name		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

LanguageByIetfTag: Visma Severa Language by IETF Tag

Catalog: Severa

Schema: API

Label: Language by IETF Tag

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

Filters on API Server: ietfLanguageTag

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function LanguageByIetfTag. 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
ietfLanguageTag		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
GUID	varchar2	GUID		
IETFLANGUAGETAG	varchar2	IETF Language Tag		
NAME	varchar2	Name		

LeadSourceByGUID: Visma Severa Lead Source by GUID

Catalog: Severa

Schema: API

Label: Lead Source by GUID

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Filters on API Server: leadSourceGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function LeadSourceByGUID. 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
leadSourceGUID		□		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
NAME	varchar2	Name		

LeadSources: Visma Severa Lead Sources

Catalog: Severa

Schema: API

Label: Lead Sources

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

Table Columns

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

Name	Data Type	Label	Required	Documentation
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
NAME	varchar2	Name		

OverTimeByGUID: Visma Severa Overtime by GUID

Catalog: Severa

Schema: API

Label: Overtime by GUID

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

Filters on API Server: overTimeGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function OverTimeByGUID. 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
overTimeGUID		□		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CODE	varchar2	Code		
GUID	varchar2	GUID		
INCLUDEINFLEXTIME	boolean	Include in Flextime		
ISACTIVE	boolean	Is Active		
ISWORKTIME	boolean	Is Work Time		
NAME	varchar2	Name		
PERCENTAGE	number	Percentage		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Overtimes: Visma Severa Overtimes

Catalog: Severa

Schema: API

Label: Overtimes

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

Table Columns

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

Name	Data Type	Label	Required	Documentation
CODE	varchar2	Code		
GUID	varchar2	GUID		
INCLUDEINFLEXTIME	boolean	Include in Flextime		
ISACTIVE	boolean	Is Active		
ISWORKTIME	boolean	Is Work Time		
NAME	varchar2	Name		
PERCENTAGE	number	Percentage		

PhaseByGUID: Visma Severa Phase by GUID

Catalog: Severa

Schema: API

Label: Phase by GUID

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

Filters on API Server: phaseGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function PhaseByGUID. 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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
phaseGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CASEGUID	varchar2	Project GUID		
CODE	varchar2	Code		
DEADLINE	datetime	Deadline		
DESCRIPTION	varchar2	Ddescription		
FLATRATE_ADDITIONALHOURSUNITPRICE	number	Flat Rate Additional Hours Unit Price		
FLATRATE_AREADDITIONALHOURSBILLABLE	boolean	Flat Rate Are additional Hours Billable		
FLATRATE_HOURS	number	Flat Rate Hours		
FLATRATE_PRICE	number	Flat Rate Price		
GUID	varchar2	GUID		
ISCOMPLETED	boolean	Is Completed		
ISLOCKED	boolean	Is Locked		
NAME	varchar2	Name		
ORIGINALDEADLINE	datetime	Original Deadline		
ORIGINALPLANNEDSTARTDATE	datetime	Original Planned Start Date		
ORIGINALWORKESTIMATE	number	Original Work Estimate		
OWNERUSERGUID	varchar2	Owner User GUID		
PARENTPHASEGUID	varchar2	Parent Phase GUID		
PLANNEDSTARTDATE	datetime	Planned Start Date		
PRICEPERHOUR	number	Price per Hour		
SORTORDER	number	Sort Order		
WORKESTIMATE	number	Work Estimate		
WORKTYPEGUID	varchar2	Work Type GUID		

PhaseMembersByPhaseGUID: Visma Severa Phase Members by Phase GUID

Catalog: Severa

Schema: API

Label: Phase Members by Phase GUID

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Filters on API Server: phaseGUID- onlyActive

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function PhaseMembersByPhaseGUID. 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
onlyActive		☐		
phaseGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
STRING	varchar2			

PhasesByCaseGUID: Visma Severa Phases by Project GUID

Catalog: Severa

Schema: API

Label: Phases by Project GUID

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

Filters on API Server: caseGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function PhasesByCaseGUID. A value must be provided at all times for required parameters, but

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
caseGUID		□		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CASEGUID	varchar2	Project GUID		
CODE	varchar2	Code		
DEADLINE	datetime	Deadline		
DESCRIPTION	varchar2	Ddescription		
FLATRATE_ADDITIONALHOURSUNITPRICE	number	Flat Rate Additional Hours Unit Price		
FLATRATE_AREADDITIONALHOURSBILLABLE	boolean	Flat Rate Are additional Hours Billable		
FLATRATE_HOURS	number	Flat Rate Hours		
FLATRATE_PRICE	number	Flat Rate Price		
GUID	varchar2	GUID		
ISCOMPLETED	boolean	Is Completed		
ISLOCKED	boolean	Is Locked		
NAME	varchar2	Name		
ORIGINALDEADLINE	datetime	Original Deadline		
ORIGINALPLANNEDSTARTDATE	datetime	Original Planned Start Date		
ORIGINALWORKESTIMATE	number	Original Work Estimate		
OWNERUSERGUID	varchar2	Owner User GUID		
PARENTPHASEGUID	varchar2	Parent Phase GUID		
PLANNEDSTARTDATE	datetime	Planned Start Date		
PRICEPERHOUR	number	Price per Hour		
SORTORDER	number	Sort Order		
WORKESTIMATE	number	Work Estimate		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
WORKTYPEGUID	varchar2	Work Type GUID		

PhasesByCode: Visma Severa Phases by Code

Catalog: Severa

Schema: API

Label: Phases by Code

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

Filters on API Server: phaseCode- caseGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function PhasesByCode. 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
caseGUID		☐		
phaseCode		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CASEGUID	varchar2	Project GUID		
CODE	varchar2	Code		
DEADLINE	datetime	Deadline		
DESCRIPTION	varchar2	Ddescription		
FLATRATE_ADDITIONALHOURSUNITPRICE	number	Flat Rate Additional Hours Unit Price		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
FLATRATE_AREADDITIONALHOURSBILLABLE	boolean	Flat Rate Are additional Hours Billable		
FLATRATE_HOURS	number	Flat Rate Hours		
FLATRATE_PRICE	number	Flat Rate Price		
GUID	varchar2	GUID		
ISCOMPLETED	boolean	Is Completed		
ISLOCKED	boolean	Is Locked		
NAME	varchar2	Name		
ORIGINALDEADLINE	datetime	Original Deadline		
ORIGINALPLANNEDSTARTDATE	datetime	Original Planned Start Date		
ORIGINALWORKESTIMATE	number	Original Work Estimate		
OWNERUSERGUID	varchar2	Owner User GUID		
PARENTPHASEGUID	varchar2	Parent Phase GUID		
PLANNEDSTARTDATE	datetime	Planned Start Date		
PRICEPERHOUR	number	Price per Hour		
SORTORDER	number	Sort Order		
WORKESTIMATE	number	Work Estimate		
WORKTYPEGUID	varchar2	Work Type GUID		

PhasesChangedSince: Visma Severa Phases Changed since

Catalog: Severa

Schema: API

Label: Phases Changed since

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

Filters on API Server: businessUnitGUID- startDateTime- options

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function PhasesChangedSince. 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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
businessUnitGUID		☐		
options		☐		
startDateTime		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CASEGUID	varchar2	Project GUID		
CODE	varchar2	Code		
DEADLINE	datetime	Deadline		
DESCRIPTION	varchar2	Ddescription		
FLATRATE_ADDITIONALHOURSUNITPRICE	number	Flat Rate Additional Hours Unit Price		
FLATRATE_AREADDITIONALHOURSBILLABLE	boolean	Flat Rate Are additional Hours Billable		
FLATRATE_HOURS	number	Flat Rate Hours		
FLATRATE_PRICE	number	Flat Rate Price		
GUID	varchar2	GUID		
ISCOMPLETED	boolean	Is Completed		
ISLOCKED	boolean	Is Locked		
NAME	varchar2	Name		
ORIGINALDEADLINE	datetime	Original Deadline		
ORIGINALPLANNEDSTARTDATE	datetime	Original Planned Start Date		
ORIGINALWORKESTIMATE	number	Original Work Estimate		
OWNERUSERGUID	varchar2	Owner User GUID		
PARENTPHASEGUID	varchar2	Parent Phase GUID		
PLANNEDSTARTDATE	datetime	Planned Start Date		
PRICEPERHOUR	number	Price per Hour		
SORTORDER	number	Sort Order		
WORKESTIMATE	number	Work Estimate		
WORKTYPEGUID	varchar2	Work Type GUID		

PricelistByGUID: Visma Severa Price List by GUID

Catalog: Severa

Schema: API

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Label: Price List by GUID

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

Filters on API Server: pricelistGuid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function PricelistByGUID. 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
pricelistGuid		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CURRENCYCODE	varchar2	Currency Code		
CURRENCYGUID	varchar2	Currency GUID		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
ISINTERNAL	boolean	Is Internal		
ISVOLUMEPRICING	boolean	Is Volume Pricing		
NAME	varchar2	Name		

Pricelists: Visma Severa Price Lists

Catalog: Severa

Schema: API

Label: Price Lists

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Table Columns

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

Name	Data Type	Label	Required	Documentation
CURRENCYCODE	varchar2	Currency Code		
CURRENCYGUID	varchar2	Currency GUID		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
ISINTERNAL	boolean	Is Internal		
ISVOLUMEPRICING	boolean	Is Volume Pricing		
NAME	varchar2	Name		

ProductByCode: Visma Severa Product by Code

Catalog: Severa

Schema: API

Label: Product by Code

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

Filters on API Server: productCode

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ProductByCode. 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
productCode		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CODE	varchar2	Code		
DESCRIPTION	varchar2	Ddescription		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
MEASUREMENTUNIT	varchar2	Measurement Unit		
NAME	varchar2	Name		
PRODUCTCATEGORYGUID	varchar2	Product Category GUID		
SALESACCOUNTGUID	varchar2	Sales Account GUID		
TYPE	varchar2	Type		
UNITCOST	number	Unit Cost		
UNITPRICE	number	Unit Price		
USEINWORKTIMEENTRY	boolean	Use in Work Time Entry		
VAT	number	VAT		

ProductByCodeIncludeInactive: Visma Severa Product by Code Include Inactive

Catalog: Severa

Schema: API

Label: Product by Code Include Inactive

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

Filters on API Server: productCode

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ProductByCodeIncludeInactive. 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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
productCode		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CODE	varchar2	Code		
DESCRIPTION	varchar2	Ddescription		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
MEASUREMENTUNIT	varchar2	Measurement Unit		
NAME	varchar2	Name		
PRODUCTCATEGORYGUID	varchar2	Product Category GUID		
SALESACCOUNTGUID	varchar2	Sales Account GUID		
TYPE	varchar2	Type		
UNITCOST	number	Unit Cost		
UNITPRICE	number	Unit Price		
USEINWORKTIMEENTRY	boolean	Use in Work Time Entry		
VAT	number	VAT		

ProductByGUID: Visma Severa Product by GUID

Catalog: Severa

Schema: API

Label: Product by GUID

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

Filters on API Server: productGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ProductByGUID. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
productGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CODE	varchar2	Code		
DESCRIPTION	varchar2	Ddescription		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
MEASUREMENTUNIT	varchar2	Measurement Unit		
NAME	varchar2	Name		
PRODUCTCATEGORYGUID	varchar2	Product Category GUID		
SALESACCOUNTGUID	varchar2	Sales Account GUID		
TYPE	varchar2	Type		
UNITCOST	number	Unit Cost		
UNITPRICE	number	Unit Price		
USEINWORKTIMEENTRY	boolean	Use in Work Time Entry		
VAT	number	VAT		

ProductCategoriesChangedSince: Visma Severa Product Categories Changed since

Catalog: Severa

Schema: API

Label: Product Categories Changed since

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

Filters on API Server: startDateTime- options

Parameters of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The following parameters can be used to control the behaviour of the table function ProductCategoriesChangedSince. 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
options		☐		
startDateTime		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CODE	varchar2	Code		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
NAME	varchar2	Name		

ProductCategoryByCode: Visma Severa Product Category by Code

Catalog: Severa

Schema: API

Label: Product Category by Code

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

Filters on API Server: productCategoryCode

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ProductCategoryByCode. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
productCategoryCode		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CODE	varchar2	Code		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
NAME	varchar2	Name		

ProductCategoryByGUID: Visma Severa Product Category by GUID

Catalog: Severa

Schema: API

Label: Product Category by GUID

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

Filters on API Server: productCategoryGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ProductCategoryByGUID. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
productCategoryGUID		□		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CODE	varchar2	Code		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
NAME	varchar2	Name		

ProductCostAccountAssociationsByGUID: Visma Severa Product Cost Account Associations by GUID

Catalog: Severa

Schema: API

Label: Product Cost Account Associations by GUID

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

Filters on API Server: productGUID- includeDefault

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ProductCostAccountAssociationsByGUID. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
includeDefault		☐		
productGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
ISTRAVELTYPEDEFAULT	boolean	Is Travel Type Default		
NAME	varchar2	Name		
NUMBER	varchar2	Number		
TAXPERCENTAGE	number	Tax (%)		

ProductsChangedSince: Visma Severa Products Changed since

Catalog: Severa

Schema: API

Label: Products Changed since

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

Filters on API Server: startDateTime- options

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ProductsChangedSince. 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
options		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
startDateTime		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CODE	varchar2	Code		
DESCRIPTION	varchar2	Ddescription		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
MEASUREMENTUNIT	varchar2	Measurement Unit		
NAME	varchar2	Name		
PRODUCTCATEGORYGUID	varchar2	Product Category GUID		
SALESACCOUNTGUID	varchar2	Sales Account GUID		
TYPE	varchar2	Type		
UNITCOST	number	Unit Cost		
UNITPRICE	number	Unit Price		
USEINWORKTIMEENTRY	boolean	Use in Work Time Entry		
VAT	number	VAT		

ReimbursedHourEntriesByDate: Visma Severa Reimbursed Hour Entries by Date

Catalog: Severa

Schema: API

Label: Reimbursed Hour Entries by Date

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

Filters on API Server: businessUnitGUID- startdate- enddate- inCreditNote

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ReimbursedHourEntriesByDate. 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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
businessUnitGUID		☐		
enddate		☐		
inCreditNote		☐		
startdate		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	varchar2	Account GUID		
CASEBUSINESSUNITCODE	varchar2	Project Business Unit Code		
CASECOSTCENTERIDENTIFIER	varchar2	Project Cost Centre Identifier		
CASEGUID	varchar2	Project GUID		
CASENAME	varchar2	Project Name		
CASENUMBER	number	Project Number		
COSTCURRENCYSHORTFORM	varchar2	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Cost Currency Short Form of Organisation Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Cost Currency Short Form of User Company Currency		
DESCRIPTION	varchar2	Ddescription		
ENDTIME	datetime	End Time		
EVENTDATE	datetime	Event Date		
GUID	varchar2	GUID		
INVOICEDESCRIPTION	varchar2	Invoice Description		
INVOICEGUID	varchar2	Invoice GUID		
INVOICEQUANTITY	number	Invoice Quantity		
INVOICEROWGUID	varchar2	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
ISBILLABLE	boolean	Is Billable		
ISPRODUCTIVE	boolean	Is Productive		
ISREADONLY	boolean	Is Read-only		
OVERTIMECODE	varchar2	Overtime Code		
OVERTIMEGUID	varchar2	Overtime GUID		
PHASEGUID	varchar2	Phase GUID		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	varchar2	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Price Currency Short Form of Organisation Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Price Currency Short Form of User Company Currency		
QUANTITY	number	Quantity		
STARTTIME	datetime	Start Time		
TASKGUID	varchar2	Task GUID		
UNITCOST	number	Unit Cost		
UNITCOSTININVOICINGCURRENCY	number	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	number	Unit Cost in Organisation Currency		
UNITCOSTINUSERCOMPANYCURRENCY	number	Unit Cost in User Company Currency		
UNITPRICE	number	Unit Price		
UNITPRICEINBASECURRENCY	number	Unit Price in Base Currency		
UNITPRICEINCASECOMPANYCURRENCY	number	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	number	Unit Price in User Company Currency		
USEINVOICEQUANTITY	boolean			
USERBUSINESSUNITCODE	varchar2	User Business Unit Code		
USERCODE	varchar2	User Code		
USERFIRSTNAME	varchar2	User First Name		
USERGUID	varchar2	User GUID		
USERLASTNAME	varchar2	User Last Name		
WORKTYPECODE	varchar2	Work Type Code		
WORKTYPEGUID	varchar2	Work Type GUID		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

ReimbursedHourEntriesChangedSince: Visma Severa Reimbursed Hour Entries Changed since

Catalog: Severa

Schema: API

Label: Reimbursed Hour Entries Changed since

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

Filters on API Server: startDate- inCreditNote

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ReimbursedHourEntriesChangedSince. 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
inCreditNote		☐		
startDate		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	varchar2	Account GUID		
CASEBUSINESSUNITCODE	varchar2	Project Business Unit Code		
CASECOSTCENTERIDENTIFIER	varchar2	Project Cost Centre Identifier		
CASEGUID	varchar2	Project GUID		
CASENAME	varchar2	Project Name		
CASENUMBER	number	Project Number		
COSTCURRENCYSHORTFORM	varchar2	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Cost Currency Short Form of Invoicing Currency		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Cost Currency Short Form of Organisation Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Cost Currency Short Form of User Company Currency		
DESCRIPTION	varchar2	Ddescription		
ENDTIME	datetime	End Time		
EVENTDATE	datetime	Event Date		
GUID	varchar2	GUID		
INVOICEDESCRIPTION	varchar2	Invoice Description		
INVOICEGUID	varchar2	Invoice GUID		
INVOICEQUANTITY	number	Invoice Quantity		
INVOICEROWGUID	varchar2	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISPRODUCTIVE	boolean	Is Productive		
ISREADONLY	boolean	Is Read-only		
OVERTIMECODE	varchar2	Overtime Code		
OVERTIMEGUID	varchar2	Overtime GUID		
PHASEGUID	varchar2	Phase GUID		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	varchar2	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Price Currency Short Form of Organisation Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Price Currency Short Form of User Company Currency		
QUANTITY	number	Quantity		
STARTTIME	datetime	Start Time		
TASKGUID	varchar2	Task GUID		
UNITCOST	number	Unit Cost		
UNITCOSTININVOICINGCURRENCY	number	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	number	Unit Cost in Organisation Currency		
UNITCOSTINUSERCOMPANYCURRENCY	number	Unit Cost in User Company Currency		
UNITPRICE	number	Unit Price		
UNITPRICEINBASECURRENCY	number	Unit Price in Base Currency		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
UNITPRICEINCASECOMPANYCURRENCY	number	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	number	Unit Price in User Company Currency		
USEINVOICEQUANTITY	boolean			
USERBUSINESSUNITCODE	varchar2	User Business Unit Code		
USERCODE	varchar2	User Code		
USERFIRSTNAME	varchar2	User First Name		
USERGUID	varchar2	User GUID		
USERLASTNAME	varchar2	User Last Name		
WORKTYPECODE	varchar2	Work Type Code		
WORKTYPEGUID	varchar2	Work Type GUID		

ReimbursedHourEntryByGUID: Visma Severa Reimbursed Hour Entry by GUID

Catalog: Severa

Schema: API

Label: Reimbursed Hour Entry by GUID

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

Filters on API Server: hourEntryGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ReimbursedHourEntryByGUID. 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
hourEntryGUID		☐		

Columns of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	varchar2	Account GUID		
CASEBUSINESSUNITCODE	varchar2	Project Business Unit Code		
CASECOSTCENTERIDENTIFIER	varchar2	Project Cost Centre Identifier		
CASEGUID	varchar2	Project GUID		
CASENAME	varchar2	Project Name		
CASENUMBER	number	Project Number		
COSTCURRENCYSHORTFORM	varchar2	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Cost Currency Short Form of Organisation Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Cost Currency Short Form of User Company Currency		
DESCRIPTION	varchar2	Ddescription		
ENDTIME	datetime	End Time		
EVENTDATE	datetime	Event Date		
GUID	varchar2	GUID		
INVOICEDESCRIPTION	varchar2	Invoice Description		
INVOICEGUID	varchar2	Invoice GUID		
INVOICEQUANTITY	number	Invoice Quantity		
INVOICEROWGUID	varchar2	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISPRODUCTIVE	boolean	Is Productive		
ISREADONLY	boolean	Is Read-only		
OVERTIMECODE	varchar2	Overtime Code		
OVERTIMEGUID	varchar2	Overtime GUID		
PHASEGUID	varchar2	Phase GUID		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	varchar2	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Price Currency Short Form of Organisation Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Price Currency Short Form of User		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
		Company Currency		
QUANTITY	number	Quantity		
STARTTIME	datetime	Start Time		
TASKGUID	varchar2	Task GUID		
UNITCOST	number	Unit Cost		
UNITCOSTININVOICINGCURRENCY	number	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	number	Unit Cost in Organisation Currency		
UNITCOSTINUSERCOMPANYCURRENCY	number	Unit Cost in User Company Currency		
UNITPRICE	number	Unit Price		
UNITPRICEINBASECURRENCY	number	Unit Price in Base Currency		
UNITPRICEINCASECOMPANYCURRENCY	number	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	number	Unit Price in User Company Currency		
USEINVOICEQUANTITY	boolean			
USERBUSINESSUNITCODE	varchar2	User Business Unit Code		
USERCODE	varchar2	User Code		
USERFIRSTNAME	varchar2	User First Name		
USERGUID	varchar2	User GUID		
USERLASTNAME	varchar2	User Last Name		
WORKTYPECODE	varchar2	Work Type Code		
WORKTYPEGUID	varchar2	Work Type GUID		

ReimbursedItemByGUID: Visma Severa Reimbursed Item by GUID

Catalog: Severa

Schema: API

Label: Reimbursed Item by GUID

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

Filters on API Server: itemGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ReimbursedItemByGUID. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
itemGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
BILLINGDEPENDENCYTASKGUID	varchar2	Billing Dependency Task GUID		
BILLINGSCHEDULE	varchar2	Billing Schedule		
CASEGUID	varchar2	Project GUID		
CASENUMBER	number	Project Number		
COSTCENTERGUID	varchar2	Cost Centre GUID		
COSTCENTERIDENTIFIER	varchar2	Cost Centre Identifier		
COSTCURRENCYSHORTFORM	varchar2	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Cost Currency Short Form of Organisation Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Cost Currency Short Form of User Company Currency		
DESCRIPTION	varchar2	Ddescription		
DISPLAYPERIODSTART	datetime	Display Period Start		
EVENTDATE	datetime	Event Date		
GUID	varchar2	GUID		
INVOICEDESCRIPTION	varchar2	Invoice Description		
INVOICEGUID	varchar2	Invoice GUID		
INVOICEQUANTITY	number	Invoice Quantity		
INVOICEROWGUID	varchar2	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISOUTSOURCINGCOST	boolean	Is Out-sourcing Cost		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
ISREADONLY	boolean	Is Read-only		
ISREALISED	boolean	Is Realized		
MEASUREMENTUNIT	varchar2	Measurement Unit		
NAME	varchar2	Name		
ORDERDATE	datetime	Order Date		
PLANNEDBILLINGDATE	datetime	Planned Billing Date		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	varchar2	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Price Currency Short Form of Organisation Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Price Currency Short Form of User Company Currency		
PRODUCTCODE	varchar2	Product Code		
PRODUCTGUID	varchar2	Product GUID		
PRODUCTTYPE	varchar2	Product Type		
PURCHASEORDERNUMBER	varchar2	Purchase Order Number		
PURCHASEVAT	number	Purchase VAT		
QUANTITY	number	Quantity		
RECURRENCEENDDATE	datetime	Recurrence End Date		
RECURRENCEENDTYPE	number	Recurrence End Type		
RECURRENCEFREQUENCY	number	Recurrence Frequency		
RECURRENCESTARTDATE	datetime	Recurrence Start Date		
RECURRENCETIMES	number	Recurrence Times		
RECURRINGITEMGUID	varchar2	Recurring Item GUID		
SALESACCOUNTGUID	varchar2	Sales Account GUID		
SALESSTATUSGUID	varchar2	Sales Status GUID		
TASKGUID	varchar2	Task GUID		
TRAVELENDTIME	datetime	Travel End Time		
TRAVELSTARTTIME	datetime	Travel Start Time		
UNITCOST	number	Unit Cost		
UNITCOSTININVOICINGCURRENCY	number	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	number	Unit Cost in Organisation Currency		
UNITCOSTINUSERCOMPANYCURRENCY	number	Unit Cost in User Company Currency		
UNITCOSTVATINCLUDEDAMOUNT	number	Unit Cost VAT Included Amount		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
UNITPRICE	number	Unit Price		
UNITPRICEINBASECURRENCY	number	Unit Price in Base Currency		
UNITPRICEINCASECOMPANYCURRENCY	number	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	number	Unit Price in User Company Currency		
USERCODE	varchar2	User Code		
USERGUID	varchar2	User GUID		
VAT	number	VAT		

ReimbursedItemsChangedSince: Visma Severa Reimbursed Items Changed since

Catalog: Severa

Schema: API

Label: Reimbursed Items Changed since

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

Filters on API Server: startDate- inCreditNote

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ReimbursedItemsChangedSince. 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
inCreditNote		☐		
startDate		☐		

Columns of Table Function

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
BILLINGDEPENDENCYTASKGUID	varchar2	Billing Dependency Task GUID		
BILLINGSCHEDULE	varchar2	Billing Schedule		
CASEGUID	varchar2	Project GUID		
CASENUMBER	number	Project Number		
COSTCENTERGUID	varchar2	Cost Centre GUID		
COSTCENTERIDENTIFIER	varchar2	Cost Centre Identifier		
COSTCURRENCYSHORTFORM	varchar2	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Cost Currency Short Form of Organisation Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Cost Currency Short Form of User Company Currency		
DESCRIPTION	varchar2	Ddescription		
DISPLAYPERIODSTART	datetime	Display Period Start		
EVENTDATE	datetime	Event Date		
GUID	varchar2	GUID		
INVOICEDESCRIPTION	varchar2	Invoice Description		
INVOICEGUID	varchar2	Invoice GUID		
INVOICEQUANTITY	number	Invoice Quantity		
INVOICEROWGUID	varchar2	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISOUTSOURCINGCOST	boolean	Is Out-sourcing Cost		
ISREADONLY	boolean	Is Read-only		
ISREALISED	boolean	Is Realized		
MEASUREMENTUNIT	varchar2	Measurement Unit		
NAME	varchar2	Name		
ORDERDATE	datetime	Order Date		
PLANNEDBILLINGDATE	datetime	Planned Billing Date		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	varchar2	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Price Currency Short Form of Organisation Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Price Currency Short Form of User Company Currency		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
PRODUCTCODE	varchar2	Product Code		
PRODUCTGUID	varchar2	Product GUID		
PRODUCTTYPE	varchar2	Product Type		
PURCHASEORDERNUMBER	varchar2	Purchase Order Number		
PURCHASEVAT	number	Purchase VAT		
QUANTITY	number	Quantity		
RECURRENCEENDDATE	datetime	Recurrence End Date		
RECURRENCEENDTYPE	number	Recurrence End Type		
RECURRENCEFREQUENCY	number	Recurrence Frequency		
RECURRENCESTARTDATE	datetime	Recurrence Start Date		
RECURRENCETIMES	number	Recurrence Times		
RECURRINGITEMGUID	varchar2	Recurring Item GUID		
SALESACCOUNTGUID	varchar2	Sales Account GUID		
SALESSTATUSGUID	varchar2	Sales Status GUID		
TASKGUID	varchar2	Task GUID		
TRAVELENDTIME	datetime	Travel End Time		
TRAVELSTARTTIME	datetime	Travel Start Time		
UNITCOST	number	Unit Cost		
UNITCOSTININVOICINGCURRENCY	number	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	number	Unit Cost in Organisation Currency		
UNITCOSTINUSERCOMPANYCURRENCY	number	Unit Cost in User Company Currency		
UNITCOSTVATINCLUDEDAMOUNT	number	Unit Cost VAT Included Amount		
UNITPRICE	number	Unit Price		
UNITPRICEINBASECURRENCY	number	Unit Price in Base Currency		
UNITPRICEINCASECOMPANYCURRENCY	number	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	number	Unit Price in User Company Currency		
USERCODE	varchar2	User Code		
USERGUID	varchar2	User GUID		
VAT	number	VAT		

RemoveAccountGroupMembers: Visma Severa Remove Account Group Members

Catalog: Severa

Schema: API

Label: Remove Account Group Members

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Filters on API Server: accountGUID- accountGroupGUIDs

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function RemoveAccountGroupMembers. 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
accountGroupGUIDs		☐		
accountGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
DUMMY	varchar2	Dummy	☑	Dummy column

RemoveActivityContactParticipant: Visma Severa Remove Activity Contact Participant

Catalog: Severa

Schema: API

Label: Remove Activity Contact Participant

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

Filters on API Server: activityGUID- contactGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function RemoveActivityContactParticipant. A value must be provided at all times for required

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
activityGUID		☐		
contactGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
REMOVEACTIVITYCONTACTPARTICIPANTRESULT	boolean			

RemoveActivityResourceParticipant: Visma Severa Remove Activity Resource Participant

Catalog: Severa

Schema: API

Label: Remove Activity Resource Participant

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

Filters on API Server: activityGUID- resourceGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function RemoveActivityResourceParticipant. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
activityGUID		☐		
resourceGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
REMOVEACTIVITYRESOURCEPARTICIPANTRESULT	boolean			

RemoveActivityUserParticipant: Visma Severa Remove Activity User Participant

Catalog: Severa

Schema: API

Label: Remove Activity User Participant

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

Filters on API Server: activityGUID- userGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function RemoveActivityUserParticipant. 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
activityGUID		☐		
userGUID		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
REMOVEACTIVITYUSERPARTICIPANTRESULT	boolean			

ResourceAllocationByGUID: Visma Severa Resource Allocations by GUID

Catalog: Severa

Schema: API

Label: Resource Allocations by GUID

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

Filters on API Server: resourceAllocationGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ResourceAllocationByGUID. 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
resourceAllocationGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ALLOCATEDHOURS	number	Allocated Hours		
ALLOCATEDPERCENTAGE	number	Allocated Percentage		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
CASEGUID	varchar2	Project GUID		
ENDDATE	datetime	End Date		
GUID	varchar2	GUID		
PHASEGUID	varchar2	Phase GUID		
STARTDATE	datetime	Start Date		
USERGUID	varchar2	User GUID		

ResourceAllocations: Visma Severa Resource Allocations

Catalog: Severa

Schema: API

Label: Resource Allocations

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

Filters on API Server: caseGUIDs- userGUIDs- businessUnitGUIDs- supervisorUserGUIDs- caseStatusGUIDs- salesProcessGUIDs- caseTagGUIDs- userTagGUIDs- startDate- endDate

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ResourceAllocations. 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
businessUnitGUIDs		☐		
caseGUIDs		☐		
caseStatusGUIDs		☐		
caseTagGUIDs		☐		
endDate		☐		
salesProcessGUIDs		☐		
startDate		☐		
supervisorUserGUIDs		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
userGUIDs		☐		
userTagGUIDs		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ALLOCATEDHOURS	number	Allocated Hours		
ALLOCATEDPERCENTAGE	number	Allocated Percentage		
CASEGUID	varchar2	Project GUID		
ENDDATE	datetime	End Date		
GUID	varchar2	GUID		
PHASEGUID	varchar2	Phase GUID		
STARTDATE	datetime	Start Date		
USERGUID	varchar2	User GUID		

ResourceAllocationsByCaseGUID: Visma Severa Resource Allocations by Project GUID

Catalog: Severa

Schema: API

Label: Resource Allocations by Project GUID

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

Filters on API Server: caseGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ResourceAllocationsByCaseGUID. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
caseGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ALLOCATEDHOURS	number	Allocated Hours		
ALLOCATEDPERCENTAGE	number	Allocated Percentage		
CASEGUID	varchar2	Project GUID		
ENDDATE	datetime	End Date		
GUID	varchar2	GUID		
PHASEGUID	varchar2	Phase GUID		
STARTDATE	datetime	Start Date		
USERGUID	varchar2	User GUID		

ResourceAllocationsByCaseGUIDUserGUID: Visma Severa Resource Allocations by Project GUID and User GUID

Catalog: Severa

Schema: API

Label: Resource Allocations by Project GUID and User GUID

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

Filters on API Server: caseGUID- userGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ResourceAllocationsByCaseGUIDUserGUID. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
caseGUID		☐		
userGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ALLOCATEDHOURS	number	Allocated Hours		
ALLOCATEDPERCENTAGE	number	Allocated Percentage		
CASEGUID	varchar2	Project GUID		
ENDDATE	datetime	End Date		
GUID	varchar2	GUID		
PHASEGUID	varchar2	Phase GUID		
STARTDATE	datetime	Start Date		
USERGUID	varchar2	User GUID		

ResourceAllocationsByUserGUID: Visma Severa Resource Allocations by User GUID

Catalog: Severa

Schema: API

Label: Resource Allocations by User GUID

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

Filters on API Server: userGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ResourceAllocationsByUserGUID. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
userGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ALLOCATEDHOURS	number	Allocated Hours		
ALLOCATEDPERCENTAGE	number	Allocated Percentage		
CASEGUID	varchar2	Project GUID		
ENDDATE	datetime	End Date		
GUID	varchar2	GUID		
PHASEGUID	varchar2	Phase GUID		
STARTDATE	datetime	Start Date		
USERGUID	varchar2	User GUID		

ResourceAllocationsChangedSince: Visma Severa Resource Allocations Changed since

Catalog: Severa

Schema: API

Label: Resource Allocations Changed since

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

Filters on API Server: startDateTime

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ResourceAllocationsChangedSince. 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
startDateTime		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ALLOCATEDHOURS	number	Allocated Hours		
ALLOCATEDPERCENTAGE	number	Allocated Percentage		
CASEGUID	varchar2	Project GUID		
ENDDATE	datetime	End Date		
GUID	varchar2	GUID		
PHASEGUID	varchar2	Phase GUID		
STARTDATE	datetime	Start Date		
USERGUID	varchar2	User GUID		

ResourceByGUID: Visma Severa Resource by GUID

Catalog: Severa

Schema: API

Label: Resource by GUID

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

Filters on API Server: resourceGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ResourceByGUID. 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
resourceGUID		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
NAME	varchar2	Name		

Resources: Visma Severa Resources

Catalog: Severa

Schema: API

Label: Resources

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

Table Columns

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

Name	Data Type	Label	Required	Documentation
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
NAME	varchar2	Name		

SalesProcessByGUID: Visma Severa Sales Process by GUID

Catalog: Severa

Schema: API

Label: Sales Process by GUID

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

Filters on API Server: salesProcessGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function SalesProcessByGUID. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
salesProcessGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
DEFAULTPROBABILITY	number	Default Probability		
GUID	varchar2	GUID		
ICON	varchar2	Icon		
ISACTIVE	boolean	Is Active		
ISALSOITEMSTATUS	boolean	Is also Item Status		
ISINPROGRESS	boolean	Is in Progress		
ISLOST	boolean	Is Lost		
ISOFFER	boolean	Is Offer		
ISWON	boolean	Is Won		
NAME	varchar2	Name		

SalesProcesses: Visma Severa Sales Processes

Catalog: Severa

Schema: API

Label: Sales Processes

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

Table Columns

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

Name	Data Type	Label	Required	Documentation
DEFAULTPROBABILITY	number	Default Probability		
GUID	varchar2	GUID		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
ICON	varchar2	Icon		
ISACTIVE	boolean	Is Active		
ISALSOITEMSTATUS	boolean	Is also Item Status		
ISINPROGRESS	boolean	Is in Progress		
ISLOST	boolean	Is Lost		
ISOFFER	boolean	Is Offer		
ISWON	boolean	Is Won		
NAME	varchar2	Name		

SalesStatusByGUID: Visma Severa Sales Status by GUID

Catalog: Severa

Schema: API

Label: Sales Status by GUID

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

Filters on API Server: salesStatusGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function SalesStatusByGUID. 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
salesStatusGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CASEGUID	varchar2	Project GUID		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
GUID	varchar2	GUID		
SALESPROCESSGUID	varchar2	Sales Process GUID		
TIMESTAMP	datetime	Time Stamp		

SalesStatusesForCase: Visma Severa Sales Statuses for Project

Catalog: Severa

Schema: API

Label: Sales Statuses for Project

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

Filters on API Server: caseGuid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function SalesStatusesForCase. 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
caseGuid		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CASEGUID	varchar2	Project GUID		
GUID	varchar2	GUID		
SALESPROCESSGUID	varchar2	Sales Process GUID		
TIMESTAMP	datetime	Time Stamp		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Server: Visma Severa Server

Catalog: Severa

Schema: API

Label: Server

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

Table Columns

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

Name	Data Type	Label	Required	Documentation
SERVERRESULT	varchar2			

SetInvoiceNumber: Visma Severa Set Invoice Number

Catalog: Severa

Schema: API

Label: Set Invoice Number

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

Filters on API Server: invoiceGUID- invoiceNumber- referenceNumber- status

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function SetInvoiceNumber. 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
invoiceGUID		☐		
invoiceNumber		☐		
referenceNumber		☐		
status		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
DUMMY	varchar2	Dummy	☑	Dummy column

SetInvoicePaymentDate: Visma Severa Set Invoice Payment Date

Catalog: Severa

Schema: API

Label: Set Invoice Payment Date

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

Filters on API Server: invoiceGUID- paymentDate

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function SetInvoicePaymentDate. 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
invoiceGUID		☐		
paymentDate		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
DUMMY	varchar2	Dummy	☑	Dummy column

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

SetInvoiceStatus: Visma Severa Set Invoice Status

Catalog: Severa

Schema: API

Label: Set Invoice Status

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

Filters on API Server: invoiceGUID- status

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function SetInvoiceStatus. 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
invoiceGUID		☐		
status		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
DUMMY	varchar2	Dummy	☑	Dummy column

SetPhaseMemberListForPhase: Visma Severa Set Phase Member List for Phase

Catalog: Severa

Schema: API

Label: Set Phase Member List for Phase

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

Filters on API Server: phaseGUID- userGUIDs

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function SetPhaseMemberListForPhase. 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
phaseGUID		☐		
userGUIDs		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
SETPHASEMEMBERLISTFORPHASERESULT	boolean			

SetTravelReimbursementStatus: Visma Severa Set Travel Reimbursement Status

Catalog: Severa

Schema: API

Label: Set Travel Reimbursement Status

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

Filters on API Server: travelReimbursementGUID- status

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function SetTravelReimbursementStatus. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
status		☐		
travelReimbursementGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
DUMMY	varchar2	Dummy	☑	Dummy column

TagByGUID: Visma Severa Tag by GUID

Catalog: Severa

Schema: API

Label: Tag by GUID

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

Filters on API Server: tagGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function TagByGUID. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
tagGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CONTEXT	varchar2	Context		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
KEYWORD	varchar2	Keyword		
WEIGHT	number	Weight		

TagsByCaseGUID: Visma Severa Tags by Project GUID

Catalog: Severa

Schema: API

Label: Tags by Project GUID

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

Filters on API Server: caseGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function TagsByCaseGUID. 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
caseGUID		☐		

Columns of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Name	Data Type	Label	Required	Documentation
CONTEXT	varchar2	Context		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
KEYWORD	varchar2	Keyw ord		
WEIGHT	number	Weight		

TimeEntriesByDate: Visma Severa Time Entries by Date

Catalog: Severa

Schema: API

Label: Time Entries by Date

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

Filters on API Server: eventDate

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function TimeEntriesByDate. 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
eventDate		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
DESCRIPTION	varchar2	Ddescription		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
ENDTIME	datetime	End Time		
EVENTDATE	datetime	Event Date		
GUID	varchar2	GUID		
PHASEGUID	varchar2	Phase GUID		
QUANTITY	number	Quantity		
STARTTIME	datetime	Start Time		
USERGUID	varchar2	User GUID		

TimeEntriesByDateAndUserGUID: Visma Severa Time Entries by Date and User GUID

Catalog: Severa

Schema: API

Label: Time Entries by Date and User GUID

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

Filters on API Server: eventDate- userGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function TimeEntriesByDateAndUserGUID. 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
eventDate		☐		
userGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
DESCRIPTION	varchar2	Ddescription		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
ENDTIME	datetime	End Time		
EVENTDATE	datetime	Event Date		
GUID	varchar2	GUID		
PHASEGUID	varchar2	Phase GUID		
QUANTITY	number	Quantity		
STARTTIME	datetime	Start Time		
USERGUID	varchar2	User GUID		

TimeEntriesChangedSince: Visma Severa Time Entries Changed since

Catalog: Severa

Schema: API

Label: Time Entries Changed since

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

Filters on API Server: startDateTime

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function TimeEntriesChangedSince. 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
startDateTime		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
DESCRIPTION	varchar2	Ddescription		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
ENDTIME	datetime	End Time		
EVENTDATE	datetime	Event Date		
GUID	varchar2	GUID		
PHASEGUID	varchar2	Phase GUID		
QUANTITY	number	Quantity		
STARTTIME	datetime	Start Time		
USERGUID	varchar2	User GUID		

TimezoneByGUID: Visma Severa Timezone by GUID

Catalog: Severa

Schema: API

Label: Timezone by GUID

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

Filters on API Server: timezoneGuid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function TimezoneByGUID. 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
timezoneGuid		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
DIFFERENCETOGMT	number	Difference to GMT		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
GUID	varchar2	GUID		
NAME	varchar2	Name		
SYSTEMNAME	varchar2	System Name		

Timezones: Visma Severa Timezones

Catalog: Severa

Schema: API

Label: Timezones

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

Table Columns

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

Name	Data Type	Label	Required	Documentation
DIFFERENCETOGMT	number	Difference to GMT		
GUID	varchar2	GUID		
NAME	varchar2	Name		
SYSTEMNAME	varchar2	System Name		

TravelExpenseByCode: Visma Severa Travel Expense by Code

Catalog: Severa

Schema: API

Label: Travel Expense by Code

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

Filters on API Server: travelExpenseCode

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function TravelExpenseByCode. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
travelExpenseCode		□		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CODE	varchar2	Code		
COSTCURRENCYGUID	varchar2	Cost Currency GUID		
DESCRIPTION	varchar2	Ddescription		
EXPENSECLASS	varchar2	Expense Class		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
MEASUREMENTUNIT	varchar2	Measurement Unit		
NAME	varchar2	Name		
TYPE	varchar2	Type		
UNITCOST	number	Unit Cost		
UNITPRICE	number	Unit Price		
USEINWORKTIMEENTRY	boolean	Use in Work Time Entry		
VAT	number	VAT		

TravelExpenseByCodeIncludeInactive: Visma Severa Travel Expense by Code Include Inactive

Catalog: Severa

Schema: API

Label: Travel Expense by Code Include Inactive

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

Filters on API Server: travelExpenseCode

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function TravelExpenseByCodeIncludeInactive. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
travelExpenseCode		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CODE	varchar2	Code		
COSTCURRENCYGUID	varchar2	Cost Currency GUID		
DESCRIPTION	varchar2	Ddescription		
EXPENSECLASS	varchar2	Expense Class		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
MEASUREMENTUNIT	varchar2	Measurement Unit		
NAME	varchar2	Name		
TYPE	varchar2	Type		
UNITCOST	number	Unit Cost		
UNITPRICE	number	Unit Price		
USEINWORKTIMEENTRY	boolean	Use in Work Time Entry		
VAT	number	VAT		

TravelExpenseByGUID: Visma Severa Travel Expense by GUID

Catalog: Severa

Schema: API

Label: Travel Expense by GUID

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

Filters on API Server: travelExpenseGUID

Parameters of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The following parameters can be used to control the behaviour of the table function TravelExpenseByGUID. 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
travelExpenseGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CODE	varchar2	Code		
COSTCURRENCYGUID	varchar2	Cost Currency GUID		
DESCRIPTION	varchar2	Ddescription		
EXPENSECLASS	varchar2	Expense Class		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
MEASUREMENTUNIT	varchar2	Measurement Unit		
NAME	varchar2	Name		
TYPE	varchar2	Type		
UNITCOST	number	Unit Cost		
UNITPRICE	number	Unit Price		
USEINWORKTIMEENTRY	boolean	Use in Work Time Entry		
VAT	number	VAT		

TravelExpensesByDate: Visma Severa Travel Expenses by Date

Catalog: Severa

Schema: API

Label: Travel Expenses by Date

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Filters on API Server: businessUnitGUID- startDate- endDate

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function TravelExpensesByDate. 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
businessUnitGUID		☐		
endDate		☐		
startDate		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
BILLINGDEPENDENCYTASKGUID	varchar2	Billing Dependency Task GUID		
BILLINGSCHEDULE	varchar2	Billing Schedule		
CASEGUID	varchar2	Project GUID		
CASENUMBER	number	Project Number		
COSTCENTERGUID	varchar2	Cost Centre GUID		
COSTCENTERIDENTIFIER	varchar2	Cost Centre Identifier		
COSTCURRENCYSHORTFORM	varchar2	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Cost Currency Short Form of Organisation Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Cost Currency Short Form of User		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
		Company Currency		
DESCRIPTION	varchar2	Ddescription		
DISPLAYPERIODSTART	datetime	Display Period Start		
EVENTDATE	datetime	Event Date		
GUID	varchar2	GUID		
INVOICEDESCRIPTION	varchar2	Invoice Description		
INVOICEGUID	varchar2	Invoice GUID		
INVOICEQUANTITY	number	Invoice Quantity		
INVOICEROWGUID	varchar2	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISOUTSOURCINGCOST	boolean	Is Out-sourcing Cost		
ISREADONLY	boolean	Is Read-only		
ISREALISED	boolean	Is Realized		
MEASUREMENTUNIT	varchar2	Measurement Unit		
NAME	varchar2	Name		
ORDERDATE	datetime	Order Date		
PLANNEDBILLINGDATE	datetime	Planned Billing Date		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	varchar2	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Price Currency Short Form of Organisation Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Price Currency Short Form of User Company Currency		
PRODUCTCODE	varchar2	Product Code		
PRODUCTGUID	varchar2	Product GUID		
PRODUCTTYPE	varchar2	Product Type		
PURCHASEORDERNUMBER	varchar2	Purchase Order Number		
PURCHASEVAT	number	Purchase VAT		
QUANTITY	number	Quantity		
RECURRENCEENDDATE	datetime	Recurrence End Date		
RECURRENCEENDTYPE	number	Recurrence End Type		
RECURRENCEFREQUENCY	number	Recurrence Frequency		
RECURRENCESTARTDATE	datetime	Recurrence Start Date		
RECURRENCETIMES	number	Recurrence Times		
RECURRINGITEMGUID	varchar2	Recurring Item GUID		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
SALESACCOUNTGUID	varchar2	Sales Account GUID		
SALESSTATUSGUID	varchar2	Sales Status GUID		
TASKGUID	varchar2	Task GUID		
TRAVELENDTIME	datetime	Travel End Time		
TRAVELSTARTTIME	datetime	Travel Start Time		
UNITCOST	number	Unit Cost		
UNITCOSTININVOICINGCURRENCY	number	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	number	Unit Cost in Organisation Currency		
UNITCOSTINUSERCOMPANYCURRENCY	number	Unit Cost in User Company Currency		
UNITCOSTVATINCLUDEDAMOUNT	number	Unit Cost VAT Included Amount		
UNITPRICE	number	Unit Price		
UNITPRICEINBASECURRENCY	number	Unit Price in Base Currency		
UNITPRICEINCASECOMPANYCURRENCY	number	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	number	Unit Price in User Company Currency		
USERCODE	varchar2	User Code		
USERGUID	varchar2	User GUID		
VAT	number	VAT		

TravelExpensesChangedSince: Visma Severa Travel Expenses Changed since

Catalog: Severa

Schema: API

Label: Travel Expenses Changed since

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

Filters on API Server: startDateTime- options

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function TravelExpensesChangedSince. 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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
options		☐		
startDateTime		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CODE	varchar2	Code		
COSTCURRENCYGUID	varchar2	Cost Currency GUID		
DESCRIPTION	varchar2	Ddescription		
EXPENSECLASS	varchar2	Expense Class		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
MEASUREMENTUNIT	varchar2	Measurement Unit		
NAME	varchar2	Name		
TYPE	varchar2	Type		
UNITCOST	number	Unit Cost		
UNITPRICE	number	Unit Price		
USEINWORKTIMEENTRY	boolean	Use in Work Time Entry		
VAT	number	VAT		

TravelReimbursementAttachments: Visma Severa Travel Reimbursement Attachments

Catalog: Severa

Schema: API

Label: Travel Reimbursement Attachments

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

Filters on API Server: guid

Parameters of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The following parameters can be used to control the behaviour of the table function TravelReimbursementAttachments. 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
guid		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CONTENTTYPE	varchar2	Content Type		
DESCRIPTION	varchar2	Ddescription		
GUID	varchar2	GUID		
NAME	varchar2	Name		
SIZE	number	Size		

TravelReimbursementByGUID: Visma Severa Travel Reimbursement by GUID

Catalog: Severa

Schema: API

Label: Travel Reimbursement by GUID

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

Filters on API Server: travelReimbursementGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function TravelReimbursementByGUID. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
travelReimbursementGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ADVANCEPAYMENT	number	Advance Payment		
APPROVALDATE	datetime	Approval Date		
APPROVEDBYUSERGUID	varchar2	Approved by User GUID		
CURRENCYGUID	varchar2	Currency GUID		
CURRENCYSHORTFORM	varchar2	Currency Short Form		
DATE	datetime	Date		
GUID	varchar2	GUID		
NOTES	varchar2	Notes		
NUMBER	number	Number		
TRAVELREIMBURSEMENTROWS	varchar2			
TRAVELREIMBURSEMENTSTATUS	varchar2	Travel Reimbursement Status		
USERGUID	varchar2	User GUID		

TravelReimbursementByNumber: Visma Severa Travel Reimbursement by Number

Catalog: Severa

Schema: API

Label: Travel Reimbursement by Number

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

Filters on API Server: travelReimbursementNumber-travelReimbursementNumberingBusinessUnitGUID

Parameters of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The following parameters can be used to control the behaviour of the table function TravelReimbursementByNumber. 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
travelReimbursementNumber		☐		
travelReimbursementNumberingBusinessUnitGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ADVANCEPAYMENT	number	Advance Payment		
APPROVALDATE	datetime	Approval Date		
APPROVEDBYUSERGUID	varchar2	Approved by User GUID		
CURRENCYGUID	varchar2	Currency GUID		
CURRENCYSHORTFORM	varchar2	Currency Short Form		
DATE	datetime	Date		
GUID	varchar2	GUID		
NOTES	varchar2	Notes		
NUMBER	number	Number		
TRAVELREIMBURSEMENTROWS	varchar2			
TRAVELREIMBURSEMENTSTATUS	varchar2	Travel Reimbursement Status		
USERGUID	varchar2	User GUID		

TravelReimbursementPdf: Visma Severa Travel Reimbursement PDF

Catalog: Severa

Schema: API

Label: Travel Reimbursement PDF

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Filters on API Server: guid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function TravelReimbursementPdf. 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
guid		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
GETTRAVELREIMBURSEMENTPDF RESULT	blob			

TravelReimbursementsByDate: Visma Severa Travel Reimbursement by Date

Catalog: Severa

Schema: API

Label: Travel Reimbursement by Date

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

Filters on API Server: startDate- endDate- options

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function TravelReimbursementsByDate. 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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
endDate		☐		
options		☐		
startDate		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ADVANCEPAYMENT	number	Advance Payment		
APPROVALDATE	datetime	Approval Date		
APPROVEDBYUSERGUID	varchar2	Approved by User GUID		
CURRENCYGUID	varchar2	Currency GUID		
CURRENCYSHORTFORM	varchar2	Currency Short Form		
DATE	datetime	Date		
GUID	varchar2	GUID		
NOTES	varchar2	Notes		
NUMBER	number	Number		
TRAVELREIMBURSEMENTROWS	varchar2			
TRAVELREIMBURSEMENTSTATUS	varchar2	Travel Reimbursement Status		
USERGUID	varchar2	User GUID		

TravelReimbursementsByStatus: Visma Severa Travel Reimbursement by Status

Catalog: Severa

Schema: API

Label: Travel Reimbursement by Status

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Filters on API Server: status- options- businessUnitGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function TravelReimbursementsByStatus. 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
businessUnitGUID		☐		
options		☐		
status		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ADVANCEPAYMENT	number	Advance Payment		
APPROVALDATE	datetime	Approval Date		
APPROVEDBYUSERGUID	varchar2	Approved by User GUID		
CURRENCYGUID	varchar2	Currency GUID		
CURRENCYSHORTFORM	varchar2	Currency Short Form		
DATE	datetime	Date		
GUID	varchar2	GUID		
NOTES	varchar2	Notes		
NUMBER	number	Number		
TRAVELREIMBURSEMENTROWS	varchar2			
TRAVELREIMBURSEMENTSTATUS	varchar2	Travel Reimbursement Status		
USERGUID	varchar2	User GUID		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

TravelReimbursementsByStatusGUID: Visma Severa Travel Reimbursement by Status GUID

Catalog: Severa

Schema: API

Label: Travel Reimbursement by Status GUID

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

Filters on API Server: travelReimbursementStatusGUID- options

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function TravelReimbursementsByStatusGUID. 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
options		☐		
travelReimbursementStatusGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ADVANCEPAYMENT	number	Advance Payment		
APPROVALDATE	datetime	Approval Date		
APPROVEDBYUSERGUID	varchar2	Approved by User GUID		
CURRENCYGUID	varchar2	Currency GUID		
CURRENCYSHORTFORM	varchar2	Currency Short Form		
DATE	datetime	Date		
GUID	varchar2	GUID		
NOTES	varchar2	Notes		
NUMBER	number	Number		
TRAVELREIMBURSEMENTROWS	varchar2			

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
TRAVELREIMBURSEMENTSTATUS	varchar2	Travel Reimbursement Status		
USERGUID	varchar2	User GUID		

TravelReimbursementStatuses: Visma Severa Travel Reimbursement Statuses

Catalog: Severa

Schema: API

Label: Travel Reimbursement Statuses

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

Table Columns

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

Name	Data Type	Label	Required	Documentation
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
ISAPPROVED	boolean	Is Approved		
ISLOCKED	boolean	Is Locked		
NAME	varchar2	Name		
SORTORDER	number	Sort Order		

UpdateAccessRightsProfile: Visma Severa Update Access Rights Profile

Catalog: Severa

Schema: API

Label: Update Access Rights Profile

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

Filters on API Server: accessRightsProfile_GUID- accessRightsProfile_IsActive- accessRightsProfile_IsDefault- accessRightsProfile_Name- accessRightsProfile_Rights_Account- accessRightsProfile_Rights_AccountDelete- accessRightsProfile_Rights_Administrator- accessRightsProfile_Rights_Case- accessRightsProfile_Rights_CaseDelete- accessRightsProfile_Rights_CaseOwnerWorkHourApproval- accessRightsProfile_Rights_ScheduleJobs- accessRightsProfile_Rights_Sharing- accessRightsProfile_Rights_TravelReimbursement- accessRightsProfile_Rights_Users- accessRightsProfile_Rights_WorkHourApproval

Parameters of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The following parameters can be used to control the behaviour of the table function UpdateAccessRightsProfile. 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
accessRightsProfile_GUID		☐		
accessRightsProfile_IsActive		☐		
accessRightsProfile_IsDefault		☐		
accessRightsProfile_Name		☐		
accessRightsProfile_Rights_Account		☐		
accessRightsProfile_Rights_AccountDelete		☐		
accessRightsProfile_Rights_Administrator		☐		
accessRightsProfile_Rights_Case		☐		
accessRightsProfile_Rights_CaseDelete		☐		
accessRightsProfile_Rights_CaseOwnerWorkHourApproval		☐		
accessRightsProfile_Rights_ScheduleJobs		☐		
accessRightsProfile_Rights_Sharing		☐		
accessRightsProfile_Rights_TravelReimbursement		☐		
accessRightsProfile_Rights_Users		☐		
accessRightsProfile_Rights_WorkHourApproval		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
ISDEFAULT	boolean	Is Default		
NAME	varchar2	Name		
RIGHTS_ACCOUNT	varchar2			
RIGHTS_ACCOUNTDELETE	boolean			
RIGHTS_ADMINISTRATOR	varchar2			
RIGHTS_CASE	varchar2			
RIGHTS_CASEDELETE	boolean			
RIGHTS_CASEOWNERWORKHOURAPPROVAL	boolean			
RIGHTS_SCHEDULEJOBS	varchar2			
RIGHTS_SHARING	boolean			
RIGHTS_TRAVELREIMBURSEMENT	varchar2			
RIGHTS_USERS	varchar2			
RIGHTS_WORKHOURAPPROVAL	varchar2			

UpdateAccount: Visma Severa Update Account

Catalog: Severa

Schema: API

Label: Update Account

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

Filters on API Server: accountInfo_AccountGroupGUIDs- accountInfo_AccountOwnerUserGUID- accountInfo_AccountRating- accountInfo_CompanyGUID- accountInfo_CurrencyCode- accountInfo_CurrencyGUID- accountInfo_GUID- accountInfo_InsertTS- accountInfo_InvoicingVat- accountInfo_IsActive- accountInfo_IsInternal- accountInfo_IsPrivatePerson- accountInfo_LanguageCode- accountInfo_LanguageGUID- accountInfo_Name- accountInfo_Notes- accountInfo_Number- accountInfo_OverdueInterest- accountInfo_PaymentTerm- accountInfo_PricelistGUID- accountInfo_ReverseCharge- accountInfo_ReverseChargeDescription- accountInfo_UpdatedTS- accountInfo_UrlToAccount- accountInfo_eInvoiceAddress- accountInfo_eInvoiceOperator- companyInfo_AnnualRevenue- companyInfo_AnnualRevenue2- companyInfo_Email- companyInfo_Employees- companyInfo_GUID- companyInfo_HeadOfficeAddressGUID- companyInfo_HierarchyGUID- companyInfo_IndustryGUID- companyInfo_KvkNumber- companyInfo_LanguageGUID- companyInfo_Name- companyInfo_TimezoneGUID- companyInfo_VatNumber- companyInfo_Website

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UpdateAccount. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
accountInfo_AccountGroupGUIDs		☐		
accountInfo_AccountOwnerUserGUID		☐		
accountInfo_AccountRating		☐		
accountInfo_CompanyGUID		☐		
accountInfo_CurrencyCode		☐		
accountInfo_CurrencyGUID		☐		
accountInfo_eInvoiceAddress		☐		
accountInfo_eInvoiceOperator		☐		
accountInfo_GUID		☐		
accountInfo_InsertTS		☐		
accountInfo_InvoicingVat		☐		
accountInfo_IsActive		☐		
accountInfo_IsInternal		☐		
accountInfo_IsPrivatePerson		☐		
accountInfo_LanguageCode		☐		
accountInfo_LanguageGUID		☐		
accountInfo_Name		☐		
accountInfo_Notes		☐		
accountInfo_Number		☐		
accountInfo_OverdueInterest		☐		
accountInfo_PaymentTerm		☐		
accountInfo_PricelistGUID		☐		
accountInfo_ReverseCharge		☐		
accountInfo_ReverseChargeDescription		☐		
accountInfo_UpdatedTS		☐		
accountInfo_UrlToAccount		☐		
companyInfo_AnnualRevenue		☐		
companyInfo_AnnualRevenue2		☐		
companyInfo_Email		☐		
companyInfo_Employees		☐		
companyInfo_GUID		☐		
companyInfo_HeadOfficeAddressGUID		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
companyInfo_HierarchyGUID		☐		
companyInfo_IndustryGUID		☐		
companyInfo_KvkNumber		☐		
companyInfo_LanguageGUID		☐		
companyInfo_Name		☐		
companyInfo_TimezoneGUID		☐		
companyInfo_VatNumber		☐		
companyInfo_Website		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGROUPGUIDS	varchar2			
ACCOUNTOWNERUSERGUID	varchar2	Account Owner User GUID		
ACCOUNTRATING	number	Account Rating		
COMPANYGUID	varchar2	Company GUID		
CURRENCYCODE	varchar2	Currency Code		
CURRENCYGUID	varchar2	Currency GUID		
EINVOICEADDRESS	varchar2	e-invoice Address		
EINVOICEOPERATOR	varchar2	e-invoice Operator		
GUID	varchar2	GUID		
INSERTTS	datetime	Inserted		
INVOICINGVAT	number	Invoicing VAT		
ISACTIVE	boolean	Is Active		
ISINTERNAL	boolean	Is Internal		
ISPRIVATEPERSON	boolean			
LANGUAGECODE	varchar2	Language Code		
LANGUAGEGUID	varchar2	Language GUID		
NAME	varchar2	Name		
NOTES	varchar2	Notes		
NUMBER	number	Number		
OVERDUEINTEREST	number	Overdue Interest		
PAYMENTTERM	number	Payment Term		
PRICELISTGUID	varchar2	Pricelist GUID		
REVERSECHARGE	boolean	Reverse Charge		
REVERSECHARGEDESCRIPTION	varchar2	Reverse Charge Description		
UPDATEDTS	datetime	Updated		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
URLTOACCOUNT	varchar2	URL to Account		

UpdateActivity: Visma Severa Update Activity

Catalog: Severa

Schema: API

Label: Update Activity

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

Filters on API Server: activityInfo_AccountGUID- activityInfo_ActivityTypeCode- activityInfo_ActivityTypeGUID- activityInfo_CaseGUID- activityInfo_EndDateTime- activityInfo_GUID- activityInfo_IsAllDay- activityInfo_IsClosed- activityInfo_IsDuration- activityInfo_IsPrivate- activityInfo_IsReadOnly- activityInfo_IsUnassigned- activityInfo_Location- activityInfo_Name- activityInfo_Notes- activityInfo_OwnerUserBusinessUnitCode- activityInfo_OwnerUserCode- activityInfo_OwnerUserFirstName- activityInfo_OwnerUserGUID- activityInfo_OwnerUserLastName- activityInfo_StartDateTime- activityInfo_TaskGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UpdateActivity. 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
activityInfo_AccountGUID		☐		
activityInfo_ActivityTypeCode		☐		
activityInfo_ActivityTypeGUID		☐		
activityInfo_CaseGUID		☐		
activityInfo_EndDateTime		☐		
activityInfo_GUID		☐		
activityInfo_IsAllDay		☐		
activityInfo_IsClosed		☐		
activityInfo_IsDuration		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
activityInfo_IsPrivate		☐		
activityInfo_IsReadOnly		☐		
activityInfo_IsUnassigned		☐		
activityInfo_Location		☐		
activityInfo_Name		☐		
activityInfo_Notes		☐		
activityInfo_OwnerUserBusinessUnitCode		☐		
activityInfo_OwnerUserCode		☐		
activityInfo_OwnerUserFirstName		☐		
activityInfo_OwnerUserGUID		☐		
activityInfo_OwnerUserLastName		☐		
activityInfo_StartDateTime		☐		
activityInfo_TaskGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	varchar2	Account GUID		
ACTIVITYTYPECODE	varchar2	Activity Type Code		
ACTIVITYTYPEGUID	varchar2	Activity Type GUID		
CASEGUID	varchar2	Project GUID		
ENDDATETIME	datetime	End Date Time		
GUID	varchar2	GUID		
ISALLDAY	boolean	Is All Day		
ISCLOSED	boolean	Is Closed		
ISDURATION	boolean	Is Duration		
ISPRIVATE	boolean	Is Private		
ISREADONLY	boolean	Is Read-only		
ISUNASSIGNED	boolean	Is Unassigned		
LOCATION	varchar2	Location		
NAME	varchar2	Name		
NOTES	varchar2	Notes		
OWNERUSERBUSINESSUNITCODE	varchar2	Owner User Business Unit Code		
OWNERUSERCODE	varchar2	Owner User Code		
OWNERUSERFIRSTNAME	varchar2	Owner User First Name		
OWNERUSERGUID	varchar2	Owner User GUID		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
OWNERUSERLASTNAME	varchar2	Owner User Last Name		
STARTDATETIME	datetime	Start Date Time		
TASKGUID	varchar2	Task GUID		

UpdateActivityType: Visma Severa Update Activity Type

Catalog: Severa

Schema: API

Label: Update Activity Type

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

Filters on API Server: activityTypeInfo_Category- activityTypeInfo_Code- activityTypeInfo_GUID- activityTypeInfo_Icon- activityTypeInfo_IsActive- activityTypeInfo_IsDefault- activityTypeInfo_IsPaidLeave- activityTypeInfo_Name

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UpdateActivityType. 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
activityTypeInfo_Category		☐		
activityTypeInfo_Code		☐		
activityTypeInfo_GUID		☐		
activityTypeInfo_Icon		☐		
activityTypeInfo_IsActive		☐		
activityTypeInfo_IsDefault		☐		
activityTypeInfo_IsPaidLeave		☐		
activityTypeInfo_Name		☐		

Columns of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Name	Data Type	Label	Required	Documentation
CATEGORY	varchar2	Category		
CODE	varchar2	Code		
GUID	varchar2	GUID		
ICON	varchar2	Icon		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
ISPAIDLEAVE	boolean	Is Paid Leave		
NAME	varchar2	Name		

UpdateAddress: Visma Severa Update Address

Catalog: Severa

Schema: API

Label: Update Address

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

Filters on API Server: addressInfo_Addressline- addressInfo_Addressline2- addressInfo_Addressline3- addressInfo_City- addressInfo_CompanyGUID- addressInfo_CountryCode- addressInfo_CountryGUID- addressInfo_CountryName- addressInfo_CountryRegionGUID- addressInfo_CountryRegionName- addressInfo_Fax- addressInfo_GUID- addressInfo_IsBillingAddress- addressInfo_IsPostalAddress- addressInfo_IsVisitAddress- addressInfo_Phone- addressInfo_PostalCode

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UpdateAddress. 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
addressInfo_Addressline		☐		
addressInfo_Addressline2		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
addressInfo_Addressline3		☐		
addressInfo_City		☐		
addressInfo_CompanyGUID		☐		
addressInfo_CountryCode		☐		
addressInfo_CountryGUID		☐		
addressInfo_CountryName		☐		
addressInfo_CountryRegionGUID		☐		
addressInfo_CountryRegionName		☐		
addressInfo_Fax		☐		
addressInfo_GUID		☐		
addressInfo_IsBillingAddress		☐		
addressInfo_IsPostalAddress		☐		
addressInfo_IsVisitAddress		☐		
addressInfo_Phone		☐		
addressInfo_PostalCode		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ADDRESSLINE	varchar2	Addressline		
ADDRESSLINE2	varchar2	Addressline 2		
ADDRESSLINE3	varchar2	Addressline 3		
CITY	varchar2	City		
COMPANYGUID	varchar2	Company GUID		
COUNTRYCODE	varchar2	Country Code		
COUNTRYGUID	varchar2	Country GUID		
COUNTRYNAME	varchar2	Country Name		
COUNTRYREGIONGUID	varchar2	Country Region GUID		
COUNTRYREGIONNAME	varchar2	Country Region Name		
FAX	varchar2	Fax		
GUID	varchar2	GUID		
ISBILLINGADDRESS	boolean	Is Billing Address		
ISPOSTALADDRESS	boolean	Is Postal Address		
ISVISITADDRESS	boolean	Is Visit Address		
PHONE	varchar2	Phone		
POSTALCODE	varchar2	Postal Code		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

UpdateBillingAndExpenseForecast: Visma Severa Update Billing and Expense Forecast

Catalog: Severa

Schema: API

Label: Update Billing and Expense Forecast

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

Filters on API Server: billingAndExpenseForecastInfo_BillingForecast- billingAndExpenseForecastInfo_ExpenseForecast- billingAndExpenseForecastInfo_GUID- billingAndExpenseForecastInfo_InhouseRevenueForecast- billingAndExpenseForecastInfo_Month- billingAndExpenseForecastInfo_Notes- billingAndExpenseForecastInfo_OutsourcingRevenueForecast- billingAndExpenseForecastInfo_Year

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UpdateBillingAndExpenseForecast. 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
billingAndExpenseForecastInfo_BillingForecast		☐		
billingAndExpenseForecastInfo_ExpenseForecast		☐		
billingAndExpenseForecastInfo_GUID		☐		
billingAndExpenseForecastInfo_InhouseRevenueForecast		☐		
billingAndExpenseForecastInfo_Month		☐		
billingAndExpenseForecastInfo_Notes		☐		
billingAndExpenseForecastInfo_OutsourcingRevenueForecast		☐		
billingAndExpenseForecastInfo_Year		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
BILLINGFORECAST	number	Billing Forecast		
EXPENSEFORECAST	number	Expense Forecast		
GUID	varchar2	GUID		
INHOUSEREVENUEFORECAST	number	Inhouse Revenue Forecast		
MONTH	number	Month		
NOTES	varchar2	Notes		
OUTSOURCINGREVENUEFORECAST	number	Outsourcing Revenue Forecast		
YEAR	number	Year		

UpdateBusinessUnit: Visma Severa Update Business Unit

Catalog: Severa

Schema: API

Label: Update Business Unit

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

Filters on API Server: businessUnitInfo_Code- businessUnitInfo_CostCenterGUID- businessUnitInfo_CurrencyCode- businessUnitInfo_CurrencyGUID- businessUnitInfo_FormattingCulture- businessUnitInfo_GUID- businessUnitInfo_InsertTS- businessUnitInfo_IsActive- businessUnitInfo_LanguageCode- businessUnitInfo_LanguageGUID- businessUnitInfo_Name- businessUnitInfo_NextInvoiceNumber- businessUnitInfo_OverDueInterest- businessUnitInfo_ParentBusinessUnitGUID- businessUnitInfo_PaymentTerm- businessUnitInfo_ShowLogoOnInvoices- businessUnitInfo_ShowLogoOnPdfReports- businessUnitInfo_TimeZoneGUID- businessUnitInfo_UpdatedTS- businessUnitInfo_VatNumber

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UpdateBusinessUnit. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
businessUnitInfo_Code		☐		
businessUnitInfo_CostCenterGUID		☐		
businessUnitInfo_CurrencyCode		☐		
businessUnitInfo_CurrencyGUID		☐		
businessUnitInfo_FormattingCulture		☐		
businessUnitInfo_GUID		☐		
businessUnitInfo_InsertTS		☐		
businessUnitInfo_IsActive		☐		
businessUnitInfo_LanguageCode		☐		
businessUnitInfo_LanguageGUID		☐		
businessUnitInfo_Name		☐		
businessUnitInfo_NextInvoiceNumber		☐		
businessUnitInfo_OverDueInterest		☐		
businessUnitInfo_ParentBusinessUnitGUID		☐		
businessUnitInfo_PaymentTerm		☐		
businessUnitInfo_Show LogoOnInvoices		☐		
businessUnitInfo_Show LogoOnPdf Reports		☐		
businessUnitInfo_TimeZoneGUID		☐		
businessUnitInfo_UpdatedTS		☐		
businessUnitInfo_VatNumber		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CODE	varchar2	Code		
COSTCENTERGUID	varchar2	Cost Centre GUID		
CURRENCYCODE	varchar2	Currency Code		
CURRENCYGUID	varchar2	Currency GUID		
FORMATTINGCULTURE	varchar2	Formatting Culture		
GUID	varchar2	GUID		
INSERTTS	datetime	Inserted		
ISACTIVE	boolean	Is Active		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
LANGUAGECODE	varchar2	Language Code		
LANGUAGEGUID	varchar2	Language GUID		
NAME	varchar2	Name		
NEXTINVOICENUMBER	number	Next Invoice Number		
OVERDUEINTEREST	number	Overdue Interest		
PARENTBUSINESSUNITGUID	varchar2	Parent Business Unit GUID		
PAYMENTTERM	number	Payment Term		
SHOWLOGOONINVOICES	boolean	Show Logo on Invoices		
SHOWLOGOONPDFREPORTS	boolean	Show Logo on PDF Reports		
TIMEZONEGUID	varchar2	Timezone GUID		
UPDATEDTS	datetime	Updated		
VATNUMBER	varchar2	VAT Number		

UpdateCase: Visma Severa Update Project

Catalog: Severa

Schema: API

Label: Update Project

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

Filters on API Server: caseInfo_AccountGUID- caseInfo_AccountName- caseInfo_AccountNumber- caseInfo_BillingAddress- caseInfo_BillingAddressGUID- caseInfo_BillingInternalNotes- caseInfo_BillingNotesBeforeGrid- caseInfo_BusinessUnitGUID- caseInfo_BusinessUnitName- caseInfo_BusinessUnitNumber- caseInfo_CaseCompanyCurrencyGUID- caseInfo_CaseNumber- caseInfo_CaseOwnerUserCode- caseInfo_CaseOwnerUserGUID- caseInfo_ClosedDate- caseInfo_CompletionEstimate- caseInfo_ContactGUID- caseInfo_CostCenterGUID- caseInfo_CostCenterNumber- caseInfo_CurrencyGUID- caseInfo_CurrencyShortform- caseInfo_CurrencyShortformOfCaseCompanyCurrency- caseInfo_CurrentCaseStatusDescription- caseInfo_CurrentCaseStatusGUID- caseInfo_CurrentCaseStatusInsertTs- caseInfo_DeadlineDate- caseInfo_Description- caseInfo_ExpectedValue- caseInfo_GUID- caseInfo_IncludeExceptions- caseInfo_IncludeOverTime- caseInfo_InsertTs- caseInfo_InternalName- caseInfo_InvoiceTemplateGUID- caseInfo_IsClosed- caseInfo_IsDailyAllowanceBillable- caseInfo_IsInternal- caseInfo_IsMileageBillable- caseInfo_IsOtherTravelExpensesBillable- caseInfo_LatestEstimationDate- caseInfo_LeadRating- caseInfo_LeadSourceGUID- caseInfo_Name- caseInfo_OrderNumber- caseInfo_OurReference- caseInfo_OverdueInterest- caseInfo_PaymentTerm- caseInfo_PricelistGUID- caseInfo_PricingRule- caseInfo_PricingType- caseInfo_Priority- caseInfo_Probability- caseInfo_RootTaskGUID- caseInfo_SalesCloseDate- caseInfo_SalesPersonUserCode- caseInfo_SalesPersonUserGUID- caseInfo_SalesProcessGUID- caseInfo_SalesStatusGUID- caseInfo_SharedToExtranet- caseInfo_StartDate-

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

caseInfo_UpdatedTs- caseInfo_UrlToCase- caseInfo_UseDefaultProductsInWorkTimeEntry- caseInfo_UseDefaultWorkTypes- caseInfo_YourReference

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UpdateCase. 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
caseInfo_AccountGUID		☐		
caseInfo_AccountName		☐		
caseInfo_AccountNumber		☐		
caseInfo_BillingAddress		☐		
caseInfo_BillingAddressGUID		☐		
caseInfo_BillingInternalNotes		☐		
caseInfo_BillingNotesBeforeGrid		☐		
caseInfo_BusinessUnitGUID		☐		
caseInfo_BusinessUnitName		☐		
caseInfo_BusinessUnitNumber		☐		
caseInfo_CaseCompanyCurrencyGUID		☐		
caseInfo_CaseNumber		☐		
caseInfo_CaseOwnerUserCode		☐		
caseInfo_CaseOwnerUserGUID		☐		
caseInfo_ClosedDate		☐		
caseInfo_CompletionEstimate		☐		
caseInfo_ContactGUID		☐		
caseInfo_CostCenterGUID		☐		
caseInfo_CostCenterNumber		☐		
caseInfo_CurrencyGUID		☐		
caseInfo_CurrencyShortform		☐		
caseInfo_CurrencyShortformOfCaseCompanyCurrency		☐		
caseInfo_CurrentCaseStatusDescription		☐		
caseInfo_CurrentCaseStatusGUID		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
caseInfo_CurrentCaseStatusInsertTs		☐		
caseInfo_DeadlineDate		☐		
caseInfo_Description		☐		
caseInfo_ExpectedValue		☐		
caseInfo_GUID		☐		
caseInfo_IncludeExceptions		☐		
caseInfo_IncludeOverTime		☐		
caseInfo_InsertTs		☐		
caseInfo_InternalName		☐		
caseInfo_InvoiceTemplateGUID		☐		
caseInfo_IsClosed		☐		
caseInfo_IsDailyAllowanceBillable		☐		
caseInfo_IsInternal		☐		
caseInfo_IsMileageBillable		☐		
caseInfo_IsOtherTravelExpensesBillable		☐		
caseInfo_LatestEstimationDate		☐		
caseInfo_LeadRating		☐		
caseInfo_LeadSourceGUID		☐		
caseInfo_Name		☐		
caseInfo_OrderNumber		☐		
caseInfo_OurReference		☐		
caseInfo_OverdueInterest		☐		
caseInfo_PaymentTerm		☐		
caseInfo_PricelistGUID		☐		
caseInfo_PricingRule		☐		
caseInfo_PricingType		☐		
caseInfo_Priority		☐		
caseInfo_Probability		☐		
caseInfo_RootTaskGUID		☐		
caseInfo_SalesCloseDate		☐		
caseInfo_SalesPersonUserCode		☐		
caseInfo_SalesPersonUserGUID		☐		
caseInfo_SalesProcessGUID		☐		
caseInfo_SalesStatusGUID		☐		
caseInfo_SharedToExtranet		☐		
caseInfo_StartDate		☐		
caseInfo_UpdatedTs		☐		
caseInfo_UrlToCase		☐		
caseInfo_UseDefaultProductsInWorkTimeEntry		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
caseInfo_UseDefaultWorkTypes		☐		
caseInfo_YourReference		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	varchar2	Account GUID		
ACCOUNTNAME	varchar2	Account Name		
ACCOUNTNUMBER	number	Account Number		
BILLINGADDRESS	varchar2	Billing Address		
BILLINGADDRESSGUID	varchar2	Billing Address GUID		
BILLINGINTERNALNOTES	varchar2	Billing Internal Notes		
BILLINGNOTESBEFOREGRID	varchar2	Billing Notes before Grid		
BUSINESSUNITGUID	varchar2	Business Unit GUID		
BUSINESSUNITNAME	varchar2	Business Unit Name		
BUSINESSUNITNUMBER	varchar2	Business Unit Number		
CASECOMPANYCURRENCYGUID	varchar2	Project Company Currency GUID		
CASENUMBER	number	Project Number		
CASEOWNERUSERCODE	varchar2	Project Owner User Code		
CASEOWNERUSERGUID	varchar2	Project Owner User GUID		
CLOSEDDATE	datetime	Closed Date		
COMPLETIONESTIMATE	number	Completion Estimate		
CONTACTGUID	varchar2	Contact GUID		
COSTCENTERGUID	varchar2	Cost Centre GUID		
COSTCENTERNUMBER	varchar2	Cost Centre Number		
CURRENCYGUID	varchar2	Currency GUID		
CURRENCYSHORTFORM	varchar2	Currency Short Form		
CURRENCYSHORTFORMOFCASECOMPANYCURRENCY	varchar2	Currency Short Form of Project Company Currency		
CURRENTCASESTATUSDESCRIPTION	varchar2	Current Project Status Description		
CURRENTCASESTATUSGUID	varchar2	Current Project Status GUID		
CURRENTCASESTATUSINSERTTS	datetime	Current Project Status Inserted		
DEADLINEDATE	datetime	Deadline Date		
DESCRIPTION	varchar2	Ddescription		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
EXPECTEDVALUE	number	Expected Value		
GUID	varchar2	GUID		
INCLUDEEXCEPTIONS	boolean	Include Exceptions		
INCLUDEOVERTIME	boolean	Include Overtime		
INSERTTS	datetime	Inserted		
INTERNALNAME	varchar2	Internal Name		
INVOICETEMPLATEGUID	varchar2	Invoice Template GUID		
ISCLOSED	boolean	Is Closed		
ISDAILYALLOWANCEBILLABLE	boolean	Is Daily Allowance Billable		
ISINTERNAL	boolean	Is Internal		
ISMILEAGEBILLABLE	boolean	Is Mileage Billable		
ISOTHERTRAVELEXPENSESBILLABLE	boolean	Are Other Travel Expenses Billable		
LATESTESTIMATIONDATE	datetime	Latest Estimation Date		
LEADRATING	number	Lead Rating		
LEADSOURCEGUID	varchar2	Lead Source GUID		
NAME	varchar2	Name		
ORDERNUMBER	varchar2	Order Number		
OURREFERENCE	varchar2	Our Reference		
OVERDUEINTEREST	number	Overdue Interest		
PAYMENTTERM	number	Payment Term		
PRICELISTGUID	varchar2	Pricelist GUID		
PRICINGRULE	varchar2	Pricing Rule		
PRICINGTYPE	varchar2	Pricing Type		
PRIORITY	number	Priority		
PROBABILITY	number	Probability		
ROOTTASKGUID	varchar2	Root Task GUID		
SALESCLOSEDATE	datetime	Sales Close Date		
SALESPERSONUSERCODE	varchar2	Sales Person User Code		
SALESPERSONUSERGUID	varchar2	Sales Person User GUID		
SALESPROCESSGUID	varchar2	Sales Process GUID		
SALESSTATUSGUID	varchar2	Sales Status GUID		
SHAREDTOEXTRANET	boolean	Shared to Extranet		
STARTDATE	datetime	Start Date		
UPDATEDTS	datetime	Updated		
URLTOCASE	varchar2	URL to Project		
USEDEFAULTPRODUCTSINWORKTIMEENTRY	boolean	Use Default Products in Work Time Entry		
USEDEFAULTWORKTYPES	boolean	Use Default Work Types		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
YOURREFERENCE	varchar2	Your Reference		

UpdateCaseStatus: Visma Severa Update Project Status

Catalog: Severa

Schema: API

Label: Update Project Status

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

Filters on API Server: caseStatusInfo_GUID- caseStatusInfo_Icon- caseStatusInfo_IsActive- caseStatusInfo_Name

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UpdateCaseStatus. 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
caseStatusInfo_GUID		☐		
caseStatusInfo_Icon		☐		
caseStatusInfo_IsActive		☐		
caseStatusInfo_Name		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
GUID	varchar2	GUID		
ICON	varchar2	Icon		
ISACTIVE	boolean	Is Active		
NAME	varchar2	Name		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

UpdateContact: Visma Severa Update Contact

Catalog: Severa

Schema: API

Label: Update Contact

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

Filters on API Server: contactInfo_AccountGUID- contactInfo_AddressGUID- contactInfo_CommunicationMethods- contactInfo_ContactRoleGUID- contactInfo_DateOfBirth- contactInfo_Description- contactInfo_FirstName- contactInfo_GUID- contactInfo_IsActive- contactInfo_JobTitle- contactInfo_Keywords- contactInfo_LanguageGUID- contactInfo_LastName- contactInfo_Salutation- contactInfo_SatisfactionLevel- contactInfo_TimeZoneGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UpdateContact. 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
contactInfo_AccountGUID		☐		
contactInfo_AddressGUID		☐		
contactInfo_CommunicationMethods		☐		
contactInfo_ContactRoleGUID		☐		
contactInfo_DateOfBirth		☐		
contactInfo_Description		☐		
contactInfo_FirstName		☐		
contactInfo_GUID		☐		
contactInfo_IsActive		☐		
contactInfo_JobTitle		☐		
contactInfo_Keywords		☐		
contactInfo_LanguageGUID		☐		
contactInfo_LastName		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
contactInfo_Salutation		☐		
contactInfo_SatisfactionLevel		☐		
contactInfo_TimeZoneGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	varchar2	Account GUID		
ADDRESSGUID	varchar2	Address GUID		
COMMUNICATIONMETHODS	varchar2			
CONTACTROLEGUID	varchar2	Contact Role GUID		
DATEOFBIRTH	datetime	Date of Birth		
DESCRIPTION	varchar2	Ddescription		
FIRSTNAME	varchar2	First Name		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
JOBTITLE	varchar2	Job Title		
KEYWORDS	varchar2			
LANGUAGEGUID	varchar2	Language GUID		
LASTNAME	varchar2	Last Name		
SALUTATION	varchar2	Salutation		
SATISFACTIONLEVEL	number	Satisfaction Level		
TIMEZONEGUID	varchar2	Timezone GUID		

UpdateCustomer: Visma Severa Update Customer

Catalog: Severa

Schema: API

Label: Update Customer

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

Filters on API Server: customer_AccountGroupGUIDs- customer_AccountOwnerUserGUID- customer_AccountRating- customer_AnnualRevenue- customer_AnnualRevenue2- customer_CompanyGUID- customer_CurrencyCode- customer_CurrencyGUID- customer_Email- customer_Employees- customer_GUID- customer_HeadOfficeAddressGUID- customer_IndustryGUID- customer_InsertTS- customer_InvoicingVat- customer_IsActive- customer_IsInternal- customer_IsPrivatePerson- customer_LanguageCode- customer_LanguageGUID- customer_Name- customer_Notes- customer_Number- customer_OverdueInterest- customer_PaymentTerm-

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

customer_PricelistGUID- customer_ReverseCharge- customer_ReverseChargeDescription- customer_TimezoneGUID- customer_UpdatedTS- customer_UrlToAccount- customer_VatNumber- customer_Website- customer_eInvoiceAddress- customer_eInvoiceOperator

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UpdateCustomer. 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
customer_AccountGroupGUIDs		☐		
customer_AccountOwnerUserGUID		☐		
customer_AccountRating		☐		
customer_AnnualRevenue		☐		
customer_AnnualRevenue2		☐		
customer_CompanyGUID		☐		
customer_CurrencyCode		☐		
customer_CurrencyGUID		☐		
customer_eInvoiceAddress		☐		
customer_eInvoiceOperator		☐		
customer_Email		☐		
customer_Employees		☐		
customer_GUID		☐		
customer_HeadOfficeAddressGUID		☐		
customer_IndustryGUID		☐		
customer_InsertTS		☐		
customer_InvoicingVat		☐		
customer_IsActive		☐		
customer_IsInternal		☐		
customer_IsPrivatePerson		☐		
customer_LanguageCode		☐		
customer_LanguageGUID		☐		
customer_Name		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
customer_Notes		☐		
customer_Number		☐		
customer_OverdueInterest		☐		
customer_PaymentTerm		☐		
customer_PricelistGUID		☐		
customer_ReverseCharge		☐		
customer_ReverseChargeDescription		☐		
customer_TimezoneGUID		☐		
customer_UpdatedTS		☐		
customer_UrlToAccount		☐		
customer_VatNumber		☐		
customer_Website		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGROUPGUIDS	varchar2			
ACCOUNTOWNERUSERGUID	varchar2	Account Owner User GUID		
ACCOUNTRATING	number	Account Rating		
ANNUALREVENUE	number	Annual Revenue		
ANNUALREVENUE2	number			
COMPANYGUID	varchar2	Company GUID		
CURRENCYCODE	varchar2	Currency Code		
CURRENCYGUID	varchar2	Currency GUID		
EINVOICEADDRESS	varchar2	e-invoice Address		
EINVOICEOPERATOR	varchar2	e-invoice Operator		
EMAIL	varchar2	Email		
EMPLOYEES	number	Employees		
GUID	varchar2	GUID		
HEADOFFICEADDRESSGUID	varchar2	Head Office Address GUID		
INDUSTRYGUID	varchar2	Industry GUID		
INSERTTS	datetime	Inserted		
INVOICINGVAT	number	Invoicing VAT		
ISACTIVE	boolean	Is Active		
ISINTERNAL	boolean	Is Internal		
ISPRIVATEPERSON	boolean			

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
LANGUAGECODE	varchar2	Language Code		
LANGUAGEGUID	varchar2	Language GUID		
NAME	varchar2	Name		
NOTES	varchar2	Notes		
NUMBER	number	Number		
OVERDUEINTEREST	number	Overdue Interest		
PAYMENTTERM	number	Payment Term		
PRICELISTGUID	varchar2	Pricelist GUID		
REVERSECHARGE	boolean	Reverse Charge		
REVERSECHARGEDESCRIPTION	varchar2	Reverse Charge Description		
TIMEZONEGUID	varchar2	Timezone GUID		
UPDATEDTS	datetime	Updated		
URLTOACCOUNT	varchar2	URL to Account		
VATNUMBER	varchar2	VAT Number		
WEBSITE	varchar2	Website		

UpdateEmployment: Visma Severa Update Employment

Catalog: Severa

Schema: API

Label: Update Employment

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

Filters on API Server: employmentInfo_CompanyBusinessUnitCode- employmentInfo_CompanyBusinessUnitGUID- employmentInfo_CompanyBusinessUnitName- employmentInfo_DailyHours- employmentInfo_EndDate- employmentInfo_GUID- employmentInfo_HourCost- employmentInfo_IsOvertimeAllowed- employmentInfo_StartDate- employmentInfo_Title

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UpdateEmployment. 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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
employmentInfo_CompanyBusinessUnitCode		☐		
employmentInfo_CompanyBusinessUnitGUID		☐		
employmentInfo_CompanyBusinessUnitName		☐		
employmentInfo_DailyHours		☐		
employmentInfo_EndDate		☐		
employmentInfo_GUID		☐		
employmentInfo_HourCost		☐		
employmentInfo_IsOvertimeAllowed		☐		
employmentInfo_StartDate		☐		
employmentInfo_Title		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
COMPANYBUSINESSUNITCODE	varchar2	Company Business Unit Code		
COMPANYBUSINESSUNITGUID	varchar2	Company Business Unit GUID		
COMPANYBUSINESSUNITNAME	varchar2	Company Business Unit Name		
DAILYHOURS	number	Daily Hours		
ENDDATE	datetime	End Date		
GUID	varchar2	GUID		
HOURCOST	number	Hour Cost		
ISOVERTIMEALLOWED	boolean	Is Overtime Allowed		
STARTDATE	datetime	Start Date		
TITLE	varchar2	Title		

UpdateHourEntry: Visma Severa Update Hour Entry

Catalog: Severa

Schema: API

Label: Update Hour Entry

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Filters on API Server: hourEntryInfo_AccountGUID- hourEntryInfo_CaseBusinessUnitCode- hourEntryInfo_CaseCostCenterIdentifier- hourEntryInfo_CaseGUID- hourEntryInfo_CaseName- hourEntryInfo_CaseNumber- hourEntryInfo_CostCurrencyShortform- hourEntryInfo_CostCurrencyShortformOfInvoicingCurrency- hourEntryInfo_CostCurrencyShortformOfOrganizationCurrency- hourEntryInfo_CostCurrencyShortformOfUserCompanyCurrency- hourEntryInfo_Description- hourEntryInfo_EndTime- hourEntryInfo_EventDate- hourEntryInfo_GUID- hourEntryInfo_InvoiceDescription- hourEntryInfo_InvoiceGUID- hourEntryInfo_InvoiceQuantity- hourEntryInfo_InvoiceRowGUID- hourEntryInfo_IsApproved- hourEntryInfo_IsBillable- hourEntryInfo_IsProductive- hourEntryInfo_IsReadOnly- hourEntryInfo_OverTimeGUID- hourEntryInfo_OvertimeCode- hourEntryInfo_PhaseGUID- hourEntryInfo_PriceCurrencyShortformOfCaseCompanyCurrency- hourEntryInfo_PriceCurrencyShortformOfInvoicingCurrency- hourEntryInfo_PriceCurrencyShortformOfOrganizationCurrency- hourEntryInfo_PriceCurrencyShortformOfUserCompanyCurrency- hourEntryInfo_Quantity- hourEntryInfo_StartTime- hourEntryInfo_TaskGUID- hourEntryInfo_UnitCost- hourEntryInfo_UnitCostInInvoicingCurrency- hourEntryInfo_UnitCostInOrganizationCurrency- hourEntryInfo_UnitCostInUserCompanyCurrency- hourEntryInfo_UnitPrice- hourEntryInfo_UnitPriceInBaseCurrency- hourEntryInfo_UnitPriceInCaseCompanyCurrency- hourEntryInfo_UnitPriceInUserCompanyCurrency- hourEntryInfo_UseInvoiceQuantity- hourEntryInfo_UserBusinessUnitCode- hourEntryInfo_UserCode- hourEntryInfo_UserFirstName- hourEntryInfo_UserGUID- hourEntryInfo_UserLastName- hourEntryInfo_WorkTypeCode- hourEntryInfo_WorkTypeGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UpdateHourEntry. 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
hourEntryInfo_AccountGUID		☐		
hourEntryInfo_CaseBusinessUnitCode		☐		
hourEntryInfo_CaseCostCenterIdentifier		☐		
hourEntryInfo_CaseGUID		☐		
hourEntryInfo_CaseName		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
hourEntryInfo_CaseNumber		☐		
hourEntryInfo_CostCurrencyShortform		☐		
hourEntryInfo_CostCurrencyShortformOfInvoicingCurrency		☐		
hourEntryInfo_CostCurrencyShortformOfOrganizationCurrency		☐		
hourEntryInfo_CostCurrencyShortformOfUserCompanyCurrency		☐		
hourEntryInfo_Description		☐		
hourEntryInfo_EndTime		☐		
hourEntryInfo_EventDate		☐		
hourEntryInfo_GUID		☐		
hourEntryInfo_InvoiceDescription		☐		
hourEntryInfo_InvoiceGUID		☐		
hourEntryInfo_InvoiceQuantity		☐		
hourEntryInfo_InvoiceRow GUID		☐		
hourEntryInfo_IsApproved		☐		
hourEntryInfo_IsBillable		☐		
hourEntryInfo_IsProductive		☐		
hourEntryInfo_IsReadOnly		☐		
hourEntryInfo_OvertimeCode		☐		
hourEntryInfo_OverTimeGUID		☐		
hourEntryInfo_PhaseGUID		☐		
hourEntryInfo_PriceCurrencyShortformOfCaseCompanyCurrency		☐		
hourEntryInfo_PriceCurrencyShortformOfInvoicingCurrency		☐		
hourEntryInfo_PriceCurrencyShortformOfOrganizationCurrency		☐		
hourEntryInfo_PriceCurrencyShortformOfUserCompanyCurrency		☐		
hourEntryInfo_Quantity		☐		
hourEntryInfo_StartTime		☐		
hourEntryInfo_TaskGUID		☐		
hourEntryInfo_UnitCost		☐		
hourEntryInfo_UnitCostInInvoicingCurrency		☐		
hourEntryInfo_UnitCostInOrganizationCurrency		☐		
hourEntryInfo_UnitCostInUserCompanyCurrency		☐		
hourEntryInfo_UnitPrice		☐		
hourEntryInfo_UnitPriceInBaseCurrency		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
hourEntryInfo_UnitPriceInCaseCompanyCurrency		☐		
hourEntryInfo_UnitPriceInUserCompanyCurrency		☐		
hourEntryInfo_UseInvoiceQuantity		☐		
hourEntryInfo_UserBusinessUnitCode		☐		
hourEntryInfo_UserCode		☐		
hourEntryInfo_UserFirstName		☐		
hourEntryInfo_UserGUID		☐		
hourEntryInfo_UserLastName		☐		
hourEntryInfo_WorkTypeCode		☐		
hourEntryInfo_WorkTypeGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	varchar2	Account GUID		
CASEBUSINESSUNITCODE	varchar2	Project Business Unit Code		
CASECOSTCENTERIDENTIFIER	varchar2	Project Cost Centre Identifier		
CASEGUID	varchar2	Project GUID		
CASENAME	varchar2	Project Name		
CASENUMBER	number	Project Number		
COSTCURRENCYSHORTFORM	varchar2	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Cost Currency Short Form of Organisation Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Cost Currency Short Form of User Company Currency		
DESCRIPTION	varchar2	Ddescription		
ENDTIME	datetime	End Time		
EVENTDATE	datetime	Event Date		
GUID	varchar2	GUID		
INVOICEDESCRIPTION	varchar2	Invoice Description		
INVOICEGUID	varchar2	Invoice GUID		
INVOICEQUANTITY	number	Invoice Quantity		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
INVOICEROWGUID	varchar2	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISPRODUCTIVE	boolean	Is Productive		
ISREADONLY	boolean	Is Read-only		
OVERTIMECODE	varchar2	Overtime Code		
OVERTIMEGUID	varchar2	Overtime GUID		
PHASEGUID	varchar2	Phase GUID		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	varchar2	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Price Currency Short Form of Organisation Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Price Currency Short Form of User Company Currency		
QUANTITY	number	Quantity		
STARTTIME	datetime	Start Time		
TASKGUID	varchar2	Task GUID		
UNITCOST	number	Unit Cost		
UNITCOSTININVOICINGCURRENCY	number	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	number	Unit Cost in Organisation Currency		
UNITCOSTINUSERCOMPANYCURRENCY	number	Unit Cost in User Company Currency		
UNITPRICE	number	Unit Price		
UNITPRICEINBASECURRENCY	number	Unit Price in Base Currency		
UNITPRICEINCASECOMPANYCURRENCY	number	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	number	Unit Price in User Company Currency		
USEINVOICEQUANTITY	boolean			
USERBUSINESSUNITCODE	varchar2	User Business Unit Code		
USERCODE	varchar2	User Code		
USERFIRSTNAME	varchar2	User First Name		
USERGUID	varchar2	User GUID		
USERLASTNAME	varchar2	User Last Name		
WORKTYPECODE	varchar2	Work Type Code		
WORKTYPEGUID	varchar2	Work Type GUID		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

UpdatePhase: Visma Severa Update Phase

Catalog: Severa

Schema: API

Label: Update Phase

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

Filters on API Server: phaseInfo_CaseGUID- phaseInfo_Code- phaseInfo_Deadline- phaseInfo_Description- phaseInfo_FlatRate_AdditionalHoursUnitPrice- phaseInfo_FlatRate_AreAdditionalHoursBillable- phaseInfo_FlatRate_Hours- phaseInfo_FlatRate_Price- phaseInfo_GUID- phaseInfo_IsCompleted- phaseInfo_IsLocked- phaseInfo_Name- phaseInfo_OriginalDeadline- phaseInfo_OriginalPlannedStartDate- phaseInfo_OriginalWorkEstimate- phaseInfo_OwnerUserGUID- phaseInfo_ParentPhaseGUID- phaseInfo_PlannedStartDate- phaseInfo_PricePerHour- phaseInfo_SortOrder- phaseInfo_WorkEstimate- phaseInfo_WorkTypeGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UpdatePhase. 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
phaseInfo_CaseGUID		☐		
phaseInfo_Code		☐		
phaseInfo_Deadline		☐		
phaseInfo_Description		☐		
phaseInfo_FlatRate_AdditionalHoursUnitPrice		☐		
phaseInfo_FlatRate_AreAdditionalHoursBillable		☐		
phaseInfo_FlatRate_Hours		☐		
phaseInfo_FlatRate_Price		☐		
phaseInfo_GUID		☐		
phaseInfo_IsCompleted		☐		
phaseInfo_IsLocked		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
phaseInfo_Name		☐		
phaseInfo_OriginalDeadline		☐		
phaseInfo_OriginalPlannedStartDate		☐		
phaseInfo_OriginalWorkEstimate		☐		
phaseInfo_OwnerUserGUID		☐		
phaseInfo_ParentPhaseGUID		☐		
phaseInfo_PlannedStartDate		☐		
phaseInfo_PricePerHour		☐		
phaseInfo_SortOrder		☐		
phaseInfo_WorkEstimate		☐		
phaseInfo_WorkTypeGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CASEGUID	varchar2	Project GUID		
CODE	varchar2	Code		
DEADLINE	datetime	Deadline		
DESCRIPTION	varchar2	Ddescription		
FLATRATE_ADDITIONALHOURSUNITPRICE	number	Flat Rate Additional Hours Unit Price		
FLATRATE_AREADDITIONALHOURSBILLABLE	boolean	Flat Rate Are additional Hours Billable		
FLATRATE_HOURS	number	Flat Rate Hours		
FLATRATE_PRICE	number	Flat Rate Price		
GUID	varchar2	GUID		
ISCOMPLETED	boolean	Is Completed		
ISLOCKED	boolean	Is Locked		
NAME	varchar2	Name		
ORIGINALDEADLINE	datetime	Original Deadline		
ORIGINALPLANNEDSTARTDATE	datetime	Original Planned Start Date		
ORIGINALWORKESTIMATE	number	Original Work Estimate		
OWNERUSERGUID	varchar2	Owner User GUID		
PARENTPHASEGUID	varchar2	Parent Phase GUID		
PLANNEDSTARTDATE	datetime	Planned Start Date		
PRICEPERHOUR	number	Price per Hour		
SORTORDER	number	Sort Order		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
WORKESTIMATE	number	Work Estimate		
WORKTYPEGUID	varchar2	Work Type GUID		

UpdateProduct: Visma Severa Update Product

Catalog: Severa

Schema: API

Label: Update Product

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

Filters on API Server: productInfo_Code- productInfo_Description- productInfo_GUID- productInfo_IsActive- productInfo_MeasurementUnit- productInfo_Name- productInfo_Type- productInfo_UnitCost- productInfo_UnitPrice- productInfo_UseInWorkTimeEntry- productInfo_VAT- productInfo_ProductCategoryGUID- productInfo_SalesAccountGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UpdateProduct. 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
productInfo_Code		☐		
productInfo_Description		☐		
productInfo_GUID		☐		
productInfo_IsActive		☐		
productInfo_MeasurementUnit		☐		
productInfo_Name		☐		
productInfo_ProductCategoryGUID		☐		
productInfo_SalesAccountGUID		☐		
productInfo_Type		☐		
productInfo_UnitCost		☐		
productInfo_UnitPrice		☐		
productInfo_UseInWorkTimeEntry		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
productInfo_VAT		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CODE	varchar2	Code		
DESCRIPTION	varchar2	Ddescription		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
MEASUREMENTUNIT	varchar2	Measurement Unit		
NAME	varchar2	Name		
PRODUCTCATEGORYGUID	varchar2	Product Category GUID		
SALESACCOUNTGUID	varchar2	Sales Account GUID		
TYPE	varchar2	Type		
UNITCOST	number	Unit Cost		
UNITPRICE	number	Unit Price		
USEINWORKTIMEENTRY	boolean	Use in Work Time Entry		
VAT	number	VAT		

UpdateProductCategory: Visma Severa Update Product Category

Catalog: Severa

Schema: API

Label: Update Product Category

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

Filters on API Server: productCategoryInfo_Code- productCategoryInfo_GUID- productCategoryInfo_IsActive- productCategoryInfo_IsDefault- productCategoryInfo_Name

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UpdateProductCategory. 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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
productCategoryInfo_Code		☐		
productCategoryInfo_GUID		☐		
productCategoryInfo_IsActive		☐		
productCategoryInfo_IsDefault		☐		
productCategoryInfo_Name		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CODE	varchar2	Code		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
NAME	varchar2	Name		

UpdateResource: Visma Severa Update Resource

Catalog: Severa

Schema: API

Label: Update Resource

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

Filters on API Server: resourceInfo_GUID- resourceInfo_IsActive- resourceInfo_Name

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UpdateResource. 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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
resourceInfo_GUID		☐		
resourceInfo_IsActive		☐		
resourceInfo_Name		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
NAME	varchar2	Name		

UpdateResourceAllocation: Visma Severa Update Resource Allocation

Catalog: Severa

Schema: API

Label: Update Resource Allocation

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

Filters on API Server: resourceAllocationInfo_AllocatedHours- resourceAllocationInfo_AllocatedPercentage- resourceAllocationInfo_CaseGUID- resourceAllocationInfo_EndDate- resourceAllocationInfo_GUID- resourceAllocationInfo_PhaseGUID- resourceAllocationInfo_StartDate- resourceAllocationInfo_UserGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UpdateResourceAllocation. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
resourceAllocationInfo_AllocatedHours		☐		
resourceAllocationInfo_AllocatedPercentage		☐		
resourceAllocationInfo_CaseGUID		☐		
resourceAllocationInfo_EndDate		☐		
resourceAllocationInfo_GUID		☐		
resourceAllocationInfo_PhaseGUID		☐		
resourceAllocationInfo_StartDate		☐		
resourceAllocationInfo_UserGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ALLOCATEDHOURS	number	Allocated Hours		
ALLOCATEDPERCENTAGE	number	Allocated Percentage		
CASEGUID	varchar2	Project GUID		
ENDDATE	datetime	End Date		
GUID	varchar2	GUID		
PHASEGUID	varchar2	Phase GUID		
STARTDATE	datetime	Start Date		
USERGUID	varchar2	User GUID		

UpdateTravelExpense: Visma Severa Update Travel Expense

Catalog: Severa

Schema: API

Label: Update Travel Expense

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

Filters on API Server: travelExpenseInfo_Code- travelExpenseInfo_Description- travelExpenseInfo_GUID- travelExpenseInfo_IsActive- travelExpenseInfo_MeasurementUnit- travelExpenseInfo_Name- travelExpenseInfo_Type- travelExpenseInfo_UnitCost- travelExpenseInfo_UnitPrice- travelExpenseInfo_UseInWorkTimeEntry-

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

travelExpenseInfo_VAT- travelExpenseInfo_CostCurrencyGUID- travelExpenseInfo_ExpenseClass

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UpdateTravelExpense. 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
travelExpenseInfo_Code		☐		
travelExpenseInfo_CostCurrencyGUID		☐		
travelExpenseInfo_Description		☐		
travelExpenseInfo_ExpenseClass		☐		
travelExpenseInfo_GUID		☐		
travelExpenseInfo_IsActive		☐		
travelExpenseInfo_MeasurementUnit		☐		
travelExpenseInfo_Name		☐		
travelExpenseInfo_Type		☐		
travelExpenseInfo_UnitCost		☐		
travelExpenseInfo_UnitPrice		☐		
travelExpenseInfo_UseInWorkTimeEntry		☐		
travelExpenseInfo_VAT		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CODE	varchar2	Code		
COSTCURRENCYGUID	varchar2	Cost Currency GUID		
DESCRIPTION	varchar2	Ddescription		
EXPENSECLASS	varchar2	Expense Class		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
MEASUREMENTUNIT	varchar2	Measurement Unit		
NAME	varchar2	Name		
TYPE	varchar2	Type		
UNITCOST	number	Unit Cost		
UNITPRICE	number	Unit Price		
USEINWORKTIMEENTRY	boolean	Use in Work Time Entry		
VAT	number	VAT		

UpdateUser: Visma Severa Update User

Catalog: Severa

Schema: API

Label: Update User

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

Filters on API Server: userInfo_Addressline- userInfo_BankAccountNumber- userInfo_BusinessUnitCode- userInfo_BusinessUnitGUID- userInfo_BusinessUnitName- userInfo_Code- userInfo_CountryCode- userInfo_CountryGUID- userInfo_CountryRegionGUID- userInfo_CountryRegionName- userInfo_Culture- userInfo_DefaultActivityTypeGUID- userInfo_Email- userInfo_FirstName- userInfo_GUID- userInfo_IsActive- userInfo_LanguageCode- userInfo_LanguageGUID- userInfo_LastName- userInfo_LicenseType- userInfo_Notes- userInfo_OrganizationGUID- userInfo_Phone- userInfo_PhotoFileGUID- userInfo_PostOffice- userInfo_PostalCode- userInfo_ProfileGUID- userInfo_Salutation- userInfo_SendDailyMail- userInfo_SendWeeklyMail- userInfo_SocialSecurityNumber- userInfo_SuperiorUserGUID- userInfo_TimezoneGUID- userInfo_Title- userInfo_WorktypeGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UpdateUser. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
userInfo_Addressline		☐		
userInfo_BankAccountNumber		☐		
userInfo_BusinessUnitCode		☐		
userInfo_BusinessUnitGUID		☐		
userInfo_BusinessUnitName		☐		
userInfo_Code		☐		
userInfo_CountryCode		☐		
userInfo_CountryGUID		☐		
userInfo_CountryRegionGUID		☐		
userInfo_CountryRegionName		☐		
userInfo_Culture		☐		
userInfo_DefaultActivityTypeGUID		☐		
userInfo_Email		☐		
userInfo_FirstName		☐		
userInfo_GUID		☐		
userInfo_IsActive		☐		
userInfo_LanguageCode		☐		
userInfo_LanguageGUID		☐		
userInfo_LastName		☐		
userInfo_LicenseType		☐		
userInfo_Notes		☐		
userInfo_OrganizationGUID		☐		
userInfo_Phone		☐		
userInfo_PhotoFileGUID		☐		
userInfo_PostalCode		☐		
userInfo_PostOffice		☐		
userInfo_ProfileGUID		☐		
userInfo_Salutation		☐		
userInfo_SendDailyMail		☐		
userInfo_SendWeeklyMail		☐		
userInfo_SocialSecurityNumber		☐		
userInfo_SuperiorUserGUID		☐		
userInfo_TimezoneGUID		☐		
userInfo_Title		☐		
userInfo_WorktypeGUID		☐		

Columns of Table Function

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
ADDRESSLINE	varchar2	Addressline		
BANKACCOUNTNUMBER	varchar2	Bank Account Number		
BUSINESSUNITCODE	varchar2	Business Unit Code		
BUSINESSUNITGUID	varchar2	Business Unit GUID		
BUSINESSUNITNAME	varchar2	Business Unit Name		
CODE	varchar2	Code		
COUNTRYCODE	varchar2	Country Code		
COUNTRYGUID	varchar2	Country GUID		
COUNTRYREGIONGUID	varchar2	Country Region GUID		
COUNTRYREGIONNAME	varchar2	Country Region Name		
CULTURE	varchar2	Culture		
DEFAULTACTIVITYTYPEGUID	varchar2	Default Activity Type GUID		
EMAIL	varchar2	Email		
FIRSTNAME	varchar2	First Name		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
LANGUAGECODE	varchar2	Language Code		
LANGUAGEGUID	varchar2	Language GUID		
LASTNAME	varchar2	Last Name		
LICENSETYPE	varchar2			
NOTES	varchar2	Notes		
ORGANIZATIONGUID	varchar2	Organisation GUID		
PHONE	varchar2	Phone		
PHOTOFILEGUID	varchar2	Photo File GUID		
POSTALCODE	varchar2	Postal Code		
POSTOFFICE	varchar2	Post Office		
PROFILEGUID	varchar2	Profile GUID		
SALUTATION	varchar2	Salutation		
SENDDAILYMAIL	boolean	Send Daily Mail		
SENDWEEKLYMAIL	boolean	Send Weekly Mail		
SOCIALSECURITYNUMBER	varchar2	Social Security Number		
SUPERIORUSERGUID	varchar2	Superior User GUID		
TIMEZONEGUID	varchar2	Timezone GUID		
TITLE	varchar2	Title		
WORKTYPEGUID	varchar2	Work Type GUID		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

UpdateUserRightsByUserGUID: Visma Severa Update User Rights by User GUID

Catalog: Severa

Schema: API

Label: Update User Rights by User GUID

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

Filters on API Server: guid- rights_API- rights_Rights_Account- rights_Rights_AccountDelete- rights_Rights_Administrator- rights_Rights_Case- rights_Rights_CaseDelete- rights_Rights_CaseOwnerWorkHourApproval- rights_Rights_ScheduleJobs- rights_Rights_Sharing- rights_Rights_TravelReimbursement- rights_Rights_Users- rights_Rights_WorkHourApproval

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UpdateUserRightsByUserGUID. 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
guid		☐		
rights_API		☐		
rights_Rights_Account		☐		
rights_Rights_AccountDelete		☐		
rights_Rights_Administrator		☐		
rights_Rights_Case		☐		
rights_Rights_CaseDelete		☐		
rights_Rights_CaseOwnerWorkHourApproval		☐		
rights_Rights_ScheduleJobs		☐		
rights_Rights_Sharing		☐		
rights_Rights_TravelReimbursement		☐		
rights_Rights_Users		☐		
rights_Rights_WorkHourApproval		☐		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
API	boolean	API		
RIGHTS_ACCOUNT	varchar2			
RIGHTS_ACCOUNTDELETE	boolean			
RIGHTS_ADMINISTRATOR	varchar2			
RIGHTS_CASE	varchar2			
RIGHTS_CASEDELETE	boolean			
RIGHTS_CASEOWNERWORKHOURAPPROVAL	boolean			
RIGHTS_SCHEDULEJOBS	varchar2			
RIGHTS_SHARING	boolean			
RIGHTS_TRAVELREIMBURSEMENT	varchar2			
RIGHTS_USERS	varchar2			
RIGHTS_WORKHOURAPPROVAL	varchar2			

UpdateWorkType: Visma Severa Update Work Type

Catalog: Severa

Schema: API

Label: Update Work Type

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

Filters on API Server: workTypeInfo_Code- workTypeInfo_GUID- workTypeInfo_IsActive- workTypeInfo_IsDefault- workTypeInfo_IsProductive- workTypeInfo_Name- workTypeInfo_SalesAccountGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UpdateWorkType. 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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
workTypeInfo_Code		☐		
workTypeInfo_GUID		☐		
workTypeInfo_IsActive		☐		
workTypeInfo_IsDefault		☐		
workTypeInfo_IsProductive		☐		
workTypeInfo_Name		☐		
workTypeInfo_SalesAccountGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CODE	varchar2	Code		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
ISPRODUCTIVE	boolean	Is Productive		
NAME	varchar2	Name		
SALESACCOUNTGUID	varchar2	Sales Account GUID		

UserByCode: Visma Severa User by Code

Catalog: Severa

Schema: API

Label: User by Code

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

Filters on API Server: userCode

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UserByCode. 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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
userCode		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ADDRESSLINE	varchar2	Addressline		
BANKACCOUNTNUMBER	varchar2	Bank Account Number		
BUSINESSUNITCODE	varchar2	Business Unit Code		
BUSINESSUNITGUID	varchar2	Business Unit GUID		
BUSINESSUNITNAME	varchar2	Business Unit Name		
CODE	varchar2	Code		
COUNTRYCODE	varchar2	Country Code		
COUNTRYGUID	varchar2	Country GUID		
COUNTRYREGIONGUID	varchar2	Country Region GUID		
COUNTRYREGIONNAME	varchar2	Country Region Name		
CULTURE	varchar2	Culture		
DEFAULTACTIVITYTYPEGUID	varchar2	Default Activity Type GUID		
EMAIL	varchar2	Email		
FIRSTNAME	varchar2	First Name		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
LANGUAGECODE	varchar2	Language Code		
LANGUAGEGUID	varchar2	Language GUID		
LASTNAME	varchar2	Last Name		
LICENSETYPE	varchar2			
NOTES	varchar2	Notes		
ORGANIZATIONGUID	varchar2	Organisation GUID		
PHONE	varchar2	Phone		
PHOTOFILEGUID	varchar2	Photo File GUID		
POSTALCODE	varchar2	Postal Code		
POSTOFFICE	varchar2	Post Office		
PROFILEGUID	varchar2	Profile GUID		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
SALUTATION	varchar2	Salutation		
SENDDAILYMAIL	boolean	Send Daily Mail		
SENDWEEKLYMAIL	boolean	Send Weekly Mail		
SOCIALSECURITYNUMBER	varchar2	Social Security Number		
SUPERIORUSERGUID	varchar2	Superior User GUID		
TIMEZONEGUID	varchar2	Timezone GUID		
TITLE	varchar2	Title		
WORKTYPEGUID	varchar2	Work Type GUID		

UserByGUID: Visma Severa User by GUID

Catalog: Severa

Schema: API

Label: User by GUID

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

Filters on API Server: userGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UserByGUID. 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
userGUID		☐		

Columns of Table Function

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
ADDRESSLINE	varchar2	Addressline		
BANKACCOUNTNUMBER	varchar2	Bank Account Number		
BUSINESSUNITCODE	varchar2	Business Unit Code		
BUSINESSUNITGUID	varchar2	Business Unit GUID		
BUSINESSUNITNAME	varchar2	Business Unit Name		
CODE	varchar2	Code		
COUNTRYCODE	varchar2	Country Code		
COUNTRYGUID	varchar2	Country GUID		
COUNTRYREGIONGUID	varchar2	Country Region GUID		
COUNTRYREGIONNAME	varchar2	Country Region Name		
CULTURE	varchar2	Culture		
DEFAULTACTIVITYTYPEGUID	varchar2	Default Activity Type GUID		
EMAIL	varchar2	Email		
FIRSTNAME	varchar2	First Name		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
LANGUAGECODE	varchar2	Language Code		
LANGUAGEGUID	varchar2	Language GUID		
LASTNAME	varchar2	Last Name		
LICENSETYPE	varchar2			
NOTES	varchar2	Notes		
ORGANIZATIONGUID	varchar2	Organisation GUID		
PHONE	varchar2	Phone		
PHOTOFILEGUID	varchar2	Photo File GUID		
POSTALCODE	varchar2	Postal Code		
POSTOFFICE	varchar2	Post Office		
PROFILEGUID	varchar2	Profile GUID		
SALUTATION	varchar2	Salutation		
SENDDAILYMAIL	boolean	Send Daily Mail		
SENDWEEKLYMAIL	boolean	Send Weekly Mail		
SOCIALSECURITYNUMBER	varchar2	Social Security Number		
SUPERIORUSERGUID	varchar2	Superior User GUID		
TIMEZONEGUID	varchar2	Timezone GUID		
TITLE	varchar2	Title		
WORKTYPEGUID	varchar2	Work Type GUID		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

UserByName: Visma Severa User by Name

Catalog: Severa

Schema: API

Label: User by Name

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

Filters on API Server: firstName- lastName

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UserByName. 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
firstName		☐		
lastName		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ADDRESSLINE	varchar2	Addressline		
BANKACCOUNTNUMBER	varchar2	Bank Account Number		
BUSINESSUNITCODE	varchar2	Business Unit Code		
BUSINESSUNITGUID	varchar2	Business Unit GUID		
BUSINESSUNITNAME	varchar2	Business Unit Name		
CODE	varchar2	Code		
COUNTRYCODE	varchar2	Country Code		
COUNTRYGUID	varchar2	Country GUID		
COUNTRYREGIONGUID	varchar2	Country Region GUID		
COUNTRYREGIONNAME	varchar2	Country Region Name		
CULTURE	varchar2	Culture		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
DEFAULTACTIVITYTYPEGUID	varchar2	Default Activity Type GUID		
EMAIL	varchar2	Email		
FIRSTNAME	varchar2	First Name		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
LANGUAGECODE	varchar2	Language Code		
LANGUAGEGUID	varchar2	Language GUID		
LASTNAME	varchar2	Last Name		
LICENSETYPE	varchar2			
NOTES	varchar2	Notes		
ORGANIZATIONGUID	varchar2	Organisation GUID		
PHONE	varchar2	Phone		
PHOTOFILEGUID	varchar2	Photo File GUID		
POSTALCODE	varchar2	Postal Code		
POSTOFFICE	varchar2	Post Office		
PROFILEGUID	varchar2	Profile GUID		
SALUTATION	varchar2	Salutation		
SENDDAILYMAIL	boolean	Send Daily Mail		
SENDWEEKLYMAIL	boolean	Send Weekly Mail		
SOCIALSECURITYNUMBER	varchar2	Social Security Number		
SUPERIORUSERGUID	varchar2	Superior User GUID		
TIMEZONEGUID	varchar2	Timezone GUID		
TITLE	varchar2	Title		
WORKTYPEGUID	varchar2	Work Type GUID		

UserRightsByUserGUID: Visma Severa User Rights by User GUID

Catalog: Severa

Schema: API

Label: User Rights by User GUID

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

Filters on API Server: guid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UserRightsByUserGUID. 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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
guid		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
API	boolean	API		
RIGHTS_ACCOUNT	varchar2			
RIGHTS_ACCOUNTDELETE	boolean			
RIGHTS_ADMINISTRATOR	varchar2			
RIGHTS_CASE	varchar2			
RIGHTS_CASEDELETE	boolean			
RIGHTS_CASEOWNERWORKHOURAPPROVAL	boolean			
RIGHTS_SCHEDULEJOBS	varchar2			
RIGHTS_SHARING	boolean			
RIGHTS_TRAVELREIMBURSEMENT	varchar2			
RIGHTS_USERS	varchar2			
RIGHTS_WORKHOURAPPROVAL	varchar2			

Users: Visma Severa Users

Catalog: Severa

Schema: API

Label: Users

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

Filters on API Server: businessUnitGUID- includeInactive

Parameters of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The following parameters can be used to control the behaviour of the table function Users. 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
businessUnitGUID		☐		
includeInactive		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ADDRESSLINE	varchar2	Addressline		
BANKACCOUNTNUMBER	varchar2	Bank Account Number		
BUSINESSUNITCODE	varchar2	Business Unit Code		
BUSINESSUNITGUID	varchar2	Business Unit GUID		
BUSINESSUNITNAME	varchar2	Business Unit Name		
CODE	varchar2	Code		
COUNTRYCODE	varchar2	Country Code		
COUNTRYGUID	varchar2	Country GUID		
COUNTRYREGIONGUID	varchar2	Country Region GUID		
COUNTRYREGIONNAME	varchar2	Country Region Name		
CULTURE	varchar2	Culture		
DEFAULTACTIVITYTYPEGUID	varchar2	Default Activity Type GUID		
EMAIL	varchar2	Email		
FIRSTNAME	varchar2	First Name		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
LANGUAGECODE	varchar2	Language Code		
LANGUAGEGUID	varchar2	Language GUID		
LASTNAME	varchar2	Last Name		
LICENSETYPE	varchar2			

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
NOTES	varchar2	Notes		
ORGANIZATIONGUID	varchar2	Organisation GUID		
PHONE	varchar2	Phone		
PHOTOFILEGUID	varchar2	Photo File GUID		
POSTALCODE	varchar2	Postal Code		
POSTOFFICE	varchar2	Post Office		
PROFILEGUID	varchar2	Profile GUID		
SALUTATION	varchar2	Salutation		
SENDDAILYMAIL	boolean	Send Daily Mail		
SENDWEEKLYMAIL	boolean	Send Weekly Mail		
SOCIALSECURITYNUMBER	varchar2	Social Security Number		
SUPERIORUSERGUID	varchar2	Superior User GUID		
TIMEZONEGUID	varchar2	Timezone GUID		
TITLE	varchar2	Title		
WORKTYPEGUID	varchar2	Work Type GUID		

UsersByAccessRightsProfileGUID: Visma Severa Users by Access Rights Profile GUID

Catalog: Severa

Schema: API

Label: Users by Access Rights Profile GUID

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

Filters on API Server: guid

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UsersByAccessRightsProfileGUID. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
guid		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ADDRESSLINE	varchar2	Addressline		
BANKACCOUNTNUMBER	varchar2	Bank Account Number		
BUSINESSUNITCODE	varchar2	Business Unit Code		
BUSINESSUNITGUID	varchar2	Business Unit GUID		
BUSINESSUNITNAME	varchar2	Business Unit Name		
CODE	varchar2	Code		
COUNTRYCODE	varchar2	Country Code		
COUNTRYGUID	varchar2	Country GUID		
COUNTRYREGIONGUID	varchar2	Country Region GUID		
COUNTRYREGIONNAME	varchar2	Country Region Name		
CULTURE	varchar2	Culture		
DEFAULTACTIVITYTYPEGUID	varchar2	Default Activity Type GUID		
EMAIL	varchar2	Email		
FIRSTNAME	varchar2	First Name		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
LANGUAGECODE	varchar2	Language Code		
LANGUAGEGUID	varchar2	Language GUID		
LASTNAME	varchar2	Last Name		
LICENSETYPE	varchar2			
NOTES	varchar2	Notes		
ORGANIZATIONGUID	varchar2	Organisation GUID		
PHONE	varchar2	Phone		
PHOTOFILEGUID	varchar2	Photo File GUID		
POSTALCODE	varchar2	Postal Code		
POSTOFFICE	varchar2	Post Office		
PROFILEGUID	varchar2	Profile GUID		
SALUTATION	varchar2	Salutation		
SENDDAILYMAIL	boolean	Send Daily Mail		
SENDWEEKLYMAIL	boolean	Send Weekly Mail		
SOCIALSECURITYNUMBER	varchar2	Social Security Number		
SUPERIORUSERGUID	varchar2	Superior User GUID		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
TIMEZONEGUID	varchar2	Timezone GUID		
TITLE	varchar2	Title		
WORKTYPEGUID	varchar2	Work Type GUID		

UsersByGUIDs: Visma Severa Users by GUIDs

Catalog: Severa

Schema: API

Label: Users by GUIDs

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

Filters on API Server: userGUIDs

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UsersByGUIDs. 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
userGUIDs		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ADDRESSLINE	varchar2	Addressline		
BANKACCOUNTNUMBER	varchar2	Bank Account Number		
BUSINESSUNITCODE	varchar2	Business Unit Code		
BUSINESSUNITGUID	varchar2	Business Unit GUID		
BUSINESSUNITNAME	varchar2	Business Unit Name		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
CODE	varchar2	Code		
COUNTRYCODE	varchar2	Country Code		
COUNTRYGUID	varchar2	Country GUID		
COUNTRYREGIONGUID	varchar2	Country Region GUID		
COUNTRYREGIONNAME	varchar2	Country Region Name		
CULTURE	varchar2	Culture		
DEFAULTACTIVITYTYPEGUID	varchar2	Default Activity Type GUID		
EMAIL	varchar2	Email		
FIRSTNAME	varchar2	First Name		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
LANGUAGECODE	varchar2	Language Code		
LANGUAGEGUID	varchar2	Language GUID		
LASTNAME	varchar2	Last Name		
LICENSETYPE	varchar2			
NOTES	varchar2	Notes		
ORGANIZATIONGUID	varchar2	Organisation GUID		
PHONE	varchar2	Phone		
PHOTOFILEGUID	varchar2	Photo File GUID		
POSTALCODE	varchar2	Postal Code		
POSTOFFICE	varchar2	Post Office		
PROFILEGUID	varchar2	Profile GUID		
SALUTATION	varchar2	Salutation		
SENDDAILYMAIL	boolean	Send Daily Mail		
SENDWEEKLYMAIL	boolean	Send Weekly Mail		
SOCIALSECURITYNUMBER	varchar2	Social Security Number		
SUPERIORUSERGUID	varchar2	Superior User GUID		
TIMEZONEGUID	varchar2	Timezone GUID		
TITLE	varchar2	Title		
WORKTYPEGUID	varchar2	Work Type GUID		

UsersByUserKeywords: Visma Severa Users by User Keywords

Catalog: Severa

Schema: API

Label: Users by User Keywords

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Filters on API Server: businessUnitGUID- userKeywords

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UsersByUserKeywords. 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
businessUnitGUID		☐		
userKeywords		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ADDRESSLINE	varchar2	Addressline		
BANKACCOUNTNUMBER	varchar2	Bank Account Number		
BUSINESSUNITCODE	varchar2	Business Unit Code		
BUSINESSUNITGUID	varchar2	Business Unit GUID		
BUSINESSUNITNAME	varchar2	Business Unit Name		
CODE	varchar2	Code		
COUNTRYCODE	varchar2	Country Code		
COUNTRYGUID	varchar2	Country GUID		
COUNTRYREGIONGUID	varchar2	Country Region GUID		
COUNTRYREGIONNAME	varchar2	Country Region Name		
CULTURE	varchar2	Culture		
DEFAULTACTIVITYTYPEGUID	varchar2	Default Activity Type GUID		
EMAIL	varchar2	Email		
FIRSTNAME	varchar2	First Name		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
LANGUAGECODE	varchar2	Language Code		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
LANGUAGEGUID	varchar2	Language GUID		
LASTNAME	varchar2	Last Name		
LICENSETYPE	varchar2			
NOTES	varchar2	Notes		
ORGANIZATIONGUID	varchar2	Organisation GUID		
PHONE	varchar2	Phone		
PHOTOFILEGUID	varchar2	Photo File GUID		
POSTALCODE	varchar2	Postal Code		
POSTOFFICE	varchar2	Post Office		
PROFILEGUID	varchar2	Profile GUID		
SALUTATION	varchar2	Salutation		
SENDDAILYMAIL	boolean	Send Daily Mail		
SENDWEEKLYMAIL	boolean	Send Weekly Mail		
SOCIALSECURITYNUMBER	varchar2	Social Security Number		
SUPERIORUSERGUID	varchar2	Superior User GUID		
TIMEZONEGUID	varchar2	Timezone GUID		
TITLE	varchar2	Title		
WORKTYPEGUID	varchar2	Work Type GUID		

UsersChangedSince: Visma Severa Users Changed since

Catalog: Severa

Schema: API

Label: Users Changed since

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

Filters on API Server: businessUnitGUID- startDateTime- options

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UsersChangedSince. 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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
businessUnitGUID		☐		
options		☐		
startDateTime		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ADDRESSLINE	varchar2	Addressline		
BANKACCOUNTNUMBER	varchar2	Bank Account Number		
BUSINESSUNITCODE	varchar2	Business Unit Code		
BUSINESSUNITGUID	varchar2	Business Unit GUID		
BUSINESSUNITNAME	varchar2	Business Unit Name		
CODE	varchar2	Code		
COUNTRYCODE	varchar2	Country Code		
COUNTRYGUID	varchar2	Country GUID		
COUNTRYREGIONGUID	varchar2	Country Region GUID		
COUNTRYREGIONNAME	varchar2	Country Region Name		
CULTURE	varchar2	Culture		
DEFAULTACTIVITYTYPEGUID	varchar2	Default Activity Type GUID		
EMAIL	varchar2	Email		
FIRSTNAME	varchar2	First Name		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
LANGUAGECODE	varchar2	Language Code		
LANGUAGEGUID	varchar2	Language GUID		
LASTNAME	varchar2	Last Name		
LICENSETYPE	varchar2			
NOTES	varchar2	Notes		
ORGANIZATIONGUID	varchar2	Organisation GUID		
PHONE	varchar2	Phone		
PHOTOFILEGUID	varchar2	Photo File GUID		
POSTALCODE	varchar2	Postal Code		
POSTOFFICE	varchar2	Post Office		
PROFILEGUID	varchar2	Profile GUID		
SALUTATION	varchar2	Salutation		
SENDDAILYMAIL	boolean	Send Daily Mail		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
SENDWEEKLYMAIL	boolean	Send Weekly Mail		
SOCIALSECURITYNUMBER	varchar2	Social Security Number		
SUPERIORUSERGUID	varchar2	Superior User GUID		
TIMEZONEGUID	varchar2	Timezone GUID		
TITLE	varchar2	Title		
WORKTYPEGUID	varchar2	Work Type GUID		

UserTags: Visma Severa User Tags

Catalog: Severa

Schema: API

Label: User Tags

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

Table Columns

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

Name	Data Type	Label	Required	Documentation
CONTEXT	varchar2	Context		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
KEYWORD	varchar2	Keyword		
WEIGHT	number	Weight		

Version: Visma Severa Version

Catalog: Severa

Schema: API

Label: Version

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

Table Columns

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

Name	Data Type	Label	Required	Documentation
GETVERSIONRESULT	varchar2			

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

WorkDayInfoByDateRangeAndUserGUID: Visma Severa Workday Information by Date Range and User GUID

Catalog: Severa

Schema: API

Label: Workday Information by Date Range and User GUID

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

Filters on API Server: businessUnitGUID- startdate- enddate- userGUID- includeUnPaidAbsenses- includeInfoForInactiveUser

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function WorkDayInfoByDateRangeAndUserGUID. 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
businessUnitGUID		☐		
enddate		☐		
includeInfoForInactiveUser		☐		
includeUnPaidAbsenses		☐		
startdate		☐		
userGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACTIVITIES	varchar2			
DATE	datetime	Date		
ENTRIES	varchar2			
EXPECTEDHOURS	number	Expected Hours		
OVERTIMEALLOWED	boolean	Overtime Allowed		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
READONLY	boolean	Read-only		
READY	boolean	Ready		
TOTALHOURS	number	Total Hours		

WorkTypeByGUID: Visma Severa Work Type by GUID

Catalog: Severa

Schema: API

Label: Work Type by GUID

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

Filters on API Server: workTypeGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function WorkTypeByGUID. 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
workTypeGUID		□		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CODE	varchar2	Code		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
ISPRODUCTIVE	boolean	Is Productive		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
NAME	varchar2	Name		
SALESACCOUNTGUID	varchar2	Sales Account GUID		

WorkTypeByHourGUID: Visma Severa Work Type by Hour GUID

Catalog: Severa

Schema: API

Label: Work Type by Hour GUID

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

Filters on API Server: hourGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function WorkTypeByHourGUID. 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
hourGUID		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CODE	varchar2	Code		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
ISPRODUCTIVE	boolean	Is Productive		
NAME	varchar2	Name		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
SALESACCOUNTGUID	varchar2	Sales Account GUID		

WorkTypes: Visma Severa Work Types

Catalog: Severa

Schema: API

Label: Work Types

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

Table Columns

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

Name	Data Type	Label	Required	Documentation
CODE	varchar2	Code		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
ISPRODUCTIVE	boolean	Is Productive		
NAME	varchar2	Name		
SALESACCOUNTGUID	varchar2	Sales Account GUID		

WorkTypesByPhaseGUID: Visma Severa Work Types by Phase GUID

Catalog: Severa

Schema: API

Label: Work Types by Phase GUID

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

Filters on API Server: phaseGUID

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function WorkTypesByPhaseGUID. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
phaseGUID		□		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CODE	varchar2	Code		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
ISPRODUCTIVE	boolean	Is Productive		
NAME	varchar2	Name		
SALESACCOUNTGUID	varchar2	Sales Account GUID		

WorkTypesChangedSince: Visma Severa Work Types Changed since

Catalog: Severa

Schema: API

Label: Work Types Changed since

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

Filters on API Server: startDateTime- options

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function WorkTypesChangedSince. 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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
options		☐		
startDateTime		☐		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CODE	varchar2	Code		
GUID	varchar2	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
ISPRODUCTIVE	boolean	Is Productive		
NAME	varchar2	Name		
SALESACCOUNTGUID	varchar2	Sales Account GUID		

3.1.2 Schema: Native

3.1.2.1 Tables

NATIVEPLATFORMSCALARREQUESTS: Visma Severa Native Platform Scalar Requests

{res:itgen_native_platform_scalar_requests_desc}

Catalog: Severa

Schema: Native

Alias: `npt`

Label: Native Platform Scalar Requests

Documentation:

The NativePlatformScalarRequests table provides direct access to the native API protocol over an established connection to the Visma Severa API server. It will contain a new row for every row inserted with a native API request in PAYLOAD_TEXT with the results of unaltered forwarding of the payload to the Visma Severa API server.

Retrieve: true

Insert: true

Update: false

Delete: false

View Columns

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The columns of the view NATIVEPLATFORMSCALARREQUESTS 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.

Name	Data Type	Label	Required	Documentation
BLOB_PREFERRED	char	BLOB Preferred	☑	Indicator w hether a BLOB result is preferred over text.
BOL_RESPONSE_CACHE_MAX_AGE_SEC	int32	Response Cache Maximum Age (sec)	☐	Maximum age in seconds of Bridge Online response cache entries to be used.
CONTENT_TYPE	varchar2(240)	Content Type	☐	
DATE_ENDED	datetime	End Date	☑	
DATE_STARTED	datetime	Start Date	☑	
DRY_RUN	char	Run w ithout Actions	☑	
DURATION_MS	int64	Duration (ms)	☑	
ERROR_MESSAGE_CODE	varchar2(30)	Error Message Code	☐	
ERROR_MESSAGE_TEXT	varchar2(32000)	Error Message Text	☐	
FAIL_ON_ERROR	char	Fail on Error	☑	Whether to raise an exception w hen processing the native request triggered an error from the provider.
HTTP_DISK_CACHE_MAX_AGE_SEC	int32	HTTP Disk Cache Maximum Age (sec)	☐	Maximum age in seconds of HTTP disk cache entries to be used.
HTTP_DISK_CACHE_SAVE	char	Save HTTP Disk Cache	☐	Whether results can be stored in HTTP disk cache.
HTTP_DISK_CACHE_USE	char	Use HTTP Disk Cache	☐	Whether results can be fetched from HTTP disk cache.
HTTP_MEMORY_CACHE_MAX_AGE_SEC	int32	HTTP Memory Cache Maximum Age (sec)	☐	Maximum age in seconds of HTTP memory cache entries to be used.
HTTP_MEMORY_CACHE_SAVE	char	Save HTTP Memory Cache	☐	Whether results can be stored in HTTP memory cache.
HTTP_MEMORY_CACHE_USE	char	Use HTTP Memory Cache	☐	Whether results can be fetched from HTTP memory cache.
HTTP_METHOD	varchar2(30)	HTTP Method	☐	
HTTP_STATUS_CODE	numeric	HTTP Status Code	☐	
ORIG_SYSTEM_GROUP	varchar2(4000)	Original System Group	☐	
ORIG_SYSTEM_REFERENCE	varchar2(4000)	Original System Reference	☐	
PAYLOAD_TEXT	varchar2	Payload	☐	
RESULT_BLOB	blob	Result BLOB	☐	
RESULT_DATE_TIME_UTC	datetime	Result Date Time	☐	
RESULT_NUMBER	number	Result Number	☐	
RESULT_TEXT	varchar2	Result Text	☐	
SUCCESSFUL	char	Succesful	☑	
TIMEOUT_SEC	int32	Timeout (sec)	☐	Timeout in seconds.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
TRANSACTION_ID	int64	Transaction ID	☑	Incrementing ID of the transaction.
URL	varchar2(4000)	URL	☐	

3.1.3 Schema: Views

3.1.3.1 Views

HourEntries2Years: Visma Severa Hour Entries Last 2 Years

Catalog: Severa

Schema: Views

Label: Hour Entries Last 2 Years

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

View Columns

The columns of the view HourEntries2Years are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	varchar2	Account GUID		
CASEBUSINESSUNITCODE	varchar2	Project Business Unit Code		
CASECOSTCENTERIDENTIFIER	varchar2	Project Cost Centre Identifier		
CASEGUID	varchar2	Project GUID		
CASENAME	varchar2	Project Name		
CASENUMBER	number	Project Number		
COSTCURRENCYSHORTFORM	varchar2	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Cost Currency Short Form of Organisation Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Cost Currency Short Form of User Company Currency		
cse_ACCOUNTGUID	varchar2	Account GUID		
cse_ACCOUNTNAME	varchar2	Account Name		
cse_ACCOUNTNUMBER	number	Account Number		
cse_BILLINGADDRESS	varchar2	Billing Address		
cse_BILLINGADDRESSGUID	varchar2	Billing Address GUID		
cse_BILLINGINTERNALNOTES	varchar2	Billing Internal Notes		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
cse_BILLINGNOTESBEFOREGRID	varchar2	Billing Notes before Grid		
cse_BUSINESSUNITGUID	varchar2	Business Unit GUID		
cse_BUSINESSUNITNAME	varchar2	Business Unit Name		
cse_BUSINESSUNITNUMBER	varchar2	Business Unit Number		
cse_CASECOMPANYCURRENCYGUID	varchar2	Project Company Currency GUID		
cse_CASENUMBER	number	Project Number		
cse_CASEOWNERUSERCODE	varchar2	Project Owner User Code		
cse_CASEOWNERUSERGUID	varchar2	Project Owner User GUID		
cse_CLOSEDDATE	datetime	Closed Date		
cse_COMPLETIONESTIMATE	number	Completion Estimate		
cse_CONTACTGUID	varchar2	Contact GUID		
cse_COSTCENTERGUID	varchar2	Cost Centre GUID		
cse_COSTCENTERNUMBER	varchar2	Cost Centre Number		
cse_CURRENCYGUID	varchar2	Currency GUID		
cse_CURRENCYSHORTFORM	varchar2	Currency Short Form		
cse_CURRENCYSHORTFORMOFCASECOMPANYCURRENCY	varchar2	Currency Short Form of Project Company Currency		
cse_CURRENTCASESTATUSDESCRIPTION	varchar2	Current Project Status Description		
cse_CURRENTCASESTATUSGUID	varchar2	Current Project Status GUID		
cse_CURRENTCASESTATUSINSERTTS	datetime	Current Project Status Inserted		
cse_DEADLINEDATE	datetime	Deadline Date		
cse_DESCRIPTION	varchar2	Ddescription		
cse_EXPECTEDVALUE	number	Expected Value		
cse_GUID	varchar2	GUID		
cse_INCLUDEEXCEPTIONS	boolean	Include Exceptions		
cse_INCLUDEOVERTIME	boolean	Include Overtime		
cse_INSERTTS	datetime	Inserted		
cse_INTERNALNAME	varchar2	Internal Name		
cse_INVOICETEMPLATEGUID	varchar2	Invoice Template GUID		
cse_ISCLOSED	boolean	Is Closed		
cse_ISDAILYALLOWANCEBILLABLE	boolean	Is Daily Allowance Billable		
cse_ISINTERNAL	boolean	Is Internal		
cse_ISMILEAGEBILLABLE	boolean	Is Mileage Billable		
cse_ISOTHERTRAVELEXPENSESBILLABLE	boolean	Are Other Travel Expenses Billable		
cse_LATESTESTIMATIONDATE	datetime	Latest Estimation Date		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
cse_LEADRATING	number	Lead Rating		
cse_LEADSOURCEGUID	varchar2	Lead Source GUID		
cse_NAME	varchar2	Name		
cse_ORDERNUMBER	varchar2	Order Number		
cse_OURREFERENCE	varchar2	Our Reference		
cse_OVERDUEINTEREST	number	Overdue Interest		
cse_PAYMENTTERM	number	Payment Term		
cse_PRICELISTGUID	varchar2	Pricelist GUID		
cse_PRICINGRULE	varchar2	Pricing Rule		
cse_PRICINGTYPE	varchar2	Pricing Type		
cse_PRIORITY	number	Priority		
cse_PROBABILITY	number	Probability		
cse_ROOTTASKGUID	varchar2	Root Task GUID		
cse_SALESCLOSEDATE	datetime	Sales Close Date		
cse_SALESPERSONUSERCODE	varchar2	Sales Person User Code		
cse_SALESPERSONUSERGUID	varchar2	Sales Person User GUID		
cse_SALESPROCESSGUID	varchar2	Sales Process GUID		
cse_SALESSTATUSGUID	varchar2	Sales Status GUID		
cse_SHAREDTOEXTRANET	boolean	Shared to Extranet		
cse_STARTDATE	datetime	Start Date		
cse_UPDATEDTS	datetime	Updated		
cse_URLTOCASE	varchar2	URL to Project		
cse_USEDEFAULTPRODUCTSINWORKTIMEENTRY	boolean	Use Default Products in Work Time Entry		
cse_USEDEFAULTWORKTYPES	boolean	Use Default Work Types		
cse_YOURREFERENCE	varchar2	Your Reference		
DESCRIPTION	varchar2	Ddescription		
ENDTIME	datetime	End Time		
EVENTDATE	datetime	Event Date		
GUID	varchar2	GUID		
INVOICEDESCRIPTION	varchar2	Invoice Description		
INVOICEGUID	varchar2	Invoice GUID		
INVOICEQUANTITY	number	Invoice Quantity		
INVOICEROWGUID	varchar2	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISPRODUCTIVE	boolean	Is Productive		
ISREADONLY	boolean	Is Read-only		
OVERTIMECODE	varchar2	Overtime Code		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
OVERTIMEGUID	varchar2	Overtime GUID		
PHASEGUID	varchar2	Phase GUID		
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY	varchar2	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	varchar2	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	varchar2	Price Currency Short Form of Organisation Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	varchar2	Price Currency Short Form of User Company Currency		
QUANTITY	number	Quantity		
STARTTIME	datetime	Start Time		
TASKGUID	varchar2	Task GUID		
UNITCOST	number	Unit Cost		
UNITCOSTININVOICINGCURRENCY	number	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	number	Unit Cost in Organisation Currency		
UNITCOSTINUSERCOMPANYCURRENCY	number	Unit Cost in User Company Currency		
UNITPRICE	number	Unit Price		
UNITPRICEINBASECURRENCY	number	Unit Price in Base Currency		
UNITPRICEINCASECOMPANYCURRENCY	number	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	number	Unit Price in User Company Currency		
USEINVOICEQUANTITY	boolean			
USERBUSINESSUNITCODE	varchar2	User Business Unit Code		
USERCODE	varchar2	User Code		
USERFIRSTNAME	varchar2	User First Name		
USERGUID	varchar2	User GUID		
USERLASTNAME	varchar2	User Last Name		
WORKTYPECODE	varchar2	Work Type Code		
WORKTYPEGUID	varchar2	Work Type GUID		

Invoices: Visma Severa Invoices

Catalog: Severa

Schema: Views

Label: Invoices

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

View Columns

The columns of the view Invoices are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	varchar2	Account GUID		
ACCOUNTNAME	varchar2	Account Name		
ACCOUNTNUMBER	number	Account Number		
BUSINESSUNITCODE	varchar2	Business Unit Code		
BUSINESSUNITGUID	varchar2	Business Unit GUID		
BUSINESSUNITNAME	varchar2	Business Unit Name		
BUYERORGANIZATION_BUYERCONTACTPERSONNAME	varchar2			
BUYERORGANIZATION_BUYERELECTRONICINVOICEADDRESS	varchar2			
BUYERORGANIZATION_BUYERELECTRONICINVOICEOPERATOR	varchar2			
BUYERORGANIZATION_BUYEREMAILADDRESSIDENTIFIER	varchar2			
BUYERORGANIZATION_BUYERORGANISATIONNAME	varchar2			
BUYERORGANIZATION_BUYERORGANISATIONTAXCODE	varchar2			
BUYERORGANIZATION_BUYERPARTYIDENTIFIER	varchar2			
BUYERORGANIZATION_BUYERPHONENUMBERIDENTIFIER	varchar2			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRY	varchar2			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRYCODE	varchar2			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	varchar2			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_STREETNAME	varchar2			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_TOWNNAME	varchar2			
CULTURE	varchar2	Culture		
CURRENCYGUID	varchar2	Currency GUID		
CURRENCYNAME	varchar2	Currency Name		
CURRENCYRATE	number	Currency Rate		
CURRENCYSHORTFORM	varchar2	Currency Short Form		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
DATE	datetime	Date		
DESCRIPTION	varchar2	Ddescription		
DUEDATE	datetime	Due Date		
ENTRYDATE	datetime	Entry Date		
GUID	varchar2	GUID		
INVOICEALLOWEDCASEGUIDS	varchar2			
INVOICECONFIG_BILLEXPENSE	varchar2			
INVOICECONFIG_BILLHOUR	varchar2			
INVOICECONFIG_BILLPRODUCT	varchar2			
INVOICECONFIG_EXPENSEITEMDESCRIPTIONFORMAT	varchar2			
INVOICECONFIG_HOURITEMDESCRIPTIONFORMAT	varchar2			
INVOICECONFIG_HOURPRIMARYGROUPBY	varchar2			
INVOICECONFIG_HOURSECONDARYGROUPBY	varchar2			
INVOICECONFIG_ISTAXFREE	boolean			
INVOICECONFIG_PRODUCTITEMDESCRIPTIONFORMAT	varchar2			
INVOICECONFIG_PRODUCTPRIMARYGROUPBY	varchar2			
INVOICECONFIG_PRODUCTSECONDARYGROUPBY	varchar2			
INVOICECONFIG_SHOWATTACHPRICEEXCLUDINGVAT	boolean			
INVOICECONFIG_SHOWATTACHUNIT	boolean			
INVOICECONFIG_SHOWATTACHUNITPRICE	boolean			
INVOICECONFIG_SHOWATTACHVAT	boolean			
INVOICECONFIG_SHOWLOGOONINVOICE	boolean			
INVOICECONFIG_SHOWPRICEEXCLUDINGVAT	boolean			
INVOICECONFIG_SHOWQUANTITY	boolean			
INVOICECONFIG_SHOWUNIT	boolean			
INVOICECONFIG_SHOWUNITPRICE	boolean			
INVOICECONFIG_SHOWVAT	boolean			
INVOICECONFIG_TAXFREEDESCRIPTION	varchar2			
INVOICECONFIG_TRAVELPRIMARYGROUPBY	varchar2			
INVOICECONFIG_TRAVELSECONDARYGROUPBY	varchar2			
INVOICEDETAILS_INVOICEDATE	datetime			

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
INVOICEDETAILS_INVOICEDUEDATE	datetime			
INVOICEDETAILS_INVOICENUMBER	number			
INVOICEDETAILS_INVOICETOTALVATAMOUNT	number			
INVOICEDETAILS_INVOICETOTALVATEXCLUDEDAMOUNT	number			
INVOICEDETAILS_INVOICETOTALVATINCLUDEDAMOUNT	number			
INVOICEDETAILS_INVOICETYPETEXT	varchar2			
INVOICEDETAILS_ORDERIDENTIFIER	varchar2			
INVOICEDETAILS_ORIGINCODE	varchar2			
INVOICEDETAILS_PAYMENTOVERDUEFINEFREETEXT	varchar2			
INVOICEDETAILS_PAYMENTOVERDUEFINEPERCENT	number			
INVOICEDETAILS_PAYMENTTERMSFREETEXT	varchar2			
INVOICEDETAILS_SELLERREFERENCEIDENTIFIER	varchar2			
INVOICEDETAILS_VATBASEAMOUNT	number			
INVOICEDETAILS_VATRATEAMOUNT	number			
INVOICEDETAILS_VATRATEPERCENT	number			
INVOICENUMBERINGBUSINESSUNITGUID	varchar2	Invoice Numbering Business Unit GUID		
INVOICEROWS	varchar2			
INVOICESTATUS	varchar2	Invoice Status		
ISCREDITNOTE	boolean	Is Credit Note		
ISREIMBURSED	boolean	Is Reimbursed		
ISTAXFREE	boolean	Is Tax Free		
LANGUAGECODE	varchar2	Language Code		
LANGUAGEGUID	varchar2	Language GUID		
MAINCASEGUID	varchar2	Main Case GUID		
NOTES	varchar2	Notes		
NOTESAFTERINVOICEROWS	varchar2	Notes after Invoice Lines		
NOTESBEFOREINVOICEROWS	varchar2	Notes before Invoice Lines		
NUMBER	number	Number		
ORDERNUMBER	varchar2	Order Number		
OURREFERENCE	varchar2	Our Reference		
OVERDUEINTEREST	number	Overdue Interest		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
PAYMENTDATE	datetime	Payment Date		
PAYMENTSTATUS	varchar2	Payment Status		
PAYMENTTERM	number	Payment Term		
PONUMBER	varchar2	PO Number		
RECEIVERADDRESSLINE	varchar2	Receiver Addressline		
RECEIVERCITY	varchar2	Receiver City		
RECEIVERCONTACTEMAIL	varchar2	Receiver Contact Email		
RECEIVERCONTACTNAME	varchar2	Receiver Contact Name		
RECEIVERCOUNTRY	varchar2	Receiver Country		
RECEIVERCUSTOMERNAME	varchar2	Receiver Customer Name		
RECEIVERPOSTALCODE	varchar2	Receiver Postal Code		
RECEIVERSTATE	varchar2	Receiver State		
RECEIVERVATNUMBER	varchar2	Receiver VAT Number		
REFERENCENUMBER	varchar2	Reference Number		
REIMBURSEINVOICEGUID	varchar2	Reimburse Invoice GUID		
SELLERCONTACT_SELLERACCOUNTDETAILS	varchar2			
SELLERCONTACT_SELLERCONTACTPERSONNAME	varchar2			
SELLERCONTACT_SELLEREMAILADDRESSIDENTIFIER	varchar2			
SELLERCONTACT_SELLERFAXNUMBER	varchar2			
SELLERCONTACT_SELLERFREETEXT	varchar2			
SELLERCONTACT_SELLERHOMETOWNNAME	varchar2			
SELLERCONTACT_SELLERPHONENUMBER	varchar2			
SELLERCONTACT_SELLERWEBADDRESSIDENTIFIER	varchar2			
SELLERORGANIZATION_SELLEREINVOICEADDRESS	varchar2			
SELLERORGANIZATION_SELLEREINVOICEOPERATOR	varchar2			
SELLERORGANIZATION_SELLERORGANISATIONNAME	varchar2			
SELLERORGANIZATION_SELLERORGANISATIONTAXCODE	varchar2			
SELLERORGANIZATION_SELLERPARTYIDENTIFIER	varchar2			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRY	varchar2			

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRYCODE	varchar2			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	varchar2			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_STREETNAME	varchar2			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_TOWNNAME	varchar2			
SENDERADDRESSLINE	varchar2	Sender Addressline		
SENDERCITY	varchar2	Sender City		
SENDERCONTACTNAME	varchar2	Sender Contact Name		
SENDERCOUNTRY	varchar2	Sender Country		
SENDERNAME	varchar2	Sender Name		
SENDERPOSTALCODE	varchar2	Sender Postal Code		
SENDERSTATE	varchar2	Sender State		
SENDERVATNUMBER	varchar2	Sender VAT Number		
TAXFREEDESCRIPTION	varchar2	Tax Free Description		
TITLE	varchar2	Title		
VIRTUALBANKBARCODE	varchar2	Virtual Bank Barcode		
YOURREFERENCE	varchar2	Your Reference		

4 Package: dcr_metadata

4.1 Procedures

4.1.1 dcr_metadata.get_partitions: Visma Severa Data container metadata package

Get all partitions.

Documentation:

List all partitions.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Index

- A -

Access Rights Profile by GUID 16
Access Rights Profiles 17
accessRightsProfile_GUID 43, 329
accessRightsProfile_IsActive 43, 329
accessRightsProfile_IsDefault 43, 329
accessRightsProfile_Name 43, 329
accessRightsProfile_Rights_Account 43, 329
accessRightsProfile_Rights_AccountDelete 43, 329
accessRightsProfile_Rights_Administrator 43, 329
accessRightsProfile_Rights_Case 43, 329
accessRightsProfile_Rights_CaseDelete 43, 329
accessRightsProfile_Rights_CaseOwnerWorkHourApproval 43, 329
accessRightsProfile_Rights_ScheduleJobs 43, 329
accessRightsProfile_Rights_Sharing 43, 329
accessRightsProfile_Rights_TravelReimbursement 43, 329
accessRightsProfile_Rights_Users 43, 329
accessRightsProfile_Rights_WorkHourApproval 43, 329
AccessRightsProfileByGUID 16
AccessRightsProfiles 17
Account by GUID 18
Account by Name 19
Account by Number 20
Account by VAT Number 22
Account Group by Account GUID 24
Account Groups 23
Account GUID 32, 34, 36, 50, 61, 67, 80, 120, 122, 125, 129, 132, 135, 140, 141, 143, 144, 149, 189, 192, 194, 196, 199, 201, 203, 207, 212, 217, 222, 227, 232, 238, 277, 280, 282, 334, 342, 348, 353, 394, 397
Account Name 61, 120, 122, 125, 129, 132, 135, 149, 203, 207, 212, 222, 227, 232, 238, 342, 394, 397
Account Number 61, 120, 122, 125, 129, 132, 135, 149, 207, 212, 222, 227, 232, 238, 342, 394, 397
Account Owner User GUID 18, 19, 20, 22, 26, 27, 29, 30, 44, 70, 158, 160, 161, 331, 349
Account Rating 18, 19, 20, 22, 26, 27, 29, 30, 44, 70, 158, 160, 161, 331, 349
AccountByGUID 18
AccountByName 19
AccountByNumber 20
AccountByVatNumber 22
ACCOUNTGROUPGUIDS 18, 19, 20, 22, 26, 27, 29, 30, 44, 49, 70, 158, 160, 161, 289, 331, 349
accountGroupInfo_GUID 48
accountGroupInfo_IsActive 48
accountGroupInfo_Name 48
accountGroupInfo_ParentGUID 48
AccountGroups 23
AccountGroupsByAccountGUID 24
ACCOUNTGUID 18, 24, 32, 34, 36, 49, 50, 61, 67, 80, 111, 120, 122, 125, 129, 132, 135, 140, 141, 143, 144, 149, 158, 187, 188, 189, 192, 194, 196, 199, 201, 203, 207, 212, 217, 222, 227, 232, 238, 277, 280, 282, 289, 334, 342, 348, 353, 394, 397
accountInfo_AccountGroupGUIDs 44, 331
accountInfo_AccountOwnerUserGUID 44, 331
accountInfo_AccountRating 44, 331
accountInfo_CompanyGUID 44, 331
accountInfo_CurrencyCode 44, 331
accountInfo_CurrencyGUID 44, 331
accountInfo_eInvoiceAddress 44, 331
accountInfo_eInvoiceOperator 44, 331
accountInfo_GUID 44, 331
accountInfo_InsertTS 44, 331
accountInfo_InvoicingVat 44, 331
accountInfo_IsActive 44, 331
accountInfo_IsInternal 44, 331
accountInfo_IsPrivatePerson 44, 331
accountInfo_LanguageCode 44, 331
accountInfo_LanguageGUID 44, 331
accountInfo_Name 44, 331
accountInfo_Notes 44, 331
accountInfo_Number 44, 331
accountInfo_OverdueInterest 44, 331
accountInfo_PaymentTerm 44, 331
accountInfo_PricelistGUID 44, 331
accountInfo_ReverseCharge 44, 331
accountInfo_ReverseChargeDescription 44, 331
accountInfo_UpdatedTS 44, 331
accountInfo_UrlToAccount 44, 331
ACCOUNTNAME 19, 61, 120, 122, 125, 129, 132, 135, 149, 203, 207, 212, 222, 227, 232, 238, 342, 397
ACCOUNTNUMBER 20, 61, 120, 122, 125, 129, 132, 135, 149, 207, 212, 222, 227, 232, 238, 342, 397
ACCOUNTOWNERUSERGUID 18, 19, 20, 22, 26, 27, 29, 30, 44, 70, 158, 160, 161, 331, 349
ACCOUNTRATING 18, 19, 20, 22, 26, 27, 29, 30, 44, 70, 158, 160, 161, 331, 349
Accounts 25
Accounts by Company GUID 26
Accounts by Insert Date 27
Accounts Changed since 29
Accounts Changed since Opts. 30
AccountsByCompanyGUID 26
AccountsByInsertDate 27

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

AccountsChangedSince 29
AccountsChangedSinceOpts 30
Active Lead Sources 32
ActiveLeadSources 32
ACTIVITIES 387
Activities by Date 32
ActivitiesByDate 32
Activity by GUID 34
Activity Contact Participants 35
Activity Instances 36
Activity Participants 38
Activity Resource Participants 38
Activity Type by GUID 39
Activity Type Code 32, 34, 36, 50, 334
Activity Type GUID 32, 34, 36, 50, 334
Activity Types 40
Activity User Participants 41
ActivityByGUID 34
activityCategory 32
ActivityContactParticipants 35
activityGUID 34, 35, 38, 41, 52, 53, 55, 164, 290, 291, 292
activityInfo_AccountGUID 50, 334
activityInfo_ActivityTypeCode 50, 334
activityInfo_ActivityTypeGUID 50, 334
activityInfo_CaseGUID 50, 334
activityInfo_EndDateTime 50, 334
activityInfo_GUID 50, 334
activityInfo_IsAllDay 50, 334
activityInfo_IsClosed 50, 334
activityInfo_IsDuration 50, 334
activityInfo_IsPrivate 50, 334
activityInfo_IsReadOnly 50, 334
activityInfo_IsUnassigned 50, 334
activityInfo_Location 50, 334
activityInfo_Name 50, 334
activityInfo_Notes 50, 334
activityInfo_OwnerUserBusinessUnitCode 50, 334
activityInfo_OwnerUserCode 50, 334
activityInfo_OwnerUserFirstName 50, 334
activityInfo_OwnerUserGUID 50, 334
activityInfo_OwnerUserLastName 50, 334
activityInfo_StartDateTime 50, 334
activityInfo_TaskGUID 50, 334
ActivityInstances 36
ActivityParticipants 38
ActivityResourceParticipants 38
ActivityTypeByGUID 39
ACTIVITYTYPECODE 32, 34, 36, 50, 334
ACTIVITYTYPEGUID 32, 34, 36, 39, 50, 165, 334
activityTypeInfo_Category 54, 336
activityTypeInfo_Code 54, 336
activityTypeInfo_GUID 54, 336
activityTypeInfo_Icon 54, 336
activityTypeInfo_IsActive 54, 336
activityTypeInfo_IsDefault 54, 336
activityTypeInfo_IsPaidLeave 54, 336
activityTypeInfo_Name 54, 336
ActivityTypes 40
ActivityUserParticipants 41
Add New Access Rights Profile 43
Add New Account 44
Add New Account Group 48
Add New Account Group Members 49
Add New Activity 50
Add New Activity Contact Participant 52
Add New Activity Resource Participant 53
Add New Activity Type 54
Add New Activity User Participant 55
Add New Address 56
Add New Billing and Expense Forecast to Project 57
Add New Business Unit 59
Add New Contact 67
Add New Contact Tag 69
Add New Customer 70
Add New Employment 73
Add New Expense to Product by PO Number 74
Add New File Tag 79
Add New Hour Entry 80
Add New Industry 84
Add New Invoice Log Entry 85
Add New Item 86
Add New Items 91
Add New Phase 94
Add New Product 96
Add New Product Category 97
Add New Project 61
Add New Project Status 65
Add New Project Tag 66
Add New Resource 98
Add New Resource Allocation 99
Add New Resource Allocations 101
Add New Sales Process 102
Add New Sales Status 103
Add New Travel Expense 104
Add New User 105
Add New User Tag 108
Add New Work Type 109
Add Project Member User 42
AddCaseMemberUser 42
AddNewAccessRightsProfile 43

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

AddNewAccount 44
AddNewAccountGroup 48
AddNewAccountGroupMembers 49
AddNewActivity 50
AddNewActivityContactParticipant 52
ADDNEWACTIVITYCONTACTPARTICIPANTRESULT 52
AddNewActivityResourceParticipant 53
ADDNEWACTIVITYRESOURCEPARTICIPANTRESULT 53
AddNewActivityType 54
AddNewActivityUserParticipant 55
ADDNEWACTIVITYUSERPARTICIPANTRESULT 55
AddNewAddress 56
AddNewBillingAndExpenseForecastToCase 57
AddNewBusinessUnit 59
AddNewCase 61
AddNewCaseStatus 65
AddNewCaseTag 66
AddNewContact 67
AddNewContactTag 69
AddNewCustomer 70
AddNewEmployment 73
AddNewExpenseToProductByPoNumber 74
AddNewFileTag 79
AddNewHourEntry 80
AddNewIndustry 84
AddNewInvoiceLogEntry 85
AddNewItem 86
AddNewItems 91
AddNewPhase 94
AddNewProduct 96
AddNewProductCategory 97
AddNewResource 98
AddNewResourceAllocation 99
AddNewResourceAllocations 101
AddNewSalesProcess 102
AddNewSalesStatus 103
AddNewTravelExpense 104
AddNewUser 105
AddNewUserTag 108
AddNewWorkType 109
Address by GUID 110
Address GUID 67, 140, 141, 143, 144, 348
AddressByGUID 110
Addresses Changed since 111
AddressesChangedSince 111
ADDRESSGUID 67, 110, 140, 141, 143, 144, 348
addressInfo_Addressline 44, 56, 337
addressInfo_Addressline2 44, 56, 337
addressInfo_Addressline3 44, 56, 337
addressInfo_City 44, 56, 337
addressInfo_CompanyGUID 44, 56, 337
addressInfo_CountryCode 44, 56, 337
addressInfo_CountryGUID 44, 56, 337
addressInfo_CountryName 44, 56, 337
addressInfo_CountryRegionGUID 44, 56, 337
addressInfo_CountryRegionName 44, 56, 337
addressInfo_Fax 44, 56, 337
addressInfo_GUID 44, 56, 337
addressInfo_IsBillingAddress 44, 56, 337
addressInfo_IsPostalAddress 44, 56, 337
addressInfo_IsVisitAddress 44, 56, 337
addressInfo_Phone 44, 56, 337
addressInfo_PostalCode 44, 56, 337
ADDRESSLINE 56, 105, 110, 111, 113, 128, 187, 188, 337, 366, 371, 373, 375, 377, 379, 381, 382, 384
Addressline 2 56, 110, 111, 113, 187, 188, 337
Addressline 3 56, 110, 111, 113, 187, 188, 337
ADDRESSLINE2 56, 110, 111, 113, 187, 188, 337
ADDRESSLINE3 56, 110, 111, 113, 187, 188, 337
Advance Payment 322, 323, 325, 326, 328
ADVANCEPAYMENT 322, 323, 325, 326, 328
Allocated Hours 99, 101, 293, 294, 295, 296, 297, 298, 363
Allocated Percentage 99, 101, 293, 294, 295, 296, 297, 298, 363
ALLOCATEDHOURS 99, 101, 293, 294, 295, 296, 297, 298, 363
ALLOCATEDPERCENTAGE 99, 101, 293, 294, 295, 296, 297, 298, 363
analysis-enforce-row-uniqueness 2
Annual Revenue 70, 139, 158, 160, 161, 349
ANNUALREVENUE 70, 139, 158, 160, 161, 349
ANNUALREVENUE2 70, 139, 158, 160, 161, 349
API 369, 376
api-url 2
Approval Date 322, 323, 325, 326, 328
APPROVALDATE 322, 323, 325, 326, 328
Approved by User GUID 322, 323, 325, 326, 328
APPROVEDBYUSERGUID 322, 323, 325, 326, 328
Are Other Travel Expenses Billable 61, 120, 122, 125, 129, 132, 135, 342, 394
attachmentGuid 205

- B -

Bank Account Number 105, 128, 366, 371, 373, 375, 377, 379, 381, 382, 384
BANKACCOUNTNUMBER 105, 128, 366, 371, 373, 375, 377, 379, 381, 382, 384
Billing Address 61, 113, 120, 122, 125, 129, 132, 135, 342, 394

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Billing Address GUID 61, 120, 122, 125, 129, 132, 135, 342, 394
Billing and Expense Forecast by Project GUID 113
Billing Dependency Task GUID 74, 86, 91, 243, 246, 249, 251, 254, 284, 287, 317
Billing Forecast 57, 113, 339
Billing Internal Notes 61, 120, 122, 125, 129, 132, 135, 342, 394
Billing Notes before Grid 61, 120, 122, 125, 129, 132, 135, 342, 394
Billing Schedule 74, 86, 91, 243, 246, 249, 251, 254, 284, 287, 317
BILLINGADDRESS 61, 113, 120, 122, 125, 129, 132, 135, 342
BILLINGADDRESSGUID 61, 120, 122, 125, 129, 132, 135, 342
BillingAndExpenseForecastByCaseGUID 113
billingAndExpenseForecastInfo_BillingForecast 57, 339
billingAndExpenseForecastInfo_ExpenseForecast 57, 339
billingAndExpenseForecastInfo_GUID 57, 339
billingAndExpenseForecastInfo_InhouseRevenueForecast 57, 339
billingAndExpenseForecastInfo_Month 57, 339
billingAndExpenseForecastInfo_Notes 57, 339
billingAndExpenseForecastInfo_OutsourcingRevenueForecast 57, 339
billingAndExpenseForecastInfo_Year 57, 339
BILLINGDEPENDENCYTASKGUID 74, 86, 91, 243, 246, 249, 251, 254, 284, 287, 317
BILLINGFORECAST 57, 113, 339
BILLINGINTERNALNOTES 61, 120, 122, 125, 129, 132, 135, 342
BILLINGNOTESBEFOREGRID 61, 120, 122, 125, 129, 132, 135, 342
BILLINGSCHEDULE 74, 86, 91, 243, 246, 249, 251, 254, 284, 287, 317
BLOB Preferred 392
BLOB_PREFERRED 392
BOL_RESPONSE_CACHE_MAX_AGE_SEC 392
bulk-delete-page-size-rows 2
bulk-insert-page-size-bytes 2
bulk-insert-page-size-rows 2
Business Unit by Code 114
Business Unit by GUID 116
Business Unit Code 105, 128, 149, 207, 212, 222, 227, 232, 238, 366, 371, 373, 375, 377, 379, 381, 382, 384, 397
Business Unit GUID 61, 105, 120, 122, 125, 128, 129, 132, 135, 149, 207, 212, 222, 227, 232, 238, 342, 366, 371, 373, 375, 377, 379, 381, 382, 384, 394, 397
Business Unit Hierarchy 117
Business Unit Name 61, 105, 120, 122, 125, 128, 129, 132, 135, 149, 207, 212, 222, 227, 232, 238, 342, 366, 371, 373, 375, 377, 379, 381, 382, 384, 394, 397
Business Unit Number 61, 120, 122, 125, 129, 132, 135, 342, 394
Business Units Changed since 118
BusinessUnitByCode 114
BusinessUnitByGUID 116
BUSINESSUNITCODE 105, 114, 128, 149, 207, 212, 222, 227, 232, 238, 366, 371, 373, 375, 377, 379, 381, 382, 384, 397
BUSINESSUNITGUID 32, 61, 105, 116, 118, 120, 122, 125, 128, 129, 132, 135, 149, 185, 186, 192, 194, 196, 207, 212, 222, 227, 232, 238, 266, 277, 317, 326, 342, 366, 371, 373, 375, 377, 379, 381, 382, 384, 387, 397
businessUnitGUIDs 294
BusinessUnitHierarchy 117
businessUnitInfo_Code 59, 340
businessUnitInfo_CostCenterGUID 59, 340
businessUnitInfo_CurrencyCode 59, 340
businessUnitInfo_CurrencyGUID 59, 340
businessUnitInfo_FormattingCulture 59, 340
businessUnitInfo_GUID 59, 340
businessUnitInfo_InsertTS 59, 340
businessUnitInfo_IsActive 59, 340
businessUnitInfo_LanguageCode 59, 340
businessUnitInfo_LanguageGUID 59, 340
businessUnitInfo_Name 59, 340
businessUnitInfo_NextInvoiceNumber 59, 340
businessUnitInfo_OverDueInterest 59, 340
businessUnitInfo_ParentBusinessUnitGUID 59, 340
businessUnitInfo_PaymentTerm 59, 340
businessUnitInfo_ShowLogoOnInvoices 59, 340
businessUnitInfo_ShowLogoOnPdfReports 59, 340
businessUnitInfo_TimeZoneGUID 59, 340
businessUnitInfo_UpdatedTS 59, 340
businessUnitInfo_VatNumber 59, 340
BUSINESSUNITNAME 61, 105, 120, 122, 125, 128, 129, 132, 135, 149, 207, 212, 222, 227, 232, 238, 342, 366, 371, 373, 375, 377, 379, 381, 382, 384, 397
BUSINESSUNITNUMBER 61, 120, 122, 125, 129, 132, 135, 342
BusinessUnitsChangedSince 118
BUYERORGANIZATION_BUYERCONTACTPERSONNAME 149, 207, 212, 222, 227, 232, 238, 397
BUYERORGANIZATION_BUYERELECTRONICINVOICEADDRESS 149, 207, 212, 222, 227, 232, 238, 397
BUYERORGANIZATION_BUYERELECTRONICINVOICEOPERATOR 149, 207, 212, 222, 227, 232, 238, 397

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

BUYERORGANIZATION_BUYEREMAILADDRESSIDENTIFIER 149, 207, 212, 222, 227, 232, 238, 397
BUYERORGANIZATION_BUYERORGANISATIONNAME 149, 207, 212, 222, 227, 232, 238, 397
BUYERORGANIZATION_BUYERORGANISATIONTAXCODE 149, 207, 212, 222, 227, 232, 238, 397
BUYERORGANIZATION_BUYERPARTYIDENTIFIER 149, 207, 212, 222, 227, 232, 238, 397
BUYERORGANIZATION_BUYERPHONENUMBERIDENTIFIER 149, 207, 212, 222, 227, 232, 238, 397
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRY 149, 207, 212, 222, 227, 232, 238, 397
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRYCODE 149, 207, 212, 222, 227, 232, 238, 397
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER 149, 207, 212, 222, 227, 232, 238, 397
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_STREETNAME 149, 207, 212, 222, 227, 232, 238, 397
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_TOWNNAME 149, 207, 212, 222, 227, 232, 238, 397

- C -

CASEBUSINESSUNITCODE 80, 189, 192, 194, 196, 199, 201, 217, 277, 280, 282, 353, 394
CaseByGUID 120
CaseByNumber 122
CaseByTaskGUID 125
CASECOMPANYCURRENCYGUID 61, 120, 122, 125, 129, 132, 135, 342
CASECOSTCENTERIDENTIFIER 80, 189, 192, 194, 196, 199, 201, 217, 277, 280, 282, 353, 394
CASEGUID 32, 34, 36, 42, 50, 57, 66, 74, 80, 86, 91, 94, 99, 101, 103, 113, 120, 128, 166, 180, 182, 189, 192, 194, 196, 199, 201, 203, 217, 243, 246, 248, 251, 254, 261, 263, 265, 266, 277, 280, 282, 284, 287, 293, 294, 295, 296, 297, 298, 302, 303, 309, 317, 334, 353, 358, 363, 394
caseGUIDs 294
caseInfo_AccountGUID 61, 342
caseInfo_AccountName 61, 342
caseInfo_AccountNumber 61, 342
caseInfo_BillingAddress 61, 342
caseInfo_BillingAddressGUID 61, 342
caseInfo_BillingInternalNotes 61, 342
caseInfo_BillingNotesBeforeGrid 61, 342
caseInfo_BusinessUnitGUID 61, 342
caseInfo_BusinessUnitName 61, 342
caseInfo_BusinessUnitNumber 61, 342
caseInfo_CaseCompanyCurrencyGUID 61, 342
caseInfo_CaseNumber 61, 342
caseInfo_CaseOwnerUserCode 61, 342
caseInfo_CaseOwnerUserGUID 61, 342
caseInfo_ClosedDate 61, 342
caseInfo_CompletionEstimate 61, 342
caseInfo_ContactGUID 61, 342
caseInfo_CostCenterGUID 61, 342
caseInfo_CostCenterNumber 61, 342
caseInfo_CurrencyGUID 61, 342
caseInfo_CurrencyShortform 61, 342
caseInfo_CurrencyShortformOfCaseCompanyCurrency 61, 342
caseInfo_CurrentCaseStatusDescription 61, 342
caseInfo_CurrentCaseStatusGUID 61, 342
caseInfo_CurrentCaseStatusInsertTs 61, 342
caseInfo_DeadlineDate 61, 342
caseInfo_Description 61, 342
caseInfo_ExpectedValue 61, 342
caseInfo_GUID 61, 342
caseInfo_IncludeExceptions 61, 342
caseInfo_IncludeOverTime 61, 342
caseInfo_InsertTs 61, 342
caseInfo_InternalName 61, 342
caseInfo_InvoiceTemplateGUID 61, 342
caseInfo_IsClosed 61, 342
caseInfo_IsDailyAllowanceBillable 61, 342
caseInfo_IsInternal 61, 342
caseInfo_IsMileageBillable 61, 342
caseInfo_IsOtherTravelExpensesBillable 61, 342
caseInfo_LatestEstimationDate 61, 342
caseInfo_LeadRating 61, 342
caseInfo_LeadSourceGUID 61, 342
caseInfo_Name 61, 342
caseInfo_OrderNumber 61, 342
caseInfo_OurReference 61, 342
caseInfo_OverdueInterest 61, 342
caseInfo_PaymentTerm 61, 342
caseInfo_PricelistGUID 61, 342
caseInfo_PricingRule 61, 342
caseInfo_PricingType 61, 342
caseInfo_Priority 61, 342
caseInfo_Probability 61, 342
caseInfo_RootTaskGUID 61, 342
caseInfo_SalesCloseDate 61, 342
caseInfo_SalesPersonUserCode 61, 342
caseInfo_SalesPersonUserGUID 61, 342
caseInfo_SalesProcessGUID 61, 342
caseInfo_SalesStatusGUID 61, 342
caseInfo_SharedToExtranet 61, 342
caseInfo_StartDate 61, 342

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

caseInfo_UpdatedTs 61, 342
caseInfo_UrlToCase 61, 342
caseInfo_UseDefaultProductsInWorkTimeEntry 61, 342
caseInfo_UseDefaultWorkTypes 61, 342
caseInfo_YourReference 61, 342
CaseMemberUsersByCaseGUID 128
CASENAME 80, 189, 192, 194, 196, 199, 201, 217, 277, 280, 282, 353, 394
CASENUMBER 61, 74, 80, 86, 91, 120, 122, 125, 129, 132, 135, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 342, 353, 394
CASEOWNERUSERCODE 61, 120, 122, 125, 129, 132, 135, 342
CASEOWNERUSERGUID 61, 120, 122, 125, 129, 132, 135, 342
CASERIGHTS_EDITCASESTATUS 182
CASERIGHTS_SHOWBILLING 182
CASERIGHTS_SHOWCASESTATUS 182
CASERIGHTS_SHOWFILESANDLINKS 182
CASERIGHTS_SHOWGANTTCHART 182
CASERIGHTS_SHOWPHASETREE 182
CASERIGHTS_SHOWREADYTOBILL 182
CASERIGHTS_SHOWWORKHOURESTIMATES 182
CASERIGHTS_SHOWWORKHOURS 182
CASERIGHTS_SHOWWORKHOURSTARTANDENDTIME 182
CASES 129, 219
CasesByAccountGUID 132
CasesChangedSince 135
CaseStatusByGUID 137
CaseStatuses 138
caseStatusGUID 137, 167
caseStatusGUIDs 294
caseStatusInfo_GUID 65, 347
caseStatusInfo_Icon 65, 347
caseStatusInfo_IsActive 65, 347
caseStatusInfo_Name 65, 347
caseTagGUIDs 294
CaseTags 139
CATEGORY 39, 40, 54, 336
Child Phases 203
CHILDPHASES 203
CITY 56, 110, 111, 113, 187, 188, 337
Closed Date 61, 120, 122, 125, 129, 132, 135, 342, 394
CLOSEDDATE 61, 120, 122, 125, 129, 132, 135, 342
CODE 39, 40, 54, 59, 84, 94, 96, 97, 104, 105, 109, 114, 116, 117, 118, 128, 204, 260, 261, 263, 265, 266, 269, 270, 271, 272, 273, 274, 276, 314, 315, 316, 320, 336, 340, 358, 360, 361, 364, 366, 370, 371, 373, 375, 377, 379, 381, 382, 384, 388, 389, 390, 391
COMMUNICATIONMETHODS 67, 140, 141, 143, 144, 348
Company Business Unit Code 73, 177, 178, 352
Company Business Unit GUID 73, 177, 178, 352
Company Business Unit Name 73, 177, 178, 352
Company by GUID 139
Company GUID 18, 19, 20, 22, 26, 27, 29, 30, 44, 56, 70, 110, 111, 113, 158, 160, 161, 187, 188, 331, 337, 349
COMPANYBUSINESSUNITCODE 73, 177, 178, 352
COMPANYBUSINESSUNITGUID 73, 177, 178, 352
COMPANYBUSINESSUNITNAME 73, 177, 178, 352
CompanyByGUID 139
COMPANYGUID 18, 19, 20, 22, 26, 27, 29, 30, 44, 56, 70, 110, 111, 113, 139, 158, 160, 161, 187, 188, 331, 337, 349
companyInfo_AnnualRevenue 44, 331
companyInfo_AnnualRevenue2 44, 331
companyInfo_Email 44, 331
companyInfo_Employees 44, 331
companyInfo_GUID 44, 331
companyInfo_HeadOfficeAddressGUID 44, 331
companyInfo_HierarchyGUID 44, 331
companyInfo_IndustryGUID 44, 331
companyInfo_KvkNumber 44, 331
companyInfo_LanguageGUID 44, 331
companyInfo_Name 44, 331
companyInfo_TimezoneGUID 44, 331
companyInfo_VatNumber 44, 331
companyInfo_Website 44, 331
Completion Estimate 61, 120, 122, 125, 129, 132, 135, 342, 394
COMPLETIONESTIMATE 61, 120, 122, 125, 129, 132, 135, 342
Contact by GUID 140
Contact GUID 35, 38, 41, 61, 120, 122, 125, 129, 132, 135, 342, 394
Contact Role GUID 67, 140, 141, 143, 144, 348
ContactByGUID 140
CONTACTGUID 35, 38, 41, 52, 61, 69, 120, 122, 125, 129, 132, 135, 140, 168, 182, 290, 342
contactInfo_AccountGUID 44, 67, 348
contactInfo_AddressGUID 44, 67, 348
contactInfo_CommunicationMethods 44, 67, 348
contactInfo_ContactRoleGUID 44, 67, 348
contactInfo_DateOfBirth 44, 67, 348

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

contactInfo_Description 44, 67, 348
contactInfo_FirstName 44, 67, 348
contactInfo_GUID 44, 67, 348
contactInfo_IsActive 44, 67, 348
contactInfo_JobTitle 44, 67, 348
contactInfo_Keywords 44, 67, 348
contactInfo_LanguageGUID 44, 67, 348
contactInfo_LastName 44, 67, 348
contactInfo_Salutation 44, 67, 348
contactInfo_SatisfactionLevel 44, 67, 348
contactInfo_TimeZoneGUID 44, 67, 348
CONTACTRIGHTS_APPROVAL 182
CONTACTRIGHTS_COMMENTS 182
CONTACTRIGHTS_FILEUPLOAD 182
CONTACTROLEGUID 67, 140, 141, 143, 144, 348
CONTACTS 180
Contacts by Account GUID 141
Contacts by Communication Method Type 143
Contacts Changed since 144
ContactsByAccountGUID 141
ContactsByCommunicationMethodType 143
ContactsChangedSince 144
Content Type 183, 185, 321, 392
CONTENT_TYPE 392
CONTENTTYPE 183, 185, 321
CONTEXT 66, 69, 79, 85, 108, 139, 308, 309, 386
Cost Centre by GUID 145
Cost Centre GUID 59, 61, 74, 86, 91, 114, 116, 117, 118, 120, 122, 125, 129, 132, 135, 243, 246, 249, 251, 254, 284, 287, 317, 340, 342, 394
Cost Centre Identifier 74, 86, 91, 243, 246, 249, 251, 254, 284, 287, 317
Cost Centre Number 61, 120, 122, 125, 129, 132, 135, 342, 394
Cost Centres 146
Cost Currency GUID 104, 314, 315, 316, 320, 364
Cost Currency Short Form 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 353, 394
Cost Currency Short Form of Invoicing Currency 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 353, 394
Cost Currency Short Form of Organisation Currency 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 353, 394
Cost Currency Short Form of User Company Currency 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 353, 394
CostCenterByGUID 145
COSTCENTERGUID 59, 61, 74, 86, 91, 114, 116, 117, 118, 120, 122, 125, 129, 132, 135, 145, 243, 246, 249, 251, 254, 284, 287, 317, 340, 342
COSTCENTERIDENTIFIER 74, 86, 91, 243, 246, 249, 251, 254, 284, 287, 317
COSTCENTERNUMBER 61, 120, 122, 125, 129, 132, 135, 342
CostCenters 146
COSTCURRENCYGUID 104, 314, 315, 316, 320, 364
COSTCURRENCYSHORTFORM 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 353, 394
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 353, 394
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 353, 394
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 353, 394
count 25, 160
Country by GUID 146
Country by ISO 147
Country by Name 148
Country Code 56, 105, 110, 111, 113, 128, 187, 188, 337, 366, 371, 373, 375, 377, 379, 381, 382, 384
Country GUID 56, 105, 110, 111, 113, 128, 187, 188, 337, 366, 371, 373, 375, 377, 379, 381, 382, 384
Country Name 56, 110, 111, 113, 187, 188, 337
Country Region GUID 56, 105, 110, 111, 113, 128, 187, 188, 337, 366, 371, 373, 375, 377, 379, 381, 382, 384
Country Region Name 56, 105, 110, 111, 113, 128, 187, 188, 337, 366, 371, 373, 375, 377, 379, 381, 382, 384
CountryByGUID 146
CountryByISO 147
CountryByName 148
COUNTRYCODE 56, 105, 110, 111, 113, 128, 187, 188, 337, 366, 371, 373, 375, 377, 379, 381, 382, 384
COUNTRYGUID 56, 105, 110, 111, 113, 128, 146, 187, 188, 337, 366, 371, 373, 375, 377, 379, 381, 382, 384
COUNTRYNAME 56, 110, 111, 113, 148, 187, 188, 337
COUNTRYREGIONGUID 56, 105, 110, 111, 113, 128, 187, 188, 337, 366, 371, 373, 375, 377, 379, 381, 382, 384

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

COUNTRYREGIONNAME 56, 105, 110, 111, 113, 128, 187, 188, 337, 366, 371, 373, 375, 377, 379, 381, 382, 384
Create Invoices 149
CreateInvoices 149
criteria_AccountGroupGuids 160
criteria_InternalName 129
criteria_OnlyActiveCases 129
criteria_OnlyCustomerCases 129
criteria_OnlyInactiveCases 129
criteria_OnlyInternalCases 129
criteria_SalesProcess 129
criteria_Tags 129
cse_ACCOUNTGUID 394
cse_ACCOUNTNAME 394
cse_ACCOUNTNUMBER 394
cse_BILLINGADDRESS 394
cse_BILLINGADDRESSGUID 394
cse_BILLINGINTERNALNOTES 394
cse_BILLINGNOTESBEFOREGRID 394
cse_BUSINESSUNITGUID 394
cse_BUSINESSUNITNAME 394
cse_BUSINESSUNITNUMBER 394
cse_CASECOMPANYCURRENCYGUID 394
cse_CASENUMBER 394
cse_CASEOWNERUSERCODE 394
cse_CASEOWNERUSERGUID 394
cse_CLOSEDDATE 394
cse_COMPLETIONESTIMATE 394
cse_CONTACTGUID 394
cse_COSTCENTERGUID 394
cse_COSTCENTERNUMBER 394
cse_CURRENCYGUID 394
cse_CURRENCYSHORTFORM 394
cse_CURRENCYSHORTFORMOFCASECOMPANYCURRENCY 394
cse_CURRENTCASESTATUSDESCRIPTION 394
cse_CURRENTCASESTATUSGUID 394
cse_CURRENTCASESTATUSINSERTTS 394
cse_DEADLINEDATE 394
cse_DESCRIPTION 394
cse_EXPECTEDVALUE 394
cse_GUID 394
cse_INCLUDEEXCEPTIONS 394
cse_INCLUDEOVERTIME 394
cse_INSERTTS 394
cse_INTERNALNAME 394
cse_INVOICETEMPLATEGUID 394
cse_ISCLOSED 394
cse_ISDAILYALLOWANCEBILLABLE 394
cse_ISINTERNAL 394
cse_ISMILEAGEBILLABLE 394
cse_ISOTHERTRAVELEXPENSESBILLABLE 394
cse_LATESTESTIMATIONDATE 394
cse_LEADRATING 394
cse_LEADSOURCEGUID 394
cse_NAME 394
cse_ORDERNUMBER 394
cse_OURREFERENCE 394
cse_OVERDUEINTEREST 394
cse_PAYMENTTERM 394
cse_PRICELISTGUID 394
cse_PRICINGRULE 394
cse_PRICINGTYPE 394
cse_PRIORITY 394
cse_PROBABILITY 394
cse_ROOTTASKGUID 394
cse_SALESCLOSEDATE 394
cse_SALESPERSONUSERCODE 394
cse_SALESPERSONUSERGUID 394
cse_SALESPROCESSGUID 394
cse_SALESSTATUSGUID 394
cse_SHAREDTOEXTRANET 394
cse_STARTDATE 394
cse_UPDATEDTS 394
cse_URLTOCASE 394
cse_USEDEFAULTPRODUCTSINWORKTIMEENTRY 394
cse_USEDEFAULTWORKTYPES 394
cse_YOURREFERENCE 394
CULTURE 105, 128, 149, 207, 212, 222, 227, 232, 238, 366, 371, 373, 375, 377, 379, 381, 382, 384, 397
Currencies 156
Currency by GUID 156
Currency by ISO Code 157
Currency Code 18, 19, 20, 22, 26, 27, 29, 30, 44, 59, 70, 114, 116, 117, 118, 158, 160, 161, 267, 268, 331, 340, 349
Currency GUID 18, 19, 20, 22, 26, 27, 29, 30, 44, 59, 61, 70, 114, 116, 117, 118, 120, 122, 125, 129, 132, 135, 149, 158, 160, 161, 207, 212, 222, 227, 232, 238, 267, 268, 322, 323, 325, 326, 328, 331, 340, 342, 349, 394, 397
Currency Name 149, 207, 212, 222, 227, 232, 238, 397
Currency Rate 149, 207, 212, 222, 227, 232, 238, 397
Currency Short Form 61, 120, 122, 125, 129, 132, 135, 149, 207, 212, 222, 227, 232, 238, 322, 323, 325, 326, 328, 342, 394, 397
Currency Short Form of Project Company Currency 61, 120, 122, 125, 129, 132, 135, 342, 394
CurrencyByGUID 156
CurrencyByIsoCode 157

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

CURRENCYCODE 18, 19, 20, 22, 26, 27, 29, 30, 44, 59, 70, 86, 114, 116, 117, 118, 158, 160, 161, 267, 268, 331, 340, 349
CURRENCYGUID 18, 19, 20, 22, 26, 27, 29, 30, 44, 59, 61, 70, 114, 116, 117, 118, 120, 122, 125, 129, 132, 135, 149, 156, 158, 160, 161, 207, 212, 222, 227, 232, 238, 267, 268, 322, 323, 325, 326, 328, 331, 340, 342, 349, 397
currencyIsoCode 157
CURRENCYNAME 149, 207, 212, 222, 227, 232, 238, 397
CURRENCYRATE 149, 207, 212, 222, 227, 232, 238, 397
CURRENCYSHORTFORM 61, 120, 122, 125, 129, 132, 135, 149, 207, 212, 222, 227, 232, 238, 322, 323, 325, 326, 328, 342, 397
CURRENCYSHORTFORMOFCASECOMPANYCURRENCY 61, 120, 122, 125, 129, 132, 135, 342
Current Project Status Description 61, 120, 122, 125, 129, 132, 135, 342, 394
Current Project Status GUID 61, 120, 122, 125, 129, 132, 135, 342, 394
Current Project Status Inserted 61, 120, 122, 125, 129, 132, 135, 342, 394
CURRENTCASESTATUSDESCRIPTION 61, 120, 122, 125, 129, 132, 135, 342
CURRENTCASESTATUSGUID 61, 120, 122, 125, 129, 132, 135, 342
CURRENTCASESTATUSINSERTTS 61, 120, 122, 125, 129, 132, 135, 342
Customer by GUID 158
customer_AccountGroupGUIDs 70, 349
customer_AccountOwnerUserGUID 70, 349
customer_AccountRating 70, 349
customer_AnnualRevenue 70, 349
customer_AnnualRevenue2 70, 349
customer_CompanyGUID 70, 349
customer_CurrencyCode 70, 349
customer_CurrencyGUID 70, 349
customer_eInvoiceAddress 70, 349
customer_eInvoiceOperator 70, 349
customer_Email 70, 349
customer_Employees 70, 349
customer_GUID 70, 349
customer_HeadOfficeAddressGUID 70, 349
customer_IndustryGUID 70, 349
customer_InsertTS 70, 349
customer_InvoicingVat 70, 349
customer_IsActive 70, 349
customer_IsInternal 70, 349
customer_IsPrivatePerson 70, 349
customer_LanguageCode 70, 349
customer_LanguageGUID 70, 349
customer_Name 70, 349
customer_Notes 70, 349
customer_Number 70, 349
customer_OverdueInterest 70, 349
customer_PaymentTerm 70, 349
customer_PricelistGUID 70, 349
customer_ReverseCharge 70, 349
customer_ReverseChargeDescription 70, 349
customer_TimezoneGUID 70, 349
customer_UpdatedTS 70, 349
customer_UrlToAccount 70, 349
customer_VatNumber 70, 349
customer_Website 70, 349
CustomerByGUID 158
Customers 160
Customers Changed since 161
CustomersChangedSince 161

- D -

Daily Hours 73, 177, 178, 352
DAILYHOURS 73, 177, 178, 352
Database 163
Database Driver 1
DATABASERESULT 163
DATE 149, 207, 212, 222, 227, 232, 238, 322, 323, 325, 326, 328, 387, 397
Date of Birth 67, 140, 141, 143, 144, 348
DATE_ENDED 392
DATE_STARTED 392
DATEOFBIRTH 67, 140, 141, 143, 144, 348
Ddescription 61, 67, 74, 80, 86, 91, 94, 96, 104, 120, 122, 125, 129, 132, 135, 140, 141, 143, 144, 149, 183, 185, 189, 192, 194, 196, 199, 201, 207, 212, 217, 222, 227, 232, 238, 243, 246, 249, 251, 254, 261, 263, 265, 266, 269, 270, 271, 276, 277, 280, 282, 284, 287, 310, 311, 312, 314, 315, 316, 317, 320, 321, 342, 348, 353, 358, 360, 364, 394, 397
DEADLINE 94, 261, 263, 265, 266, 358
Deadline Date 61, 120, 122, 125, 129, 132, 135, 342, 394
DEADLINEDATE 61, 120, 122, 125, 129, 132, 135, 342
Default Activity Type GUID 105, 128, 366, 371, 373, 375, 377, 379, 381, 382, 384
Default Language Code 146, 147, 148
Default Language GUID 146, 147, 148
Default Probability 102, 300, 301
DEFAULTACTIVITYTYPEGUID 105, 128, 366, 371, 373, 375, 377, 379, 381, 382, 384
DEFAULTLANGUAGECODE 146, 147, 148
DEFAULTLANGUAGEGUID 146, 147, 148

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

DEFAULTPROBABILITY 102, 300, 301
Delete Access Rights Profile by GUID 163
Delete Activity 164
Delete Activity Type 165
Delete Contact 168
Delete Employment 168
Delete Hour Entry 169
Delete Item 170
Delete Phase 171
Delete Product 172
Delete Product Category 172
Delete Project 166
Delete Project Status 167
Delete Resource 173
Delete Resource Allocation 174
Delete Tag 175
Delete Travel Expense 176
DeleteAccessRightsProfileByGUID 163
DeleteActivity 164
DELETEACTIVITYRESULT 164
DeleteActivityType 165
DELETEACTIVITYTYPERESULT 165
DeleteCase 166
DELETECASERESULT 166
DeleteCaseStatus 167
DELETECASESTATUSRESULT 167
DeleteContact 168
DELETECONTACTRESULT 168
DeleteEmployment 168
DELETEEMPLOYMENTRESULT 168
DeleteHourEntry 169
DELETEHOURENTRYRESULT 169
DeleteItem 170
DELETEITEMRESULT 170
DeletePhase 171
DELETEPHASERESULT 171
DeleteProduct 172
DeleteProductCategory 172
DELETEPRODUCTCATEGORYRESULT 172
DELETEPRODUCTRESULT 172
DeleteResource 173
DeleteResourceAllocation 174
DELETERESOURCEALLOCATIONRESULT 174
DELETERESOURCERESULT 173
DeleteTag 175
DELETETAGRESULT 175
DeleteTravelExpense 176
DELETETRAVELEXPENSERESULT 176
DESCRIPTION 61, 67, 74, 80, 86, 91, 94, 96, 104, 120, 122, 125, 129, 132, 135, 140, 141, 143, 144, 149, 183, 185, 189, 192, 194, 196, 199, 201, 207, 212, 217, 222, 227, 232, 238, 243, 246, 249, 251, 254, 261, 263, 265, 266, 269, 270, 271, 276, 277, 280, 282, 284, 287, 310, 311, 312, 314, 315, 316, 317, 320, 321, 342, 348, 353, 358, 360, 364, 394, 397
Difference to GMT 313, 314
DIFFERENCETOGMT 313, 314
Disable User 176
DisableUser 176
DISABLEUSERRESULT 176
Display Period Start 74, 86, 91, 243, 246, 249, 251, 254, 284, 287, 317
DISPLAYPERIODSTART 74, 86, 91, 243, 246, 249, 251, 254, 284, 287, 317
download-error-400-bad-request-max-tries 2
download-error-400-bad-request-sleep-initial-ms 2
download-error-400-bad-request-sleep-max-ms 2
download-error-400-bad-request-sleep-multiplicator 2
download-error-403-forbidden-max-tries 2
download-error-403-forbidden-sleep-initial-ms 2
download-error-403-forbidden-sleep-max-ms 2
download-error-403-forbidden-sleep-multiplicator 2
download-error-408-request-timeout-max-tries 2
download-error-408-request-timeout-sleep-initial-ms 2
download-error-408-request-timeout-sleep-max-ms 2
download-error-408-request-timeout-sleep-multiplicator 2
download-error-422-bad-request-max-tries 2
download-error-422-bad-request-sleep-initial-ms 2
download-error-422-bad-request-sleep-max-ms 2
download-error-422-bad-request-sleep-multiplicator 2
download-error-429-too-many-requests-max-tries 2
download-error-429-too-many-requests-sleep-initial-ms 2
download-error-429-too-many-requests-sleep-max-ms 2
download-error-429-too-many-requests-sleep-multiplicator 2
download-error-500-internal-server-error-max-tries 2
download-error-500-internal-server-error-sleep-initial-ms 2
download-error-500-internal-server-error-sleep-max-ms 2
download-error-500-internal-server-error-sleep-multiplicator 2
download-error-502-server-unavailable-max-tries 2
download-error-502-server-unavailable-sleep-initial-ms 2

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

download-error-502-server-unavailable-sleep-max-ms 2
download-error-502-server-unavailable-sleep-multiplicator 2
download-error-503-server-unavailable-max-tries 2
download-error-503-server-unavailable-sleep-initial-ms 2
download-error-503-server-unavailable-sleep-max-ms 2
download-error-503-server-unavailable-sleep-multiplicator 2
download-error-504-gateway-timeout-max-tries 2
download-error-504-gateway-timeout-sleep-initial-ms 2
download-error-504-gateway-timeout-sleep-max-ms 2
download-error-504-gateway-timeout-sleep-multiplicator 2
download-error-590-network-connect-timeout-max-tries 2
download-error-590-network-connect-timeout-sleep-initial-ms 2
download-error-590-network-connect-timeout-sleep-max-ms 2
download-error-590-network-connect-timeout-sleep-multiplicator 2
download-error-599-network-connect-timeout-max-tries 2
download-error-599-network-connect-timeout-sleep-initial-ms 2
download-error-599-network-connect-timeout-sleep-max-ms 2
download-error-599-network-connect-timeout-sleep-multiplicator 2
download-error-argument-exception-max-tries 2
download-error-argument-exception-sleep-initial-ms 2
download-error-argument-exception-sleep-max-ms 2
download-error-argument-exception-sleep-multiplicator 2
download-error-internet-down-max-tries 2
download-error-internet-down-sleep-initial-ms 2
download-error-internet-down-sleep-max-ms 2
download-error-internet-down-sleep-multiplicator 2
download-error-io-exception-max-tries 2
download-error-io-exception-sleep-initial-ms 2
download-error-io-exception-sleep-max-ms 2
download-error-io-exception-sleep-multiplicator 2
download-error-json-exception-max-tries 2
download-error-json-exception-sleep-initial-ms 2
download-error-json-exception-sleep-max-ms 2
download-error-json-exception-sleep-multiplicator 2
download-error-name-resolution-failure-max-tries 2
download-error-name-resolution-failure-sleep-initial-ms 2
download-error-name-resolution-failure-sleep-max-ms 2
download-error-name-resolution-failure-sleep-multiplicator 2
download-error-other-exception-max-tries 2
download-error-other-exception-sleep-initial-ms 2
download-error-other-exception-sleep-max-ms 2
download-error-other-exception-sleep-multiplicator 2
download-error-socket-exception-max-tries 2
download-error-socket-exception-sleep-initial-ms 2
download-error-socket-exception-sleep-max-ms 2
download-error-socket-exception-sleep-multiplicator 2
download-error-web-exception-max-tries 2
download-error-web-exception-sleep-initial-ms 2
download-error-web-exception-sleep-max-ms 2
download-error-web-exception-sleep-multiplicator 2
download-error-web-not-found-max-tries 2
download-error-web-not-found-sleep-initial-ms 2
download-error-web-not-found-sleep-max-ms 2
download-error-web-not-found-sleep-multiplicator 2
download-error-web-not-implemented-max-tries 2
download-error-web-not-implemented-sleep-initial-ms 2
download-error-web-not-implemented-sleep-max-ms 2
download-error-web-not-implemented-sleep-multiplicator 2
download-error-web-timeout-max-tries 2
download-error-web-timeout-sleep-initial-ms 2
download-error-web-timeout-sleep-max-ms 2
download-error-web-timeout-sleep-multiplicator 2
download-error-web-unauthorized-max-tries 2
download-error-web-unauthorized-sleep-initial-ms 2
download-error-web-unauthorized-sleep-max-ms 2
download-error-web-unauthorized-sleep-multiplicator 2
download-error-web-unknown-max-tries 2
download-error-web-unknown-sleep-initial-ms 2
download-error-web-unknown-sleep-max-ms 2
download-error-web-unknown-sleep-multiplicator 2
DRY_RUN 392
Due Date 149, 207, 212, 222, 227, 232, 238, 397
DUEDATE 149, 207, 212, 222, 227, 232, 238, 397
DUMMY 42, 49, 85, 163, 289, 304, 305, 306, 307
Duration (ms) 392
DURATION_MS 392

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

- E -

e-invoice Address 18, 19, 20, 22, 26, 27, 29, 30, 44, 70, 158, 160, 161, 331, 349
e-invoice Operator 18, 19, 20, 22, 26, 27, 29, 30, 44, 70, 158, 160, 161, 331, 349
EINVOICEADDRESS 18, 19, 20, 22, 26, 27, 29, 30, 44, 70, 158, 160, 161, 331, 349
EINVOICEOPERATOR 18, 19, 20, 22, 26, 27, 29, 30, 44, 70, 158, 160, 161, 331, 349
EMAIL 70, 105, 128, 139, 158, 160, 161, 349, 366, 371, 373, 375, 377, 379, 381, 382, 384
EMPLOYEES 70, 139, 158, 160, 161, 349
Employment by GUID 177
EmploymentByGUID 177
employmentGUID 168, 177
employmentInfo_CompanyBusinessUnitCode 73, 352
employmentInfo_CompanyBusinessUnitGUID 73, 352
employmentInfo_CompanyBusinessUnitName 73, 352
employmentInfo_DailyHours 73, 352
employmentInfo_EndDate 73, 352
employmentInfo_GUID 73, 352
employmentInfo_HourCost 73, 352
employmentInfo_IsOvertimeAllowed 73, 352
employmentInfo_StartDate 73, 352
employmentInfo_Title 73, 352
Employments by User GUID 178
EmploymentsByUserGUID 178
Enable User 179
EnableUser 179
ENABLEUSERRESULT 179
End Date 73, 99, 101, 177, 178, 293, 294, 295, 296, 297, 298, 352, 363, 392
End Date Time 32, 34, 36, 50, 334
End Time 80, 189, 192, 194, 196, 199, 201, 217, 277, 280, 282, 310, 311, 312, 353, 394
enddate 32, 73, 99, 101, 177, 178, 189, 192, 194, 196, 222, 227, 277, 293, 294, 295, 296, 297, 298, 317, 325, 352, 363, 387
ENDDATETIME 32, 34, 36, 50, 334
endsAfter 36
endsBefore 36
ENDTIME 80, 189, 192, 194, 196, 199, 201, 217, 277, 280, 282, 310, 311, 312, 353, 394
Entity Schema Version 180
EntitySchemaVersion 180
ENTRIES 387
Entry Date 149, 207, 212, 222, 227, 232, 238, 397
ENTRYDATE 149, 207, 212, 222, 227, 232, 238, 397
Error Message Code 392
Error Message Text 392
ERROR_MESSAGE_CODE 392
ERROR_MESSAGE_TEXT 392
Event Date 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 310, 311, 312, 317, 353, 394
EVENTDATE 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 310, 311, 312, 317, 353, 394
eventTimeUtc 85
Expected Hours 387
Expected Value 61, 120, 122, 125, 129, 132, 135, 342, 394
EXPECTEDHOURS 387
EXPECTEDVALUE 61, 120, 122, 125, 129, 132, 135, 342
Expense Class 104, 314, 315, 316, 320, 364
Expense Forecast 57, 113, 339
EXPENSECLASS 104, 314, 315, 316, 320, 364
EXPENSEFORECAST 57, 113, 339
Extranet Contact Project Rights by Contact GUID 182
Extranet Project by Project GUID 180
ExtranetCaseByCaseGUID 180
ExtranetContactCaseRightsByContactGUID 182

- F -

Fail on Error 392
FAIL_ON_ERROR 392
FAX 56, 110, 111, 113, 187, 188, 337
File by GUID 183
File Contents by GUID 184
FileByGUID 183
FileBytesByGUID 184
fileGUID 79
Files 185
Financial Invoice 185, 186
Financial Invoices by Status 185
Financial Invoices by Status and Date 186
FINVOICE 185, 186
FinvoicesByStatus 185
FinvoicesByStatusAndDate 186
finvoiceVersion 186
First Account Billing Addresses 187
First Name 67, 105, 128, 140, 141, 143, 144, 348, 366, 371, 373, 375, 377, 379, 381, 382, 384
FirstAccountsBillingAddress 187

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

FIRSTNAME 67, 105, 128, 140, 141, 143, 144, 348, 366, 371, 373, 375, 377, 379, 381, 382, 384
firstRow 36
Flat Rate Additional Hours Unit Price 94, 261, 263, 265, 266, 358
Flat Rate Are additional Hours Billable 94, 261, 263, 265, 266, 358
Flat Rate Hours 94, 261, 263, 265, 266, 358
Flat Rate Price 94, 261, 263, 265, 266, 358
FLATRATE_ADDITIONALHOURSUNITPRICE 94, 261, 263, 265, 266, 358
FLATRATE_AREADDITIONALHOURSBILLABLE 94, 261, 263, 265, 266, 358
FLATRATE_HOURS 94, 261, 263, 265, 266, 358
FLATRATE_PRICE 94, 261, 263, 265, 266, 358
force-case-sensitive-identifiers 2
forced-casing-identifiers 2
Formatting Culture 59, 114, 116, 117, 118, 340
FORMATTINGCULTURE 59, 114, 116, 117, 118, 340

- G -

getAttachmentOnly 220, 221
GETENTITYSCHEMAVERSIONRESULT 180
GETFILEBYTESBYGUIDRESULT 184
GETINVOICEATTACHMENTFILERESULT 205
GETINVOICEPDFBYINVOICEGUIDRESULT 221
GETINVOICEPDFRESULT 220
GETTRAVELREIMBURSEMENTPDFRESULT 324
GETVERSIONRESULT 386
GUID 16, 17, 18, 19, 20, 22, 23, 24, 25, 26, 27, 29, 30, 32, 34, 36, 39, 40, 43, 44, 48, 50, 54, 56, 57, 59, 61, 65, 66, 67, 69, 70, 73, 74, 79, 80, 84, 86, 91, 94, 96, 97, 98, 99, 101, 102, 103, 104, 105, 108, 109, 110, 111, 113, 114, 116, 117, 118, 120, 122, 125, 128, 129, 132, 135, 137, 138, 139, 140, 141, 143, 144, 145, 146, 147, 148, 149, 156, 157, 158, 160, 161, 163, 177, 178, 180, 183, 184, 185, 186, 187, 188, 189, 192, 194, 196, 199, 201, 203, 204, 206, 207, 212, 217, 222, 227, 232, 238, 243, 246, 249, 251, 254, 257, 258, 259, 260, 261, 263, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 280, 282, 284, 287, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 320, 321, 322, 323, 324, 325, 326, 328, 329, 331, 334, 336, 337, 339, 340, 342, 347, 348, 349, 352, 353, 358, 360, 361, 362, 363, 364, 366, 369, 370, 371, 373, 375, 376, 377, 379, 381, 382, 384, 386, 388, 389, 390, 391, 394, 397

- H -

Has Invoice Number 243
HASINVOICENUMBER 243
Head Office Address by Account GUID 188
Head Office Address GUID 70, 139, 158, 160, 161, 349
HeadOfficeAddressByAccountGUID 188
HEADOFFICEADDRESSGUID 70, 139, 158, 160, 161, 349
Hierarchy GUID 139
HIERARCHYGUID 139
Hour Cost 73, 177, 178, 352
Hour Entries by Date 192
Hour Entries by Date and User 194
Hour Entries by Project GUID 189
Hour Entries Changed since 199
Hour Entries Last 2 Years 394
Hour Entry by Date and User GUID 196
Hour Entry by GUID 201
Hour Entry Phases by User GUID 203
HOURCOST 73, 177, 178, 352
HourEntries2Years 394
HourEntriesByCaseGUID 189
HourEntriesByDate 192
HourEntriesByDateAndUser 194
HourEntriesByDateAndUserGUID 196
HourEntriesChangedSince 199
HourEntryByGUID 201
hourEntryGUID 169, 201, 282
hourEntryInfo_AccountGUID 80, 353
hourEntryInfo_CaseBusinessUnitCode 80, 353
hourEntryInfo_CaseCostCenterIdentifier 80, 353
hourEntryInfo_CaseGUID 80, 353
hourEntryInfo_CaseName 80, 353
hourEntryInfo_CaseNumber 80, 353
hourEntryInfo_CostCurrencyShortform 80, 353
hourEntryInfo_CostCurrencyShortformOfInvoicingCurrency 80, 353
hourEntryInfo_CostCurrencyShortformOfOrganizationCurrency 80, 353
hourEntryInfo_CostCurrencyShortformOfUserCompanyCurrency 80, 353
hourEntryInfo_Description 80, 353
hourEntryInfo_EndTime 80, 353
hourEntryInfo_EventDate 80, 353
hourEntryInfo_GUID 80, 353
hourEntryInfo_InvoiceDescription 80, 353
hourEntryInfo_InvoiceGUID 80, 353
hourEntryInfo_InvoiceQuantity 80, 353
hourEntryInfo_InvoiceRowGUID 80, 353
hourEntryInfo_IsApproved 80, 353
hourEntryInfo_IsBillable 80, 353
hourEntryInfo_IsProductive 80, 353
hourEntryInfo_IsReadOnly 80, 353

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

hourEntryInfo_OvertimeCode 80, 353
hourEntryInfo_OverTimeGUID 80, 353
hourEntryInfo_PhaseGUID 80, 353
hourEntryInfo_PriceCurrencyShortformOfCaseCompanyCurrency 80, 353
hourEntryInfo_PriceCurrencyShortformOfInvoicingCurrency 80, 353
hourEntryInfo_PriceCurrencyShortformOfOrganizationCurrency 80, 353
hourEntryInfo_PriceCurrencyShortformOfUserCompanyCurrency 80, 353
hourEntryInfo_Quantity 80, 353
hourEntryInfo_StartTime 80, 353
hourEntryInfo_TaskGUID 80, 353
hourEntryInfo_UnitCost 80, 353
hourEntryInfo_UnitCostInInvoicingCurrency 80, 353
hourEntryInfo_UnitCostInOrganizationCurrency 80, 353
hourEntryInfo_UnitCostInUserCompanyCurrency 80, 353
hourEntryInfo_UnitPrice 80, 353
hourEntryInfo_UnitPriceInBaseCurrency 80, 353
hourEntryInfo_UnitPriceInCaseCompanyCurrency 80, 353
hourEntryInfo_UnitPriceInUserCompanyCurrency 80, 353
hourEntryInfo_UseInvoiceQuantity 80, 353
hourEntryInfo_UserBusinessUnitCode 80, 353
hourEntryInfo_UserCode 80, 353
hourEntryInfo_UserFirstName 80, 353
hourEntryInfo_UserGUID 80, 353
hourEntryInfo_UserLastName 80, 353
hourEntryInfo_WorkTypeCode 80, 353
hourEntryInfo_WorkTypeGUID 80, 353
HourEntryPhasesByUserGUID 203
hourGUID 389
HTTP Disk Cache Maximum Age (sec) 392
HTTP Memory Cache Maximum Age (sec) 392
HTTP Method 392
HTTP Status Code 392
HTTP_DISK_CACHE_MAX_AGE_SEC 392
HTTP_DISK_CACHE_SAVE 392
HTTP_DISK_CACHE_USE 392
HTTP_MEMORY_CACHE_MAX_AGE_SEC 392
HTTP_MEMORY_CACHE_SAVE 392
HTTP_MEMORY_CACHE_USE 392
HTTP_METHOD 392
HTTP_STATUS_CODE 392
http-disk-cache-compression-level 2
http-disk-cache-directory 2
http-disk-cache-ignore-write-errors 2
http-disk-cache-max-age-sec 2
http-get-timeout-max-ms 2
http-get-timeout-ms 2
http-memory-cache-compression-level 2
http-memory-cache-max-age-sec 2
http-post-timeout-max-ms 2
http-post-timeout-ms 2
http-public-disk-cache-directory 2

I

ICON 39, 40, 54, 65, 102, 137, 138, 300, 301, 336, 347
IDENTIFIER 145, 146
IETF Language Tag 257, 258
IETFLANGUAGETAG 257, 258
ignoreDuplicates 85
ignore-http-400-errors 2
ignore-http-401-errors 2
ignore-http-402-errors 2
ignore-http-403-errors 2
ignore-http-404-errors 2
ignore-http-422-errors 2
ignore-http-429-errors 2
ignore-http-500-errors 2
ignore-http-502-errors 2
ignore-http-503-errors 2
Include Exceptions 61, 120, 122, 125, 129, 132, 135, 342, 394
Include in Flextime 260, 261
Include Overtime 61, 120, 122, 125, 129, 132, 135, 342, 394
includeDefault 275
INCLUDEEXCEPTIONS 61, 120, 122, 125, 129, 132, 135, 342
includeInactive 377
INCLUDEINFLEXTIME 260, 261
includeInfoForInactiveUser 387
INCLUDEOVERTIME 61, 120, 122, 125, 129, 132, 135, 342
includeUnPaidAbsenses 387
inCreditNote 277, 280, 287
Industries 204
Industry GUID 70, 139, 158, 160, 161, 349
INDUSTRYGUID 70, 139, 158, 160, 161, 349
industryInfo_Code 84
industryInfo_GUID 84
industryInfo_IsActive 84
industryInfo_Name 84
Inhouse Revenue Forecast 57, 113, 339
INHOUSEREVENUEFORECAST 57, 113, 339

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Inserted 18, 19, 20, 22, 26, 27, 29, 30, 44, 59, 61, 70, 114, 116, 117, 118, 120, 122, 125, 129, 132, 135, 158, 160, 161, 331, 340, 342, 349, 394
insertEndDate 27, 251
insertPosition 65
insertStartDate 27, 251
INSERTTS 18, 19, 20, 22, 26, 27, 29, 30, 44, 59, 61, 70, 114, 116, 117, 118, 120, 122, 125, 129, 132, 135, 158, 160, 161, 331, 340, 342, 349
Internal Name 61, 120, 122, 125, 129, 132, 135, 342, 394
INTERNALNAME 61, 120, 122, 125, 129, 132, 135, 342
invalid-json-on-get-max-tries 2
invalid-json-on-get-sleep-initial-ms 2
invalid-json-on-get-sleep-max-ms 2
invalid-json-on-get-sleep-multiplicator 2
invalid-json-on-post-max-tries 2
invalid-json-on-post-sleep-initial-ms 2
invalid-json-on-post-sleep-max-ms 2
invalid-json-on-post-sleep-multiplicator 2
invantive-sql-compress-sparse-arrays 2
invantive-sql-correct-invalid-date 2
invantive-sql-execution-profile-disk-path 2
invantive-sql-execution-profile-to-disk 2
invantive-sql-forward-filters-to-data-containers 2
invantive-sql-share-byte-arrays 2
invantive-sql-share-strings 2
invantive-sql-shuffle-fetch-results-data-containers 2
invantive-use-cache 2
Invoice Attachment File 205
Invoice Attachment File GUIDs 206
Invoice by GUID 207
Invoice by Number 212
Invoice Description 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 353, 394
Invoice GUID 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 353, 394
Invoice Line GUID 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 353, 394
Invoice Numbering Business Unit GUID 149, 207, 212, 222, 227, 232, 238, 397
Invoice Origin Details by Invoice GUID 219
Invoice PDF 220
Invoice PDF by Invoice GUID 221
Invoice Quantity 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 353, 394
Invoice Status 149, 207, 212, 222, 227, 232, 238, 397
Invoice Statuses 243
Invoice Template GUID 61, 120, 122, 125, 129, 132, 135, 342, 394
INVOICEALLOWEDCASEGUIDS 149, 207, 212, 222, 227, 232, 238, 397
InvoiceAttachmentFile 205
InvoiceAttachmentFileGUIDs 206
InvoiceByGUID 207
InvoiceByNumber 212
INVOICECONFIG_BILLEXPENSE 149, 207, 212, 222, 227, 232, 238, 397
INVOICECONFIG_BILLHOUR 149, 207, 212, 222, 227, 232, 238, 397
INVOICECONFIG_BILLPRODUCT 149, 207, 212, 222, 227, 232, 238, 397
INVOICECONFIG_EXPENSEITEMDESCRIPTIONFORMAT 149, 207, 212, 222, 227, 232, 238, 397
INVOICECONFIG_HOURITEMDESCRIPTIONFORMAT 149, 207, 212, 222, 227, 232, 238, 397
INVOICECONFIG_HOURPRIMARYGROUPBY 149, 207, 212, 222, 227, 232, 238, 397
INVOICECONFIG_HOURSECONDARYGROUPBY 149, 207, 212, 222, 227, 232, 238, 397
INVOICECONFIG_ISTAXFREE 149, 207, 212, 222, 227, 232, 238, 397
INVOICECONFIG_PRODUCTITEMDESCRIPTIONFORMAT 149, 207, 212, 222, 227, 232, 238, 397
INVOICECONFIG_PRODUCTPRIMARYGROUPBY 149, 207, 212, 222, 227, 232, 238, 397
INVOICECONFIG_PRODUCTSECONDARYGROUPBY 149, 207, 212, 222, 227, 232, 238, 397
INVOICECONFIG_SHOWATTACHPRICEEXCLUDINGVAT 149, 207, 212, 222, 227, 232, 238, 397
INVOICECONFIG_SHOWATTACHUNIT 149, 207, 212, 222, 227, 232, 238, 397
INVOICECONFIG_SHOWATTACHUNITPRICE 149, 207, 212, 222, 227, 232, 238, 397
INVOICECONFIG_SHOWATTACHVAT 149, 207, 212, 222, 227, 232, 238, 397
INVOICECONFIG_SHOWLOGOONINVOICE 149, 207, 212, 222, 227, 232, 238, 397
INVOICECONFIG_SHOWPRICEEXCLUDINGVAT 149, 207, 212, 222, 227, 232, 238, 397
INVOICECONFIG_SHOWQUANTITY 149, 207, 212, 222, 227, 232, 238, 397
INVOICECONFIG_SHOWUNIT 149, 207, 212, 222, 227, 232, 238, 397
INVOICECONFIG_SHOWUNITPRICE 149, 207, 212, 222, 227, 232, 238, 397
INVOICECONFIG_SHOWVAT 149, 207, 212, 222, 227, 232, 238, 397
INVOICECONFIG_TAXFREEDESCRIPTION 149, 207, 212, 222, 227, 232, 238, 397

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

INVOICECONFIG_TRAVELPRIMARYGROUPBY 149, 207, 212, 222, 227, 232, 238, 397
INVOICECONFIG_TRAVELSECONDARYGROUPBY 149, 207, 212, 222, 227, 232, 238, 397
Invoiced Hour Entries by Project GUID 217
invoiceDateEnd 186
invoiceDateStart 186
INVOICEDESCRIPTION 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 353, 394
INVOICEDETAILS_INVOICEDATE 149, 207, 212, 222, 227, 232, 238, 397
INVOICEDETAILS_INVOICEDUEDATE 149, 207, 212, 222, 227, 232, 238, 397
INVOICEDETAILS_INVOICENUMBER 149, 207, 212, 222, 227, 232, 238, 397
INVOICEDETAILS_INVOICETOTALVATAMOUNT 149, 207, 212, 222, 227, 232, 238, 397
INVOICEDETAILS_INVOICETOTALVATEXCLUDEDAMOUNT 149, 207, 212, 222, 227, 232, 238, 397
INVOICEDETAILS_INVOICETOTALVATINCLUDEDAMOUNT 149, 207, 212, 222, 227, 232, 238, 397
INVOICEDETAILS_INVOICETYPETEXT 149, 207, 212, 222, 227, 232, 238, 397
INVOICEDETAILS_ORDERIDENTIFIER 149, 207, 212, 222, 227, 232, 238, 397
INVOICEDETAILS_ORIGINCODE 149, 207, 212, 222, 227, 232, 238, 397
INVOICEDETAILS_PAYMENTOVERDUEFINEFREETEXT 149, 207, 212, 222, 227, 232, 238, 397
INVOICEDETAILS_PAYMENTOVERDUEFINEPERCENT 149, 207, 212, 222, 227, 232, 238, 397
INVOICEDETAILS_PAYMENTTERMSFREETEXT 149, 207, 212, 222, 227, 232, 238, 397
INVOICEDETAILS_SELLERREFERENCEIDENTIFIER 149, 207, 212, 222, 227, 232, 238, 397
INVOICEDETAILS_VATBASEAMOUNT 149, 207, 212, 222, 227, 232, 238, 397
INVOICEDETAILS_VATRATEAMOUNT 149, 207, 212, 222, 227, 232, 238, 397
INVOICEDETAILS_VATRATEPERCENT 149, 207, 212, 222, 227, 232, 238, 397
InvoicedHourEntriesByCaseGUID 217
INVOICEGUID 74, 80, 85, 86, 91, 189, 192, 194, 196, 199, 201, 206, 207, 217, 219, 221, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 304, 305, 306, 317, 353, 394
invoiceInformation_Date 149
invoiceInformation_DueDate 149
invoiceInformation_InvoiceStatus 149
invoiceInformation_ItemGUIDs 149
invoiceInformation_LanguageGUID 149
invoiceInformation_Notes 149
invoiceInformation_OurReference 149
invoiceInformation_OverDueInterest 149
invoiceInformation_PaymentTerm 149
invoiceInformation_ReceiverAddressline 149
invoiceInformation_ReceiverCity 149
invoiceInformation_ReceiverContactEmail 149
invoiceInformation_ReceiverContactName 149
invoiceInformation_ReceiverCountry 149
invoiceInformation_ReceiverCustomerName 149
invoiceInformation_ReceiverPostalCode 149
invoiceInformation_ReceiverState 149
invoiceInformation_ReceiverVatNumber 149
invoiceInformation_SenderAddressline 149
invoiceInformation_SenderCity 149
invoiceInformation_SenderContactName 149
invoiceInformation_SenderCountry 149
invoiceInformation_SenderName 149
invoiceInformation_SenderPostalCode 149
invoiceInformation_SenderState 149
invoiceInformation_SenderVatNumber 149
invoiceInformation_YourReference 149
invoiceNumber 212, 220, 304
INVOICENUMBERINGBUSINESSUNITGUID 149, 207, 212, 220, 222, 227, 232, 238, 397
InvoiceOriginDetailsByInvoiceGUID 219
InvoicePdf 220
InvoicePdfByInvoiceGUID 221
INVOICEQUANTITY 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 353, 394
INVOICEROWGUID 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 353, 394
INVOICEROWORIGINS 219
INVOICEROWS 149, 207, 212, 222, 227, 232, 238, 397
Invoices 397
Invoices by Date 222
Invoices by Date and Status 227
Invoices by Status 232
Invoices by Status and Business Unit 238
InvoicesByDate 222
InvoicesByDateAndStatus 227
InvoicesByStatus 232
InvoicesByStatusAndBusinessUnit 238
INVOICESTATUS 149, 186, 207, 212, 222, 227, 232, 238, 397
InvoiceStatuses 243
INVOICETEMPLATEGUID 61, 120, 122, 125, 129, 132, 135, 342
Invoicing VAT 18, 19, 20, 22, 26, 27, 29, 30, 44, 70, 158, 160, 161, 331, 349

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

INVOICINGVAT 18, 19, 20, 22, 26, 27, 29, 30, 44, 70, 158, 160, 161, 331, 349
Is Active 16, 17, 18, 19, 20, 22, 23, 24, 26, 27, 29, 30, 32, 39, 40, 43, 44, 48, 54, 59, 65, 66, 67, 69, 70, 79, 84, 96, 97, 98, 102, 104, 105, 108, 109, 114, 116, 117, 118, 128, 137, 138, 139, 140, 141, 143, 144, 145, 146, 156, 157, 158, 160, 161, 204, 243, 258, 259, 260, 261, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 299, 300, 301, 308, 309, 314, 315, 316, 320, 329, 331, 336, 340, 347, 348, 349, 360, 361, 362, 364, 366, 370, 371, 373, 375, 377, 379, 381, 382, 384, 386, 388, 389, 390, 391
Is All Day 32, 34, 36, 50, 334
Is also Item Status 102, 300, 301
Is Approved 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 329, 353, 394
Is Billable 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 353, 394
Is Billing Address 56, 110, 111, 113, 187, 188, 337
Is Closed 32, 34, 36, 50, 61, 120, 122, 125, 129, 132, 135, 334, 342, 394
Is Completed 94, 261, 263, 265, 266, 358
Is Credit Note 149, 207, 212, 222, 227, 232, 238, 397
Is Daily Allowance Billable 61, 120, 122, 125, 129, 132, 135, 342, 394
Is Default 16, 17, 39, 40, 43, 54, 97, 109, 145, 146, 243, 267, 268, 272, 273, 274, 329, 336, 361, 370, 388, 389, 390, 391
Is Duration 32, 34, 36, 50, 334
Is EU 146, 147, 148
Is in Progress 102, 300, 301
Is Internal 18, 19, 20, 22, 26, 27, 29, 30, 44, 61, 70, 120, 122, 125, 129, 132, 135, 158, 160, 161, 267, 268, 331, 342, 349, 394
Is Locked 94, 203, 261, 263, 265, 266, 329, 358
Is Lost 102, 300, 301
Is Mileage Billable 61, 120, 122, 125, 129, 132, 135, 342, 394
Is Offer 102, 300, 301
Is Out-sourcing Cost 74, 86, 91, 243, 246, 249, 251, 254, 284, 287, 317
Is Overtime Allowed 73, 177, 178, 352
Is Paid 243
Is Paid Leave 39, 40, 54, 336
Is Postal Address 56, 110, 111, 113, 187, 188, 337
Is Private 32, 34, 36, 50, 334
Is Productive 80, 109, 189, 192, 194, 196, 199, 201, 217, 277, 280, 282, 353, 370, 388, 389, 390, 391, 394
Is Read-only 32, 34, 36, 50, 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 334, 353, 394
Is Realized 74, 86, 91, 243, 246, 249, 251, 254, 284, 287, 317
Is Reimbursed 149, 207, 212, 222, 227, 232, 238, 397
Is Sent 243
Is Tax Free 149, 207, 212, 222, 227, 232, 238, 397
Is Travel Type Default 275
Is Unassigned 32, 34, 36, 50, 334
Is Visit Address 56, 110, 111, 113, 187, 188, 337
Is Volume Pricing 267, 268
Is Waiting Payment 243
Is Won 102, 300, 301
Is Work Time 260, 261
ISACTIVE 16, 17, 18, 19, 20, 22, 23, 24, 26, 27, 29, 30, 32, 39, 40, 43, 44, 48, 54, 59, 65, 66, 67, 69, 70, 79, 84, 96, 97, 98, 102, 104, 105, 108, 109, 114, 116, 117, 118, 128, 137, 138, 139, 140, 141, 143, 144, 145, 146, 156, 157, 158, 160, 161, 204, 243, 258, 259, 260, 261, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 299, 300, 301, 308, 309, 314, 315, 316, 320, 329, 331, 336, 340, 347, 348, 349, 360, 361, 362, 364, 366, 370, 371, 373, 375, 377, 379, 381, 382, 384, 386, 388, 389, 390, 391
ISALLDAY 32, 34, 36, 50, 334
ISALSOITEMSTATUS 102, 300, 301
ISAPPROVED 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 329, 353, 394
ISBILLABLE 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 353, 394
ISBILLINGADDRESS 56, 110, 111, 113, 187, 188, 337
ISCLOSED 32, 34, 36, 50, 61, 120, 122, 125, 129, 132, 135, 334, 342
ISCOMPLETED 94, 261, 263, 265, 266, 358
ISCREDITNOTE 149, 207, 212, 222, 227, 232, 238, 397
ISDAILYALLOWANCEBILLABLE 61, 120, 122, 125, 129, 132, 135, 342
ISDEFAULT 16, 17, 39, 40, 43, 54, 97, 109, 145, 146, 243, 267, 268, 272, 273, 274, 329, 336, 361, 370, 388, 389, 390, 391
ISDURATION 32, 34, 36, 50, 334
ISEU 146, 147, 148
ISINPROGRESS 102, 300, 301
ISINTERNAL 18, 19, 20, 22, 26, 27, 29, 30, 44, 61, 70, 120, 122, 125, 129, 132, 135, 158, 160, 161, 267, 268, 331, 342, 349
ISLOCKED 94, 203, 261, 263, 265, 266, 329, 358
ISLOST 102, 300, 301
ISMILEAGEBILLABLE 61, 120, 122, 125, 129, 132, 135, 342
ISO Code 156, 157

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

ISO Code A2 146, 147, 148
ISO Code A3 146, 147, 148
ISOCODE 156, 157
ISOCODEA2 146, 147, 148
ISOCODEA3 146, 147, 148
ISOFFER 102, 300, 301
isoName 147
ISOTHERTRAVELEXPENSESBILLABLE 61, 120, 122, 125, 129, 132, 135, 342
ISOUTSOURCINGCOST 74, 86, 91, 243, 246, 249, 251, 254, 284, 287, 317
ISOVERTIMEALLOWED 73, 177, 178, 352
ISPAID 243
ISPAIDLEAVE 39, 40, 54, 336
ISPOSTALADDRESS 56, 110, 111, 113, 187, 188, 337
ISPRIVATE 32, 34, 36, 50, 334
ISPRIVATEPERSON 18, 19, 20, 22, 26, 27, 29, 30, 44, 70, 158, 160, 161, 331, 349
ISPRODUCTIVE 80, 109, 189, 192, 194, 196, 199, 201, 217, 277, 280, 282, 353, 370, 388, 389, 390, 391, 394
ISREADONLY 32, 34, 36, 50, 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 334, 353, 394
ISREALISED 74, 86, 91, 243, 246, 249, 251, 254, 284, 287, 317
ISREIMBURSED 149, 207, 212, 222, 227, 232, 238, 397
ISSENT 243
ISTAXFREE 149, 207, 212, 222, 227, 232, 238, 397
ISTRAVELTYPEDEFAULT 275
ISUNASSIGNED 32, 34, 36, 50, 334
ISVISITADDRESS 56, 110, 111, 113, 187, 188, 337
ISVOLUMEPRICING 267, 268
ISWAITINGPAYMENT 243
ISWON 102, 300, 301
ISWORKTIME 260, 261
Item by GUID 243
Item by Purchase Order Number 246
ItemByGUID 243
ItemByPurchaseOrderNumber 246
itemGUID 170, 243, 284
itemInfo_BillingDependencyTaskGUID 74, 86
itemInfo_BillingSchedule 74, 86
itemInfo_CaseGUID 74, 86
itemInfo_CaseNumber 74, 86
itemInfo_CostCenterGUID 74, 86
itemInfo_CostCenterIdentifier 74, 86
itemInfo_CostCurrencyShortform 74, 86
itemInfo_CostCurrencyShortformOfInvoicingCurrency 74, 86
itemInfo_CostCurrencyShortformOfOrganizationCurrency 74, 86
itemInfo_CostCurrencyShortformOfUserCompanyCurrency 74, 86
itemInfo_Description 74, 86
itemInfo_DisplayPeriodStart 74, 86
itemInfo_EventDate 74, 86
itemInfo_GUID 74, 86
itemInfo_InvoiceDescription 74, 86
itemInfo_InvoiceGUID 74, 86
itemInfo_InvoiceQuantity 74, 86
itemInfo_InvoiceRowGUID 74, 86
itemInfo_IsApproved 74, 86
itemInfo_IsBillable 74, 86
itemInfo_IsOutSourcingCost 74, 86
itemInfo_IsReadOnly 74, 86
itemInfo_IsRealised 74, 86
itemInfo_MeasurementUnit 74, 86
itemInfo_Name 74, 86
itemInfo_OrderDate 74, 86
itemInfo_PlannedBillingDate 74, 86
itemInfo_PriceCurrencyShortformOfCaseCompanyCurrency 74, 86
itemInfo_PriceCurrencyShortformOfInvoicingCurrency 74, 86
itemInfo_PriceCurrencyShortformOfOrganizationCurrency 74, 86
itemInfo_PriceCurrencyShortformOfUserCompanyCurrency 74, 86
itemInfo_ProductCode 74, 86
itemInfo_ProductGUID 74, 86
itemInfo_ProductType 74, 86
itemInfo_PurchaseOrderNumber 74, 86
itemInfo_PurchaseVAT 74, 86
itemInfo_Quantity 74, 86
itemInfo_RecurrenceEndDate 74, 86
itemInfo_RecurrenceEndType 74, 86
itemInfo_RecurrenceFrequency 74, 86
itemInfo_RecurrenceStartDate 74, 86
itemInfo_RecurrenceTimes 74, 86
itemInfo_RecurringItemGUID 74, 86
itemInfo_SalesAccountGUID 74, 86
itemInfo_SalesStatusGUID 74, 86
itemInfo_TaskGUID 74, 86
itemInfo_TravelEndTime 74, 86
itemInfo_TravelStartTime 74, 86
itemInfo_UnitCost 74, 86
itemInfo_UnitCostInInvoicingCurrency 74, 86
itemInfo_UnitCostInOrganizationCurrency 74, 86
itemInfo_UnitCostInUserCompanyCurrency 74, 86
itemInfo_UnitCostVatIncludedAmount 74, 86
itemInfo_UnitPrice 74, 86

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

itemInfo_UnitPriceInBaseCurrency 74, 86
itemInfo_UnitPriceInCaseCompanyCurrency 74, 86
itemInfo_UnitPriceInUserCompanyCurrency 74, 86
itemInfo_UserCode 74, 86
itemInfo_UserGUID 74, 86
itemInfo_VAT 74, 86
items 91
Items by Insert Date 251
Items by Project GUID 249
Items Changed since 254
ItemsByCaseGUID 249
ItemsByInsertDate 251
ItemsChangedSince 254

- J -

Job Title 67, 140, 141, 143, 144, 348
JOBTITLE 67, 140, 141, 143, 144, 348

- K -

KEYWORD 66, 69, 79, 108, 139, 308, 309, 386
KEYWORDS 67, 140, 141, 143, 144, 348
KVKNUMBER 139

- L -

Language by GUID 257
Language by IETF Tag 258
Language Code 18, 19, 20, 22, 26, 27, 29, 30, 44, 59, 70, 105, 114, 116, 117, 118, 128, 149, 158, 160, 161, 207, 212, 222, 227, 232, 238, 331, 340, 349, 366, 371, 373, 375, 377, 379, 381, 382, 384, 397
Language GUID 18, 19, 20, 22, 26, 27, 29, 30, 44, 59, 67, 70, 105, 114, 116, 117, 118, 128, 139, 140, 141, 143, 144, 149, 158, 160, 161, 207, 212, 222, 227, 232, 238, 331, 340, 348, 349, 366, 371, 373, 375, 377, 379, 381, 382, 384, 397
LanguageByGUID 257
LanguageByIetfTag 258
LANGUAGECODE 18, 19, 20, 22, 26, 27, 29, 30, 44, 59, 70, 105, 114, 116, 117, 118, 128, 149, 158, 160, 161, 207, 212, 222, 227, 232, 238, 331, 340, 349, 366, 371, 373, 375, 377, 379, 381, 382, 384, 397
LANGUAGEGUID 18, 19, 20, 22, 26, 27, 29, 30, 44, 59, 67, 70, 105, 114, 116, 117, 118, 128, 139, 140, 141, 143, 144, 149, 158, 160, 161, 207, 212, 222, 227, 232, 238, 257, 331, 340, 348, 349, 366, 371, 373, 375, 377, 379, 381, 382, 384, 397
Last Name 67, 105, 128, 140, 141, 143, 144, 348, 366, 371, 373, 375, 377, 379, 381, 382, 384
LASTNAME 67, 105, 128, 140, 141, 143, 144, 348, 366, 371, 373, 375, 377, 379, 381, 382, 384
Latest Estimation Date 61, 120, 122, 125, 129, 132, 135, 342, 394
LATESTESTIMATIONDATE 61, 120, 122, 125, 129, 132, 135, 342
Lead Rating 61, 120, 122, 125, 129, 132, 135, 342, 394
Lead Source by GUID 258
Lead Source GUID 61, 120, 122, 125, 129, 132, 135, 342, 394
Lead Sources 259
LEADRATING 61, 120, 122, 125, 129, 132, 135, 342
LeadSourceByGUID 258
LEADSOURCEGUID 61, 120, 122, 125, 129, 132, 135, 258, 342
LeadSources 259
LICENSETYPE 105, 128, 366, 371, 373, 375, 377, 379, 381, 382, 384
LOCATION 32, 34, 36, 50, 334
log-native-calls-to-disk-max-events 2
log-native-calls-to-disk-max-seconds 2
log-native-calls-to-disk-on-error 2
log-native-calls-to-disk-on-success 2
log-native-calls-to-trace 2

- M -

Main Case GUID 149, 207, 212, 222, 227, 232, 238, 397
MAINCASEGUID 149, 207, 212, 222, 227, 232, 238, 397
maximum-length-identifiers 2
maxInvoices 232
maxRows 36
max-url-length-accepted 2
max-url-length-desired 2
Measurement Unit 74, 86, 91, 96, 104, 243, 246, 249, 251, 254, 269, 270, 271, 276, 284, 287, 314, 315, 316, 317, 320, 360, 364
MEASUREMENTUNIT 74, 86, 91, 96, 104, 243, 246, 249, 251, 254, 269, 270, 271, 276, 284, 287, 314, 315, 316, 317, 320, 360, 364
members 94
memberStatus 55
MIMETYPE 206
MIME-type 206
minimum-length-text 2
MONTH 57, 113, 339

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

- N -

Name 16, 17, 18, 19, 20, 22, 23, 24, 25, 26, 27, 29, 30, 32, 34, 35, 36, 38, 39, 40, 41, 43, 44, 48, 50, 54, 59, 61, 65, 70, 74, 84, 86, 91, 94, 96, 97, 98, 102, 104, 109, 114, 116, 117, 118, 120, 122, 125, 129, 132, 135, 137, 138, 139, 145, 146, 147, 148, 158, 160, 161, 183, 185, 203, 204, 206, 243, 246, 249, 251, 254, 257, 258, 259, 260, 261, 263, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 284, 287, 299, 300, 301, 313, 314, 315, 316, 317, 320, 321, 329, 331, 334, 336, 340, 342, 347, 349, 358, 360, 361, 362, 364, 370, 388, 389, 390, 391, 394
Native Platform Scalar Requests 392
NATIVEPLATFORMSCALARREQUESTS 392
Next Invoice Number 59, 114, 116, 117, 118, 340
NEXTINVOICENUMBER 59, 114, 116, 117, 118, 340
NOTES 18, 19, 20, 22, 26, 27, 29, 30, 32, 34, 36, 44, 50, 57, 70, 105, 113, 128, 149, 158, 160, 161, 207, 212, 222, 227, 232, 238, 322, 323, 325, 326, 328, 331, 334, 339, 349, 366, 371, 373, 375, 377, 379, 381, 382, 384, 397
Notes after Invoice Lines 149, 207, 212, 222, 227, 232, 238, 397
Notes before Invoice Lines 149, 207, 212, 222, 227, 232, 238, 397
NOTESAFTERINVOICEROWS 149, 207, 212, 222, 227, 232, 238, 397
NOTESBEFOREINVOICEROWS 149, 207, 212, 222, 227, 232, 238, 397
npt 392
NUMBER 18, 19, 20, 22, 25, 26, 27, 29, 30, 44, 72, 149, 158, 160, 161, 207, 212, 222, 227, 232, 238, 275, 322, 323, 325, 326, 328, 331, 349, 397

- O -

oauth-unauthorized-max-tries 2
oauth-unauthorized-sleep-initial-ms 2
oauth-unauthorized-sleep-max-ms 2
oauth-unauthorized-sleep-multiplicator 2
onlyActive 262
options 23, 24, 25, 29, 30, 111, 117, 118, 135, 144, 160, 161, 185, 186, 222, 227, 232, 266, 272, 276, 320, 325, 326, 328, 384, 391
Order Date 74, 86, 91, 243, 246, 249, 251, 254, 284, 287, 317
Order Number 61, 120, 122, 125, 129, 132, 135, 149, 207, 212, 222, 227, 232, 238, 342, 394, 397
ORDERDATE 74, 86, 91, 243, 246, 249, 251, 254, 284, 287, 317
ORDERNUMBER 61, 120, 122, 125, 129, 132, 135, 149, 207, 212, 222, 227, 232, 238, 342, 397
Organisation GUID 105, 128, 366, 371, 373, 375, 377, 379, 381, 382, 384
ORGANIZATIONGUID 105, 128, 366, 371, 373, 375, 377, 379, 381, 382, 384
ORIG_SYSTEM_GROUP 392
ORIG_SYSTEM_REFERENCE 392
Original Deadline 94, 261, 263, 265, 266, 358
Original Planned Start Date 94, 261, 263, 265, 266, 358
Original System Group 392
Original System Reference 392
Original Work Estimate 94, 261, 263, 265, 266, 358
ORIGINALDEADLINE 94, 261, 263, 265, 266, 358
ORIGINALPLANNEDSTARTDATE 94, 261, 263, 265, 266, 358
ORIGINALWORKESTIMATE 94, 261, 263, 265, 266, 358
Our Reference 61, 120, 122, 125, 129, 132, 135, 149, 207, 212, 222, 227, 232, 238, 342, 394, 397
OURREFERENCE 61, 120, 122, 125, 129, 132, 135, 149, 207, 212, 222, 227, 232, 238, 342, 397
Outsourcing Revenue Forecast 57, 113, 339
OUTSOURCINGREVENUEFORECAST 57, 113, 339
Overdue Interest 18, 19, 20, 22, 26, 27, 29, 30, 44, 59, 61, 70, 114, 116, 117, 118, 120, 122, 125, 129, 132, 135, 149, 158, 160, 161, 207, 212, 222, 227, 232, 238, 331, 340, 342, 349, 394, 397
OVERDUEINTEREST 18, 19, 20, 22, 26, 27, 29, 30, 44, 59, 61, 70, 114, 116, 117, 118, 120, 122, 125, 129, 132, 135, 149, 158, 160, 161, 207, 212, 222, 227, 232, 238, 331, 340, 342, 349, 397
Overtime Allowed 387
Overtime by GUID 260
Overtime Code 80, 189, 192, 194, 196, 199, 201, 217, 277, 280, 282, 353, 394
Overtime GUID 80, 189, 192, 194, 196, 199, 201, 217, 277, 280, 282, 353, 394
OVERTIMEALLOWED 387
OverTimeByGUID 260
OVERTIMECODE 80, 189, 192, 194, 196, 199, 201, 217, 277, 280, 282, 353, 394
OVERTIMEGUID 80, 189, 192, 194, 196, 199, 201, 217, 260, 277, 280, 282, 353, 394
Overtimes 261
Owner User Business Unit Code 32, 34, 36, 50, 334
Owner User Code 32, 34, 36, 50, 334
Owner User First Name 32, 34, 36, 50, 334
Owner User GUID 32, 34, 36, 50, 94, 261, 263, 265, 266, 334, 358
Owner User Last Name 32, 34, 36, 50, 334

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

OWNERUSERBUSINESSUNITCODE 32, 34, 36, 50, 334
OWNERUSERCODE 32, 34, 36, 50, 334
OWNERUSERFIRSTNAME 32, 34, 36, 50, 334
OWNERUSERGUID 32, 34, 36, 50, 94, 261, 263, 265, 266, 334, 358
OWNERUSERLASTNAME 32, 34, 36, 50, 334

- P -

Parent Business Unit GUID 59, 114, 116, 117, 118, 340
Parent GUID 23, 24, 48
Parent Phase GUID 94, 203, 261, 263, 265, 266, 358
PARENTBUSINESSUNITGUID 59, 114, 116, 117, 118, 340
PARENTGUID 23, 24, 48
PARENTPHASEGUID 94, 203, 261, 263, 265, 266, 358
partition-slot-based-rate-limit-length-ms 2
partition-slot-based-rate-limit-slots 2
Payload 392
PAYLOAD_TEXT 392
Payment Date 149, 207, 212, 222, 227, 232, 238, 397
Payment Status 149, 207, 212, 222, 227, 232, 238, 397
Payment Term 18, 19, 20, 22, 26, 27, 29, 30, 44, 59, 61, 70, 114, 116, 117, 118, 120, 122, 125, 129, 132, 135, 149, 158, 160, 161, 207, 212, 222, 227, 232, 238, 331, 340, 342, 349, 394, 397
PAYMENTDATE 149, 207, 212, 222, 227, 232, 238, 305, 397
PAYMENTSTATUS 149, 207, 212, 222, 227, 232, 238, 397
PAYMENTTERM 18, 19, 20, 22, 26, 27, 29, 30, 44, 59, 61, 70, 114, 116, 117, 118, 120, 122, 125, 129, 132, 135, 149, 158, 160, 161, 207, 212, 222, 227, 232, 238, 331, 340, 342, 349, 397
PERCENTAGE 260, 261
Phase by GUID 261
Phase GUID 80, 99, 101, 189, 192, 194, 196, 199, 201, 217, 277, 280, 282, 293, 294, 295, 296, 297, 298, 310, 311, 312, 353, 363, 394
Phase Members by Phase GUID 262
PhaseByGUID 261
phaseCode 265
PHASEGUID 80, 99, 101, 171, 189, 192, 194, 196, 199, 201, 217, 261, 262, 277, 280, 282, 293, 294, 295, 296, 297, 298, 306, 310, 311, 312, 353, 363, 390, 394
phaseGUIDToTransferItemsTo 171
phaseInfo_CaseGUID 94, 358
phaseInfo_Code 94, 358
phaseInfo_Deadline 94, 358
phaseInfo_Description 94, 358
phaseInfo_FlatRate_AdditionalHoursUnitPrice 94, 358
phaseInfo_FlatRate_AreAdditionalHoursBillable 94, 358
phaseInfo_FlatRate_Hours 94, 358
phaseInfo_FlatRate_Price 94, 358
phaseInfo_GUID 94, 358
phaseInfo_IsCompleted 94, 358
phaseInfo_IsLocked 94, 358
phaseInfo_Name 94, 358
phaseInfo_OriginalDeadline 94, 358
phaseInfo_OriginalPlannedStartDate 94, 358
phaseInfo_OriginalWorkEstimate 94, 358
phaseInfo_OwnerUserGUID 94, 358
phaseInfo_ParentPhaseGUID 94, 358
phaseInfo_PlannedStartDate 94, 358
phaseInfo_PricePerHour 94, 358
phaseInfo_SortOrder 94, 358
phaseInfo_WorkEstimate 94, 358
phaseInfo_WorkTypeGUID 94, 358
PhaseMembersByPhaseGUID 262
Phases by Code 265
Phases by Project GUID 263
Phases Changed since 266
PhasesByCaseGUID 263
PhasesByCode 265
PhasesChangedSince 266
PHONE 56, 105, 110, 111, 113, 128, 187, 188, 337, 366, 371, 373, 375, 377, 379, 381, 382, 384
Photo File GUID 105, 128, 366, 371, 373, 375, 377, 379, 381, 382, 384
PHOTOFILEGUID 105, 128, 366, 371, 373, 375, 377, 379, 381, 382, 384
Planned Billing Date 74, 86, 91, 243, 246, 249, 251, 254, 284, 287, 317
Planned Start Date 94, 261, 263, 265, 266, 358
PLANNEDBILLINGDATE 74, 86, 91, 243, 246, 249, 251, 254, 284, 287, 317
PLANNEDSTARTDATE 94, 261, 263, 265, 266, 358
PO Number 149, 207, 212, 222, 227, 232, 238, 397
PONUMBER 74, 149, 207, 212, 222, 227, 232, 238, 246, 397
Post Office 105, 128, 366, 371, 373, 375, 377, 379, 381, 382, 384
Postal Code 56, 105, 110, 111, 113, 128, 187, 188, 337, 366, 371, 373, 375, 377, 379, 381, 382, 384
POSTALCODE 56, 105, 110, 111, 113, 128, 187, 188, 337, 366, 371, 373, 375, 377, 379, 381, 382, 384
POSTOFFICE 105, 128, 366, 371, 373, 375, 377, 379, 381, 382, 384

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

pre-request-delay-ms 2
Price Currency Short Form of Invoicing Currency 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 353, 394
Price Currency Short Form of Organisation Currency 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 353, 394
Price Currency Short Form of Project Company Currency 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 353, 394
Price Currency Short Form of User Company Currency 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 353, 394
Price List by GUID 267
Price Lists 268
Price per Hour 94, 261, 263, 265, 266, 358
PRICECURRENCYSHORTFORMOFCASECOMPANYCURRENCY 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 353, 394
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 353, 394
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 353, 394
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 353, 394
Pricelist GUID 18, 19, 20, 22, 26, 27, 29, 30, 44, 61, 70, 120, 122, 125, 129, 132, 135, 158, 160, 161, 331, 342, 349, 394
PricelistByGUID 267
PRICELISTGUID 18, 19, 20, 22, 26, 27, 29, 30, 44, 61, 70, 120, 122, 125, 129, 132, 135, 158, 160, 161, 267, 331, 342, 349
Pricelists 268
PRICEPERHOUR 94, 261, 263, 265, 266, 358
Pricing Rule 61, 120, 122, 125, 129, 132, 135, 342, 394
Pricing Type 61, 120, 122, 125, 129, 132, 135, 342, 394
PRICINGRULE 61, 120, 122, 125, 129, 132, 135, 342
PRICINGTYPE 61, 120, 122, 125, 129, 132, 135, 342
Print Order 206
PRINTORDER 206
Priority 61, 120, 122, 125, 129, 132, 135, 342, 394
Probability 61, 120, 122, 125, 129, 132, 135, 342, 394
Product by Code 269
Product by Code Include Inactive 270
Product by GUID 271
Product Categories Changed since 272
Product Category by Code 273
Product Category by GUID 274
Product Category GUID 96, 269, 270, 271, 276, 360
Product Code 74, 86, 91, 243, 246, 249, 251, 254, 284, 287, 317
Product Cost Account Associations by GUID 275
Product GUID 74, 86, 91, 243, 246, 249, 251, 254, 284, 287, 317
Product Type 74, 86, 91, 243, 246, 249, 251, 254, 284, 287, 317
ProductByCode 269
ProductByCodeIncludeInactive 270
ProductByGUID 271
ProductCategoriesChangedSince 272
ProductCategoryByCode 273
ProductCategoryByGUID 274
productCategoryCode 273
PRODUCTCATEGORYGUID 96, 172, 269, 270, 271, 274, 276, 360
productCategoryInfo_Code 97, 361
productCategoryInfo_GUID 97, 361
productCategoryInfo_IsActive 97, 361
productCategoryInfo_IsDefault 97, 361
productCategoryInfo_Name 97, 361
PRODUCTCODE 74, 86, 91, 243, 246, 249, 251, 254, 269, 270, 284, 287, 317
ProductCostAccountAssociationsByGUID 275
PRODUCTGUID 74, 86, 91, 172, 243, 246, 249, 251, 254, 271, 275, 284, 287, 317
productInfo_Code 96, 360
productInfo_Description 96, 360
productInfo_GUID 96, 360
productInfo_IsActive 96, 360
productInfo_MeasurementUnit 96, 360
productInfo_Name 96, 360
productInfo_ProductCategoryGUID 96, 360
productInfo_SalesAccountGUID 96, 360
productInfo_Type 96, 360
productInfo_UnitCost 96, 360
productInfo_UnitPrice 96, 360
productInfo_UseInWorkTimeEntry 96, 360
productInfo_VAT 96, 360
Products Changed since 276
ProductsChangedSince 276

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

PRODUCTTYPE 74, 86, 91, 243, 246, 249, 251, 254, 284, 287, 317
Profile GUID 105, 128, 366, 371, 373, 375, 377, 379, 381, 382, 384
PROFILEGUID 105, 128, 366, 371, 373, 375, 377, 379, 381, 382, 384
Project Business Unit Code 80, 189, 192, 194, 196, 199, 201, 217, 277, 280, 282, 353, 394
Project by GUID 120
Project by Number 122
Project by Task GUID 125
Project Company Currency GUID 61, 120, 122, 125, 129, 132, 135, 342, 394
Project Cost Centre Identifier 80, 189, 192, 194, 196, 199, 201, 217, 277, 280, 282, 353, 394
Project GUID 32, 34, 36, 50, 74, 80, 86, 91, 94, 99, 101, 103, 182, 189, 192, 194, 196, 199, 201, 203, 217, 243, 246, 249, 251, 254, 261, 263, 265, 266, 277, 280, 282, 284, 287, 293, 294, 295, 296, 297, 298, 302, 303, 317, 334, 353, 358, 363, 394
Project Member Users by Project GUID 128
Project Name 80, 189, 192, 194, 196, 199, 201, 217, 277, 280, 282, 353, 394
Project Number 61, 74, 80, 86, 91, 120, 122, 125, 129, 132, 135, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 342, 353, 394
Project Owner User Code 61, 120, 122, 125, 129, 132, 135, 342, 394
Project Owner User GUID 61, 120, 122, 125, 129, 132, 135, 342, 394
Project Status by GUID 137
Project Statuses 138
Project Tags 139
Projects 129, 219
Projects by Account GUID 132
Projects Changed since 135
Purchase Order Number 74, 86, 91, 243, 246, 249, 251, 254, 284, 287, 317
Purchase VAT 74, 86, 91, 243, 246, 249, 251, 254, 284, 287, 317
PURCHASEORDERNUMBER 74, 86, 91, 243, 246, 249, 251, 254, 284, 287, 317
PURCHASEVAT 74, 86, 91, 243, 246, 249, 251, 254, 284, 287, 317

- Q -

QUANTITY 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 310, 311, 312, 317, 353, 394

- R -

RATE 156, 157
READONLY 387
Read-only 387
READY 387
Receiver Addressline 149, 207, 212, 222, 227, 232, 238, 397
Receiver City 149, 207, 212, 222, 227, 232, 238, 397
Receiver Contact Email 149, 207, 212, 222, 227, 232, 238, 397
Receiver Contact Name 149, 207, 212, 222, 227, 232, 238, 397
Receiver Country 149, 207, 212, 222, 227, 232, 238, 397
Receiver Customer Name 149, 207, 212, 222, 227, 232, 238, 397
Receiver Postal Code 149, 207, 212, 222, 227, 232, 238, 397
Receiver State 149, 207, 212, 222, 227, 232, 238, 397
Receiver VAT Number 149, 207, 212, 222, 227, 232, 238, 397
RECEIVERADDRESSLINE 149, 207, 212, 222, 227, 232, 238, 397
RECEIVERCITY 149, 207, 212, 222, 227, 232, 238, 397
RECEIVERCONTACTEMAIL 149, 207, 212, 222, 227, 232, 238, 397
RECEIVERCONTACTNAME 149, 207, 212, 222, 227, 232, 238, 397
RECEIVERCOUNTRY 149, 207, 212, 222, 227, 232, 238, 397
RECEIVERCUSTOMERNAME 149, 207, 212, 222, 227, 232, 238, 397
RECEIVERPOSTALCODE 149, 207, 212, 222, 227, 232, 238, 397
RECEIVERSTATE 149, 207, 212, 222, 227, 232, 238, 397
RECEIVERVATNUMBER 149, 207, 212, 222, 227, 232, 238, 397
Recurrence End Date 74, 86, 91, 243, 246, 249, 251, 254, 284, 287, 317
Recurrence End Type 74, 86, 91, 243, 246, 249, 251, 254, 284, 287, 317
Recurrence Frequency 74, 86, 91, 243, 246, 249, 251, 254, 284, 287, 317
Recurrence Start Date 74, 86, 91, 243, 246, 249, 251, 254, 284, 287, 317
Recurrence Times 74, 86, 91, 243, 246, 249, 251, 254, 284, 287, 317

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

RECURRENCEENDDATE 74, 86, 91, 243, 246, 249, 251, 254, 284, 287, 317
RECURRENCEENDTYPE 74, 86, 91, 243, 246, 249, 251, 254, 284, 287, 317
RECURRENCEFREQUENCY 74, 86, 91, 243, 246, 249, 251, 254, 284, 287, 317
RECURRENCESTARTDATE 74, 86, 91, 243, 246, 249, 251, 254, 284, 287, 317
RECURRENCETIMES 74, 86, 91, 243, 246, 249, 251, 254, 284, 287, 317
Recurring Item GUID 74, 86, 91, 243, 246, 249, 251, 254, 284, 287, 317
RECURRINGITEMGUID 74, 86, 91, 243, 246, 249, 251, 254, 284, 287, 317
Reference Number 149, 207, 212, 222, 227, 232, 238, 397
REFERENCENUMBER 149, 207, 212, 222, 227, 232, 238, 304, 397
Reimburse Invoice GUID 149, 207, 212, 222, 227, 232, 238, 397
Reimbursed Hour Entries by Date 277
Reimbursed Hour Entries Changed since 280
Reimbursed Hour Entry by GUID 282
Reimbursed Item by GUID 284
Reimbursed Items Changed since 287
ReimbursedHourEntriesByDate 277
ReimbursedHourEntriesChangedSince 280
ReimbursedHourEntryByGUID 282
ReimbursedItemByGUID 284
ReimbursedItemsChangedSince 287
REIMBURSEINVOICEGUID 149, 207, 212, 222, 227, 232, 238, 397
Remove Account Group Members 289
Remove Activity Contact Participant 290
Remove Activity Resource Participant 291
Remove Activity User Participant 292
RemoveAccountGroupMembers 289
RemoveActivityContactParticipant 290
REMOVEACTIVITYCONTACTPARTICIPANTRESULT 290
RemoveActivityResourceParticipant 291
REMOVEACTIVITYRESOURCEPARTICIPANTRESULT 291
RemoveActivityUserParticipant 292
REMOVEACTIVITYUSERPARTICIPANTRESULT 292
replacementActivityTypeGUID 165
replacementCaseStatusGUID 167
requested-maximum-usable-partitions 2
requested-page-size 2
requests-parallel-max 2
Resource Allocations 294
Resource Allocations by GUID 293
Resource Allocations by Project GUID 295
Resource Allocations by Project GUID and User GUID 296
Resource Allocations by User GUID 297
Resource Allocations Changed since 298
Resource by GUID 299
Resource GUID 35, 38, 41
ResourceAllocationByGUID 293
resourceAllocationGUID 174, 293
resourceAllocationInfo_AllocatedHours 99, 363
resourceAllocationInfo_AllocatedPercentage 99, 363
resourceAllocationInfo_CaseGUID 99, 363
resourceAllocationInfo_EndDate 99, 363
resourceAllocationInfo_GUID 99, 363
resourceAllocationInfo_PhaseGUID 99, 363
resourceAllocationInfo_StartDate 99, 363
resourceAllocationInfo_UserGUID 99, 363
ResourceAllocations 101, 294
ResourceAllocationsByCaseGUID 295
ResourceAllocationsByCaseGUIDUserGUID 296
ResourceAllocationsByUserGUID 297
ResourceAllocationsChangedSince 298
ResourceByGUID 299
resourceGUID 35, 38, 41, 53, 173, 291, 299
resourceInfo_GUID 98, 362
resourceInfo_IsActive 98, 362
resourceInfo_Name 98, 362
Resources 300
Response Cache Maximum Age (sec) 392
Result BLOB 392
Result Date Time 392
Result Number 392
Result Text 392
RESULT_BLOB 392
RESULT_DATE_TIME_UTC 392
RESULT_NUMBER 392
RESULT_TEXT 392
result-set-memory-cache 2
Reverse Charge 18, 19, 20, 22, 26, 27, 29, 30, 44, 70, 158, 160, 161, 331, 349
Reverse Charge Description 18, 19, 20, 22, 26, 27, 29, 30, 44, 70, 158, 160, 161, 331, 349
REVERSECHARGE 18, 19, 20, 22, 26, 27, 29, 30, 44, 70, 158, 160, 161, 331, 349
REVERSECHARGEDESCRIPTION 18, 19, 20, 22, 26, 27, 29, 30, 44, 70, 158, 160, 161, 331, 349
RIGHTS_ACCOUNT 16, 17, 43, 329, 369, 376
RIGHTS_ACCOUNTDELETE 16, 17, 43, 329, 369, 376
RIGHTS_ADMINISTRATOR 16, 17, 43, 329, 369, 376

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

rights_API 369
RIGHTS_CASE 16, 17, 43, 329, 369, 376
RIGHTS_CASEDELETE 16, 17, 43, 329, 369, 376
RIGHTS_CASEOWNERWORKHOURAPPROVAL 16, 17, 43, 329, 369, 376
RIGHTS_EDITCASESTATUS 180
rights_Rights_Account 369
rights_Rights_AccountDelete 369
rights_Rights_Administrator 369
rights_Rights_Case 369
rights_Rights_CaseDelete 369
rights_Rights_CaseOwnerWorkHourApproval 369
rights_Rights_ScheduleJobs 369
rights_Rights_Sharing 369
rights_Rights_TravelReimbursement 369
rights_Rights_Users 369
rights_Rights_WorkHourApproval 369
RIGHTS_SCHEDULEJOBS 16, 17, 43, 329, 369, 376
RIGHTS_SHARING 16, 17, 43, 329, 369, 376
RIGHTS_SHOWBILLING 180
RIGHTS_SHOWCASESTATUS 180
RIGHTS_SHOWFILESANDLINKS 180
RIGHTS_SHOWGANTTCHART 180
RIGHTS_SHOWPHASETREE 180
RIGHTS_SHOWREADYTOBILL 180
RIGHTS_SHOWWORKHOURESTIMATES 180
RIGHTS_SHOWWORKHOURS 180
RIGHTS_SHOWWORKHOURSTARTANDENDTIME 180
RIGHTS_TRAVELREIMBURSEMENT 16, 17, 43, 329, 369, 376
RIGHTS_USERS 16, 17, 43, 329, 369, 376
RIGHTS_WORKHOURAPPROVAL 16, 17, 43, 329, 369, 376
Root Task GUID 61, 120, 122, 125, 129, 132, 135, 342, 394
rootBusinessUnitGuid 117
ROOTTASKGUID 61, 120, 122, 125, 129, 132, 135, 342
Run without Actions 392

- S -

Sales Account GUID 74, 86, 91, 96, 109, 243, 246, 249, 251, 254, 269, 270, 271, 276, 284, 287, 317, 360, 370, 388, 389, 390, 391
Sales Close Date 61, 120, 122, 125, 129, 132, 135, 342, 394
Sales Person User Code 61, 120, 122, 125, 129, 132, 135, 342, 394
Sales Person User GUID 61, 120, 122, 125, 129, 132, 135, 342, 394
Sales Process by GUID 300
Sales Process GUID 61, 103, 120, 122, 125, 129, 132, 135, 302, 303, 342, 394
Sales Processes 301
Sales Status by GUID 302
Sales Status GUID 61, 74, 86, 91, 120, 122, 125, 129, 132, 135, 243, 246, 249, 251, 254, 284, 287, 317, 342, 394
Sales Statuses for Project 303
SALESACCOUNTGUID 74, 86, 91, 96, 109, 243, 246, 249, 251, 254, 269, 270, 271, 276, 284, 287, 317, 360, 370, 388, 389, 390, 391
SALESCLOSEDATE 61, 120, 122, 125, 129, 132, 135, 342
SALESPERSONUSERCODE 61, 120, 122, 125, 129, 132, 135, 342
SALESPERSONUSERGUID 61, 120, 122, 125, 129, 132, 135, 342
SalesProcessByGUID 300
SalesProcesses 301
SALESPROCESSGUID 61, 103, 120, 122, 125, 129, 132, 135, 300, 302, 303, 342
salesProcessGUIDs 294
salesProcessInfo_DefaultProbability 102
salesProcessInfo_GUID 102
salesProcessInfo_Icon 102
salesProcessInfo_IsActive 102
salesProcessInfo_IsAlsoItemStatus 102
salesProcessInfo_IsInProgress 102
salesProcessInfo_IsLost 102
salesProcessInfo_IsOffer 102
salesProcessInfo_IsWon 102
salesProcessInfo_Name 102
SalesStatusByGUID 302
SalesStatusesForCase 303
SALESSTATUSGUID 61, 74, 86, 91, 120, 122, 125, 129, 132, 135, 243, 246, 249, 251, 254, 284, 287, 302, 317, 342
salesStatusInfo_CaseGUID 103
salesStatusInfo_GUID 103
salesStatusInfo_SalesProcessGUID 103
salesStatusInfo_TimeStamp 103
SALUTATION 67, 105, 128, 140, 141, 143, 144, 348, 366, 371, 373, 375, 377, 379, 381, 382, 384
Satisfaction Level 67, 140, 141, 143, 144, 348
SATISFACTIONLEVEL 67, 140, 141, 143, 144, 348
Save HTTP Disk Cache 392
Save HTTP Memory Cache 392
SELLERCONTACT_SELLERACCOUNTDETAILS 149, 207, 212, 222, 227, 232, 238, 397

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

SELLERCONTACT_SELLERCONTACTPERSONNAME 149, 207, 212, 222, 227, 232, 238, 397
SELLERCONTACT_SELLEREMAILADDRESSIDENTIFIER 149, 207, 212, 222, 227, 232, 238, 397
SELLERCONTACT_SELLERFAXNUMBER 149, 207, 212, 222, 227, 232, 238, 397
SELLERCONTACT_SELLERFREETEXT 149, 207, 212, 222, 227, 232, 238, 397
SELLERCONTACT_SELLERHOMETOWNNAME 149, 207, 212, 222, 227, 232, 238, 397
SELLERCONTACT_SELLERPHONENUMBER 149, 207, 212, 222, 227, 232, 238, 397
SELLERCONTACT_SELLERWEBADDRESSIDENTIFIER 149, 207, 212, 222, 227, 232, 238, 397
SELLERORGANIZATION_SELLEREINVOICEADDRESS 149, 207, 212, 222, 227, 232, 238, 397
SELLERORGANIZATION_SELLEREINVOICEOPERATOR 149, 207, 212, 222, 227, 232, 238, 397
SELLERORGANIZATION_SELLERORGANISATIONNAME 149, 207, 212, 222, 227, 232, 238, 397
SELLERORGANIZATION_SELLERORGANISATIONTAXCODE 149, 207, 212, 222, 227, 232, 238, 397
SELLERORGANIZATION_SELLERPARTYIDENTIFIER 149, 207, 212, 222, 227, 232, 238, 397
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRY 149, 207, 212, 222, 227, 232, 238, 397
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRYCODE 149, 207, 212, 222, 227, 232, 238, 397
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER 149, 207, 212, 222, 227, 232, 238, 397
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_STREETNAME 149, 207, 212, 222, 227, 232, 238, 397
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_TOWNNAME 149, 207, 212, 222, 227, 232, 238, 397
Send Daily Mail 105, 128, 366, 371, 373, 375, 377, 379, 381, 382, 384
Send Weekly Mail 105, 128, 366, 371, 373, 375, 377, 379, 381, 382, 384
SENDDAILYMAIL 105, 128, 366, 371, 373, 375, 377, 379, 381, 382, 384
Sender Addressline 149, 207, 212, 222, 227, 232, 238, 397
Sender City 149, 207, 212, 222, 227, 232, 238, 397
Sender Contact Name 149, 207, 212, 222, 227, 232, 238, 397
Sender Country 149, 207, 212, 222, 227, 232, 238, 397
Sender Name 149, 207, 212, 222, 227, 232, 238, 397
Sender Postal Code 149, 207, 212, 222, 227, 232, 238, 397
Sender State 149, 207, 212, 222, 227, 232, 238, 397
Sender VAT Number 149, 207, 212, 222, 227, 232, 238, 397
SENDERADDRESSLINE 149, 207, 212, 222, 227, 232, 238, 397
SENDERCITY 149, 207, 212, 222, 227, 232, 238, 397
SENDERCONTACTNAME 149, 207, 212, 222, 227, 232, 238, 397
SENDERCOUNTRY 149, 207, 212, 222, 227, 232, 238, 397
SENDERNAME 149, 207, 212, 222, 227, 232, 238, 397
SENDERPOSTALCODE 149, 207, 212, 222, 227, 232, 238, 397
SENDERSTATE 149, 207, 212, 222, 227, 232, 238, 397
SENDERVATNUMBER 149, 207, 212, 222, 227, 232, 238, 397
SENDWEEKLYMAIL 105, 128, 366, 371, 373, 375, 377, 379, 381, 382, 384
SendWelcomeEmail 105
Server 304
SERVERRESULT 304
Set Invoice Number 304
Set Invoice Payment Date 305
Set Invoice Status 306
Set Phase Member List for Phase 306
Set Travel Reimbursement Status 307
SetInvoiceNumber 304
SetInvoicePaymentDate 305
SetInvoiceStatus 306
SetPhaseMemberListForPhase 306
SETPHASEMEMBERLISTFORPHASERESULT 306
SetTravelReimbursementStatus 307
severa 1
Shared to Extranet 61, 120, 122, 125, 129, 132, 135, 342, 394
SHAREDTOEXTRANET 61, 120, 122, 125, 129, 132, 135, 342
Show Logo on Invoices 59, 114, 116, 117, 118, 340
Show Logo on PDF Reports 59, 114, 116, 117, 118, 340
SHOWLOGOONINVOICES 59, 114, 116, 117, 118, 340
SHOWLOGOONPDFREPORTS 59, 114, 116, 117, 118, 340
simulate-http-400-errors 2
simulate-http-400-errors-percentage 2
simulate-http-401-errors 2

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

simulate-http-401-errors-percentage 2
simulate-http-403-errors 2
simulate-http-403-errors-percentage 2
simulate-http-408-errors 2
simulate-http-408-errors-percentage 2
simulate-http-429-errors 2
simulate-http-429-errors-percentage 2
simulate-http-500-errors 2
simulate-http-500-errors-percentage 2
simulate-http-502-errors 2
simulate-http-502-errors-percentage 2
simulate-http-503-errors 2
simulate-http-503-errors-percentage 2
simulate-http-504-errors 2
simulate-http-504-errors-percentage 2
simulate-http-522-errors 2
simulate-http-522-errors-percentage 2
simulate-http-524-errors 2
simulate-http-524-errors-percentage 2
simulate-http-protocol-errors 2
simulate-http-protocol-errors-percentage 2
simulate-http-timeout-errors 2
simulate-http-timeout-errors-percentage 2
SIZE 183, 185, 321
slot-based-rate-limit-length-ms 2
slot-based-rate-limit-slots 2
Social Security Number 105, 128, 366, 371, 373, 375, 377, 379, 381, 382, 384
SOCIALSECURITYNUMBER 105, 128, 366, 371, 373, 375, 377, 379, 381, 382, 384
Sort Order 94, 243, 261, 263, 265, 266, 329, 358
SORTORDER 94, 243, 261, 263, 265, 266, 329, 358
source 85
standardize-identifiers 2
standardize-identifiers-casing 2
Start Date 61, 73, 99, 101, 120, 122, 125, 129, 132, 135, 177, 178, 293, 294, 295, 296, 297, 298, 342, 352, 363, 392, 394
Start Date Time 32, 34, 36, 50, 334
Start Time 80, 189, 192, 194, 196, 199, 201, 217, 277, 280, 282, 310, 311, 312, 353, 394
startdate 32, 61, 73, 99, 101, 120, 122, 125, 129, 132, 135, 177, 178, 189, 192, 194, 196, 199, 222, 227, 254, 277, 280, 287, 293, 294, 295, 296, 297, 298, 317, 325, 342, 352, 363, 387
startDateTime 29, 30, 32, 34, 36, 50, 111, 118, 135, 144, 161, 266, 272, 276, 298, 312, 320, 334, 384, 391
startFromIndex 25, 160
startsAfter 36
startsBefore 36
STARTTIME 80, 189, 192, 194, 196, 199, 201, 217, 277, 280, 282, 310, 311, 312, 353, 394
status 35, 38, 41, 185, 227, 232, 238, 304, 306, 307, 326
STRING 262
Succesful 392
SUCCESSFUL 392
Superior User GUID 105, 128, 366, 371, 373, 375, 377, 379, 381, 382, 384
SUPERIORUSERGUID 105, 128, 366, 371, 373, 375, 377, 379, 381, 382, 384
supervisorUserGUIDs 294
SYMBOL 156, 157
System Name 313, 314
SYSTEMNAME 313, 314

- T -

Tag by GUID 308
TagByGUID 308
tagGUID 175, 308
tagInfo_Context 66, 69, 79, 108
tagInfo_GUID 66, 69, 79, 108
tagInfo_IsActive 66, 69, 79, 108
tagInfo_Keyword 66, 69, 79, 108
tagInfo_Weight 66, 69, 79, 108
Tags by Project GUID 309
TagsByCaseGUID 309
Task GUID 32, 34, 36, 50, 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 334, 353, 394
TASKGUID 32, 34, 36, 50, 74, 80, 86, 91, 125, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 334, 353, 394
Tax (%) 275
Tax Free Description 149, 207, 212, 222, 227, 232, 238, 397
TAXFREEDESCRIPTION 149, 207, 212, 222, 227, 232, 238, 397
TAXPERCENTAGE 275
text 85
Time Entries by Date 310
Time Entries by Date and User GUID 311
Time Entries Changed since 312
Time Stamp 103, 302, 303
TimeEntriesByDate 310
TimeEntriesByDateAndUserGUID 311
TimeEntriesChangedSince 312
Timeout (sec) 392
TIMEOUT_SEC 392
TIMESTAMP 103, 302, 303
Timezone by GUID 313

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Timezone GUID 59, 67, 70, 105, 114, 116, 117, 118, 128, 139, 140, 141, 143, 144, 146, 147, 148, 158, 160, 161, 340, 348, 349, 366, 371, 373, 375, 377, 379, 381, 382, 384
TimezoneByGUID 313
TIMEZONEGUID 59, 67, 70, 105, 114, 116, 117, 118, 128, 139, 140, 141, 143, 144, 146, 147, 148, 158, 160, 161, 313, 340, 348, 349, 366, 371, 373, 375, 377, 379, 381, 382, 384
Timezones 314
TITLE 73, 105, 128, 149, 177, 178, 207, 212, 222, 227, 232, 238, 352, 366, 371, 373, 375, 377, 379, 381, 382, 384, 397
Total Hours 387
TOTALHOURS 387
Transaction ID 392
TRANSACTION_ID 392
Translated Name 146, 147, 148
TRANSLATEDNAME 146, 147, 148
Travel End Time 74, 86, 91, 243, 246, 249, 251, 254, 284, 287, 317
Travel Expense by Code 314
Travel Expense by Code Include Inactive 315
Travel Expense by GUID 316
Travel Expenses by Date 317
Travel Expenses Changed since 320
Travel Reimbursement Attachments 321
Travel Reimbursement by Date 325
Travel Reimbursement by GUID 322
Travel Reimbursement by Number 323
Travel Reimbursement by Status 326
Travel Reimbursement by Status GUID 328
Travel Reimbursement PDF 324
Travel Reimbursement Status 322, 323, 325, 326, 328
Travel Reimbursement Statuses 329
Travel Start Time 74, 86, 91, 243, 246, 249, 251, 254, 284, 287, 317
TRAVELENDTIME 74, 86, 91, 243, 246, 249, 251, 254, 284, 287, 317
TravelExpenseByCode 314
TravelExpenseByCodeIncludeInactive 315
TravelExpenseByGUID 316
travelExpenseCode 314, 315
travelExpenseGUID 176, 316
travelExpenseInfo_Code 104, 364
travelExpenseInfo_CostCurrencyGUID 104, 364
travelExpenseInfo_Description 104, 364
travelExpenseInfo_ExpenseClass 104, 364
travelExpenseInfo_GUID 104, 364
travelExpenseInfo_IsActive 104, 364
travelExpenseInfo_MeasurementUnit 104, 364
travelExpenseInfo_Name 104, 364
travelExpenseInfo_Type 104, 364
travelExpenseInfo_UnitCost 104, 364
travelExpenseInfo_UnitPrice 104, 364
travelExpenseInfo_UseInWorkTimeEntry 104, 364
travelExpenseInfo_VAT 104, 364
TravelExpensesByDate 317
TravelExpensesChangedSince 320
TravelReimbursementAttachments 321
TravelReimbursementByGUID 322
TravelReimbursementByNumber 323
travelReimbursementGUID 307, 322
travelReimbursementNumber 323
travelReimbursementNumberingBusinessUnitGUID 323
TravelReimbursementPdf 324
TRAVELREIMBURSEMENTROWS 322, 323, 325, 326, 328
TravelReimbursementsByDate 325
TravelReimbursementsByStatus 326
TravelReimbursementsByStatusGUID 328
TRAVELREIMBURSEMENTSTATUS 322, 323, 325, 326, 328
TravelReimbursementStatuses 329
travelReimbursementStatusGUID 328
TRAVELSTARTTIME 74, 86, 91, 243, 246, 249, 251, 254, 284, 287, 317
TYPE 96, 104, 143, 269, 270, 271, 276, 314, 315, 316, 320, 360, 364

- U -

Unit Cost 74, 80, 86, 91, 96, 104, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 269, 270, 271, 276, 277, 280, 282, 284, 287, 314, 315, 316, 317, 320, 353, 360, 364, 394
Unit Cost in Invoicing Currency 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 353, 394
Unit Cost in Organisation Currency 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 353, 394
Unit Cost in User Company Currency 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 353, 394
Unit Cost VAT Included Amount 74, 86, 91, 243, 246, 249, 251, 254, 284, 287, 317
Unit Price 74, 80, 86, 91, 96, 104, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 269, 270, 271, 276, 277, 280, 282, 284, 287, 314, 315, 316, 317, 320, 353, 360, 364, 394
Unit Price in Base Currency 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 353, 394

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Unit Price in Project Company Currency 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 353, 394
Unit Price in User Company Currency 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 353, 394
UNITCOST 74, 80, 86, 91, 96, 104, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 269, 270, 271, 276, 277, 280, 282, 284, 287, 314, 315, 316, 317, 320, 353, 360, 364, 394
UNITCOSTININVOICINGCURRENCY 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 353, 394
UNITCOSTINORGANIZATIONCURRENCY 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 353, 394
UNITCOSTINUSERCOMPANYCURRENCY 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 353, 394
UNITCOSTVATINCLUDEDAMOUNT 74, 86, 91, 243, 246, 249, 251, 254, 284, 287, 317
UNITPRICE 74, 80, 86, 91, 96, 104, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 269, 270, 271, 276, 277, 280, 282, 284, 287, 314, 315, 316, 317, 320, 353, 360, 364, 394
UNITPRICEINBASECURRENCY 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 353, 394
UNITPRICEINCASECOMPANYCURRENCY 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 353, 394
UNITPRICEINUSERCOMPANYCURRENCY 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 353, 394
Update Access Rights Profile 329
Update Account 331
Update Activity 334
Update Activity Type 336
Update Address 337
Update Billing and Expense Forecast 339
Update Business Unit 340
Update Contact 348
Update Customer 349
Update Employment 352
Update Hour Entry 353
Update Phase 358
Update Product 360
Update Product Category 361
Update Project 342
Update Project Status 347
Update Resource 362
Update Resource Allocation 363
Update Travel Expense 364
Update User 366
Update User Rights by User GUID 369
Update Work Type 370
UpdateAccessRightsProfile 329
UpdateAccount 331
UpdateActivity 334
UpdateActivityType 336
UpdateAddress 337
UpdateBillingAndExpenseForecast 339
UpdateBusinessUnit 340
UpdateCase 342
UpdateCaseStatus 347
UpdateContact 348
UpdateCustomer 349
Updated 18, 19, 20, 22, 26, 27, 29, 30, 44, 59, 61, 70, 114, 116, 117, 118, 120, 122, 125, 129, 132, 135, 158, 160, 161, 331, 340, 342, 349, 394
UPDATEDTS 18, 19, 20, 22, 26, 27, 29, 30, 44, 59, 61, 70, 114, 116, 117, 118, 120, 122, 125, 129, 132, 135, 158, 160, 161, 331, 340, 342, 349
UpdateEmployment 352
UpdateHourEntry 353
UpdatePhase 358
UpdateProduct 360
UpdateProductCategory 361
UpdateResource 362
UpdateResourceAllocation 363
UpdateTravelExpense 364
UpdateUser 366
UpdateUserRightsByUserGUID 369
UpdateWorkType 370
URL 392
URL to Account 18, 19, 20, 22, 26, 27, 29, 30, 44, 70, 158, 160, 161, 331, 349
URL to Project 61, 120, 122, 125, 129, 132, 135, 342, 394
URLTOACCOUNT 18, 19, 20, 22, 26, 27, 29, 30, 44, 70, 158, 160, 161, 331, 349
URLTOCASE 61, 120, 122, 125, 129, 132, 135, 342
Use Default Products in Work Time Entry 61, 120, 122, 125, 129, 132, 135, 342, 394
Use Default Work Types 61, 120, 122, 125, 129, 132, 135, 342, 394
Use HTTP Disk Cache 392
Use HTTP Memory Cache 392
Use in Work Time Entry 96, 104, 269, 270, 271, 276, 314, 315, 316, 320, 360, 364
USEDEFAULTPRODUCTSINWORKTIMEENTRY 61, 120, 122, 125, 129, 132, 135, 342
USEDEFAULTWORKTYPES 61, 120, 122, 125, 129, 132, 135, 342
use-http-disk-cache 2
use-http-disk-cache-read 2
use-http-disk-cache-write 2

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

use-http-memory-cache 2
use-http-memory-cache-read 2
use-http-memory-cache-write 2
USEINVOICEQUANTITY 80, 189, 192, 194, 196, 199, 201, 217, 277, 280, 282, 353, 394
USEINWORKTIMEENTRY 96, 104, 269, 270, 271, 276, 314, 315, 316, 320, 360, 364
use-metadata-memory-cache 2
User Business Unit Code 80, 189, 192, 194, 196, 199, 201, 217, 277, 280, 282, 353, 394
User by Code 371
User by GUID 373
User by Name 375
User Code 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 353, 394
User First Name 80, 189, 192, 194, 196, 199, 201, 217, 277, 280, 282, 353, 394
User GUID 35, 38, 41, 74, 80, 86, 91, 99, 101, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 293, 294, 295, 296, 297, 298, 310, 311, 312, 317, 322, 323, 325, 326, 328, 353, 363, 394
User Last Name 80, 189, 192, 194, 196, 199, 201, 217, 277, 280, 282, 353, 394
User Rights by User GUID 376
User Tags 386
USERBUSINESSUNITCODE 80, 189, 192, 194, 196, 199, 201, 217, 277, 280, 282, 353, 394
UserByCode 371
UserByGUID 373
UserByName 375
USERCODE 74, 80, 86, 91, 189, 192, 194, 196, 199, 201, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 317, 353, 371, 394
use-result-memory-cache 2
USERFIRSTNAME 80, 189, 192, 194, 196, 199, 201, 217, 277, 280, 282, 353, 394
USERGUID 35, 36, 38, 41, 42, 55, 73, 74, 80, 86, 91, 99, 101, 108, 176, 178, 179, 189, 192, 194, 196, 199, 201, 203, 217, 243, 246, 249, 251, 254, 277, 280, 282, 284, 287, 292, 293, 294, 295, 296, 297, 298, 310, 311, 312, 317, 322, 323, 325, 326, 328, 353, 363, 373, 387, 394
userGUIDs 294, 306, 381
userInfo_Addressline 105, 366
userInfo_BankAccountNumber 105, 366
userInfo_BusinessUnitCode 105, 366
userInfo_BusinessUnitGUID 105, 366
userInfo_BusinessUnitName 105, 366
userInfo_Code 105, 366
userInfo_CountryCode 105, 366
userInfo_CountryGUID 105, 366
userInfo_CountryRegionGUID 105, 366
userInfo_CountryRegionName 105, 366
userInfo_Culture 105, 366
userInfo_DefaultActivityTypeGUID 105, 366
userInfo_Email 105, 366
userInfo_FirstName 105, 366
userInfo_GUID 105, 366
userInfo_IsActive 105, 366
userInfo_LanguageCode 105, 366
userInfo_LanguageGUID 105, 366
userInfo_LastName 105, 366
userInfo_LicenseType 105, 366
userInfo_Notes 105, 366
userInfo_OrganizationGUID 105, 366
userInfo_Phone 105, 366
userInfo_PhotoFileGUID 105, 366
userInfo_PostalCode 105, 366
userInfo_PostOffice 105, 366
userInfo_ProfileGUID 105, 366
userInfo_Salutation 105, 366
userInfo_SendDailyMail 105, 366
userInfo_SendWeeklyMail 105, 366
userInfo_SocialSecurityNumber 105, 366
userInfo_SuperiorUserGUID 105, 366
userInfo_TimezoneGUID 105, 366
userInfo_Title 105, 366
userInfo_WorktypeGUID 105, 366
userKeywords 382
USERLASTNAME 80, 189, 192, 194, 196, 199, 201, 217, 277, 280, 282, 353, 394
UserRightsByUserGUID 376
Users 377
Users by Access Rights Profile GUID 379
Users by GUIDs 381
Users by User Keywords 382
Users Changed since 384
UsersByAccessRightsProfileGUID 379
UsersByGUIDs 381
UsersByUserKeywords 382
UsersChangedSince 384
userTagGUIDs 294
UserTags 386

- V -

value 143
VAT 74, 86, 91, 96, 104, 243, 246, 249, 251, 254, 269, 270, 271, 276, 284, 287, 314, 315, 316, 317, 320, 360, 364
VAT Number 25, 59, 70, 114, 116, 117, 118, 139, 158, 160, 161, 340, 349

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

VATNUMBER 22, 25, 59, 70, 114, 116, 117, 118, 139, 158, 160, 161, 340, 349
Version 386
Virtual Bank Barcode 149, 207, 212, 222, 227, 232, 238, 397
VIRTUALBANKBARCODE 149, 207, 212, 222, 227, 232, 238, 397
Visma Severa 1, 16, 17, 18, 19, 20, 22, 23, 24, 25, 26, 27, 29, 30, 32, 34, 35, 36, 38, 39, 40, 41, 42, 43, 44, 48, 49, 50, 52, 53, 54, 55, 56, 57, 59, 61, 65, 66, 67, 69, 70, 73, 74, 79, 80, 84, 85, 86, 91, 94, 96, 97, 98, 99, 101, 102, 103, 104, 105, 108, 109, 110, 111, 113, 114, 116, 117, 118, 120, 122, 125, 128, 129, 132, 135, 137, 138, 139, 140, 141, 143, 144, 145, 146, 147, 148, 149, 156, 157, 158, 160, 161, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 182, 183, 184, 185, 186, 187, 188, 189, 192, 194, 196, 199, 201, 203, 204, 205, 206, 207, 212, 217, 219, 220, 221, 222, 227, 232, 238, 243, 246, 249, 251, 254, 257, 258, 259, 260, 261, 262, 263, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 280, 282, 284, 287, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 320, 321, 322, 323, 324, 325, 326, 328, 329, 331, 334, 336, 337, 339, 340, 342, 347, 348, 349, 352, 353, 358, 360, 361, 362, 363, 364, 366, 369, 370, 371, 373, 375, 376, 377, 379, 381, 382, 384, 386, 387, 388, 389, 390, 391, 392, 394, 397
VismaSevera 1

- W -

WEBSITE 70, 139, 158, 160, 161, 349
WEIGHT 66, 69, 79, 108, 139, 308, 309, 386
Work Estimate 94, 261, 263, 265, 266, 358
Work Type by GUID 388
Work Type by Hour GUID 389
Work Type Code 80, 189, 192, 194, 196, 199, 201, 217, 277, 280, 282, 353, 394
Work Type GUID 80, 94, 105, 128, 189, 192, 194, 196, 199, 201, 203, 217, 261, 263, 265, 266, 277, 280, 282, 353, 358, 366, 371, 373, 375, 377, 379, 381, 382, 384, 394
Work Types 390
Work Types by Phase GUID 390
Work Types Changed since 391
Workday Information by Date Range and User GUID 387
WorkDayInfoByDateRangeAndUserGUID 387
WORKESTIMATE 94, 261, 263, 265, 266, 358
WorkTypeByGUID 388
WorkTypeByHourGUID 389
WORKTYPECODE 80, 189, 192, 194, 196, 199, 201, 217, 277, 280, 282, 353, 394
WORKTYPEGUID 80, 94, 105, 128, 189, 192, 194, 196, 199, 201, 203, 217, 261, 263, 265, 266, 277, 280, 282, 353, 358, 366, 371, 373, 375, 377, 379, 381, 382, 384, 388, 394
workTypeInfo_Code 109, 370
workTypeInfo_GUID 109, 370
workTypeInfo_IsActive 109, 370
workTypeInfo_IsDefault 109, 370
workTypeInfo_IsProductive 109, 370
workTypeInfo_Name 109, 370
workTypeInfo_SalesAccountGUID 109, 370
WorkTypes 390
WorkTypesByPhaseGUID 390
WorkTypesChangedSince 391

- Y -

YEAR 57, 113, 339
Your Reference 61, 120, 122, 125, 129, 132, 135, 149, 207, 212, 222, 227, 232, 238, 342, 394, 397
YOURREFERENCE 61, 120, 122, 125, 129, 132, 135, 149, 207, 212, 222, 227, 232, 238, 342, 397

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.



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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.
24.0.92
29/06/2026, 19:56
ISBN 978-94-6302-000-8: