

Visma Severa API Data Model

for use with Invantive SQL



Copyright

(C) Copyright 2004-2023 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 and 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	SQL Driver Attributes for Visma Severa API	2
3	Schema: API	6
3.1	Tables	6
3.1.1	AccessRightsProfileByGUID: Visma Severa Access Rights Profile by GUID	6
3.1.2	AccessRightsProfiles: Visma Severa Access Rights Profiles	7
3.1.3	AccountByGUID: Visma Severa Account by GUID	8
3.1.4	AccountByName: Visma Severa Account by Name	9
3.1.5	AccountByNumber: Visma Severa Account by Number	11
3.1.6	AccountByVatNumber: Visma Severa Account by VAT Number	12
3.1.7	AccountGroups: Visma Severa Account Groups	14
3.1.8	AccountGroupsByAccountGUID: Visma Severa Account Group by Account GUID	14
3.1.9	Accounts: Visma Severa Accounts	15
3.1.10	AccountsByCompanyGUID: Visma Severa Accounts by Company GUID	16
3.1.11	AccountsByInsertDate: Visma Severa Accounts by Insert Date	18
3.1.12	AccountsChangedSince: Visma Severa Accounts Changed since	19
3.1.13	AccountsChangedSinceOpts: Visma Severa Accounts Changed since Opts.	21
3.1.14	ActiveLeadSources: Visma Severa Active Lead Sources	22
3.1.15	ActivitiesByDate: Visma Severa Activities by Date	22
3.1.16	ActivityByGUID: Visma Severa Activity by GUID	24
3.1.17	ActivityContactParticipants: Visma Severa Activity Contact Participants	25
3.1.18	ActivityInstances: Visma Severa Activity Instances	26
3.1.19	ActivityParticipants: Visma Severa Activity Participants	28
3.1.20	ActivityResourceParticipants: Visma Severa Activity Resource Participants	29
3.1.21	ActivityTypeByGUID: Visma Severa Activity Type by GUID	30
3.1.22	ActivityTypes: Visma Severa Activity Types	31
3.1.23	ActivityUserParticipants: Visma Severa Activity User Participants	31
3.1.24	AddCaseMemberUser: Visma Severa Add Project Member User	32
3.1.25	AddNew AccessRightsProfile: Visma Severa Add New Access Rights Profile	33
3.1.26	AddNew Account: Visma Severa Add New Account	35
3.1.27	AddNew AccountGroup: Visma Severa Add New Account Group	38
3.1.28	AddNew AccountGroupMembers: Visma Severa Add New Account Group Members	39
3.1.29	AddNew Activity: Visma Severa Add New Activity	40
3.1.30	AddNew ActivityContactParticipant: Visma Severa Add New Activity Contact Participant	42
3.1.31	AddNew ActivityResourceParticipant: Visma Severa Add New Activity Resource Participant	43
3.1.32	AddNew ActivityType: Visma Severa Add New Activity Type	44
3.1.33	AddNew ActivityUserParticipant: Visma Severa Add New Activity User Participant	45
3.1.34	AddNew Address: Visma Severa Add New Address	46
3.1.35	AddNew BillingAndExpenseForecastToCase: Visma Severa Add New Billing and Expense Forecast to Project	
3.1.36	AddNew BusinessUnit: Visma Severa Add New Business Unit	49
3.1.37	AddNew Case: Visma Severa Add New Project	51
3.1.38	AddNew CaseStatus: Visma Severa Add New Project Status	56
3.1.39	AddNew CaseTag: Visma Severa Add New Project Tag	57
3.1.40	AddNew Contact: Visma Severa Add New Contact	58
3.1.41	AddNew ContactTag: Visma Severa Add New Contact Tag	59
3.1.42	AddNew Customer: Visma Severa Add New Customer	60
3.1.43	AddNew Employment: Visma Severa Add New Employment	63
3.1.44	AddNew ExpenseToProductByPoNumber: Visma Severa Add New Expense to Product by PO Number ..	64
3.1.45	AddNew FileTag: Visma Severa Add New File Tag	69
3.1.46	AddNew HourEntry: Visma Severa Add New Hour Entry	70

3.1.47	AddNew Industry: Visma Severa Add New Industry	74
3.1.48	AddNew InvoiceLogEntry: Visma Severa Add New Invoice Log Entry	75
3.1.49	AddNew Item: Visma Severa Add New Item	76
3.1.50	AddNew Items: Visma Severa Add New Items	81
3.1.51	AddNew Phase: Visma Severa Add New Phase	84
3.1.52	AddNew Product: Visma Severa Add New Product	86
3.1.53	AddNew ProductCategory: Visma Severa Add New Product Category	87
3.1.54	AddNew Resource: Visma Severa Add New Resource	88
3.1.55	AddNew ResourceAllocation: Visma Severa Add New Resource Allocation	89
3.1.56	AddNew ResourceAllocations: Visma Severa Add New Resource Allocations	91
3.1.57	AddNew SalesProcess: Visma Severa Add New Sales Process	92
3.1.58	AddNew SalesStatus: Visma Severa Add New Sales Status	93
3.1.59	AddNew TravelExpense: Visma Severa Add New Travel Expense	94
3.1.60	AddNew User: Visma Severa Add New User	95
3.1.61	AddNew UserTag: Visma Severa Add New User Tag	98
3.1.62	AddNew WorkType: Visma Severa Add New Work Type	99
3.1.63	AddressByGUID: Visma Severa Address by GUID	100
3.1.64	AddressesChangedSince: Visma Severa Addresses Changed since	101
3.1.65	BillingAddress: Visma Severa Billing Address	103
3.1.66	BillingAndExpenseForecastByCaseGUID: Visma Severa Billing and Expense Forecast by Project GUID	103
3.1.67	BusinessUnitByCode: Visma Severa Business Unit by Code	104
3.1.68	BusinessUnitByGUID: Visma Severa Business Unit by GUID	106
3.1.69	BusinessUnitHierarchy: Visma Severa Business Unit Hierarchy	107
3.1.70	BusinessUnitsChangedSince: Visma Severa Business Units Changed since	108
3.1.71	CaseByGUID: Visma Severa Project by GUID	110
3.1.72	CaseByNumber: Visma Severa Project by Number	112
3.1.73	CaseByTaskGUID: Visma Severa Project by Task GUID	115
3.1.74	CaseMemberUsersByCaseGUID: Visma Severa Project Member Users by Project GUID	118
3.1.75	Cases: Visma Severa Projects	119
3.1.76	CasesByAccountGUID: Visma Severa Projects by Account GUID	122
3.1.77	CasesChangedSince: Visma Severa Projects Changed since	125
3.1.78	CaseStatusByGUID: Visma Severa Project Status by GUID	127
3.1.79	CaseStatuses: Visma Severa Project Statuses	128
3.1.80	CaseTags: Visma Severa Project Tags	129
3.1.81	CompanyByGUID: Visma Severa Company by GUID	129
3.1.82	ContactByGUID: Visma Severa Contact by GUID	130
3.1.83	ContactsByAccountGUID: Visma Severa Contacts by Account GUID	131
3.1.84	ContactsByCommunicationMethodType: Visma Severa Contacts by Communication Method Type	133
3.1.85	ContactsChangedSince: Visma Severa Contacts Changed since	134
3.1.86	CostCenterByGUID: Visma Severa Cost Center by GUID	135
3.1.87	CostCenters: Visma Severa Cost Centers	136
3.1.88	CountryByGUID: Visma Severa Country by GUID	136
3.1.89	CountryByISO: Visma Severa Country by ISO	137
3.1.90	CountryByName: Visma Severa Country by Name	138
3.1.91	CreateInvoices: Visma Severa Create Invoices	139
3.1.92	Currencies: Visma Severa Currencies	146
3.1.93	CurrencyByGUID: Visma Severa Currency by GUID	146
3.1.94	CurrencyByIsoCode: Visma Severa Currency by ISO Code	147
3.1.95	CustomerByGUID: Visma Severa Customer by GUID	148
3.1.96	Customers: Visma Severa Customers	149
3.1.97	CustomersChangedSince: Visma Severa Customers Changed since	151
3.1.98	Database: Visma Severa Database	153
3.1.99	DeleteAccessRightsProfileByGUID: Visma Severa Delete Access Rights Profile by GUID	153
3.1.100	DeleteActivity: Visma Severa Delete Activity	154
3.1.101	DeleteActivityType: Visma Severa Delete Activity Type	155
3.1.102	DeleteCase: Visma Severa Delete Project	156
3.1.103	DeleteCaseStatus: Visma Severa Delete Project Status	156
3.1.104	DeleteContact: Visma Severa Delete Contact	157
3.1.105	DeleteEmployment: Visma Severa Delete Employment	158

3.1.106	DeleteHourEntry: Visma Severa Delete Hour Entry	159
3.1.107	DeleteItem: Visma Severa Delete Item	160
3.1.108	DeletePhase: Visma Severa Delete Phase	160
3.1.109	DeleteProduct: Visma Severa Delete Product	161
3.1.110	DeleteProductCategory: Visma Severa Delete Product Category	162
3.1.111	DeleteResource: Visma Severa Delete Resource	163
3.1.112	DeleteResourceAllocation: Visma Severa Delete Resource Allocation	164
3.1.113	DeleteTag: Visma Severa Delete Tag	165
3.1.114	DeleteTravelExpense: Visma Severa Delete Travel Expense	165
3.1.115	DisableUser: Visma Severa Disable User	166
3.1.116	EmploymentByGUID: Visma Severa Employment by GUID	167
3.1.117	EmploymentsByUserGUID: Visma Severa Employments by User GUID	168
3.1.118	EnableUser: Visma Severa Enable User	169
3.1.119	EntitySchemaVersion: Visma Severa Entity Schema Version	170
3.1.120	ExtranetCaseByCaseGUID: Visma Severa Extranet Project by Project GUID	170
3.1.121	ExtranetContactCaseRightsByContactGUID: Visma Severa Extranet Contact Project Rights by Contact GUID	170
3.1.122	FileByGUID: Visma Severa File by GUID	173
3.1.123	FileBytesByGUID: Visma Severa File Contents by GUID	173
3.1.124	Files: Visma Severa Files	174
3.1.125	FinvoicesByStatus: Visma Severa Financial Invoices by Status	175
3.1.126	FinvoicesByStatusAndDate: Visma Severa Financial Invoices by Status and Date	176
3.1.127	FirstAccountsBillingAddress: Visma Severa First Account Billing Addresses	177
3.1.128	HeadOfficeAddressByAccountGUID: Visma Severa Head Office Address by Account GUID	178
3.1.129	HourEntriesByCaseGUID: Visma Severa Hour Entries by Project GUID	179
3.1.130	HourEntriesByDate: Visma Severa Hour Entries by Date	181
3.1.131	HourEntriesByDateAndUser: Visma Severa Hour Entries by Date and User	184
3.1.132	HourEntriesByDateAndUserGUID: Visma Severa Hour Entry by Date and User GUID	186
3.1.133	HourEntriesChangedSince: Visma Severa Hour Entries Changed since	189
3.1.134	HourEntryByGUID: Visma Severa Hour Entry by GUID	191
3.1.135	HourEntryPhasesByUserGUID: Visma Severa Hour Entry Phases by User GUID	193
3.1.136	Industries: Visma Severa Industries	194
3.1.137	InvoiceAttachmentFile: Visma Severa Invoice Attachment File	195
3.1.138	InvoiceAttachmentFileGUIDs: Visma Severa Invoice Attachment File GUIDs	195
3.1.139	InvoiceByGUID: Visma Severa Invoice by GUID	196
3.1.140	InvoiceByNumber: Visma Severa Invoice by Number	202
3.1.141	InvoicedHourEntriesByCaseGUID: Visma Severa Invoiced Hour Entries by Project GUID	207
3.1.142	InvoiceOriginDetailsByInvoiceGUID: Visma Severa Invoice Origin Details by Invoice GUID	209
3.1.143	InvoicePdf: Visma Severa Invoice PDF	210
3.1.144	InvoicePdfByInvoiceGUID: Visma Severa Invoice PDF by Invoice GUID	211
3.1.145	InvoicesByDate: Visma Severa Invoices by Date	212
3.1.146	InvoicesByDateAndStatus: Visma Severa Invoices by Date and Status	217
3.1.147	InvoicesByStatus: Visma Severa Invoices by Status	222
3.1.148	InvoicesByStatusAndBusinessUnit: Visma Severa Invoices by Status and Business Unit	228
3.1.149	InvoiceStatuses: Visma Severa Invoice Statuses	233
3.1.150	ItemByGUID: Visma Severa Item by GUID	233
3.1.151	ItemByPurchaseOrderNumber: Visma Severa Item by Purchase Order Number	236
3.1.152	ItemsByCaseGUID: Visma Severa Items by Project GUID	239
3.1.153	ItemsByInsertDate: Visma Severa Items by Insert Date	241
3.1.154	ItemsChangedSince: Visma Severa Items Changed since	244
3.1.155	LanguageByGUID: Visma Severa Language by GUID	247
3.1.156	LanguageByIetfTag: Visma Severa Language by IETF Tag	248
3.1.157	LeadSourceByGUID: Visma Severa Lead Source by GUID	248
3.1.158	LeadSources: Visma Severa Lead Sources	249
3.1.159	OverTimeByGUID: Visma Severa Overtime by GUID	250
3.1.160	Overtimes: Visma Severa Overtimes	251
3.1.161	PhaseByGUID: Visma Severa Phase by GUID	251
3.1.162	PhaseMembersByPhaseGUID: Visma Severa Phase Members by Phase GUID	252
3.1.163	PhasesByCaseGUID: Visma Severa Phases by Project GUID	253
3.1.164	PhasesByCode: Visma Severa Phases by Code	255

3.1.165	PhasesChangedSince: Visma Severa Phases Changed since	256
3.1.166	PricelistByGUID: Visma Severa Price List by GUID	257
3.1.167	Pricelists: Visma Severa Price Lists	258
3.1.168	ProductByCode: Visma Severa Product by Code	259
3.1.169	ProductByCodeIncludeInactive: Visma Severa Product by Code Include Inactive	260
3.1.170	ProductByGUID: Visma Severa Product by GUID	261
3.1.171	ProductCategoriesChangedSince: Visma Severa Product Categories Changed since	262
3.1.172	ProductCategoryByCode: Visma Severa Product Category by Code	263
3.1.173	ProductCategoryByGUID: Visma Severa Product Category by GUID	264
3.1.174	ProductCostAccountAssociationsByGUID: Visma Severa Product Cost Account Associations by GUID	265
3.1.175	ProductsChangedSince: Visma Severa Products Changed since	266
3.1.176	ReimbursedHourEntriesByDate: Visma Severa Reimbursed Hour Entries by Date	267
3.1.177	ReimbursedHourEntriesChangedSince: Visma Severa Reimbursed Hour Entries Changed since	270
3.1.178	ReimbursedHourEntryByGUID: Visma Severa Reimbursed Hour Entry by GUID	272
3.1.179	ReimbursedItemByGUID: Visma Severa Reimbursed Item by GUID	274
3.1.180	ReimbursedItemsChangedSince: Visma Severa Reimbursed Items Changed since	277
3.1.181	RemoveAccountGroupMembers: Visma Severa Remove Account Group Members	280
3.1.182	RemoveActivityContactParticipant: Visma Severa Remove Activity Contact Participant	280
3.1.183	RemoveActivityResourceParticipant: Visma Severa Remove Activity Resource Participant	281
3.1.184	RemoveActivityUserParticipant: Visma Severa Remove Activity User Participant	282
3.1.185	ResourceAllocationByGUID: Visma Severa Resource Allocations by GUID	283
3.1.186	ResourceAllocations: Visma Severa Resource Allocations	284
3.1.187	ResourceAllocationsByCaseGUID: Visma Severa Resource Allocations by Project GUID	285
3.1.188	ResourceAllocationsByCaseGUIDUserGUID: Visma Severa Resource Allocations by Project GUID and User GUID	
3.1.189	ResourceAllocationsByUserGUID: Visma Severa Resource Allocations by User GUID	287
3.1.190	ResourceAllocationsChangedSince: Visma Severa Resource Allocations Changed since	288
3.1.191	ResourceByGUID: Visma Severa Resource by GUID	289
3.1.192	Resources: Visma Severa Resources	290
3.1.193	SalesProcessByGUID: Visma Severa Sales Process by GUID	290
3.1.194	SalesProcesses: Visma Severa Sales Processes	291
3.1.195	SalesStatusByGUID: Visma Severa Sales Status by GUID	292
3.1.196	SalesStatusesForCase: Visma Severa Sales Statuses for Project	293
3.1.197	Server: Visma Severa Server	294
3.1.198	SetInvoiceNumber: Visma Severa Set Invoice Number	294
3.1.199	SetInvoicePaymentDate: Visma Severa Set Invoice Payment Date	295
3.1.200	SetInvoiceStatus: Visma Severa Set Invoice Status	296
3.1.201	SetPhaseMemberListForPhase: Visma Severa Set Phase Member List for Phase	296
3.1.202	SetTravelReimbursementStatus: Visma Severa Set Travel Reimbursement Status	297
3.1.203	TagByGUID: Visma Severa Tag by GUID	298
3.1.204	TagsByCaseGUID: Visma Severa Tags by Project GUID	299
3.1.205	TimeEntriesByDate: Visma Severa Time Entries by Date	300
3.1.206	TimeEntriesByDateAndUserGUID: Visma Severa Time Entries by Date and User GUID	301
3.1.207	TimeEntriesChangedSince: Visma Severa Time Entries Changed since	302
3.1.208	TimezoneByGUID: Visma Severa Timezone by GUID	303
3.1.209	Timezones: Visma Severa Timezones	304
3.1.210	TravelExpenseByCode: Visma Severa Travel Expense by Code	304
3.1.211	TravelExpenseByCodeIncludeInactive: Visma Severa Travel Expense by Code Include Inactive	305
3.1.212	TravelExpenseByGUID: Visma Severa Travel Expense by GUID	306
3.1.213	TravelExpensesByDate: Visma Severa Travel Expenses by Date	307
3.1.214	TravelExpensesChangedSince: Visma Severa Travel Expenses Changed since	310
3.1.215	TravelReimbursementAttachments: Visma Severa Travel Reimbursement Attachments	311
3.1.216	TravelReimbursementByGUID: Visma Severa Travel Reimbursement by GUID	312
3.1.217	TravelReimbursementByNumber: Visma Severa Travel Reimbursement by Number	313
3.1.218	TravelReimbursementPdf: Visma Severa Travel Reimbursement PDF	314
3.1.219	TravelReimbursementsByDate: Visma Severa Travel Reimbursement by Date	315
3.1.220	TravelReimbursementsByStatus: Visma Severa Travel Reimbursement by Status	316
3.1.221	TravelReimbursementsByStatusGUID: Visma Severa Travel Reimbursement by Status GUID	318
3.1.222	TravelReimbursementStatuses: Visma Severa Travel Reimbursement Statuses	319
3.1.223	UpdateAccessRightsProfile: Visma Severa Update Access Rights Profile	319

3.1.224	UpdateAccount: Visma Severa Update Account	321
3.1.225	UpdateActivity: Visma Severa Update Activity	324
3.1.226	UpdateActivityType: Visma Severa Update Activity Type	326
3.1.227	UpdateAddress: Visma Severa Update Address	327
3.1.228	UpdateBillingAndExpenseForecast: Visma Severa Update Billing and Expense Forecast	328
3.1.229	UpdateBusinessUnit: Visma Severa Update Business Unit	330
3.1.230	UpdateCase: Visma Severa Update Project	332
3.1.231	UpdateCaseStatus: Visma Severa Update Project Status	337
3.1.232	UpdateContact: Visma Severa Update Contact	338
3.1.233	UpdateCustomer: Visma Severa Update Customer	339
3.1.234	UpdateEmployment: Visma Severa Update Employment	342
3.1.235	UpdateHourEntry: Visma Severa Update Hour Entry	343
3.1.236	UpdatePhase: Visma Severa Update Phase	348
3.1.237	UpdateProduct: Visma Severa Update Product	350
3.1.238	UpdateProductCategory: Visma Severa Update Product Category	351
3.1.239	UpdateResource: Visma Severa Update Resource	352
3.1.240	UpdateResourceAllocation: Visma Severa Update Resource Allocation	353
3.1.241	UpdateTravelExpense: Visma Severa Update Travel Expense	354
3.1.242	UpdateUser: Visma Severa Update User	356
3.1.243	UpdateUserRightsByUserGUID: Visma Severa Update User Rights by User GUID	359
3.1.244	UpdateWorkType: Visma Severa Update Work Type	360
3.1.245	UserByCode: Visma Severa User by Code	361
3.1.246	UserByGUID: Visma Severa User by GUID	363
3.1.247	UserByName: Visma Severa User by Name	365
3.1.248	UserRightsByUserGUID: Visma Severa User Rights by User GUID	366
3.1.249	Users: Visma Severa Users	367
3.1.250	UsersByAccessRightsProfileGUID: Visma Severa Users by Access Rights Profile GUID	369
3.1.251	UsersByGUIDs: Visma Severa Users by GUIDs	371
3.1.252	UsersByUserKeywords: Visma Severa Users by User Keywords	372
3.1.253	UsersChangedSince: Visma Severa Users Changed since	374
3.1.254	UserTags: Visma Severa User Tags	376
3.1.255	Version: Visma Severa Version	376
3.1.256	WorkDayInfoByDateRangeAndUserGUID: Visma Severa Workday Information by Date Range and User GUID	
3.1.257	WorkTypeByGUID: Visma Severa Work Type by GUID	378
3.1.258	WorkTypeByHourGUID: Visma Severa Work Type by Hour GUID	379
3.1.259	WorkTypes: Visma Severa Work Types	380
3.1.260	WorkTypesByPhaseGUID: Visma Severa Work Types by Phase GUID	380
3.1.261	WorkTypesChangedSince: Visma Severa Work Types Changed since	381
4	Schema: Native	382
4.1	Tables	382
4.1.1	NATIVEPLATFORMSCALARREQUESTS: Visma Severa Native Platform Scalar Requests	382
5	Schema: Views	384
5.1	Views	384
5.1.1	HourEntries2Years: Visma Severa Hour Entries Last 2 Years	384
5.1.2	Invoices: Visma Severa Invoices	387
	Index	393

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 5299 columns.

Visma Severa API Clients

Invantive UniversalSQL is available on many user interfaces ("clients" in traditional server-client paradigm). 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.

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](#).

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`

Driver code for use in settings.xml

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

Updated 10-06-2024 19:00 using Invantive UniversalSQL version 24.1.3-BETA+4689.

2 SQL Driver Attributes for Visma Severa 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.

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 Driver's 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-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 occurs during retrieval of data.		10000	✓	✓	✓	
download-error-other-exception-sleep-max-ms	Maximum sleep in milliseconds between retries when an unqualified error occurs during retrieval of data.		300000	✓	✓	✓	
download-error-other-exception-sleep-multiplicator	Multiplication factor for sleep between retries when an unqualified error occurs during retrieval of data.		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.	Shared	5	✓	✓	✓	
http-disk-cache-directory	Directory where HTTP cache is stored.	Shared	C:\Users\guido\Inventive\Cache\http\guido\shared	✓	✓	✓	
http-disk-cache-ignore-write-errors	Whether to ignore write errors to disk cache.	Shared	False	✓	✓	✓	
http-disk-cache-max-age-sec	Maximum acceptable age in seconds for use of data in the HTTP disk cache.	Shared	2592000	✓	✓	✓	

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Driver's File	Set from Log On
http-get-timeout-max-ms	HTTP GET timeout maximum 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 timeout maximum on retry (ms)		58000	✓	✓	✓	
http-post-timeout-ms	HTTP POST timeout (ms)		57000	✓	✓	✓	
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	itgen_pae_invantive_sql_execution_profile_disk_path	SQL Engine V1	c:\temp\profiles	✓	✓	✓	
invantive-sql-execution-profile-to-disk	itgen_pae_invantive_sql_execution_profile_to_disk	SQL Engine V1	True	✓	✓	✓	
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	✓	✓	✓	

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
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	✓	✓	✓	
max-url-length-accepted	The maximum accepted URL length before raising an error.	Shared	8000	✓	✓	✓	
max-url-length-desired	The maximum desired URL length.	Shared	8000	✓	✓	✓	
maximum-length-identifiers	Non-default maximum length in characters of identifier names.	Shared		✓	✓	✓	
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		✓			✓
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-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			✓		
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-casing	Rewrite all identifiers to the recommended standard platform-specific casing when changing a data model on a case-dependent platform.	Shared	True	✓	✓	✓	
standardize-identifiers	Rewrite all identifiers to the preferred standards as configured by standardize-identifiers-casing and maximum-length-identifiers.	Shared	True	✓	✓	✓	
use-http-disk-cache-read	Whether to use HTTP responses from previous queries stored on disk to answer the current query.	Shared	False	✓	✓	✓	

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Driver's File	Set from Log On
use-http-disk-cache-write	Whether to memorize HTTP responses on disk.	Shared	False	✓	✓	✓	
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	✓	✓	✓	

3 Schema: API

3.1 Tables

3.1.1 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	string	<input type="checkbox"/>		

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	string	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
NAME	string	Name		
RIGHTS_ACCOUNT	string			
RIGHTS_ACCOUNTDELETE	boolean			
RIGHTS_ADMINISTRATOR	string			
RIGHTS_CASE	string			
RIGHTS_CASEDELETE	boolean			
RIGHTS_CASEOWNERWORKHOUR APPROVAL	boolean			
RIGHTS_SCHEDULEJOBS	string			
RIGHTS_SHARING	boolean			
RIGHTS_TRAVELREIMBURSEMENT	string			
RIGHTS_USERS	string			
RIGHTS_WORKHOURAPPROVAL	string			

3.1.2 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 Invariant UniversalSQL driver 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	string	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		

Name	Data Type	Label	Required	Documentation
NAME	string	Name		
RIGHTS_ACCOUNT	string			
RIGHTS_ACCOUNTDELETE	boolean			
RIGHTS_ADMINISTRATOR	string			
RIGHTS_CASE	string			
RIGHTS_CASEDELETE	boolean			
RIGHTS_CASEOWNERWORKHOUR APPROVAL	boolean			
RIGHTS_SCHEDULEJOBS	string			
RIGHTS_SHARING	boolean			
RIGHTS_TRAVELREIMBURSEMENT	string			
RIGHTS_USERS	string			
RIGHTS_WORKHOURAPPROVAL	string			

3.1.3 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	string	<input type="checkbox"/>		

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	string			
ACCOUNTOWNERUSERGUID	string	Account Owner User GUID		
ACCOUNTRATING	int32	Account Rating		
COMPANYGUID	string	Company GUID		
CURRENCYCODE	string	Currency Code		
CURRENCYGUID	string	Currency GUID		
EINVOICEADDRESS	string	e-invoice Address		
EINVOICEOPERATOR	string	e-invoice Operator		
GUID	string	GUID		
INSERTTS	datetime	Inserted		
INVOICINGVAT	decimal	Invoicing VAT		
ISACTIVE	boolean	Is Active		
ISINTERNAL	boolean	Is Internal		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
NAME	string	Name		
NOTES	string	Notes		
NUMBER	int64	Number		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTTERM	int32	Payment Term		
PRICELISTGUID	string	Pricelist GUID		
REVERSECHARGE	boolean	Reverse Charge		
REVERSECHARGEDESCRIPTION	string	Reverse Charge Description		
UPDATEDTS	datetime	Updated		
URLTOACCOUNT	string	URL to Account		

3.1.4 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.

Name	Data Type	Required	Default Value	Documentation
accountName	string	<input type="checkbox"/>		

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	string			
ACCOUNTOWNERUSERGUID	string	Account Owner User GUID		
ACCOUNTRATING	int32	Account Rating		
COMPANYGUID	string	Company GUID		
CURRENCYCODE	string	Currency Code		
CURRENCYGUID	string	Currency GUID		
EINVOICEADDRESS	string	e-invoice Address		
EINVOICEOPERATOR	string	e-invoice Operator		
GUID	string	GUID		
INSERTTS	datetime	Inserted		
INVOICINGVAT	decimal	Invoicing VAT		
ISACTIVE	boolean	Is Active		
ISINTERNAL	boolean	Is Internal		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
NAME	string	Name		
NOTES	string	Notes		
NUMBER	int64	Number		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTTERM	int32	Payment Term		

Name	Data Type	Label	Required	Documentation
PRICELISTGUID	string	Pricelist GUID		
REVERSECHARGE	boolean	Reverse Charge		
REVERSECHARGEDESCRIPTION	string	Reverse Charge Description		
UPDATEDTS	datetime	Updated		
URLTOACCOUNT	string	URL to Account		

3.1.5 AccountByNumber: Visma Severa Account by Number

Catalog: Severa

Schema: API

Label: Account 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: 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	int64	<input type="checkbox"/>		

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	string			
ACCOUNTOWNERUSERGUID	string	Account Owner User GUID		

Name	Data Type	Label	Required	Documentation
ACCOUNTRATING	int32	Account Rating		
COMPANYGUID	string	Company GUID		
CURRENCYCODE	string	Currency Code		
CURRENCYGUID	string	Currency GUID		
EINVOICEADDRESS	string	e-invoice Address		
EINVOICEOPERATOR	string	e-invoice Operator		
GUID	string	GUID		
INSERTTS	datetime	Inserted		
INVOICINGVAT	decimal	Invoicing VAT		
ISACTIVE	boolean	Is Active		
ISINTERNAL	boolean	Is Internal		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
NAME	string	Name		
NOTES	string	Notes		
NUMBER	int64	Number		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTTERM	int32	Payment Term		
PRICELISTGUID	string	Pricelist GUID		
REVERSECHARGE	boolean	Reverse Charge		
REVERSECHARGEDESCRIPTION	string	Reverse Charge Description		
UPDATEDTS	datetime	Updated		
URLTOACCOUNT	string	URL to Account		

3.1.6 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	string	<input type="checkbox"/>		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGROUPGUIDS	string			
ACCOUNTOWNERUSERGUID	string	Account Owner User GUID		
ACCOUNTRATING	int32	Account Rating		
COMPANYGUID	string	Company GUID		
CURRENCYCODE	string	Currency Code		
CURRENCYGUID	string	Currency GUID		
EINVOICEADDRESS	string	e-invoice Address		
EINVOICEOPERATOR	string	e-invoice Operator		
GUID	string	GUID		
INSERTTS	datetime	Inserted		
INVOICINGVAT	decimal	Invoicing VAT		
ISACTIVE	boolean	Is Active		
ISINTERNAL	boolean	Is Internal		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
NAME	string	Name		
NOTES	string	Notes		
NUMBER	int64	Number		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTTERM	int32	Payment Term		
PRICELISTGUID	string	Pricelist GUID		
REVERSECHARGE	boolean	Reverse Charge		
REVERSECHARGEDESCRIPTION	string	Reverse Charge Description		
UPDATEDTS	datetime	Updated		
URLTOACCOUNT	string	URL to Account		

3.1.7 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

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	int32	<input type="checkbox"/>		

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	string	GUID		
ISACTIVE	boolean	Is Active		
NAME	string	Name		
PARENTGUID	string	Parent GUID		

3.1.8 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 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	string	<input type="checkbox"/>		
options	int32	<input type="checkbox"/>		

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	string	GUID		
ISACTIVE	boolean	Is Active		
NAME	string	Name		
PARENTGUID	string	Parent GUID		

3.1.9 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 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	int32	<input type="checkbox"/>		
options	string	<input type="checkbox"/>		
startFromIndex	int32	<input type="checkbox"/>		

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	string	GUID		
NAME	string	Name		
NUMBER	int64	Number		
VATNUMBER	string	VAT Number		

3.1.10 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	string	<input type="checkbox"/>		

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	string			
ACCOUNTOWNERUSERGUID	string	Account Owner User GUID		
ACCOUNTRATING	int32	Account Rating		
COMPANYGUID	string	Company GUID		
CURRENCYCODE	string	Currency Code		
CURRENCYGUID	string	Currency GUID		
EINVOICEADDRESS	string	e-invoice Address		
EINVOICEOPERATOR	string	e-invoice Operator		
GUID	string	GUID		
INSERTTS	datetime	Inserted		
INVOICINGVAT	decimal	Invoicing VAT		
ISACTIVE	boolean	Is Active		
ISINTERNAL	boolean	Is Internal		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
NAME	string	Name		
NOTES	string	Notes		
NUMBER	int64	Number		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTTERM	int32	Payment Term		
PRICELISTGUID	string	Pricelist GUID		
REVERSECHARGE	boolean	Reverse Charge		

Name	Data Type	Label	Required	Documentation
REVERSECHARGEDESCRIPTION	string	Reverse Charge Description		
UPDATEDTS	datetime	Updated		
URLTOACCOUNT	string	URL to Account		

3.1.11 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.

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	datetime	<input type="checkbox"/>		
insertStartDate	datetime	<input type="checkbox"/>		

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	string			
ACCOUNTOWNERUSERGUID	string	Account Owner User GUID		
ACCOUNTRATING	int32	Account Rating		

Name	Data Type	Label	Required	Documentation
COMPANYGUID	string	Company GUID		
CURRENCYCODE	string	Currency Code		
CURRENCYGUID	string	Currency GUID		
EINVOICEADDRESS	string	e-invoice Address		
EINVOICEOPERATOR	string	e-invoice Operator		
GUID	string	GUID		
INSERTTS	datetime	Inserted		
INVOICINGVAT	decimal	Invoicing VAT		
ISACTIVE	boolean	Is Active		
ISINTERNAL	boolean	Is Internal		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
NAME	string	Name		
NOTES	string	Notes		
NUMBER	int64	Number		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTTERM	int32	Payment Term		
PRICELISTGUID	string	Pricelist GUID		
REVERSECHARGE	boolean	Reverse Charge		
REVERSECHARGEDESCRIPTION	string	Reverse Charge Description		
UPDATEDTS	datetime	Updated		
URLTOACCOUNT	string	URL to Account		

3.1.12 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	int32	<input type="checkbox"/>		
startDateTime	datetime	<input type="checkbox"/>		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGROUPGUIDS	string			
ACCOUNTOWNERUSERGUID	string	Account Owner User GUID		
ACCOUNTRATING	int32	Account Rating		
COMPANYGUID	string	Company GUID		
CURRENCYCODE	string	Currency Code		
CURRENCYGUID	string	Currency GUID		
EINVOICEADDRESS	string	e-invoice Address		
EINVOICEOPERATOR	string	e-invoice Operator		
GUID	string	GUID		
INSERTTS	datetime	Inserted		
INVOICINGVAT	decimal	Invoicing VAT		
ISACTIVE	boolean	Is Active		
ISINTERNAL	boolean	Is Internal		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
NAME	string	Name		
NOTES	string	Notes		
NUMBER	int64	Number		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTTERM	int32	Payment Term		
PRICELISTGUID	string	Pricelist GUID		
REVERSECHARGE	boolean	Reverse Charge		
REVERSECHARGEDESCRIPTION	string	Reverse Charge Description		
UPDATEDTS	datetime	Updated		
URLTOACCOUNT	string	URL to Account		

3.1.13 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

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	string	<input type="checkbox"/>		
startDateTime	datetime	<input type="checkbox"/>		

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	string			
ACCOUNTOWNERUSERGUID	string	Account Owner User GUID		
ACCOUNTRATING	int32	Account Rating		
COMPANYGUID	string	Company GUID		
CURRENCYCODE	string	Currency Code		
CURRENCYGUID	string	Currency GUID		
EINVOICEADDRESS	string	e-invoice Address		
EINVOICEOPERATOR	string	e-invoice Operator		

Name	Data Type	Label	Required	Documentation
GUID	string	GUID		
INSERTTS	datetime	Inserted		
INVOICINGVAT	decimal	Invoicing VAT		
ISACTIVE	boolean	Is Active		
ISINTERNAL	boolean	Is Internal		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
NAME	string	Name		
NOTES	string	Notes		
NUMBER	int64	Number		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTTERM	int32	Payment Term		
PRICELISTGUID	string	Pricelist GUID		
REVERSECHARGE	boolean	Reverse Charge		
REVERSECHARGEDESCRIPTION	string	Reverse Charge Description		
UPDATEDTS	datetime	Updated		
URLTOACCOUNT	string	URL to Account		

3.1.14 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	string	GUID		
ISACTIVE	boolean	Is Active		
NAME	string	Name		

3.1.15 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.

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	string	<input type="checkbox"/>		
businessUnitGuid	string	<input type="checkbox"/>		
endDate	datetime	<input type="checkbox"/>		
startDate	datetime	<input type="checkbox"/>		

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	string	Account GUID		
ACTIVITYTYPECODE	string	Activity Type Code		
ACTIVITYTYPEGUID	string	Activity Type GUID		
CASEGUID	string	Project GUID		
ENDDATETIME	datetime	End Date Time		
GUID	string	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	string	Location		

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

3.1.16 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	string	<input type="checkbox"/>		

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	string	Account GUID		
ACTIVITYTYPECODE	string	Activity Type Code		
ACTIVITYTYPEGUID	string	Activity Type GUID		
CASEGUID	string	Project GUID		
ENDDATETIME	datetime	End Date Time		
GUID	string	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	string	Location		
NAME	string	Name		
NOTES	string	Notes		
OWNERUSERBUSINESSUNITCODE	string	Owner User Business Unit Code		
OWNERUSERCODE	string	Owner User Code		
OWNERUSERFIRSTNAME	string	Owner User First Name		
OWNERUSERGUID	string	Owner User GUID		
OWNERUSERLASTNAME	string	Owner User Last Name		
STARTDATETIME	datetime	Start Date Time		
TASKGUID	string	Task GUID		

3.1.17 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.

Name	Data Type	Required	Default Value	Documentation
activityGUID	string	<input type="checkbox"/>		

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	string	Contact GUID		
NAME	string	Name		
RESOURCEGUID	string	Resource GUID		
STATUS	int16	Status		
USERGUID	string	User GUID		

3.1.18 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	string	<input type="checkbox"/>		
activityTypeGUID	string	<input type="checkbox"/>		
caseGUID	string	<input type="checkbox"/>		
endsAfter	datetime	<input type="checkbox"/>		
endsBefore	datetime	<input type="checkbox"/>		
firstRow	int32	<input type="checkbox"/>		
maxRow s	int32	<input type="checkbox"/>		
startsAfter	datetime	<input type="checkbox"/>		
startsBefore	datetime	<input type="checkbox"/>		
userGUID	string	<input type="checkbox"/>		

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	string	Account GUID		
ACTIVITYTYPECODE	string	Activity Type Code		
ACTIVITYTYPEGUID	string	Activity Type GUID		
CASEGUID	string	Project GUID		
ENDDATETIME	datetime	End Date Time		
GUID	string	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	string	Location		
NAME	string	Name		
NOTES	string	Notes		
OWNERUSERBUSINESSUNITCODE	string	Owner User Business Unit Code		
OWNERUSERCODE	string	Owner User Code		

Name	Data Type	Label	Required	Documentation
OWNERUSERFIRSTNAME	string	Owner User First Name		
OWNERUSERGUID	string	Owner User GUID		
OWNERUSERLASTNAME	string	Owner User Last Name		
STARTDATETIME	datetime	Start Date Time		
TASKGUID	string	Task GUID		

3.1.19 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	string	<input type="checkbox"/>		

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	string	Contact GUID		
NAME	string	Name		

Name	Data Type	Label	Required	Documentation
RESOURCEGUID	string	Resource GUID		
STATUS	int16	Status		
USERGUID	string	User GUID		

3.1.20 ActivityResourceParticipants: Visma Severa Activity Resource Participants

Catalog: Severa

Schema: API

Label: Activity Resource 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 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	string	<input type="checkbox"/>		

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	string	Contact GUID		
NAME	string	Name		
RESOURCEGUID	string	Resource GUID		
STATUS	int16	Status		
USERGUID	string	User GUID		

3.1.21 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

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	string	<input type="checkbox"/>		

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	string	Category		
CODE	string	Code		
GUID	string	GUID		
ICON	string	Icon		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
ISPAIDLEAVE	boolean	Is Paid Leave		
NAME	string	Name		

3.1.22 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

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

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

3.1.23 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	string	<input type="checkbox"/>		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CONTACTGUID	string	Contact GUID		
NAME	string	Name		
RESOURCEGUID	string	Resource GUID		
STATUS	int16	Status		
USERGUID	string	User GUID		

3.1.24 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	string	<input type="checkbox"/>		

Name	Data Type	Required	Default Value	Documentation
userGuid	string	<input type="checkbox"/>		

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	string	Dummy	<input checked="" type="checkbox"/>	Dummy column

3.1.25 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	string	<input type="checkbox"/>		

Name	Data Type	Required	Default Value	Documentation
accessRightsProfile_IsActive	boolean	<input type="checkbox"/>		
accessRightsProfile_IsDefault	boolean	<input type="checkbox"/>		
accessRightsProfile_Name	string	<input type="checkbox"/>		
accessRightsProfile_Rights_Account	string	<input type="checkbox"/>		
accessRightsProfile_Rights_AccountDelete	boolean	<input type="checkbox"/>		
accessRightsProfile_Rights_Administrator	string	<input type="checkbox"/>		
accessRightsProfile_Rights_Case	string	<input type="checkbox"/>		
accessRightsProfile_Rights_CaseDelete	boolean	<input type="checkbox"/>		
accessRightsProfile_Rights_CaseOwnerWorkHourApproval	boolean	<input type="checkbox"/>		
accessRightsProfile_Rights_ScheduleJobs	string	<input type="checkbox"/>		
accessRightsProfile_Rights_Sharing	boolean	<input type="checkbox"/>		
accessRightsProfile_Rights_TravelReimbursement	string	<input type="checkbox"/>		
accessRightsProfile_Rights_Users	string	<input type="checkbox"/>		
accessRightsProfile_Rights_WorkHourApproval	string	<input type="checkbox"/>		

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	string	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
NAME	string	Name		
RIGHTS_ACCOUNT	string			
RIGHTS_ACCOUNTDELETE	boolean			
RIGHTS_ADMINISTRATOR	string			
RIGHTS_CASE	string			
RIGHTS_CASEDELETE	boolean			
RIGHTS_CASEOWNERWORKHOURAPPROVAL	boolean			
RIGHTS_SCHEDULEJOBS	string			
RIGHTS_SHARING	boolean			
RIGHTS_TRAVELREIMBURSEMENT	string			
RIGHTS_USERS	string			

Name	Data Type	Label	Required	Documentation
RIGHTS_WORKHOURAPPROVAL	string			

3.1.26 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-
 accountInfo_CompanyGUID- accountInfo_CurrencyCode- accountInfo_CurrencyGUID-
 accountInfo_GUID- accountInfo_InsertTS- accountInfo_InvoicingVat- accountInfo_IsActive-
 accountInfo_IsInternal- 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_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	string	<input type="checkbox"/>		
accountInfo_AccountOwnerUserGUID	string	<input type="checkbox"/>		
accountInfo_AccountRating	int32	<input type="checkbox"/>		
accountInfo_CompanyGUID	string	<input type="checkbox"/>		
accountInfo_CurrencyCode	string	<input type="checkbox"/>		
accountInfo_CurrencyGUID	string	<input type="checkbox"/>		
accountInfo_eInvoiceAddress	string	<input type="checkbox"/>		
accountInfo_eInvoiceOperator	string	<input type="checkbox"/>		
accountInfo_GUID	string	<input type="checkbox"/>		
accountInfo_InsertTS	datetime	<input type="checkbox"/>		
accountInfo_InvoicingVat	decimal	<input type="checkbox"/>		
accountInfo_IsActive	boolean	<input type="checkbox"/>		
accountInfo_IsInternal	boolean	<input type="checkbox"/>		
accountInfo_LanguageCode	string	<input type="checkbox"/>		
accountInfo_LanguageGUID	string	<input type="checkbox"/>		
accountInfo_Name	string	<input type="checkbox"/>		
accountInfo_Notes	string	<input type="checkbox"/>		
accountInfo_Number	int64	<input type="checkbox"/>		
accountInfo_OverdueInterest	decimal	<input type="checkbox"/>		
accountInfo_PaymentTerm	int32	<input type="checkbox"/>		
accountInfo_PricelistGUID	string	<input type="checkbox"/>		
accountInfo_ReverseCharge	boolean	<input type="checkbox"/>		
accountInfo_ReverseChargeDescription	string	<input type="checkbox"/>		
accountInfo_UpdatedTS	datetime	<input type="checkbox"/>		
accountInfo_UrlToAccount	string	<input type="checkbox"/>		
addressInfo_Addressline	string	<input type="checkbox"/>		
addressInfo_Addressline2	string	<input type="checkbox"/>		
addressInfo_Addressline3	string	<input type="checkbox"/>		
addressInfo_City	string	<input type="checkbox"/>		
addressInfo_CompanyGUID	string	<input type="checkbox"/>		
addressInfo_CountryCode	string	<input type="checkbox"/>		
addressInfo_CountryGUID	string	<input type="checkbox"/>		
addressInfo_CountryName	string	<input type="checkbox"/>		
addressInfo_CountryRegionGUID	string	<input type="checkbox"/>		
addressInfo_CountryRegionName	string	<input type="checkbox"/>		
addressInfo_Fax	string	<input type="checkbox"/>		
addressInfo_GUID	string	<input type="checkbox"/>		

Name	Data Type	Required	Default Value	Documentation
addressInfo_IsBillingAddress	boolean	<input type="checkbox"/>		
addressInfo_IsPostalAddress	boolean	<input type="checkbox"/>		
addressInfo_IsVisitAddress	boolean	<input type="checkbox"/>		
addressInfo_Phone	string	<input type="checkbox"/>		
addressInfo_PostalCode	string	<input type="checkbox"/>		
companyInfo_AnnualRevenue	int32	<input type="checkbox"/>		
companyInfo_AnnualRevenue2	int64	<input type="checkbox"/>		
companyInfo_Email	string	<input type="checkbox"/>		
companyInfo_Employees	int32	<input type="checkbox"/>		
companyInfo_GUID	string	<input type="checkbox"/>		
companyInfo_HeadOfficeAddress GUID	string	<input type="checkbox"/>		
companyInfo_HierarchyGUID	string	<input type="checkbox"/>		
companyInfo_IndustryGUID	string	<input type="checkbox"/>		
companyInfo_LanguageGUID	string	<input type="checkbox"/>		
companyInfo_Name	string	<input type="checkbox"/>		
companyInfo_TimezoneGUID	string	<input type="checkbox"/>		
companyInfo_VatNumber	string	<input type="checkbox"/>		
companyInfo_Website	string	<input type="checkbox"/>		
contactInfo_AccountGUID	string	<input type="checkbox"/>		
contactInfo_AddressGUID	string	<input type="checkbox"/>		
contactInfo_CommunicationMethods	string	<input type="checkbox"/>		
contactInfo_ContactRoleGUID	string	<input type="checkbox"/>		
contactInfo_DateOfBirth	datetime	<input type="checkbox"/>		
contactInfo_Description	string	<input type="checkbox"/>		
contactInfo_FirstName	string	<input type="checkbox"/>		
contactInfo_GUID	string	<input type="checkbox"/>		
contactInfo_IsActive	boolean	<input type="checkbox"/>		
contactInfo_JobTitle	string	<input type="checkbox"/>		
contactInfo_Keywords	string	<input type="checkbox"/>		
contactInfo_LanguageGUID	string	<input type="checkbox"/>		
contactInfo_LastName	string	<input type="checkbox"/>		
contactInfo_Salutation	string	<input type="checkbox"/>		
contactInfo_SatisfactionLevel	int32	<input type="checkbox"/>		
contactInfo_TimeZoneGUID	string	<input type="checkbox"/>		

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	string			
ACCOUNTOWNERUSERGUID	string	Account Owner User GUID		
ACCOUNTRATING	int32	Account Rating		
COMPANYGUID	string	Company GUID		
CURRENCYCODE	string	Currency Code		
CURRENCYGUID	string	Currency GUID		
EINVOICEADDRESS	string	e-invoice Address		
EINVOICEOPERATOR	string	e-invoice Operator		
GUID	string	GUID		
INSERTTS	datetime	Inserted		
INVOICINGVAT	decimal	Invoicing VAT		
ISACTIVE	boolean	Is Active		
ISINTERNAL	boolean	Is Internal		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
NAME	string	Name		
NOTES	string	Notes		
NUMBER	int64	Number		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTTERM	int32	Payment Term		
PRICELISTGUID	string	Pricelist GUID		
REVERSECHARGE	boolean	Reverse Charge		
REVERSECHARGEDESCRIPTION	string	Reverse Charge Description		
UPDATEDTS	datetime	Updated		
URLTOACCOUNT	string	URL to Account		

3.1.27 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.

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	string	<input type="checkbox"/>		
accountGroupInfo_IsActive	boolean	<input type="checkbox"/>		
accountGroupInfo_Name	string	<input type="checkbox"/>		
accountGroupInfo_ParentGUID	string	<input type="checkbox"/>		

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	string	GUID		
ISACTIVE	boolean	Is Active		
NAME	string	Name		
PARENTGUID	string	Parent GUID		

3.1.28 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 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	string	<input type="checkbox"/>		
accountGUID	string	<input type="checkbox"/>		

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	string	Dummy	<input checked="" type="checkbox"/>	Dummy column

3.1.29 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.

Name	Data Type	Required	Default Value	Documentation
activityInfo_AccountGUID	string	<input type="checkbox"/>		
activityInfo_ActivityTypeCode	string	<input type="checkbox"/>		
activityInfo_ActivityTypeGUID	string	<input type="checkbox"/>		
activityInfo_CaseGUID	string	<input type="checkbox"/>		
activityInfo_EndDateTime	datetime	<input type="checkbox"/>		
activityInfo_GUID	string	<input type="checkbox"/>		
activityInfo_IsAllDay	boolean	<input type="checkbox"/>		
activityInfo_IsClosed	boolean	<input type="checkbox"/>		
activityInfo_IsDuration	boolean	<input type="checkbox"/>		
activityInfo_IsPrivate	boolean	<input type="checkbox"/>		
activityInfo_IsReadOnly	boolean	<input type="checkbox"/>		
activityInfo_IsUnassigned	boolean	<input type="checkbox"/>		
activityInfo_Location	string	<input type="checkbox"/>		
activityInfo_Name	string	<input type="checkbox"/>		
activityInfo_Notes	string	<input type="checkbox"/>		
activityInfo_OwnerUserBusinessUnitCode	string	<input type="checkbox"/>		
activityInfo_OwnerUserCode	string	<input type="checkbox"/>		
activityInfo_OwnerUserFirstName	string	<input type="checkbox"/>		
activityInfo_OwnerUserGUID	string	<input type="checkbox"/>		
activityInfo_OwnerUserLastName	string	<input type="checkbox"/>		
activityInfo_StartDateTime	datetime	<input type="checkbox"/>		
activityInfo_TaskGUID	string	<input type="checkbox"/>		

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	string	Account GUID		
ACTIVITYTYPECODE	string	Activity Type Code		
ACTIVITYTYPEGUID	string	Activity Type GUID		
CASEGUID	string	Project GUID		
ENDDATETIME	datetime	End Date Time		
GUID	string	GUID		

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

3.1.30 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	string	<input type="checkbox"/>		
contactGUID	string	<input type="checkbox"/>		

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			

3.1.31 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 Invariantive UniversalSQL driver 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	string	<input type="checkbox"/>		
resourceGUID	string	<input type="checkbox"/>		

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
<code>ADDNEWACTIVITYRESOURCEPARTICIPANTRESULT</code>	boolean			

3.1.32 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
<code>activityTypeInfo_Category</code>	string	<input type="checkbox"/>		
<code>activityTypeInfo_Code</code>	string	<input type="checkbox"/>		
<code>activityTypeInfo_GUID</code>	string	<input type="checkbox"/>		
<code>activityTypeInfo_Icon</code>	string	<input type="checkbox"/>		
<code>activityTypeInfo_IsActive</code>	boolean	<input type="checkbox"/>		
<code>activityTypeInfo_IsDefault</code>	boolean	<input type="checkbox"/>		
<code>activityTypeInfo_IsPaidLeave</code>	boolean	<input type="checkbox"/>		
<code>activityTypeInfo_Name</code>	string	<input type="checkbox"/>		

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	string	Category		
CODE	string	Code		
GUID	string	GUID		
ICON	string	Icon		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
ISPAIDLEAVE	boolean	Is Paid Leave		
NAME	string	Name		

3.1.33 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 Invariantive UniversalSQL driver 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	string	<input type="checkbox"/>		
memberStatus	int16	<input type="checkbox"/>		
userGUID	string	<input type="checkbox"/>		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ADDNEWACTIVITYUSERPARTICIPANTRESULT	boolean			

3.1.34 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	string	<input type="checkbox"/>		
addressInfo_Addressline2	string	<input type="checkbox"/>		
addressInfo_Addressline3	string	<input type="checkbox"/>		
addressInfo_City	string	<input type="checkbox"/>		
addressInfo_CompanyGUID	string	<input type="checkbox"/>		
addressInfo_CountryCode	string	<input type="checkbox"/>		
addressInfo_CountryGUID	string	<input type="checkbox"/>		

Name	Data Type	Required	Default Value	Documentation
addressInfo_CountryName	string	<input type="checkbox"/>		
addressInfo_CountryRegionGUID	string	<input type="checkbox"/>		
addressInfo_CountryRegionName	string	<input type="checkbox"/>		
addressInfo_Fax	string	<input type="checkbox"/>		
addressInfo_GUID	string	<input type="checkbox"/>		
addressInfo_IsBillingAddress	boolean	<input type="checkbox"/>		
addressInfo_IsPostalAddress	boolean	<input type="checkbox"/>		
addressInfo_IsVisitAddress	boolean	<input type="checkbox"/>		
addressInfo_Phone	string	<input type="checkbox"/>		
addressInfo_PostalCode	string	<input type="checkbox"/>		

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

3.1.35 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.

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	decimal	<input type="checkbox"/>		
billingAndExpenseForecastInfo_ExpenseForecast	decimal	<input type="checkbox"/>		
billingAndExpenseForecastInfo_GUID	string	<input type="checkbox"/>		
billingAndExpenseForecastInfo_InhouseRevenueForecast	decimal	<input type="checkbox"/>		
billingAndExpenseForecastInfo_Month	int32	<input type="checkbox"/>		
billingAndExpenseForecastInfo_Notes	string	<input type="checkbox"/>		
billingAndExpenseForecastInfo_OutsourcingRevenueForecast	decimal	<input type="checkbox"/>		
billingAndExpenseForecastInfo_Year	int32	<input type="checkbox"/>		
caseGuid	string	<input type="checkbox"/>		

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	decimal	Billing Forecast		
EXPENSEFORECAST	decimal	Expense Forecast		
GUID	string	GUID		
INHOUSEREVENUEFORECAST	decimal	Inhouse Revenue Forecast		
MONTH	int32	Month		
NOTES	string	Notes		
OUTSOURCINGREVENUEFORECAST	decimal	Outsourcing Revenue Forecast		
YEAR	int32	Year		

3.1.36 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_OverDuelInterest- 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	string	<input type="checkbox"/>		
businessUnitInfo_CostCenterGUID	string	<input type="checkbox"/>		
businessUnitInfo_CurrencyCode	string	<input type="checkbox"/>		
businessUnitInfo_CurrencyGUID	string	<input type="checkbox"/>		
businessUnitInfo_FormattingCulture	string	<input type="checkbox"/>		
businessUnitInfo_GUID	string	<input type="checkbox"/>		
businessUnitInfo_InsertTS	datetime	<input type="checkbox"/>		
businessUnitInfo_IsActive	boolean	<input type="checkbox"/>		
businessUnitInfo_LanguageCode	string	<input type="checkbox"/>		
businessUnitInfo_LanguageGUID	string	<input type="checkbox"/>		
businessUnitInfo_Name	string	<input type="checkbox"/>		
businessUnitInfo_NextInvoiceNumber	int32	<input type="checkbox"/>		
businessUnitInfo_OverDueInterest	decimal	<input type="checkbox"/>		
businessUnitInfo_ParentBusinessUnitGUID	string	<input type="checkbox"/>		
businessUnitInfo_PaymentTerm	int32	<input type="checkbox"/>		
businessUnitInfo_Show Logo On Invoices	boolean	<input type="checkbox"/>		
businessUnitInfo_Show Logo On Pdf Reports	boolean	<input type="checkbox"/>		
businessUnitInfo_TimeZoneGUID	string	<input type="checkbox"/>		
businessUnitInfo_UpdatedTS	datetime	<input type="checkbox"/>		
businessUnitInfo_VatNumber	string	<input type="checkbox"/>		

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	string	Code		
COSTCENTERGUID	string	Cost Center GUID		
CURRENCYCODE	string	Currency Code		
CURRENCYGUID	string	Currency GUID		
FORMATTINGCULTURE	string	Formatting Culture		
GUID	string	GUID		
INSERTTS	datetime	Inserted		
ISACTIVE	boolean	Is Active		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
NAME	string	Name		
NEXTINVOICENUMBER	int32	Next Invoice Number		

Name	Data Type	Label	Required	Documentation
OVERDUEINTEREST	decimal	Overdue Interest		
PARENTBUSINESSUNITGUID	string	Parent Business Unit GUID		
PAYMENTTERM	int32	Payment Term		
SHOWLOGOONINVOICES	boolean	Show Logo on Invoices		
SHOWLOGOONPDFREPORTS	boolean	Show Logo on PDF Reports		
TIMEZONEGUID	string	Timezone GUID		
UPDATEDTS	datetime	Updated		
VATNUMBER	string	VAT Number		

3.1.37 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: caselInfo_AccountGUID- caselInfo_AccountName- caselInfo_AccountNumber- caselInfo_BillingAddress- caselInfo_BillingAddressGUID- caselInfo_BillingInternalNotes- caselInfo_BillingNotesBeforeGrid- caselInfo_BusinessUnitGUID- caselInfo_BusinessUnitName- caselInfo_BusinessUnitNumber- caselInfo_CaseCompanyCurrencyGUID- caselInfo_CaseNumber- caselInfo_CaseOwnerUserCode- caselInfo_CaseOwnerUserGUID- caselInfo_ClosedDate- caselInfo_CompletionEstimate- caselInfo_ContactGUID- caselInfo_CostCenterGUID- caselInfo_CostCenterNumber- caselInfo_CurrencyGUID- caselInfo_CurrencyShortform- caselInfo_CurrencyShortformOfCaseCompanyCurrency- caselInfo_CurrentCaseStatusDescription- caselInfo_CurrentCaseStatusGUID- caselInfo_CurrentCaseStatusInsertTs- caselInfo_DeadlineDate- caselInfo_Description- caselInfo_ExpectedValue- caselInfo_GUID- caselInfo_IncludeExceptions- caselInfo_IncludeOverTime- caselInfo_InsertTs- caselInfo_InternalName- caselInfo_InvoiceTemplateGUID- caselInfo_IsClosed- caselInfo_IsDailyAllowanceBillable- caselInfo_IsInternal- caselInfo_IsMileageBillable- caselInfo_IsOtherTravelExpensesBillable- caselInfo_LatestEstimationDate- caselInfo_LeadRating- caselInfo_LeadSourceGUID- caselInfo_Name- caselInfo_OrderNumber- caselInfo_OurReference- caselInfo_OverdueInterest- caselInfo_PaymentTerm- caselInfo_PricelistGUID- caselInfo_PricingRule- caselInfo_PricingType- caselInfo_Priority- caselInfo_Probability- caselInfo_RootTaskGUID- caselInfo_SalesCloseDate- caselInfo_SalesPersonUserCode- caselInfo_SalesPersonUserGUID- caselInfo_SalesProcessGUID- caselInfo_SalesStatusGUID- caselInfo_SharedToExtranet- caselInfo_StartDate- caselInfo_UpdatedTs- caselInfo_UrlToCase- caselInfo_UseDefaultProductsInWorkTimeEntry- caselInfo_UseDefaultWorkTypes- caselInfo_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 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	string	<input type="checkbox"/>		
caseInfo_AccountName	string	<input type="checkbox"/>		
caseInfo_AccountNumber	int64	<input type="checkbox"/>		
caseInfo_BillingAddress	string	<input type="checkbox"/>		
caseInfo_BillingAddressGUID	string	<input type="checkbox"/>		
caseInfo_BillingInternalNotes	string	<input type="checkbox"/>		
caseInfo_BillingNotesBeforeGrid	string	<input type="checkbox"/>		
caseInfo_BusinessUnitGUID	string	<input type="checkbox"/>		
caseInfo_BusinessUnitName	string	<input type="checkbox"/>		
caseInfo_BusinessUnitNumber	string	<input type="checkbox"/>		
caseInfo_CaseCompanyCurrencyGUID	string	<input type="checkbox"/>		
caseInfo_CaseNumber	int64	<input type="checkbox"/>		
caseInfo_CaseOwnerUserCode	string	<input type="checkbox"/>		
caseInfo_CaseOwnerUserGUID	string	<input type="checkbox"/>		
caseInfo_ClosedDate	datetime	<input type="checkbox"/>		
caseInfo_CompletionEstimate	int32	<input type="checkbox"/>		
caseInfo_ContactGUID	string	<input type="checkbox"/>		
caseInfo_CostCenterGUID	string	<input type="checkbox"/>		
caseInfo_CostCenterNumber	string	<input type="checkbox"/>		
caseInfo_CurrencyGUID	string	<input type="checkbox"/>		
caseInfo_CurrencyShortform	string	<input type="checkbox"/>		
caseInfo_CurrencyShortformOfCaseCompanyCurrency	string	<input type="checkbox"/>		
caseInfo_CurrentCaseStatusDescription	string	<input type="checkbox"/>		
caseInfo_CurrentCaseStatusGUID	string	<input type="checkbox"/>		
caseInfo_CurrentCaseStatusInserts	datetime	<input type="checkbox"/>		
caseInfo_DeadlineDate	datetime	<input type="checkbox"/>		
caseInfo_Description	string	<input type="checkbox"/>		

Name	Data Type	Required	Default Value	Documentation
caseInfo_ExpectedValue	decimal	<input type="checkbox"/>		
caseInfo_GUID	string	<input type="checkbox"/>		
caseInfo_IncludeExceptions	boolean	<input type="checkbox"/>		
caseInfo_IncludeOverTime	boolean	<input type="checkbox"/>		
caseInfo_InsertTs	datetime	<input type="checkbox"/>		
caseInfo_InternalName	string	<input type="checkbox"/>		
caseInfo_InvoiceTemplateGUID	string	<input type="checkbox"/>		
caseInfo_IsClosed	boolean	<input type="checkbox"/>		
caseInfo_IsDailyAllow anceBillable	boolean	<input type="checkbox"/>		
caseInfo_IsInternal	boolean	<input type="checkbox"/>		
caseInfo_IsMileageBillable	boolean	<input type="checkbox"/>		
caseInfo_IsOtherTravelExpensesBillable	boolean	<input type="checkbox"/>		
caseInfo_LatestEstimationDate	datetime	<input type="checkbox"/>		
caseInfo_LeadRating	int32	<input type="checkbox"/>		
caseInfo_LeadSourceGUID	string	<input type="checkbox"/>		
caseInfo_Name	string	<input type="checkbox"/>		
caseInfo_OrderNumber	string	<input type="checkbox"/>		
caseInfo_OurReference	string	<input type="checkbox"/>		
caseInfo_OverdueInterest	decimal	<input type="checkbox"/>		
caseInfo_PaymentTerm	int32	<input type="checkbox"/>		
caseInfo_PricelistGUID	string	<input type="checkbox"/>		
caseInfo_PricingRule	string	<input type="checkbox"/>		
caseInfo_PricingType	string	<input type="checkbox"/>		
caseInfo_Priority	int32	<input type="checkbox"/>		
caseInfo_Probability	int32	<input type="checkbox"/>		
caseInfo_RootTaskGUID	string	<input type="checkbox"/>		
caseInfo_SalesCloseDate	datetime	<input type="checkbox"/>		
caseInfo_SalesPersonUserCode	string	<input type="checkbox"/>		
caseInfo_SalesPersonUserGUID	string	<input type="checkbox"/>		
caseInfo_SalesProcessGUID	string	<input type="checkbox"/>		
caseInfo_SalesStatusGUID	string	<input type="checkbox"/>		
caseInfo_SharedToExtranet	boolean	<input type="checkbox"/>		
caseInfo_StartDate	datetime	<input type="checkbox"/>		
caseInfo_UpdatedTs	datetime	<input type="checkbox"/>		
caseInfo_UrlToCase	string	<input type="checkbox"/>		
caseInfo_UseDefaultProductsInWorkTimeEntry	boolean	<input type="checkbox"/>		
caseInfo_UseDefaultWorkTypes	boolean	<input type="checkbox"/>		
caseInfo_YourReference	string	<input type="checkbox"/>		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	string	Account GUID		
ACCOUNTNAME	string	Account Name		
ACCOUNTNUMBER	int64	Account Number		
BILLINGADDRESS	string	Billing Address		
BILLINGADDRESSGUID	string	Billing Address GUID		
BILLINGINTERNALNOTES	string	Billing Internal Notes		
BILLINGNOTESBEFOREGRID	string	Billing Notes before Grid		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
BUSINESSUNITNUMBER	string	Business Unit Number		
CASECOMPANYCURRENCYGUID	string	Project Company Currency GUID		
CASENUMBER	int64	Project Number		
CASEOWNERUSERCODE	string	Project Owner User Code		
CASEOWNERUSERGUID	string	Project Owner User GUID		
CLOSEDDATE	datetime	Closed Date		
COMPLETIONESTIMATE	int32	Completion Estimate		
CONTACTGUID	string	Contact GUID		
COSTCENTERGUID	string	Cost Center GUID		
COSTCENTERNUMBER	string	Cost Center Number		
CURRENCYGUID	string	Currency GUID		
CURRENCYSHORTFORM	string	Currency Short Form		
CURRENCYSHORTFORMOFCASECOMPANYCURRENCY	string	Currency Short Form of Project Company Currency		
CURRENTCASESTATUSDESCRIPTION	string	Current Project Status Description		
CURRENTCASESTATUSGUID	string	Current Project Status GUID		
CURRENTCASESTATUSINSERTTS	datetime	Current Project Status Inserted		
DEADLINE DATE	datetime	Deadline Date		
DESCRIPTION	string	Description		
EXPECTEDVALUE	decimal	Expected Value		
GUID	string	GUID		
INCLUDEEXCEPTIONS	boolean	Include Exceptions		
INCLUDEOVERTIME	boolean	Include Overtime		

Name	Data Type	Label	Required	Documentation
INSERTTS	datetime	Inserted		
INTERNALNAME	string	Internal Name		
INVOICETEMPLATEGUID	string	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	int32	Lead Rating		
LEADSOURCEGUID	string	Lead Source GUID		
NAME	string	Name		
ORDERNUMBER	string	Order Number		
OURREFERENCE	string	Our Reference		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTTERM	int32	Payment Term		
PRICELISTGUID	string	Pricelist GUID		
PRICINGRULE	string	Pricing Rule		
PRICINGTYPE	string	Pricing Type		
PRIORITY	int32	Priority		
PROBABILITY	int32	Probability		
ROOTTASKGUID	string	Root Task GUID		
SALESCLOSEDATE	datetime	Sales Close Date		
SALESPERSONUSERCODE	string	Sales Person User Code		
SALESPERSONUSERGUID	string	Sales Person User GUID		
SALESPROCESSGUID	string	Sales Process GUID		
SALESSTATUSGUID	string	Sales Status GUID		
SHAREDTOEXTRANET	boolean	Shared to Extranet		
STARTDATE	datetime	Start Date		
UPDATEDTS	datetime	Updated		
URLTOCASE	string	URL to Project		
USEDEFAULTPRODUCTSINWORKTIMEENTRY	boolean	Use Default Products in Work Time Entry		
USEDEFAULTWORKTYPES	boolean	Use Default Work Types		
YOURREFERENCE	string	Your Reference		

3.1.38 AddNewCaseStatus: Visma Severa Add New Project Status

Catalog: Severa

Schema: API

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	string	<input type="checkbox"/>		
caseStatusInfo_Icon	string	<input type="checkbox"/>		
caseStatusInfo_IsActive	boolean	<input type="checkbox"/>		
caseStatusInfo_Name	string	<input type="checkbox"/>		
insertPosition	int32	<input type="checkbox"/>		

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	string	GUID		
ICON	string	Icon		
ISACTIVE	boolean	Is Active		
NAME	string	Name		

3.1.39 AddNewCaseTag: Visma Severa Add New Project Tag

Catalog: Severa

Schema: API

Label: Add New Project 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: 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	string	<input type="checkbox"/>		
tagInfo_Context	string	<input type="checkbox"/>		
tagInfo_GUID	string	<input type="checkbox"/>		
tagInfo_IsActive	boolean	<input type="checkbox"/>		
tagInfo_Keyword	string	<input type="checkbox"/>		
tagInfo_Weight	int32	<input type="checkbox"/>		

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	string	Context		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
KEYWORD	string	Keyw ord		
WEIGHT	int32	Weight		

3.1.40 AddNewContact: Visma Severa Add New Contact

Catalog: Severa

Schema: API

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	string	<input type="checkbox"/>		
contactInfo_AddressGUID	string	<input type="checkbox"/>		
contactInfo_CommunicationMethods	string	<input type="checkbox"/>		
contactInfo_ContactRoleGUID	string	<input type="checkbox"/>		
contactInfo_DateOfBirth	datetime	<input type="checkbox"/>		
contactInfo_Description	string	<input type="checkbox"/>		
contactInfo_FirstName	string	<input type="checkbox"/>		
contactInfo_GUID	string	<input type="checkbox"/>		
contactInfo_IsActive	boolean	<input type="checkbox"/>		
contactInfo_JobTitle	string	<input type="checkbox"/>		
contactInfo_Keyw ords	string	<input type="checkbox"/>		
contactInfo_LanguageGUID	string	<input type="checkbox"/>		
contactInfo_LastName	string	<input type="checkbox"/>		
contactInfo_Salutation	string	<input type="checkbox"/>		
contactInfo_SatisfactionLevel	int32	<input type="checkbox"/>		

Name	Data Type	Required	Default Value	Documentation
contactInfo_TimeZoneGUID	string	<input type="checkbox"/>		

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	string	Account GUID		
ADDRESSGUID	string	Address GUID		
COMMUNICATIONMETHODS	string			
CONTACTROLEGUID	string	Contact Role GUID		
DATEOFBIRTH	datetime	Date of Birth		
DESCRIPTION	string	Ddescription		
FIRSTNAME	string	First Name		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
JOBTITLE	string	Job Title		
KEYWORDS	string			
LANGUAGEGUID	string	Language GUID		
LASTNAME	string	Last Name		
SALUTATION	string	Salutation		
SATISFACTIONLEVEL	int32	Satisfaction Level		
TIMEZONEGUID	string	Timezone GUID		

3.1.41 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.

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	string	<input type="checkbox"/>		
tagInfo_Context	string	<input type="checkbox"/>		
tagInfo_GUID	string	<input type="checkbox"/>		
tagInfo_IsActive	boolean	<input type="checkbox"/>		
tagInfo_Keyw ord	string	<input type="checkbox"/>		
tagInfo_Weight	int32	<input type="checkbox"/>		

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	string	Context		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
KEYWORD	string	Keyw ord		
WEIGHT	int32	Weight		

3.1.42 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_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

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	string	<input type="checkbox"/>		
customer_AccountOwnerUserGUID	string	<input type="checkbox"/>		
customer_AccountRating	int32	<input type="checkbox"/>		
customer_AnnualRevenue	int32	<input type="checkbox"/>		
customer_AnnualRevenue2	int64	<input type="checkbox"/>		
customer_CompanyGUID	string	<input type="checkbox"/>		
customer_CurrencyCode	string	<input type="checkbox"/>		
customer_CurrencyGUID	string	<input type="checkbox"/>		
customer_eInvoiceAddress	string	<input type="checkbox"/>		
customer_eInvoiceOperator	string	<input type="checkbox"/>		
customer_Email	string	<input type="checkbox"/>		
customer_Employees	int32	<input type="checkbox"/>		
customer_GUID	string	<input type="checkbox"/>		
customer_HeadOfficeAddressGUID	string	<input type="checkbox"/>		
customer_IndustryGUID	string	<input type="checkbox"/>		
customer_InsertTS	datetime	<input type="checkbox"/>		
customer_InvoicingVat	decimal	<input type="checkbox"/>		
customer_IsActive	boolean	<input type="checkbox"/>		
customer_IsInternal	boolean	<input type="checkbox"/>		
customer_LanguageCode	string	<input type="checkbox"/>		
customer_LanguageGUID	string	<input type="checkbox"/>		
customer_Name	string	<input type="checkbox"/>		
customer_Notes	string	<input type="checkbox"/>		
customer_Number	int64	<input type="checkbox"/>		

Name	Data Type	Required	Default Value	Documentation
customer_OverdueInterest	decimal	<input type="checkbox"/>		
customer_PaymentTerm	int32	<input type="checkbox"/>		
customer_PricelistGUID	string	<input type="checkbox"/>		
customer_ReverseCharge	boolean	<input type="checkbox"/>		
customer_ReverseChargeDescription	string	<input type="checkbox"/>		
customer_TimezoneGUID	string	<input type="checkbox"/>		
customer_UpdatedTS	datetime	<input type="checkbox"/>		
customer_UrIToAccount	string	<input type="checkbox"/>		
customer_VatNumber	string	<input type="checkbox"/>		
customer_Website	string	<input type="checkbox"/>		

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	string			
ACCOUNTOWNERUSERGUID	string	Account Owner User GUID		
ACCOUNTRATING	int32	Account Rating		
ANNUALREVENUE	int32	Annual Revenue		
ANNUALREVENUE2	int64			
COMPANYGUID	string	Company GUID		
CURRENCYCODE	string	Currency Code		
CURRENCYGUID	string	Currency GUID		
EINVOICEADDRESS	string	e-invoice Address		
EINVOICEOPERATOR	string	e-invoice Operator		
EMAIL	string	Email		
EMPLOYEES	int32	Employees		
GUID	string	GUID		
HEADOFFICEADDRESSGUID	string	Head Office Address GUID		
INDUSTRYGUID	string	Industry GUID		
INSERTTS	datetime	Inserted		
INVOICINGVAT	decimal	Invoicing VAT		
ISACTIVE	boolean	Is Active		
ISINTERNAL	boolean	Is Internal		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
NAME	string	Name		

Name	Data Type	Label	Required	Documentation
NOTES	string	Notes		
NUMBER	int64	Number		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTTERM	int32	Payment Term		
PRICELISTGUID	string	Pricelist GUID		
REVERSECHARGE	boolean	Reverse Charge		
REVERSECHARGEDESCRIPTION	string	Reverse Charge Description		
TIMEZONEGUID	string	Timezone GUID		
UPDATEDTS	datetime	Updated		
URLTOACCOUNT	string	URL to Account		
VATNUMBER	string	VAT Number		
WEBSITE	string	Website		

3.1.43 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.

Name	Data Type	Required	Default Value	Documentation
employmentInfo_CompanyBusinessUnitCode	string	<input type="checkbox"/>		
employmentInfo_CompanyBusinessUnitGUID	string	<input type="checkbox"/>		
employmentInfo_CompanyBusinessUnitName	string	<input type="checkbox"/>		
employmentInfo_DailyHours	decimal	<input type="checkbox"/>		
employmentInfo_EndDate	datetime	<input type="checkbox"/>		
employmentInfo_GUID	string	<input type="checkbox"/>		
employmentInfo_HourCost	decimal	<input type="checkbox"/>		
employmentInfo_IsOvertimeAllowed	boolean	<input type="checkbox"/>		
employmentInfo_StartDate	datetime	<input type="checkbox"/>		
employmentInfo_Title	string	<input type="checkbox"/>		
userGUID	string	<input type="checkbox"/>		

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	string	Company Business Unit Code		
COMPANYBUSINESSUNITGUID	string	Company Business Unit GUID		
COMPANYBUSINESSUNITNAME	string	Company Business Unit Name		
DAILYHOURS	decimal	Daily Hours		
ENDDATE	datetime	End Date		
GUID	string	GUID		
HOURLCOST	decimal	Hour Cost		
ISOVERTIMEALLOWED	boolean	Is Overtime Allowed		
STARTDATE	datetime	Start Date		
TITLE	string	Title		

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

Catalog: Severa

Schema: API

Label: Add New Expense to Product by PO 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- 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	string	<input type="checkbox"/>		
itemInfo_BillingSchedule	string	<input type="checkbox"/>		
itemInfo_CaseGUID	string	<input type="checkbox"/>		
itemInfo_CaseNumber	int64	<input type="checkbox"/>		

Name	Data Type	Required	Default Value	Documentation
itemInfo_CostCenterGUID	string	<input type="checkbox"/>		
itemInfo_CostCenterIdentifier	string	<input type="checkbox"/>		
itemInfo_CostCurrencyShortform	string	<input type="checkbox"/>		
itemInfo_CostCurrencyShortformOf InvoicingCurrency	string	<input type="checkbox"/>		
itemInfo_CostCurrencyShortformOf OrganizationCurrency	string	<input type="checkbox"/>		
itemInfo_CostCurrencyShortformOf UserCompanyCurrency	string	<input type="checkbox"/>		
itemInfo_Description	string	<input type="checkbox"/>		
itemInfo_DisplayPeriodStart	datetime	<input type="checkbox"/>		
itemInfo_EventDate	datetime	<input type="checkbox"/>		
itemInfo_GUID	string	<input type="checkbox"/>		
itemInfo_InvoiceDescription	string	<input type="checkbox"/>		
itemInfo_InvoiceGUID	string	<input type="checkbox"/>		
itemInfo_InvoiceQuantity	decimal	<input type="checkbox"/>		
itemInfo_InvoiceRow GUID	string	<input type="checkbox"/>		
itemInfo_IsApproved	boolean	<input type="checkbox"/>		
itemInfo_IsBillable	boolean	<input type="checkbox"/>		
itemInfo_IsOutSourcingCost	boolean	<input type="checkbox"/>		
itemInfo_IsReadOnly	boolean	<input type="checkbox"/>		
itemInfo_IsRealised	boolean	<input type="checkbox"/>		
itemInfo_MeasurementUnit	string	<input type="checkbox"/>		
itemInfo_Name	string	<input type="checkbox"/>		
itemInfo_OrderDate	datetime	<input type="checkbox"/>		
itemInfo_PlannedBillingDate	datetime	<input type="checkbox"/>		
itemInfo_PriceCurrencyShortformOf CaseCompanyCurrency	string	<input type="checkbox"/>		
itemInfo_PriceCurrencyShortformOf InvoicingCurrency	string	<input type="checkbox"/>		
itemInfo_PriceCurrencyShortformOf OrganizationCurrency	string	<input type="checkbox"/>		
itemInfo_PriceCurrencyShortformOf UserCompanyCurrency	string	<input type="checkbox"/>		
itemInfo_ProductCode	string	<input type="checkbox"/>		
itemInfo_ProductGUID	string	<input type="checkbox"/>		
itemInfo_ProductType	string	<input type="checkbox"/>		
itemInfo_PurchaseOrderNumber	string	<input type="checkbox"/>		
itemInfo_PurchaseVAT	decimal	<input type="checkbox"/>		
itemInfo_Quantity	decimal	<input type="checkbox"/>		
itemInfo_RecurrenceEndDate	datetime	<input type="checkbox"/>		
itemInfo_RecurrenceEndType	int32	<input type="checkbox"/>		
itemInfo_RecurrenceFrequency	int32	<input type="checkbox"/>		

Name	Data Type	Required	Default Value	Documentation
itemInfo_RecurrenceStartDate	datetime	<input type="checkbox"/>		
itemInfo_RecurrenceTimes	int32	<input type="checkbox"/>		
itemInfo_RecurringItemGUID	string	<input type="checkbox"/>		
itemInfo_SalesAccountGUID	string	<input type="checkbox"/>		
itemInfo_SalesStatusGUID	string	<input type="checkbox"/>		
itemInfo_TaskGUID	string	<input type="checkbox"/>		
itemInfo_TravelEndTime	datetime	<input type="checkbox"/>		
itemInfo_TravelStartTime	datetime	<input type="checkbox"/>		
itemInfo_UnitCost	decimal	<input type="checkbox"/>		
itemInfo_UnitCostInInvoicingCurrency	decimal	<input type="checkbox"/>		
itemInfo_UnitCostInOrganizationCurrency	decimal	<input type="checkbox"/>		
itemInfo_UnitCostInUserCompanyCurrency	decimal	<input type="checkbox"/>		
itemInfo_UnitCostVatIncludedAmount	decimal	<input type="checkbox"/>		
itemInfo_UnitPrice	decimal	<input type="checkbox"/>		
itemInfo_UnitPriceInBaseCurrency	decimal	<input type="checkbox"/>		
itemInfo_UnitPriceInCaseCompanyCurrency	decimal	<input type="checkbox"/>		
itemInfo_UnitPriceInUserCompanyCurrency	decimal	<input type="checkbox"/>		
itemInfo_UserCode	string	<input type="checkbox"/>		
itemInfo_UserGUID	string	<input type="checkbox"/>		
itemInfo_VAT	decimal	<input type="checkbox"/>		
poNumber	string	<input type="checkbox"/>		

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	string	Billing Dependency Task GUID		
BILLINGSCHEDULE	string	Billing Schedule		
CASEGUID	string	Project GUID		
CASENUMBER	int64	Project Number		
COSTCENTERGUID	string	Cost Center GUID		
COSTCENTERIDENTIFIER	string	Cost Center Identifier		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing		

Name	Data Type	Label	Required	Documentation
		Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User Company Currency		
DESCRIPTION	string	Description		
DISPLAYPERIODSTART	datetime	Display Period Start		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVICEROWGUID	string	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	string	Measurement Unit		
NAME	string	Name		
ORDERDATE	datetime	Order Date		
PLANNEDBILLINGDATE	datetime	Planned Billing Date		
PRICECURRENCYSHORTFORMOFPROJECTCOMPANYCURRENCY	string	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Price Currency Short Form of Organization Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Price Currency Short Form of User Company Currency		
PRODUCTCODE	string	Product Code		
PRODUCTGUID	string	Product GUID		
PRODUCTTYPE	string	Product Type		
PURCHASEORDERNUMBER	string	Purchase Order Number		
PURCHASEVAT	decimal	Purchase VAT		
QUANTITY	decimal	Quantity		
RECURRENCEENDDATE	datetime	Recurrence End Date		
RECURRENCEENDTYPE	int32	Recurrence End Type		

Name	Data Type	Label	Required	Documentation
RECURRENCFREQUENCY	int32	Recurrence Frequency		
RECURRENCESTARTDATE	datetime	Recurrence Start Date		
RECURRENCETIMES	int32	Recurrence Times		
RECURRINGITEMGUID	string	Recurring Item GUID		
SALESACCOUNTGUID	string	Sales Account GUID		
SALESSTATUSGUID	string	Sales Status GUID		
TASKGUID	string	Task GUID		
TRAVELENDTIME	datetime	Travel End Time		
TRAVELSTARTTIME	datetime	Travel Start Time		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	decimal	Unit Cost in Organization Currency		
UNITCOSTINUSERCOMPANYCURRENCY	decimal	Unit Cost in User Company Currency		
UNITCOSTVATINCLUDEDAMOUNT	decimal	Unit Cost VAT Included Amount		
UNITPRICE	decimal	Unit Price		
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		
UNITPRICEINPROJECTCOMPANYCURRENCY	decimal	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	decimal	Unit Price in User Company Currency		
USERCODE	string	User Code		
USERGUID	string	User GUID		
VAT	decimal	VAT		

3.1.45 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

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	string	<input type="checkbox"/>		
tagInfo_Context	string	<input type="checkbox"/>		
tagInfo_GUID	string	<input type="checkbox"/>		
tagInfo_IsActive	boolean	<input type="checkbox"/>		
tagInfo_Keyw ord	string	<input type="checkbox"/>		
tagInfo_Weight	int32	<input type="checkbox"/>		

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	string	Context		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
KEYWORD	string	Keyw ord		
WEIGHT	int32	Weight		

3.1.46 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-

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	string	<input type="checkbox"/>		
hourEntryInfo_CaseBusinessUnitCode	string	<input type="checkbox"/>		
hourEntryInfo_CaseCostCenterIdentifier	string	<input type="checkbox"/>		
hourEntryInfo_CaseGUID	string	<input type="checkbox"/>		
hourEntryInfo_CaseName	string	<input type="checkbox"/>		
hourEntryInfo_CaseNumber	int64	<input type="checkbox"/>		
hourEntryInfo_CostCurrencyShortform	string	<input type="checkbox"/>		
hourEntryInfo_CostCurrencyShortformOfInvoicingCurrency	string	<input type="checkbox"/>		
hourEntryInfo_CostCurrencyShortformOfOrganizationCurrency	string	<input type="checkbox"/>		

Name	Data Type	Required	Default Value	Documentation
hourEntryInfo_CostCurrencyShortformOfUserCompanyCurrency	string	<input type="checkbox"/>		
hourEntryInfo_Description	string	<input type="checkbox"/>		
hourEntryInfo_EndTime	datetime	<input type="checkbox"/>		
hourEntryInfo_EventDate	datetime	<input type="checkbox"/>		
hourEntryInfo_GUID	string	<input type="checkbox"/>		
hourEntryInfo_InvoiceDescription	string	<input type="checkbox"/>		
hourEntryInfo_InvoiceGUID	string	<input type="checkbox"/>		
hourEntryInfo_InvoiceQuantity	decimal	<input type="checkbox"/>		
hourEntryInfo_InvoiceRow GUID	string	<input type="checkbox"/>		
hourEntryInfo_IsApproved	boolean	<input type="checkbox"/>		
hourEntryInfo_IsBillable	boolean	<input type="checkbox"/>		
hourEntryInfo_IsProductive	boolean	<input type="checkbox"/>		
hourEntryInfo_IsReadOnly	boolean	<input type="checkbox"/>		
hourEntryInfo_OvertimeCode	string	<input type="checkbox"/>		
hourEntryInfo_OverTimeGUID	string	<input type="checkbox"/>		
hourEntryInfo_PhaseGUID	string	<input type="checkbox"/>		
hourEntryInfo_PriceCurrencyShortformOfCaseCompanyCurrency	string	<input type="checkbox"/>		
hourEntryInfo_PriceCurrencyShortformOfInvoicingCurrency	string	<input type="checkbox"/>		
hourEntryInfo_PriceCurrencyShortformOfOrganizationCurrency	string	<input type="checkbox"/>		
hourEntryInfo_PriceCurrencyShortformOfUserCompanyCurrency	string	<input type="checkbox"/>		
hourEntryInfo_Quantity	decimal	<input type="checkbox"/>		
hourEntryInfo_StartTime	datetime	<input type="checkbox"/>		
hourEntryInfo_TaskGUID	string	<input type="checkbox"/>		
hourEntryInfo_UnitCost	decimal	<input type="checkbox"/>		
hourEntryInfo_UnitCostInInvoicingCurrency	decimal	<input type="checkbox"/>		
hourEntryInfo_UnitCostInOrganizationCurrency	decimal	<input type="checkbox"/>		
hourEntryInfo_UnitCostInUserCompanyCurrency	decimal	<input type="checkbox"/>		
hourEntryInfo_UnitPrice	decimal	<input type="checkbox"/>		
hourEntryInfo_UnitPriceInBaseCurrency	decimal	<input type="checkbox"/>		
hourEntryInfo_UnitPriceInCaseCompanyCurrency	decimal	<input type="checkbox"/>		
hourEntryInfo_UnitPriceInUserCompanyCurrency	decimal	<input type="checkbox"/>		
hourEntryInfo_UseInvoiceQuantity	boolean	<input type="checkbox"/>		
hourEntryInfo_UserBusinessUnitCode	string	<input type="checkbox"/>		

Name	Data Type	Required	Default Value	Documentation
hourEntryInfo_UserCode	string	<input type="checkbox"/>		
hourEntryInfo_UserFirstName	string	<input type="checkbox"/>		
hourEntryInfo_UserGUID	string	<input type="checkbox"/>		
hourEntryInfo_UserLastName	string	<input type="checkbox"/>		
hourEntryInfo_WorkTypeCode	string	<input type="checkbox"/>		
hourEntryInfo_WorkTypeGUID	string	<input type="checkbox"/>		

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	string	Account GUID		
CASEBUSINESSUNITCODE	string	Project Business Unit Code		
CASECOSTCENTERIDENTIFIER	string	Project Cost Center Identifier		
CASEGUID	string	Project GUID		
CASENAME	string	Project Name		
CASENUMBER	int64	Project Number		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User Company Currency		
DESCRIPTION	string	Ddescription		
ENDTIME	datetime	End Time		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVICEROWGUID	string	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISPRODUCTIVE	boolean	Is Productive		
ISREADONLY	boolean	Is Read-only		
OVERTIMECODE	string	Overtime Code		

Name	Data Type	Label	Required	Documentation
OVERTIMEGUID	string	Overtime GUID		
PHASEGUID	string	Phase GUID		
PRICECURRENCYSHORTFORMOF ASECOMPANYCURRENCY	string	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFI NVOICINGCURRENCY	string	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOF ORGANIZATIONCURRENCY	string	Price Currency Short Form of Organization Currency		
PRICECURRENCYSHORTFORMOFU SERCOMPANYCURRENCY	string	Price Currency Short Form of User Company Currency		
QUANTITY	decimal	Quantity		
STARTTIME	datetime	Start Time		
TASKGUID	string	Task GUID		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURR ENCY	decimal	Unit Cost in Organization Currency		
UNITCOSTINUSERCOMPANYCURR ENCY	decimal	Unit Cost in User Company Currency		
UNITPRICE	decimal	Unit Price		
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		
UNITPRICEINASECOMPANYCURR ENCY	decimal	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURR ENCY	decimal	Unit Price in User Company Currency		
USEINVOICEQUANTITY	boolean			
USERBUSINESSUNITCODE	string	User Business Unit Code		
USERCODE	string	User Code		
USERFIRSTNAME	string	User First Name		
USERGUID	string	User GUID		
USERLASTNAME	string	User Last Name		
WORKTYPECODE	string	Work Type Code		
WORKTYPEGUID	string	Work Type GUID		

3.1.47 AddNewIndustry: Visma Severa Add New Industry

Catalog: Severa

Schema: API

Label: Add New Industry

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	string	<input type="checkbox"/>		
industryInfo_GUID	string	<input type="checkbox"/>		
industryInfo_IsActive	boolean	<input type="checkbox"/>		
industryInfo_Name	string	<input type="checkbox"/>		

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	string	Code		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
NAME	string	Name		

3.1.48 AddNewInvoiceLogEntry: Visma Severa Add New Invoice Log Entry

Catalog: Severa

Schema: API

Label: Add New Invoice Log 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: 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	string	<input type="checkbox"/>		
eventTimeUtc	datetime	<input type="checkbox"/>		
ignoreDuplicates	boolean	<input type="checkbox"/>		
invoiceGuid	string	<input type="checkbox"/>		
source	string	<input type="checkbox"/>		
text	string	<input type="checkbox"/>		

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	string	Dummy	<input checked="" type="checkbox"/>	Dummy column

3.1.49 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.

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	string	<input type="checkbox"/>		
itemInfo_BillingDependencyTaskGUID	string	<input type="checkbox"/>		
itemInfo_BillingSchedule	string	<input type="checkbox"/>		
itemInfo_CaseGUID	string	<input type="checkbox"/>		
itemInfo_CaseNumber	int64	<input type="checkbox"/>		
itemInfo_CostCenterGUID	string	<input type="checkbox"/>		

Name	Data Type	Required	Default Value	Documentation
itemInfo_CostCenterIdentifier	string	<input type="checkbox"/>		
itemInfo_CostCurrencyShortform	string	<input type="checkbox"/>		
itemInfo_CostCurrencyShortformOf InvoicingCurrency	string	<input type="checkbox"/>		
itemInfo_CostCurrencyShortformOf OrganizationCurrency	string	<input type="checkbox"/>		
itemInfo_CostCurrencyShortformOf UserCompanyCurrency	string	<input type="checkbox"/>		
itemInfo_Description	string	<input type="checkbox"/>		
itemInfo_DisplayPeriodStart	datetime	<input type="checkbox"/>		
itemInfo_EventDate	datetime	<input type="checkbox"/>		
itemInfo_GUID	string	<input type="checkbox"/>		
itemInfo_InvoiceDescription	string	<input type="checkbox"/>		
itemInfo_InvoiceGUID	string	<input type="checkbox"/>		
itemInfo_InvoiceQuantity	decimal	<input type="checkbox"/>		
itemInfo_InvoiceRow GUID	string	<input type="checkbox"/>		
itemInfo_IsApproved	boolean	<input type="checkbox"/>		
itemInfo_IsBillable	boolean	<input type="checkbox"/>		
itemInfo_IsOutSourcingCost	boolean	<input type="checkbox"/>		
itemInfo_IsReadOnly	boolean	<input type="checkbox"/>		
itemInfo_IsRealised	boolean	<input type="checkbox"/>		
itemInfo_MeasurementUnit	string	<input type="checkbox"/>		
itemInfo_Name	string	<input type="checkbox"/>		
itemInfo_OrderDate	datetime	<input type="checkbox"/>		
itemInfo_PlannedBillingDate	datetime	<input type="checkbox"/>		
itemInfo_PriceCurrencyShortformOf CaseCompanyCurrency	string	<input type="checkbox"/>		
itemInfo_PriceCurrencyShortformOf InvoicingCurrency	string	<input type="checkbox"/>		
itemInfo_PriceCurrencyShortformOf OrganizationCurrency	string	<input type="checkbox"/>		
itemInfo_PriceCurrencyShortformOf UserCompanyCurrency	string	<input type="checkbox"/>		
itemInfo_ProductCode	string	<input type="checkbox"/>		
itemInfo_ProductGUID	string	<input type="checkbox"/>		
itemInfo_ProductType	string	<input type="checkbox"/>		
itemInfo_PurchaseOrderNumber	string	<input type="checkbox"/>		
itemInfo_PurchaseVAT	decimal	<input type="checkbox"/>		
itemInfo_Quantity	decimal	<input type="checkbox"/>		
itemInfo_RecurrenceEndDate	datetime	<input type="checkbox"/>		
itemInfo_RecurrenceEndType	int32	<input type="checkbox"/>		
itemInfo_RecurrenceFrequency	int32	<input type="checkbox"/>		
itemInfo_RecurrenceStartDate	datetime	<input type="checkbox"/>		

Name	Data Type	Required	Default Value	Documentation
itemInfo_RecurrenceTimes	int32	<input type="checkbox"/>		
itemInfo_RecurringItemGUID	string	<input type="checkbox"/>		
itemInfo_SalesAccountGUID	string	<input type="checkbox"/>		
itemInfo_SalesStatusGUID	string	<input type="checkbox"/>		
itemInfo_TaskGUID	string	<input type="checkbox"/>		
itemInfo_TravelEndTime	datetime	<input type="checkbox"/>		
itemInfo_TravelStartTime	datetime	<input type="checkbox"/>		
itemInfo_UnitCost	decimal	<input type="checkbox"/>		
itemInfo_UnitCostInInvoicingCurrency	decimal	<input type="checkbox"/>		
itemInfo_UnitCostInOrganizationCurrency	decimal	<input type="checkbox"/>		
itemInfo_UnitCostInUserCompanyCurrency	decimal	<input type="checkbox"/>		
itemInfo_UnitCostVatIncludedAmount	decimal	<input type="checkbox"/>		
itemInfo_UnitPrice	decimal	<input type="checkbox"/>		
itemInfo_UnitPriceInBaseCurrency	decimal	<input type="checkbox"/>		
itemInfo_UnitPriceInCaseCompanyCurrency	decimal	<input type="checkbox"/>		
itemInfo_UnitPriceInUserCompanyCurrency	decimal	<input type="checkbox"/>		
itemInfo_UserCode	string	<input type="checkbox"/>		
itemInfo_UserGUID	string	<input type="checkbox"/>		
itemInfo_VAT	decimal	<input type="checkbox"/>		

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	string	Billing Dependency Task GUID		
BILLINGSCHEDULE	string	Billing Schedule		
CASEGUID	string	Project GUID		
CASENUMBER	int64	Project Number		
COSTCENTERGUID	string	Cost Center GUID		
COSTCENTERIDENTIFIER	string	Cost Center Identifier		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing Currency		

Name	Data Type	Label	Required	Documentation
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User Company Currency		
DESCRIPTION	string	Description		
DISPLAYPERIODSTART	datetime	Display Period Start		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVICEROWGUID	string	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	string	Measurement Unit		
NAME	string	Name		
ORDERDATE	datetime	Order Date		
PLANNEDBILLINGDATE	datetime	Planned Billing Date		
PRICECURRENCYSHORTFORMOFPROJECTCOMPANYCURRENCY	string	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Price Currency Short Form of Organization Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Price Currency Short Form of User Company Currency		
PRODUCTCODE	string	Product Code		
PRODUCTGUID	string	Product GUID		
PRODUCTTYPE	string	Product Type		
PURCHASEORDERNUMBER	string	Purchase Order Number		
PURCHASEVAT	decimal	Purchase VAT		
QUANTITY	decimal	Quantity		
RECURRENCEENDDATE	datetime	Recurrence End Date		
RECURRENCEENDTYPE	int32	Recurrence End Type		
RECURRENCFREQUENCY	int32	Recurrence Frequency		

Name	Data Type	Label	Required	Documentation
RECURRENCESTARTDATE	datetime	Recurrence Start Date		
RECURRENCETIMES	int32	Recurrence Times		
RECURRINGITEMGUID	string	Recurring Item GUID		
SALESACCOUNTGUID	string	Sales Account GUID		
SALESSTATUSGUID	string	Sales Status GUID		
TASKGUID	string	Task GUID		
TRAVELENDTIME	datetime	Travel End Time		
TRAVELSTARTTIME	datetime	Travel Start Time		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	decimal	Unit Cost in Organization Currency		
UNITCOSTINUSERCOMPANYCURRENCY	decimal	Unit Cost in User Company Currency		
UNITCOSTVATINCLUDEDAMOUNT	decimal	Unit Cost VAT Included Amount		
UNITPRICE	decimal	Unit Price		
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		
UNITPRICEINPROJECTCOMPANYCURRENCY	decimal	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	decimal	Unit Price in User Company Currency		
USERCODE	string	User Code		
USERGUID	string	User GUID		
VAT	decimal	VAT		

3.1.50 AddNewItem: 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 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
items	string	<input type="checkbox"/>		

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	string	Billing Dependency Task GUID		
BILLINGSCHEDULE	string	Billing Schedule		
CASEGUID	string	Project GUID		
CASENUMBER	int64	Project Number		
COSTCENTERGUID	string	Cost Center GUID		
COSTCENTERIDENTIFIER	string	Cost Center Identifier		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User Company Currency		
DESCRIPTION	string	Ddescription		
DISPLAYPERIODSTART	datetime	Display Period Start		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVICEROWGUID	string	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISOUTSOURCINGCOST	boolean	Is Out-sourcing Cost		

Name	Data Type	Label	Required	Documentation
ISREADONLY	boolean	Is Read-only		
ISREALISED	boolean	Is Realized		
MEASUREMENTUNIT	string	Measurement Unit		
NAME	string	Name		
ORDERDATE	datetime	Order Date		
PLANNEDBILLINGDATE	datetime	Planned Billing Date		
PRICECURRENCYSHORTFORMOF COMPANYCURRENCY	string	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOF INVOICINGCURRENCY	string	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOF ORGANIZATIONCURRENCY	string	Price Currency Short Form of Organization Currency		
PRICECURRENCYSHORTFORMOF USERCOMPANYCURRENCY	string	Price Currency Short Form of User Company Currency		
PRODUCTCODE	string	Product Code		
PRODUCTGUID	string	Product GUID		
PRODUCTTYPE	string	Product Type		
PURCHASEORDERNUMBER	string	Purchase Order Number		
PURCHASEVAT	decimal	Purchase VAT		
QUANTITY	decimal	Quantity		
RECURRENCEENDDATE	datetime	Recurrence End Date		
RECURRENCEENDTYPE	int32	Recurrence End Type		
RECURRENCEFREQUENCY	int32	Recurrence Frequency		
RECURRENCESTARTDATE	datetime	Recurrence Start Date		
RECURRENCETIMES	int32	Recurrence Times		
RECURRINGITEMGUID	string	Recurring Item GUID		
SALESACCOUNTGUID	string	Sales Account GUID		
SALESSTATUSGUID	string	Sales Status GUID		
TASKGUID	string	Task GUID		
TRAVELENDTIME	datetime	Travel End Time		
TRAVELSTARTTIME	datetime	Travel Start Time		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	decimal	Unit Cost in Organization Currency		
UNITCOSTINUSERCOMPANYCURRENCY	decimal	Unit Cost in User Company Currency		
UNITCOSTVATINCLUDEDAMOUNT	decimal	Unit Cost VAT Included Amount		

Name	Data Type	Label	Required	Documentation
UNITPRICE	decimal	Unit Price		
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		
UNITPRICEINCOMPANYCURRENCY	decimal	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	decimal	Unit Price in User Company Currency		
USERCODE	string	User Code		
USERGUID	string	User GUID		
VAT	decimal	VAT		

3.1.51 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	string	<input type="checkbox"/>		
phaseInfo_CaseGUID	string	<input type="checkbox"/>		

Name	Data Type	Required	Default Value	Documentation
phaseInfo_Code	string	<input type="checkbox"/>		
phaseInfo_Deadline	datetime	<input type="checkbox"/>		
phaseInfo_Description	string	<input type="checkbox"/>		
phaseInfo_FlatRate_AdditionalHoursUnitPrice	decimal	<input type="checkbox"/>		
phaseInfo_FlatRate_AreAdditionalHoursBillable	boolean	<input type="checkbox"/>		
phaseInfo_FlatRate_Hours	decimal	<input type="checkbox"/>		
phaseInfo_FlatRate_Price	decimal	<input type="checkbox"/>		
phaseInfo_GUID	string	<input type="checkbox"/>		
phaseInfo_IsCompleted	boolean	<input type="checkbox"/>		
phaseInfo_IsLocked	boolean	<input type="checkbox"/>		
phaseInfo_Name	string	<input type="checkbox"/>		
phaseInfo_OriginalDeadline	datetime	<input type="checkbox"/>		
phaseInfo_OriginalPlannedStartDate	datetime	<input type="checkbox"/>		
phaseInfo_OriginalWorkEstimate	decimal	<input type="checkbox"/>		
phaseInfo_OwnerUserGUID	string	<input type="checkbox"/>		
phaseInfo_ParentPhaseGUID	string	<input type="checkbox"/>		
phaseInfo_PlannedStartDate	datetime	<input type="checkbox"/>		
phaseInfo_PricePerHour	decimal	<input type="checkbox"/>		
phaseInfo_SortOrder	int32	<input type="checkbox"/>		
phaseInfo_WorkEstimate	decimal	<input type="checkbox"/>		
phaseInfo_WorkTypeGUID	string	<input type="checkbox"/>		

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	string	Project GUID		
CODE	string	Code		
DEADLINE	datetime	Deadline		
DESCRIPTION	string	Ddescription		
FLATRATE_ADDITIONALHOURSUNITPRICE	decimal	Flat Rate Additional Hours Unit Price		
FLATRATE_AREADDITIONALHOURSBILLABLE	boolean	Flat Rate Are additional Hours Billable		
FLATRATE_HOURS	decimal	Flat Rate Hours		
FLATRATE_PRICE	decimal	Flat Rate Price		
GUID	string	GUID		

Name	Data Type	Label	Required	Documentation
ISCOMPLETED	boolean	Is Completed		
ISLOCKED	boolean	Is Locked		
NAME	string	Name		
ORIGINALDEADLINE	datetime	Original Deadline		
ORIGINALPLANNEDSTARTDATE	datetime	Original Planned Start Date		
ORIGINALWORKESTIMATE	decimal	Original Work Estimate		
OWNERUSERGUID	string	Owner User GUID		
PARENTPHASEGUID	string	Parent Phase GUID		
PLANNEDSTARTDATE	datetime	Planned Start Date		
PRICEPERHOUR	decimal	Price per Hour		
SORTORDER	int32	Sort Order		
WORKESTIMATE	decimal	Work Estimate		
WORKTYPEGUID	string	Work Type GUID		

3.1.52 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.

Name	Data Type	Required	Default Value	Documentation
productInfo_Code	string	<input type="checkbox"/>		
productInfo_Description	string	<input type="checkbox"/>		
productInfo_GUID	string	<input type="checkbox"/>		
productInfo_IsActive	boolean	<input type="checkbox"/>		
productInfo_MeasurementUnit	string	<input type="checkbox"/>		
productInfo_Name	string	<input type="checkbox"/>		
productInfo_ProductCategoryGUID	string	<input type="checkbox"/>		
productInfo_SalesAccountGUID	string	<input type="checkbox"/>		
productInfo_Type	string	<input type="checkbox"/>		
productInfo_UnitCost	decimal	<input type="checkbox"/>		
productInfo_UnitPrice	decimal	<input type="checkbox"/>		
productInfo_UseInWorkTimeEntry	boolean	<input type="checkbox"/>		
productInfo_VAT	decimal	<input type="checkbox"/>		

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	string	Code		
DESCRIPTION	string	Ddescription		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
MEASUREMENTUNIT	string	Measurement Unit		
NAME	string	Name		
PRODUCTCATEGORY GUID	string	Product Category GUID		
SALESACCOUNTGUID	string	Sales Account GUID		
TYPE	string	Type		
UNITCOST	decimal	Unit Cost		
UNITPRICE	decimal	Unit Price		
USEINWORKTIMEENTRY	boolean	Use in Work Time Entry		
VAT	decimal	VAT		

3.1.53 AddNewProductCategory: Visma Severa Add New Product Category

Catalog: Severa

Schema: API

Label: Add New 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 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	string	<input type="checkbox"/>		
productCategoryInfo_GUID	string	<input type="checkbox"/>		
productCategoryInfo_IsActive	boolean	<input type="checkbox"/>		
productCategoryInfo_IsDefault	boolean	<input type="checkbox"/>		
productCategoryInfo_Name	string	<input type="checkbox"/>		

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	string	Code		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
NAME	string	Name		

3.1.54 AddNewResource: Visma Severa Add New Resource

Catalog: Severa

Schema: API

Label: Add New 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 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	string	<input type="checkbox"/>		
resourceInfo_IsActive	boolean	<input type="checkbox"/>		
resourceInfo_Name	string	<input type="checkbox"/>		

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	string	GUID		
ISACTIVE	boolean	Is Active		
NAME	string	Name		

3.1.55 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-

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	decimal	<input type="checkbox"/>		
resourceAllocationInfo_AllocatedPercentage	int32	<input type="checkbox"/>		
resourceAllocationInfo_CaseGUID	string	<input type="checkbox"/>		
resourceAllocationInfo_EndDate	datetime	<input type="checkbox"/>		
resourceAllocationInfo_GUID	string	<input type="checkbox"/>		
resourceAllocationInfo_PhaseGUID	string	<input type="checkbox"/>		
resourceAllocationInfo_StartDate	datetime	<input type="checkbox"/>		
resourceAllocationInfo_UserGUID	string	<input type="checkbox"/>		

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	decimal	Allocated Hours		
ALLOCATEDPERCENTAGE	int32	Allocated Percentage		
CASEGUID	string	Project GUID		
ENDDATE	datetime	End Date		
GUID	string	GUID		
PHASEGUID	string	Phase GUID		
STARTDATE	datetime	Start Date		
USERGUID	string	User GUID		

3.1.56 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	string	<input type="checkbox"/>		

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	decimal	Allocated Hours		
ALLOCATEDPERCENTAGE	int32	Allocated Percentage		
CASEGUID	string	Project GUID		
ENDDATE	datetime	End Date		
GUID	string	GUID		
PHASEGUID	string	Phase GUID		
STARTDATE	datetime	Start Date		
USERGUID	string	User GUID		

3.1.57 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	int32	<input type="checkbox"/>		
salesProcessInfo_GUID	string	<input type="checkbox"/>		
salesProcessInfo_Icon	string	<input type="checkbox"/>		
salesProcessInfo_IsActive	boolean	<input type="checkbox"/>		
salesProcessInfo_IsAlsoItemStatus	boolean	<input type="checkbox"/>		
salesProcessInfo_IsInProgress	boolean	<input type="checkbox"/>		
salesProcessInfo_IsLost	boolean	<input type="checkbox"/>		
salesProcessInfo_IsOffer	boolean	<input type="checkbox"/>		
salesProcessInfo_IsWon	boolean	<input type="checkbox"/>		
salesProcessInfo_Name	string	<input type="checkbox"/>		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
DEFAULTPROBABILITY	int32	Default Probability		
GUID	string	GUID		
ICON	string	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	string	Name		

3.1.58 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	string	<input type="checkbox"/>		
salesStatusInfo_GUID	string	<input type="checkbox"/>		
salesStatusInfo_SalesProcessGUID	string	<input type="checkbox"/>		
salesStatusInfo_TimeStamp	datetime	<input type="checkbox"/>		

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	string	Project GUID		
GUID	string	GUID		
SALESPROCESSGUID	string	Sales Process GUID		
TIMESTAMP	datetime	Time Stamp		

3.1.59 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_UselnWorkTimeEntry-` `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
<code>travelExpenseInfo_Code</code>	string	<input type="checkbox"/>		
<code>travelExpenseInfo_CostCurrencyGUID</code>	string	<input type="checkbox"/>		

Name	Data Type	Required	Default Value	Documentation
travelExpenseInfo_Description	string	<input type="checkbox"/>		
travelExpenseInfo_ExpenseClass	string	<input type="checkbox"/>		
travelExpenseInfo_GUID	string	<input type="checkbox"/>		
travelExpenseInfo_IsActive	boolean	<input type="checkbox"/>		
travelExpenseInfo_MeasurementUnit	string	<input type="checkbox"/>		
travelExpenseInfo_Name	string	<input type="checkbox"/>		
travelExpenseInfo_Type	string	<input type="checkbox"/>		
travelExpenseInfo_UnitCost	decimal	<input type="checkbox"/>		
travelExpenseInfo_UnitPrice	decimal	<input type="checkbox"/>		
travelExpenseInfo_UselnWorkTimeEntry	boolean	<input type="checkbox"/>		
travelExpenseInfo_VAT	decimal	<input type="checkbox"/>		

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	string	Code		
COSTCURRENCYGUID	string	Cost Currency GUID		
DESCRIPTION	string	Ddescription		
EXPENSECLASS	string	Expense Class		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
MEASUREMENTUNIT	string	Measurement Unit		
NAME	string	Name		
TYPE	string	Type		
UNITCOST	decimal	Unit Cost		
UNITPRICE	decimal	Unit Price		
USEINWORKTIMEENTRY	boolean	Use in Work Time Entry		
VAT	decimal	VAT		

3.1.60 AddNewUser: Visma Severa Add New User

Catalog: Severa

Schema: API

Label: Add New 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- 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	boolean	<input type="checkbox"/>		
userInfo_Addressline	string	<input type="checkbox"/>		
userInfo_BankAccountNumber	string	<input type="checkbox"/>		
userInfo_BusinessUnitCode	string	<input type="checkbox"/>		
userInfo_BusinessUnitGUID	string	<input type="checkbox"/>		
userInfo_BusinessUnitName	string	<input type="checkbox"/>		
userInfo_Code	string	<input type="checkbox"/>		
userInfo_CountryCode	string	<input type="checkbox"/>		
userInfo_CountryGUID	string	<input type="checkbox"/>		
userInfo_CountryRegionGUID	string	<input type="checkbox"/>		
userInfo_CountryRegionName	string	<input type="checkbox"/>		
userInfo_Culture	string	<input type="checkbox"/>		
userInfo_DefaultActivityTypeGUID	string	<input type="checkbox"/>		
userInfo_Email	string	<input type="checkbox"/>		
userInfo_FirstName	string	<input type="checkbox"/>		
userInfo_GUID	string	<input type="checkbox"/>		

Name	Data Type	Required	Default Value	Documentation
userInfo_IsActive	boolean	<input type="checkbox"/>		
userInfo_LanguageCode	string	<input type="checkbox"/>		
userInfo_LanguageGUID	string	<input type="checkbox"/>		
userInfo_LastName	string	<input type="checkbox"/>		
userInfo_LicenseType	string	<input type="checkbox"/>		
userInfo_Notes	string	<input type="checkbox"/>		
userInfo_OrganizationGUID	string	<input type="checkbox"/>		
userInfo_Phone	string	<input type="checkbox"/>		
userInfo_PhotoFileGUID	string	<input type="checkbox"/>		
userInfo_PostalCode	string	<input type="checkbox"/>		
userInfo_PostOffice	string	<input type="checkbox"/>		
userInfo_ProfileGUID	string	<input type="checkbox"/>		
userInfo_Salutation	string	<input type="checkbox"/>		
userInfo_SendDailyMail	boolean	<input type="checkbox"/>		
userInfo_SendWeeklyMail	boolean	<input type="checkbox"/>		
userInfo_SocialSecurityNumber	string	<input type="checkbox"/>		
userInfo_SuperiorUserGUID	string	<input type="checkbox"/>		
userInfo_TimezoneGUID	string	<input type="checkbox"/>		
userInfo_Title	string	<input type="checkbox"/>		
userInfo_WorktypeGUID	string	<input type="checkbox"/>		

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

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

3.1.61 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

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	string	<input type="checkbox"/>		
tagInfo_GUID	string	<input type="checkbox"/>		
tagInfo_IsActive	boolean	<input type="checkbox"/>		
tagInfo_Keyw ord	string	<input type="checkbox"/>		
tagInfo_Weight	int32	<input type="checkbox"/>		
userGUID	string	<input type="checkbox"/>		

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	string	Context		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
KEYWORD	string	Keyw ord		
WEIGHT	int32	Weight		

3.1.62 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.

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	string	<input type="checkbox"/>		
workTypeInfo_GUID	string	<input type="checkbox"/>		
workTypeInfo_IsActive	boolean	<input type="checkbox"/>		
workTypeInfo_IsDefault	boolean	<input type="checkbox"/>		
workTypeInfo_IsProductive	boolean	<input type="checkbox"/>		
workTypeInfo_Name	string	<input type="checkbox"/>		
workTypeInfo_SalesAccountGUID	string	<input type="checkbox"/>		

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	string	Code		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
ISPRODUCTIVE	boolean	Is Productive		
NAME	string	Name		
SALESACCOUNTGUID	string	Sales Account GUID		

3.1.63 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

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	string	<input type="checkbox"/>		

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

3.1.64 AddressesChangedSince: Visma Severa Addresses Changed since

Catalog: Severa

Schema: API

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	string	<input type="checkbox"/>		
options	int32	<input type="checkbox"/>		
startDateTime	datetime	<input type="checkbox"/>		

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	string	Addressline		
ADDRESSLINE2	string	Addressline 2		
ADDRESSLINE3	string	Addressline 3		
CITY	string	City		
COMPANYGUID	string	Company GUID		
COUNTRYCODE	string	Country Code		
COUNTRYGUID	string	Country GUID		
COUNTRYNAME	string	Country Name		
COUNTRYREGIONGUID	string	Country Region GUID		
COUNTRYREGIONNAME	string	Country Region Name		
FAX	string	Fax		
GUID	string	GUID		
ISBILLINGADDRESS	boolean	Is Billing Address		

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

3.1.65 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 Invariantive UniversalSQL driver 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	string	Addressline		
ADDRESSLINE2	string	Addressline 2		
ADDRESSLINE3	string	Addressline 3		
CITY	string	City		
COMPANYGUID	string	Company GUID		
COUNTRYCODE	string	Country Code		
COUNTRYGUID	string	Country GUID		
COUNTRYNAME	string	Country Name		
COUNTRYREGIONGUID	string	Country Region GUID		
COUNTRYREGIONNAME	string	Country Region Name		
FAX	string	Fax		
GUID	string	GUID		
ISBILLINGADDRESS	boolean	Is Billing Address		
ISPOSTALADDRESS	boolean	Is Postal Address		
ISVISITADDRESS	boolean	Is Visit Address		
PHONE	string	Phone		
POSTALCODE	string	Postal Code		

3.1.66 BillingAndExpenseForecastByCaseGUID: Visma Severa Billing and Expense Forecast by Project GUID

Catalog: Severa

Schema: API

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	string	<input type="checkbox"/>		

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	decimal	Billing Forecast		
EXPENSEFORECAST	decimal	Expense Forecast		
GUID	string	GUID		
INHOUSEREVENUEFORECAST	decimal	Inhouse Revenue Forecast		
MONTH	int32	Month		
NOTES	string	Notes		
OUTSOURCINGREVENUEFORECAST	decimal	Outsourcing Revenue Forecast		
YEAR	int32	Year		

3.1.67 BusinessUnitByCode: Visma Severa Business Unit by Code

Catalog: Severa

Schema: API

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	string	<input type="checkbox"/>		

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	string	Code		
COSTCENTERGUID	string	Cost Center GUID		
CURRENCYCODE	string	Currency Code		
CURRENCYGUID	string	Currency GUID		
FORMATTINGCULTURE	string	Formatting Culture		
GUID	string	GUID		
INSERTTS	datetime	Inserted		
ISACTIVE	boolean	Is Active		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
NAME	string	Name		
NEXTINVOICENUMBER	int32	Next Invoice Number		
OVERDUEINTEREST	decimal	Overdue Interest		
PARENTBUSINESSUNITGUID	string	Parent Business Unit GUID		

Name	Data Type	Label	Required	Documentation
PAYMENTTERM	int32	Payment Term		
SHOWLOGOONINVOICES	boolean	Show Logo on Invoices		
SHOWLOGOONPDFREPORTS	boolean	Show Logo on PDF Reports		
TIMEZONEGUID	string	Timezone GUID		
UPDATEDTS	datetime	Updated		
VATNUMBER	string	VAT Number		

3.1.68 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	string	<input type="checkbox"/>		

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	string	Code		

Name	Data Type	Label	Required	Documentation
COSTCENTERGUID	string	Cost Center GUID		
CURRENCY CODE	string	Currency Code		
CURRENCY GUID	string	Currency GUID		
FORMATTINGCULTURE	string	Formatting Culture		
GUID	string	GUID		
INSERTTS	datetime	Inserted		
ISACTIVE	boolean	Is Active		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
NAME	string	Name		
NEXTINVOICENUMBER	int32	Next Invoice Number		
OVERDUEINTEREST	decimal	Overdue Interest		
PARENTBUSINESSUNITGUID	string	Parent Business Unit GUID		
PAYMENTTERM	int32	Payment Term		
SHOWLOGOONINVOICES	boolean	Show Logo on Invoices		
SHOWLOGOONPDFREPORTS	boolean	Show Logo on PDF Reports		
TIMEZONEGUID	string	Timezone GUID		
UPDATEDTS	datetime	Updated		
VATNUMBER	string	VAT Number		

3.1.69 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.

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	int32	<input type="checkbox"/>		
rootBusinessUnitGuid	string	<input type="checkbox"/>		

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	string	Code		
COSTCENTERGUID	string	Cost Center GUID		
CURRENCYCODE	string	Currency Code		
CURRENCYGUID	string	Currency GUID		
FORMATTINGCULTURE	string	Formatting Culture		
GUID	string	GUID		
INSERTTS	datetime	Inserted		
ISACTIVE	boolean	Is Active		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
NAME	string	Name		
NEXTINVOICENUMBER	int32	Next Invoice Number		
OVERDUEINTEREST	decimal	Overdue Interest		
PARENTBUSINESSUNITGUID	string	Parent Business Unit GUID		
PAYMENTTERM	int32	Payment Term		
SHOWLOGOONINVOICES	boolean	Show Logo on Invoices		
SHOWLOGOONPDFREPORTS	boolean	Show Logo on PDF Reports		
TIMEZONEGUID	string	Timezone GUID		
UPDATETS	datetime	Updated		
VATNUMBER	string	VAT Number		

3.1.70 BusinessUnitsChangedSince: Visma Severa Business Units Changed since

Catalog: Severa

Schema: API

Label: Business Units 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 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	string	<input type="checkbox"/>		
options	int32	<input type="checkbox"/>		
startDateTime	datetime	<input type="checkbox"/>		

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	string	Code		
COSTCENTERGUID	string	Cost Center GUID		
CURRENCYCODE	string	Currency Code		
CURRENCYGUID	string	Currency GUID		
FORMATTINGCULTURE	string	Formatting Culture		
GUID	string	GUID		
INSERTTS	datetime	Inserted		
ISACTIVE	boolean	Is Active		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
NAME	string	Name		
NEXTINVOICENUMBER	int32	Next Invoice Number		
OVERDUEINTEREST	decimal	Overdue Interest		
PARENTBUSINESSUNITGUID	string	Parent Business Unit GUID		

Name	Data Type	Label	Required	Documentation
PAYMENTTERM	int32	Payment Term		
SHOWLOGOONINVOICES	boolean	Show Logo on Invoices		
SHOWLOGOONPDFREPORTS	boolean	Show Logo on PDF Reports		
TIMEZONEGUID	string	Timezone GUID		
UPDATEDTS	datetime	Updated		
VATNUMBER	string	VAT Number		

3.1.71 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	string	<input type="checkbox"/>		

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	string	Account GUID		

Name	Data Type	Label	Required	Documentation
ACCOUNTNAME	string	Account Name		
ACCOUNTNUMBER	int64	Account Number		
BILLINGADDRESS	string	Billing Address		
BILLINGADDRESSGUID	string	Billing Address GUID		
BILLINGINTERNALNOTES	string	Billing Internal Notes		
BILLINGNOTESBEFOREGRID	string	Billing Notes before Grid		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
BUSINESSUNITNUMBER	string	Business Unit Number		
CASECOMPANYCURRENCYGUID	string	Project Company Currency GUID		
CASENUMBER	int64	Project Number		
CASEOWNERUSERCODE	string	Project Owner User Code		
CASEOWNERUSERGUID	string	Project Owner User GUID		
CLOSEDDATE	datetime	Closed Date		
COMPLETIONESTIMATE	int32	Completion Estimate		
CONTACTGUID	string	Contact GUID		
COSTCENTERGUID	string	Cost Center GUID		
COSTCENTERNUMBER	string	Cost Center Number		
CURRENCYGUID	string	Currency GUID		
CURRENCYSHORTFORM	string	Currency Short Form		
CURRENCYSHORTFORMOFCASECOMPANYCURRENCY	string	Currency Short Form of Project Company Currency		
CURRENTCASESTATUSDESCRIPTION	string	Current Project Status Description		
CURRENTCASESTATUSGUID	string	Current Project Status GUID		
CURRENTCASESTATUSINSERTTS	datetime	Current Project Status Inserted		
DEADLINEDATE	datetime	Deadline Date		
DESCRIPTION	string	Description		
EXPECTEDVALUE	decimal	Expected Value		
GUID	string	GUID		
INCLUDEEXCEPTIONS	boolean	Include Exceptions		
INCLUDEOVERTIME	boolean	Include Overtime		
INSERTTS	datetime	Inserted		
INTERNALNAME	string	Internal Name		
INVOICETEMPLATEGUID	string	Invoice Template GUID		
ISCLOSED	boolean	Is Closed		

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	int32	Lead Rating		
LEADSOURCEGUID	string	Lead Source GUID		
NAME	string	Name		
ORDERNUMBER	string	Order Number		
OURREFERENCE	string	Our Reference		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTTERM	int32	Payment Term		
PRICELISTGUID	string	Pricelist GUID		
PRINGRULE	string	Pricing Rule		
PRINGTYPE	string	Pricing Type		
PRIORITY	int32	Priority		
PROBABILITY	int32	Probability		
ROOTTASKGUID	string	Root Task GUID		
SALESCLOSEDATE	datetime	Sales Close Date		
SALESPERSONUSERCODE	string	Sales Person User Code		
SALESPERSONUSERGUID	string	Sales Person User GUID		
SALESPROCESSGUID	string	Sales Process GUID		
SALESSTATUSGUID	string	Sales Status GUID		
SHAREDTOEXTRANET	boolean	Shared to Extranet		
STARTDATE	datetime	Start Date		
UPDATEDTS	datetime	Updated		
URLTOCASE	string	URL to Project		
USEDEFAULTPRODUCTSINWORKTIMEENTRY	boolean	Use Default Products in Work Time Entry		
USEDEFAULTWORKTYPES	boolean	Use Default Work Types		
YOURREFERENCE	string	Your Reference		

3.1.72 CaseByNumber: Visma Severa Project by Number

Catalog: Severa

Schema: API

Label: Project 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: 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	int64	<input type="checkbox"/>		

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	string	Account GUID		
ACCOUNTNAME	string	Account Name		
ACCOUNTNUMBER	int64	Account Number		
BILLINGADDRESS	string	Billing Address		
BILLINGADDRESSGUID	string	Billing Address GUID		
BILLINGINTERNALNOTES	string	Billing Internal Notes		
BILLINGNOTESBEFOREGRID	string	Billing Notes before Grid		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
BUSINESSUNITNUMBER	string	Business Unit Number		
CASECOMPANYCURRENCYGUID	string	Project Company Currency GUID		
CASENUMBER	int64	Project Number		
CASEOWNERUSERCODE	string	Project Owner User Code		
CASEOWNERUSERGUID	string	Project Owner User GUID		

Name	Data Type	Label	Required	Documentation
CLOSEDDATE	datetime	Closed Date		
COMPLETIONESTIMATE	int32	Completion Estimate		
CONTACTGUID	string	Contact GUID		
COSTCENTERGUID	string	Cost Center GUID		
COSTCENTERNUMBER	string	Cost Center Number		
CURRENCYGUID	string	Currency GUID		
CURRENCYSHORTFORM	string	Currency Short Form		
CURRENCYSHORTFORMOFCASECOMPANYCURRENCY	string	Currency Short Form of Project Company Currency		
CURRENTCASESTATUSDESCRIPTION	string	Current Project Status Description		
CURRENTCASESTATUSGUID	string	Current Project Status GUID		
CURRENTCASESTATUSINSERTTS	datetime	Current Project Status Inserted		
DEADLINEDATE	datetime	Deadline Date		
DESCRIPTION	string	Description		
EXPECTEDVALUE	decimal	Expected Value		
GUID	string	GUID		
INCLUDEEXCEPTIONS	boolean	Include Exceptions		
INCLUDEOVERTIME	boolean	Include Overtime		
INSERTTS	datetime	Inserted		
INTERNALNAME	string	Internal Name		
INVOICETEMPLATEGUID	string	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	int32	Lead Rating		
LEADSOURCEGUID	string	Lead Source GUID		
NAME	string	Name		
ORDERNUMBER	string	Order Number		
OURREFERENCE	string	Our Reference		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTTERM	int32	Payment Term		
PRICELISTGUID	string	Pricelist GUID		
PRICINGRULE	string	Pricing Rule		
PRICINGTYPE	string	Pricing Type		

Name	Data Type	Label	Required	Documentation
PRIORITY	int32	Priority		
PROBABILITY	int32	Probability		
ROOTTASKGUID	string	Root Task GUID		
SALESCLOSEDATE	datetime	Sales Close Date		
SALESPERSONUSERCODE	string	Sales Person User Code		
SALESPERSONUSERGUID	string	Sales Person User GUID		
SALESPROCESSGUID	string	Sales Process GUID		
SALESSTATUSGUID	string	Sales Status GUID		
SHAREDTOEXTRANET	boolean	Shared to Extranet		
STARTDATE	datetime	Start Date		
UPDATEDTS	datetime	Updated		
URLTOCASE	string	URL to Project		
USEDEFAULTPRODUCTSINWORKTIMEENTRY	boolean	Use Default Products in Work Time Entry		
USEDEFAULTWORKTYPES	boolean	Use Default Work Types		
YOURREFERENCE	string	Your Reference		

3.1.73 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.

Name	Data Type	Required	Default Value	Documentation
taskGuid	string	<input type="checkbox"/>		

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	string	Account GUID		
ACCOUNTNAME	string	Account Name		
ACCOUNTNUMBER	int64	Account Number		
BILLINGADDRESS	string	Billing Address		
BILLINGADDRESSGUID	string	Billing Address GUID		
BILLINGINTERNALNOTES	string	Billing Internal Notes		
BILLINGNOTESBEFOREGRID	string	Billing Notes before Grid		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
BUSINESSUNITNUMBER	string	Business Unit Number		
CASECOMPANYCURRENCYGUID	string	Project Company Currency GUID		
CASENUMBER	int64	Project Number		
CASEOWNERUSERCODE	string	Project Owner User Code		
CASEOWNERUSERGUID	string	Project Owner User GUID		
CLOSEDDATE	datetime	Closed Date		
COMPLETIONESTIMATE	int32	Completion Estimate		
CONTACTGUID	string	Contact GUID		
COSTCENTERGUID	string	Cost Center GUID		
COSTCENTERNUMBER	string	Cost Center Number		
CURRENCYGUID	string	Currency GUID		
CURRENCYSHORTFORM	string	Currency Short Form		
CURRENCYSHORTFORMOFCASECOMPANYCURRENCY	string	Currency Short Form of Project Company Currency		
CURRENTCASESTATUSDESCRIPTION	string	Current Project Status Description		
CURRENTCASESTATUSGUID	string	Current Project Status GUID		
CURRENTCASESTATUSINSERTTS	datetime	Current Project Status Inserted		
DEADLINEDATE	datetime	Deadline Date		
DESCRIPTION	string	Ddescription		
EXPECTEDVALUE	decimal	Expected Value		

Name	Data Type	Label	Required	Documentation
GUID	string	GUID		
INCLUDEEXCEPTIONS	boolean	Include Exceptions		
INCLUDEOVERTIME	boolean	Include Overtime		
INSERTTS	datetime	Inserted		
INTERNALNAME	string	Internal Name		
INVOICETEMPLATEGUID	string	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	int32	Lead Rating		
LEADSOURCEGUID	string	Lead Source GUID		
NAME	string	Name		
ORDERNUMBER	string	Order Number		
OURREFERENCE	string	Our Reference		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTTERM	int32	Payment Term		
PRICELISTGUID	string	Pricelist GUID		
PRICINGRULE	string	Pricing Rule		
PRICINGTYPE	string	Pricing Type		
PRIORITY	int32	Priority		
PROBABILITY	int32	Probability		
ROOTTASKGUID	string	Root Task GUID		
SALESCLOSEDATE	datetime	Sales Close Date		
SALESPERSONUSERCODE	string	Sales Person User Code		
SALESPERSONUSERGUID	string	Sales Person User GUID		
SALESPROCESSGUID	string	Sales Process GUID		
SALESSTATUSGUID	string	Sales Status GUID		
SHAREDTOEXTRANET	boolean	Shared to Extranet		
STARTDATE	datetime	Start Date		
UPDATEDTS	datetime	Updated		
URLTOCASE	string	URL to Project		
USEDEFAULTPRODUCTSINWORKTIMEENTRY	boolean	Use Default Products in Work Time Entry		
USEDEFAULTWORKTYPES	boolean	Use Default Work Types		
YOURREFERENCE	string	Your Reference		

3.1.74 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	string	<input type="checkbox"/>		

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	string	Addressline		
BANKACCOUNTNUMBER	string	Bank Account Number		
BUSINESSUNITCODE	string	Business Unit Code		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
CODE	string	Code		
COUNTRYCODE	string	Country Code		
COUNTRY GUID	string	Country GUID		
COUNTRYREGIONGUID	string	Country Region GUID		
COUNTRYREGIONNAME	string	Country Region Name		

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

3.1.75 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

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	string	<input type="checkbox"/>		
criteria_InternalName	string	<input type="checkbox"/>		
criteria_OnlyActiveCases	boolean	<input type="checkbox"/>		
criteria_OnlyCustomerCases	boolean	<input type="checkbox"/>		
criteria_OnlyInactiveCases	boolean	<input type="checkbox"/>		
criteria_OnlyInternalCases	boolean	<input type="checkbox"/>		
criteria_SalesProcess	string	<input type="checkbox"/>		
criteria_Tags	string	<input type="checkbox"/>		

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	string	Account GUID		
ACCOUNTNAME	string	Account Name		
ACCOUNTNUMBER	int64	Account Number		
BILLINGADDRESS	string	Billing Address		
BILLINGADDRESSGUID	string	Billing Address GUID		
BILLINGINTERNALNOTES	string	Billing Internal Notes		
BILLINGNOTESBEFOREGRID	string	Billing Notes before Grid		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
BUSINESSUNITNUMBER	string	Business Unit Number		
CASECOMPANYCURRENCYGUID	string	Project Company Currency GUID		
CASENUMBER	int64	Project Number		
CASEOWNERUSERCODE	string	Project Owner User Code		

Name	Data Type	Label	Required	Documentation
CASEOWNERUSERGUID	string	Project Owner User GUID		
CLOSEDDATE	datetime	Closed Date		
COMPLETIONESTIMATE	int32	Completion Estimate		
CONTACTGUID	string	Contact GUID		
COSTCENTERGUID	string	Cost Center GUID		
COSTCENTERNUMBER	string	Cost Center Number		
CURRENCYGUID	string	Currency GUID		
CURRENCYSHORTFORM	string	Currency Short Form		
CURRENCYSHORTFORMOFCASECOMPANYCURRENCY	string	Currency Short Form of Project Company Currency		
CURRENTCASESTATUSDESCRIPTION	string	Current Project Status Description		
CURRENTCASESTATUSGUID	string	Current Project Status GUID		
CURRENTCASESTATUSINSERTTS	datetime	Current Project Status Inserted		
DEADLINE DATE	datetime	Deadline Date		
DESCRIPTION	string	Description		
EXPECTEDVALUE	decimal	Expected Value		
GUID	string	GUID		
INCLUDEEXCEPTIONS	boolean	Include Exceptions		
INCLUDEOVERTIME	boolean	Include Overtime		
INSERTTS	datetime	Inserted		
INTERNALNAME	string	Internal Name		
INVOICETEMPLATEGUID	string	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	int32	Lead Rating		
LEADSOURCEGUID	string	Lead Source GUID		
NAME	string	Name		
ORDERNUMBER	string	Order Number		
OURREFERENCE	string	Our Reference		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTTERM	int32	Payment Term		
PRICELISTGUID	string	Pricelist GUID		
PRICINGRULE	string	Pricing Rule		

Name	Data Type	Label	Required	Documentation
PRICINGTYPE	string	Pricing Type		
PRIORITY	int32	Priority		
PROBABILITY	int32	Probability		
ROOTTASKGUID	string	Root Task GUID		
SALESCLOSEDATE	datetime	Sales Close Date		
SALESPERSONUSERCODE	string	Sales Person User Code		
SALESPERSONUSERGUID	string	Sales Person User GUID		
SALESPROCESSGUID	string	Sales Process GUID		
SALESSTATUSGUID	string	Sales Status GUID		
SHAREDTOEXTRANET	boolean	Shared to Extranet		
STARTDATE	datetime	Start Date		
UPDATEDTS	datetime	Updated		
URLTOCASE	string	URL to Project		
USEDEFAULTPRODUCTSINWORKTIMEENTRY	boolean	Use Default Products in Work Time Entry		
USEDEFAULTWORKTYPES	boolean	Use Default Work Types		
YOURREFERENCE	string	Your Reference		

3.1.76 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

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	string	<input type="checkbox"/>		
businessUnitGUID	string	<input type="checkbox"/>		

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	string	Account GUID		
ACCOUNTNAME	string	Account Name		
ACCOUNTNUMBER	int64	Account Number		
BILLINGADDRESS	string	Billing Address		
BILLINGADDRESSGUID	string	Billing Address GUID		
BILLINGINTERNALNOTES	string	Billing Internal Notes		
BILLINGNOTESBEFOREGRID	string	Billing Notes before Grid		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
BUSINESSUNITNUMBER	string	Business Unit Number		
CASECOMPANYCURRENCYGUID	string	Project Company Currency GUID		
CASENUMBER	int64	Project Number		
CASEOWNERUSERCODE	string	Project Owner User Code		
CASEOWNERUSERGUID	string	Project Owner User GUID		
CLOSEDDATE	datetime	Closed Date		
COMPLETIONESTIMATE	int32	Completion Estimate		
CONTACTGUID	string	Contact GUID		
COSTCENTERGUID	string	Cost Center GUID		
COSTCENTERNUMBER	string	Cost Center Number		
CURRENCYGUID	string	Currency GUID		
CURRENCYSHORTFORM	string	Currency Short Form		
CURRENCYSHORTFORMOFCASECOMPANYCURRENCY	string	Currency Short Form of Project Company Currency		
CURRENTCASESTATUSDESCRIPTION	string	Current Project Status Description		
CURRENTCASESTATUSGUID	string	Current Project Status GUID		
CURRENTCASESTATUSINSERTTS	datetime	Current Project Status Inserted		

Name	Data Type	Label	Required	Documentation
DEADLINE DATE	datetime	Deadline Date		
DESCRIPTION	string	Description		
EXPECTED VALUE	decimal	Expected Value		
GUID	string	GUID		
INCLUDE EXCEPTIONS	boolean	Include Exceptions		
INCLUDE OVERTIME	boolean	Include Overtime		
INSERT TS	datetime	Inserted		
INTERNAL NAME	string	Internal Name		
INVOICE TEMPLATE GUID	string	Invoice Template GUID		
IS CLOSED	boolean	Is Closed		
IS DAILY ALLOWANCE BILLABLE	boolean	Is Daily Allowance Billable		
IS INTERNAL	boolean	Is Internal		
IS MILEAGE BILLABLE	boolean	Is Mileage Billable		
IS OTHER TRAVEL EXPENSES BILLABLE	boolean	Are Other Travel Expenses Billable		
LATEST ESTIMATION DATE	datetime	Latest Estimation Date		
LEAD RATING	int32	Lead Rating		
LEAD SOURCE GUID	string	Lead Source GUID		
NAME	string	Name		
ORDER NUMBER	string	Order Number		
OUR REFERENCE	string	Our Reference		
OVERDUE INTEREST	decimal	Overdue Interest		
PAYMENT TERM	int32	Payment Term		
PRICELIST GUID	string	Pricelist GUID		
PRICING RULE	string	Pricing Rule		
PRICING TYPE	string	Pricing Type		
PRIORITY	int32	Priority		
PROBABILITY	int32	Probability		
ROOT TASK GUID	string	Root Task GUID		
SALES CLOSE DATE	datetime	Sales Close Date		
SALES PERSON USER CODE	string	Sales Person User Code		
SALES PERSON USER GUID	string	Sales Person User GUID		
SALES PROCESS GUID	string	Sales Process GUID		
SALES STATUS GUID	string	Sales Status GUID		
SHARED TO EXTRANET	boolean	Shared to Extranet		
START DATE	datetime	Start Date		
UPDATED TS	datetime	Updated		
URL TO CASE	string	URL to Project		
USE DEFAULT PRODUCTS IN WORK TIME ENTRY	boolean	Use Default Products in Work Time Entry		

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

3.1.77 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 Invariantive UniversalSQL driver 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	string	<input type="checkbox"/>		
options	int32	<input type="checkbox"/>		
startDateTime	datetime	<input type="checkbox"/>		

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	string	Account GUID		
ACCOUNTNAME	string	Account Name		
ACCOUNTNUMBER	int64	Account Number		
BILLINGADDRESS	string	Billing Address		

Name	Data Type	Label	Required	Documentation
BILLINGADDRESSGUID	string	Billing Address GUID		
BILLINGINTERNALNOTES	string	Billing Internal Notes		
BILLINGNOTESBEFOREGRID	string	Billing Notes before Grid		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
BUSINESSUNITNUMBER	string	Business Unit Number		
CASECOMPANYCURRENCYGUID	string	Project Company Currency GUID		
CASENUMBER	int64	Project Number		
CASEOWNERUSERCODE	string	Project Owner User Code		
CASEOWNERUSERGUID	string	Project Owner User GUID		
CLOSEDDATE	datetime	Closed Date		
COMPLETIONESTIMATE	int32	Completion Estimate		
CONTACTGUID	string	Contact GUID		
COSTCENTERGUID	string	Cost Center GUID		
COSTCENTERNUMBER	string	Cost Center Number		
CURRENCYGUID	string	Currency GUID		
CURRENCYSHORTFORM	string	Currency Short Form		
CURRENCYSHORTFORMOFCASECOMPANYCURRENCY	string	Currency Short Form of Project Company Currency		
CURRENTCASESTATUSDESCRIPTION	string	Current Project Status Description		
CURRENTCASESTATUSGUID	string	Current Project Status GUID		
CURRENTCASESTATUSINSERTTS	datetime	Current Project Status Inserted		
DEADLINE DATE	datetime	Deadline Date		
DESCRIPTION	string	Description		
EXPECTEDVALUE	decimal	Expected Value		
GUID	string	GUID		
INCLUDEEXCEPTIONS	boolean	Include Exceptions		
INCLUDEOVERTIME	boolean	Include Overtime		
INSERTTS	datetime	Inserted		
INTERNALNAME	string	Internal Name		
INVOICETEMPLATEGUID	string	Invoice Template GUID		
ISCLOSED	boolean	Is Closed		
ISDAILYALLOWANCEBILLABLE	boolean	Is Daily Allowance Billable		
ISINTERNAL	boolean	Is Internal		
ISMILEAGEBILLABLE	boolean	Is Mileage Billable		

Name	Data Type	Label	Required	Documentation
ISOTHERTRAVELXPENSESBILLABLE	boolean	Are Other Travel Expenses Billable		
LATESTESTIMATIONDATE	datetime	Latest Estimation Date		
LEADRATING	int32	Lead Rating		
LEADSOURCEGUID	string	Lead Source GUID		
NAME	string	Name		
ORDERNUMBER	string	Order Number		
OURREFERENCE	string	Our Reference		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTTERM	int32	Payment Term		
PRICELISTGUID	string	Pricelist GUID		
PRINGRULE	string	Pricing Rule		
PRINGTYPE	string	Pricing Type		
PRIORITY	int32	Priority		
PROBABILITY	int32	Probability		
ROOTTASKGUID	string	Root Task GUID		
SALESCLOSEDATE	datetime	Sales Close Date		
SALESPERSONUSERCODE	string	Sales Person User Code		
SALESPERSONUSERGUID	string	Sales Person User GUID		
SALESPROCESSGUID	string	Sales Process GUID		
SALESSTATUSGUID	string	Sales Status GUID		
SHAREDTOEXTRANET	boolean	Shared to Extranet		
STARTDATE	datetime	Start Date		
UPDATEDTS	datetime	Updated		
URLTOCASE	string	URL to Project		
USEDEFAULTPRODUCTSINWORKTIMEENTRY	boolean	Use Default Products in Work Time Entry		
USEDEFAULTWORKTYPES	boolean	Use Default Work Types		
YOURREFERENCE	string	Your Reference		

3.1.78 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

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	string	<input type="checkbox"/>		

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	string	GUID		
ICON	string	Icon		
ISACTIVE	boolean	Is Active		
NAME	string	Name		

3.1.79 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	string	GUID		
ICON	string	Icon		

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

3.1.80 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	string	Context		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
KEYWORD	string	Keyw ord		
WEIGHT	int32	Weight		

3.1.81 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.

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	string	<input type="checkbox"/>		

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	int32	Annual Revenue		
ANNUALREVENUE2	int64			
EMAIL	string	Email		
EMPLOYEES	int32	Employees		
GUID	string	GUID		
HEADOFFICEADDRESSGUID	string	Head Office Address GUID		
HIERARCHYGUID	string	Hierarchy GUID		
INDUSTRYGUID	string	Industry GUID		
LANGUAGEGUID	string	Language GUID		
NAME	string	Name		
TIMEZONEGUID	string	Timezone GUID		
VATNUMBER	string	VAT Number		
WEBSITE	string	Website		

3.1.82 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-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	string	<input type="checkbox"/>		

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	string	Account GUID		
ADDRESSGUID	string	Address GUID		
COMMUNICATIONMETHODS	string			
CONTACTROLEGUID	string	Contact Role GUID		
DATEOFBIRTH	datetime	Date of Birth		
DESCRIPTION	string	Ddescription		
FIRSTNAME	string	First Name		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
JOBTITLE	string	Job Title		
KEYWORDS	string			
LANGUAGEGUID	string	Language GUID		
LASTNAME	string	Last Name		
SALUTATION	string	Salutation		
SATISFACTIONLEVEL	int32	Satisfaction Level		
TIMEZONEGUID	string	Timezone GUID		

3.1.83 ContactsByAccountGUID: Visma Severa Contacts by Account GUID

Catalog: Severa

Schema: API

Label: Contacts 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 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	string	<input type="checkbox"/>		

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	string	Account GUID		
ADDRESSGUID	string	Address GUID		
COMMUNICATIONMETHODS	string			
CONTACTROLEGUID	string	Contact Role GUID		
DATEOFBIRTH	datetime	Date of Birth		
DESCRIPTION	string	Ddescription		
FIRSTNAME	string	First Name		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
JOBTITLE	string	Job Title		
KEYWORDS	string			
LANGUAGEGUID	string	Language GUID		
LASTNAME	string	Last Name		
SALUTATION	string	Salutation		
SATISFACTIONLEVEL	int32	Satisfaction Level		
TIMEZONEGUID	string	Timezone GUID		

3.1.84 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 Invariantive UniversalSQL driver 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	string	<input type="checkbox"/>		
value	string	<input type="checkbox"/>		

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	string	Account GUID		
ADDRESSGUID	string	Address GUID		
COMMUNICATIONMETHODS	string			
CONTACTROLEGUID	string	Contact Role GUID		
DATEOFBIRTH	datetime	Date of Birth		
DESCRIPTION	string	Ddescription		
FIRSTNAME	string	First Name		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
JOBTITLE	string	Job Title		

Name	Data Type	Label	Required	Documentation
KEYWORDS	string			
LANGUAGEGUID	string	Language GUID		
LASTNAME	string	Last Name		
SALUTATION	string	Salutation		
SATISFACTIONLEVEL	int32	Satisfaction Level		
TIMEZONEGUID	string	Timezone GUID		

3.1.85 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	string	<input type="checkbox"/>		
options	int32	<input type="checkbox"/>		
startDateTime	datetime	<input type="checkbox"/>		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	string	Account GUID		

Name	Data Type	Label	Required	Documentation
ADDRESSGUID	string	Address GUID		
COMMUNICATIONMETHODS	string			
CONTACTROLEGUID	string	Contact Role GUID		
DATEOFBIRTH	datetime	Date of Birth		
DESCRIPTION	string	Ddescription		
FIRSTNAME	string	First Name		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
JOBTITLE	string	Job Title		
KEYWORDS	string			
LANGUAGEGUID	string	Language GUID		
LASTNAME	string	Last Name		
SALUTATION	string	Salutation		
SATISFACTIONLEVEL	int32	Satisfaction Level		
TIMEZONEGUID	string	Timezone GUID		

3.1.86 CostCenterByGUID: Visma Severa Cost Center by GUID

Catalog: Severa

Schema: API

Label: Cost Center 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.

Name	Data Type	Required	Default Value	Documentation
costCenterGuid	string	<input type="checkbox"/>		

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	string	GUID		
IDENTIFIER	string	Identifier		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
NAME	string	Name		

3.1.87 CostCenters: Visma Severa Cost Centers

Catalog: Severa

Schema: API

Label: Cost Centers

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	string	GUID		
IDENTIFIER	string	Identifier		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
NAME	string	Name		

3.1.88 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.

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	string	<input type="checkbox"/>		

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	string	Default Language Code		
DEFAULTLANGUAGEGUID	string	Default Language GUID		
GUID	string	GUID		
ISEU	boolean	Is EU		
ISOCODEA2	string	ISO Code A2		
ISOCODEA3	string	ISO Code A3		
NAME	string	Name		
TIMEZONEGUID	string	Timezone GUID		
TRANSLATEDNAME	string	Translated Name		

3.1.89 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.

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	string	<input type="checkbox"/>		

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	string	Default Language Code		
DEFAULTLANGUAGEGUID	string	Default Language GUID		
GUID	string	GUID		
ISEU	boolean	Is EU		
ISOCODEA2	string	ISO Code A2		
ISOCODEA3	string	ISO Code A3		
NAME	string	Name		
TIMEZONEGUID	string	Timezone GUID		
TRANSLATEDNAME	string	Translated Name		

3.1.90 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 Invariant UniversalSQL driver 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: `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	string	<input type="checkbox"/>		

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	string	Default Language Code		
DEFAULTLANGUAGEGUID	string	Default Language GUID		
GUID	string	GUID		
ISEU	boolean	Is EU		
ISOCODEA2	string	ISO Code A2		
ISOCODEA3	string	ISO Code A3		
NAME	string	Name		
TIMEZONEGUID	string	Timezone GUID		
TRANSLATEDNAME	string	Translated Name		

3.1.91 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.

Filters on API Server: invoiceInformation_Date- invoiceInformation_DueDate- invoiceInformation_InvoiceStatus- invoiceInformation_ItemGUIDs- invoiceInformation_LanguageGUID- invoiceInformation_Notes-

invoiceInformation_OurReference- invoiceInformation_OverDuelInterest-
 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	datetime	<input type="checkbox"/>		
invoiceInformation_DueDate	datetime	<input type="checkbox"/>		
invoiceInformation_InvoiceStatus	string	<input type="checkbox"/>		
invoiceInformation_ItemGUIDs	string	<input type="checkbox"/>		
invoiceInformation_LanguageGUID	string	<input type="checkbox"/>		
invoiceInformation_Notes	string	<input type="checkbox"/>		
invoiceInformation_OurReference	string	<input type="checkbox"/>		
invoiceInformation_OverDuelInterest	decimal	<input type="checkbox"/>		
invoiceInformation_PaymentTerm	int32	<input type="checkbox"/>		
invoiceInformation_ReceiverAddressline	string	<input type="checkbox"/>		
invoiceInformation_ReceiverCity	string	<input type="checkbox"/>		
invoiceInformation_ReceiverContactEmail	string	<input type="checkbox"/>		
invoiceInformation_ReceiverContactName	string	<input type="checkbox"/>		
invoiceInformation_ReceiverCountry	string	<input type="checkbox"/>		
invoiceInformation_ReceiverCustomerName	string	<input type="checkbox"/>		

Name	Data Type	Required	Default Value	Documentation
invoiceInformation_ReceiverPostalCode	string	<input type="checkbox"/>		
invoiceInformation_ReceiverState	string	<input type="checkbox"/>		
invoiceInformation_ReceiverVatNumber	string	<input type="checkbox"/>		
invoiceInformation_SenderAddressline	string	<input type="checkbox"/>		
invoiceInformation_SenderCity	string	<input type="checkbox"/>		
invoiceInformation_SenderContactName	string	<input type="checkbox"/>		
invoiceInformation_SenderCountry	string	<input type="checkbox"/>		
invoiceInformation_SenderName	string	<input type="checkbox"/>		
invoiceInformation_SenderPostalCode	string	<input type="checkbox"/>		
invoiceInformation_SenderState	string	<input type="checkbox"/>		
invoiceInformation_SenderVatNumber	string	<input type="checkbox"/>		
invoiceInformation_YourReference	string	<input type="checkbox"/>		

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	string	Account GUID		
ACCOUNTNAME	string	Account Name		
ACCOUNTNUMBER	int64	Account Number		
BUSINESSUNITCODE	string	Business Unit Code		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
BUYERORGANIZATION_BUYERCONTACTPERSONNAME	string			
BUYERORGANIZATION_BUYERELECTRONICINVOICEADDRESS	string			
BUYERORGANIZATION_BUYERELECTRONICINVOICEOPERATOR	string			
BUYERORGANIZATION_BUYEREMAILADDRESSIDENTIFIER	string			
BUYERORGANIZATION_BUYERORGANISATIONNAME	string			
BUYERORGANIZATION_BUYERORGANISATIONTAXCODE	string			
BUYERORGANIZATION_BUYERPARTYIDENTIFIER	string			

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

Name	Data Type	Label	Required	Documentation
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	string			
INVOICECONFIG_TRAVELPRIMARYGROUPBY	string			
INVOICECONFIG_TRAVELSECONDARYGROUPBY	string			
INVOICEDETAILS_INVOICEDATE	datetime			
INVOICEDETAILS_INVOICEDUEDATE	datetime			
INVOICEDETAILS_INVOICENUMBER	int32			
INVOICEDETAILS_INVOICETOTALVATA MOUNT	decimal			
INVOICEDETAILS_INVOICETOTALVATEXCLUDEDAMOUNT	decimal			
INVOICEDETAILS_INVOICETOTALVATINCLUDEDAMOUNT	decimal			
INVOICEDETAILS_INVOICETYPETEXT	string			
INVOICEDETAILS_ORDERIDENTIFIER	string			
INVOICEDETAILS_ORIGINC CODE	string			
INVOICEDETAILS_PAYMENTOVERDUDEFINEFREETEXT	string			
INVOICEDETAILS_PAYMENTOVERDUDEFINEPERCENT	decimal			
INVOICEDETAILS_PAYMENTTERMSFREETEXT	string			
INVOICEDETAILS_SELLERREFERENCEIDENTIFIER	string			
INVOICEDETAILS_VATBASEAMOUNT	decimal			
INVOICEDETAILS_VATRATEAMOUNT	decimal			

Name	Data Type	Label	Required	Documentation
INVOICEDETAILS_VATRATEPERCENT	int32			
INVICENUMBERINGBUSINESSUNITGUID	string	Invoice Numbering Business Unit GUID		
INVOICEROWS	string			
INVOICESTATUS	string	Invoice Status		
ISCREDITNOTE	boolean	Is Credit Note		
ISREIMBURSED	boolean	Is Reimbursed		
ISTAXFREE	boolean	Is Tax Free		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
MAINCASEGUID	string	Main Case GUID		
NOTES	string	Notes		
NOTESAFTERINVOICEROWS	string	Notes after Invoice Lines		
NOTESBEFOREINVOICEROWS	string	Notes before Invoice Lines		
NUMBER	int32	Number		
ORDERNUMBER	string	Order Number		
OURREFERENCE	string	Our Reference		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTDATE	datetime	Payment Date		
PAYMENTSTATUS	string	Payment Status		
PAYMENTTERM	int32	Payment Term		
PONUMBER	string	PO Number		
RECEIVERADDRESSLINE	string	Receiver Addressline		
RECEIVERCITY	string	Receiver City		
RECEIVERCONTACTEMAIL	string	Receiver Contact Email		
RECEIVERCONTACTNAME	string	Receiver Contact Name		
RECEIVERCOUNTRY	string	Receiver Country		
RECEIVERCUSTOMERNAME	string	Receiver Customer Name		
RECEIVERPOSTALCODE	string	Receiver Postal Code		
RECEIVERSTATE	string	Receiver State		
RECEIVERVATNUMBER	string	Receiver VAT Number		
REFERENCENUMBER	string	Reference Number		
REIMBURSEINVOICEGUID	string	Reimburse Invoice GUID		
SELLERCONTACT_SELLERACCOUNTDETAILS	string			
SELLERCONTACT_SELLERCONTACTPERSONNAME	string			

Name	Data Type	Label	Required	Documentation
SELLERCONTACT_SELLEREMAILADDRESSIDENTIFIER	string			
SELLERCONTACT_SELLERFAXNUMBER	string			
SELLERCONTACT_SELLERFREEXT	string			
SELLERCONTACT_SELLERHOMETOWNNAME	string			
SELLERCONTACT_SELLERPHONE NUMBER	string			
SELLERCONTACT_SELLERWEBADDRESSIDENTIFIER	string			
SELLERORGANIZATION_SELLERINVOICEADDRESS	string			
SELLERORGANIZATION_SELLERINVOICEOPERATOR	string			
SELLERORGANIZATION_SELLERORGANISATIONNAME	string			
SELLERORGANIZATION_SELLERORGANISATIONTAXCODE	string			
SELLERORGANIZATION_SELLERPARTYIDENTIFIER	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRY	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRYCODE	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_STREETNAME	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_TOWNNAME	string			
SENDERADDRESSLINE	string	Sender Addressline		
SENDERCITY	string	Sender City		
SENDERCONTACTNAME	string	Sender Contact Name		
SENDERCOUNTRY	string	Sender Country		
SENDERNAME	string	Sender Name		
SENDERPOSTALCODE	string	Sender Postal Code		
SENDERSTATE	string	Sender State		
SENDERVATNUMBER	string	Sender VAT Number		
TAXFREEDESCRIPTION	string	Tax Free Description		
TITLE	string	Title		
VIRTUALBANKBARCODE	string	Virtual Bank Barcode		

Name	Data Type	Label	Required	Documentation
YOURREFERENCE	string	Your Reference		

3.1.92 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	string	GUID		
ISACTIVE	boolean	Is Active		
ISOCODE	string	ISO Code		
RATE	decimal	Rate		
SYMBOL	string	Symbol		

3.1.93 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 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	string	<input type="checkbox"/>		

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	string	GUID		
ISACTIVE	boolean	Is Active		
ISOCODE	string	ISO Code		
RATE	decimal	Rate		
SYMBOL	string	Symbol		

3.1.94 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 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	string	<input type="checkbox"/>		

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	string	GUID		
ISACTIVE	boolean	Is Active		
ISOCODE	string	ISO Code		
RATE	decimal	Rate		
SYMBOL	string	Symbol		

3.1.95 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	string	<input type="checkbox"/>		

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	string			
ACCOUNTOWNERUSERGUID	string	Account Owner User GUID		
ACCOUNTRATING	int32	Account Rating		
ANNUALREVENUE	int32	Annual Revenue		
ANNUALREVENUE2	int64			
COMPANYGUID	string	Company GUID		
CURRENCYCODE	string	Currency Code		
CURRENCYGUID	string	Currency GUID		
EINVOICEADDRESS	string	e-invoice Address		
EINVOICEOPERATOR	string	e-invoice Operator		
EMAIL	string	Email		
EMPLOYEES	int32	Employees		
GUID	string	GUID		
HEADOFFICEADDRESSGUID	string	Head Office Address GUID		
INDUSTRYGUID	string	Industry GUID		
INSERTTS	datetime	Inserted		
INVOICINGVAT	decimal	Invoicing VAT		
ISACTIVE	boolean	Is Active		
ISINTERNAL	boolean	Is Internal		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
NAME	string	Name		
NOTES	string	Notes		
NUMBER	int64	Number		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTTERM	int32	Payment Term		
PRICELISTGUID	string	Pricelist GUID		
REVERSECHARGE	boolean	Reverse Charge		
REVERSECHARGEDESCRIPTION	string	Reverse Charge Description		
TIMEZONEGUID	string	Timezone GUID		
UPDATEDTS	datetime	Updated		
URLTOACCOUNT	string	URL to Account		
VATNUMBER	string	VAT Number		
WEBSITE	string	Website		

3.1.96 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	int32	<input type="checkbox"/>		
criteria_AccountGroupGuids	string	<input type="checkbox"/>		
options	int32	<input type="checkbox"/>		
startFromIndex	int32	<input type="checkbox"/>		

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	string			
ACCOUNTOWNERUSERGUID	string	Account Owner User GUID		
ACCOUNTRATING	int32	Account Rating		
ANNUALREVENUE	int32	Annual Revenue		
ANNUALREVENUE2	int64			
COMPANYGUID	string	Company GUID		
CURRENCYCODE	string	Currency Code		
CURRENCYGUID	string	Currency GUID		
EINVOICEADDRESS	string	e-invoice Address		
EINVOICEOPERATOR	string	e-invoice Operator		
EMAIL	string	Email		

Name	Data Type	Label	Required	Documentation
EMPLOYEES	int32	Employees		
GUID	string	GUID		
HEADOFFICEADDRESSGUID	string	Head Office Address GUID		
INDUSTRYGUID	string	Industry GUID		
INSERTTS	datetime	Inserted		
INVOICINGVAT	decimal	Invoicing VAT		
ISACTIVE	boolean	Is Active		
ISINTERNAL	boolean	Is Internal		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
NAME	string	Name		
NOTES	string	Notes		
NUMBER	int64	Number		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTTERM	int32	Payment Term		
PRICELISTGUID	string	Pricelist GUID		
REVERSECHARGE	boolean	Reverse Charge		
REVERSECHARGEDESCRIPTION	string	Reverse Charge Description		
TIMEZONEGUID	string	Timezone GUID		
UPDATETS	datetime	Updated		
URLTOACCOUNT	string	URL to Account		
VATNUMBER	string	VAT Number		
WEBSITE	string	Website		

3.1.97 CustomersChangedSince: Visma Severa Customers Changed since

Catalog: Severa

Schema: API

Label: Customers 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 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	int32	<input type="checkbox"/>		
startDateTime	datetime	<input type="checkbox"/>		

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	string			
ACCOUNTOWNERUSERGUID	string	Account Owner User GUID		
ACCOUNTRATING	int32	Account Rating		
ANNUALREVENUE	int32	Annual Revenue		
ANNUALREVENUE2	int64			
COMPANYGUID	string	Company GUID		
CURRENCYCODE	string	Currency Code		
CURRENCYGUID	string	Currency GUID		
EINVOICEADDRESS	string	e-invoice Address		
EINVOICEOPERATOR	string	e-invoice Operator		
EMAIL	string	Email		
EMPLOYEES	int32	Employees		
GUID	string	GUID		
HEADOFFICEADDRESSGUID	string	Head Office Address GUID		
INDUSTRYGUID	string	Industry GUID		
INSERTTS	datetime	Inserted		
INVOICINGVAT	decimal	Invoicing VAT		
ISACTIVE	boolean	Is Active		
ISINTERNAL	boolean	Is Internal		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
NAME	string	Name		
NOTES	string	Notes		
NUMBER	int64	Number		

Name	Data Type	Label	Required	Documentation
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTTERM	int32	Payment Term		
PRICELISTGUID	string	Pricelist GUID		
REVERSECHARGE	boolean	Reverse Charge		
REVERSECHARGEDESCRIPTION	string	Reverse Charge Description		
TIMEZONEGUID	string	Timezone GUID		
UPDATEDTS	datetime	Updated		
URLTOACCOUNT	string	URL to Account		
VATNUMBER	string	VAT Number		
WEBSITE	string	Website		

3.1.98 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	string			

3.1.99 DeleteAccessRightsProfileByGUID: Visma Severa Delete Access Rights Profile by GUID

Catalog: Severa

Schema: API

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	string	<input type="checkbox"/>		

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	string	Dummy	<input checked="" type="checkbox"/>	Dummy column

3.1.100 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 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	string	<input type="checkbox"/>		

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			

3.1.101 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 Invariant UniversalSQL driver 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 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	string	<input type="checkbox"/>		
replacementActivityTypeGUID	string	<input type="checkbox"/>		

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
DELETEACTIVITYTYPEPERESULT	boolean			

3.1.102 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	string	<input type="checkbox"/>		

Columns of Table Function

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			

3.1.103 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	string	<input type="checkbox"/>		
replacementCaseStatusGUID	string	<input type="checkbox"/>		

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			

3.1.104 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	string	<input type="checkbox"/>		

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			

3.1.105 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

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	string	<input type="checkbox"/>		

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			

3.1.106 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 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	string	<input type="checkbox"/>		

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			

3.1.107 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	string	<input type="checkbox"/>		

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			

3.1.108 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	string	<input type="checkbox"/>		
phaseGUIDToTransferItemsTo	string	<input type="checkbox"/>		

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			

3.1.109 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	string	<input type="checkbox"/>		

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			

3.1.110 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

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	string	<input type="checkbox"/>		

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
DELETEPRODUCTCATEGORYRESU LT	boolean			

3.1.111 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 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	string	<input type="checkbox"/>		

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			

3.1.112 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	string	<input type="checkbox"/>		

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			

3.1.113 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	string	<input type="checkbox"/>		

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			

3.1.114 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	string	<input type="checkbox"/>		

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			

3.1.115 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

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	string	<input type="checkbox"/>		

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			

3.1.116 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 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	string	<input type="checkbox"/>		

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	string	Company Business Unit Code		
COMPANYBUSINESSUNITGUID	string	Company Business Unit GUID		
COMPANYBUSINESSUNITNAME	string	Company Business Unit Name		
DAILYHOURS	decimal	Daily Hours		
ENDDATE	datetime	End Date		
GUID	string	GUID		
HOURLCOST	decimal	Hour Cost		
ISOVERTIMEALLOWED	boolean	Is Overtime Allowed		
STARTDATE	datetime	Start Date		
TITLE	string	Title		

3.1.117 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.

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	string	<input type="checkbox"/>		

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	string	Company Business Unit Code		
COMPANYBUSINESSUNITGUID	string	Company Business Unit GUID		
COMPANYBUSINESSUNITNAME	string	Company Business Unit Name		
DAILYHOURS	decimal	Daily Hours		
ENDDATE	datetime	End Date		
GUID	string	GUID		
HOURLCOST	decimal	Hour Cost		
ISOVERTIMEALLOWED	boolean	Is Overtime Allowed		
STARTDATE	datetime	Start Date		
TITLE	string	Title		

3.1.118 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-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	string	<input type="checkbox"/>		

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			

3.1.119 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	string			

3.1.120 ExtranetCaseByCaseGUID: Visma Severa Extranet Project by Project GUID

Catalog: Severa

Schema: API

Label: Extranet Project 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 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	string	<input type="checkbox"/>		

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	string			
GUID	string	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			

3.1.121 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	string	<input type="checkbox"/>		

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	string	Project GUID		
CASERIGHTS_EDITCASESTATUS	boolean			
CASERIGHTS_SHOWBILLING	boolean			
CASERIGHTS_SHOWCASESTATUS	boolean			
CASERIGHTS_SHOWFILESANDLINKS	boolean			
CASERIGHTS_SHOWGANTTCHAR T	boolean			
CASERIGHTS_SHOWPHASETREE	boolean			
CASERIGHTS_SHOWREADYTOBILL	boolean			
CASERIGHTS_SHOWWORKHOUR ESTIMATES	boolean			
CASERIGHTS_SHOWWORKHOURS	boolean			
CASERIGHTS_SHOWWORKHOUR STARTANDENDTIME	boolean			
CONTACTRIGHTS_APPROVAL	boolean			
CONTACTRIGHTS_COMMENTS	boolean			
CONTACTRIGHTS_FILEUPLOAD	boolean			

3.1.122 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	string	<input type="checkbox"/>		

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	string	Content Type		
DESCRIPTION	string	Ddescription		
GUID	string	GUID		
NAME	string	Name		
SIZE	int32	Size		

3.1.123 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	string	<input type="checkbox"/>		

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	byte[]			

3.1.124 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	string	Content Type		
DESCRIPTION	string	Ddescription		

Name	Data Type	Label	Required	Documentation
GUID	string	GUID		
NAME	string	Name		
SIZE	int32	Size		

3.1.125 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	string	<input type="checkbox"/>		
options	int32	<input type="checkbox"/>		
status	string	<input type="checkbox"/>		

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	string	Financial Invoice		
GUID	string	GUID		

3.1.126 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: fininvoiceVersion- 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	string	<input type="checkbox"/>		
fininvoiceVersion	string	<input type="checkbox"/>		
invoiceDateEnd	datetime	<input type="checkbox"/>		
invoiceDateStart	datetime	<input type="checkbox"/>		
invoiceStatus	string	<input type="checkbox"/>		
options	int32	<input type="checkbox"/>		

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	string	Financial Invoice		
GUID	string	GUID		

3.1.127 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 Invariantive UniversalSQL driver 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	string	<input type="checkbox"/>		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ADDRESSLINE	string	Addressline		
ADDRESSLINE2	string	Addressline 2		
ADDRESSLINE3	string	Addressline 3		
CITY	string	City		
COMPANYGUID	string	Company GUID		
COUNTRYCODE	string	Country Code		
COUNTRYGUID	string	Country GUID		
COUNTRYNAME	string	Country Name		
COUNTRYREGIONGUID	string	Country Region GUID		
COUNTRYREGIONNAME	string	Country Region Name		
FAX	string	Fax		
GUID	string	GUID		

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

3.1.128 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.

Name	Data Type	Required	Default Value	Documentation
accountGUID	string	<input type="checkbox"/>		

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	string	Addressline		
ADDRESSLINE2	string	Addressline 2		
ADDRESSLINE3	string	Addressline 3		

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

3.1.129 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.

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	string	<input type="checkbox"/>		
endDate	datetime	<input type="checkbox"/>		

Name	Data Type	Required	Default Value	Documentation
startDate	datetime	<input type="checkbox"/>		

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	string	Account GUID		
CASEBUSINESSUNITCODE	string	Project Business Unit Code		
CASECOSTCENTERIDENTIFIER	string	Project Cost Center Identifier		
CASEGUID	string	Project GUID		
CASENAME	string	Project Name		
CASENUMBER	int64	Project Number		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User Company Currency		
DESCRIPTION	string	Ddescription		
ENDTIME	datetime	End Time		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVICEROWGUID	string	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISPRODUCTIVE	boolean	Is Productive		
ISREADONLY	boolean	Is Read-only		
OVERTIMECODE	string	Overtime Code		
OVERTIMEGUID	string	Overtime GUID		
PHASEGUID	string	Phase GUID		
PRICECURRENCYSHORTFORMOFPROJECTCOMPANYCURRENCY	string	Price Currency Short Form of Project Company Currency		

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

3.1.130 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	string	<input type="checkbox"/>		
enddate	datetime	<input type="checkbox"/>		
startdate	datetime	<input type="checkbox"/>		

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	string	Account GUID		
CASEBUSINESSUNITCODE	string	Project Business Unit Code		
CASECOSTCENTERIDENTIFIER	string	Project Cost Center Identifier		
CASEGUID	string	Project GUID		
CASENAME	string	Project Name		
CASENUMBER	int64	Project Number		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User Company Currency		
DESCRIPTION	string	Ddescription		
ENDTIME	datetime	End Time		

Name	Data Type	Label	Required	Documentation
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVICEROWGUID	string	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISPRODUCTIVE	boolean	Is Productive		
ISREADONLY	boolean	Is Read-only		
OVERTIMECODE	string	Overtime Code		
OVERTIMEGUID	string	Overtime GUID		
PHASEGUID	string	Phase GUID		
PRICECURRENCYSHORTFORMOF ASECOMPANYCURRENCY	string	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOF INVOICINGCURRENCY	string	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOF ORGANIZATIONCURRENCY	string	Price Currency Short Form of Organization Currency		
PRICECURRENCYSHORTFORMOF USERCOMPANYCURRENCY	string	Price Currency Short Form of User Company Currency		
QUANTITY	decimal	Quantity		
STARTTIME	datetime	Start Time		
TASKGUID	string	Task GUID		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	decimal	Unit Cost in Organization Currency		
UNITCOSTINUSERCOMPANYCURRENCY	decimal	Unit Cost in User Company Currency		
UNITPRICE	decimal	Unit Price		
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		
UNITPRICEINPROJECTCOMPANYCURRENCY	decimal	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	decimal	Unit Price in User Company Currency		
USEINVOICEQUANTITY	boolean			
USERBUSINESSUNITCODE	string	User Business Unit Code		
USERCODE	string	User Code		

Name	Data Type	Label	Required	Documentation
USERFIRSTNAME	string	User First Name		
USERGUID	string	User GUID		
USERLASTNAME	string	User Last Name		
WORKTYPECODE	string	Work Type Code		
WORKTYPEGUID	string	Work Type GUID		

3.1.131 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	string	<input type="checkbox"/>		
enddate	datetime	<input type="checkbox"/>		
startdate	datetime	<input type="checkbox"/>		
userGUID	string	<input type="checkbox"/>		

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	string	Account GUID		

Name	Data Type	Label	Required	Documentation
CASEBUSINESSUNITCODE	string	Project Business Unit Code		
CASECOSTCENTERIDENTIFIER	string	Project Cost Center Identifier		
CASEGUID	string	Project GUID		
CASENAME	string	Project Name		
CASENUMBER	int64	Project Number		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User Company Currency		
DESCRIPTION	string	Ddescription		
ENDTIME	datetime	End Time		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVICEROWGUID	string	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISPRODUCTIVE	boolean	Is Productive		
ISREADONLY	boolean	Is Read-only		
OVERTIMECODE	string	Overtime Code		
OVERTIMEGUID	string	Overtime GUID		
PHASEGUID	string	Phase GUID		
PRICECURRENCYSHORTFORMOFPROJECTCOMPANYCURRENCY	string	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Price Currency Short Form of Organization Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Price Currency Short Form of User Company Currency		
QUANTITY	decimal	Quantity		
STARTTIME	datetime	Start Time		

Name	Data Type	Label	Required	Documentation
TASKGUID	string	Task GUID		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	decimal	Unit Cost in Organization Currency		
UNITCOSTINUSERCOMPANYCURRENCY	decimal	Unit Cost in User Company Currency		
UNITPRICE	decimal	Unit Price		
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		
UNITPRICEINPROJECTCOMPANYCURRENCY	decimal	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	decimal	Unit Price in User Company Currency		
USEINVOICEQUANTITY	boolean			
USERBUSINESSUNITCODE	string	User Business Unit Code		
USERCODE	string	User Code		
USERFIRSTNAME	string	User First Name		
USERGUID	string	User GUID		
USERLASTNAME	string	User Last Name		
WORKTYPECODE	string	Work Type Code		
WORKTYPEGUID	string	Work Type GUID		

3.1.132 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.

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	string	<input type="checkbox"/>		
enddate	datetime	<input type="checkbox"/>		
startdate	datetime	<input type="checkbox"/>		
userGUID	string	<input type="checkbox"/>		

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	string	Account GUID		
CASEBUSINESSUNITCODE	string	Project Business Unit Code		
CASECOSTCENTERIDENTIFIER	string	Project Cost Center Identifier		
CASEGUID	string	Project GUID		
CASENAME	string	Project Name		
CASENUMBER	int64	Project Number		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User Company Currency		
DESCRIPTION	string	Ddescription		
ENDTIME	datetime	End Time		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVOICEROWGUID	string	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		

Name	Data Type	Label	Required	Documentation
ISBILLABLE	boolean	Is Billable		
ISPRODUCTIVE	boolean	Is Productive		
ISREADONLY	boolean	Is Read-only		
OVERTIMECODE	string	Overtime Code		
OVERTIMEGUID	string	Overtime GUID		
PHASEGUID	string	Phase GUID		
PRICECURRENCYSHORTFORMOFCS ASECOMPANYCURRENCY	string	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFI NVOICINGCURRENCY	string	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFO RGANIZATIONCURRENCY	string	Price Currency Short Form of Organization Currency		
PRICECURRENCYSHORTFORMOFU SERCOMPANYCURRENCY	string	Price Currency Short Form of User Company Currency		
QUANTITY	decimal	Quantity		
STARTTIME	datetime	Start Time		
TASKGUID	string	Task GUID		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURR ENCY	decimal	Unit Cost in Organization Currency		
UNITCOSTINUSERCOMPANYCURR ENCY	decimal	Unit Cost in User Company Currency		
UNITPRICE	decimal	Unit Price		
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		
UNITPRICEINPROJECTCOMPANYCURR ENCY	decimal	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURR ENCY	decimal	Unit Price in User Company Currency		
USEINVOICEQUANTITY	boolean			
USERBUSINESSUNITCODE	string	User Business Unit Code		
USERCODE	string	User Code		
USERFIRSTNAME	string	User First Name		
USERGUID	string	User GUID		
USERLASTNAME	string	User Last Name		
WORKTYPECODE	string	Work Type Code		
WORKTYPEGUID	string	Work Type GUID		

3.1.133 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 Invariantive UniversalSQL driver 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	datetime	<input type="checkbox"/>		

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	string	Account GUID		
CASEBUSINESSUNITCODE	string	Project Business Unit Code		
CASECOSTCENTERIDENTIFIER	string	Project Cost Center Identifier		
CASEGUID	string	Project GUID		
CASENAME	string	Project Name		
CASENUMBER	int64	Project Number		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing Currency		

Name	Data Type	Label	Required	Documentation
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User Company Currency		
DESCRIPTION	string	Ddescription		
ENDTIME	datetime	End Time		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVICEROWGUID	string	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISPRODUCTIVE	boolean	Is Productive		
ISREADONLY	boolean	Is Read-only		
OVERTIMECODE	string	Overtime Code		
OVERTIMEGUID	string	Overtime GUID		
PHASEGUID	string	Phase GUID		
PRICECURRENCYSHORTFORMOFPROJECTCOMPANYCURRENCY	string	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Price Currency Short Form of Organization Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Price Currency Short Form of User Company Currency		
QUANTITY	decimal	Quantity		
STARTTIME	datetime	Start Time		
TASKGUID	string	Task GUID		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	decimal	Unit Cost in Organization Currency		
UNITCOSTINUSERCOMPANYCURRENCY	decimal	Unit Cost in User Company Currency		
UNITPRICE	decimal	Unit Price		
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		

Name	Data Type	Label	Required	Documentation
UNITPRICEINCA SECOMPANY CURRENCY	decimal	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANY CURRENCY	decimal	Unit Price in User Company Currency		
USEINVOICEQUANTITY	boolean			
USERBUSINESSUNITCODE	string	User Business Unit Code		
USERCODE	string	User Code		
USERFIRSTNAME	string	User First Name		
USERGUID	string	User GUID		
USERLASTNAME	string	User Last Name		
WORKTYPECODE	string	Work Type Code		
WORKTYPEGUID	string	Work Type GUID		

3.1.134 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 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	string	<input type="checkbox"/>		

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	string	Account GUID		
CASEBUSINESSUNITCODE	string	Project Business Unit Code		
CASECOSTCENTERIDENTIFIER	string	Project Cost Center Identifier		
CASEGUID	string	Project GUID		
CASENAME	string	Project Name		
CASENUMBER	int64	Project Number		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User Company Currency		
DESCRIPTION	string	Description		
ENDTIME	datetime	End Time		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVICEROWGUID	string	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISPRODUCTIVE	boolean	Is Productive		
ISREADONLY	boolean	Is Read-only		
OVERTIMECODE	string	Overtime Code		
OVERTIMEGUID	string	Overtime GUID		
PHASEGUID	string	Phase GUID		
PRICECURRENCYSHORTFORMOFPROJECTCOMPANYCURRENCY	string	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Price Currency Short Form of Organization Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Price Currency Short Form of User Company Currency		

Name	Data Type	Label	Required	Documentation
		Company Currency		
QUANTITY	decimal	Quantity		
STARTTIME	datetime	Start Time		
TASKGUID	string	Task GUID		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	decimal	Unit Cost in Organization Currency		
UNITCOSTINUSERCOMPANYCURRENCY	decimal	Unit Cost in User Company Currency		
UNITPRICE	decimal	Unit Price		
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		
UNITPRICEINPROJECTCOMPANYCURRENCY	decimal	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	decimal	Unit Price in User Company Currency		
USEINVOICEQUANTITY	boolean			
USERBUSINESSUNITCODE	string	User Business Unit Code		
USERCODE	string	User Code		
USERFIRSTNAME	string	User First Name		
USERGUID	string	User GUID		
USERLASTNAME	string	User Last Name		
WORKTYPECODE	string	Work Type Code		
WORKTYPEGUID	string	Work Type GUID		

3.1.135 HourEntryPhasesByUserGUID: Visma Severa Hour Entry Phases by User GUID

Catalog: Severa

Schema: API

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	string	<input type="checkbox"/>		

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	string	Account GUID		
ACCOUNTNAME	string	Account Name		
CASEGUID	string	Project GUID		
CHILDPHASES	string	Child Phases		
GUID	string	GUID		
ISLOCKED	boolean	Is Locked		
NAME	string	Name		
PARENTPHASEGUID	string	Parent Phase GUID		
WORKTYPEGUID	string	Work Type GUID		

3.1.136 Industries: Visma Severa Industries

Catalog: Severa

Schema: API

Label: Industries

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	string	Code		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
NAME	string	Name		

3.1.137 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	string	<input type="checkbox"/>		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
GETINVOICEATTACHMENTFILERE ULT	byte[]			

3.1.138 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	string	<input type="checkbox"/>		

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	string	GUID		
MIMETYPE	string	MIME-type		
NAME	string	Name		
PRINTORDER	int32	Print Order		

3.1.139 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	string	<input type="checkbox"/>		

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	string	Account GUID		
ACCOUNTNAME	string	Account Name		
ACCOUNTNUMBER	int64	Account Number		
BUSINESSUNITCODE	string	Business Unit Code		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
BUYERORGANIZATION_BUYERCONTACTPERSONNAME	string			
BUYERORGANIZATION_BUYERELECTRONICINVOICEADDRESS	string			
BUYERORGANIZATION_BUYERELECTRONICINVOICEOPERATOR	string			
BUYERORGANIZATION_BUYEREMAILADDRESSIDENTIFIER	string			
BUYERORGANIZATION_BUYERORGANIZATIONNAME	string			
BUYERORGANIZATION_BUYERORGANIZATIONTAXCODE	string			
BUYERORGANIZATION_BUYERPARTYIDENTIFIER	string			
BUYERORGANIZATION_BUYERPHONENUMBERIDENTIFIER	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRY	string			

Name	Data Type	Label	Required	Documentation
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRYCODE	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_STREETNAME	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_TOWNNAME	string			
CULTURE	string	Culture		
CURRENCYGUID	string	Currency GUID		
CURRENCYNAME	string	Currency Name		
CURRENCYRATE	decimal	Currency Rate		
CURRENCYSHORTFORM	string	Currency Short Form		
DATE	datetime	Date		
DESCRIPTION	string	Ddescription		
DUEDATE	datetime	Due Date		
ENTRYDATE	datetime	Entry Date		
GUID	string	GUID		
INVOICEALLOWEDCATEGORIESGUIDS	string			
INVOICECONFIG_BILLEXPENSE	string			
INVOICECONFIG_BILLHOUR	string			
INVOICECONFIG_BILLPRODUCT	string			
INVOICECONFIG_EXPENSEITEMDESCRIPTIONFORMAT	string			
INVOICECONFIG_HOURITEMDESCRIPTIONFORMAT	string			
INVOICECONFIG_HOURPRIMARYGROUPBY	string			
INVOICECONFIG_HOURSECONDARYGROUPBY	string			
INVOICECONFIG_ISTAXFREE	boolean			
INVOICECONFIG_PRODUCTITEMDESCRIPTIONFORMAT	string			
INVOICECONFIG_PRODUCTPRIMARYGROUPBY	string			
INVOICECONFIG_PRODUCTSECONDARYGROUPBY	string			
INVOICECONFIG_SHOWATTACHPRICEEXCLUDINGVAT	boolean			
INVOICECONFIG_SHOWATTACHUNIT	boolean			
INVOICECONFIG_SHOWATTACHUNITPRICE	boolean			

Name	Data Type	Label	Required	Documentation
INVOICECONFIG_SHOWATTACHVAT	boolean			
INVOICECONFIG_SHOWLOGOONINVOICE	boolean			
INVOICECONFIG_SHOWPRICEEXCLUDINGVAT	boolean			
INVOICECONFIG_SHOWQUANTITY	boolean			
INVOICECONFIG_SHOWUNIT	boolean			
INVOICECONFIG_SHOWUNITPRICE	boolean			
INVOICECONFIG_SHOWVAT	boolean			
INVOICECONFIG_TAXFREEDESCRIPTION	string			
INVOICECONFIG_TRAVELPRIMARYGROUPBY	string			
INVOICECONFIG_TRAVELSECONDARYGROUPBY	string			
INVOICEDETAILS_INVOICEDATE	datetime			
INVOICEDETAILS_INVOICEDUEDATE	datetime			
INVOICEDETAILS_INVOICENUMBER	int32			
INVOICEDETAILS_INVOICETOTALVATAMOUNT	decimal			
INVOICEDETAILS_INVOICETOTALVATEXCLUDEDAMOUNT	decimal			
INVOICEDETAILS_INVOICETOTALVATINCLUDEDAMOUNT	decimal			
INVOICEDETAILS_INVOICETYPETEXT	string			
INVOICEDETAILS_ORDERIDENTIFIER	string			
INVOICEDETAILS_ORIGINCODE	string			
INVOICEDETAILS_PAYMENTOVERDUEDEFINEFREETEXT	string			
INVOICEDETAILS_PAYMENTOVERDUEDEFINEPERCENT	decimal			
INVOICEDETAILS_PAYMENTTERMSFREETEXT	string			
INVOICEDETAILS_SELLERREFERENCEIDENTIFIER	string			
INVOICEDETAILS_VATBASEAMOUNT	decimal			
INVOICEDETAILS_VATRATEAMOUNT	decimal			
INVOICEDETAILS_VATRATEPERCENT	int32			
INVICENUMBERINGBUSINESSUNITGUID	string	Invoice Numbering Business Unit GUID		
INVICEROWS	string			

Name	Data Type	Label	Required	Documentation
INVOICESTATUS	string	Invoice Status		
ISREDITNOTE	boolean	Is Credit Note		
ISREIMBURSED	boolean	Is Reimbursed		
ISTAXFREE	boolean	Is Tax Free		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
MAINCASEGUID	string	Main Case GUID		
NOTES	string	Notes		
NOTESAFTERINVOICEROWS	string	Notes after Invoice Lines		
NOTESBEFOREINVOICEROWS	string	Notes before Invoice Lines		
NUMBER	int32	Number		
ORDERNUMBER	string	Order Number		
OURREFERENCE	string	Our Reference		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTDATE	datetime	Payment Date		
PAYMENTSTATUS	string	Payment Status		
PAYMENTTERM	int32	Payment Term		
PONUMBER	string	PO Number		
RECEIVERADDRESSLINE	string	Receiver Addressline		
RECEIVERCITY	string	Receiver City		
RECEIVERCONTACTEMAIL	string	Receiver Contact Email		
RECEIVERCONTACTNAME	string	Receiver Contact Name		
RECEIVERCOUNTRY	string	Receiver Country		
RECEIVERCUSTOMERNAME	string	Receiver Customer Name		
RECEIVERPOSTALCODE	string	Receiver Postal Code		
RECEIVERSTATE	string	Receiver State		
RECEIVERVATNUMBER	string	Receiver VAT Number		
REFERENCENUMBER	string	Reference Number		
REIMBURSEINVOICEGUID	string	Reimburse Invoice GUID		
SELLERCONTACT_SELLERACCOUNTDETAILS	string			
SELLERCONTACT_SELLERCONTACTPERSONNAME	string			
SELLERCONTACT_SELLEREMAILADDRESSIDENTIFIER	string			
SELLERCONTACT_SELLERFAXNUMBER	string			
SELLERCONTACT_SELLERFREEXT	string			

Name	Data Type	Label	Required	Documentation
SELLERCONTACT_SELLERHOMETOWNNAME	string			
SELLERCONTACT_SELLERPHONE NUMBER	string			
SELLERCONTACT_SELLERWEBADDRESSIDENTIFIER	string			
SELLERORGANIZATION_SELLERINVOICEADDRESS	string			
SELLERORGANIZATION_SELLERINVOICEOPERATOR	string			
SELLERORGANIZATION_SELLERORGANISATIONNAME	string			
SELLERORGANIZATION_SELLERORGANISATIONTAXCODE	string			
SELLERORGANIZATION_SELLERPARTYIDENTIFIER	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRY	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRYCODE	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_STREETNAME	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_TOWNNAME	string			
SENDERADDRESSLINE	string	Sender Addressline		
SENDERCITY	string	Sender City		
SENDERCONTACTNAME	string	Sender Contact Name		
SENDERCOUNTRY	string	Sender Country		
SENDERNAME	string	Sender Name		
SENDERPOSTALCODE	string	Sender Postal Code		
SENDERSTATE	string	Sender State		
SENDERVATNUMBER	string	Sender VAT Number		
TAXFREEDESCRIPTION	string	Tax Free Description		
TITLE	string	Title		
VIRTUALBANKBARCODE	string	Virtual Bank Barcode		
YOURREFERENCE	string	Your Reference		

3.1.140 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	string	<input type="checkbox"/>		
invoiceNumber	int32	<input type="checkbox"/>		

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	string	Account GUID		
ACCOUNTNAME	string	Account Name		
ACCOUNTNUMBER	int64	Account Number		
BUSINESSUNITCODE	string	Business Unit Code		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
BUYERORGANIZATION_BUYERCONTACTPERSONNAME	string			
BUYERORGANIZATION_BUYERELECTRONICINVOICEADDRESS	string			
BUYERORGANIZATION_BUYERELECTRONICINVOICEOPERATOR	string			

Name	Data Type	Label	Required	Documentation
BUYERORGANIZATION_BUYEREMAILADDRESSIDENTIFIER	string			
BUYERORGANIZATION_BUYERORGANISATIONNAME	string			
BUYERORGANIZATION_BUYERORGANISATIONTAXCODE	string			
BUYERORGANIZATION_BUYERPARTYIDENTIFIER	string			
BUYERORGANIZATION_BUYERPHONENUMBERIDENTIFIER	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRY	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRYCODE	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_STREETNAME	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_TOWNNAME	string			
CULTURE	string	Culture		
CURRENCYGUID	string	Currency GUID		
CURRENCYNAME	string	Currency Name		
CURRENCYRATE	decimal	Currency Rate		
CURRENCYSHORTFORM	string	Currency Short Form		
DATE	datetime	Date		
DESCRIPTION	string	Description		
DUEDATE	datetime	Due Date		
ENTRYDATE	datetime	Entry Date		
GUID	string	GUID		
INVOICEALLOWEDCATEGORIESGUIDS	string			
INVOICECONFIG_BILLEXPENSE	string			
INVOICECONFIG_BILLHOUR	string			
INVOICECONFIG_BILLPRODUCT	string			
INVOICECONFIG_EXPENSEITEMDESCRIPTIONFORMAT	string			
INVOICECONFIG_HOURITEMDESCRIPTIONFORMAT	string			
INVOICECONFIG_HOURPRIMARYGROUPBY	string			
INVOICECONFIG_HOURSECONDARYGROUPBY	string			
INVOICECONFIG_ISTAXFREE	boolean			

Name	Data Type	Label	Required	Documentation
INVOICECONFIG_PRODUCTITEMDESCRIPTIONFORMAT	string			
INVOICECONFIG_PRODUCTPRIMARYGROUPBY	string			
INVOICECONFIG_PRODUCTSECONDARYGROUPBY	string			
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	string			
INVOICECONFIG_TRAVELPRIMARYGROUPBY	string			
INVOICECONFIG_TRAVELSECONDARYGROUPBY	string			
INVOICEDETAILS_INVOICEDATE	datetime			
INVOICEDETAILS_INVOICEDUEDATE	datetime			
INVOICEDETAILS_INVOICENUMBER	int32			
INVOICEDETAILS_INVOICETOTALVATA MOUNT	decimal			
INVOICEDETAILS_INVOICETOTALVATEXCLUDEDAMOUNT	decimal			
INVOICEDETAILS_INVOICETOTALVATINCLUDEDAMOUNT	decimal			
INVOICEDETAILS_INVOICETYPETEXT	string			
INVOICEDETAILS_ORDERIDENTIFIER	string			
INVOICEDETAILS_ORIGINCODE	string			
INVOICEDETAILS_PAYMENTOVERDUDEFINEFREETEXT	string			
INVOICEDETAILS_PAYMENTOVERDUDEFINEPERCENT	decimal			

Name	Data Type	Label	Required	Documentation
INVOICEDETAILS_PAYMENTTERMSFREETEXT	string			
INVOICEDETAILS_SELLERREFERENCEIDENTIFIER	string			
INVOICEDETAILS_VATBASEAMOUNT	decimal			
INVOICEDETAILS_VATRATEAMOUNT	decimal			
INVOICEDETAILS_VATRATEPERCENT	int32			
INVICENUMBERINGBUSINESSUNITGUID	string	Invoice Numbering Business Unit GUID		
INVOICEROWS	string			
INVOICESTATUS	string	Invoice Status		
ISCREDITNOTE	boolean	Is Credit Note		
ISREIMBURSED	boolean	Is Reimbursed		
ISTAXFREE	boolean	Is Tax Free		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
MAINCASEGUID	string	Main Case GUID		
NOTES	string	Notes		
NOTESAFTERINVOICEROWS	string	Notes after Invoice Lines		
NOTESBEFOREINVOICEROWS	string	Notes before Invoice Lines		
NUMBER	int32	Number		
ORDERNUMBER	string	Order Number		
OURREFERENCE	string	Our Reference		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTDATE	datetime	Payment Date		
PAYMENTSTATUS	string	Payment Status		
PAYMENTTERM	int32	Payment Term		
PONUMBER	string	PO Number		
RECEIVERADDRESSLINE	string	Receiver Addressline		
RECEIVERCITY	string	Receiver City		
RECEIVERCONTACTEMAIL	string	Receiver Contact Email		
RECEIVERCONTACTNAME	string	Receiver Contact Name		
RECEIVERCOUNTRY	string	Receiver Country		
RECEIVERCUSTOMERNAME	string	Receiver Customer Name		
RECEIVERPOSTALCODE	string	Receiver Postal Code		
RECEIVERSTATE	string	Receiver State		
RECEIVERVATNUMBER	string	Receiver VAT Number		

Name	Data Type	Label	Required	Documentation
REFERENCENUMBER	string	Reference Number		
REIMBURSEINVOICEGUID	string	Reimburse Invoice GUID		
SELLERCONTACT_SELLERACCOUNTDETAILS	string			
SELLERCONTACT_SELLERCONTACTPERSONNAME	string			
SELLERCONTACT_SELLEREMAILADDRESSIDENTIFIER	string			
SELLERCONTACT_SELLERFAXNUMBER	string			
SELLERCONTACT_SELLERFREEXT	string			
SELLERCONTACT_SELLERHOMETOWNNAME	string			
SELLERCONTACT_SELLERPHONE NUMBER	string			
SELLERCONTACT_SELLERWEBADDRESSIDENTIFIER	string			
SELLERORGANIZATION_SELLERINVOICEADDRESS	string			
SELLERORGANIZATION_SELLERINVOICEOPERATOR	string			
SELLERORGANIZATION_SELLERORGANISATIONNAME	string			
SELLERORGANIZATION_SELLERORGANISATIONTAXCODE	string			
SELLERORGANIZATION_SELLERPARTYIDENTIFIER	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRY	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRYCODE	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_STREETNAME	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_TOWNNAME	string			
SENDERADDRESSLINE	string	Sender Addressline		
SENDERCITY	string	Sender City		
SENDERCONTACTNAME	string	Sender Contact Name		
SENDERCOUNTRY	string	Sender Country		
SENDERNAME	string	Sender Name		

Name	Data Type	Label	Required	Documentation
SENDERPOSTALCODE	string	Sender Postal Code		
SENDERSTATE	string	Sender State		
SENDERVATNUMBER	string	Sender VAT Number		
TAXFREEDESCRIPTION	string	Tax Free Description		
TITLE	string	Title		
VIRTUALBANKBARCODE	string	Virtual Bank Barcode		
YOURREFERENCE	string	Your Reference		

3.1.141 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.

Name	Data Type	Required	Default Value	Documentation
caseGuid	string	<input type="checkbox"/>		

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	string	Account GUID		

Name	Data Type	Label	Required	Documentation
CASEBUSINESSUNITCODE	string	Project Business Unit Code		
CASECOSTCENTERIDENTIFIER	string	Project Cost Center Identifier		
CASEGUID	string	Project GUID		
CASENAME	string	Project Name		
CASENUMBER	int64	Project Number		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User Company Currency		
DESCRIPTION	string	Ddescription		
ENDTIME	datetime	End Time		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVICEROWGUID	string	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISPRODUCTIVE	boolean	Is Productive		
ISREADONLY	boolean	Is Read-only		
OVERTIMECODE	string	Overtime Code		
OVERTIMEGUID	string	Overtime GUID		
PHASEGUID	string	Phase GUID		
PRICECURRENCYSHORTFORMOFPROJECTCOMPANYCURRENCY	string	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Price Currency Short Form of Organization Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Price Currency Short Form of User Company Currency		
QUANTITY	decimal	Quantity		
STARTTIME	datetime	Start Time		

Name	Data Type	Label	Required	Documentation
TASKGUID	string	Task GUID		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	decimal	Unit Cost in Organization Currency		
UNITCOSTINUSERCOMPANYCURRENCY	decimal	Unit Cost in User Company Currency		
UNITPRICE	decimal	Unit Price		
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		
UNITPRICEINPROJECTCOMPANYCURRENCY	decimal	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	decimal	Unit Price in User Company Currency		
USEINVOICEQUANTITY	boolean			
USERBUSINESSUNITCODE	string	User Business Unit Code		
USERCODE	string	User Code		
USERFIRSTNAME	string	User First Name		
USERGUID	string	User GUID		
USERLASTNAME	string	User Last Name		
WORKTYPECODE	string	Work Type Code		
WORKTYPEGUID	string	Work Type GUID		

3.1.142 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

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	string	<input type="checkbox"/>		

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	string	Projects		
INVOICEROWORIGINS	string			

3.1.143 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.

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	string	<input type="checkbox"/>		
invoiceNumber	int32	<input type="checkbox"/>		
invoiceNumberingBusinessUnitGUID	string	<input type="checkbox"/>		

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	byte[]			

3.1.144 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 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	string	<input type="checkbox"/>		
invoiceGuid	string	<input type="checkbox"/>		

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	byte[]			

3.1.145 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	string	<input type="checkbox"/>		
endDate	datetime	<input type="checkbox"/>		
options	int32	<input type="checkbox"/>		
startDate	datetime	<input type="checkbox"/>		

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	string	Account GUID		
ACCOUNTNAME	string	Account Name		

Name	Data Type	Label	Required	Documentation
ACCOUNTNUMBER	int64	Account Number		
BUSINESSUNITCODE	string	Business Unit Code		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
BUYERORGANIZATION_BUYERCONTACTPERSONNAME	string			
BUYERORGANIZATION_BUYERELECTRONICINVOICEADDRESS	string			
BUYERORGANIZATION_BUYERELECTRONICINVOICEOPERATOR	string			
BUYERORGANIZATION_BUYEREMAILADDRESSIDENTIFIER	string			
BUYERORGANIZATION_BUYERORGANISATIONNAME	string			
BUYERORGANIZATION_BUYERORGANISATIONTAXCODE	string			
BUYERORGANIZATION_BUYERPARTYIDENTIFIER	string			
BUYERORGANIZATION_BUYERPHONENUMBERIDENTIFIER	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRY	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRYCODE	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_STREETNAME	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_TOWNNAME	string			
CULTURE	string	Culture		
CURRENCYGUID	string	Currency GUID		
CURRENCYNAME	string	Currency Name		
CURRENCYRATE	decimal	Currency Rate		
CURRENCYSHORTFORM	string	Currency Short Form		
DATE	datetime	Date		
DESCRIPTION	string	Ddescription		
DUEDATE	datetime	Due Date		
ENTRYDATE	datetime	Entry Date		
GUID	string	GUID		
INVOICEALLOWEDCATEGORIESGUIDS	string			
INVOICECONFIG_BILLEXPENSE	string			

Name	Data Type	Label	Required	Documentation
INVOICECONFIG_BILLHOUR	string			
INVOICECONFIG_BILLPRODUCT	string			
INVOICECONFIG_EXPENSEITEMDESCRIPTIONFORMAT	string			
INVOICECONFIG_HOURITEMDESCRIPTIONFORMAT	string			
INVOICECONFIG_HOURPRIMARYGROUPBY	string			
INVOICECONFIG_HOURSECONDARYGROUPBY	string			
INVOICECONFIG_ISTAXFREE	boolean			
INVOICECONFIG_PRODUCTITEMDESCRIPTIONFORMAT	string			
INVOICECONFIG_PRODUCTPRIMARYGROUPBY	string			
INVOICECONFIG_PRODUCTSECONDARYGROUPBY	string			
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	string			
INVOICECONFIG_TRAVELPRIMARYGROUPBY	string			
INVOICECONFIG_TRAVELSECONDARYGROUPBY	string			
INVOICEDETAILS_INVOICEDATE	datetime			
INVOICEDETAILS_INVOICEDUEDATE	datetime			
INVOICEDETAILS_INVOICENUMBER	int32			
INVOICEDETAILS_INVOICETOTALVATAMOUNT	decimal			
INVOICEDETAILS_INVOICETOTALVATEXCLUDEDAMOUNT	decimal			

Name	Data Type	Label	Required	Documentation
INVOICEDETAILS_INVOICETOTALV ATAINCLUDEDAMOUNT	decimal			
INVOICEDETAILS_INVOICETYPETE XT	string			
INVOICEDETAILS_ORDERIDENTIFIE R	string			
INVOICEDETAILS_ORIGINC <small>ODE</small>	string			
INVOICEDETAILS_PAYMENTOVER DUEFINEFREETEXT	string			
INVOICEDETAILS_PAYMENTOVER DUEFINEPERCENT	decimal			
INVOICEDETAILS_PAYMENTTERM SFREETEXT	string			
INVOICEDETAILS_SELLERREFERE NCEIDENTIFIER	string			
INVOICEDETAILS_VATBASEAMOU NT	decimal			
INVOICEDETAILS_VATR <small>ATE</small> AMOU NT	decimal			
INVOICEDETAILS_VATR <small>ATE</small> PERCE NT	int32			
INVICENUMBERINGBUSINESSUNI T <small>GUID</small>	string	Invoice Numbering Business Unit GUID		
INVOICER <small>OWS</small>	string			
INVOICE <small>STATUS</small>	string	Invoice Status		
IS <small>CREDIT</small> NOTE	boolean	Is Credit Note		
IS <small>REIMBURSED</small>	boolean	Is Reimbursed		
IS <small>TAX</small> FREE	boolean	Is Tax Free		
LANGUA <small>G</small> CODE	string	Language Code		
LANGUA <small>G</small> GUID	string	Language GUID		
MA <small>IN</small> CASE <small>GUID</small>	string	Main Case GUID		
NOT <small>ES</small>	string	Notes		
NOT <small>ES</small> A <small>FTER</small> INVOICER <small>OWS</small>	string	Notes after Invoice Lines		
NOT <small>ES</small> B <small>EFORE</small> INVOICER <small>OWS</small>	string	Notes before Invoice Lines		
NUM <small>BER</small>	int32	Number		
ORD <small>ER</small> NUM <small>BER</small>	string	Order Number		
OUR <small>REFERENCE</small>	string	Our Reference		
OVER <small>DUE</small> INTEREST	decimal	Overdue Interest		
P <small>AYMENT</small> DATE	datetime	Payment Date		
P <small>AYMENT</small> STATUS	string	Payment Status		
P <small>AYMENT</small> TERM	int32	Payment Term		
P <small>ONUMBER</small>	string	PO Number		
RECEIV <small>ER</small> ADDRESS <small>LINE</small>	string	Receiver Addressline		

Name	Data Type	Label	Required	Documentation
RECEIVERCITY	string	Receiver City		
RECEIVERCONTACTEMAIL	string	Receiver Contact Email		
RECEIVERCONTACTNAME	string	Receiver Contact Name		
RECEIVERCOUNTRY	string	Receiver Country		
RECEIVERCUSTOMERNAME	string	Receiver Customer Name		
RECEIVERPOSTALCODE	string	Receiver Postal Code		
RECEIVERSTATE	string	Receiver State		
RECEIVERVATNUMBER	string	Receiver VAT Number		
REFERENCENUMBER	string	Reference Number		
REIMBURSEINVOICEGUID	string	Reimburse Invoice GUID		
SELLERCONTACT_SELLERACCOUNTDETAILS	string			
SELLERCONTACT_SELLERCONTACTPERSONNAME	string			
SELLERCONTACT_SELLEREMAILADDRESSIDENTIFIER	string			
SELLERCONTACT_SELLERFAXNUMBER	string			
SELLERCONTACT_SELLERFREEXT	string			
SELLERCONTACT_SELLERHOMETOWNNAME	string			
SELLERCONTACT_SELLERPHONE NUMBER	string			
SELLERCONTACT_SELLERWEBADDRESSIDENTIFIER	string			
SELLERORGANIZATION_SELLERINVOICEADDRESS	string			
SELLERORGANIZATION_SELLERINVOICEOPERATOR	string			
SELLERORGANIZATION_SELLERORGANISATIONNAME	string			
SELLERORGANIZATION_SELLERORGANISATIONTAXCODE	string			
SELLERORGANIZATION_SELLERPARTYIDENTIFIER	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRY	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRYCODE	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	string			

Name	Data Type	Label	Required	Documentation
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_STREETNAME	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_TOWNNAME	string			
SENDERADDRESSLINE	string	Sender Addressline		
SENDERCITY	string	Sender City		
SENDERCONTACTNAME	string	Sender Contact Name		
SENDERCOUNTRY	string	Sender Country		
SENDERNAME	string	Sender Name		
SENDERPOSTALCODE	string	Sender Postal Code		
SENDERSTATE	string	Sender State		
SENDERVATNUMBER	string	Sender VAT Number		
TAXFREEDESCRIPTION	string	Tax Free Description		
TITLE	string	Title		
VIRTUALBANKBARCODE	string	Virtual Bank Barcode		
YOURREFERENCE	string	Your Reference		

3.1.146 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.

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	string	<input type="checkbox"/>		
endDate	datetime	<input type="checkbox"/>		
options	int32	<input type="checkbox"/>		
startDate	datetime	<input type="checkbox"/>		
status	string	<input type="checkbox"/>		

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	string	Account GUID		
ACCOUNTNAME	string	Account Name		
ACCOUNTNUMBER	int64	Account Number		
BUSINESSUNITCODE	string	Business Unit Code		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
BUYERORGANIZATION_BUYERCONTACTPERSONNAME	string			
BUYERORGANIZATION_BUYERELECTRONICINVOICEADDRESS	string			
BUYERORGANIZATION_BUYERELECTRONICINVOICEOPERATOR	string			
BUYERORGANIZATION_BUYEREMAILADDRESSIDENTIFIER	string			
BUYERORGANIZATION_BUYERORGANISATIONNAME	string			
BUYERORGANIZATION_BUYERORGANISATIONTAXCODE	string			
BUYERORGANIZATION_BUYERPARTYIDENTIFIER	string			
BUYERORGANIZATION_BUYERPHONENUMBERIDENTIFIER	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRY	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRYCODE	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_STREETNAME	string			

Name	Data Type	Label	Required	Documentation
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_TOWNNAME	string			
CULTURE	string	Culture		
CURRENCYGUID	string	Currency GUID		
CURRENCYNAME	string	Currency Name		
CURRENCYRATE	decimal	Currency Rate		
CURRENCYSHORTFORM	string	Currency Short Form		
DATE	datetime	Date		
DESCRIPTION	string	Description		
DUEDATE	datetime	Due Date		
ENTRYDATE	datetime	Entry Date		
GUID	string	GUID		
INVOICEALLOWEDCASEGUIDS	string			
INVOICECONFIG_BILLEXPENSE	string			
INVOICECONFIG_BILLHOUR	string			
INVOICECONFIG_BILLPRODUCT	string			
INVOICECONFIG_EXPENSEITEMDESCRIPTIONFORMAT	string			
INVOICECONFIG_HOURITEMDESCRIPTIONFORMAT	string			
INVOICECONFIG_HOURPRIMARYGROUPBY	string			
INVOICECONFIG_HOURSECONDARYGROUPBY	string			
INVOICECONFIG_ISTAXFREE	boolean			
INVOICECONFIG_PRODUCTITEMDESCRIPTIONFORMAT	string			
INVOICECONFIG_PRODUCTPRIMARYGROUPBY	string			
INVOICECONFIG_PRODUCTSECONDARYGROUPBY	string			
INVOICECONFIG_SHOWATTACHPRICEEXCLUDINGVAT	boolean			
INVOICECONFIG_SHOWATTACHUNIT	boolean			
INVOICECONFIG_SHOWATTACHUNITPRICE	boolean			
INVOICECONFIG_SHOWATTACHVAT	boolean			
INVOICECONFIG_SHOWLOGOONINVOICE	boolean			
INVOICECONFIG_SHOWPRICEEXCLUDINGVAT	boolean			
INVOICECONFIG_SHOWQUANTITY	boolean			
INVOICECONFIG_SHOWUNIT	boolean			

Name	Data Type	Label	Required	Documentation
INVOICECONFIG_SHOWUNITPRICE	boolean			
INVOICECONFIG_SHOWVAT	boolean			
INVOICECONFIG_TAXFREEDESCRIPTION	string			
INVOICECONFIG_TRAVELPRIMARYGROUPBY	string			
INVOICECONFIG_TRAVELSECONDARYGROUPBY	string			
INVOICEDetails_INVOICEDATE	datetime			
INVOICEDetails_INVOICEDUEDATE	datetime			
INVOICEDetails_INVOICENUMBER	int32			
INVOICEDetails_INVOICETOTALVATAmount	decimal			
INVOICEDetails_INVOICETOTALVATEXCLUDEDAMOUNT	decimal			
INVOICEDetails_INVOICETOTALVATINCLUDEDAMOUNT	decimal			
INVOICEDetails_INVOICETYPETEXT	string			
INVOICEDetails_ORDERIDENTIFIER	string			
INVOICEDetails_ORIGINCODE	string			
INVOICEDetails_PAYMENTOVERDUDEFREEFREETEXT	string			
INVOICEDetails_PAYMENTOVERDUDEFINEPERCENT	decimal			
INVOICEDetails_PAYMENTTERMSFREETEXT	string			
INVOICEDetails_SELLERREFERENCEIDENTIFIER	string			
INVOICEDetails_VATBASEAMOUNT	decimal			
INVOICEDetails_VATRATEAMOUNT	decimal			
INVOICEDetails_VATRATEPERCENT	int32			
INVICENUMBERINGBUSINESSUNITGUID	string	Invoice Numbering Business Unit GUID		
INVOICEROWS	string			
INVOICESTATUS	string	Invoice Status		
ISCREDITNOTE	boolean	Is Credit Note		
ISREIMBURSED	boolean	Is Reimbursed		
ISTAXFREE	boolean	Is Tax Free		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
MAINCASEGUID	string	Main Case GUID		

Name	Data Type	Label	Required	Documentation
NOTES	string	Notes		
NOTESAFTERINVOICEROWS	string	Notes after Invoice Lines		
NOTESBEFOREINVOICEROWS	string	Notes before Invoice Lines		
NUMBER	int32	Number		
ORDERNUMBER	string	Order Number		
OURREFERENCE	string	Our Reference		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTDATE	datetime	Payment Date		
PAYMENTSTATUS	string	Payment Status		
PAYMENTTERM	int32	Payment Term		
PONUMBER	string	PO Number		
RECEIVERADDRESSLINE	string	Receiver Addressline		
RECEIVERCITY	string	Receiver City		
RECEIVERCONTACTEMAIL	string	Receiver Contact Email		
RECEIVERCONTACTNAME	string	Receiver Contact Name		
RECEIVERCOUNTRY	string	Receiver Country		
RECEIVERCUSTOMERNAME	string	Receiver Customer Name		
RECEIVERPOSTALCODE	string	Receiver Postal Code		
RECEIVERSTATE	string	Receiver State		
RECEIVERVATNUMBER	string	Receiver VAT Number		
REFERENCENUMBER	string	Reference Number		
REIMBURSEINVOICEGUID	string	Reimburse Invoice GUID		
SELLERCONTACT_SELLERACCOUNTDETAILS	string			
SELLERCONTACT_SELLERCONTACTPERSONNAME	string			
SELLERCONTACT_SELLEREMAILADDRESSIDENTIFIER	string			
SELLERCONTACT_SELLERFAXNUMBER	string			
SELLERCONTACT_SELLERFREEXT	string			
SELLERCONTACT_SELLERHOMETOWNNAME	string			
SELLERCONTACT_SELLERPHONENUMBER	string			
SELLERCONTACT_SELLERWEBADDRESSIDENTIFIER	string			
SELLERORGANIZATION_SELLERINVOICEADDRESS	string			

Name	Data Type	Label	Required	Documentation
SELLERORGANIZATION_SELLERINVOICEOPERATOR	string			
SELLERORGANIZATION_SELLERORGANISATIONNAME	string			
SELLERORGANIZATION_SELLERORGANISATIONTAXCODE	string			
SELLERORGANIZATION_SELLERPARTYIDENTIFIER	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRY	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRYCODE	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_STREETNAME	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_TOWNNAME	string			
SENDERADDRESSLINE	string	Sender Addressline		
SENDERCITY	string	Sender City		
SENDERCONTACTNAME	string	Sender Contact Name		
SENDERCOUNTRY	string	Sender Country		
SENDERNAME	string	Sender Name		
SENDERPOSTALCODE	string	Sender Postal Code		
SENDERSTATE	string	Sender State		
SENDERVATNUMBER	string	Sender VAT Number		
TAXFREEDESCRIPTION	string	Tax Free Description		
TITLE	string	Title		
VIRTUALBANKBARCODE	string	Virtual Bank Barcode		
YOURREFERENCE	string	Your Reference		

3.1.147 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.

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	string	<input type="checkbox"/>		
maxInvoices	int32	<input type="checkbox"/>		
options	int32	<input type="checkbox"/>		
status	string	<input type="checkbox"/>		

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	string	Account GUID		
ACCOUNTNAME	string	Account Name		
ACCOUNTNUMBER	int64	Account Number		
BUSINESSUNITCODE	string	Business Unit Code		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
BUYERORGANIZATION_BUYERCONTACTPERSONNAME	string			
BUYERORGANIZATION_BUYERELECTRONICINVOICEADDRESS	string			
BUYERORGANIZATION_BUYERELECTRONICINVOICEOPERATOR	string			
BUYERORGANIZATION_BUYEREMAILADDRESSIDENTIFIER	string			
BUYERORGANIZATION_BUYERORGANIZATIONNAME	string			
BUYERORGANIZATION_BUYERORGANIZATIONTAXCODE	string			
BUYERORGANIZATION_BUYERPARTYIDENTIFIER	string			

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

Name	Data Type	Label	Required	Documentation
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	string			
INVOICECONFIG_TRAVELPRIMARYGROUPBY	string			
INVOICECONFIG_TRAVELSECONDARYGROUPBY	string			
INVOICEDetails_INVOICEDATE	datetime			
INVOICEDetails_INVOICEDUEDATE	datetime			
INVOICEDetails_INVOICENUMBER	int32			
INVOICEDetails_INVOICETOTALVATA MOUNT	decimal			
INVOICEDetails_INVOICETOTALVATEXCLUDEDAMOUNT	decimal			
INVOICEDetails_INVOICETOTALVATA INCLUDEDAMOUNT	decimal			
INVOICEDetails_INVOICETYPETEXT	string			
INVOICEDetails_ORDERIDENTIFIER	string			
INVOICEDetails_ORIGINCODE	string			
INVOICEDetails_PAYMENTOVERDUDEFINEFREETEXT	string			
INVOICEDetails_PAYMENTOVERDUDEFINEPERCENT	decimal			
INVOICEDetails_PAYMENTTERMSFREETEXT	string			
INVOICEDetails_SELLERREFERENCEIDENTIFIER	string			
INVOICEDetails_VATBASEAMOUNT	decimal			
INVOICEDetails_VATRATEAMOUNT	decimal			

Name	Data Type	Label	Required	Documentation
INVOICEDETAILS_VATRATEPERCENT	int32			
INVICENUMBERINGBUSINESSUNITGUID	string	Invoice Numbering Business Unit GUID		
INVOICEROWS	string			
INVOICESTATUS	string	Invoice Status		
ISCREDITNOTE	boolean	Is Credit Note		
ISREIMBURSED	boolean	Is Reimbursed		
ISTAXFREE	boolean	Is Tax Free		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
MAINCASEGUID	string	Main Case GUID		
NOTES	string	Notes		
NOTESAFTERINVOICEROWS	string	Notes after Invoice Lines		
NOTESBEFOREINVOICEROWS	string	Notes before Invoice Lines		
NUMBER	int32	Number		
ORDERNUMBER	string	Order Number		
OURREFERENCE	string	Our Reference		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTDATE	datetime	Payment Date		
PAYMENTSTATUS	string	Payment Status		
PAYMENTTERM	int32	Payment Term		
PONUMBER	string	PO Number		
RECEIVERADDRESSLINE	string	Receiver Addressline		
RECEIVERCITY	string	Receiver City		
RECEIVERCONTACTEMAIL	string	Receiver Contact Email		
RECEIVERCONTACTNAME	string	Receiver Contact Name		
RECEIVERCOUNTRY	string	Receiver Country		
RECEIVERCUSTOMERNAME	string	Receiver Customer Name		
RECEIVERPOSTALCODE	string	Receiver Postal Code		
RECEIVERSTATE	string	Receiver State		
RECEIVERVATNUMBER	string	Receiver VAT Number		
REFERENCENUMBER	string	Reference Number		
REIMBURSEINVOICEGUID	string	Reimburse Invoice GUID		
SELLERCONTACT_SELLERACCOUNTDETAILS	string			
SELLERCONTACT_SELLERCONTACTPERSONNAME	string			

Name	Data Type	Label	Required	Documentation
SELLERCONTACT_SELLEREMAILADDRESSIDENTIFIER	string			
SELLERCONTACT_SELLERFAXNUMBER	string			
SELLERCONTACT_SELLERFREEXT	string			
SELLERCONTACT_SELLERHOMETOWNNAME	string			
SELLERCONTACT_SELLERPHONE NUMBER	string			
SELLERCONTACT_SELLERWEBADDRESSIDENTIFIER	string			
SELLERORGANIZATION_SELLERINVOICEADDRESS	string			
SELLERORGANIZATION_SELLERINVOICEOPERATOR	string			
SELLERORGANIZATION_SELLERORGANISATIONNAME	string			
SELLERORGANIZATION_SELLERORGANISATIONTAXCODE	string			
SELLERORGANIZATION_SELLERPARTYIDENTIFIER	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRY	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRYCODE	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_STREETNAME	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_TOWNNAME	string			
SENDERADDRESSLINE	string	Sender Addressline		
SENDERCITY	string	Sender City		
SENDERCONTACTNAME	string	Sender Contact Name		
SENDERCOUNTRY	string	Sender Country		
SENDERNAME	string	Sender Name		
SENDERPOSTALCODE	string	Sender Postal Code		
SENDERSTATE	string	Sender State		
SENDERVATNUMBER	string	Sender VAT Number		
TAXFREEDESCRIPTION	string	Tax Free Description		
TITLE	string	Title		
VIRTUALBANKBARCODE	string	Virtual Bank Barcode		

Name	Data Type	Label	Required	Documentation
YOURREFERENCE	string	Your Reference		

3.1.148 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	string	<input type="checkbox"/>		
status	string	<input type="checkbox"/>		

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	string	Account GUID		
ACCOUNTNAME	string	Account Name		
ACCOUNTNUMBER	int64	Account Number		
BUSINESSUNITCODE	string	Business Unit Code		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		

Name	Data Type	Label	Required	Documentation
BUYERORGANIZATION_BUYERCONTACTPERSONNAME	string			
BUYERORGANIZATION_BUYERELECTRONICINVOICEADDRESS	string			
BUYERORGANIZATION_BUYERELECTRONICINVOICEOPERATOR	string			
BUYERORGANIZATION_BUYEREMAILADDRESSIDENTIFIER	string			
BUYERORGANIZATION_BUYERORGANIZATIONNAME	string			
BUYERORGANIZATION_BUYERORGANIZATIONTAXCODE	string			
BUYERORGANIZATION_BUYERPARTYIDENTIFIER	string			
BUYERORGANIZATION_BUYERPHONENUMBERIDENTIFIER	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRY	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRYCODE	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_STREETNAME	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_TOWNNAME	string			
CULTURE	string	Culture		
CURRENCYGUID	string	Currency GUID		
CURRENCYNAME	string	Currency Name		
CURRENCYRATE	decimal	Currency Rate		
CURRENCYSHORTFORM	string	Currency Short Form		
DATE	datetime	Date		
DESCRIPTION	string	Ddescription		
DUEDATE	datetime	Due Date		
ENTRYDATE	datetime	Entry Date		
GUID	string	GUID		
INVOICEALLOWEDCATEGORIESGUIDS	string			
INVOICECONFIG_BILLEXPENSE	string			
INVOICECONFIG_BILLHOUR	string			
INVOICECONFIG_BILLPRODUCT	string			
INVOICECONFIG_EXPENSEITEMDESCRIPTIONFORMAT	string			

Name	Data Type	Label	Required	Documentation
INVOICECONFIG_HOURITEMDESCRIPTIONFORMAT	string			
INVOICECONFIG_HOURPRIMARYGROUPBY	string			
INVOICECONFIG_HOURSECONDARYGROUPBY	string			
INVOICECONFIG_ISTAXFREE	boolean			
INVOICECONFIG_PRODUCTITEMDESCRIPTIONFORMAT	string			
INVOICECONFIG_PRODUCTPRIMARYGROUPBY	string			
INVOICECONFIG_PRODUCTSECONDARYGROUPBY	string			
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	string			
INVOICECONFIG_TRAVELPRIMARYGROUPBY	string			
INVOICECONFIG_TRAVELSECONDARYGROUPBY	string			
INVOICEDETAILS_INVOICEDATE	datetime			
INVOICEDETAILS_INVOICEDUEDATE	datetime			
INVOICEDETAILS_INVOICENUMBER	int32			
INVOICEDETAILS_INVOICETOTALVATAMOUNT	decimal			
INVOICEDETAILS_INVOICETOTALVATEXCLUDEDAMOUNT	decimal			
INVOICEDETAILS_INVOICETOTALVATINCLUDEDAMOUNT	decimal			
INVOICEDETAILS_INVOICETYPETEXT	string			

Name	Data Type	Label	Required	Documentation
INVOICEDetails_OrderIdentifier	string			
INVOICEDetails_OriginCode	string			
INVOICEDetails_PaymentOverdueDefinitionText	string			
INVOICEDetails_PaymentOverdueDefinitionPercent	decimal			
INVOICEDetails_PaymentTermDefinitionText	string			
INVOICEDetails_SellerReferenceIdentifier	string			
INVOICEDetails_VATBaseAmount	decimal			
INVOICEDetails_VATRateAmount	decimal			
INVOICEDetails_VATRatePercent	int32			
InvoiceNumberingBusinessUnitGUID	string	Invoice Numbering Business Unit GUID		
INVOICERows	string			
INVOICEStatus	string	Invoice Status		
ISCreditNote	boolean	Is Credit Note		
ISReimbursed	boolean	Is Reimbursed		
ISTaxFree	boolean	Is Tax Free		
LanguageCode	string	Language Code		
LanguageGUID	string	Language GUID		
MainCaseGUID	string	Main Case GUID		
NOTES	string	Notes		
NOTESAfterInvoiceRows	string	Notes after Invoice Lines		
NOTESBeforeInvoiceRows	string	Notes before Invoice Lines		
NUMBER	int32	Number		
ORDERNumber	string	Order Number		
OURReference	string	Our Reference		
OVERDUEInterest	decimal	Overdue Interest		
PAYMENTDate	datetime	Payment Date		
PAYMENTStatus	string	Payment Status		
PAYMENTTerm	int32	Payment Term		
PONumber	string	PO Number		
RECEIVERAddressLine	string	Receiver Addressline		
RECEIVERCity	string	Receiver City		
RECEIVERContactEmail	string	Receiver Contact Email		
RECEIVERContactName	string	Receiver Contact Name		

Name	Data Type	Label	Required	Documentation
RECEIVERCOUNTRY	string	Receiver Country		
RECEIVERCUSTOMERNAME	string	Receiver Customer Name		
RECEIVERPOSTALCODE	string	Receiver Postal Code		
RECEIVERSTATE	string	Receiver State		
RECEIVERVATNUMBER	string	Receiver VAT Number		
REFERENCENUMBER	string	Reference Number		
REIMBURSEINVOICEGUID	string	Reimburse Invoice GUID		
SELLERCONTACT_SELLERACCOUNTDETAILS	string			
SELLERCONTACT_SELLERCONTACTPERSONNAME	string			
SELLERCONTACT_SELLEREMAILADDRESSIDENTIFIER	string			
SELLERCONTACT_SELLERFAXNUMBER	string			
SELLERCONTACT_SELLERFREEXT	string			
SELLERCONTACT_SELLERHOMETOWNNAME	string			
SELLERCONTACT_SELLERPHONE NUMBER	string			
SELLERCONTACT_SELLERWEBADDRESSIDENTIFIER	string			
SELLERORGANIZATION_SELLERINVOICEADDRESS	string			
SELLERORGANIZATION_SELLERINVOICEOPERATOR	string			
SELLERORGANIZATION_SELLERORGANISATIONNAME	string			
SELLERORGANIZATION_SELLERORGANISATIONTAXCODE	string			
SELLERORGANIZATION_SELLERPARTYIDENTIFIER	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRY	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRYCODE	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_STREETNAME	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_TOWNNAME	string			

Name	Data Type	Label	Required	Documentation
SENDERADDRESSLINE	string	Sender Addressline		
SENDERCITY	string	Sender City		
SENDERCONTACTNAME	string	Sender Contact Name		
SENDERCOUNTRY	string	Sender Country		
SENDERNAME	string	Sender Name		
SENDERPOSTALCODE	string	Sender Postal Code		
SENDERSTATE	string	Sender State		
SENDERVATNUMBER	string	Sender VAT Number		
TAXFREEDESCRIPTION	string	Tax Free Description		
TITLE	string	Title		
VIRTUALBANKBARCODE	string	Virtual Bank Barcode		
YOURREFERENCE	string	Your Reference		

3.1.149 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	string	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	string	Name		
SORTORDER	int32	Sort Order		

3.1.150 ItemByGUID: Visma Severa Item by GUID

Catalog: Severa

Schema: API

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	string	<input type="checkbox"/>		

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	string	Billing Dependency Task GUID		
BILLINGSCHEDULE	string	Billing Schedule		
CASEGUID	string	Project GUID		
CASENUMBER	int64	Project Number		
COSTCENTERGUID	string	Cost Center GUID		
COSTCENTERIDENTIFIER	string	Cost Center Identifier		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User Company Currency		

Name	Data Type	Label	Required	Documentation
DESCRIPTION	string	Ddescription		
DISPLAYPERIODSTART	datetime	Display Period Start		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVICEROWGUID	string	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	string	Measurement Unit		
NAME	string	Name		
ORDERDATE	datetime	Order Date		
PLANNEDBILLINGDATE	datetime	Planned Billing Date		
PRICECURRENCYSHORTFORMOF ASECOMPANYCURRENCY	string	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOF INVOICINGCURRENCY	string	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOF ORGANIZATIONCURRENCY	string	Price Currency Short Form of Organization Currency		
PRICECURRENCYSHORTFORMOF USERCOMPANYCURRENCY	string	Price Currency Short Form of User Company Currency		
PRODUCTCODE	string	Product Code		
PRODUCTGUID	string	Product GUID		
PRODUCTTYPE	string	Product Type		
PURCHASEORDERNUMBER	string	Purchase Order Number		
PURCHASEVAT	decimal	Purchase VAT		
QUANTITY	decimal	Quantity		
RECURRENCEENDDATE	datetime	Recurrence End Date		
RECURRENCEENDTYPE	int32	Recurrence End Type		
RECURRENCEFREQUENCY	int32	Recurrence Frequency		
RECURRENCESTARTDATE	datetime	Recurrence Start Date		
RECURRENCETIMES	int32	Recurrence Times		
RECURRINGITEMGUID	string	Recurring Item GUID		
SALESAccountGUID	string	Sales Account GUID		

Name	Data Type	Label	Required	Documentation
SALESSTATUSGUID	string	Sales Status GUID		
TASKGUID	string	Task GUID		
TRAVELENDTIME	datetime	Travel End Time		
TRAVELSTARTTIME	datetime	Travel Start Time		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	decimal	Unit Cost in Organization Currency		
UNITCOSTINUSERCOMPANYCURRENCY	decimal	Unit Cost in User Company Currency		
UNITCOSTVATINCLUDEDAMOUNT	decimal	Unit Cost VAT Included Amount		
UNITPRICE	decimal	Unit Price		
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		
UNITPRICEINPROJECTCOMPANYCURRENCY	decimal	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	decimal	Unit Price in User Company Currency		
USERCODE	string	User Code		
USERGUID	string	User GUID		
VAT	decimal	VAT		

3.1.151 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.

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	string	<input type="checkbox"/>		

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	string	Billing Dependency Task GUID		
BILLINGSCHEDULE	string	Billing Schedule		
CASEGUID	string	Project GUID		
CASENUMBER	int64	Project Number		
COSTCENTERGUID	string	Cost Center GUID		
COSTCENTERIDENTIFIER	string	Cost Center Identifier		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User Company Currency		
DESCRIPTION	string	Ddescription		
DISPLAYPERIODSTART	datetime	Display Period Start		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVICEROWGUID	string	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	string	Measurement Unit		
NAME	string	Name		

Name	Data Type	Label	Required	Documentation
ORDERDATE	datetime	Order Date		
PLANNEDBILLINGDATE	datetime	Planned Billing Date		
PRICECURRENCY SHORTFORM OF COMPANY CURRENCY	string	Price Currency Short Form of Project Company Currency		
PRICECURRENCY SHORTFORM OF INVOICING CURRENCY	string	Price Currency Short Form of Invoicing Currency		
PRICECURRENCY SHORTFORM OF ORGANIZATION CURRENCY	string	Price Currency Short Form of Organization Currency		
PRICECURRENCY SHORTFORM OF USER COMPANY CURRENCY	string	Price Currency Short Form of User Company Currency		
PRODUCTCODE	string	Product Code		
PRODUCTGUID	string	Product GUID		
PRODUCTTYPE	string	Product Type		
PURCHASEORDERNUMBER	string	Purchase Order Number		
PURCHASEVAT	decimal	Purchase VAT		
QUANTITY	decimal	Quantity		
RECURRENCEENDDATE	datetime	Recurrence End Date		
RECURRENCEENDTYPE	int32	Recurrence End Type		
RECURRENCEFREQUENCY	int32	Recurrence Frequency		
RECURRENCESTARTDATE	datetime	Recurrence Start Date		
RECURRENCETIMES	int32	Recurrence Times		
RECURRINGITEMGUID	string	Recurring Item GUID		
SALESACCOUNTGUID	string	Sales Account GUID		
SALESSTATUSGUID	string	Sales Status GUID		
TASKGUID	string	Task GUID		
TRAVELENDTIME	datetime	Travel End Time		
TRAVELSTARTTIME	datetime	Travel Start Time		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	decimal	Unit Cost in Organization Currency		
UNITCOSTINUSERCOMPANYCURRENCY	decimal	Unit Cost in User Company Currency		
UNITCOSTVATINCLUDEDAMOUNT	decimal	Unit Cost VAT Included Amount		
UNITPRICE	decimal	Unit Price		
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		
UNITPRICEINCOMPANYCURRENCY	decimal	Unit Price in Project Company Currency		

Name	Data Type	Label	Required	Documentation
UNITPRICEINUSERCOMPANYCURRENCY	decimal	Unit Price in User Company Currency		
USERCODE	string	User Code		
USERGUID	string	User GUID		
VAT	decimal	VAT		

3.1.152 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	string	<input type="checkbox"/>		

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	string	Billing Dependency Task GUID		
BILLINGSCHEDULE	string	Billing Schedule		
CASEGUID	string	Project GUID		

Name	Data Type	Label	Required	Documentation
CASENUMBER	int64	Project Number		
COSTCENTERGUID	string	Cost Center GUID		
COSTCENTERIDENTIFIER	string	Cost Center Identifier		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User Company Currency		
DESCRIPTION	string	Description		
DISPLAYPERIODSTART	datetime	Display Period Start		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVICEROWGUID	string	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	string	Measurement Unit		
NAME	string	Name		
ORDERDATE	datetime	Order Date		
PLANNEDBILLINGDATE	datetime	Planned Billing Date		
PRICECURRENCYSHORTFORMOFPROJECTCOMPANYCURRENCY	string	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Price Currency Short Form of Organization Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Price Currency Short Form of User Company Currency		
PRODUCTCODE	string	Product Code		
PRODUCTGUID	string	Product GUID		
PRODUCTTYPE	string	Product Type		

Name	Data Type	Label	Required	Documentation
PURCHASEORDERNUMBER	string	Purchase Order Number		
PURCHASEVAT	decimal	Purchase VAT		
QUANTITY	decimal	Quantity		
RECURRENCEENDDATE	datetime	Recurrence End Date		
RECURRENCEENDTYPE	int32	Recurrence End Type		
RECURRENCEFREQUENCY	int32	Recurrence Frequency		
RECURRENCESTARTDATE	datetime	Recurrence Start Date		
RECURRENCETIMES	int32	Recurrence Times		
RECURRINGITEMGUID	string	Recurring Item GUID		
SALESACCOUNTGUID	string	Sales Account GUID		
SALESSTATUSGUID	string	Sales Status GUID		
TASKGUID	string	Task GUID		
TRAVELENDTIME	datetime	Travel End Time		
TRAVELSTARTTIME	datetime	Travel Start Time		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	decimal	Unit Cost in Organization Currency		
UNITCOSTINUSERCOMPANYCURRENCY	decimal	Unit Cost in User Company Currency		
UNITCOSTVATINCLUDEDAMOUNT	decimal	Unit Cost VAT Included Amount		
UNITPRICE	decimal	Unit Price		
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		
UNITPRICEINPROJECTCOMPANYCURRENCY	decimal	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	decimal	Unit Price in User Company Currency		
USERCODE	string	User Code		
USERGUID	string	User GUID		
VAT	decimal	VAT		

3.1.153 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.

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	datetime	<input type="checkbox"/>		
insertStartDate	datetime	<input type="checkbox"/>		

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	string	Billing Dependency Task GUID		
BILLINGSCHEDULE	string	Billing Schedule		
CASEGUID	string	Project GUID		
CASENUMBER	int64	Project Number		
COSTCENTERGUID	string	Cost Center GUID		
COSTCENTERIDENTIFIER	string	Cost Center Identifier		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User Company Currency		
DESCRIPTION	string	Ddescription		
DISPLAYPERIODSTART	datetime	Display Period Start		
EVENTDATE	datetime	Event Date		

Name	Data Type	Label	Required	Documentation
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVOICEROWGUID	string	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	string	Measurement Unit		
NAME	string	Name		
ORDERDATE	datetime	Order Date		
PLANNEDBILLINGDATE	datetime	Planned Billing Date		
PRICECURRENCYSHORTFORMOF COMPANYS CURRENCY	string	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOF INVOICING CURRENCY	string	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOF ORGANIZATION CURRENCY	string	Price Currency Short Form of Organization Currency		
PRICECURRENCYSHORTFORMOF USER COMPANY CURRENCY	string	Price Currency Short Form of User Company Currency		
PRODUCTCODE	string	Product Code		
PRODUCTGUID	string	Product GUID		
PRODUCTTYPE	string	Product Type		
PURCHASEORDERNUMBER	string	Purchase Order Number		
PURCHASEVAT	decimal	Purchase VAT		
QUANTITY	decimal	Quantity		
RECURRENCEENDDATE	datetime	Recurrence End Date		
RECURRENCEENDTYPE	int32	Recurrence End Type		
RECURRENCEFREQUENCY	int32	Recurrence Frequency		
RECURRENCESTARTDATE	datetime	Recurrence Start Date		
RECURRENCETIMES	int32	Recurrence Times		
RECURRINGITEMGUID	string	Recurring Item GUID		
SALESACCOUNTGUID	string	Sales Account GUID		
SALESSTATUSGUID	string	Sales Status GUID		
TASKGUID	string	Task GUID		
TRAVELENDTIME	datetime	Travel End Time		

Name	Data Type	Label	Required	Documentation
TRAVELSTARTTIME	datetime	Travel Start Time		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	decimal	Unit Cost in Organization Currency		
UNITCOSTINUSERCOMPANYCURRENCY	decimal	Unit Cost in User Company Currency		
UNITCOSTVATINCLUDEDAMOUNT	decimal	Unit Cost VAT Included Amount		
UNITPRICE	decimal	Unit Price		
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		
UNITPRICEINPROJECTCOMPANYCURRENCY	decimal	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	decimal	Unit Price in User Company Currency		
USERCODE	string	User Code		
USERGUID	string	User GUID		
VAT	decimal	VAT		

3.1.154 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.

Name	Data Type	Required	Default Value	Documentation
startDate	datetime	<input type="checkbox"/>		

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	string	Billing Dependency Task GUID		
BILLINGSCHEDULE	string	Billing Schedule		
CASEGUID	string	Project GUID		
CASENUMBER	int64	Project Number		
COSTCENTERGUID	string	Cost Center GUID		
COSTCENTERIDENTIFIER	string	Cost Center Identifier		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User Company Currency		
DESCRIPTION	string	Ddescription		
DISPLAYPERIODSTART	datetime	Display Period Start		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVOICEROWGUID	string	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	string	Measurement Unit		
NAME	string	Name		
ORDERDATE	datetime	Order Date		
PLANNEDBILLINGDATE	datetime	Planned Billing Date		
PRICECURRENCYSHORTFORMOFPROJECTCOMPANYCURRENCY	string	Price Currency Short Form of Project		

Name	Data Type	Label	Required	Documentation
		Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Price Currency Short Form of Organization Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Price Currency Short Form of User Company Currency		
PRODUCTCODE	string	Product Code		
PRODUCTGUID	string	Product GUID		
PRODUCTTYPE	string	Product Type		
PURCHASEORDERNUMBER	string	Purchase Order Number		
PURCHASEVAT	decimal	Purchase VAT		
QUANTITY	decimal	Quantity		
RECURRENCEENDDATE	datetime	Recurrence End Date		
RECURRENCEENDTYPE	int32	Recurrence End Type		
RECURRENCEFREQUENCY	int32	Recurrence Frequency		
RECURRENCESTARTDATE	datetime	Recurrence Start Date		
RECURRENCETIMES	int32	Recurrence Times		
RECURRINGITEMGUID	string	Recurring Item GUID		
SALESACCOUNTGUID	string	Sales Account GUID		
SALESSTATUSGUID	string	Sales Status GUID		
TASKGUID	string	Task GUID		
TRAVELENDTIME	datetime	Travel End Time		
TRAVELSTARTTIME	datetime	Travel Start Time		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	decimal	Unit Cost in Organization Currency		
UNITCOSTINUSERCOMPANYCURRENCY	decimal	Unit Cost in User Company Currency		
UNITCOSTVATINCLUDEDAMOUNT	decimal	Unit Cost VAT Included Amount		
UNITPRICE	decimal	Unit Price		
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		
UNITPRICEINPROJECTCOMPANYCURRENCY	decimal	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	decimal	Unit Price in User Company Currency		
USERCODE	string	User Code		

Name	Data Type	Label	Required	Documentation
USERGUID	string	User GUID		
VAT	decimal	VAT		

3.1.155 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	string	<input type="checkbox"/>		

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	string	GUID		
IETFLANGUAGETAG	string	IETF Language Tag		
NAME	string	Name		

3.1.156 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	string	<input type="checkbox"/>		

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	string	GUID		
IETFLANGUAGETAG	string	IETF Language Tag		
NAME	string	Name		

3.1.157 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.

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	string	<input type="checkbox"/>		

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	string	GUID		
ISACTIVE	boolean	Is Active		
NAME	string	Name		

3.1.158 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	string	GUID		
ISACTIVE	boolean	Is Active		

Name	Data Type	Label	Required	Documentation
NAME	string	Name		

3.1.159 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	string	<input type="checkbox"/>		

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	string	Code		
GUID	string	GUID		
INCLUDEINFLEXTIME	boolean	Include in Flextime		
ISACTIVE	boolean	Is Active		
ISWORKTIME	boolean	Is Work Time		
NAME	string	Name		
PERCENTAGE	decimal	Percentage		

3.1.160 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	string	Code		
GUID	string	GUID		
INCLUDEINFLEXTIME	boolean	Include in Flextime		
ISACTIVE	boolean	Is Active		
ISWORKTIME	boolean	Is Work Time		
NAME	string	Name		
PERCENTAGE	decimal	Percentage		

3.1.161 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

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	string	<input type="checkbox"/>		

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	string	Project GUID		
CODE	string	Code		
DEADLINE	datetime	Deadline		
DESCRIPTION	string	Ddescription		
FLATRATE_ADDITIONALHOURSUNITPRICE	decimal	Flat Rate Additional Hours Unit Price		
FLATRATE_AREADDITIONALHOURSBILLABLE	boolean	Flat Rate Are additional Hours Billable		
FLATRATE_HOURS	decimal	Flat Rate Hours		
FLATRATE_PRICE	decimal	Flat Rate Price		
GUID	string	GUID		
ISCOMPLETED	boolean	Is Completed		
ISLOCKED	boolean	Is Locked		
NAME	string	Name		
ORIGINALDEADLINE	datetime	Original Deadline		
ORIGINALPLANNEDSTARTDATE	datetime	Original Planned Start Date		
ORIGINALWORKESTIMATE	decimal	Original Work Estimate		
OWNERUSERGUID	string	Owner User GUID		
PARENTPHASEGUID	string	Parent Phase GUID		
PLANNEDSTARTDATE	datetime	Planned Start Date		
PRICEPERHOUR	decimal	Price per Hour		
SORTORDER	int32	Sort Order		
WORKESTIMATE	decimal	Work Estimate		
WORKTYPEGUID	string	Work Type GUID		

3.1.162 PhaseMembersByPhaseGUID: Visma Severa Phase Members by Phase GUID

Catalog: Severa

Schema: API

Label: Phase Members 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- 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	boolean	<input type="checkbox"/>		
phaseGUID	string	<input type="checkbox"/>		

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	string			

3.1.163 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

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	string	<input type="checkbox"/>		

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	string	Project GUID		
CODE	string	Code		
DEADLINE	datetime	Deadline		
DESCRIPTION	string	Ddescription		
FLA TRATE_ADDITIONALHOURSUNITPRICE	decimal	Flat Rate Additional Hours Unit Price		
FLA TRATE_AREADDITIONALHOURS BILLABLE	boolean	Flat Rate Are additional Hours Billable		
FLA TRATE_HOURS	decimal	Flat Rate Hours		
FLA TRATE_PRICE	decimal	Flat Rate Price		
GUID	string	GUID		
ISCOMPLETED	boolean	Is Completed		
ISLOCKED	boolean	Is Locked		
NAME	string	Name		
ORIGINALDEADLINE	datetime	Original Deadline		
ORIGINALPLANNEDSTARTDATE	datetime	Original Planned Start Date		
ORIGINALWORKESTIMATE	decimal	Original Work Estimate		
OWNERUSERGUID	string	Owner User GUID		
PARENTPHASEGUID	string	Parent Phase GUID		
PLANNEDSTARTDATE	datetime	Planned Start Date		
PRICEPERHOUR	decimal	Price per Hour		
SORTORDER	int32	Sort Order		
WORKESTIMATE	decimal	Work Estimate		

Name	Data Type	Label	Required	Documentation
WORKTYPEGUID	string	Work Type GUID		

3.1.164 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	string	<input type="checkbox"/>		
phaseCode	string	<input type="checkbox"/>		

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	string	Project GUID		
CODE	string	Code		
DEADLINE	datetime	Deadline		
DESCRIPTION	string	Ddescription		
FLATRATE_ADDITIONALHOURSUNITPRICE	decimal	Flat Rate Additional Hours Unit Price		

Name	Data Type	Label	Required	Documentation
FLATRATE_AREADDITIONALHOURS BILLABLE	boolean	Flat Rate Are additional Hours Billable		
FLATRATE_HOURS	decimal	Flat Rate Hours		
FLATRATE_PRICE	decimal	Flat Rate Price		
GUID	string	GUID		
ISCOMPLETED	boolean	Is Completed		
ISLOCKED	boolean	Is Locked		
NAME	string	Name		
ORIGINALDEADLINE	datetime	Original Deadline		
ORIGINALPLANNEDSTARTDATE	datetime	Original Planned Start Date		
ORIGINALWORKESTIMATE	decimal	Original Work Estimate		
OWNERUSERGUID	string	Owner User GUID		
PARENTPHASEGUID	string	Parent Phase GUID		
PLANNEDSTARTDATE	datetime	Planned Start Date		
PRICEPERHOUR	decimal	Price per Hour		
SORTORDER	int32	Sort Order		
WORKESTIMATE	decimal	Work Estimate		
WORKTYPEGUID	string	Work Type GUID		

3.1.165 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

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	string	<input type="checkbox"/>		
options	int32	<input type="checkbox"/>		
startDateTime	datetime	<input type="checkbox"/>		

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	string	Project GUID		
CODE	string	Code		
DEADLINE	datetime	Deadline		
DESCRIPTION	string	Ddescription		
FLATRATE_ADDITIONALHOURSUNITPRICE	decimal	Flat Rate Additional Hours Unit Price		
FLATRATE_AREADDITIONALHOURSBILLABLE	boolean	Flat Rate Are additional Hours Billable		
FLATRATE_HOURS	decimal	Flat Rate Hours		
FLATRATE_PRICE	decimal	Flat Rate Price		
GUID	string	GUID		
ISCOMPLETED	boolean	Is Completed		
ISLOCKED	boolean	Is Locked		
NAME	string	Name		
ORIGINALDEADLINE	datetime	Original Deadline		
ORIGINALPLANNEDSTARTDATE	datetime	Original Planned Start Date		
ORIGINALWORKESTIMATE	decimal	Original Work Estimate		
OWNERUSERGUID	string	Owner User GUID		
PARENTPHASEGUID	string	Parent Phase GUID		
PLANNEDSTARTDATE	datetime	Planned Start Date		
PRICEPERHOUR	decimal	Price per Hour		
SORTORDER	int32	Sort Order		
WORKESTIMATE	decimal	Work Estimate		
WORKTYPEGUID	string	Work Type GUID		

3.1.166 PricelistByGUID: Visma Severa Price List by GUID

Catalog: Severa

Schema: API

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	string	<input type="checkbox"/>		

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	string	Currency Code		
CURRENCYGUID	string	Currency GUID		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
ISINTERNAL	boolean	Is Internal		
ISVOLUMEPRICING	boolean	Is Volume Pricing		
NAME	string	Name		

3.1.167 Pricelists: Visma Severa Price Lists

Catalog: Severa

Schema: API

Label: Price Lists

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	string	Currency Code		
CURRENCYGUID	string	Currency GUID		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
ISINTERNAL	boolean	Is Internal		
ISVOLUMEPRICING	boolean	Is Volume Pricing		
NAME	string	Name		

3.1.168 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	string	<input type="checkbox"/>		

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	string	Code		
DESCRIPTION	string	Ddescription		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
MEASUREMENTUNIT	string	Measurement Unit		
NAME	string	Name		
PRODUCTCATEGORY GUID	string	Product Category GUID		
SALESACCOUNTGUID	string	Sales Account GUID		
TYPE	string	Type		
UNITCOST	decimal	Unit Cost		
UNITPRICE	decimal	Unit Price		
USEINWORKTIMEENTRY	boolean	Use in Work Time Entry		
VAT	decimal	VAT		

3.1.169 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

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	string	<input type="checkbox"/>		

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	string	Code		
DESCRIPTION	string	Ddescription		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
MEASUREMENTUNIT	string	Measurement Unit		
NAME	string	Name		
PRODUCTCATEGORY GUID	string	Product Category GUID		
SALESACCOUNTGUID	string	Sales Account GUID		
TYPE	string	Type		
UNITCOST	decimal	Unit Cost		
UNITPRICE	decimal	Unit Price		
USEINWORKTIMEENTRY	boolean	Use in Work Time Entry		
VAT	decimal	VAT		

3.1.170 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.

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	string	<input type="checkbox"/>		

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	string	Code		
DESCRIPTION	string	Ddescription		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
MEASUREMENTUNIT	string	Measurement Unit		
NAME	string	Name		
PRODUCTCATEGORY GUID	string	Product Category GUID		
SALESACCOUNTGUID	string	Sales Account GUID		
TYPE	string	Type		
UNITCOST	decimal	Unit Cost		
UNITPRICE	decimal	Unit Price		
USEINWORKTIMEENTRY	boolean	Use in Work Time Entry		
VAT	decimal	VAT		

3.1.171 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

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	int32	<input type="checkbox"/>		
startDateTime	datetime	<input type="checkbox"/>		

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	string	Code		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
NAME	string	Name		

3.1.172 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.

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	string	<input type="checkbox"/>		

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	string	Code		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
NAME	string	Name		

3.1.173 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.

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	string	<input type="checkbox"/>		

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	string	Code		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
NAME	string	Name		

3.1.174 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 Invariant UniversalSQL driver 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.

Name	Data Type	Required	Default Value	Documentation
includeDefault	boolean	<input type="checkbox"/>		
productGUID	string	<input type="checkbox"/>		

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	string	GUID		
ISACTIVE	boolean	Is Active		
ISTRAVELTYPEDEFAULT	boolean	Is Travel Type Default		
NAME	string	Name		
NUMBER	string	Number		
TAXPERCENTAGE	decimal	Tax (%)		

3.1.175 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	int32	<input type="checkbox"/>		

Name	Data Type	Required	Default Value	Documentation
startDateTime	datetime	<input type="checkbox"/>		

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	string	Code		
DESCRIPTION	string	Ddescription		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
MEASUREMENTUNIT	string	Measurement Unit		
NAME	string	Name		
PRODUCTCATEGORY GUID	string	Product Category GUID		
SALESACCOUNTGUID	string	Sales Account GUID		
TYPE	string	Type		
UNITCOST	decimal	Unit Cost		
UNITPRICE	decimal	Unit Price		
USEINWORKTIMEENTRY	boolean	Use in Work Time Entry		
VAT	decimal	VAT		

3.1.176 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

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	string	<input type="checkbox"/>		
enddate	datetime	<input type="checkbox"/>		
inCreditNote	boolean	<input type="checkbox"/>		
startdate	datetime	<input type="checkbox"/>		

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	string	Account GUID		
CASEBUSINESSUNITCODE	string	Project Business Unit Code		
CASECOSTCENTERIDENTIFIER	string	Project Cost Center Identifier		
CASEGUID	string	Project GUID		
CASENAME	string	Project Name		
CASENUMBER	int64	Project Number		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User Company Currency		
DESCRIPTION	string	Ddescription		
ENDTIME	datetime	End Time		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVICEROWGUID	string	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		

Name	Data Type	Label	Required	Documentation
ISBILLABLE	boolean	Is Billable		
ISPRODUCTIVE	boolean	Is Productive		
ISREADONLY	boolean	Is Read-only		
OVERTIMECODE	string	Overtime Code		
OVERTIMEGUID	string	Overtime GUID		
PHASEGUID	string	Phase GUID		
PRICECURRENCYSHORTFORMOF COMPANYS CURRENCY	string	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOF INVOICINGCURRENCY	string	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOF ORGANIZATIONCURRENCY	string	Price Currency Short Form of Organization Currency		
PRICECURRENCYSHORTFORMOF USERCOMPANYS CURRENCY	string	Price Currency Short Form of User Company Currency		
QUANTITY	decimal	Quantity		
STARTTIME	datetime	Start Time		
TASKGUID	string	Task GUID		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	decimal	Unit Cost in Organization Currency		
UNITCOSTINUSERCOMPANYS CURRENCY	decimal	Unit Cost in User Company Currency		
UNITPRICE	decimal	Unit Price		
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		
UNITPRICEINCOMPANYS CURRENCY	decimal	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYS CURRENCY	decimal	Unit Price in User Company Currency		
USEINVOICEQUANTITY	boolean			
USERBUSINESSUNITCODE	string	User Business Unit Code		
USERCODE	string	User Code		
USERFIRSTNAME	string	User First Name		
USERGUID	string	User GUID		
USERLASTNAME	string	User Last Name		
WORKTYPECODE	string	Work Type Code		
WORKTYPEGUID	string	Work Type GUID		

3.1.177 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 Invariantive UniversalSQL driver 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	boolean	<input type="checkbox"/>		
startDate	datetime	<input type="checkbox"/>		

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	string	Account GUID		
CASEBUSINESSUNITCODE	string	Project Business Unit Code		
CASECOSTCENTERIDENTIFIER	string	Project Cost Center Identifier		
CASEGUID	string	Project GUID		
CASENAME	string	Project Name		
CASENUMBER	int64	Project Number		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		

Name	Data Type	Label	Required	Documentation
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User Company Currency		
DESCRIPTION	string	Description		
ENDTIME	datetime	End Time		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVICEROWGUID	string	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISPRODUCTIVE	boolean	Is Productive		
ISREADONLY	boolean	Is Read-only		
OVERTIMECODE	string	Overtime Code		
OVERTIMEGUID	string	Overtime GUID		
PHASEGUID	string	Phase GUID		
PRICECURRENCYSHORTFORMOFPROJECTCOMPANYCURRENCY	string	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Price Currency Short Form of Organization Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Price Currency Short Form of User Company Currency		
QUANTITY	decimal	Quantity		
STARTTIME	datetime	Start Time		
TASKGUID	string	Task GUID		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	decimal	Unit Cost in Organization Currency		
UNITCOSTINUSERCOMPANYCURRENCY	decimal	Unit Cost in User Company Currency		
UNITPRICE	decimal	Unit Price		

Name	Data Type	Label	Required	Documentation
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		
UNITPRICEINCOMPANYCURRENCY	decimal	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	decimal	Unit Price in User Company Currency		
USEINVOICEQUANTITY	boolean			
USERBUSINESSUNITCODE	string	User Business Unit Code		
USERCODE	string	User Code		
USERFIRSTNAME	string	User First Name		
USERGUID	string	User GUID		
USERLASTNAME	string	User Last Name		
WORKTYPECODE	string	Work Type Code		
WORKTYPEGUID	string	Work Type GUID		

3.1.178 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	string	<input type="checkbox"/>		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ACCOUNTGUID	string	Account GUID		
CASEBUSINESSUNITCODE	string	Project Business Unit Code		
CASECOSTCENTERIDENTIFIER	string	Project Cost Center Identifier		
CASEGUID	string	Project GUID		
CASENAME	string	Project Name		
CASENUMBER	int64	Project Number		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User Company Currency		
DESCRIPTION	string	Ddescription		
ENDTIME	datetime	End Time		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVICEROWGUID	string	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISPRODUCTIVE	boolean	Is Productive		
ISREADONLY	boolean	Is Read-only		
OVERTIMECODE	string	Overtime Code		
OVERTIMEGUID	string	Overtime GUID		
PHASEGUID	string	Phase GUID		
PRICECURRENCYSHORTFORMOFPROJECTCOMPANYCURRENCY	string	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Price Currency Short Form of Organization		

Name	Data Type	Label	Required	Documentation
		Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Price Currency Short Form of User Company Currency		
QUANTITY	decimal	Quantity		
STARTTIME	datetime	Start Time		
TASKGUID	string	Task GUID		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	decimal	Unit Cost in Organization Currency		
UNITCOSTINUSERCOMPANYCURRENCY	decimal	Unit Cost in User Company Currency		
UNITPRICE	decimal	Unit Price		
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		
UNITPRICEINPROJECTCOMPANYCURRENCY	decimal	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	decimal	Unit Price in User Company Currency		
USEINVOICEQUANTITY	boolean			
USERBUSINESSUNITCODE	string	User Business Unit Code		
USERCODE	string	User Code		
USERFIRSTNAME	string	User First Name		
USERGUID	string	User GUID		
USERLASTNAME	string	User Last Name		
WORKTYPECODE	string	Work Type Code		
WORKTYPEGUID	string	Work Type GUID		

3.1.179 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.

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	string	<input type="checkbox"/>		

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	string	Billing Dependency Task GUID		
BILLINGSCHEDULE	string	Billing Schedule		
CASEGUID	string	Project GUID		
CASENUMBER	int64	Project Number		
COSTCENTERGUID	string	Cost Center GUID		
COSTCENTERIDENTIFIER	string	Cost Center Identifier		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User Company Currency		
DESCRIPTION	string	Ddescription		
DISPLAYPERIODSTART	datetime	Display Period Start		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVICEROWGUID	string	Invoice Line GUID		

Name	Data Type	Label	Required	Documentation
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISOUTSOURCINGCOST	boolean	Is Out-sourcing Cost		
ISREADONLY	boolean	Is Read-only		
ISREALISED	boolean	Is Realized		
MEASUREMENTUNIT	string	Measurement Unit		
NAME	string	Name		
ORDERDATE	datetime	Order Date		
PLANNEDBILLINGDATE	datetime	Planned Billing Date		
PRICECURRENCYSHORTFORMOFC ASECOMPANYCURRENCY	string	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFI NVOICINGCURRENCY	string	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFO RGANIZATIONCURRENCY	string	Price Currency Short Form of Organization Currency		
PRICECURRENCYSHORTFORMOFU SERCOMPANYCURRENCY	string	Price Currency Short Form of User Company Currency		
PRODUCTCODE	string	Product Code		
PRODUCTGUID	string	Product GUID		
PRODUCTTYPE	string	Product Type		
PURCHASEORDERNUMBER	string	Purchase Order Number		
PURCHASEVAT	decimal	Purchase VAT		
QUANTITY	decimal	Quantity		
RECURRENCEENDDATE	datetime	Recurrence End Date		
RECURRENCEENDTYPE	int32	Recurrence End Type		
RECURRENCFREQUENCY	int32	Recurrence Frequency		
RECURRENCESTARTDATE	datetime	Recurrence Start Date		
RECURRENCETIMES	int32	Recurrence Times		
RECURRINGITEMGUID	string	Recurring Item GUID		
SALESACCOUNTGUID	string	Sales Account GUID		
SALESSTATUSGUID	string	Sales Status GUID		
TASKGUID	string	Task GUID		
TRAVELENDTIME	datetime	Travel End Time		
TRAVELSTARTTIME	datetime	Travel Start Time		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURR ENCY	decimal	Unit Cost in Organization Currency		

Name	Data Type	Label	Required	Documentation
UNITCOSTINUSERCOMPANYCURRENCY	decimal	Unit Cost in User Company Currency		
UNITCOSTVATINCLUDEDAMOUNT	decimal	Unit Cost VAT Included Amount		
UNITPRICE	decimal	Unit Price		
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		
UNITPRICEINPROJECTCOMPANYCURRENCY	decimal	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	decimal	Unit Price in User Company Currency		
USERCODE	string	User Code		
USERGUID	string	User GUID		
VAT	decimal	VAT		

3.1.180 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	boolean	<input type="checkbox"/>		
startDate	datetime	<input type="checkbox"/>		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
BILLINGDEPENDENCYTASKGUID	string	Billing Dependency Task GUID		
BILLINGSCHEDULE	string	Billing Schedule		
CASEGUID	string	Project GUID		
CASENUMBER	int64	Project Number		
COSTCENTERGUID	string	Cost Center GUID		
COSTCENTERIDENTIFIER	string	Cost Center Identifier		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User Company Currency		
DESCRIPTION	string	Ddescription		
DISPLAYPERIODSTART	datetime	Display Period Start		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVICEROWGUID	string	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	string	Measurement Unit		
NAME	string	Name		
ORDERDATE	datetime	Order Date		
PLANNEDBILLINGDATE	datetime	Planned Billing Date		
PRICECURRENCYSHORTFORMOFPROJECTCOMPANYCURRENCY	string	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Price Currency Short Form of Invoicing Currency		

Name	Data Type	Label	Required	Documentation
PRICECURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Price Currency Short Form of Organization Currency		
PRICECURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Price Currency Short Form of User Company Currency		
PRODUCTCODE	string	Product Code		
PRODUCTGUID	string	Product GUID		
PRODUCTTYPE	string	Product Type		
PURCHASEORDERNUMBER	string	Purchase Order Number		
PURCHASEVAT	decimal	Purchase VAT		
QUANTITY	decimal	Quantity		
RECURRENCEENDDATE	datetime	Recurrence End Date		
RECURRENCEENDTYPE	int32	Recurrence End Type		
RECURRENCFREQUENCY	int32	Recurrence Frequency		
RECURRENCESTARTDATE	datetime	Recurrence Start Date		
RECURRENCETIMES	int32	Recurrence Times		
RECURRINGITEMGUID	string	Recurring Item GUID		
SALESACCOUNTGUID	string	Sales Account GUID		
SALESSTATUSGUID	string	Sales Status GUID		
TASKGUID	string	Task GUID		
TRAVELENDTIME	datetime	Travel End Time		
TRAVELSTARTTIME	datetime	Travel Start Time		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	decimal	Unit Cost in Organization Currency		
UNITCOSTINUSERCOMPANYCURRENCY	decimal	Unit Cost in User Company Currency		
UNITCOSTVATINCLUDEDAMOUNT	decimal	Unit Cost VAT Included Amount		
UNITPRICE	decimal	Unit Price		
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		
UNITPRICEINPROJECTCOMPANYCURRENCY	decimal	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	decimal	Unit Price in User Company Currency		
USERCODE	string	User Code		
USERGUID	string	User GUID		
VAT	decimal	VAT		

3.1.181 RemoveAccountGroupMembers: Visma Severa Remove Account Group Members

Catalog: Severa

Schema: API

Label: Remove 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 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	string	<input type="checkbox"/>		
accountGUID	string	<input type="checkbox"/>		

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	string	Dummy	<input checked="" type="checkbox"/>	Dummy column

3.1.182 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 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	string	<input type="checkbox"/>		
contactGUID	string	<input type="checkbox"/>		

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			

3.1.183 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.

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	string	<input type="checkbox"/>		
resourceGUID	string	<input type="checkbox"/>		

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
<code>REMOVEACTIVITYRESOURCEPARTICIPANTRESULT</code>	boolean			

3.1.184 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 Invariant UniversalSQL driver 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	string	<input type="checkbox"/>		
userGUID	string	<input type="checkbox"/>		

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			

3.1.185 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	string	<input type="checkbox"/>		

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	decimal	Allocated Hours		
ALLOCATEDPERCENTAGE	int32	Allocated Percentage		
CASEGUID	string	Project GUID		
ENDDATE	datetime	End Date		
GUID	string	GUID		
PHASEGUID	string	Phase GUID		
STARTDATE	datetime	Start Date		
USERGUID	string	User GUID		

3.1.186 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	string	<input type="checkbox"/>		
caseGUIDs	string	<input type="checkbox"/>		
caseStatusGUIDs	string	<input type="checkbox"/>		
caseTagGUIDs	string	<input type="checkbox"/>		
endDate	datetime	<input type="checkbox"/>		
salesProcessGUIDs	string	<input type="checkbox"/>		

Name	Data Type	Required	Default Value	Documentation
startDate	datetime	<input type="checkbox"/>		
supervisorUserGUIDs	string	<input type="checkbox"/>		
userGUIDs	string	<input type="checkbox"/>		
userTagGUIDs	string	<input type="checkbox"/>		

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	decimal	Allocated Hours		
ALLOCATEDPERCENTAGE	int32	Allocated Percentage		
CASEGUID	string	Project GUID		
ENDDATE	datetime	End Date		
GUID	string	GUID		
PHASEGUID	string	Phase GUID		
STARTDATE	datetime	Start Date		
USERGUID	string	User GUID		

3.1.187 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.

Name	Data Type	Required	Default Value	Documentation
caseGUID	string	<input type="checkbox"/>		

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	decimal	Allocated Hours		
ALLOCATEDPERCENTAGE	int32	Allocated Percentage		
CASEGUID	string	Project GUID		
ENDDATE	datetime	End Date		
GUID	string	GUID		
PHASEGUID	string	Phase GUID		
STARTDATE	datetime	Start Date		
USERGUID	string	User GUID		

3.1.188 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.

Name	Data Type	Required	Default Value	Documentation
caseGUID	string	<input type="checkbox"/>		
userGUID	string	<input type="checkbox"/>		

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	decimal	Allocated Hours		
ALLOCATEDPERCENTAGE	int32	Allocated Percentage		
CASEGUID	string	Project GUID		
ENDDATE	datetime	End Date		
GUID	string	GUID		
PHASEGUID	string	Phase GUID		
STARTDATE	datetime	Start Date		
USERGUID	string	User GUID		

3.1.189 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.

Name	Data Type	Required	Default Value	Documentation
userGUID	string	<input type="checkbox"/>		

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	decimal	Allocated Hours		
ALLOCATEDPERCENTAGE	int32	Allocated Percentage		
CASEGUID	string	Project GUID		
ENDDATE	datetime	End Date		
GUID	string	GUID		
PHASEGUID	string	Phase GUID		
STARTDATE	datetime	Start Date		
USERGUID	string	User GUID		

3.1.190 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	datetime	<input type="checkbox"/>		

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	decimal	Allocated Hours		
ALLOCATEDPERCENTAGE	int32	Allocated Percentage		
CASEGUID	string	Project GUID		
ENDDATE	datetime	End Date		
GUID	string	GUID		
PHASEGUID	string	Phase GUID		
STARTDATE	datetime	Start Date		
USERGUID	string	User GUID		

3.1.191 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	string	<input type="checkbox"/>		

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	string	GUID		
ISACTIVE	boolean	Is Active		
NAME	string	Name		

3.1.192 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	string	GUID		
ISACTIVE	boolean	Is Active		
NAME	string	Name		

3.1.193 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.

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	string	<input type="checkbox"/>		

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	int32	Default Probability		
GUID	string	GUID		
ICON	string	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	string	Name		

3.1.194 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	int32	Default Probability		
GUID	string	GUID		

Name	Data Type	Label	Required	Documentation
ICON	string	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	string	Name		

3.1.195 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	string	<input type="checkbox"/>		

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	string	Project GUID		

Name	Data Type	Label	Required	Documentation
GUID	string	GUID		
SALESPROCESSGUID	string	Sales Process GUID		
TIMESTAMP	datetime	Time Stamp		

3.1.196 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	string	<input type="checkbox"/>		

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	string	Project GUID		
GUID	string	GUID		
SALESPROCESSGUID	string	Sales Process GUID		
TIMESTAMP	datetime	Time Stamp		

3.1.197 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	string			

3.1.198 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	string	<input type="checkbox"/>		
invoiceNumber	int32	<input type="checkbox"/>		
referenceNumber	string	<input type="checkbox"/>		
status	string	<input type="checkbox"/>		

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	string	Dummy	<input checked="" type="checkbox"/>	Dummy column

3.1.199 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	string	<input type="checkbox"/>		
paymentDate	datetime	<input type="checkbox"/>		

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	string	Dummy	<input checked="" type="checkbox"/>	Dummy column

3.1.200 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	string	<input type="checkbox"/>		
status	string	<input type="checkbox"/>		

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	string	Dummy	<input checked="" type="checkbox"/>	Dummy column

3.1.201 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

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	string	<input type="checkbox"/>		
userGUIDs	string	<input type="checkbox"/>		

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			

3.1.202 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.

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	string	<input type="checkbox"/>		
travelReimbursementGUID	string	<input type="checkbox"/>		

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	string	Dummy	<input checked="" type="checkbox"/>	Dummy column

3.1.203 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.

Name	Data Type	Required	Default Value	Documentation
tagGUID	string	<input type="checkbox"/>		

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	string	Context		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
KEYWORD	string	Keyw ord		
WEIGHT	int32	Weight		

3.1.204 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	string	<input type="checkbox"/>		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CONTEXT	string	Context		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
KEYWORD	string	Keyw ord		
WEIGHT	int32	Weight		

3.1.205 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	datetime	<input type="checkbox"/>		

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	string	Ddescription		

Name	Data Type	Label	Required	Documentation
ENDTIME	datetime	End Time		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
PHASEGUID	string	Phase GUID		
QUANTITY	decimal	Quantity		
STARTTIME	datetime	Start Time		
USERGUID	string	User GUID		

3.1.206 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	datetime	<input type="checkbox"/>		
userGUID	string	<input type="checkbox"/>		

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	string	Ddescription		

Name	Data Type	Label	Required	Documentation
ENDTIME	datetime	End Time		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
PHASEGUID	string	Phase GUID		
QUANTITY	decimal	Quantity		
STARTTIME	datetime	Start Time		
USERGUID	string	User GUID		

3.1.207 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	datetime	<input type="checkbox"/>		

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	string	Ddescription		

Name	Data Type	Label	Required	Documentation
ENDTIME	datetime	End Time		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
PHASEGUID	string	Phase GUID		
QUANTITY	decimal	Quantity		
STARTTIME	datetime	Start Time		
USERGUID	string	User GUID		

3.1.208 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 Invariant UniversalSQL driver 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	string	<input type="checkbox"/>		

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	decimal	Difference to GMT		

Name	Data Type	Label	Required	Documentation
GUID	string	GUID		
NAME	string	Name		
SYSTEMNAME	string	System Name		

3.1.209 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	decimal	Difference to GMT		
GUID	string	GUID		
NAME	string	Name		
SYSTEMNAME	string	System Name		

3.1.210 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.

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	string	<input type="checkbox"/>		

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	string	Code		
COSTCURRENCYGUID	string	Cost Currency GUID		
DESCRIPTION	string	Ddescription		
EXPENSECLASS	string	Expense Class		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
MEASUREMENTUNIT	string	Measurement Unit		
NAME	string	Name		
TYPE	string	Type		
UNITCOST	decimal	Unit Cost		
UNITPRICE	decimal	Unit Price		
USEINWORKTIMEENTRY	boolean	Use in Work Time Entry		
VAT	decimal	VAT		

3.1.211 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.

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	string	<input type="checkbox"/>		

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	string	Code		
COSTCURRENCYGUID	string	Cost Currency GUID		
DESCRIPTION	string	Description		
EXPENSECLASS	string	Expense Class		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
MEASUREMENTUNIT	string	Measurement Unit		
NAME	string	Name		
TYPE	string	Type		
UNITCOST	decimal	Unit Cost		
UNITPRICE	decimal	Unit Price		
USEINWORKTIMEENTRY	boolean	Use in Work Time Entry		
VAT	decimal	VAT		

3.1.212 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 Invariant UniversalSQL driver 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 `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	string	<input type="checkbox"/>		

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	string	Code		
COSTCURRENCYGUID	string	Cost Currency GUID		
DESCRIPTION	string	Ddescription		
EXPENSECLASS	string	Expense Class		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
MEASUREMENTUNIT	string	Measurement Unit		
NAME	string	Name		
TYPE	string	Type		
UNITCOST	decimal	Unit Cost		
UNITPRICE	decimal	Unit Price		
USEINWORKTIMEENTRY	boolean	Use in Work Time Entry		
VAT	decimal	VAT		

3.1.213 TravelExpensesByDate: Visma Severa Travel Expenses by Date

Catalog: Severa

Schema: API

Label: Travel Expenses 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 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	string	<input type="checkbox"/>		
endDate	datetime	<input type="checkbox"/>		
startDate	datetime	<input type="checkbox"/>		

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	string	Billing Dependency Task GUID		
BILLINGSCHEDULE	string	Billing Schedule		
CASEGUID	string	Project GUID		
CASENUMBER	int64	Project Number		
COSTCENTERGUID	string	Cost Center GUID		
COSTCENTERIDENTIFIER	string	Cost Center Identifier		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User		

Name	Data Type	Label	Required	Documentation
		Company Currency		
DESCRIPTION	string	Ddescription		
DISPLAYPERIODSTART	datetime	Display Period Start		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVICEROWGUID	string	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	string	Measurement Unit		
NAME	string	Name		
ORDERDATE	datetime	Order Date		
PLANNEDBILLINGDATE	datetime	Planned Billing Date		
PRICECURRENCYSHORTFORMOFCS ASECOMPANYCURRENCY	string	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOFI NVOICINGCURRENCY	string	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOFO RGANIZATIONCURRENCY	string	Price Currency Short Form of Organization Currency		
PRICECURRENCYSHORTFORMOFU SERCOMPANYCURRENCY	string	Price Currency Short Form of User Company Currency		
PRODUCTCODE	string	Product Code		
PRODUCTGUID	string	Product GUID		
PRODUCTTYPE	string	Product Type		
PURCHASEORDERNUMBER	string	Purchase Order Number		
PURCHASEVAT	decimal	Purchase VAT		
QUANTITY	decimal	Quantity		
RECURRENCEENDDATE	datetime	Recurrence End Date		
RECURRENCEENDTYPE	int32	Recurrence End Type		
RECURRENCEFREQUENCY	int32	Recurrence Frequency		
RECURRENCESTARTDATE	datetime	Recurrence Start Date		
RECURRENCETIMES	int32	Recurrence Times		
RECURRINGITEMGUID	string	Recurring Item GUID		

Name	Data Type	Label	Required	Documentation
SALESAccountGUID	string	Sales Account GUID		
SALESSTATUSGUID	string	Sales Status GUID		
TASKGUID	string	Task GUID		
TRAVELENDTIME	datetime	Travel End Time		
TRAVELSTARTTIME	datetime	Travel Start Time		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	decimal	Unit Cost in Organization Currency		
UNITCOSTINUSERCOMPANYCURRENCY	decimal	Unit Cost in User Company Currency		
UNITCOSTVATINCLUDEDAMOUNT	decimal	Unit Cost VAT Included Amount		
UNITPRICE	decimal	Unit Price		
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		
UNITPRICEINPROJECTCOMPANYCURRENCY	decimal	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYCURRENCY	decimal	Unit Price in User Company Currency		
USERCODE	string	User Code		
USERGUID	string	User GUID		
VAT	decimal	VAT		

3.1.214 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

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	int32	<input type="checkbox"/>		
startDateTime	datetime	<input type="checkbox"/>		

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	string	Code		
COSTCURRENCYGUID	string	Cost Currency GUID		
DESCRIPTION	string	Ddescription		
EXPENSECLASS	string	Expense Class		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
MEASUREMENTUNIT	string	Measurement Unit		
NAME	string	Name		
TYPE	string	Type		
UNITCOST	decimal	Unit Cost		
UNITPRICE	decimal	Unit Price		
USEINWORKTIMEENTRY	boolean	Use in Work Time Entry		
VAT	decimal	VAT		

3.1.215 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

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	string	<input type="checkbox"/>		

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	string	Content Type		
DESCRIPTION	string	Ddescription		
GUID	string	GUID		
NAME	string	Name		
SIZE	int32	Size		

3.1.216 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.

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	string	<input type="checkbox"/>		

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	decimal	Advance Payment		
APPROVALDATE	datetime	Approval Date		
APPROVEDBYUSERGUID	string	Approved by User GUID		
CURRENCYGUID	string	Currency GUID		
CURRENCYSHORTFORM	string	Currency Short Form		
DATE	datetime	Date		
GUID	string	GUID		
NOTES	string	Notes		
NUMBER	int32	Number		
TRAVELREIMBURSEMENTROWS	string			
TRAVELREIMBURSEMENTSTATUS	string	Travel Reimbursement Status		
USERGUID	string	User GUID		

3.1.217 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

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
<code>travelReimbursementNumber</code>	int32	<input type="checkbox"/>		
<code>travelReimbursementNumberingBusinessUnitGUID</code>	string	<input type="checkbox"/>		

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
<code>ADVANCEPAYMENT</code>	decimal	Advance Payment		
<code>APPROVALDATE</code>	datetime	Approval Date		
<code>APPROVEDBYUSERGUID</code>	string	Approved by User GUID		
<code>CURRENCYGUID</code>	string	Currency GUID		
<code>CURRENCYSHORTFORM</code>	string	Currency Short Form		
<code>DATE</code>	datetime	Date		
<code>GUID</code>	string	GUID		
<code>NOTES</code>	string	Notes		
<code>NUMBER</code>	int32	Number		
<code>TRAVELREIMBURSEMENTROWS</code>	string			
<code>TRAVELREIMBURSEMENTSTATUS</code>	string	Travel Reimbursement Status		
<code>USERGUID</code>	string	User GUID		

3.1.218 `TravelReimbursementPdf`: Visma Severa Travel Reimbursement PDF

Catalog: Severa

Schema: API

Label: Travel Reimbursement 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: 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	string	<input type="checkbox"/>		

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	byte[]			

3.1.219 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

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	datetime	<input type="checkbox"/>		
options	int32	<input type="checkbox"/>		
startDate	datetime	<input type="checkbox"/>		

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	decimal	Advance Payment		
APPROVALDATE	datetime	Approval Date		
APPROVEDBYUSERGUID	string	Approved by User GUID		
CURRENCYGUID	string	Currency GUID		
CURRENCYSHORTFORM	string	Currency Short Form		
DATE	datetime	Date		
GUID	string	GUID		
NOTES	string	Notes		
NUMBER	int32	Number		
TRAVELREIMBURSEMENTROWS	string			
TRAVELREIMBURSEMENTSTATUS	string	Travel Reimbursement Status		
USERGUID	string	User GUID		

3.1.220 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 Invariant UniversalSQL driver 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- 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	string	<input type="checkbox"/>		
options	int32	<input type="checkbox"/>		
status	string	<input type="checkbox"/>		

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	decimal	Advance Payment		
APPROVALDATE	datetime	Approval Date		
APPROVEDBYUSERGUID	string	Approved by User GUID		
CURRENCYGUID	string	Currency GUID		
CURRENCYSHORTFORM	string	Currency Short Form		
DATE	datetime	Date		
GUID	string	GUID		
NOTES	string	Notes		
NUMBER	int32	Number		
TRAVELREIMBURSEMENTROWS	string			
TRAVELREIMBURSEMENTSTATUS	string	Travel Reimbursement Status		
USERGUID	string	User GUID		

3.1.221 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 Invariant UniversalSQL driver 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	int32	<input type="checkbox"/>		
travelReimbursementStatusGUID	string	<input type="checkbox"/>		

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	decimal	Advance Payment		
APPROVALDATE	datetime	Approval Date		
APPROVEDBYUSERGUID	string	Approved by User GUID		
CURRENCYGUID	string	Currency GUID		
CURRENCYSHORTFORM	string	Currency Short Form		
DATE	datetime	Date		
GUID	string	GUID		
NOTES	string	Notes		
NUMBER	int32	Number		

Name	Data Type	Label	Required	Documentation
TRAVELREIMBURSEMENTROWS	string			
TRAVELREIMBURSEMENTSTATUS	string	Travel Reimbursement Status		
USERGUID	string	User GUID		

3.1.222 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	string	GUID		
ISACTIVE	boolean	Is Active		
ISAPPROVED	boolean	Is Approved		
ISLOCKED	boolean	Is Locked		
NAME	string	Name		
SORTORDER	int32	Sort Order		

3.1.223 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

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	string	<input type="checkbox"/>		
accessRightsProfile_IsActive	boolean	<input type="checkbox"/>		
accessRightsProfile_IsDefault	boolean	<input type="checkbox"/>		
accessRightsProfile_Name	string	<input type="checkbox"/>		
accessRightsProfile_Rights_Account	string	<input type="checkbox"/>		
accessRightsProfile_Rights_AccountDelete	boolean	<input type="checkbox"/>		
accessRightsProfile_Rights_Administrator	string	<input type="checkbox"/>		
accessRightsProfile_Rights_Case	string	<input type="checkbox"/>		
accessRightsProfile_Rights_CaseDelete	boolean	<input type="checkbox"/>		
accessRightsProfile_Rights_CaseOwnerWorkHourApproval	boolean	<input type="checkbox"/>		
accessRightsProfile_Rights_ScheduleJobs	string	<input type="checkbox"/>		
accessRightsProfile_Rights_Sharing	boolean	<input type="checkbox"/>		
accessRightsProfile_Rights_TravelReimbursement	string	<input type="checkbox"/>		
accessRightsProfile_Rights_Users	string	<input type="checkbox"/>		
accessRightsProfile_Rights_WorkHourApproval	string	<input type="checkbox"/>		

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	string	GUID		

Name	Data Type	Label	Required	Documentation
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
NAME	string	Name		
RIGHTS_ACCOUNT	string			
RIGHTS_ACCOUNTDELETE	boolean			
RIGHTS_ADMINISTRATOR	string			
RIGHTS_CASE	string			
RIGHTS_CASEDELETE	boolean			
RIGHTS_CASEOWNERWORKHOUR APPROVAL	boolean			
RIGHTS_SCHEDULEJOBS	string			
RIGHTS_SHARING	boolean			
RIGHTS_TRAVELREIMBURSEMENT	string			
RIGHTS_USERS	string			
RIGHTS_WORKHOURAPPROVAL	string			

3.1.224 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_LanguageCode- accountInfo_LanguageGUID-
 accountInfo_Name- accountInfo_Notes- accountInfo_Number- accountInfo_OverdueInterest-
 accountInfo_PaymentTerm- accountInfo_PricelistGUID- accountInfo_ReverseCharge-
 accountInfo_ReverseChargeDescription- accountInfo_UpdatedTS-
 accountInfo_UrIToAccount- accountInfo_eInvoiceAddress- accountInfo_eInvoiceOperator-
 companyInfo_AnnualRevenue- companyInfo_AnnualRevenue2- companyInfo_Email-
 companyInfo_Employees- companyInfo_GUID- companyInfo_HeadOfficeAddressGUID-
 companyInfo_HierarchyGUID- companyInfo_IndustryGUID- 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.

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	string	<input type="checkbox"/>		
accountInfo_AccountOwnerUserGUID	string	<input type="checkbox"/>		
accountInfo_AccountRating	int32	<input type="checkbox"/>		
accountInfo_CompanyGUID	string	<input type="checkbox"/>		
accountInfo_CurrencyCode	string	<input type="checkbox"/>		
accountInfo_CurrencyGUID	string	<input type="checkbox"/>		
accountInfo_eInvoiceAddress	string	<input type="checkbox"/>		
accountInfo_eInvoiceOperator	string	<input type="checkbox"/>		
accountInfo_GUID	string	<input type="checkbox"/>		
accountInfo_InsertTS	datetime	<input type="checkbox"/>		
accountInfo_InvoicingVat	decimal	<input type="checkbox"/>		
accountInfo_IsActive	boolean	<input type="checkbox"/>		
accountInfo_IsInternal	boolean	<input type="checkbox"/>		
accountInfo_LanguageCode	string	<input type="checkbox"/>		
accountInfo_LanguageGUID	string	<input type="checkbox"/>		
accountInfo_Name	string	<input type="checkbox"/>		
accountInfo_Notes	string	<input type="checkbox"/>		
accountInfo_Number	int64	<input type="checkbox"/>		
accountInfo_OverdueInterest	decimal	<input type="checkbox"/>		
accountInfo_PaymentTerm	int32	<input type="checkbox"/>		
accountInfo_PricelistGUID	string	<input type="checkbox"/>		
accountInfo_ReverseCharge	boolean	<input type="checkbox"/>		
accountInfo_ReverseChargeDescription	string	<input type="checkbox"/>		
accountInfo_UpdatedTS	datetime	<input type="checkbox"/>		
accountInfo_UrlToAccount	string	<input type="checkbox"/>		
companyInfo_AnnualRevenue	int32	<input type="checkbox"/>		
companyInfo_AnnualRevenue2	int64	<input type="checkbox"/>		
companyInfo_Email	string	<input type="checkbox"/>		
companyInfo_Employees	int32	<input type="checkbox"/>		
companyInfo_GUID	string	<input type="checkbox"/>		
companyInfo_HeadOfficeAddressGUID	string	<input type="checkbox"/>		
companyInfo_HierarchyGUID	string	<input type="checkbox"/>		

Name	Data Type	Required	Default Value	Documentation
companyInfo_IndustryGUID	string	<input type="checkbox"/>		
companyInfo_LanguageGUID	string	<input type="checkbox"/>		
companyInfo_Name	string	<input type="checkbox"/>		
companyInfo_TimezoneGUID	string	<input type="checkbox"/>		
companyInfo_VatNumber	string	<input type="checkbox"/>		
companyInfo_Website	string	<input type="checkbox"/>		

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	string			
ACCOUNTOWNERUSERGUID	string	Account Owner User GUID		
ACCOUNTRATING	int32	Account Rating		
COMPANYGUID	string	Company GUID		
CURRENCYCODE	string	Currency Code		
CURRENCYGUID	string	Currency GUID		
EINVOICEADDRESS	string	e-invoice Address		
EINVOICEOPERATOR	string	e-invoice Operator		
GUID	string	GUID		
INSERTTS	datetime	Inserted		
INVOICINGVAT	decimal	Invoicing VAT		
ISACTIVE	boolean	Is Active		
ISINTERNAL	boolean	Is Internal		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
NAME	string	Name		
NOTES	string	Notes		
NUMBER	int64	Number		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTTERM	int32	Payment Term		
PRICELISTGUID	string	Pricelist GUID		
REVERSECHARGE	boolean	Reverse Charge		
REVERSECHARGEDESCRIPTION	string	Reverse Charge Description		
UPDATEDTS	datetime	Updated		
URLTOACCOUNT	string	URL to Account		

3.1.225 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	string	<input type="checkbox"/>		
activityInfo_ActivityTypeCode	string	<input type="checkbox"/>		
activityInfo_ActivityTypeGUID	string	<input type="checkbox"/>		
activityInfo_CaseGUID	string	<input type="checkbox"/>		
activityInfo_EndDateTime	datetime	<input type="checkbox"/>		
activityInfo_GUID	string	<input type="checkbox"/>		
activityInfo_IsAllDay	boolean	<input type="checkbox"/>		
activityInfo_IsClosed	boolean	<input type="checkbox"/>		
activityInfo_IsDuration	boolean	<input type="checkbox"/>		
activityInfo_IsPrivate	boolean	<input type="checkbox"/>		
activityInfo_IsReadOnly	boolean	<input type="checkbox"/>		
activityInfo_IsUnassigned	boolean	<input type="checkbox"/>		
activityInfo_Location	string	<input type="checkbox"/>		
activityInfo_Name	string	<input type="checkbox"/>		

Name	Data Type	Required	Default Value	Documentation
activityInfo_Notes	string	<input type="checkbox"/>		
activityInfo_OwnerUserBusinessUnitCode	string	<input type="checkbox"/>		
activityInfo_OwnerUserCode	string	<input type="checkbox"/>		
activityInfo_OwnerUserFirstName	string	<input type="checkbox"/>		
activityInfo_OwnerUserGUID	string	<input type="checkbox"/>		
activityInfo_OwnerUserLastName	string	<input type="checkbox"/>		
activityInfo_StartDateTime	datetime	<input type="checkbox"/>		
activityInfo_TaskGUID	string	<input type="checkbox"/>		

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	string	Account GUID		
ACTIVITYTYPECODE	string	Activity Type Code		
ACTIVITYTYPEGUID	string	Activity Type GUID		
CASEGUID	string	Project GUID		
ENDDATETIME	datetime	End Date Time		
GUID	string	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	string	Location		
NAME	string	Name		
NOTES	string	Notes		
OWNERUSERBUSINESSUNITCODE	string	Owner User Business Unit Code		
OWNERUSERCODE	string	Owner User Code		
OWNERUSERFIRSTNAME	string	Owner User First Name		
OWNERUSERGUID	string	Owner User GUID		
OWNERUSERLASTNAME	string	Owner User Last Name		
STARTDATETIME	datetime	Start Date Time		
TASKGUID	string	Task GUID		

3.1.226 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	string	<input type="checkbox"/>		
activityTypeInfo_Code	string	<input type="checkbox"/>		
activityTypeInfo_GUID	string	<input type="checkbox"/>		
activityTypeInfo_Icon	string	<input type="checkbox"/>		
activityTypeInfo_IsActive	boolean	<input type="checkbox"/>		
activityTypeInfo_IsDefault	boolean	<input type="checkbox"/>		
activityTypeInfo_IsPaidLeave	boolean	<input type="checkbox"/>		
activityTypeInfo_Name	string	<input type="checkbox"/>		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
CATEGORY	string	Category		
CODE	string	Code		
GUID	string	GUID		

Name	Data Type	Label	Required	Documentation
ICON	string	Icon		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
ISPAIDLEAVE	boolean	Is Paid Leave		
NAME	string	Name		

3.1.227 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	string	<input type="checkbox"/>		
addressInfo_Addressline2	string	<input type="checkbox"/>		
addressInfo_Addressline3	string	<input type="checkbox"/>		
addressInfo_City	string	<input type="checkbox"/>		
addressInfo_CompanyGUID	string	<input type="checkbox"/>		
addressInfo_CountryCode	string	<input type="checkbox"/>		
addressInfo_CountryGUID	string	<input type="checkbox"/>		

Name	Data Type	Required	Default Value	Documentation
addressInfo_CountryName	string	<input type="checkbox"/>		
addressInfo_CountryRegionGUID	string	<input type="checkbox"/>		
addressInfo_CountryRegionName	string	<input type="checkbox"/>		
addressInfo_Fax	string	<input type="checkbox"/>		
addressInfo_GUID	string	<input type="checkbox"/>		
addressInfo_IsBillingAddress	boolean	<input type="checkbox"/>		
addressInfo_IsPostalAddress	boolean	<input type="checkbox"/>		
addressInfo_IsVisitAddress	boolean	<input type="checkbox"/>		
addressInfo_Phone	string	<input type="checkbox"/>		
addressInfo_PostalCode	string	<input type="checkbox"/>		

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

3.1.228 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	decimal	<input type="checkbox"/>		
billingAndExpenseForecastInfo_ExpenseForecast	decimal	<input type="checkbox"/>		
billingAndExpenseForecastInfo_GUID	string	<input type="checkbox"/>		
billingAndExpenseForecastInfo_InhouseRevenueForecast	decimal	<input type="checkbox"/>		
billingAndExpenseForecastInfo_Month	int32	<input type="checkbox"/>		
billingAndExpenseForecastInfo_Notes	string	<input type="checkbox"/>		
billingAndExpenseForecastInfo_OutsourcingRevenueForecast	decimal	<input type="checkbox"/>		
billingAndExpenseForecastInfo_Year	int32	<input type="checkbox"/>		

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	decimal	Billing Forecast		
EXPENSEFORECAST	decimal	Expense Forecast		
GUID	string	GUID		
INHOUSEREVENUEFORECAST	decimal	Inhouse Revenue Forecast		
MONTH	int32	Month		
NOTES	string	Notes		
OUTSOURCINGREVENUEFORECAST	decimal	Outsourcing Revenue Forecast		
YEAR	int32	Year		

3.1.229 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_OverDuelInterest- 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.

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	string	<input type="checkbox"/>		
businessUnitInfo_CostCenterGUID	string	<input type="checkbox"/>		
businessUnitInfo_CurrencyCode	string	<input type="checkbox"/>		
businessUnitInfo_CurrencyGUID	string	<input type="checkbox"/>		
businessUnitInfo_FormattingCulture	string	<input type="checkbox"/>		
businessUnitInfo_GUID	string	<input type="checkbox"/>		
businessUnitInfo_InsertTS	datetime	<input type="checkbox"/>		
businessUnitInfo_IsActive	boolean	<input type="checkbox"/>		
businessUnitInfo_LanguageCode	string	<input type="checkbox"/>		
businessUnitInfo_LanguageGUID	string	<input type="checkbox"/>		
businessUnitInfo_Name	string	<input type="checkbox"/>		
businessUnitInfo_NextInvoiceNumber	int32	<input type="checkbox"/>		
businessUnitInfo_OverDueInterest	decimal	<input type="checkbox"/>		
businessUnitInfo_ParentBusinessUnitGUID	string	<input type="checkbox"/>		
businessUnitInfo_PaymentTerm	int32	<input type="checkbox"/>		
businessUnitInfo_Show Logo On Invoices	boolean	<input type="checkbox"/>		
businessUnitInfo_Show Logo On Pdf Reports	boolean	<input type="checkbox"/>		
businessUnitInfo_TimeZoneGUID	string	<input type="checkbox"/>		
businessUnitInfo_UpdatedTS	datetime	<input type="checkbox"/>		
businessUnitInfo_VatNumber	string	<input type="checkbox"/>		

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	string	Code		
COSTCENTERGUID	string	Cost Center GUID		
CURRENCYCODE	string	Currency Code		
CURRENCYGUID	string	Currency GUID		
FORMATTINGCULTURE	string	Formatting Culture		
GUID	string	GUID		
INSERTTS	datetime	Inserted		
ISACTIVE	boolean	Is Active		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
NAME	string	Name		
NEXTINVOICENUMBER	int32	Next Invoice Number		

Name	Data Type	Label	Required	Documentation
OVERDUEINTEREST	decimal	Overdue Interest		
PARENTBUSINESSUNITGUID	string	Parent Business Unit GUID		
PAYMENTTERM	int32	Payment Term		
SHOWLOGOONINVOICES	boolean	Show Logo on Invoices		
SHOWLOGOONPDFREPORTS	boolean	Show Logo on PDF Reports		
TIMEZONEGUID	string	Timezone GUID		
UPDATEDTS	datetime	Updated		
VATNUMBER	string	VAT Number		

3.1.230 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: caselInfo_AccountGUID- caselInfo_AccountName- caselInfo_AccountNumber- caselInfo_BillingAddress- caselInfo_BillingAddressGUID- caselInfo_BillingInternalNotes- caselInfo_BillingNotesBeforeGrid- caselInfo_BusinessUnitGUID- caselInfo_BusinessUnitName- caselInfo_BusinessUnitNumber- caselInfo_CaseCompanyCurrencyGUID- caselInfo_CaseNumber- caselInfo_CaseOwnerUserCode- caselInfo_CaseOwnerUserGUID- caselInfo_ClosedDate- caselInfo_CompletionEstimate- caselInfo_ContactGUID- caselInfo_CostCenterGUID- caselInfo_CostCenterNumber- caselInfo_CurrencyGUID- caselInfo_CurrencyShortform- caselInfo_CurrencyShortformOfCaseCompanyCurrency- caselInfo_CurrentCaseStatusDescription- caselInfo_CurrentCaseStatusGUID- caselInfo_CurrentCaseStatusInsertTs- caselInfo_DeadlineDate- caselInfo_Description- caselInfo_ExpectedValue- caselInfo_GUID- caselInfo_IncludeExceptions- caselInfo_IncludeOverTime- caselInfo_InsertTs- caselInfo_InternalName- caselInfo_InvoiceTemplateGUID- caselInfo_IsClosed- caselInfo_IsDailyAllowanceBillable- caselInfo_IsInternal- caselInfo_IsMileageBillable- caselInfo_IsOtherTravelExpensesBillable- caselInfo_LatestEstimationDate- caselInfo_LeadRating- caselInfo_LeadSourceGUID- caselInfo_Name- caselInfo_OrderNumber- caselInfo_OurReference- caselInfo_OverdueInterest- caselInfo_PaymentTerm- caselInfo_PricelistGUID- caselInfo_PricingRule- caselInfo_PricingType- caselInfo_Priority- caselInfo_Probability- caselInfo_RootTaskGUID- caselInfo_SalesCloseDate- caselInfo_SalesPersonUserCode- caselInfo_SalesPersonUserGUID- caselInfo_SalesProcessGUID- caselInfo_SalesStatusGUID- caselInfo_SharedToExtranet- caselInfo_StartDate- caselInfo_UpdatedTs- caselInfo_UrlToCase- caselInfo_UseDefaultProductsInWorkTimeEntry- caselInfo_UseDefaultWorkTypes- caselInfo_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	string	<input type="checkbox"/>		
caseInfo_AccountName	string	<input type="checkbox"/>		
caseInfo_AccountNumber	int64	<input type="checkbox"/>		
caseInfo_BillingAddress	string	<input type="checkbox"/>		
caseInfo_BillingAddressGUID	string	<input type="checkbox"/>		
caseInfo_BillingInternalNotes	string	<input type="checkbox"/>		
caseInfo_BillingNotesBeforeGrid	string	<input type="checkbox"/>		
caseInfo_BusinessUnitGUID	string	<input type="checkbox"/>		
caseInfo_BusinessUnitName	string	<input type="checkbox"/>		
caseInfo_BusinessUnitNumber	string	<input type="checkbox"/>		
caseInfo_CaseCompanyCurrencyGUID	string	<input type="checkbox"/>		
caseInfo_CaseNumber	int64	<input type="checkbox"/>		
caseInfo_CaseOwnerUserCode	string	<input type="checkbox"/>		
caseInfo_CaseOwnerUserGUID	string	<input type="checkbox"/>		
caseInfo_ClosedDate	datetime	<input type="checkbox"/>		
caseInfo_CompletionEstimate	int32	<input type="checkbox"/>		
caseInfo_ContactGUID	string	<input type="checkbox"/>		
caseInfo_CostCenterGUID	string	<input type="checkbox"/>		
caseInfo_CostCenterNumber	string	<input type="checkbox"/>		
caseInfo_CurrencyGUID	string	<input type="checkbox"/>		
caseInfo_CurrencyShortform	string	<input type="checkbox"/>		
caseInfo_CurrencyShortformOfCaseCompanyCurrency	string	<input type="checkbox"/>		
caseInfo_CurrentCaseStatusDescription	string	<input type="checkbox"/>		
caseInfo_CurrentCaseStatusGUID	string	<input type="checkbox"/>		
caseInfo_CurrentCaseStatusInserts	datetime	<input type="checkbox"/>		
caseInfo_DeadlineDate	datetime	<input type="checkbox"/>		
caseInfo_Description	string	<input type="checkbox"/>		

Name	Data Type	Required	Default Value	Documentation
caseInfo_ExpectedValue	decimal	<input type="checkbox"/>		
caseInfo_GUID	string	<input type="checkbox"/>		
caseInfo_IncludeExceptions	boolean	<input type="checkbox"/>		
caseInfo_IncludeOverTime	boolean	<input type="checkbox"/>		
caseInfo_InsertTs	datetime	<input type="checkbox"/>		
caseInfo_InternalName	string	<input type="checkbox"/>		
caseInfo_InvoiceTemplateGUID	string	<input type="checkbox"/>		
caseInfo_IsClosed	boolean	<input type="checkbox"/>		
caseInfo_IsDailyAllow anceBillable	boolean	<input type="checkbox"/>		
caseInfo_IsInternal	boolean	<input type="checkbox"/>		
caseInfo_IsMileageBillable	boolean	<input type="checkbox"/>		
caseInfo_IsOtherTravelExpensesBillable	boolean	<input type="checkbox"/>		
caseInfo_LatestEstimationDate	datetime	<input type="checkbox"/>		
caseInfo_LeadRating	int32	<input type="checkbox"/>		
caseInfo_LeadSourceGUID	string	<input type="checkbox"/>		
caseInfo_Name	string	<input type="checkbox"/>		
caseInfo_OrderNumber	string	<input type="checkbox"/>		
caseInfo_OurReference	string	<input type="checkbox"/>		
caseInfo_OverdueInterest	decimal	<input type="checkbox"/>		
caseInfo_PaymentTerm	int32	<input type="checkbox"/>		
caseInfo_PricelistGUID	string	<input type="checkbox"/>		
caseInfo_PricingRule	string	<input type="checkbox"/>		
caseInfo_PricingType	string	<input type="checkbox"/>		
caseInfo_Priority	int32	<input type="checkbox"/>		
caseInfo_Probability	int32	<input type="checkbox"/>		
caseInfo_RootTaskGUID	string	<input type="checkbox"/>		
caseInfo_SalesCloseDate	datetime	<input type="checkbox"/>		
caseInfo_SalesPersonUserCode	string	<input type="checkbox"/>		
caseInfo_SalesPersonUserGUID	string	<input type="checkbox"/>		
caseInfo_SalesProcessGUID	string	<input type="checkbox"/>		
caseInfo_SalesStatusGUID	string	<input type="checkbox"/>		
caseInfo_SharedToExtranet	boolean	<input type="checkbox"/>		
caseInfo_StartDate	datetime	<input type="checkbox"/>		
caseInfo_UpdatedTs	datetime	<input type="checkbox"/>		
caseInfo_UrlToCase	string	<input type="checkbox"/>		
caseInfo_UseDefaultProductsInWorkTimeEntry	boolean	<input type="checkbox"/>		
caseInfo_UseDefaultWorkTypes	boolean	<input type="checkbox"/>		
caseInfo_YourReference	string	<input type="checkbox"/>		

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	string	Account GUID		
ACCOUNTNAME	string	Account Name		
ACCOUNTNUMBER	int64	Account Number		
BILLINGADDRESS	string	Billing Address		
BILLINGADDRESSGUID	string	Billing Address GUID		
BILLINGINTERNALNOTES	string	Billing Internal Notes		
BILLINGNOTESBEFOREGRID	string	Billing Notes before Grid		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
BUSINESSUNITNUMBER	string	Business Unit Number		
CASECOMPANYCURRENCYGUID	string	Project Company Currency GUID		
CASENUMBER	int64	Project Number		
CASEOWNERUSERCODE	string	Project Owner User Code		
CASEOWNERUSERGUID	string	Project Owner User GUID		
CLOSEDDATE	datetime	Closed Date		
COMPLETIONESTIMATE	int32	Completion Estimate		
CONTACTGUID	string	Contact GUID		
COSTCENTERGUID	string	Cost Center GUID		
COSTCENTERNUMBER	string	Cost Center Number		
CURRENCYGUID	string	Currency GUID		
CURRENCYSHORTFORM	string	Currency Short Form		
CURRENCYSHORTFORMOFCASECOMPANYCURRENCY	string	Currency Short Form of Project Company Currency		
CURRENTCASESTATUSDESCRIPTION	string	Current Project Status Description		
CURRENTCASESTATUSGUID	string	Current Project Status GUID		
CURRENTCASESTATUSINSERTTS	datetime	Current Project Status Inserted		
DEADLINE DATE	datetime	Deadline Date		
DESCRIPTION	string	Description		
EXPECTEDVALUE	decimal	Expected Value		
GUID	string	GUID		
INCLUDEEXCEPTIONS	boolean	Include Exceptions		
INCLUDEOVERTIME	boolean	Include Overtime		

Name	Data Type	Label	Required	Documentation
INSERTTS	datetime	Inserted		
INTERNALNAME	string	Internal Name		
INVOICETEMPLATEGUID	string	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	int32	Lead Rating		
LEADSOURCEGUID	string	Lead Source GUID		
NAME	string	Name		
ORDERNUMBER	string	Order Number		
OURREFERENCE	string	Our Reference		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTTERM	int32	Payment Term		
PRICELISTGUID	string	Pricelist GUID		
PRICINGRULE	string	Pricing Rule		
PRICINGTYPE	string	Pricing Type		
PRIORITY	int32	Priority		
PROBABILITY	int32	Probability		
ROOTTASKGUID	string	Root Task GUID		
SALESCLOSEDATE	datetime	Sales Close Date		
SALESPERSONUSERCODE	string	Sales Person User Code		
SALESPERSONUSERGUID	string	Sales Person User GUID		
SALESPROCESSGUID	string	Sales Process GUID		
SALESSTATUSGUID	string	Sales Status GUID		
SHAREDTOEXTRANET	boolean	Shared to Extranet		
STARTDATE	datetime	Start Date		
UPDATEDTS	datetime	Updated		
URLTOCASE	string	URL to Project		
USEDEFAULTPRODUCTSINWORKTIMEENTRY	boolean	Use Default Products in Work Time Entry		
USEDEFAULTWORKTYPES	boolean	Use Default Work Types		
YOURREFERENCE	string	Your Reference		

3.1.231 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	string	<input type="checkbox"/>		
caseStatusInfo_Icon	string	<input type="checkbox"/>		
caseStatusInfo_IsActive	boolean	<input type="checkbox"/>		
caseStatusInfo_Name	string	<input type="checkbox"/>		

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	string	GUID		
ICON	string	Icon		
ISACTIVE	boolean	Is Active		
NAME	string	Name		

3.1.232 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	string	<input type="checkbox"/>		
contactInfo_AddressGUID	string	<input type="checkbox"/>		
contactInfo_CommunicationMethods	string	<input type="checkbox"/>		
contactInfo_ContactRoleGUID	string	<input type="checkbox"/>		
contactInfo_DateOfBirth	datetime	<input type="checkbox"/>		
contactInfo_Description	string	<input type="checkbox"/>		
contactInfo_FirstName	string	<input type="checkbox"/>		
contactInfo_GUID	string	<input type="checkbox"/>		
contactInfo_IsActive	boolean	<input type="checkbox"/>		
contactInfo_JobTitle	string	<input type="checkbox"/>		
contactInfo_Keyw ords	string	<input type="checkbox"/>		
contactInfo_LanguageGUID	string	<input type="checkbox"/>		
contactInfo_LastName	string	<input type="checkbox"/>		
contactInfo_Salutation	string	<input type="checkbox"/>		
contactInfo_SatisfactionLevel	int32	<input type="checkbox"/>		

Name	Data Type	Required	Default Value	Documentation
contactInfo_TimeZoneGUID	string	<input type="checkbox"/>		

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	string	Account GUID		
ADDRESSGUID	string	Address GUID		
COMMUNICATIONMETHODS	string			
CONTACTROLEGUID	string	Contact Role GUID		
DATEOFBIRTH	datetime	Date of Birth		
DESCRIPTION	string	Ddescription		
FIRSTNAME	string	First Name		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
JOBTITLE	string	Job Title		
KEYWORDS	string			
LANGUAGEGUID	string	Language GUID		
LASTNAME	string	Last Name		
SALUTATION	string	Salutation		
SATISFACTIONLEVEL	int32	Satisfaction Level		
TIMEZONEGUID	string	Timezone GUID		

3.1.233 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_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

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	string	<input type="checkbox"/>		
customer_AccountOwnerUserGUID	string	<input type="checkbox"/>		
customer_AccountRating	int32	<input type="checkbox"/>		
customer_AnnualRevenue	int32	<input type="checkbox"/>		
customer_AnnualRevenue2	int64	<input type="checkbox"/>		
customer_CompanyGUID	string	<input type="checkbox"/>		
customer_CurrencyCode	string	<input type="checkbox"/>		
customer_CurrencyGUID	string	<input type="checkbox"/>		
customer_eInvoiceAddress	string	<input type="checkbox"/>		
customer_eInvoiceOperator	string	<input type="checkbox"/>		
customer_Email	string	<input type="checkbox"/>		
customer_Employees	int32	<input type="checkbox"/>		
customer_GUID	string	<input type="checkbox"/>		
customer_HeadOfficeAddressGUID	string	<input type="checkbox"/>		
customer_IndustryGUID	string	<input type="checkbox"/>		
customer_InsertTS	datetime	<input type="checkbox"/>		
customer_InvoicingVat	decimal	<input type="checkbox"/>		
customer_IsActive	boolean	<input type="checkbox"/>		
customer_IsInternal	boolean	<input type="checkbox"/>		
customer_LanguageCode	string	<input type="checkbox"/>		
customer_LanguageGUID	string	<input type="checkbox"/>		
customer_Name	string	<input type="checkbox"/>		
customer_Notes	string	<input type="checkbox"/>		
customer_Number	int64	<input type="checkbox"/>		

Name	Data Type	Required	Default Value	Documentation
customer_OverdueInterest	decimal	<input type="checkbox"/>		
customer_PaymentTerm	int32	<input type="checkbox"/>		
customer_PricelistGUID	string	<input type="checkbox"/>		
customer_ReverseCharge	boolean	<input type="checkbox"/>		
customer_ReverseChargeDescription	string	<input type="checkbox"/>		
customer_TimezoneGUID	string	<input type="checkbox"/>		
customer_UpdatedTS	datetime	<input type="checkbox"/>		
customer_UrIToAccount	string	<input type="checkbox"/>		
customer_VatNumber	string	<input type="checkbox"/>		
customer_Website	string	<input type="checkbox"/>		

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	string			
ACCOUNTOWNERUSERGUID	string	Account Owner User GUID		
ACCOUNTRATING	int32	Account Rating		
ANNUALREVENUE	int32	Annual Revenue		
ANNUALREVENUE2	int64			
COMPANYGUID	string	Company GUID		
CURRENCYCODE	string	Currency Code		
CURRENCYGUID	string	Currency GUID		
EINVOICEADDRESS	string	e-invoice Address		
EINVOICEOPERATOR	string	e-invoice Operator		
EMAIL	string	Email		
EMPLOYEES	int32	Employees		
GUID	string	GUID		
HEADOFFICEADDRESSGUID	string	Head Office Address GUID		
INDUSTRYGUID	string	Industry GUID		
INSERTTS	datetime	Inserted		
INVOICINGVAT	decimal	Invoicing VAT		
ISACTIVE	boolean	Is Active		
ISINTERNAL	boolean	Is Internal		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
NAME	string	Name		

Name	Data Type	Label	Required	Documentation
NOTES	string	Notes		
NUMBER	int64	Number		
OVERDUEINTEREST	decimal	Overdue Interest		
PAYMENTTERM	int32	Payment Term		
PRICELISTGUID	string	Pricelist GUID		
REVERSECHARGE	boolean	Reverse Charge		
REVERSECHARGEDESCRIPTION	string	Reverse Charge Description		
TIMEZONEGUID	string	Timezone GUID		
UPDATEDTS	datetime	Updated		
URLTOACCOUNT	string	URL to Account		
VATNUMBER	string	VAT Number		
WEBSITE	string	Website		

3.1.234 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 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	string	<input type="checkbox"/>		
employmentInfo_CompanyBusinessUnitGUID	string	<input type="checkbox"/>		
employmentInfo_CompanyBusinessUnitName	string	<input type="checkbox"/>		
employmentInfo_DailyHours	decimal	<input type="checkbox"/>		
employmentInfo_EndDate	datetime	<input type="checkbox"/>		
employmentInfo_GUID	string	<input type="checkbox"/>		
employmentInfo_HourCost	decimal	<input type="checkbox"/>		
employmentInfo_IsOvertimeAllowed	boolean	<input type="checkbox"/>		
employmentInfo_StartDate	datetime	<input type="checkbox"/>		
employmentInfo_Title	string	<input type="checkbox"/>		

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	string	Company Business Unit Code		
COMPANYBUSINESSUNITGUID	string	Company Business Unit GUID		
COMPANYBUSINESSUNITNAME	string	Company Business Unit Name		
DAILYHOURS	decimal	Daily Hours		
ENDDATE	datetime	End Date		
GUID	string	GUID		
HOURLCOST	decimal	Hour Cost		
ISOVERTIMEALLOWED	boolean	Is Overtime Allowed		
STARTDATE	datetime	Start Date		
TITLE	string	Title		

3.1.235 UpdateHourEntry: Visma Severa Update Hour Entry

Catalog: Severa

Schema: API

Label: Update 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- 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	string	<input type="checkbox"/>		
hourEntryInfo_CaseBusinessUnitCode	string	<input type="checkbox"/>		
hourEntryInfo_CaseCostCenterIdentifier	string	<input type="checkbox"/>		
hourEntryInfo_CaseGUID	string	<input type="checkbox"/>		
hourEntryInfo_CaseName	string	<input type="checkbox"/>		
hourEntryInfo_CaseNumber	int64	<input type="checkbox"/>		
hourEntryInfo_CostCurrencyShortform	string	<input type="checkbox"/>		

Name	Data Type	Required	Default Value	Documentation
hourEntryInfo_CostCurrencyShortformOfInvoicingCurrency	string	<input type="checkbox"/>		
hourEntryInfo_CostCurrencyShortformOfOrganizationCurrency	string	<input type="checkbox"/>		
hourEntryInfo_CostCurrencyShortformOfUserCompanyCurrency	string	<input type="checkbox"/>		
hourEntryInfo_Description	string	<input type="checkbox"/>		
hourEntryInfo_EndTime	datetime	<input type="checkbox"/>		
hourEntryInfo_EventDate	datetime	<input type="checkbox"/>		
hourEntryInfo_GUID	string	<input type="checkbox"/>		
hourEntryInfo_InvoiceDescription	string	<input type="checkbox"/>		
hourEntryInfo_InvoiceGUID	string	<input type="checkbox"/>		
hourEntryInfo_InvoiceQuantity	decimal	<input type="checkbox"/>		
hourEntryInfo_InvoiceRow GUID	string	<input type="checkbox"/>		
hourEntryInfo_IsApproved	boolean	<input type="checkbox"/>		
hourEntryInfo_IsBillable	boolean	<input type="checkbox"/>		
hourEntryInfo_IsProductive	boolean	<input type="checkbox"/>		
hourEntryInfo_IsReadOnly	boolean	<input type="checkbox"/>		
hourEntryInfo_OvertimeCode	string	<input type="checkbox"/>		
hourEntryInfo_OverTimeGUID	string	<input type="checkbox"/>		
hourEntryInfo_PhaseGUID	string	<input type="checkbox"/>		
hourEntryInfo_PriceCurrencyShortformOfCaseCompanyCurrency	string	<input type="checkbox"/>		
hourEntryInfo_PriceCurrencyShortformOfInvoicingCurrency	string	<input type="checkbox"/>		
hourEntryInfo_PriceCurrencyShortformOfOrganizationCurrency	string	<input type="checkbox"/>		
hourEntryInfo_PriceCurrencyShortformOfUserCompanyCurrency	string	<input type="checkbox"/>		
hourEntryInfo_Quantity	decimal	<input type="checkbox"/>		
hourEntryInfo_StartTime	datetime	<input type="checkbox"/>		
hourEntryInfo_TaskGUID	string	<input type="checkbox"/>		
hourEntryInfo_UnitCost	decimal	<input type="checkbox"/>		
hourEntryInfo_UnitCostInInvoicingCurrency	decimal	<input type="checkbox"/>		
hourEntryInfo_UnitCostInOrganizationCurrency	decimal	<input type="checkbox"/>		
hourEntryInfo_UnitCostInUserCompanyCurrency	decimal	<input type="checkbox"/>		
hourEntryInfo_UnitPrice	decimal	<input type="checkbox"/>		
hourEntryInfo_UnitPriceInBaseCurrency	decimal	<input type="checkbox"/>		
hourEntryInfo_UnitPriceInCaseCompanyCurrency	decimal	<input type="checkbox"/>		

Name	Data Type	Required	Default Value	Documentation
hourEntryInfo_UnitPriceInUserCompanyCurrency	decimal	<input type="checkbox"/>		
hourEntryInfo_UseInvoiceQuantity	boolean	<input type="checkbox"/>		
hourEntryInfo_UserBusinessUnitCode	string	<input type="checkbox"/>		
hourEntryInfo_UserCode	string	<input type="checkbox"/>		
hourEntryInfo_UserFirstName	string	<input type="checkbox"/>		
hourEntryInfo_UserGUID	string	<input type="checkbox"/>		
hourEntryInfo_UserLastName	string	<input type="checkbox"/>		
hourEntryInfo_WorkTypeCode	string	<input type="checkbox"/>		
hourEntryInfo_WorkTypeGUID	string	<input type="checkbox"/>		

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	string	Account GUID		
CASEBUSINESSUNITCODE	string	Project Business Unit Code		
CASECOSTCENTERIDENTIFIER	string	Project Cost Center Identifier		
CASEGUID	string	Project GUID		
CASENAME	string	Project Name		
CASENUMBER	int64	Project Number		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User Company Currency		
DESCRIPTION	string	Description		
ENDTIME	datetime	End Time		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVICEROWGUID	string	Invoice Line GUID		

Name	Data Type	Label	Required	Documentation
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISPRODUCTIVE	boolean	Is Productive		
ISREADONLY	boolean	Is Read-only		
OVERTIMECODE	string	Overtime Code		
OVERTIMEGUID	string	Overtime GUID		
PHASEGUID	string	Phase GUID		
PRICECURRENCYSHORTFORMOF COMPANYS CURRENCY	string	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOF INVOICINGCURRENCY	string	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOF ORGANIZATIONCURRENCY	string	Price Currency Short Form of Organization Currency		
PRICECURRENCYSHORTFORMOF USERCOMPANYS CURRENCY	string	Price Currency Short Form of User Company Currency		
QUANTITY	decimal	Quantity		
STARTTIME	datetime	Start Time		
TASKGUID	string	Task GUID		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	decimal	Unit Cost in Organization Currency		
UNITCOSTINUSERCOMPANYS CURRENCY	decimal	Unit Cost in User Company Currency		
UNITPRICE	decimal	Unit Price		
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		
UNITPRICEINCOMPANYS CURRENCY	decimal	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYS CURRENCY	decimal	Unit Price in User Company Currency		
USEINVOICEQUANTITY	boolean			
USERBUSINESSUNITCODE	string	User Business Unit Code		
USERCODE	string	User Code		
USERFIRSTNAME	string	User First Name		
USERGUID	string	User GUID		
USERLASTNAME	string	User Last Name		
WORKTYPECODE	string	Work Type Code		
WORKTYPEGUID	string	Work Type GUID		

3.1.236 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	string	<input type="checkbox"/>		
phaseInfo_Code	string	<input type="checkbox"/>		
phaseInfo_Deadline	datetime	<input type="checkbox"/>		
phaseInfo_Description	string	<input type="checkbox"/>		
phaseInfo_FlatRate_AdditionalHoursUnitPrice	decimal	<input type="checkbox"/>		
phaseInfo_FlatRate_AreAdditionalHoursBillable	boolean	<input type="checkbox"/>		
phaseInfo_FlatRate_Hours	decimal	<input type="checkbox"/>		
phaseInfo_FlatRate_Price	decimal	<input type="checkbox"/>		
phaseInfo_GUID	string	<input type="checkbox"/>		
phaseInfo_IsCompleted	boolean	<input type="checkbox"/>		
phaseInfo_IsLocked	boolean	<input type="checkbox"/>		
phaseInfo_Name	string	<input type="checkbox"/>		

Name	Data Type	Required	Default Value	Documentation
phaseInfo_OriginalDeadline	datetime	<input type="checkbox"/>		
phaseInfo_OriginalPlannedStartDate	datetime	<input type="checkbox"/>		
phaseInfo_OriginalWorkEstimate	decimal	<input type="checkbox"/>		
phaseInfo_OwnerUserGUID	string	<input type="checkbox"/>		
phaseInfo_ParentPhaseGUID	string	<input type="checkbox"/>		
phaseInfo_PlannedStartDate	datetime	<input type="checkbox"/>		
phaseInfo_PricePerHour	decimal	<input type="checkbox"/>		
phaseInfo_SortOrder	int32	<input type="checkbox"/>		
phaseInfo_WorkEstimate	decimal	<input type="checkbox"/>		
phaseInfo_WorkTypeGUID	string	<input type="checkbox"/>		

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	string	Project GUID		
CODE	string	Code		
DEADLINE	datetime	Deadline		
DESCRIPTION	string	Ddescription		
FLATRATE_ADDITIONALHOURSUNITPRICE	decimal	Flat Rate Additional Hours Unit Price		
FLATRATE_AREADDITIONALHOURSBILLABLE	boolean	Flat Rate Are additional Hours Billable		
FLATRATE_HOURS	decimal	Flat Rate Hours		
FLATRATE_PRICE	decimal	Flat Rate Price		
GUID	string	GUID		
ISCOMPLETED	boolean	Is Completed		
ISLOCKED	boolean	Is Locked		
NAME	string	Name		
ORIGINALDEADLINE	datetime	Original Deadline		
ORIGINALPLANNEDSTARTDATE	datetime	Original Planned Start Date		
ORIGINALWORKESTIMATE	decimal	Original Work Estimate		
OWNERUSERGUID	string	Owner User GUID		
PARENTPHASEGUID	string	Parent Phase GUID		
PLANNEDSTARTDATE	datetime	Planned Start Date		
PRICEPERHOUR	decimal	Price per Hour		
SORTORDER	int32	Sort Order		
WORKESTIMATE	decimal	Work Estimate		

Name	Data Type	Label	Required	Documentation
WORKTYPEGUID	string	Work Type GUID		

3.1.237 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_UselnWorkTimeEntry- 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	string	<input type="checkbox"/>		
productInfo_Description	string	<input type="checkbox"/>		
productInfo_GUID	string	<input type="checkbox"/>		
productInfo_IsActive	boolean	<input type="checkbox"/>		
productInfo_MeasurementUnit	string	<input type="checkbox"/>		
productInfo_Name	string	<input type="checkbox"/>		
productInfo_ProductCategoryGUID	string	<input type="checkbox"/>		
productInfo_SalesAccountGUID	string	<input type="checkbox"/>		
productInfo_Type	string	<input type="checkbox"/>		
productInfo_UnitCost	decimal	<input type="checkbox"/>		
productInfo_UnitPrice	decimal	<input type="checkbox"/>		
productInfo_UselnWorkTimeEntry	boolean	<input type="checkbox"/>		
productInfo_VAT	decimal	<input type="checkbox"/>		

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	string	Code		
DESCRIPTION	string	Ddescription		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
MEASUREMENTUNIT	string	Measurement Unit		
NAME	string	Name		
PRODUCTCATEGORYGUID	string	Product Category GUID		
SALESACCOUNTGUID	string	Sales Account GUID		
TYPE	string	Type		
UNITCOST	decimal	Unit Cost		
UNITPRICE	decimal	Unit Price		
USEINWORKTIMEENTRY	boolean	Use in Work Time Entry		
VAT	decimal	VAT		

3.1.238 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 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	string	<input type="checkbox"/>		
productCategoryInfo_GUID	string	<input type="checkbox"/>		
productCategoryInfo_IsActive	boolean	<input type="checkbox"/>		
productCategoryInfo_IsDefault	boolean	<input type="checkbox"/>		
productCategoryInfo_Name	string	<input type="checkbox"/>		

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	string	Code		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
NAME	string	Name		

3.1.239 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 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	string	<input type="checkbox"/>		
resourceInfo_IsActive	boolean	<input type="checkbox"/>		
resourceInfo_Name	string	<input type="checkbox"/>		

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	string	GUID		
ISACTIVE	boolean	Is Active		
NAME	string	Name		

3.1.240 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.

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	decimal	<input type="checkbox"/>		
resourceAllocationInfo_AllocatedPercentage	int32	<input type="checkbox"/>		
resourceAllocationInfo_CaseGUID	string	<input type="checkbox"/>		
resourceAllocationInfo_EndDate	datetime	<input type="checkbox"/>		
resourceAllocationInfo_GUID	string	<input type="checkbox"/>		
resourceAllocationInfo_PhaseGUID	string	<input type="checkbox"/>		
resourceAllocationInfo_StartDate	datetime	<input type="checkbox"/>		
resourceAllocationInfo_UserGUID	string	<input type="checkbox"/>		

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	decimal	Allocated Hours		
ALLOCATEDPERCENTAGE	int32	Allocated Percentage		
CASEGUID	string	Project GUID		
ENDDATE	datetime	End Date		
GUID	string	GUID		
PHASEGUID	string	Phase GUID		
STARTDATE	datetime	Start Date		
USERGUID	string	User GUID		

3.1.241 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_UsedInWorkTimeEntry- 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	string	<input type="checkbox"/>		
travelExpenseInfo_CostCurrencyGUID	string	<input type="checkbox"/>		
travelExpenseInfo_Description	string	<input type="checkbox"/>		
travelExpenseInfo_ExpenseClass	string	<input type="checkbox"/>		
travelExpenseInfo_GUID	string	<input type="checkbox"/>		
travelExpenseInfo_IsActive	boolean	<input type="checkbox"/>		
travelExpenseInfo_MeasurementUnit	string	<input type="checkbox"/>		
travelExpenseInfo_Name	string	<input type="checkbox"/>		
travelExpenseInfo_Type	string	<input type="checkbox"/>		
travelExpenseInfo_UnitCost	decimal	<input type="checkbox"/>		
travelExpenseInfo_UnitPrice	decimal	<input type="checkbox"/>		
travelExpenseInfo_UseInWorkTimeEntry	boolean	<input type="checkbox"/>		
travelExpenseInfo_VAT	decimal	<input type="checkbox"/>		

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	string	Code		
COSTCURRENCYGUID	string	Cost Currency GUID		
DESCRIPTION	string	Description		
EXPENSECLASS	string	Expense Class		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		

Name	Data Type	Label	Required	Documentation
MEASUREMENTUNIT	string	Measurement Unit		
NAME	string	Name		
TYPE	string	Type		
UNITCOST	decimal	Unit Cost		
UNITPRICE	decimal	Unit Price		
USEINWORKTIMEENTRY	boolean	Use in Work Time Entry		
VAT	decimal	VAT		

3.1.242 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.

Name	Data Type	Required	Default Value	Documentation
userInfo_Addressline	string	<input type="checkbox"/>		

Name	Data Type	Required	Default Value	Documentation
userInfo_BankAccountNumber	string	<input type="checkbox"/>		
userInfo_BusinessUnitCode	string	<input type="checkbox"/>		
userInfo_BusinessUnitGUID	string	<input type="checkbox"/>		
userInfo_BusinessUnitName	string	<input type="checkbox"/>		
userInfo_Code	string	<input type="checkbox"/>		
userInfo_CountryCode	string	<input type="checkbox"/>		
userInfo_CountryGUID	string	<input type="checkbox"/>		
userInfo_CountryRegionGUID	string	<input type="checkbox"/>		
userInfo_CountryRegionName	string	<input type="checkbox"/>		
userInfo_Culture	string	<input type="checkbox"/>		
userInfo_DefaultActivityTypeGUID	string	<input type="checkbox"/>		
userInfo_Email	string	<input type="checkbox"/>		
userInfo_FirstName	string	<input type="checkbox"/>		
userInfo_GUID	string	<input type="checkbox"/>		
userInfo_IsActive	boolean	<input type="checkbox"/>		
userInfo_LanguageCode	string	<input type="checkbox"/>		
userInfo_LanguageGUID	string	<input type="checkbox"/>		
userInfo_LastName	string	<input type="checkbox"/>		
userInfo_LicenseType	string	<input type="checkbox"/>		
userInfo_Notes	string	<input type="checkbox"/>		
userInfo_OrganizationGUID	string	<input type="checkbox"/>		
userInfo_Phone	string	<input type="checkbox"/>		
userInfo_PhotoFileGUID	string	<input type="checkbox"/>		
userInfo_PostalCode	string	<input type="checkbox"/>		
userInfo_PostOffice	string	<input type="checkbox"/>		
userInfo_ProfileGUID	string	<input type="checkbox"/>		
userInfo_Salutation	string	<input type="checkbox"/>		
userInfo_SendDailyMail	boolean	<input type="checkbox"/>		
userInfo_SendWeeklyMail	boolean	<input type="checkbox"/>		
userInfo_SocialSecurityNumber	string	<input type="checkbox"/>		
userInfo_SuperiorUserGUID	string	<input type="checkbox"/>		
userInfo_TimezoneGUID	string	<input type="checkbox"/>		
userInfo_Title	string	<input type="checkbox"/>		
userInfo_WorktypeGUID	string	<input type="checkbox"/>		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ADDRESSLINE	string	Addressline		
BANKACCOUNTNUMBER	string	Bank Account Number		
BUSINESSUNITCODE	string	Business Unit Code		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
CODE	string	Code		
COUNTRYCODE	string	Country Code		
COUNTRYGUID	string	Country GUID		
COUNTRYREGIONGUID	string	Country Region GUID		
COUNTRYREGIONNAME	string	Country Region Name		
CULTURE	string	Culture		
DEFAULTACTIVITYTYPEGUID	string	Default Activity Type GUID		
EMAIL	string	Email		
FIRSTNAME	string	First Name		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
LASTNAME	string	Last Name		
LICENSETYPE	string			
NOTES	string	Notes		
ORGANIZATIONGUID	string	Organization GUID		
PHONE	string	Phone		
PHOTOFILEGUID	string	Photo File GUID		
POSTALCODE	string	Postal Code		
POSTOFFICE	string	Post Office		
PROFILEGUID	string	Profile GUID		
SALUTATION	string	Salutation		
SENDDAILYMAIL	boolean	Send Daily Mail		
SENDWEEKLYMAIL	boolean	Send Weekly Mail		
SOCIALSECURITYNUMBER	string	Social Security Number		
SUPERIORUSERGUID	string	Superior User GUID		
TIMEZONEGUID	string	Timezone GUID		
TITLE	string	Title		
WORKTYPEGUID	string	Work Type GUID		

3.1.243 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 Invariant UniversalSQL driver 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	string	<input type="checkbox"/>		
rights_API	boolean	<input type="checkbox"/>		
rights_Rights_Account	string	<input type="checkbox"/>		
rights_Rights_AccountDelete	boolean	<input type="checkbox"/>		
rights_Rights_Administrator	string	<input type="checkbox"/>		
rights_Rights_Case	string	<input type="checkbox"/>		
rights_Rights_CaseDelete	boolean	<input type="checkbox"/>		
rights_Rights_CaseOwnerWorkHourApproval	boolean	<input type="checkbox"/>		
rights_Rights_ScheduleJobs	string	<input type="checkbox"/>		
rights_Rights_Sharing	boolean	<input type="checkbox"/>		
rights_Rights_TravelReimbursement	string	<input type="checkbox"/>		
rights_Rights_Users	string	<input type="checkbox"/>		
rights_Rights_WorkHourApproval	string	<input type="checkbox"/>		

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	string			
RIGHTS_ACCOUNTDELETE	boolean			
RIGHTS_ADMINISTRATOR	string			
RIGHTS_CASE	string			
RIGHTS_CASEDELETE	boolean			
RIGHTS_CASEOWNERWORKHOUR APPROVAL	boolean			
RIGHTS_SCHEDULEJOBS	string			
RIGHTS_SHARING	boolean			
RIGHTS_TRAVELREIMBURSEMENT	string			
RIGHTS_USERS	string			
RIGHTS_WORKHOURAPPROVAL	string			

3.1.244 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

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	string	<input type="checkbox"/>		
workTypeInfo_GUID	string	<input type="checkbox"/>		
workTypeInfo_IsActive	boolean	<input type="checkbox"/>		
workTypeInfo_IsDefault	boolean	<input type="checkbox"/>		
workTypeInfo_IsProductive	boolean	<input type="checkbox"/>		
workTypeInfo_Name	string	<input type="checkbox"/>		
workTypeInfo_SalesAccountGUID	string	<input type="checkbox"/>		

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	string	Code		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
ISPRODUCTIVE	boolean	Is Productive		
NAME	string	Name		
SALESACCOUNTGUID	string	Sales Account GUID		

3.1.245 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

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	string	<input type="checkbox"/>		

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	string	Addressline		
BANKACCOUNTNUMBER	string	Bank Account Number		
BUSINESSUNITCODE	string	Business Unit Code		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
CODE	string	Code		
COUNTRYCODE	string	Country Code		
COUNTRYGUID	string	Country GUID		
COUNTRYREGIONGUID	string	Country Region GUID		
COUNTRYREGIONNAME	string	Country Region Name		
CULTURE	string	Culture		
DEFAULTACTIVITYTYPEGUID	string	Default Activity Type GUID		
EMAIL	string	Email		
FIRSTNAME	string	First Name		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
LASTNAME	string	Last Name		
LICENSETYPE	string			
NOTES	string	Notes		
ORGANIZATIONGUID	string	Organization GUID		
PHONE	string	Phone		
PHOTOFILEGUID	string	Photo File GUID		
POSTALCODE	string	Postal Code		
POSTOFFICE	string	Post Office		
PROFILEGUID	string	Profile GUID		

Name	Data Type	Label	Required	Documentation
SALUTATION	string	Salutation		
SENDDAILYMAIL	boolean	Send Daily Mail		
SENDWEEKLYMAIL	boolean	Send Weekly Mail		
SOCIALSECURITYNUMBER	string	Social Security Number		
SUPERIORUSERGUID	string	Superior User GUID		
TIMEZONEGUID	string	Timezone GUID		
TITLE	string	Title		
WORKTYPEGUID	string	Work Type GUID		

3.1.246 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	string	<input type="checkbox"/>		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ADDRESSLINE	string	Addressline		
BANKACCOUNTNUMBER	string	Bank Account Number		
BUSINESSUNITCODE	string	Business Unit Code		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
CODE	string	Code		
COUNTRYCODE	string	Country Code		
COUNTRYGUID	string	Country GUID		
COUNTRYREGIONGUID	string	Country Region GUID		
COUNTRYREGIONNAME	string	Country Region Name		
CULTURE	string	Culture		
DEFAULTACTIVITYTYPEGUID	string	Default Activity Type GUID		
EMAIL	string	Email		
FIRSTNAME	string	First Name		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
LASTNAME	string	Last Name		
LICENSETYPE	string			
NOTES	string	Notes		
ORGANIZATIONGUID	string	Organization GUID		
PHONE	string	Phone		
PHOTOFILEGUID	string	Photo File GUID		
POSTALCODE	string	Postal Code		
POSTOFFICE	string	Post Office		
PROFILEGUID	string	Profile GUID		
SALUTATION	string	Salutation		
SENDDAILYMAIL	boolean	Send Daily Mail		
SENDWEEKLYMAIL	boolean	Send Weekly Mail		
SOCIALSECURITYNUMBER	string	Social Security Number		
SUPERIORUSERGUID	string	Superior User GUID		
TIMEZONEGUID	string	Timezone GUID		
TITLE	string	Title		
WORKTYPEGUID	string	Work Type GUID		

3.1.247 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	string	<input type="checkbox"/>		
lastName	string	<input type="checkbox"/>		

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	string	Addressline		
BANKACCOUNTNUMBER	string	Bank Account Number		
BUSINESSUNITCODE	string	Business Unit Code		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
CODE	string	Code		
COUNTRYCODE	string	Country Code		
COUNTRY GUID	string	Country GUID		
COUNTRYREGIONGUID	string	Country Region GUID		
COUNTRYREGIONNAME	string	Country Region Name		
CULTURE	string	Culture		

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

3.1.248 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

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	string	<input type="checkbox"/>		

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	string			
RIGHTS_ACCOUNTDELETE	boolean			
RIGHTS_ADMINISTRATOR	string			
RIGHTS_CASE	string			
RIGHTS_CASEDELETE	boolean			
RIGHTS_CASEOWNERWORKHOUR APPROVAL	boolean			
RIGHTS_SCHEDULEJOBS	string			
RIGHTS_SHARING	boolean			
RIGHTS_TRAVELREIMBURSEMENT	string			
RIGHTS_USERS	string			
RIGHTS_WORKHOURAPPROVAL	string			

3.1.249 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

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	string	<input type="checkbox"/>		
includeInactive	boolean	<input type="checkbox"/>		

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	string	Addressline		
BANKACCOUNTNUMBER	string	Bank Account Number		
BUSINESSUNITCODE	string	Business Unit Code		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
CODE	string	Code		
COUNTRYCODE	string	Country Code		
COUNTRYGUID	string	Country GUID		
COUNTRYREGIONGUID	string	Country Region GUID		
COUNTRYREGIONNAME	string	Country Region Name		
CULTURE	string	Culture		
DEFAULTACTIVITYTYPEGUID	string	Default Activity Type GUID		
EMAIL	string	Email		
FIRSTNAME	string	First Name		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
LASTNAME	string	Last Name		
LICENSETYPE	string			

Name	Data Type	Label	Required	Documentation
NOTES	string	Notes		
ORGANIZATIONGUID	string	Organization GUID		
PHONE	string	Phone		
PHOTOFILEGUID	string	Photo File GUID		
POSTALCODE	string	Postal Code		
POSTOFFICE	string	Post Office		
PROFILEGUID	string	Profile GUID		
SALUTATION	string	Salutation		
SENDDAILYMAIL	boolean	Send Daily Mail		
SENDWEEKLYMAIL	boolean	Send Weekly Mail		
SOCIALSECURITYNUMBER	string	Social Security Number		
SUPERIORUSERGUID	string	Superior User GUID		
TIMEZONEGUID	string	Timezone GUID		
TITLE	string	Title		
WORKTYPEGUID	string	Work Type GUID		

3.1.250 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.

Name	Data Type	Required	Default Value	Documentation
guid	string	<input type="checkbox"/>		

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	string	Addressline		
BANKACCOUNTNUMBER	string	Bank Account Number		
BUSINESSUNITCODE	string	Business Unit Code		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
CODE	string	Code		
COUNTRYCODE	string	Country Code		
COUNTRYGUID	string	Country GUID		
COUNTRYREGIONGUID	string	Country Region GUID		
COUNTRYREGIONNAME	string	Country Region Name		
CULTURE	string	Culture		
DEFAULTACTIVITYTYPEGUID	string	Default Activity Type GUID		
EMAIL	string	Email		
FIRSTNAME	string	First Name		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
LASTNAME	string	Last Name		
LICENSETYPE	string			
NOTES	string	Notes		
ORGANIZATIONGUID	string	Organization GUID		
PHONE	string	Phone		
PHOTOFILEGUID	string	Photo File GUID		
POSTALCODE	string	Postal Code		
POSTOFFICE	string	Post Office		
PROFILEGUID	string	Profile GUID		
SALUTATION	string	Salutation		
SENDDAILYMAIL	boolean	Send Daily Mail		
SENDWEEKLYMAIL	boolean	Send Weekly Mail		
SOCIALSECURITYNUMBER	string	Social Security Number		
SUPERIORUSERGUID	string	Superior User GUID		

Name	Data Type	Label	Required	Documentation
TIMEZONEGUID	string	Timezone GUID		
TITLE	string	Title		
WORKTYPEGUID	string	Work Type GUID		

3.1.251 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	string	<input type="checkbox"/>		

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	string	Addressline		
BANKACCOUNTNUMBER	string	Bank Account Number		
BUSINESSUNITCODE	string	Business Unit Code		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		

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

3.1.252 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.

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	string	<input type="checkbox"/>		
userKeywords	string	<input type="checkbox"/>		

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	string	Addressline		
BANKACCOUNTNUMBER	string	Bank Account Number		
BUSINESSUNITCODE	string	Business Unit Code		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
CODE	string	Code		
COUNTRYCODE	string	Country Code		
COUNTRYGUID	string	Country GUID		
COUNTRYREGIONGUID	string	Country Region GUID		
COUNTRYREGIONNAME	string	Country Region Name		
CULTURE	string	Culture		
DEFAULTACTIVITYTYPEGUID	string	Default Activity Type GUID		
EMAIL	string	Email		
FIRSTNAME	string	First Name		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
LANGUAGECODE	string	Language Code		

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

3.1.253 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

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	string	<input type="checkbox"/>		
options	int32	<input type="checkbox"/>		
startDateTime	datetime	<input type="checkbox"/>		

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	string	Addressline		
BANKACCOUNTNUMBER	string	Bank Account Number		
BUSINESSUNITCODE	string	Business Unit Code		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
CODE	string	Code		
COUNTRYCODE	string	Country Code		
COUNTRYGUID	string	Country GUID		
COUNTRYREGIONGUID	string	Country Region GUID		
COUNTRYREGIONNAME	string	Country Region Name		
CULTURE	string	Culture		
DEFAULTACTIVITYTYPEGUID	string	Default Activity Type GUID		
EMAIL	string	Email		
FIRSTNAME	string	First Name		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
LASTNAME	string	Last Name		
LICENSETYPE	string			
NOTES	string	Notes		
ORGANIZATIONGUID	string	Organization GUID		
PHONE	string	Phone		
PHOTOFILEGUID	string	Photo File GUID		
POSTALCODE	string	Postal Code		
POSTOFFICE	string	Post Office		
PROFILEGUID	string	Profile GUID		
SALUTATION	string	Salutation		
SENDDAILYMAIL	boolean	Send Daily Mail		

Name	Data Type	Label	Required	Documentation
SENDWEEKLYMAIL	boolean	Send Weekly Mail		
SOCIALSECURITYNUMBER	string	Social Security Number		
SUPERIORUSERGUID	string	Superior User GUID		
TIMEZONEGUID	string	Timezone GUID		
TITLE	string	Title		
WORKTYPEGUID	string	Work Type GUID		

3.1.254 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	string	Context		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
KEYWORD	string	Keyw ord		
WEIGHT	int32	Weight		

3.1.255 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	string			

3.1.256 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	string	<input type="checkbox"/>		
enddate	datetime	<input type="checkbox"/>		
includeInfoForInactiveUser	boolean	<input type="checkbox"/>		
includeUnPaidAbsenses	boolean	<input type="checkbox"/>		
startdate	datetime	<input type="checkbox"/>		
userGUID	string	<input type="checkbox"/>		

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	string			
DATE	datetime	Date		
ENTRIES	string			
EXPECTEDHOURS	decimal	Expected Hours		
OVERTIMEALLOWED	boolean	Overtime Allowed		

Name	Data Type	Label	Required	Documentation
READONLY	boolean	Read-only		
READY	boolean	Ready		
TOTALHOURS	decimal	Total Hours		

3.1.257 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	string	<input type="checkbox"/>		

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	string	Code		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
ISPRODUCTIVE	boolean	Is Productive		

Name	Data Type	Label	Required	Documentation
NAME	string	Name		
SALESACCOUNTGUID	string	Sales Account GUID		

3.1.258 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	string	<input type="checkbox"/>		

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	string	Code		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
ISPRODUCTIVE	boolean	Is Productive		
NAME	string	Name		

Name	Data Type	Label	Required	Documentation
SALESACCOUNTGUID	string	Sales Account GUID		

3.1.259 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	string	Code		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
ISPRODUCTIVE	boolean	Is Productive		
NAME	string	Name		
SALESACCOUNTGUID	string	Sales Account GUID		

3.1.260 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.

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	string	<input type="checkbox"/>		

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	string	Code		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
ISPRODUCTIVE	boolean	Is Productive		
NAME	string	Name		
SALESACCOUNTGUID	string	Sales Account GUID		

3.1.261 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

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	int32	<input type="checkbox"/>		
startDateTime	datetime	<input type="checkbox"/>		

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	string	Code		
GUID	string	GUID		
ISACTIVE	boolean	Is Active		
ISDEFAULT	boolean	Is Default		
ISPRODUCTIVE	boolean	Is Productive		
NAME	string	Name		
SALESACCOUNTGUID	string	Sales Account GUID		

4 Schema: Native

4.1 Tables

4.1.1 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

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	boolean	BLOB Preferred	<input checked="" type="checkbox"/>	Indicator whether a BLOB result is preferred over text.
BOL_RESPONSE_CACHE_MAX_AGE_SEC	int32	Response Cache Maximum Age (sec)	<input type="checkbox"/>	Maximum age in seconds of Bridge Online response cache entries to be used.
CONTENT_TYPE	string(240)	Content Type	<input type="checkbox"/>	
DATE_ENDED	datetime	End Date	<input checked="" type="checkbox"/>	
DATE_STARTED	datetime	Start Date	<input checked="" type="checkbox"/>	
DRY_RUN	boolean	Run without Actions	<input checked="" type="checkbox"/>	
DURATION_MS	int64	Duration (ms)	<input checked="" type="checkbox"/>	
ERROR_MESSAGE_CODE	string(30)	Error Message Code	<input type="checkbox"/>	
ERROR_MESSAGE_TEXT	string(32000)	Error Message Text	<input type="checkbox"/>	
FAIL_ON_ERROR	boolean	Fail on Error	<input checked="" type="checkbox"/>	Whether to raise an exception when processing the native request triggered an error from the provider.
HTTP_DISK_CACHE_MAX_AGE_SEC	int32	HTTP Disk Cache Maximum Age (sec)	<input type="checkbox"/>	Maximum age in seconds of HTTP disk cache entries to be used.
HTTP_DISK_CACHE_SAVE	boolean	Save HTTP Disk Cache	<input type="checkbox"/>	Whether results can be stored in HTTP disk cache.
HTTP_DISK_CACHE_USE	boolean	Use HTTP Disk Cache	<input type="checkbox"/>	Whether results can be fetched from HTTP disk cache.
HTTP_MEMORY_CACHE_MAX_AGE_SEC	int32	HTTP Memory Cache Maximum Age (sec)	<input type="checkbox"/>	Maximum age in seconds of HTTP memory cache entries to be used.
HTTP_MEMORY_CACHE_SAVE	boolean	Save HTTP Memory Cache	<input type="checkbox"/>	Whether results can be stored in HTTP memory cache.
HTTP_MEMORY_CACHE_USE	boolean	Use HTTP Memory Cache	<input type="checkbox"/>	Whether results can be fetched from HTTP memory cache.
HTTP_METHOD	string(30)	HTTP Method	<input type="checkbox"/>	
HTTP_STATUS_CODE	int32	HTTP Status Code	<input type="checkbox"/>	
ORIG_SYSTEM_GROUP	string(4000)	Original System Group	<input type="checkbox"/>	
ORIG_SYSTEM_REFERENCE	string(4000)	Original System Reference	<input type="checkbox"/>	
PAYLOAD_TEXT	string	Payload	<input type="checkbox"/>	
RESULT_BLOB	byte[]	Result BLOB	<input type="checkbox"/>	
RESULT_DATE_TIME_UTC	datetime	Result Date Time	<input type="checkbox"/>	
RESULT_NUMBER	decimal	Result Number	<input type="checkbox"/>	
RESULT_TEXT	string	Result Text	<input type="checkbox"/>	
SUCCESSFUL	boolean	Successful	<input checked="" type="checkbox"/>	
TIMEOUT_SEC	int32	Timeout (sec)	<input type="checkbox"/>	Timeout in seconds.
TRANSACTION_ID	int64	Transaction ID	<input checked="" type="checkbox"/>	Incrementing ID of the transaction.

Name	Data Type	Label	Required	Documentation
URL	string(4000)	URL	<input type="checkbox"/>	

5 Schema: Views

5.1 Views

5.1.1 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	string	Account GUID		
CASEBUSINESSUNITCODE	string	Project Business Unit Code		
CASECOSTCENTERIDENTIFIER	string	Project Cost Center Identifier		
CASEGUID	string	Project GUID		
CASENAME	string	Project Name		
CASENUMBER	int64	Project Number		
COSTCURRENCYSHORTFORM	string	Cost Currency Short Form		
COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	string	Cost Currency Short Form of Invoicing Currency		
COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	string	Cost Currency Short Form of Organization Currency		
COSTCURRENCYSHORTFORMOFUSERCOMPANYCURRENCY	string	Cost Currency Short Form of User Company Currency		
cse_ACCOUNTGUID	string	Account GUID		
cse_ACCOUNTNAME	string	Account Name		
cse_ACCOUNTNUMBER	int64	Account Number		
cse_BILLINGADDRESS	string	Billing Address		
cse_BILLINGADDRESSGUID	string	Billing Address GUID		
cse_BILLINGINTERNALNOTES	string	Billing Internal Notes		
cse_BILLINGNOTESBEFOREGRID	string	Billing Notes before Grid		

Name	Data Type	Label	Required	Documentation
cse_BUSINESSUNITGUID	string	Business Unit GUID		
cse_BUSINESSUNITNAME	string	Business Unit Name		
cse_BUSINESSUNITNUMBER	string	Business Unit Number		
cse_CASECOMPANYCURRENCYGUID	string	Project Company Currency GUID		
cse_CASENUMBER	int64	Project Number		
cse_CASEOWNERUSERCODE	string	Project Owner User Code		
cse_CASEOWNERUSERGUID	string	Project Owner User GUID		
cse_CLOSEDDATE	datetime	Closed Date		
cse_COMPLETIONESTIMATE	int32	Completion Estimate		
cse_CONTACTGUID	string	Contact GUID		
cse_COSTCENTERGUID	string	Cost Center GUID		
cse_COSTCENTERNUMBER	string	Cost Center Number		
cse_CURRENCYGUID	string	Currency GUID		
cse_CURRENCYSHORTFORM	string	Currency Short Form		
cse_CURRENCYSHORTFORMOFCASECOMPANYCURRENCY	string	Currency Short Form of Project Company Currency		
cse_CURRENTCASESTATUSDESCRIPTION	string	Current Project Status Description		
cse_CURRENTCASESTATUSGUID	string	Current Project Status GUID		
cse_CURRENTCASESTATUSINSERTED	datetime	Current Project Status Inserted		
cse_DEADLINEDATE	datetime	Deadline Date		
cse_DESCRIPTION	string	Description		
cse_EXPECTEDVALUE	decimal	Expected Value		
cse_GUID	string	GUID		
cse_INCLUDEEXCEPTIONS	boolean	Include Exceptions		
cse_INCLUDEOVERTIME	boolean	Include Overtime		
cse_INSERTED	datetime	Inserted		
cse_INTERNALNAME	string	Internal Name		
cse_INVOICETEMPLATEGUID	string	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		
cse_LEADRATING	int32	Lead Rating		

Name	Data Type	Label	Required	Documentation
cse_LEADSOURCEGUID	string	Lead Source GUID		
cse_NAME	string	Name		
cse_ORDERNUMBER	string	Order Number		
cse_OURREFERENCE	string	Our Reference		
cse_OVERDUEINTEREST	decimal	Overdue Interest		
cse_PAYMENTTERM	int32	Payment Term		
cse_PRICELISTGUID	string	Pricelist GUID		
cse_PRICINGRULE	string	Pricing Rule		
cse_PRICINGTYPE	string	Pricing Type		
cse_PRIORITY	int32	Priority		
cse_PROBABILITY	int32	Probability		
cse_ROOTTASKGUID	string	Root Task GUID		
cse_SALESCLOSEDATE	datetime	Sales Close Date		
cse_SALESPERSONUSERCODE	string	Sales Person User Code		
cse_SALESPERSONUSERGUID	string	Sales Person User GUID		
cse_SALESPROCESSGUID	string	Sales Process GUID		
cse_SALESSTATUSGUID	string	Sales Status GUID		
cse_SHAREDTOEXTRANET	boolean	Shared to Extranet		
cse_STARTDATE	datetime	Start Date		
cse_UPDATEDTS	datetime	Updated		
cse_URLTOCASE	string	URL to Project		
cse_USEDEFAULTPRODUCTSINWORKTIMEENTRY	boolean	Use Default Products in Work Time Entry		
cse_USEDEFAULTWORKTYPES	boolean	Use Default Work Types		
cse_YOURREFERENCE	string	Your Reference		
DESCRIPTION	string	Description		
ENDTIME	datetime	End Time		
EVENTDATE	datetime	Event Date		
GUID	string	GUID		
INVOICEDESCRIPTION	string	Invoice Description		
INVOICEGUID	string	Invoice GUID		
INVOICEQUANTITY	decimal	Invoice Quantity		
INVICEROWGUID	string	Invoice Line GUID		
ISAPPROVED	boolean	Is Approved		
ISBILLABLE	boolean	Is Billable		
ISPRODUCTIVE	boolean	Is Productive		
ISREADONLY	boolean	Is Read-only		
OVERTIMECODE	string	Overtime Code		
OVERTIMEGUID	string	Overtime GUID		

Name	Data Type	Label	Required	Documentation
PHASEGUID	string	Phase GUID		
PRICECURRENCYSHORTFORMOF COMPANYS CURRENCY	string	Price Currency Short Form of Project Company Currency		
PRICECURRENCYSHORTFORMOF INVOICINGCURRENCY	string	Price Currency Short Form of Invoicing Currency		
PRICECURRENCYSHORTFORMOF ORGANIZATIONCURRENCY	string	Price Currency Short Form of Organization Currency		
PRICECURRENCYSHORTFORMOF USERCOMPANYS CURRENCY	string	Price Currency Short Form of User Company Currency		
QUANTITY	decimal	Quantity		
STARTTIME	datetime	Start Time		
TASKGUID	string	Task GUID		
UNITCOST	decimal	Unit Cost		
UNITCOSTININVOICINGCURRENCY	decimal	Unit Cost in Invoicing Currency		
UNITCOSTINORGANIZATIONCURRENCY	decimal	Unit Cost in Organization Currency		
UNITCOSTINUSERCOMPANYS CURRENCY	decimal	Unit Cost in User Company Currency		
UNITPRICE	decimal	Unit Price		
UNITPRICEINBASECURRENCY	decimal	Unit Price in Base Currency		
UNITPRICEINPROJECTCOMPANYS CURRENCY	decimal	Unit Price in Project Company Currency		
UNITPRICEINUSERCOMPANYS CURRENCY	decimal	Unit Price in User Company Currency		
USEINVOICEQUANTITY	boolean			
USERBUSINESSUNITCODE	string	User Business Unit Code		
USERCODE	string	User Code		
USERFIRSTNAME	string	User First Name		
USERGUID	string	User GUID		
USERLASTNAME	string	User Last Name		
WORKTYPECODE	string	Work Type Code		
WORKTYPEGUID	string	Work Type GUID		

5.1.2 Invoices: Visma Severa Invoices

Catalog: Severa

Schema: Views

Label: Invoices

This is a read-only view. The Visma Severa API may not support changing the data or the Invariantive UniversalSQL driver 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	string	Account GUID		
ACCOUNTNAME	string	Account Name		
ACCOUNTNUMBER	int64	Account Number		
BUSINESSUNITCODE	string	Business Unit Code		
BUSINESSUNITGUID	string	Business Unit GUID		
BUSINESSUNITNAME	string	Business Unit Name		
BUYERORGANIZATION_BUYERCONTACTPERSONNAME	string			
BUYERORGANIZATION_BUYERELECTRONICINVOICEADDRESS	string			
BUYERORGANIZATION_BUYERELECTRONICINVOICEOPERATOR	string			
BUYERORGANIZATION_BUYEREMAILADDRESSIDENTIFIER	string			
BUYERORGANIZATION_BUYERORGANISATIONNAME	string			
BUYERORGANIZATION_BUYERORGANISATIONTAXCODE	string			
BUYERORGANIZATION_BUYERPARTYIDENTIFIER	string			
BUYERORGANIZATION_BUYERPHONENUMBERIDENTIFIER	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRY	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRYCODE	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_STREETNAME	string			
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_TOWNNAME	string			
CULTURE	string	Culture		
CURRENCYGUID	string	Currency GUID		
CURRENCYNAME	string	Currency Name		
CURRENCYRATE	decimal	Currency Rate		
CURRENCYSHORTFORM	string	Currency Short Form		

Name	Data Type	Label	Required	Documentation
DATE	datetime	Date		
DESCRIPTION	string	Ddescription		
DUEDATE	datetime	Due Date		
ENTRYDATE	datetime	Entry Date		
GUID	string	GUID		
INVOICEALLOWEDCASEGUIDS	string			
INVOICECONFIG_BILLEXPENSE	string			
INVOICECONFIG_BILLHOUR	string			
INVOICECONFIG_BILLPRODUCT	string			
INVOICECONFIG_EXPENSEITEMDESCRIPTIONFORMAT	string			
INVOICECONFIG_HOURITEMDESCRIPTIONFORMAT	string			
INVOICECONFIG_HOURPRIMARYGROUPBY	string			
INVOICECONFIG_HOURSECONDARYGROUPBY	string			
INVOICECONFIG_ISTAXFREE	boolean			
INVOICECONFIG_PRODUCTITEMDESCRIPTIONFORMAT	string			
INVOICECONFIG_PRODUCTPRIMARYGROUPBY	string			
INVOICECONFIG_PRODUCTSECONDARYGROUPBY	string			
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	string			
INVOICECONFIG_TRAVELPRIMARYGROUPBY	string			
INVOICECONFIG_TRAVELSECONDARYGROUPBY	string			
INVOICEDETAILS_INVOICEDATE	datetime			

Name	Data Type	Label	Required	Documentation
INVOICEDETAILS_INVOICEDUEDATE	datetime			
INVOICEDETAILS_INVOICENUMBER	int32			
INVOICEDETAILS_INVOICETOTALVATAAMOUNT	decimal			
INVOICEDETAILS_INVOICETOTALVATEXCLUDEDAMOUNT	decimal			
INVOICEDETAILS_INVOICETOTALVATAINCLUDEDAMOUNT	decimal			
INVOICEDETAILS_INVOICETYPETEXT	string			
INVOICEDETAILS_ORDERIDENTIFIER	string			
INVOICEDETAILS_ORIGINCOD	string			
INVOICEDETAILS_PAYMENTOVERDUDEFINEFREETEXT	string			
INVOICEDETAILS_PAYMENTOVERDUDEFINEPERCENT	decimal			
INVOICEDETAILS_PAYMENTTERMSFREETEXT	string			
INVOICEDETAILS_SELLERREFERENCEIDENTIFIER	string			
INVOICEDETAILS_VATBASEAMOUNT	decimal			
INVOICEDETAILS_VATRATEAMOUNT	decimal			
INVOICEDETAILS_VATRATEPERCENT	int32			
INVOICENUMBERINGBUSINESSUNITGUID	string	Invoice Numbering Business Unit GUID		
INVOICEROWS	string			
INVOICESTATUS	string	Invoice Status		
ISCREDITNOTE	boolean	Is Credit Note		
ISREIMBURSED	boolean	Is Reimbursed		
ISTAXFREE	boolean	Is Tax Free		
LANGUAGECODE	string	Language Code		
LANGUAGEGUID	string	Language GUID		
MAINCASEGUID	string	Main Case GUID		
NOTES	string	Notes		
NOTESAFTERINVOICEROWS	string	Notes after Invoice Lines		
NOTESBEFOREINVOICEROWS	string	Notes before Invoice Lines		
NUMBER	int32	Number		
ORDERNUMBER	string	Order Number		
OURREFERENCE	string	Our Reference		
OVERDUEINTEREST	decimal	Overdue Interest		

Name	Data Type	Label	Required	Documentation
PAYMENTDATE	datetime	Payment Date		
PAYMENTSTATUS	string	Payment Status		
PAYMENTTERM	int32	Payment Term		
PONUMBER	string	PO Number		
RECEIVERADDRESSLINE	string	Receiver Addressline		
RECEIVERCITY	string	Receiver City		
RECEIVERCONTACTEMAIL	string	Receiver Contact Email		
RECEIVERCONTACTNAME	string	Receiver Contact Name		
RECEIVERCOUNTRY	string	Receiver Country		
RECEIVERCUSTOMERNAME	string	Receiver Customer Name		
RECEIVERPOSTALCODE	string	Receiver Postal Code		
RECEIVERSTATE	string	Receiver State		
RECEIVERVATNUMBER	string	Receiver VAT Number		
REFERENCENUMBER	string	Reference Number		
REIMBURSEINVOICEGUID	string	Reimburse Invoice GUID		
SELLERCONTACT_SELLERACCOUNTDETAILS	string			
SELLERCONTACT_SELLERCONTACTPERSONNAME	string			
SELLERCONTACT_SELLEREMAILADDRESSIDENTIFIER	string			
SELLERCONTACT_SELLERFAXNUMBER	string			
SELLERCONTACT_SELLERFREEXT	string			
SELLERCONTACT_SELLERHOMETOWNNAME	string			
SELLERCONTACT_SELLERPHONE NUMBER	string			
SELLERCONTACT_SELLERWEBADDRESSIDENTIFIER	string			
SELLERORGANIZATION_SELLERINVOICEADDRESS	string			
SELLERORGANIZATION_SELLERINVOICEOPERATOR	string			
SELLERORGANIZATION_SELLERORGANISATIONNAME	string			
SELLERORGANIZATION_SELLERORGANISATIONTAXCODE	string			
SELLERORGANIZATION_SELLERPARTYIDENTIFIER	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRY	string			

Name	Data Type	Label	Required	Documentation
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRYCODE	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_STREETNAME	string			
SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_TOWNNAME	string			
SENDERADDRESSLINE	string	Sender Addressline		
SENDERCITY	string	Sender City		
SENDERCONTACTNAME	string	Sender Contact Name		
SENDERCOUNTRY	string	Sender Country		
SENDERNAME	string	Sender Name		
SENDERPOSTALCODE	string	Sender Postal Code		
SENDERSTATE	string	Sender State		
SENDERVATNUMBER	string	Sender VAT Number		
TAXFREEDESCRIPTION	string	Tax Free Description		
TITLE	string	Title		
VIRTUALBANKBARCODE	string	Virtual Bank Barcode		
YOURREFERENCE	string	Your Reference		

Index

- A -

- Access Rights Profile by GUID 6
- Access Rights Profiles 7
- accessRightsProfile_GUID 33, 319
- accessRightsProfile_IsActive 33, 319
- accessRightsProfile_IsDefault 33, 319
- accessRightsProfile_Name 33, 319
- accessRightsProfile_Rights_Account 33, 319
- accessRightsProfile_Rights_AccountDelete 33, 319
- accessRightsProfile_Rights_Administrator 33, 319
- accessRightsProfile_Rights_Case 33, 319
- accessRightsProfile_Rights_CaseDelete 33, 319
- accessRightsProfile_Rights_CaseOwnerWorkHourApproval 33, 319
- accessRightsProfile_Rights_ScheduleJobs 33, 319
- accessRightsProfile_Rights_Sharing 33, 319
- accessRightsProfile_Rights_TravelReimbursement 33, 319
- accessRightsProfile_Rights_Users 33, 319
- accessRightsProfile_Rights_WorkHourApproval 33, 319
- AccessRightsProfileByGUID 6
- AccessRightsProfiles 7
- Account by GUID 8
- Account by Name 9
- Account by Number 11
- Account by VAT Number 12
- Account Group by Account GUID 14
- Account Groups 14
- Account GUID 22, 24, 26, 40, 51, 58, 70, 110, 112, 115, 119, 122, 125, 130, 131, 133, 134, 139, 179, 181, 184, 186, 189, 191, 193, 196, 202, 207, 212, 217, 222, 228, 267, 270, 272, 324, 332, 338, 343, 384, 387
- Account Name 51, 110, 112, 115, 119, 122, 125, 139, 193, 196, 202, 212, 217, 222, 228, 332, 384, 387
- Account Number 51, 110, 112, 115, 119, 122, 125, 139, 196, 202, 212, 217, 222, 228, 332, 384, 387
- Account Owner User GUID 8, 9, 11, 12, 16, 18, 19, 21, 35, 60, 148, 149, 151, 321, 339
- Account Rating 8, 9, 11, 12, 16, 18, 19, 21, 35, 60, 148, 149, 151, 321, 339
- AccountByGUID 8
- AccountByName 9
- AccountByNumber 11
- AccountByVatNumber 12
- ACCOUNTGROUPGUIDS 8, 9, 11, 12, 16, 18, 19, 21, 35, 39, 60, 148, 149, 151, 280, 321, 339
- accountGroupInfo_GUID 38
- accountGroupInfo_IsActive 38
- accountGroupInfo_Name 38
- accountGroupInfo_ParentGUID 38
- AccountGroups 14
- AccountGroupsByAccountGUID 14
- ACCOUNTGUID 8, 14, 22, 24, 26, 39, 40, 51, 58, 70, 101, 110, 112, 115, 119, 122, 125, 130, 131, 133, 134, 139, 148, 177, 178, 179, 181, 184, 186, 189, 191, 193, 196, 202, 207, 212, 217, 222, 228, 267, 270, 272, 280, 324, 332, 338, 343, 384, 387
- accountInfo_AccountGroupGUIDs 35, 321
- accountInfo_AccountOwnerUserGUID 35, 321
- accountInfo_AccountRating 35, 321
- accountInfo_CompanyGUID 35, 321
- accountInfo_CurrencyCode 35, 321
- accountInfo_CurrencyGUID 35, 321
- accountInfo_eInvoiceAddress 35, 321
- accountInfo_eInvoiceOperator 35, 321
- accountInfo_GUID 35, 321
- accountInfo_InsertTS 35, 321
- accountInfo_InvoicingVat 35, 321
- accountInfo_IsActive 35, 321
- accountInfo_IsInternal 35, 321
- accountInfo_LanguageCode 35, 321
- accountInfo_LanguageGUID 35, 321
- accountInfo_Name 35, 321
- accountInfo_Notes 35, 321
- accountInfo_Number 35, 321
- accountInfo_OverdueInterest 35, 321
- accountInfo_PaymentTerm 35, 321
- accountInfo_PricelistGUID 35, 321
- accountInfo_ReverseCharge 35, 321
- accountInfo_ReverseChargeDescription 35, 321
- accountInfo_UpdatedTS 35, 321
- accountInfo_UrlToAccount 35, 321
- ACCOUNTNAME 9, 51, 110, 112, 115, 119, 122, 125, 139, 193, 196, 202, 212, 217, 222, 228, 332, 387
- ACCOUNTNUMBER 11, 51, 110, 112, 115, 119, 122, 125, 139, 196, 202, 212, 217, 222, 228, 332, 387
- ACCOUNTOWNERUSERGUID 8, 9, 11, 12, 16, 18, 19, 21, 35, 60, 148, 149, 151, 321, 339
- ACCOUNTRATING 8, 9, 11, 12, 16, 18, 19, 21, 35, 60, 148, 149, 151, 321, 339
- Accounts 15
- Accounts by Company GUID 16
- Accounts by Insert Date 18
- Accounts Changed since 19
- Accounts Changed since Opts. 21
- AccountsByCompanyGUID 16
- AccountsByInsertDate 18
- AccountsChangedSince 19

- AccountsChangedSinceOpts 21
- Active Lead Sources 22
- ActiveLeadSources 22
- ACTIVITIES 377
- Activities by Date 22
- ActivitiesByDate 22
- Activity by GUID 24
- Activity Contact Participants 25
- Activity Instances 26
- Activity Participants 28
- Activity Resource Participants 29
- Activity Type by GUID 30
- Activity Type Code 22, 24, 26, 40, 324
- Activity Type GUID 22, 24, 26, 40, 324
- Activity Types 31
- Activity User Participants 31
- ActivityByGUID 24
- activityCategory 22
- ActivityContactParticipants 25
- activityGUID 24, 25, 28, 29, 31, 42, 43, 45, 154, 280, 281, 282
- activityInfo_AccountGUID 40, 324
- activityInfo_ActivityTypeCode 40, 324
- activityInfo_ActivityTypeGUID 40, 324
- activityInfo_CaseGUID 40, 324
- activityInfo_EndDateTime 40, 324
- activityInfo_GUID 40, 324
- activityInfo_IsAllDay 40, 324
- activityInfo_IsClosed 40, 324
- activityInfo_IsDuration 40, 324
- activityInfo_IsPrivate 40, 324
- activityInfo_IsReadOnly 40, 324
- activityInfo_IsUnassigned 40, 324
- activityInfo_Location 40, 324
- activityInfo_Name 40, 324
- activityInfo_Notes 40, 324
- activityInfo_OwnerUserBusinessUnitCode 40, 324
- activityInfo_OwnerUserCode 40, 324
- activityInfo_OwnerUserFirstName 40, 324
- activityInfo_OwnerUserGUID 40, 324
- activityInfo_OwnerUserLastName 40, 324
- activityInfo_StartDateTime 40, 324
- activityInfo_TaskGUID 40, 324
- ActivityInstances 26
- ActivityParticipants 28
- ActivityResourceParticipants 29
- ActivityTypeByGUID 30
- ACTIVITYTYPECODE 22, 24, 26, 40, 324
- ACTIVITYTYPEGUID 22, 24, 26, 30, 40, 155, 324
- activityTypeInfo_Category 44, 326
- activityTypeInfo_Code 44, 326
- activityTypeInfo_GUID 44, 326
- activityTypeInfo_Icon 44, 326
- activityTypeInfo_IsActive 44, 326
- activityTypeInfo_IsDefault 44, 326
- activityTypeInfo_IsPaidLeave 44, 326
- activityTypeInfo_Name 44, 326
- ActivityTypes 31
- ActivityUserParticipants 31
- Add New Access Rights Profile 33
- Add New Account 35
- Add New Account Group 38
- Add New Account Group Members 39
- Add New Activity 40
- Add New Activity Contact Participant 42
- Add New Activity Resource Participant 43
- Add New Activity Type 44
- Add New Activity User Participant 45
- Add New Address 46
- Add New Billing and Expense Forecast to Project 47
- Add New Business Unit 49
- Add New Contact 58
- Add New Contact Tag 59
- Add New Customer 60
- Add New Employment 63
- Add New Expense to Product by PO Number 64
- Add New File Tag 69
- Add New Hour Entry 70
- Add New Industry 74
- Add New Invoice Log Entry 75
- Add New Item 76
- Add New Items 81
- Add New Phase 84
- Add New Product 86
- Add New Product Category 87
- Add New Project 51
- Add New Project Status 56
- Add New Project Tag 57
- Add New Resource 88
- Add New Resource Allocation 89
- Add New Resource Allocations 91
- Add New Sales Process 92
- Add New Sales Status 93
- Add New Travel Expense 94
- Add New User 95
- Add New User Tag 98
- Add New Work Type 99
- Add Project Member User 32
- AddCaseMemberUser 32
- AddNewAccessRightsProfile 33
- AddNewAccount 35

AddNewAccountGroup	38	addressInfo_CompanyGUID	35, 46, 327
AddNewAccountGroupMembers	39	addressInfo_CountryCode	35, 46, 327
AddNewActivity	40	addressInfo_CountryGUID	35, 46, 327
AddNewActivityContactParticipant	42	addressInfo_CountryName	35, 46, 327
ADDNEWACTIVITYCONTACTPARTICIPANTRESULT	42	addressInfo_CountryRegionGUID	35, 46, 327
AddNewActivityResourceParticipant	43	addressInfo_CountryRegionName	35, 46, 327
ADDNEWACTIVITYRESOURCEPARTICIPANTRESULT	43	addressInfo_Fax	35, 46, 327
AddNewActivityType	44	addressInfo_GUID	35, 46, 327
AddNewActivityUserParticipant	45	addressInfo_IsBillingAddress	35, 46, 327
ADDNEWACTIVITYUSERPARTICIPANTRESULT	45	addressInfo_IsPostalAddress	35, 46, 327
AddNewAddress	46	addressInfo_IsVisitAddress	35, 46, 327
AddNewBillingAndExpenseForecastToCase	47	addressInfo_Phone	35, 46, 327
AddNewBusinessUnit	49	addressInfo_PostalCode	35, 46, 327
AddNewCase	51	ADDRESSLINE	46, 95, 100, 101, 103, 118, 177, 178, 327, 356, 361, 363, 365, 367, 369, 371, 372, 374
AddNewCaseStatus	56	Addressline 2	46, 100, 101, 103, 177, 178, 327
AddNewCaseTag	57	Addressline 3	46, 100, 101, 103, 177, 178, 327
AddNewContact	58	ADDRESSLINE2	46, 100, 101, 103, 177, 178, 327
AddNewContactTag	59	ADDRESSLINE3	46, 100, 101, 103, 177, 178, 327
AddNewCustomer	60	Advance Payment	312, 313, 315, 316, 318
AddNewEmployment	63	ADVANCEPAYMENT	312, 313, 315, 316, 318
AddNewExpenseToProductByPoNumber	64	Allocated Hours	89, 91, 283, 284, 285, 286, 287, 288, 353
AddNewFileTag	69	Allocated Percentage	89, 91, 283, 284, 285, 286, 287, 288, 353
AddNewHourEntry	70	ALLOCATEDHOURS	89, 91, 283, 284, 285, 286, 287, 288, 353
AddNewIndustry	74	ALLOCATEDPERCENTAGE	89, 91, 283, 284, 285, 286, 287, 288, 353
AddNewInvoiceLogEntry	75	analysis-enforce-row-uniqueness	2
AddNewItem	76	Annual Revenue	60, 129, 148, 149, 151, 339
AddNewItems	81	ANNUALREVENUE	60, 129, 148, 149, 151, 339
AddNewPhase	84	ANNUALREVENUE2	60, 129, 148, 149, 151, 339
AddNewProduct	86	API	359, 366
AddNewProductCategory	87	api-url	2
AddNewResource	88	Approval Date	312, 313, 315, 316, 318
AddNewResourceAllocation	89	APPROVALDATE	312, 313, 315, 316, 318
AddNewResourceAllocations	91	Approved by User GUID	312, 313, 315, 316, 318
AddNewSalesProcess	92	APPROVEDBYUSERGUID	312, 313, 315, 316, 318
AddNewSalesStatus	93	Are Other Travel Expenses Billable	51, 110, 112, 115, 119, 122, 125, 332, 384
AddNewTravelExpense	94	attachmentGuid	195
AddNewUser	95		
AddNewUserTag	98		
AddNewWorkType	99		
Address by GUID	100		
Address GUID	58, 130, 131, 133, 134, 338		
AddressByGUID	100		
Addresses Changed since	101		
AddressesChangedSince	101		
ADDRESSGUID	58, 100, 130, 131, 133, 134, 338	Bank Account Number	95, 118, 356, 361, 363, 365, 367, 369, 371, 372, 374
addressInfo_Addressline	35, 46, 327	BANKACCOUNTNUMBER	95, 118, 356, 361, 363, 365, 367, 369, 371, 372, 374
addressInfo_Addressline2	35, 46, 327	Billing Address	51, 103, 110, 112, 115, 119, 122, 125, 332, 384
addressInfo_Addressline3	35, 46, 327		
addressInfo_City	35, 46, 327		

- B -

Billing Address GUID 51, 110, 112, 115, 119, 122, 125, 332, 384
 Billing and Expense Forecast by Project GUID 103, 356, 361, 363, 365, 367, 369, 371, 372, 374, 384, 387
 Billing Dependency Task GUID 64, 76, 81, 233, 236, 239, 241, 244, 274, 277, 307
 Billing Forecast 47, 103, 328
 Billing Internal Notes 51, 110, 112, 115, 119, 122, 125, 332, 384
 Billing Notes before Grid 51, 110, 112, 115, 119, 122, 125, 332, 384
 Billing Schedule 64, 76, 81, 233, 236, 239, 241, 244, 274, 277, 307
 BILLINGADDRESS 51, 103, 110, 112, 115, 119, 122, 125, 332
 BILLINGADDRESSGUID 51, 110, 112, 115, 119, 122, 125, 332
 BillingAndExpenseForecastByCaseGUID 103
 billingAndExpenseForecastInfo_BillingForecast 47, 328
 billingAndExpenseForecastInfo_ExpenseForecast 47, 328
 billingAndExpenseForecastInfo_GUID 47, 328
 billingAndExpenseForecastInfo_InhouseRevenueForecast 47, 328
 billingAndExpenseForecastInfo_Month 47, 328
 billingAndExpenseForecastInfo_Notes 47, 328
 billingAndExpenseForecastInfo_OutsourcingRevenueForecast 47, 328
 billingAndExpenseForecastInfo_Year 47, 328
 BILLINGDEPENDENCYTASKGUID 64, 76, 81, 233, 236, 239, 241, 244, 274, 277, 307
 BILLINGFORECAST 47, 103, 328
 BILLINGINTERNALNOTES 51, 110, 112, 115, 119, 122, 125, 332
 BILLINGNOTESBEFOREGRID 51, 110, 112, 115, 119, 122, 125, 332
 BILLINGSCHEDULE 64, 76, 81, 233, 236, 239, 241, 244, 274, 277, 307
 BLOB Preferred 382
 BLOB_PREFERRED 382
 BOL_RESPONSE_CACHE_MAX_AGE_SEC 382
 bulk-delete-page-size-rows 2
 bulk-insert-page-size-bytes 2
 bulk-insert-page-size-rows 2
 Business Unit by Code 104
 Business Unit by GUID 106
 Business Unit Code 95, 118, 139, 196, 202, 212, 217, 222, 228, 356, 361, 363, 365, 367, 369, 371, 372, 374, 387
 Business Unit GUID 51, 95, 110, 112, 115, 118, 119, 122, 125, 139, 196, 202, 212, 217, 222, 228, 332, 356, 361, 363, 365, 367, 369, 371, 372, 374, 384, 387
 Business Unit Hierarchy 107
 Business Unit Name 51, 95, 110, 112, 115, 118, 119, 122, 125, 139, 196, 202, 212, 217, 222, 228, 332, 356, 361, 363, 365, 367, 369, 371, 372, 374, 384, 387
 Business Unit Number 51, 110, 112, 115, 119, 122, 125, 332, 384
 Business Units Changed since 108
 BusinessUnitByCode 104
 BusinessUnitByGUID 106
 BUSINESSUNITCODE 95, 104, 118, 139, 196, 202, 212, 217, 222, 228, 356, 361, 363, 365, 367, 369, 371, 372, 374, 387
 BUSINESSUNITGUID 22, 51, 95, 106, 108, 110, 112, 115, 118, 119, 122, 125, 139, 175, 176, 181, 184, 186, 196, 202, 212, 217, 222, 228, 256, 267, 307, 316, 332, 356, 361, 363, 365, 367, 369, 371, 372, 374, 377, 387
 businessUnitGUIDs 284
 BusinessUnitHierarchy 107
 businessUnitInfo_Code 49, 330
 businessUnitInfo_CostCenterGUID 49, 330
 businessUnitInfo_CurrencyCode 49, 330
 businessUnitInfo_CurrencyGUID 49, 330
 businessUnitInfo_FormattingCulture 49, 330
 businessUnitInfo_GUID 49, 330
 businessUnitInfo_InsertTS 49, 330
 businessUnitInfo_IsActive 49, 330
 businessUnitInfo_LanguageCode 49, 330
 businessUnitInfo_LanguageGUID 49, 330
 businessUnitInfo_Name 49, 330
 businessUnitInfo_NextInvoiceNumber 49, 330
 businessUnitInfo_OverDueInterest 49, 330
 businessUnitInfo_ParentBusinessUnitGUID 49, 330
 businessUnitInfo_PaymentTerm 49, 330
 businessUnitInfo_ShowLogoOnInvoices 49, 330
 businessUnitInfo_ShowLogoOnPdfReports 49, 330
 businessUnitInfo_TimeZoneGUID 49, 330
 businessUnitInfo_UpdatedTS 49, 330
 businessUnitInfo_VatNumber 49, 330
 BUSINESSUNITNAME 51, 95, 110, 112, 115, 118, 119, 122, 125, 139, 196, 202, 212, 217, 222, 228, 332, 356, 361, 363, 365, 367, 369, 371, 372, 374, 387
 BUSINESSUNITNUMBER 51, 110, 112, 115, 119, 122, 125, 332
 BusinessUnitsChangedSince 108
 BUYERORGANIZATION_BUYERCONTACTPERSONNAME 139, 196, 202, 212, 217, 222, 228, 387
 BUYERORGANIZATION_BUYERELECTRONICINVOICEADDRESS 139, 196, 202, 212, 217, 222, 228, 387
 BUYERORGANIZATION_BUYERELECTRONICINVOICEOPERATOR 139, 196, 202, 212, 217, 222, 228, 387

BUYERORGANIZATION_BUYEREMAILADDRESSIDENTIFIER	139, 196, 202, 212, 217, 222, 228, 387	caseInfo_CaseCompanyCurrencyGUID	51, 332
BUYERORGANIZATION_BUYERORGANISATIONNAME	139, 196, 202, 212, 217, 222, 228, 387	caseInfo_CaseNumber	51, 332
BUYERORGANIZATION_BUYERORGANISATIONTAXCODE	139, 196, 202, 212, 217, 222, 228, 387	caseInfo_CaseOwnerUserCode	51, 332
BUYERORGANIZATION_BUYERPARTYIDENTIFIER	139, 196, 202, 212, 217, 222, 228, 387	caseInfo_CaseOwnerUserGUID	51, 332
BUYERORGANIZATION_BUYERPHONENUMBERIDENTIFIER	139, 196, 202, 212, 217, 222, 228, 387	caseInfo_ClosedDate	51, 332
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRY	139, 196, 202, 212, 217, 222, 228, 387	caseInfo_CompletionEstimate	51, 332
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_COUNTRYCODE	139, 196, 202, 212, 217, 222, 228, 387	caseInfo_ContactGUID	51, 332
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER	139, 196, 202, 212, 217, 222, 228, 387	caseInfo_CostCenterGUID	51, 332
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_STREETNAME	139, 196, 202, 212, 217, 222, 228, 387	caseInfo_CostCenterNumber	51, 332
BUYERORGANIZATION_BUYERPOSTALADDRESSDETAILS_TOWNNAME	139, 196, 202, 212, 217, 222, 228, 387	caseInfo_CurrencyGUID	51, 332
CASEBUSINESSUNITCODE	70, 179, 181, 184, 186, 189, 191, 207, 267, 270, 272, 343, 384	caseInfo_CurrencyShortform	51, 332
CaseByGUID	110	caseInfo_CurrencyShortformOfCaseCompanyCurrency	51, 332
CaseByNumber	112	caseInfo_CurrentCaseStatusDescription	51, 332
CaseByTaskGUID	115	caseInfo_CurrentCaseStatusGUID	51, 332
CASECOMPANYCURRENCYGUID	51, 110, 112, 115, 119, 122, 125, 332	caseInfo_CurrentCaseStatusInsertTs	51, 332
CASECOSTCENTERIDENTIFIER	70, 179, 181, 184, 186, 189, 191, 207, 267, 270, 272, 343, 384	caseInfo_DeadlineDate	51, 332
CASEGUID	22, 24, 26, 32, 40, 47, 57, 64, 70, 76, 81, 84, 89, 91, 93, 103, 110, 118, 156, 170, 171, 179, 181, 184, 186, 189, 191, 193, 207, 233, 236, 239, 244, 251, 253, 255, 256, 267, 270, 272, 274, 277, 283, 284, 285, 286, 287, 288, 292, 293, 299, 307, 324, 343, 348, 353, 384	caseInfo_Description	51, 332
caseGUIDs	284	caseInfo_ExpectedValue	51, 332
caseInfo_AccountGUID	51, 332	caseInfo_GUID	51, 332
caseInfo_AccountName	51, 332	caseInfo_IncludeExceptions	51, 332
caseInfo_AccountNumber	51, 332	caseInfo_IncludeOverTime	51, 332
caseInfo_BillingAddress	51, 332	caseInfo_InsertTs	51, 332
caseInfo_BillingAddressGUID	51, 332	caseInfo_InternalName	51, 332
caseInfo_BillingInternalNotes	51, 332	caseInfo_InvoiceTemplateGUID	51, 332
caseInfo_BillingNotesBeforeGrid	51, 332	caseInfo_IsClosed	51, 332
caseInfo_BusinessUnitGUID	51, 332	caseInfo_IsDailyAllowanceBillable	51, 332
caseInfo_BusinessUnitName	51, 332	caseInfo_IsInternal	51, 332
caseInfo_BusinessUnitNumber	51, 332	caseInfo_IsMileageBillable	51, 332
		caseInfo_IsOtherTravelExpensesBillable	51, 332
		caseInfo_LatestEstimationDate	51, 332
		caseInfo_LeadRating	51, 332
		caseInfo_LeadSourceGUID	51, 332
		caseInfo_Name	51, 332
		caseInfo_OrderNumber	51, 332
		caseInfo_OurReference	51, 332
		caseInfo_OverdueInterest	51, 332
		caseInfo_PaymentTerm	51, 332
		caseInfo_PricelistGUID	51, 332
		caseInfo_PricingRule	51, 332
		caseInfo_PricingType	51, 332
		caseInfo_Priority	51, 332
		caseInfo_Probability	51, 332
		caseInfo_RootTaskGUID	51, 332
		caseInfo_SalesCloseDate	51, 332
		caseInfo_SalesPersonUserCode	51, 332
		caseInfo_SalesPersonUserGUID	51, 332
		caseInfo_SalesProcessGUID	51, 332
		caseInfo_SalesStatusGUID	51, 332
		caseInfo_SharedToExtranet	51, 332
		caseInfo_StartDate	51, 332

- C -

caseInfo_UpdatedTs 51, 332
 caseInfo_UrlToCase 51, 332
 caseInfo_UseDefaultProductsInWorkTimeEntry 51, 332
 caseInfo_UseDefaultWorkTypes 51, 332
 caseInfo_YourReference 51, 332
 CaseMemberUsersByCaseGUID 118
 CASENAME 70, 179, 181, 184, 186, 189, 191, 207, 267, 270, 272, 343, 384
 CASENUMBER 51, 64, 70, 76, 81, 110, 112, 115, 119, 122, 125, 179, 181, 184, 186, 189, 191, 207, 235, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 332, 343, 384
 CASEOWNERUSERCODE 51, 110, 112, 115, 119, 122, 125, 332
 CASEOWNERUSERGUID 51, 110, 112, 115, 119, 122, 125, 332
 CASERIGHTS_EDITCASESTATUS 171
 CASERIGHTS_SHOWBILLING 171
 CASERIGHTS_SHOWCASESTATUS 171
 CASERIGHTS_SHOWFILESANDLINKS 171
 CASERIGHTS_SHOWGANTTCHART 171
 CASERIGHTS_SHOWPHASETREE 171
 CASERIGHTS_SHOWREADYTOBILL 171
 CASERIGHTS_SHOWWORKHOURESTIMATES 171
 CASERIGHTS_SHOWWORKHOURS 171
 CASERIGHTS_SHOWWORKHOURSTARTANDENDTIME 171
 CASES 119, 209
 CasesByAccountGUID 122
 CasesChangedSince 125
 CaseStatusByGUID 127
 CaseStatuses 128
 caseStatusGUID 127, 156
 caseStatusGUIDs 284
 caseStatusInfo_GUID 56, 337
 caseStatusInfo_Icon 56, 337
 caseStatusInfo_IsActive 56, 337
 caseStatusInfo_Name 56, 337
 caseTagGUIDs 284
 CaseTags 129
 CATEGORY 30, 31, 44, 326
 Child Phases 193
 CHILDPHASES 193
 CITY 46, 100, 101, 103, 177, 178, 327
 Closed Date 51, 110, 112, 115, 119, 122, 125, 332, 384
 CLOSEDDATE 51, 110, 112, 115, 119, 122, 125, 332
 CODE 30, 31, 44, 49, 74, 84, 86, 87, 94, 95, 99, 104, 106, 107, 108, 118, 194, 250, 251, 253, 255, 256, 259, 260, 261, 262, 263, 264, 266, 304, 305, 306, 310, 326, 330, 348, 350, 351, 354, 356, 360, 361, 363, 365, 367, 369, 371, 372, 374, 378, 379, 380, 381
 COMMUNICATIONMETHODS 58, 130, 131, 133, 134, 338
 Company Business Unit Code 63, 167, 168, 342
 Company Business Unit GUID 63, 167, 168, 342
 Company Business Unit Name 63, 167, 168, 342
 Company by GUID 129
 Company GUID 8, 9, 11, 12, 16, 18, 19, 21, 35, 46, 60, 100, 101, 103, 148, 149, 151, 177, 178, 321, 327, 339
 COMPANYBUSINESSUNITCODE 63, 167, 168, 342
 COMPANYBUSINESSUNITGUID 63, 167, 168, 342
 COMPANYBUSINESSUNITNAME 63, 167, 168, 342
 CompanyByGUID 129
 COMPANYGUID 8, 9, 11, 12, 16, 18, 19, 21, 35, 46, 60, 100, 101, 103, 129, 148, 149, 151, 177, 178, 321, 327, 339
 companyInfo_AnnualRevenue 35, 321
 companyInfo_AnnualRevenue2 35, 321
 companyInfo_Email 35, 321
 companyInfo_Employees 35, 321
 companyInfo_GUID 35, 321
 companyInfo_HeadOfficeAddressGUID 35, 321
 companyInfo_HierarchyGUID 35, 321
 companyInfo_IndustryGUID 35, 321
 companyInfo_LanguageGUID 35, 321
 companyInfo_Name 35, 321
 companyInfo_TimezoneGUID 35, 321
 companyInfo_VatNumber 35, 321
 companyInfo_Website 35, 321
 Completion Estimate 51, 110, 112, 115, 119, 122, 125, 332, 384
 COMPLETIONESTIMATE 51, 110, 112, 115, 119, 122, 125, 332
 Contact by GUID 130
 Contact GUID 25, 28, 29, 31, 51, 110, 112, 115, 119, 122, 125, 332, 384
 Contact Role GUID 58, 130, 131, 133, 134, 338
 ContactByGUID 130
 CONTACTGUID 25, 28, 29, 31, 42, 51, 59, 110, 112, 115, 119, 122, 125, 130, 157, 171, 280, 332
 contactInfo_AccountGUID 35, 58, 338
 contactInfo_AddressGUID 35, 58, 338
 contactInfo_CommunicationMethods 35, 58, 338
 contactInfo_ContactRoleGUID 35, 58, 338
 contactInfo_DateOfBirth 35, 58, 338
 contactInfo_Description 35, 58, 338

contactInfo_FirstName	35, 58, 338	COSTCENTERGUID	49, 51, 64, 76, 81, 104, 106, 107, 108, 110, 112, 115, 119, 122, 125, 135, 233, 236, 239, 241, 244, 274, 277, 307, 330, 332
contactInfo_GUID	35, 58, 338	COSTCENTERIDENTIFIER	64, 76, 81, 233, 236, 239, 241, 244, 274, 277, 307
contactInfo_IsActive	35, 58, 338	COSTCENTERNUMBER	51, 110, 112, 115, 119, 122, 125, 332
contactInfo_JobTitle	35, 58, 338	CostCenters	136
contactInfo_Keywords	35, 58, 338	COSTCURRENCYGUID	94, 304, 305, 306, 310, 354
contactInfo_LanguageGUID	35, 58, 338	COSTCURRENCYSHORTFORM	64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 343, 384
contactInfo_LastName	35, 58, 338	COSTCURRENCYSHORTFORMOFINVOICINGCURRENCY	64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 343, 384
contactInfo_Salutation	35, 58, 338	COSTCURRENCYSHORTFORMOFORGANIZATIONCURRENCY	64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 343, 384
contactInfo_SatisfactionLevel	35, 58, 338	COSTCURRENCYSHORTFORMOFUSERCOMPANYSHORTFORMOFORGANIZATIONCURRENCY	64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 343, 384
contactInfo_TimeZoneGUID	35, 58, 338	count	15, 149
CONTACTRIGHTS_APPROVAL	171	Country by GUID	136
CONTACTRIGHTS_COMMENTS	171	Country by ISO	137
CONTACTRIGHTS_FILEUPLOAD	171	Country by Name	138
CONTACTROLEGUID	58, 130, 131, 133, 134, 338	Country Code	46, 95, 100, 101, 103, 118, 177, 178, 327, 356, 361, 363, 365, 367, 369, 371, 372, 374
CONTACTS	170	Country GUID	46, 95, 100, 101, 103, 118, 177, 178, 327, 356, 361, 363, 365, 367, 369, 371, 372, 374
Contacts by Account GUID	131	Country Name	46, 100, 101, 103, 177, 178, 327
Contacts by Communication Method Type	133	Country Region GUID	46, 95, 100, 101, 103, 118, 177, 178, 327, 356, 361, 363, 365, 367, 369, 371, 372, 374
Contacts Changed since	134	Country Region Name	46, 95, 100, 101, 103, 118, 177, 178, 327, 356, 361, 363, 365, 367, 369, 371, 372, 374
ContactsByAccountGUID	131	CountryByGUID	136
ContactsByCommunicationMethodType	133	CountryByISO	137
ContactsChangedSince	134	CountryByName	138
Content Type	173, 174, 311, 382	COUNTRYCODE	46, 95, 100, 101, 103, 118, 177, 178, 327, 356, 361, 363, 365, 367, 369, 371, 372, 374
CONTENT_TYPE	382	COUNTRYGUID	46, 95, 100, 101, 103, 118, 136, 177, 178, 327, 356, 361, 363, 365, 367, 369, 371, 372, 374
CONTENTTYPE	173, 174, 311	COUNTRYNAME	46, 100, 101, 103, 138, 177, 178, 327
CONTEXT	57, 59, 69, 75, 98, 129, 298, 299, 376	COUNTRYREGIONGUID	46, 95, 100, 101, 103, 118, 177, 178, 327, 356, 361, 363, 365, 367, 369, 371, 372, 374
Cost Center by GUID	135		
Cost Center GUID	49, 51, 64, 76, 81, 104, 106, 107, 108, 110, 112, 115, 119, 122, 125, 233, 236, 239, 241, 244, 274, 277, 307, 330, 332, 384		
Cost Center Identifier	64, 76, 81, 233, 236, 239, 241, 244, 274, 277, 307		
Cost Center Number	51, 110, 112, 115, 119, 122, 125, 332, 384		
Cost Centers	136		
Cost Currency GUID	94, 304, 305, 306, 310, 354		
Cost Currency Short Form	64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 343, 384		
Cost Currency Short Form of Invoicing Currency	64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 343, 384		
Cost Currency Short Form of Organization Currency	64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 343, 384		
Cost Currency Short Form of User Company Currency	64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 343, 384		
CostCenterByGUID	135		

COUNTRYREGIONNAME 46, 95, 100, 101, 103, 371, 372, 374
 118, 177, 178, 327, 356, 361, 363, 365, 367, 369, 371, 372, 374
 Create Invoices 139
 CreateInvoices 139
 criteria_AccountGroupGuids 149
 criteria_InternalName 119
 criteria_OnlyActiveCases 119
 criteria_OnlyCustomerCases 119
 criteria_OnlyInactiveCases 119
 criteria_OnlyInternalCases 119
 criteria_SalesProcess 119
 criteria_Tags 119
 cse_ACCOUNTGUID 384
 cse_ACCOUNTNAME 384
 cse_ACCOUNTNUMBER 384
 cse_BILLINGADDRESS 384
 cse_BILLINGADDRESSGUID 384
 cse_BILLINGINTERNALNOTES 384
 cse_BILLINGNOTESBEFOREGRID 384
 cse_BUSINESSUNITGUID 384
 cse_BUSINESSUNITNAME 384
 cse_BUSINESSUNITNUMBER 384
 cse_CASECOMPANYCURRENCYGUID 384
 cse_CASENUMBER 384
 cse_CASEOWNERUSERCODE 384
 cse_CASEOWNERUSERGUID 384
 cse_CLOSEDDATE 384
 cse_COMPLETIONESTIMATE 384
 cse_CONTACTGUID 384
 cse_COSTCENTERGUID 384
 cse_COSTCENTERNUMBER 384
 cse_CURRENCYGUID 384
 cse_CURRENCYSHORTFORM 384
 cse_CURRENCYSHORTFORMOFCASECOMPANYCURRENCY 384
 cse_CURRENTCASESTATUSDESCRIPTION 384
 cse_CURRENTCASESTATUSGUID 384
 cse_CURRENTCASESTATUSINSERTTS 384
 cse_DEADLINEDATE 384
 cse_DESCRIPTION 384
 cse_EXPECTEDVALUE 384
 cse_GUID 384
 cse_INCLUDEEXCEPTIONS 384
 cse_INCLUDEOVERTIME 384
 cse_INSERTTS 384
 cse_INTERNALNAME 384
 cse_INVOICETEMPLATEGUID 384
 cse_ISCLOSED 384
 cse_ISDAILYALLOWANCEBILLABLE 384
 cse_ISINTERNAL 384
 cse_ISMILEAGEBILLABLE 384
 cse_ISOTHERTRAVELEXPENSESBILLABLE 384
 cse_LATESTESTIMATIONDATE 384
 cse_LEADRATING 384
 cse_LEADSOURCEGUID 384
 cse_NAME 384
 cse_ORDERNUMBER 384
 cse_OURREFERENCE 384
 cse_OVERDUEINTEREST 384
 cse_PAYMENTTERM 384
 cse_PRICELISTGUID 384
 cse_PRICINGRULE 384
 cse_PRICINGTYPE 384
 cse_PRIORITY 384
 cse_PROBABILITY 384
 cse_ROOTTASKGUID 384
 cse_SALESCLOSEDATE 384
 cse_SALESPERSONUSERCODE 384
 cse_SALESPERSONUSERGUID 384
 cse_SALESPROCESSGUID 384
 cse_SALESSTATUSGUID 384
 cse_SHAREDTOEXTRANET 384
 cse_STARTDATE 384
 cse_UPDATEDTS 384
 cse_URLTOCASE 384
 cse_USEDEFAULTPRODUCTSINWORKTIMEENTRY 384
 cse_USEDEFAULTWORKTYPES 384
 cse_YOURREFERENCE 384
 CULTURE 95, 118, 139, 196, 202, 212, 217, 222, 228, 356, 361, 363, 365, 367, 369, 371, 372, 374, 387
 Currencies 146
 Currency by GUID 146
 Currency by ISO Code 147
 Currency Code 8, 9, 11, 12, 16, 18, 19, 21, 35, 49, 51, 60, 104, 106, 107, 108, 148, 149, 151, 257, 258, 321, 330, 339
 Currency GUID 8, 9, 11, 12, 16, 18, 19, 21, 35, 49, 51, 60, 104, 106, 107, 108, 110, 112, 115, 119, 122, 125, 139, 148, 149, 151, 196, 202, 212, 217, 222, 228, 257, 258, 312, 313, 315, 316, 318, 321, 330, 332, 339, 384, 387
 Currency Name 139, 196, 202, 212, 217, 222, 228, 387
 Currency Rate 139, 196, 202, 212, 217, 222, 228, 387
 Currency Short Form 51, 110, 112, 115, 119, 122, 125, 139, 196, 202, 212, 217, 222, 228, 312, 313, 315, 316, 318, 332, 384, 387
 Currency Short Form of Project Company Currency 51, 110, 112, 115, 119, 122, 125, 332, 384
 CurrencyByGUID 146
 CurrencyByIsoCode 147

CURRENCYCODE	8, 9, 11, 12, 16, 18, 19, 21, 35, 49, 60, 76, 104, 106, 107, 108, 148, 149, 151, 257, 258, 321, 330, 339	customer_Notes	60, 339
CURRENCYGUID	8, 9, 11, 12, 16, 18, 19, 21, 35, 49, 51, 60, 104, 106, 107, 108, 110, 112, 115, 119, 122, 125, 139, 146, 148, 149, 151, 196, 202, 212, 217, 222, 228, 257, 258, 312, 313, 315, 316, 318, 321, 330, 332, 339, 387	customer_Number	60, 339
CURRENCYNAME	139, 196, 202, 212, 217, 222, 228, 387	customer_OverdueInterest	60, 339
CURRENCYRATE	139, 196, 202, 212, 217, 222, 228, 387	customer_PaymentTerm	60, 339
CURRENCYSHORTFORM	51, 110, 112, 115, 119, 122, 125, 139, 196, 202, 212, 217, 222, 228, 312, 313, 315, 316, 318, 332, 387	customer_PricelistGUID	60, 339
CURRENCYSHORTFORMOFCASECOMPANYCURRENCY	51, 110, 112, 115, 119, 122, 125, 332	customer_ReverseCharge	60, 339
Current Project Status Description	51, 110, 112, 115, 119, 122, 125, 332, 384	customer_ReverseChargeDescription	60, 339
Current Project Status GUID	51, 110, 112, 115, 119, 122, 125, 332, 384	customer_TimezoneGUID	60, 339
Current Project Status Inserted	51, 110, 112, 115, 119, 122, 125, 332, 384	customer_UpdatedTS	60, 339
CURRENTCASESTATUSDESCRIPTION	51, 110, 112, 115, 119, 122, 125, 332	customer_UriToAccount	60, 339
CURRENTCASESTATUSGUID	51, 110, 112, 115, 119, 122, 125, 332	customer_VatNumber	60, 339
CURRENTCASESTATUSINSERTTS	51, 110, 112, 115, 119, 122, 125, 332	customer_Website	60, 339
Customer by GUID	148	CustomerByGUID	148
customer_AccountGroupGUIDs	60, 339	Customers	149
customer_AccountOwnerUserGUID	60, 339	Customers Changed since	151
customer_AccountRating	60, 339	CustomersChangedSince	151
customer_AnnualRevenue	60, 339	- D -	
customer_AnnualRevenue2	60, 339	Daily Hours	63, 167, 168, 342
customer_CompanyGUID	60, 339	DAILYHOURS	63, 167, 168, 342
customer_CurrencyCode	60, 339	Database	153
customer_CurrencyGUID	60, 339	Database Driver	1
customer_eInvoiceAddress	60, 339	DATABASERESULT	153
customer_eInvoiceOperator	60, 339	DATE	139, 196, 202, 212, 217, 222, 228, 312, 313, 315, 316, 318, 377, 387
customer_Email	60, 339	Date of Birth	58, 130, 131, 133, 134, 338
customer_Employees	60, 339	DATE_ENDED	382
customer_GUID	60, 339	DATE_STARTED	382
customer_HeadOfficeAddressGUID	60, 339	DATEOFBIRTH	58, 130, 131, 133, 134, 338
customer_IndustryGUID	60, 339	Ddescription	51, 58, 64, 70, 76, 81, 84, 86, 94, 110, 112, 115, 119, 122, 125, 130, 131, 133, 134, 139, 173, 174, 179, 181, 184, 186, 189, 191, 196, 202, 207, 212, 217, 222, 228, 233, 236, 239, 241, 244, 251, 253, 255, 256, 259, 260, 261, 266, 267, 270, 272, 274, 277, 300, 301, 302, 304, 305, 306, 307, 310, 311, 332, 338, 343, 348, 350, 354, 384, 387
customer_InsertTS	60, 339	DEADLINE	84, 251, 253, 255, 256, 348
customer_InvoicingVat	60, 339	Deadline Date	51, 110, 112, 115, 119, 122, 125, 332, 384
customer_IsActive	60, 339	DEADLINEDATE	51, 110, 112, 115, 119, 122, 125, 332
customer_IsInternal	60, 339	Default Activity Type GUID	95, 118, 356, 361, 363, 365, 367, 369, 371, 372, 374
customer_LanguageCode	60, 339	Default Language Code	136, 137, 138
customer_LanguageGUID	60, 339	Default Language GUID	136, 137, 138
customer_Name	60, 339	Default Probability	92, 290, 291
		DEFAULTACTIVITYTYPEGUID	95, 118, 356, 361, 363, 365, 367, 369, 371, 372, 374
		DEFAULTLANGUAGECODE	136, 137, 138
		DEFAULTLANGUAGEGUID	136, 137, 138
		DEFAULTPROBABILITY	92, 290, 291

Delete Access Rights Profile by GUID 153
 Delete Activity 154
 Delete Activity Type 155
 Delete Contact 157
 Delete Employment 158
 Delete Hour Entry 159
 Delete Item 160
 Delete Phase 160
 Delete Product 161
 Delete Product Category 162
 Delete Project 156
 Delete Project Status 156
 Delete Resource 163
 Delete Resource Allocation 164
 Delete Tag 165
 Delete Travel Expense 165
 DeleteAccessRightsProfileByGUID 153
 DeleteActivity 154
 DELETEACTIVITYRESULT 154
 DeleteActivityType 155
 DELETEACTIVITYTYPESRESULT 155
 DeleteCase 156
 DELETEDECASERESULT 156
 DeleteCaseStatus 156
 DELETEDECASESTATUSRESULT 156
 DeleteContact 157
 DELETEDCONTACTRESULT 157
 DeleteEmployment 158
 DELETEDEMPLOYMENTRESULT 158
 DeleteHourEntry 159
 DELETEDHOURENTRYRESULT 159
 DeleteItem 160
 DELETEDITEMRESULT 160
 DeletePhase 160
 DELETEDPHASERESULT 160
 DeleteProduct 161
 DeleteProductCategory 162
 DELETEDPRODUCTCATEGORYRESULT 162
 DELETEDPRODUCTRESULT 161
 DeleteResource 163
 DeleteResourceAllocation 164
 DELETEDRESOURCEALLOCATIONRESULT 164
 DELETEDRESOURCERESULT 163
 DeleteTag 165
 DELETEDTAGRESULT 165
 DeleteTravelExpense 165
 DELETEDTRAVELEXPENSERESULT 165
 DESCRIPTION 51, 58, 64, 70, 76, 81, 84, 86, 94,
 110, 112, 115, 119, 122, 125, 130, 131, 133, 134, 139,
 173, 174, 179, 181, 184, 186, 189, 191, 196, 202, 207,
 212, 217, 222, 228, 233, 236, 239, 241, 244, 251, 253,
 255, 256, 259, 260, 261, 266, 267, 270, 272, 274, 277,
 300, 301, 302, 304, 305, 306, 307, 310, 311, 332, 338,
 343, 348, 350, 354, 384, 387
 Difference to GMT 303, 304
 DIFFERENCETOGMT 303, 304
 Disable User 166
 DisableUser 166
 DISABLEUSERRESULT 166
 Display Period Start 64, 76, 81, 233, 236, 239, 241,
 244, 274, 277, 307
 DISPLAYPERIODSTART 64, 76, 81, 233, 236, 239,
 241, 244, 274, 277, 307
 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
 DRY_RUN 382
 Due Date 139, 196, 202, 212, 217, 222, 228, 387
 DUEDATE 139, 196, 202, 212, 217, 222, 228, 387
 DUMMY 32, 39, 75, 153, 280, 294, 295, 296, 297
 Duration (ms) 382
 DURATION_MS 382

- E -

 e-invoice Address 8, 9, 11, 12, 16, 18, 19, 21, 35,
 60, 148, 149, 151, 321, 339
 e-invoice Operator 8, 9, 11, 12, 16, 18, 19, 21, 35,
 60, 148, 149, 151, 321, 339
 EINVOICEADDRESS 8, 9, 11, 12, 16, 18, 19, 21,
 35, 60, 148, 149, 151, 321, 339
 EINVOICEOPERATOR 8, 9, 11, 12, 16, 18, 19, 21,
 35, 60, 148, 149, 151, 321, 339
 EMAIL 60, 95, 118, 129, 148, 149, 151, 339, 356,
 361, 363, 365, 367, 369, 371, 372, 374
 EMPLOYEES 60, 129, 148, 149, 151, 339
 Employment by GUID 167
 EmploymentByGUID 167
 employmentGUID 158, 167
 employmentInfo_CompanyBusinessUnitCode 63,
 342
 employmentInfo_CompanyBusinessUnitGUID 63,
 342
 employmentInfo_CompanyBusinessUnitName 63,
 342
 employmentInfo_DailyHours 63, 342
 employmentInfo_EndDate 63, 342
 employmentInfo_GUID 63, 342
 employmentInfo_HourCost 63, 342

employmentInfo_IsOvertimeAllowed 63, 342
 employmentInfo_StartDate 63, 342
 employmentInfo_Title 63, 342
 Employments by User GUID 168
 EmploymentsByUserGUID 168
 Enable User 169
 EnableUser 169
 ENABLEUSERRESULT 169
 End Date 63, 89, 91, 167, 168, 283, 284, 285, 286, 287, 288, 342, 353, 382
 End Date Time 22, 24, 26, 40, 324
 End Time 70, 179, 181, 184, 186, 189, 191, 207, 267, 270, 272, 300, 301, 302, 343, 384
 enddate 22, 63, 89, 91, 167, 168, 179, 181, 184, 186, 212, 217, 267, 283, 284, 285, 286, 287, 288, 307, 315, 342, 353, 377
 ENDDATETIME 22, 24, 26, 40, 324
 endsAfter 26
 endsBefore 26
 ENDDTIME 70, 179, 181, 184, 186, 189, 191, 207, 267, 270, 272, 300, 301, 302, 343, 384
 Entity Schema Version 170
 EntitySchemaVersion 170
 ENTRIES 377
 Entry Date 139, 196, 202, 212, 217, 222, 228, 387
 ENTRYDATE 139, 196, 202, 212, 217, 222, 228, 387
 Error Message Code 382
 Error Message Text 382
 ERROR_MESSAGE_CODE 382
 ERROR_MESSAGE_TEXT 382
 Event Date 64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 300, 301, 302, 307, 343, 384
 EVENTDATE 64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 300, 301, 302, 307, 343, 384
 eventTimeUtc 75
 Expected Hours 377
 Expected Value 51, 110, 112, 115, 119, 122, 125, 332, 384
 EXPECTEDHOURS 377
 EXPECTEDVALUE 51, 110, 112, 115, 119, 122, 125, 332
 Expense Class 94, 304, 305, 306, 310, 354
 Expense Forecast 47, 103, 328
 EXPENSECLASS 94, 304, 305, 306, 310, 354
 EXPENSEFORECAST 47, 103, 328
 Extranet Contact Project Rights by Contact GUID 171
 Extranet Project by Project GUID 170
 ExtranetCaseByCaseGUID 170
 ExtranetContactCaseRightsByContactGUID 171

- F -

Fail on Error 382
 FAIL_ON_ERROR 382
 FAX 46, 100, 101, 103, 177, 178, 327
 File by GUID 173
 File Contents by GUID 173
 FileByGUID 173
 FileBytesByGUID 173
 fileGUID 69
 Files 174
 Financial Invoice 175, 176
 Financial Invoices by Status 175
 Financial Invoices by Status and Date 176
 FINVOICE 175, 176
 FinvoicesByStatus 175
 FinvoicesByStatusAndDate 176
 finvoiceVersion 176
 First Account Billing Addresses 177
 First Name 58, 95, 118, 130, 131, 133, 134, 338, 356, 361, 363, 365, 367, 369, 371, 372, 374
 FirstAccountsBillingAddress 177
 FIRSTNAME 58, 95, 118, 130, 131, 133, 134, 338, 356, 361, 363, 365, 367, 369, 371, 372, 374
 firstRow 26
 Flat Rate Additional Hours Unit Price 84, 251, 253, 255, 256, 348
 Flat Rate Are additional Hours Billable 84, 251, 253, 255, 256, 348
 Flat Rate Hours 84, 251, 253, 255, 256, 348
 Flat Rate Price 84, 251, 253, 255, 256, 348
 FLATRATE_ADDITIONALHOURSUNITPRICE 84, 251, 253, 255, 256, 348
 FLATRATE_AREADDITIONALHOURSBILLABLE 84, 251, 253, 255, 256, 348
 FLATRATE_HOURS 84, 251, 253, 255, 256, 348
 FLATRATE_PRICE 84, 251, 253, 255, 256, 348
 force-case-sensitive-identifiers 2
 forced-casing-identifiers 2
 Formatting Culture 49, 104, 106, 107, 108, 330
 FORMATTINGCULTURE 49, 104, 106, 107, 108, 330

- G -

getAttachmentOnly 210, 211
 GETENTITYSCHEMAVERSIONRESULT 170
 GETFILEBYTESBYGUIDRESULT 173
 GETINVOICEATTACHMENTFILERESULT 195
 GETINVOICEPDFBYINVOICEGUIDRESULT 211

GETINVOICEPDFRESULT	210	hourEntryInfo_CaseBusinessUnitCode	70, 343
GETTRAVELREIMBURSEMENTPDFRESULT	314	hourEntryInfo_CaseCostCenterIdentifier	70, 343
GETVERSIONRESULT	376	hourEntryInfo_CaseGUID	70, 343
GUID	6, 7, 8, 9, 11, 12, 14, 15, 16, 18, 19, 21, 22, 24, 26, 30, 31, 33, 35, 38, 40, 44, 46, 47, 49, 51, 56, 57, 58, 59, 60, 63, 64, 69, 70, 74, 76, 81, 84, 86, 87, 88, 89, 91, 92, 93, 94, 95, 98, 99, 100, 101, 103, 104, 106, 107, 108, 110, 112, 115, 118, 119, 122, 125, 127, 128, 129, 130, 131, 133, 134, 135, 136, 137, 138, 139, 146, 147, 148, 149, 151, 153, 167, 168, 170, 173, 174, 175, 176, 177, 178, 179, 181, 184, 186, 189, 191, 193, 194, 195, 196, 202, 207, 212, 217, 222, 228, 233, 236, 239, 241, 244, 247, 248, 249, 250, 251, 253, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 270, 272, 274, 277, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 310, 311, 312, 313, 314, 315, 316, 318, 319, 321, 324, 326, 327, 328, 330, 332, 337, 338, 339, 342, 343, 348, 350, 351, 352, 353, 354, 356, 359, 360, 361, 363, 365, 366, 367, 369, 371, 372, 374, 376, 378, 379, 380, 381, 384, 387	hourEntryInfo_CaseName	70, 343
		hourEntryInfo_CaseNumber	70, 343
		hourEntryInfo_CostCurrencyShortform	70, 343
		hourEntryInfo_CostCurrencyShortformOfInvoicingCurrency	70, 343
		hourEntryInfo_CostCurrencyShortformOfOrganizationCurrency	70, 343
		hourEntryInfo_CostCurrencyShortformOfUserCompanyCurrency	70, 343
		hourEntryInfo_Description	70, 343
		hourEntryInfo_EndTime	70, 343
		hourEntryInfo_EventDate	70, 343
		hourEntryInfo_GUID	70, 343
		hourEntryInfo_InvoiceDescription	70, 343
		hourEntryInfo_InvoiceGUID	70, 343
		hourEntryInfo_InvoiceQuantity	70, 343
		hourEntryInfo_InvoiceRowGUID	70, 343
		hourEntryInfo_IsApproved	70, 343
		hourEntryInfo_IsBillable	70, 343
		hourEntryInfo_IsProductive	70, 343
		hourEntryInfo_IsReadOnly	70, 343
		hourEntryInfo_OvertimeCode	70, 343
		hourEntryInfo_OverTimeGUID	70, 343
		hourEntryInfo_PhaseGUID	70, 343
		hourEntryInfo_PriceCurrencyShortformOfCaseCompanyCurrency	70, 343
		hourEntryInfo_PriceCurrencyShortformOfInvoicingCurrency	70, 343
		hourEntryInfo_PriceCurrencyShortformOfOrganizationCurrency	70, 343
		hourEntryInfo_PriceCurrencyShortformOfUserCompanyCurrency	70, 343
		hourEntryInfo_Quantity	70, 343
		hourEntryInfo_StartTime	70, 343
		hourEntryInfo_TaskGUID	70, 343
		hourEntryInfo_UnitCost	70, 343
		hourEntryInfo_UnitCostInInvoicingCurrency	70, 343
		hourEntryInfo_UnitCostInOrganizationCurrency	70, 343
		hourEntryInfo_UnitCostInUserCompanyCurrency	70, 343
		hourEntryInfo_UnitPrice	70, 343
		hourEntryInfo_UnitPriceInBaseCurrency	70, 343
		hourEntryInfo_UnitPriceInCaseCompanyCurrency	70, 343
		hourEntryInfo_UnitPriceInUserCompanyCurrency	70, 343
		hourEntryInfo_UserInvoiceQuantity	70, 343
		hourEntryInfo_UserBusinessUnitCode	70, 343
		hourEntryInfo_UserCode	70, 343

- H -

Has Invoice Number	233
HASINVOICENUMBER	233
Head Office Address by Account GUID	178
Head Office Address GUID	60, 129, 148, 149, 151, 339
HeadOfficeAddressByAccountGUID	178
HEADOFFICEADDRESSGUID	60, 129, 148, 149, 151, 339
Hierarchy GUID	129
HIERARCHYGUID	129
Hour Cost	63, 167, 168, 342
Hour Entries by Date	181
Hour Entries by Date and User	184
Hour Entries by Project GUID	179
Hour Entries Changed since	189
Hour Entries Last 2 Years	384
Hour Entry by Date and User GUID	186
Hour Entry by GUID	191
Hour Entry Phases by User GUID	193
HOURCOST	63, 167, 168, 342
HourEntries2Years	384
HourEntriesByCaseGUID	179
HourEntriesByDate	181
HourEntriesByDateAndUser	184
HourEntriesByDateAndUserGUID	186
HourEntriesChangedSince	189
HourEntryByGUID	191
hourEntryGUID	159, 191, 272
hourEntryInfo_AccountGUID	70, 343

hourEntryInfo_UserFirstName	70, 343	includeUnPaidAbsenses	377
hourEntryInfo_UserGUID	70, 343	inCreditNote	267, 270, 277
hourEntryInfo_UserLastName	70, 343	Industries	194
hourEntryInfo_WorkTypeCode	70, 343	Industry GUID	60, 129, 148, 149, 151, 339
hourEntryInfo_WorkTypeGUID	70, 343	INDUSTRYGUID	60, 129, 148, 149, 151, 339
HourEntryPhasesByUserGUID	193	industryInfo_Code	74
hourGUID	379	industryInfo_GUID	74
HTTP Disk Cache Maximum Age (sec)	382	industryInfo_IsActive	74
HTTP Memory Cache Maximum Age (sec)	382	industryInfo_Name	74
HTTP Method	382	Inhouse Revenue Forecast	47, 103, 328
HTTP Status Code	382	INHOUSEREVENUEFORECAST	47, 103, 328
HTTP_DISK_CACHE_MAX_AGE_SEC	382	Inserted	8, 9, 11, 12, 16, 18, 19, 21, 35, 49, 51, 60, 104, 106, 107, 108, 110, 112, 115, 119, 122, 125, 148, 149, 151, 321, 330, 332, 339, 384
HTTP_DISK_CACHE_SAVE	382	insertEndDate	18, 241
HTTP_DISK_CACHE_USE	382	insertPosition	56
HTTP_MEMORY_CACHE_MAX_AGE_SEC	382	insertStartDate	18, 241
HTTP_MEMORY_CACHE_SAVE	382	INSERTTS	8, 9, 11, 12, 16, 18, 19, 21, 35, 49, 51, 60, 104, 106, 107, 108, 110, 112, 115, 119, 122, 125, 148, 149, 151, 321, 330, 332, 339
HTTP_MEMORY_CACHE_USE	382	Internal Name	51, 110, 112, 115, 119, 122, 125, 332, 384
HTTP_METHOD	382	INTERNALNAME	51, 110, 112, 115, 119, 122, 125, 332
HTTP_STATUS_CODE	382	invantive-sql-compress-sparse-arrays	2
http-disk-cache-compression-level	2	invantive-sql-correct-invalid-date	2
http-disk-cache-directory	2	invantive-sql-execution-profile-disk-path	2
http-disk-cache-ignore-write-errors	2	invantive-sql-execution-profile-to-disk	2
http-disk-cache-max-age-sec	2	invantive-sql-forward-filters-to-data-containers	2
http-get-timeout-max-ms	2	invantive-sql-share-byte-arrays	2
http-get-timeout-ms	2	invantive-sql-share-strings	2
http-memory-cache-compression-level	2	invantive-sql-shuffle-fetch-results-data-containers	2
http-memory-cache-max-age-sec	2	invantive-use-cache	2
http-post-timeout-max-ms	2	Invoice Attachment File	195
http-post-timeout-ms	2	Invoice Attachment File GUIDs	195
		Invoice by GUID	196
		Invoice by Number	202
		Invoice Description	64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 343, 384
		Invoice GUID	64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 343, 384
		Invoice Line GUID	64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 343, 384
		Invoice Numbering Business Unit GUID	139, 196, 202, 212, 217, 222, 228, 387
		Invoice Origin Details by Invoice GUID	209
		Invoice PDF	210
		Invoice PDF by Invoice GUID	211
ICON	30, 31, 44, 56, 92, 127, 128, 290, 291, 326, 337		
IDENTIFIER	135, 136		
IETF Language Tag	247, 248		
IETFLANGUAGETAG	247, 248		
ignoreDuplicates	75		
Include Exceptions	51, 110, 112, 115, 119, 122, 125, 332, 384		
Include in Flextime	250, 251		
Include Overtime	51, 110, 112, 115, 119, 122, 125, 332, 384		
includeDefault	265		
INCLUDEEXCEPTIONS	51, 110, 112, 115, 119, 122, 125, 332		
includeInactive	367		
INCLUDEINFLEXTIME	250, 251		
includeInfoForInactiveUser	377		
INCLUDEOVERTIME	51, 110, 112, 115, 119, 122, 125, 332		

Invoice Quantity 64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 343, 384
 Invoice Status 139, 196, 202, 212, 217, 222, 228, 387
 Invoice Statuses 233
 Invoice Template GUID 51, 110, 112, 115, 119, 125, 332, 384
 INVOICEALLOWEDCASEGUIDS 139, 196, 202, 212, 217, 222, 228, 387
 InvoiceAttachmentFile 195
 InvoiceAttachmentFileGUIDs 195
 InvoiceByGUID 196
 InvoiceByNumber 202
 INVOICECONFIG_BILLEXPENSE 139, 196, 202, 212, 217, 222, 228, 387
 INVOICECONFIG_BILLHOUR 139, 196, 202, 212, 217, 222, 228, 387
 INVOICECONFIG_BILLPRODUCT 139, 196, 202, 212, 217, 222, 228, 387
 INVOICECONFIG_EXPENSEITEMDESCRIPTIONFOR
 MAT 139, 196, 202, 212, 217, 222, 228, 387
 INVOICECONFIG_HOURITEMDESCRIPTIONFORMA
 T 139, 196, 202, 212, 217, 222, 228, 387
 INVOICECONFIG_HOURPRIMARYGROUPBY 139,
 196, 202, 212, 217, 222, 228, 387
 INVOICECONFIG_HOURSECONDARYGROUPBY
 139, 196, 202, 212, 217, 222, 228, 387
 INVOICECONFIG_ISTAXFREE 139, 196, 202, 212,
 217, 222, 228, 387
 INVOICECONFIG_PRODUCTITEMDESCRIPTIONFOR
 MAT 139, 196, 202, 212, 217, 222, 228, 387
 INVOICECONFIG_PRODUCTPRIMARYGROUPBY
 139, 196, 202, 212, 217, 222, 228, 387
 INVOICECONFIG_PRODUCTSECONDARYGROUPB
 Y 139, 196, 202, 212, 217, 222, 228, 387
 INVOICECONFIG_SHOWATTACHPRICEEXCLUDIN
 G 139, 196, 202, 212, 217, 222, 228, 387
 INVOICECONFIG_SHOWATTACHUNIT 139, 196,
 202, 212, 217, 222, 228, 387
 INVOICECONFIG_SHOWATTACHUNITPRICE 139,
 196, 202, 212, 217, 222, 228, 387
 INVOICECONFIG_SHOWATTACHVAT 139, 196,
 202, 212, 217, 222, 228, 387
 INVOICECONFIG_SHOWLOGOONINVOICE 139,
 196, 202, 212, 217, 222, 228, 387
 INVOICECONFIG_SHOWPRICEEXCLUDINGVAT
 139, 196, 202, 212, 217, 222, 228, 387
 INVOICECONFIG_SHOWQUANTITY 139, 196, 202,
 212, 217, 222, 228, 387
 INVOICECONFIG_SHOWUNIT 139, 196, 202, 212,
 217, 222, 228, 387
 INVOICECONFIG_SHOWUNITPRICE 139, 196,
 202, 212, 217, 222, 228, 387
 INVOICECONFIG_SHOWVAT 139, 196, 202, 212,
 217, 222, 228, 387
 INVOICECONFIG_SHOWVATBASEAMOUNT 139,
 196, 202, 212, 217, 222, 228, 387
 INVOICECONFIG_SHOWVATRATEAMOUNT 139, 196,
 202, 212, 217, 222, 228, 387
 INVOICECONFIG_SHOWVATRATEPERCENT 139, 196,
 202, 212, 217, 222, 228, 387
 InvoicedHourEntriesByCaseGUID 207
 INVOICEGUID 64, 70, 75, 76, 81, 179, 181, 184,
 186, 189, 191, 195, 196, 207, 209, 211, 233, 236, 239,
 241, 244, 267, 270, 272, 274, 277, 294, 295, 296, 307,
 343, 384
 invoiceInformation_Date 139
 invoiceInformation_DueDate 139
 invoiceInformation_InvoiceStatus 139
 INVOICECONFIG_SHOWVAT 139, 196, 202, 212,
 217, 222, 228, 387
 INVOICECONFIG_TAXFREEDescription 139,
 196, 202, 212, 217, 222, 228, 387
 INVOICECONFIG_TRAVELPRIMARYGROUPBY
 139, 196, 202, 212, 217, 222, 228, 387
 INVOICECONFIG_TRAVELSECONDARYGROUPBY
 139, 196, 202, 212, 217, 222, 228, 387
 Invoiced Hour Entries by Project GUID 207
 invoiceDateEnd 176
 invoiceDateStart 176
 INVOICEDESCRIPTION 64, 70, 76, 81, 179, 181,
 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267,
 270, 272, 274, 277, 307, 343, 384
 INVOICEDetails_InvoiceDate 139, 196, 202,
 212, 217, 222, 228, 387
 INVOICEDetails_InvoiceDueDate 139, 196,
 202, 212, 217, 222, 228, 387
 INVOICEDetails_InvoiceNumber 139, 196,
 202, 212, 217, 222, 228, 387
 INVOICEDetails_InvoiceTotalVATAmount
 139, 196, 202, 212, 217, 222, 228, 387
 INVOICEDetails_InvoiceTotalVATExcludedA
 MOUNT 139, 196, 202, 212, 217, 222, 228, 387
 INVOICEDetails_InvoiceTotalVATIncludedDAM
 OUNT 139, 196, 202, 212, 217, 222, 228, 387
 INVOICEDetails_InvoiceTypeText 139, 196,
 202, 212, 217, 222, 228, 387
 INVOICEDetails_OrderIdentifier 139, 196,
 202, 212, 217, 222, 228, 387
 INVOICEDetails_OriginCode 139, 196, 202,
 212, 217, 222, 228, 387
 INVOICEDetails_PaymentOverdueFineFREET
 EXT 139, 196, 202, 212, 217, 222, 228, 387
 INVOICEDetails_PaymentOverdueFinePERCE
 NT 139, 196, 202, 212, 217, 222, 228, 387
 INVOICEDetails_PaymentTermsFreeText
 139, 196, 202, 212, 217, 222, 228, 387
 INVOICEDetails_SellerReferenceIdentifier
 139, 196, 202, 212, 217, 222, 228, 387
 INVOICEDetails_VATBaseAmount 139, 196,
 202, 212, 217, 222, 228, 387
 INVOICEDetails_VATRateAmount 139, 196,
 202, 212, 217, 222, 228, 387
 INVOICEDetails_VATRatePercent 139, 196,
 202, 212, 217, 222, 228, 387
 InvoicedHourEntriesByCaseGUID 207
 INVOICEGUID 64, 70, 75, 76, 81, 179, 181, 184,
 186, 189, 191, 195, 196, 207, 209, 211, 233, 236, 239,
 241, 244, 267, 270, 272, 274, 277, 294, 295, 296, 307,
 343, 384
 invoiceInformation_Date 139
 invoiceInformation_DueDate 139
 invoiceInformation_InvoiceStatus 139

invoiceInformation_ItemGUIDs	139	INVOICETEMPLATEGUID	51, 110, 112, 115, 119, 122, 125, 332
invoiceInformation_LanguageGUID	139	Invoicing VAT	8, 9, 11, 12, 16, 18, 19, 21, 35, 60, 148, 149, 151, 321, 339
invoiceInformation_Notes	139	INVOICINGVAT	8, 9, 11, 12, 16, 18, 19, 21, 35, 60, 148, 149, 151, 321, 339
invoiceInformation_OurReference	139	Is Active	6, 7, 8, 9, 11, 12, 14, 16, 18, 19, 21, 22, 30, 31, 33, 35, 38, 44, 49, 56, 57, 58, 59, 60, 69, 74, 86, 87, 88, 92, 94, 95, 98, 99, 104, 106, 107, 108, 118, 127, 128, 129, 130, 131, 133, 134, 135, 136, 146, 147, 148, 149, 151, 194, 233, 248, 249, 250, 251, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 289, 290, 291, 298, 299, 304, 305, 306, 310, 319, 321, 326, 330, 337, 338, 339, 350, 351, 352, 354, 356, 360, 361, 363, 365, 367, 369, 371, 372, 374, 376, 378, 379, 380, 381
invoiceInformation_OverDuelInterest	139	Is All Day	22, 24, 26, 40, 324
invoiceInformation_PaymentTerm	139	Is also Item Status	92, 290, 291
invoiceInformation_ReceiverAddressline	139	Is Approved	64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 319, 343, 384
invoiceInformation_ReceiverCity	139	Is Billable	64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 343, 384
invoiceInformation_ReceiverContactEmail	139	Is Billing Address	46, 100, 101, 103, 177, 178, 327
invoiceInformation_ReceiverContactName	139	Is Closed	22, 24, 26, 40, 51, 110, 112, 115, 119, 122, 125, 324, 332, 384
invoiceInformation_ReceiverCountry	139	Is Completed	84, 251, 253, 255, 256, 348
invoiceInformation_ReceiverCustomerName	139	Is Credit Note	139, 196, 202, 212, 217, 222, 228, 387
invoiceInformation_ReceiverPostalCode	139	Is Daily Allowance Billable	51, 110, 112, 115, 119, 122, 125, 332, 384
invoiceInformation_ReceiverState	139	Is Default	6, 7, 30, 31, 33, 44, 87, 99, 135, 136, 233, 257, 258, 262, 263, 264, 319, 326, 351, 360, 378, 379, 380, 381
invoiceInformation_ReceiverVatNumber	139	Is Duration	22, 24, 26, 40, 324
invoiceInformation_SenderAddressline	139	Is EU	136, 137, 138
invoiceInformation_SenderCity	139	Is in Progress	92, 290, 291
invoiceInformation_SenderContactName	139	Is Internal	8, 9, 11, 12, 16, 18, 19, 21, 35, 51, 60, 110, 112, 115, 119, 122, 125, 148, 149, 151, 257, 258, 261, 332, 339, 384
invoiceInformation_SenderCountry	139	Is Locked	84, 193, 251, 253, 255, 256, 319, 348
invoiceInformation_SenderName	139	Is Lost	92, 290, 291
invoiceInformation_SenderPostalCode	139	Is Mileage Billable	51, 110, 112, 115, 119, 122, 125, 332, 384
invoiceInformation_SenderState	139	Is Offer	92, 290, 291
invoiceInformation_SenderVatNumber	139	Is Out-sourcing Cost	64, 76, 81, 233, 236, 239, 241, 244, 274, 277, 307
invoiceInformation_YourReference	139	Is Overtime Allowed	63, 167, 168, 342
invoiceNumber	202, 210, 294	Is Paid	233
INVOICENUMBERINGBUSINESSUNITGUID	139, 196, 202, 210, 212, 217, 222, 228, 387	Is Paid Leave	30, 31, 44, 326
InvoiceOriginDetailsByInvoiceGUID	209	Is Postal Address	46, 100, 101, 103, 177, 178, 327
InvoicePdf	210	Is Private	22, 24, 26, 40, 324
InvoicePdfByInvoiceGUID	211	Is Productive	70, 99, 179, 181, 184, 186, 189, 191, 207, 267, 270, 272, 343, 360, 378, 379, 380, 381, 384
INVOICEQUANTITY	64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 343, 384		
INVOICEROWGUID	64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 343, 384		
INVOICEROWORIGINS	209		
INVOICEROWS	139, 196, 202, 212, 217, 222, 228, 387		
Invoices	387		
Invoices by Date	212		
Invoices by Date and Status	217		
Invoices by Status	222		
Invoices by Status and Business Unit	228		
InvoicesByDate	212		
InvoicesByDateAndStatus	217		
InvoicesByStatus	222		
InvoicesByStatusAndBusinessUnit	228		
INVOICESTATUS	139, 176, 196, 202, 212, 217, 222, 228, 387		
InvoiceStatuses	233		

Is Read-only 22, 24, 26, 40, 64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 242, 267, 270, 272, 274, 277, 307, 324, 343, 384
 Is Realized 64, 76, 81, 233, 236, 239, 241, 244, 274, 277, 307
 Is Reimbursed 139, 196, 202, 212, 217, 222, 228, 387
 Is Sent 233
 Is Tax Free 139, 196, 202, 212, 217, 222, 228, 387
 Is Travel Type Default 265
 Is Unassigned 22, 24, 26, 40, 324
 Is Visit Address 46, 100, 101, 103, 177, 178, 327
 Is Volume Pricing 257, 258
 Is Waiting Payment 233
 Is Won 92, 290, 291
 Is Work Time 250, 251
 ISACTIVE 6, 7, 8, 9, 11, 12, 14, 16, 18, 19, 21, 22, 30, 31, 33, 35, 38, 44, 49, 56, 57, 58, 59, 60, 69, 74, 86, 87, 88, 92, 94, 95, 98, 99, 104, 106, 107, 108, 118, 127, 128, 129, 130, 131, 133, 134, 135, 136, 146, 147, 148, 149, 151, 194, 233, 248, 249, 250, 251, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 289, 290, 291, 298, 299, 304, 305, 306, 310, 319, 321, 326, 330, 334, 337, 338, 339, 350, 351, 352, 354, 356, 360, 361, 363, 365, 367, 369, 371, 372, 374, 376, 378, 379, 380, 381
 ISALLDAY 22, 24, 26, 40, 324
 ISALSOITEMSTATUS 92, 290, 291
 ISAPPROVED 64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 319, 343, 384
 ISBILLABLE 64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 343, 384
 ISBILLINGADDRESS 46, 100, 101, 103, 177, 178, 327
 ISCLOSED 22, 24, 26, 40, 51, 110, 112, 115, 119, 122, 125, 324, 332
 ISCOMPLETED 84, 251, 253, 255, 256, 348
 ISCREDITNOTE 139, 196, 202, 212, 217, 222, 228, 387
 ISDAILYALLOWANCEBILLABLE 51, 110, 112, 115, 119, 122, 125, 332
 ISDEFAULT 6, 7, 30, 31, 33, 44, 87, 99, 135, 136, 233, 257, 258, 262, 263, 264, 319, 326, 351, 360, 378, 379, 380, 381
 ISDURATION 22, 24, 26, 40, 324
 ISEU 136, 137, 138
 ISINPROGRESS 92, 290, 291
 ISINTERNAL 8, 9, 11, 12, 16, 18, 19, 21, 35, 51, 60, 110, 112, 115, 119, 122, 125, 148, 149, 151, 257, 258, 321, 332, 339
 ISLOCKED 84, 193, 251, 253, 255, 256, 319, 348
 ISLOST 92, 290, 291
 ISMILEAGEBILLABLE 51, 110, 112, 115, 119, 122, 125, 332
 ISO Code 146, 147
 ISO Code A2 136, 137, 138
 ISO Code A3 136, 137, 138
 ISOCODE 146, 147
 ISOCODEA2 136, 137, 138
 ISOCODEA3 136, 137, 138
 ISOFFER 92, 290, 291
 isoName 137
 ISOTHERTRAVELEXPENSESBILLABLE 51, 110, 112, 115, 119, 122, 125, 332
 ISOUTSOURCINGCOST 64, 76, 81, 233, 236, 239, 241, 244, 274, 277, 307
 ISOVERTIMEALLOWED 63, 167, 168, 342
 ISPAID 233
 ISPAIDLEAVE 30, 31, 44, 326
 ISPOSTALADDRESS 46, 100, 101, 103, 177, 178, 327
 ISPRIVATE 22, 24, 26, 40, 324
 ISPRODUCTIVE 70, 99, 179, 181, 184, 186, 189, 191, 207, 267, 270, 272, 343, 360, 378, 379, 380, 381, 384
 ISREADONLY 22, 24, 26, 40, 64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 324, 343, 384
 ISREALISED 64, 76, 81, 233, 236, 239, 241, 244, 274, 277, 307
 ISREIMBURSED 139, 196, 202, 212, 217, 222, 228, 387
 ISSENT 233
 ISTAXFREE 139, 196, 202, 212, 217, 222, 228, 387
 ISTRAVELTYPEDEFAULT 265
 ISUNASSIGNED 22, 24, 26, 40, 324
 ISVISITADDRESS 46, 100, 101, 103, 177, 178, 327
 ISVOLUMEPRICING 257, 258
 ISWAITINGPAYMENT 233
 ISWON 92, 290, 291
 ISWORKTIME 250, 251
 Item by GUID 233
 Item by Purchase Order Number 236
 ItemByGUID 233
 ItemByPurchaseOrderNumber 236
 itemGUID 160, 233, 274
 itemInfo_BillingDependencyTaskGUID 64, 76
 itemInfo_BillingSchedule 64, 76
 itemInfo_CaseGUID 64, 76
 itemInfo_CaseNumber 64, 76
 itemInfo_CostCenterGUID 64, 76
 itemInfo_CostCenterIdentifier 64, 76
 itemInfo_CostCurrencyShortform 64, 76
 itemInfo_CostCurrencyShortformOffInvoicingCurrency 64, 76

Latest Estimation Date 51, 110, 112, 115, 119, 122, 125, 332, 384

LATESTESTIMATIONDATE 51, 110, 112, 115, 119, 122, 125, 332

Lead Rating 51, 110, 112, 115, 119, 122, 125, 332, 384

Lead Source by GUID 248

Lead Source GUID 51, 110, 112, 115, 119, 122, 125, 332, 384

Lead Sources 249

LEADRATING 51, 110, 112, 115, 119, 122, 125, 332

LeadSourceByGUID 248

LEADSOURCEGUID 51, 110, 112, 115, 119, 122, 384, 125, 248, 332

LeadSources 249

LICENSETYPE 95, 118, 356, 361, 363, 365, 367, 369, 371, 372, 374

LOCATION 22, 24, 26, 40, 324

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 139, 196, 202, 212, 217, 222, 228, 387

MAINCASEGUID 139, 196, 202, 212, 217, 222, 228, 387

maximum-length-identifiers 2

maxInvoices 222

maxRows 26

max-url-length-accepted 2

max-url-length-desired 2

Measurement Unit 64, 76, 81, 86, 94, 233, 236, 239, 241, 244, 259, 260, 261, 266, 274, 277, 304, 305, 306, 307, 310, 350, 354

MEASUREMENTUNIT 64, 76, 81, 86, 94, 233, 236, 239, 241, 244, 259, 260, 261, 266, 274, 277, 304, 305, 306, 307, 310, 350, 354

members 84

memberStatus 45

MIMETYPE 195

MIME-type 195

minimum-length-text 2

MONTH 47, 103, 328

- N -

Name 6, 7, 8, 9, 11, 12, 14, 15, 16, 18, 19, 21, 22, 24, 25, 26, 28, 29, 30, 31, 33, 35, 38, 40, 44, 49, 51, 56, 60, 64, 74, 76, 81, 84, 86, 87, 88, 92, 94, 99, 104, 106, 107, 108, 110, 112, 115, 119, 122, 125, 127, 128, 129, 135, 136, 137, 138, 148, 149, 151, 173, 174, 193, 194, 195, 233, 236, 239, 241, 244, 247, 248, 249, 250, 251, 253, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 274, 277, 289, 290, 291, 303, 304, 305, 306, 307, 310, 311, 319, 321, 324, 326, 330, 332, 337, 339, 348, 350, 351, 352, 354, 360, 378, 379, 380, 381,

Native Platform Scalar Requests 382

NATIVEPLATFORMSCALARREQUESTS 382

Next Invoice Number 49, 104, 106, 107, 108, 330

NEXTINVOICENUMBER 49, 104, 106, 107, 108, 330

NOTES 8, 9, 11, 12, 16, 18, 19, 21, 22, 24, 26, 35, 40, 47, 60, 95, 103, 118, 139, 148, 149, 151, 196, 202, 212, 217, 222, 228, 312, 313, 315, 316, 318, 321, 324, 328, 339, 356, 361, 363, 365, 367, 369, 371, 372, 374, 387

Notes after Invoice Lines 139, 196, 202, 212, 217, 222, 228, 387

Notes before Invoice Lines 139, 196, 202, 212, 217, 222, 228, 387

NOTESAFTERINVOICEROWS 139, 196, 202, 212, 217, 222, 228, 387

NOTESBEFOREINVOICEROWS 139, 196, 202, 212, 217, 222, 228, 387

npt 382

NUMBER 8, 9, 11, 12, 15, 16, 18, 19, 21, 35, 60, 139, 148, 149, 151, 196, 202, 212, 217, 222, 228, 265, 312, 313, 315, 316, 318, 321, 339, 387

- O -

onlyActive 252

options 14, 15, 19, 21, 101, 107, 108, 125, 134, 149, 151, 175, 176, 212, 217, 222, 256, 262, 266, 310, 315, 316, 318, 374, 381

Order Date 64, 76, 81, 233, 236, 239, 241, 244, 274, 277, 307

Order Number 51, 110, 112, 115, 119, 122, 125, 139, 196, 202, 212, 217, 222, 228, 332, 384, 387

ORDERDATE 64, 76, 81, 233, 236, 239, 241, 244, 274, 277, 307

ORDERNUMBER 51, 110, 112, 115, 119, 122, 125, 139, 196, 202, 212, 217, 222, 228, 332, 387

Organization GUID 95, 118, 356, 361, 363, 365, 367, 369, 371, 372, 374

ORGANIZATIONGUID	95, 118, 356, 361, 363, 365, 367, 369, 371, 372, 374	OWNERUSERGUID	22, 24, 26, 40, 84, 251, 253, 255, 256, 324, 348
ORIG_SYSTEM_GROUP	382	OWNERUSERLASTNAME	22, 24, 26, 40, 324
ORIG_SYSTEM_REFERENCE	382		
Original Deadline	84, 251, 253, 255, 256, 348		
Original Planned Start Date	84, 251, 253, 255, 256, 348		
Original System Group	382	Parent Business Unit GUID	49, 104, 106, 107, 108, 330
Original System Reference	382	Parent GUID	14, 38
Original Work Estimate	84, 251, 253, 255, 256, 348	Parent Phase GUID	84, 193, 251, 253, 255, 256, 348
ORIGINALDEADLINE	84, 251, 253, 255, 256, 348	PARENTBUSINESSUNITGUID	49, 104, 106, 107, 108, 330
ORIGINALPLANNEDSTARTDATE	84, 251, 253, 255, 256, 348	PARENTGUID	14, 38
ORIGINALWORKESTIMATE	84, 251, 253, 255, 256, 348	PARENTPHASEGUID	84, 193, 251, 253, 255, 256, 348
Our Reference	51, 110, 112, 115, 119, 122, 125, 139, 196, 202, 212, 217, 222, 228, 332, 384, 387	partition-slot-based-rate-limit-length-ms	2
OURREFERENCE	51, 110, 112, 115, 119, 122, 125, 139, 196, 202, 212, 217, 222, 228, 332, 387	partition-slot-based-rate-limit-slots	2
Outsourcing Revenue Forecast	47, 103, 328	Payload	382
OUTSOURCINGREVENUEFORECAST	47, 103, 328	PAYLOAD_TEXT	382
Overdue Interest	8, 9, 11, 12, 16, 18, 19, 21, 35, 49, 51, 60, 104, 106, 107, 108, 110, 112, 115, 119, 122, 125, 139, 148, 149, 151, 196, 202, 212, 217, 222, 228, 321, 330, 332, 339, 384, 387	Payment Date	139, 196, 202, 212, 217, 222, 228, 387
OVERDUEINTEREST	8, 9, 11, 12, 16, 18, 19, 21, 35, 49, 51, 60, 104, 106, 107, 108, 110, 112, 115, 119, 122, 125, 139, 148, 149, 151, 196, 202, 212, 217, 222, 228, 321, 330, 332, 339, 384, 387	Payment Status	139, 196, 202, 212, 217, 222, 228, 387
Overtime Allowed	377	Payment Term	8, 9, 11, 12, 16, 18, 19, 21, 35, 49, 51, 60, 104, 106, 107, 108, 110, 112, 115, 119, 122, 125, 139, 148, 149, 151, 196, 202, 212, 217, 222, 228, 321, 330, 332, 339, 384, 387
Overtime by GUID	250	PAYMENTDATE	139, 196, 202, 212, 217, 222, 228, 295, 387
Overtime Code	70, 179, 181, 184, 186, 189, 191, 207, 267, 270, 272, 343, 384	PAYMENTSTATUS	139, 196, 202, 212, 217, 222, 228, 387
Overtime GUID	70, 179, 181, 184, 186, 189, 191, 207, 267, 270, 272, 343, 384	PAYMENTTERM	8, 9, 11, 12, 16, 18, 19, 21, 35, 49, 51, 60, 104, 106, 107, 108, 110, 112, 115, 119, 122, 125, 139, 148, 149, 151, 196, 202, 212, 217, 222, 228, 321, 330, 332, 339, 387
OVERTIMEALLOWED	377	PERCENTAGE	250, 251
OverTimeByGUID	250	Phase by GUID	251
OVERTIMECODE	70, 179, 181, 184, 186, 189, 191, 207, 267, 270, 272, 343, 384	Phase GUID	70, 89, 91, 179, 181, 184, 186, 189, 191, 207, 267, 270, 272, 283, 284, 285, 286, 287, 288, 300, 301, 302, 343, 353, 384
OVERTIMEGUID	70, 179, 181, 184, 186, 189, 191, 207, 250, 267, 270, 272, 343, 384	Phase Members by Phase GUID	252
Overtimes	251	PhaseByGUID	251
Owner User Business Unit Code	22, 24, 26, 40, 324	PhaseCode	255
Owner User Code	22, 24, 26, 40, 324	PHASEGUID	70, 89, 91, 160, 179, 181, 184, 186, 189, 191, 207, 251, 252, 267, 270, 272, 283, 284, 285, 286, 287, 288, 296, 300, 301, 302, 343, 353, 380, 384
Owner User First Name	22, 24, 26, 40, 324	phaseGUIDToTransferItemsTo	160
Owner User GUID	22, 24, 26, 40, 84, 251, 253, 255, 256, 324, 348	phaseInfo_CaseGUID	84, 348
Owner User Last Name	22, 24, 26, 40, 324	phaseInfo_Code	84, 348
OWNERUSERBUSINESSUNITCODE	22, 24, 26, 40, 324	phaseInfo_Deadline	84, 348
OWNERUSERCODE	22, 24, 26, 40, 324	phaseInfo_Description	84, 348
OWNERUSERFIRSTNAME	22, 24, 26, 40, 324		

phaseInfo_FlatRate_AdditionalHoursUnitPrice 84, 348
 phaseInfo_FlatRate_AreAdditionalHoursBillable 84, 348
 phaseInfo_FlatRate_Hours 84, 348
 phaseInfo_FlatRate_Price 84, 348
 phaseInfo_GUID 84, 348
 phaseInfo_IsCompleted 84, 348
 phaseInfo_IsLocked 84, 348
 phaseInfo_Name 84, 348
 phaseInfo_OriginalDeadline 84, 348
 phaseInfo_OriginalPlannedStartDate 84, 348
 phaseInfo_OriginalWorkEstimate 84, 348
 phaseInfo_OwnerUserGUID 84, 348
 phaseInfo_ParentPhaseGUID 84, 348
 phaseInfo_PlannedStartDate 84, 348
 phaseInfo_PricePerHour 84, 348
 phaseInfo_SortOrder 84, 348
 phaseInfo_WorkEstimate 84, 348
 phaseInfo_WorkTypeGUID 84, 348
 PhaseMembersByPhaseGUID 252
 Phases by Code 255
 Phases by Project GUID 253
 Phases Changed since 256
 PhasesByCaseGUID 253
 PhasesByCode 255
 PhasesChangedSince 256
 PHONE 46, 95, 100, 101, 103, 118, 177, 178, 327, 356, 361, 363, 365, 367, 369, 371, 372, 374
 Photo File GUID 95, 118, 356, 361, 363, 365, 367, 369, 371, 372, 374
 PHOTOFILEGUID 95, 118, 356, 361, 363, 365, 367, 369, 371, 372, 374
 Planned Billing Date 64, 76, 81, 233, 236, 239, 241, 244, 274, 277, 307
 Planned Start Date 84, 251, 253, 255, 256, 348
 PLANNEDBILLINGDATE 64, 76, 81, 233, 236, 239, 241, 244, 274, 277, 307
 PLANNEDSTARTDATE 84, 251, 253, 255, 256, 348
 PO Number 139, 196, 202, 212, 217, 222, 228, 387
 PONUMBER 64, 139, 196, 202, 212, 217, 222, 228, 236, 387
 Post Office 95, 118, 356, 361, 363, 365, 367, 369, 371, 372, 374
 Postal Code 46, 95, 100, 101, 103, 118, 177, 178, 327, 356, 361, 363, 365, 367, 369, 371, 372, 374
 POSTALCODE 46, 95, 100, 101, 103, 118, 177, 178, 327, 356, 361, 363, 365, 367, 369, 371, 372, 374
 POSTOFFICE 95, 118, 356, 361, 363, 365, 367, 369, 371, 372, 374
 pre-request-delay-ms 2
 Price Currency Short Form of Invoicing Currency 64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 343, 384
 Price Currency Short Form of Organization Currency 64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 343, 384
 Price Currency Short Form of Project Company Currency 64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 343, 384
 Price Currency Short Form of User Company Currency 64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 343, 384
 Price List by GUID 257
 Price Lists 258
 Price per Hour 84, 251, 253, 255, 256, 348
 PRICECURRENCYSHORTFORMOFCASECOMPANY CURRENCY 64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 343, 384
 PRICECURRENCYSHORTFORMOFINVOICINGCURRENCY 64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 343, 384
 PRICECURRENCYSHORTFORMOFORGANIZATION CURRENCY 64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 343, 384
 PRICECURRENCYSHORTFORMOFUSERCOMPANY CURRENCY 64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 343, 384
 Pricelist GUID 8, 9, 11, 12, 16, 18, 19, 21, 35, 51, 60, 110, 112, 115, 119, 122, 125, 148, 149, 151, 321, 332, 339, 384
 PricelistByGUID 257
 PRICELISTGUID 8, 9, 11, 12, 16, 18, 19, 21, 35, 51, 60, 110, 112, 115, 119, 122, 125, 148, 149, 151, 257, 321, 332, 339
 Pricelists 258
 PRICEPERHOUR 84, 251, 253, 255, 256, 348
 Pricing Rule 51, 110, 112, 115, 119, 122, 125, 332, 384
 Pricing Type 51, 110, 112, 115, 119, 122, 125, 332, 384
 PRICINGRULE 51, 110, 112, 115, 119, 122, 125, 332
 PRICINGTYPE 51, 110, 112, 115, 119, 122, 125, 332
 Print Order 195
 PRINTORDER 195
 Priority 51, 110, 112, 115, 119, 122, 125, 332, 384

Probability	51, 110, 112, 115, 119, 122, 125, 332, 384	Profile GUID	95, 118, 356, 361, 363, 365, 367, 369, 371, 372, 374
Product by Code	259	PROFILEGUID	95, 118, 356, 361, 363, 365, 367, 369, 371, 372, 374
Product by Code Include Inactive	260	Project Business Unit Code	70, 179, 181, 184, 186, 189, 191, 207, 267, 270, 272, 343, 384
Product by GUID	261	Project by GUID	110
Product Categories Changed since	262	Project by Number	112
Product Category by Code	263	Project by Task GUID	115
Product Category by GUID	264	Project Company Currency GUID	51, 110, 112, 115, 119, 122, 125, 332, 384
Product Category GUID	86, 259, 260, 261, 266, 350	Project Cost Center Identifier	70, 179, 181, 184, 186, 189, 191, 207, 267, 270, 272, 343, 384
Product Code	64, 76, 81, 233, 236, 239, 241, 244, 274, 277, 307	Project GUID	22, 24, 26, 40, 64, 70, 76, 81, 84, 89, 91, 93, 171, 179, 181, 184, 186, 189, 191, 193, 207, 233, 236, 239, 241, 244, 251, 253, 255, 256, 267, 270, 272, 274, 277, 283, 284, 285, 286, 287, 288, 292, 293, 307, 324, 343, 348, 353, 384
Product Cost Account Associations by GUID	265	Project Member Users by Project GUID	118
Product GUID	64, 76, 81, 233, 236, 239, 241, 244, 274, 277, 307	Project Name	70, 179, 181, 184, 186, 189, 191, 207, 267, 270, 272, 343, 384
Product Type	64, 76, 81, 233, 236, 239, 241, 244, 274, 277, 307	Project Number	51, 64, 70, 76, 81, 110, 112, 115, 119, 122, 125, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 332, 343, 384
ProductByCode	259	Project Owner User Code	51, 110, 112, 115, 119, 122, 125, 332, 384
ProductByCodeIncludeInactive	260	Project Owner User GUID	51, 110, 112, 115, 119, 122, 125, 332, 384
ProductByGUID	261	Project Status by GUID	127
ProductCategoriesChangedSince	262	Project Statuses	128
ProductCategoryByCode	263	Project Tags	129
ProductCategoryByGUID	264	Projects	119, 209
productCategoryCode	263	Projects by Account GUID	122
PRODUCTCATEGORYGUID	86, 162, 259, 260, 261, 264, 266, 350	Projects Changed since	125
productCategoryInfo_Code	87, 351	Purchase Order Number	64, 76, 81, 233, 236, 239, 241, 244, 274, 277, 307
productCategoryInfo_GUID	87, 351	Purchase VAT	64, 76, 81, 233, 236, 239, 241, 244, 274, 277, 307
productCategoryInfo_IsActive	87, 351	PURCHASEORDERNUMBER	64, 76, 81, 233, 236, 239, 241, 244, 274, 277, 307
productCategoryInfo_IsDefault	87, 351	PURCHASEVAT	64, 76, 81, 233, 236, 239, 241, 244, 274, 277, 307
productCategoryInfo_Name	87, 351		
PRODUCTCODE	64, 76, 81, 233, 236, 239, 241, 244, 259, 260, 274, 277, 307		
ProductCostAccountAssociationsByGUID	265		
PRODUCTGUID	64, 76, 81, 161, 233, 236, 239, 241, 244, 261, 265, 274, 277, 307		
productInfo_Code	86, 350		
productInfo_Description	86, 350		
productInfo_GUID	86, 350		
productInfo_IsActive	86, 350		
productInfo_MeasurementUnit	86, 350		
productInfo_Name	86, 350		
productInfo_ProductCategoryGUID	86, 350		
productInfo_SalesAccountGUID	86, 350		
productInfo_Type	86, 350		
productInfo_UnitCost	86, 350		
productInfo_UnitPrice	86, 350		
productInfo_UselnWorkTimeEntry	86, 350		
productInfo_VAT	86, 350		
Products Changed since	266		
ProductsChangedSince	266		
PRODUCTTYPE	64, 76, 81, 233, 236, 239, 241, 244, 274, 277, 307		

- Q -

QUANTITY 64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 300, 301, 302, 307, 343, 384

- R -

RATE 146, 147

READONLY 377
 Read-only 377
 READY 377
 Receiver Addressline 139, 196, 202, 212, 217, 222, 228, 387
 Receiver City 139, 196, 202, 212, 217, 222, 228, 387
 Receiver Contact Email 139, 196, 202, 212, 217, 222, 228, 387
 Receiver Contact Name 139, 196, 202, 212, 217, 222, 228, 387
 Receiver Country 139, 196, 202, 212, 217, 222, 228, 387
 Receiver Customer Name 139, 196, 202, 212, 217, 222, 228, 387
 Receiver Postal Code 139, 196, 202, 212, 217, 222, 228, 387
 Receiver State 139, 196, 202, 212, 217, 222, 228, 387
 Receiver VAT Number 139, 196, 202, 212, 217, 222, 228, 387
 RECEIVERADDRESSLINE 139, 196, 202, 212, 217, 222, 228, 387
 RECEIVERCITY 139, 196, 202, 212, 217, 222, 228, 387
 RECEIVERCONTACTEMAIL 139, 196, 202, 212, 217, 222, 228, 387
 RECEIVERCONTACTNAME 139, 196, 202, 212, 217, 222, 228, 387
 RECEIVERCOUNTRY 139, 196, 202, 212, 217, 222, 228, 387
 RECEIVERCUSTOMERNAME 139, 196, 202, 212, 217, 222, 228, 387
 RECEIVERPOSTALCODE 139, 196, 202, 212, 217, 222, 228, 387
 RECEIVERSTATE 139, 196, 202, 212, 217, 222, 228, 387
 RECEIVERVATNUMBER 139, 196, 202, 212, 217, 222, 228, 387
 Recurrence End Date 64, 76, 81, 233, 236, 239, 241, 244, 274, 277, 307
 Recurrence End Type 64, 76, 81, 233, 236, 239, 241, 244, 274, 277, 307
 Recurrence Frequency 64, 76, 81, 233, 236, 239, 241, 244, 274, 277, 307
 Recurrence Start Date 64, 76, 81, 233, 236, 239, 241, 244, 274, 277, 307
 Recurrence Times 64, 76, 81, 233, 236, 239, 241, 244, 274, 277, 307
 RECURRENCEENDDATE 64, 76, 81, 233, 236, 239, 241, 244, 274, 277, 307
 RECURRENCEENDTYPE 64, 76, 81, 233, 236, 239, 241, 244, 274, 277, 307
 RECURRENCFREQUENCY 64, 76, 81, 233, 236, 239, 241, 244, 274, 277, 307
 RECURRENCESTARTDATE 64, 76, 81, 233, 236, 239, 241, 244, 274, 277, 307
 RECURRENCETIMES 64, 76, 81, 233, 236, 239, 241, 244, 274, 277, 307
 Recurring Item GUID 64, 76, 81, 233, 236, 239, 241, 244, 274, 277, 307
 RECURRINGITEMGUID 64, 76, 81, 233, 236, 239, 241, 244, 274, 277, 307
 Reference Number 139, 196, 202, 212, 217, 222, 228, 387
 REFERENCENUMBER 139, 196, 202, 212, 217, 222, 228, 294, 387
 Reimburse Invoice GUID 139, 196, 202, 212, 217, 222, 228, 387
 Reimbursed Hour Entries by Date 267
 Reimbursed Hour Entries Changed since 270
 Reimbursed Hour Entry by GUID 272
 Reimbursed Item by GUID 274
 Reimbursed Items Changed since 277
 ReimbursedHourEntriesByDate 267
 ReimbursedHourEntriesChangedSince 270
 ReimbursedHourEntryByGUID 272
 ReimbursedItemByGUID 274
 ReimbursedItemsChangedSince 277
 REIMBURSEINVOICEGUID 139, 196, 202, 212, 217, 222, 228, 387
 Remove Account Group Members 280
 Remove Activity Contact Participant 280
 Remove Activity Resource Participant 281
 Remove Activity User Participant 282
 RemoveAccountGroupMembers 280
 RemoveActivityContactParticipant 280
 REMOVEACTIVITYCONTACTPARTICIPANTRESULT 280
 RemoveActivityResourceParticipant 281
 REMOVEACTIVITYRESOURCEPARTICIPANTRESULT 281
 RemoveActivityUserParticipant 282
 REMOVEACTIVITYUSERPARTICIPANTRESULT 282
 replacementActivityTypeGUID 155
 replacementCaseStatusGUID 156
 requested-page-size 2
 requests-parallel-max 2
 Resource Allocations 284
 Resource Allocations by GUID 283
 Resource Allocations by Project GUID 285
 Resource Allocations by Project GUID and User GUID 286
 Resource Allocations by User GUID 287
 Resource Allocations Changed since 288
 Resource by GUID 289
 Resource GUID 25, 28, 29, 31

ResourceAllocationByGUID 283
 resourceAllocationGUID 164, 283
 resourceAllocationInfo_AllocatedHours 89, 353
 resourceAllocationInfo_AllocatedPercentage 89, 353
 resourceAllocationInfo_CaseGUID 89, 353
 resourceAllocationInfo_EndDate 89, 353
 resourceAllocationInfo_GUID 89, 353
 resourceAllocationInfo_PhaseGUID 89, 353
 resourceAllocationInfo_StartDate 89, 353
 resourceAllocationInfo_UserGUID 89, 353
 ResourceAllocations 91, 284
 ResourceAllocationsByCaseGUID 285
 ResourceAllocationsByCaseGUIDUserGUID 286
 ResourceAllocationsByUserGUID 287
 ResourceAllocationsChangedSince 288
 ResourceByGUID 289
 resourceGUID 25, 28, 29, 31, 43, 163, 281, 289
 resourceInfo_GUID 88, 352
 resourceInfo_IsActive 88, 352
 resourceInfo_Name 88, 352
 Resources 290
 Response Cache Maximum Age (sec) 382
 Result BLOB 382
 Result Date Time 382
 Result Number 382
 Result Text 382
 RESULT_BLOB 382
 RESULT_DATE_TIME_UTC 382
 RESULT_NUMBER 382
 RESULT_TEXT 382
 result-set-memory-cache 2
 Reverse Charge 8, 9, 11, 12, 16, 18, 19, 21, 35, 60, 148, 149, 151, 321, 339
 Reverse Charge Description 8, 9, 11, 12, 16, 18, 19, 21, 35, 60, 148, 149, 151, 321, 339
 REVERSECHARGE 8, 9, 11, 12, 16, 18, 19, 21, 35, 60, 148, 149, 151, 321, 339
 REVERSECHARGEDESCRIPTION 8, 9, 11, 12, 18, 19, 21, 35, 60, 148, 149, 151, 321, 339
 RIGHTS_ACCOUNT 6, 7, 33, 319, 359, 366
 RIGHTS_ACCOUNTDELETE 6, 7, 33, 319, 359, 366
 RIGHTS_ADMINISTRATOR 6, 7, 33, 319, 359, 366
 rights_API 359
 RIGHTS_CASE 6, 7, 33, 319, 359, 366
 RIGHTS_CASEDELETE 6, 7, 33, 319, 359, 366
 RIGHTS_CASEOWNERWORKHOURAPPROVAL 6, 7, 33, 319, 359, 366
 RIGHTS_EDITCASESTATUS 170
 rights_Rights_Account 359
 rights_Rights_AccountDelete 359
 rights_Rights_Administrator 359
 rights_Rights_Case 359
 rights_Rights_CaseDelete 359
 rights_Rights_CaseOwnerWorkHourApproval 359
 rights_Rights_ScheduleJobs 359
 rights_Rights_Sharing 359
 rights_Rights_TravelReimbursement 359
 rights_Rights_Users 359
 rights_Rights_WorkHourApproval 359
 RIGHTS_SCHEDULEJOBS 6, 7, 33, 319, 359, 366
 RIGHTS_SHARING 6, 7, 33, 319, 359, 366
 RIGHTS_SHOWBILLING 170
 RIGHTS_SHOWCASESTATUS 170
 RIGHTS_SHOWFILESANLINKS 170
 RIGHTS_SHOWGANTTCHART 170
 RIGHTS_SHOWPHASETREE 170
 RIGHTS_SHOWREADYTOBILL 170
 RIGHTS_SHOWWORKHOURESTIMATES 170
 RIGHTS_SHOWWORKHOURS 170
 RIGHTS_SHOWWORKHOURSTARTANDENDTIME 170
 RIGHTS_TRAVELREIMBURSEMENT 6, 7, 33, 319, 359, 366
 RIGHTS_USERS 6, 7, 33, 319, 359, 366
 RIGHTS_WORKHOURAPPROVAL 6, 7, 33, 319, 359, 366
 Root Task GUID 51, 110, 112, 115, 119, 122, 125, 332, 384
 rootBusinessUnitGuid 107
 ROOTTASKGUID 51, 110, 112, 115, 119, 122, 125, 332
 Run without Actions 382

- S -

Sales Account GUID 64, 76, 81, 86, 99, 233, 236, 239, 241, 244, 259, 260, 261, 266, 274, 277, 307, 350, 360, 378, 379, 380, 381
 Sales Close Date 51, 110, 112, 115, 119, 122, 125, 332, 384
 Sales Person User Code 51, 110, 112, 115, 119, 122, 125, 332, 384
 Sales Person User GUID 51, 110, 112, 115, 119, 122, 125, 332, 384
 Sales Process by GUID 290
 Sales Process GUID 51, 93, 110, 112, 115, 119, 122, 125, 292, 293, 332, 384
 Sales Processes 291
 Sales Status by GUID 292
 Sales Status GUID 51, 64, 76, 81, 110, 112, 115, 119, 122, 125, 233, 236, 239, 241, 244, 274, 277, 307, 332, 384

Sales Statuses for Project 293
 SALESACCOUNTGUID 64, 76, 81, 86, 99, 233, 236, 239, 241, 244, 259, 260, 261, 266, 274, 277, 307, 350, 360, 378, 379, 380, 381
 SALESCLOSEDATE 51, 110, 112, 115, 119, 122, 125, 332
 SALESPERSONUSERCODE 51, 110, 112, 115, 119, 122, 125, 332
 SALESPERSONUSERGUID 51, 110, 112, 115, 119, 122, 125, 332
 SalesProcessByGUID 290
 SalesProcesses 291
 SALESPROCESSGUID 51, 93, 110, 112, 115, 119, 122, 125, 290, 292, 293, 332
 salesProcessGUIDs 284
 salesProcessInfo_DefaultProbability 92
 salesProcessInfo_GUID 92
 salesProcessInfo_Icon 92
 salesProcessInfo_IsActive 92
 salesProcessInfo_IsAlsoItemStatus 92
 salesProcessInfo_IsInProgress 92
 salesProcessInfo_IsLost 92
 salesProcessInfo_IsOffer 92
 salesProcessInfo_IsWon 92
 salesProcessInfo_Name 92
 SalesStatusByGUID 292
 SalesStatusesForCase 293
 SALESSTATUSGUID 51, 64, 76, 81, 110, 112, 115, 119, 122, 125, 233, 236, 239, 241, 244, 274, 277, 292, 307, 332
 salesStatusInfo_CaseGUID 93
 salesStatusInfo_GUID 93
 salesStatusInfo_SalesProcessGUID 93
 salesStatusInfo_TimeStamp 93
 SALUTATION 58, 95, 118, 130, 131, 133, 134, 338, 356, 361, 363, 365, 367, 369, 371, 372, 374
 Satisfaction Level 58, 130, 131, 133, 134, 338
 SATISFACTIONLEVEL 58, 130, 131, 133, 134, 338
 Save HTTP Disk Cache 382
 Save HTTP Memory Cache 382
 SELLERCONTACT_SELLERACCOUNTDETAILS 139, 196, 202, 212, 217, 222, 228, 387
 SELLERCONTACT_SELLERCONTACTPERSONNAME 139, 196, 202, 212, 217, 222, 228, 387
 SELLERCONTACT_SELLEREMAILADDRESSIDENTIFIER 139, 196, 202, 212, 217, 222, 228, 387
 SELLERCONTACT_SELLERFAXNUMBER 139, 196, 202, 212, 217, 222, 228, 387
 SELLERCONTACT_SELLERFREETEXT 139, 196, 202, 212, 217, 222, 228, 387
 SELLERCONTACT_SELLERHOMETOWNNAME 139, 196, 202, 212, 217, 222, 228, 387
 SELLERCONTACT_SELLERPHONENUMBER 139, 196, 202, 212, 217, 222, 228, 387
 SELLERCONTACT_SELLERWEBADDRESSIDENTIFIER 139, 196, 202, 212, 217, 222, 228, 387
 SELLERORGANIZATION_SELLEREINVOICEADDRESS 139, 196, 202, 212, 217, 222, 228, 387
 SELLERORGANIZATION_SELLEREINVOICEOPERATOR 139, 196, 202, 212, 217, 222, 228, 387
 SELLERORGANIZATION_SELLERORGANISATIONNAME 139, 196, 202, 212, 217, 222, 228, 387
 SELLERORGANIZATION_SELLERORGANISATIONTAXCODE 139, 196, 202, 212, 217, 222, 228, 387
 SELLERORGANIZATION_SELLERPARTYIDENTIFIER 139, 196, 202, 212, 217, 222, 228, 387
 SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRY 139, 196, 202, 212, 217, 222, 228, 387
 SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_COUNTRYCODE 139, 196, 202, 212, 217, 222, 228, 387
 SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_POSTCODEIDENTIFIER 139, 196, 202, 212, 217, 222, 228, 387
 SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_STREETNAME 139, 196, 202, 212, 217, 222, 228, 387
 SELLERORGANIZATION_SELLERPOSTALADDRESSDETAILS_TOWNNAME 139, 196, 202, 212, 217, 222, 228, 387
 Send Daily Mail 95, 118, 356, 361, 363, 365, 367, 369, 371, 372, 374
 Send Weekly Mail 95, 118, 356, 361, 363, 365, 367, 369, 371, 372, 374
 SENDDAILYMAIL 95, 118, 356, 361, 363, 365, 367, 369, 371, 372, 374
 Sender Addressline 139, 196, 202, 212, 217, 222, 228, 387
 Sender City 139, 196, 202, 212, 217, 222, 228, 387
 Sender Contact Name 139, 196, 202, 212, 217, 222, 228, 387
 Sender Country 139, 196, 202, 212, 217, 222, 228, 387
 Sender Name 139, 196, 202, 212, 217, 222, 228, 387
 Sender Postal Code 139, 196, 202, 212, 217, 222, 228, 387
 Sender State 139, 196, 202, 212, 217, 222, 228, 387
 Sender VAT Number 139, 196, 202, 212, 217, 222, 228, 387
 SENDERADDRESSLINE 139, 196, 202, 212, 217, 222, 228, 387
 SENDERCITY 139, 196, 202, 212, 217, 222, 228, 387

SENDERCONTACTNAME	139, 196, 202, 212, 217, 222, 228, 387	standardize-identifiers-casing	2
SENDERCOUNTRY	139, 196, 202, 212, 217, 222, 228, 387	Start Date	51, 63, 89, 91, 110, 112, 115, 119, 122, 125, 167, 168, 283, 284, 285, 286, 287, 288, 332, 342, 353, 382, 384
SENDERNAME	139, 196, 202, 212, 217, 222, 228, 387	Start Date Time	22, 24, 26, 40, 324
SENDERPOSTALCODE	139, 196, 202, 212, 217, 222, 228, 387	Start Time	70, 179, 181, 184, 186, 189, 191, 207, 267, 270, 272, 300, 301, 302, 343, 384
SENDERSTATE	139, 196, 202, 212, 217, 222, 228, 387	startdate	22, 51, 63, 89, 91, 110, 112, 115, 119, 122, 125, 167, 168, 179, 181, 184, 186, 189, 212, 217, 244, 267, 270, 277, 283, 284, 285, 286, 287, 288, 307, 315, 332, 342, 353, 377
SENDERVATNUMBER	139, 196, 202, 212, 217, 222, 228, 387	startDateTime	19, 21, 22, 24, 26, 40, 101, 108, 125, 134, 151, 256, 262, 266, 288, 302, 310, 324, 374, 384
SENDWEEKLYMAIL	95, 118, 356, 361, 363, 365, 367, 369, 371, 372, 374	startFromIndex	15, 149
sendWelcomeEmail	95	startsAfter	26
Server	294	startsBefore	26
SERVERRESULT	294	STARTTIME	70, 179, 181, 184, 186, 189, 191, 207, 267, 270, 272, 300, 301, 302, 343, 384
Set Invoice Number	294	status	25, 28, 29, 31, 175, 217, 222, 228, 294, 296, 297, 316
Set Invoice Payment Date	295	STRING	252
Set Invoice Status	296	Succesful	382
Set Phase Member List for Phase	296	SUCCESSFUL	382
Set Travel Reimbursement Status	297	Superior User GUID	95, 118, 356, 361, 363, 365, 367, 369, 371, 372, 374
SetInvoiceNumber	294	SUPERIORUSERGUID	95, 118, 356, 361, 363, 365, 367, 369, 371, 372, 374
SetInvoicePaymentDate	295	supervisorUserGUIDs	284
SetInvoiceStatus	296	SYMBOL	146, 147
SetPhaseMemberListForPhase	296	System Name	303, 304
SETPHASEMEMBERLISTFORPHASERESULT	296	SYSTEMNAME	303, 304
SetTravelReimbursementStatus	297		
severa	1		
Shared to Extranet	51, 110, 112, 115, 119, 122, 125, 332, 384		
SHAREDTOEXTRANET	51, 110, 112, 115, 119, 122, 125, 332		
Show Logo on Invoices	49, 104, 106, 107, 108, 330	tag by GUID	298
Show Logo on PDF Reports	49, 104, 106, 107, 108, 330	tagByGUID	298
SHOWLOGOONINVOICES	49, 104, 106, 107, 108, 330	tagGUID	165, 298
SHOWLOGOONPDFREPORTS	49, 104, 106, 107, 108, 330	tagInfo_Context	57, 59, 69, 98
SIZE	173, 174, 311	tagInfo_GUID	57, 59, 69, 98
slot-based-rate-limit-length-ms	2	tagInfo_IsActive	57, 59, 69, 98
slot-based-rate-limit-slots	2	tagInfo_Keyword	57, 59, 69, 98
Social Security Number	95, 118, 356, 361, 363, 365, 367, 369, 371, 372, 374	tagInfo_Weight	57, 59, 69, 98
SOCIALSECURITYNUMBER	95, 118, 356, 361, 363, 365, 367, 369, 371, 372, 374	Tags by Project GUID	299
Sort Order	84, 233, 251, 253, 255, 256, 319, 348	TagsByCaseGUID	299
SORTORDER	84, 233, 251, 253, 255, 256, 319, 348	Task GUID	22, 24, 26, 40, 64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 324, 343, 384
source	75	TASKGUID	22, 24, 26, 40, 64, 70, 76, 81, 115, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 324, 343, 384
standardize-identifiers	2	Tax (%)	265
		Tax Free Description	139, 196, 202, 212, 217, 222, 228, 387

TAXFREEDescription 139, 196, 202, 212, 217, TRAVELENDTIME 64, 76, 81, 233, 236, 239, 241, 222, 228, 387
 TAXPERCENTAGE 265
 text 75
 Time Entries by Date 300
 Time Entries by Date and User GUID 301
 Time Entries Changed since 302
 Time Stamp 93, 292, 293
 TimeEntriesByDate 300
 TimeEntriesByDateAndUserGUID 301
 TimeEntriesChangedSince 302
 Timeout (sec) 382
 TIMEOUT_SEC 382
 TIMESTAMP 93, 292, 293
 Timezone by GUID 303
 Timezone GUID 49, 58, 60, 95, 104, 106, 107, 108, 118, 129, 130, 131, 133, 134, 136, 137, 138, 148, 149, 151, 330, 338, 339, 356, 361, 363, 365, 367, 369, 371, 372, 374
 TimezoneByGUID 303
 TIMEZONEGUID 49, 58, 60, 95, 104, 106, 107, 108, 118, 129, 130, 131, 133, 134, 136, 137, 138, 148, 149, 151, 303, 330, 338, 339, 356, 361, 363, 365, 367, 369, 371, 372, 374
 Timezones 304
 TITLE 63, 95, 118, 139, 167, 168, 196, 202, 212, 217, 222, 228, 342, 356, 361, 363, 365, 367, 369, 371, 372, 374, 387
 Total Hours 377
 TOTALHOURS 377
 Transaction ID 382
 TRANSACTION_ID 382
 Translated Name 136, 137, 138
 TRANSLATEDNAME 136, 137, 138
 Travel End Time 64, 76, 81, 233, 236, 239, 241, 244, 274, 277, 307
 Travel Expense by Code 304
 Travel Expense by Code Include Inactive 305
 Travel Expense by GUID 306
 Travel Expenses by Date 307
 Travel Expenses Changed since 310
 Travel Reimbursement Attachments 311
 Travel Reimbursement by Date 315
 Travel Reimbursement by GUID 312
 Travel Reimbursement by Number 313
 Travel Reimbursement by Status 316
 Travel Reimbursement by Status GUID 318
 Travel Reimbursement PDF 314
 Travel Reimbursement Status 312, 313, 315, 316, 318
 Travel Reimbursement Statuses 319
 Travel Start Time 64, 76, 81, 233, 236, 239, 241, 244, 274, 277, 307
 TravelExpenseByCode 304
 TravelExpenseByCodeIncludeInactive 305
 TravelExpenseByGUID 306
 travelExpenseCode 304, 305
 travelExpenseGUID 165, 306
 travelExpenseInfo_Code 94, 354
 travelExpenseInfo_CostCurrencyGUID 94, 354
 travelExpenseInfo_Description 94, 354
 travelExpenseInfo_ExpenseClass 94, 354
 travelExpenseInfo_GUID 94, 354
 travelExpenseInfo_IsActive 94, 354
 travelExpenseInfo_MeasurementUnit 94, 354
 travelExpenseInfo_Name 94, 354
 travelExpenseInfo_Type 94, 354
 travelExpenseInfo_UnitCost 94, 354
 travelExpenseInfo_UnitPrice 94, 354
 travelExpenseInfo_UselnWorkTimeEntry 94, 354
 travelExpenseInfo_VAT 94, 354
 TravelExpensesByDate 307
 TravelExpensesChangedSince 310
 TravelReimbursementAttachments 311
 TravelReimbursementByGUID 312
 TravelReimbursementByNumber 313
 TravelReimbursementGUID 297, 312
 travelReimbursementNumber 313
 travelReimbursementNumberingBusinessUnitGUID 313
 TravelReimbursementPdf 314
 TRAVELREIMBURSEMENTROWS 312, 313, 315, 316, 318
 TravelReimbursementsByDate 315
 TravelReimbursementsByStatus 316
 TravelReimbursementsByStatusGUID 318
 TRAVELREIMBURSEMENTSTATUS 312, 313, 315, 316, 318
 TravelReimbursementStatuses 319
 travelReimbursementStatusGUID 318
 TRAVELSTARTTIME 64, 76, 81, 233, 236, 239, 241, 244, 274, 277, 307
 TYPE 86, 94, 133, 259, 260, 261, 266, 304, 305, 306, 310, 350, 354

- U -
 Unit Cost 64, 70, 76, 81, 86, 94, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 259, 260, 261, 266, 267, 270, 272, 274, 277, 304, 305, 306, 307, 310, 343, 350, 354, 384

Unit Cost in Invoicing Currency	64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 343, 384	Update Activity Type	326
Unit Cost in Organization Currency	64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 343, 384	Update Address	327
Unit Cost in User Company Currency	64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 343, 384	Update Billing and Expense Forecast	328
Unit Cost VAT Included Amount	64, 76, 81, 233, 236, 239, 241, 244, 274, 277, 307	Update Business Unit	330
Unit Price	64, 70, 76, 81, 86, 94, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 259, 260, 261, 266, 267, 270, 272, 274, 277, 304, 305, 306, 307, 310, 343, 350, 354, 384	Update Contact	338
Unit Price in Base Currency	64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 343, 384	Update Customer	339
Unit Price in Project Company Currency	64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 343, 384	Update Employment	342
Unit Price in User Company Currency	64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 343, 384	Update Hour Entry	343
UNITCOST	64, 70, 76, 81, 86, 94, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 259, 260, 261, 266, 267, 270, 272, 274, 277, 304, 305, 306, 307, 310, 343, 350, 354, 384	Update Phase	348
UNITCOSTININVOICINGCURRENCY	64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 343, 384	Update Product	350
UNITCOSTINORGANIZATIONCURRENCY	64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 343, 384	Update Product Category	351
UNITCOSTINUSERCOMPANYCURRENCY	64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 343, 384	Update Project	332
UNITCOSTVATINCLUDEDAMOUNT	64, 76, 81, 233, 236, 239, 241, 244, 274, 277, 307	Update Project Status	337
UNITPRICE	64, 70, 76, 81, 86, 94, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 259, 260, 261, 266, 267, 270, 272, 274, 277, 304, 305, 306, 307, 310, 343, 350, 354, 384	Update Resource	352
UNITPRICEINBASECURRENCY	64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 343, 384	Update Resource Allocation	353
UNITPRICEINPROJECTCOMPANYCURRENCY	64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 343, 384	Update Travel Expense	354
UNITPRICEINUSERCOMPANYCURRENCY	64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 343, 384	Update User	356
UNITPRICETININVOICINGCURRENCY	64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 343, 384	Update User Rights by User GUID	359
UNITPRICETINORGANIZATIONCURRENCY	64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 343, 384	Update Work Type	360
UNITPRICETINUSERCOMPANYCURRENCY	64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 343, 384	UpdateAccessRightsProfile	319
UNITPRICETVATINCLUDEDAMOUNT	64, 76, 81, 233, 236, 239, 241, 244, 274, 277, 307	UpdateAccount	321
UNITPRICEINBASECURRENCY	64, 70, 76, 81, 86, 94, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 259, 260, 261, 266, 267, 270, 272, 274, 277, 304, 305, 306, 307, 310, 343, 350, 354, 384	UpdateActivity	324
UNITPRICEINCASECOMPANYCURRENCY	64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 343, 384	UpdateActivityType	326
UNITPRICEINUSERCOMPANYCURRENCY	64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 343, 384	UpdateAddress	327
UPDATEACCESSRIGHTSPROFILE	319	UpdateBillingAndExpenseForecast	328
UPDATEACCOUNT	321	UpdateBusinessUnit	330
UPDATEACTIVITY	324	UpdateCase	332
UPDATEACTIVITYTYPE	326	UpdateCaseStatus	337
UPDATEADDRESS	327	UpdateContact	338
UPDATEBILLINGANDEXPENSEFORECAST	328	UpdateCustomer	339
UPDATEBUSINESSUNIT	330	Updated	8, 9, 11, 12, 16, 18, 19, 21, 35, 49, 51, 60, 104, 106, 107, 108, 110, 112, 115, 119, 122, 125, 148, 149, 151, 321, 330, 332, 339, 384
UPDATECASE	332	UPDATEDTS	8, 9, 11, 12, 16, 18, 19, 21, 35, 49, 51, 60, 104, 106, 107, 108, 110, 112, 115, 119, 122, 125, 148, 149, 151, 321, 330, 332, 339
UPDATECASESTATUS	337	UpdateEmployment	342
UPDATECONTACT	338	UpdateHourEntry	343
UPDATECUSTOMER	339	UpdatePhase	348
UPDATED	8, 9, 11, 12, 16, 18, 19, 21, 35, 49, 51, 60, 104, 106, 107, 108, 110, 112, 115, 119, 122, 125, 148, 149, 151, 321, 330, 332, 339	UpdateProduct	350
UPDATEDTS	8, 9, 11, 12, 16, 18, 19, 21, 35, 49, 51, 60, 104, 106, 107, 108, 110, 112, 115, 119, 122, 125, 148, 149, 151, 321, 330, 332, 339	UpdateProductCategory	351
UPDATEEMPLOYMENT	342	UpdateResource	352
UPDATEHOURENTRY	343	UpdateResourceAllocation	353
UPDATEPHASE	348	UpdateTravelExpense	354
UPDATEPRODUCT	350	UpdateUser	356
UPDATEPRODUCTCATEGORY	351	UpdateUserRightsByUserGUID	359
UPDATERESOURCE	352	UpdateWorkType	360
UPDATERESOURCEALLOCATION	353	URL	382
UPDATERELEASE	354	URL to Account	8, 9, 11, 12, 16, 18, 19, 21, 35, 60, 148, 149, 151, 321, 339
UPDATEUSER	356	URL to Project	51, 110, 112, 115, 119, 122, 125, 332, 384
UPDATEUSERRIGHTSBYUSERGUID	359		
UPDATEWORKTYPE	360		

URLTOACCOUNT 8, 9, 11, 12, 16, 18, 19, 21, 35
 60, 148, 149, 151, 321, 339
 URLTOCASE 51, 110, 112, 115, 119, 122, 125, 332, 384
 Use Default Products in Work Time Entry 51, 110, 112, 115, 119, 122, 125, 332, 384
 Use Default Work Types 51, 110, 112, 115, 119, 122, 125, 332, 384
 Use HTTP Disk Cache 382
 Use HTTP Memory Cache 382
 Use in Work Time Entry 86, 94, 259, 260, 261, 266, 304, 305, 306, 310, 350, 354
 USEDEFAULTPRODUCTSINWORKTIMEENTRY 51, 110, 112, 115, 119, 122, 125, 332
 USEDEFAULTWORKTYPES 51, 110, 112, 115, 119, 122, 125, 332
 use-http-disk-cache-read 2
 use-http-disk-cache-write 2
 use-http-memory-cache-read 2
 use-http-memory-cache-write 2
 USEINVOICEQUANTITY 70, 179, 181, 184, 186, 189, 191, 207, 267, 270, 272, 343, 384
 USEINWORKTIMEENTRY 86, 94, 259, 260, 261, 266, 304, 305, 306, 310, 350, 354
 use-metadata-memory-cache 2
 User Business Unit Code 70, 179, 181, 184, 186, 189, 191, 207, 267, 270, 272, 343, 384
 User by Code 361
 User by GUID 363
 User by Name 365
 User Code 64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 343, 384
 User First Name 70, 179, 181, 184, 186, 189, 191, 207, 267, 270, 272, 343, 384
 User GUID 25, 28, 29, 31, 64, 70, 76, 81, 89, 91, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 283, 284, 285, 286, 287, 288, 300, 301, 302, 307, 312, 313, 315, 316, 318, 343, 353, 384
 User Last Name 70, 179, 181, 184, 186, 189, 191, 207, 267, 270, 272, 343, 384
 User Rights by User GUID 366
 User Tags 376
 USERBUSINESSUNITCODE 70, 179, 181, 184, 186, 189, 191, 207, 267, 270, 272, 343, 384
 UserByCode 361
 UserByGUID 363
 UserByName 365
 USERCODE 64, 70, 76, 81, 179, 181, 184, 186, 189, 191, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 307, 343, 361, 384
 use-result-memory-cache 2
 USERFIRSTNAME 70, 179, 181, 184, 186, 189, 191, 207, 267, 270, 272, 343, 384
 USERGUID 25, 26, 28, 29, 31, 32, 45, 63, 64, 70, 76, 81, 89, 91, 98, 166, 168, 169, 179, 181, 184, 186, 189, 191, 193, 207, 233, 236, 239, 241, 244, 267, 270, 272, 274, 277, 282, 283, 284, 285, 286, 287, 288, 300, 301, 302, 307, 312, 313, 315, 316, 318, 343, 353, 363, 377, 384
 userGUIDs 284, 296, 371
 userInfo_Addressline 95, 356
 userInfo_BankAccountNumber 95, 356
 userInfo_BusinessUnitCode 95, 356
 userInfo_BusinessUnitGUID 95, 356
 userInfo_BusinessUnitName 95, 356
 userInfo_Code 95, 356
 userInfo_CountryCode 95, 356
 userInfo_CountryGUID 95, 356
 userInfo_CountryRegionGUID 95, 356
 userInfo_CountryRegionName 95, 356
 userInfo_Culture 95, 356
 userInfo_DefaultActivityTypeGUID 95, 356
 userInfo_Email 95, 356
 userInfo_FirstName 95, 356
 userInfo_GUID 95, 356
 userInfo_IsActive 95, 356
 userInfo_LanguageCode 95, 356
 userInfo_LanguageGUID 95, 356
 userInfo_LastName 95, 356
 userInfo_LicenseType 95, 356
 userInfo_Notes 95, 356
 userInfo_OrganizationGUID 95, 356
 userInfo_Phone 95, 356
 userInfo_PhotoFileGUID 95, 356
 userInfo_PostalCode 95, 356
 userInfo_PostOffice 95, 356
 userInfo_ProfileGUID 95, 356
 userInfo_Salutation 95, 356
 userInfo_SendDailyMail 95, 356
 userInfo_SendWeeklyMail 95, 356
 userInfo_SocialSecurityNumber 95, 356
 userInfo_SuperiorUserGUID 95, 356
 userInfo_TimezoneGUID 95, 356
 userInfo_Title 95, 356
 userInfo_WorktypeGUID 95, 356
 userKeywords 372
 USERLASTNAME 70, 179, 181, 184, 186, 189, 191, 207, 267, 270, 272, 343, 384
 UserRightsByUserGUID 366
 Users 367
 Users by Access Rights Profile GUID 369
 Users by GUIDs 371
 Users by User Keywords 372
 Users Changed since 374
 UsersByAccessRightsProfileGUID 369

UsersByGUIDs 371
 UsersByUserKeywords 372
 UsersChangedSince 374
 userTagGUIDs 284
 UserTags 376

- V -

value 133
 VAT 64, 76, 81, 86, 94, 233, 236, 239, 241, 244, 259, 260, 261, 266, 274, 277, 304, 305, 306, 307, 310, 350, 354
 VAT Number 15, 49, 60, 104, 106, 107, 108, 129, 148, 149, 151, 330, 339
 VATNUMBER 12, 15, 49, 60, 104, 106, 107, 108, 129, 148, 149, 151, 330, 339
 Version 376
 Virtual Bank Barcode 139, 196, 202, 212, 217, 222, 228, 387
 VIRTUALBANKBARCODE 139, 196, 202, 212, 217, 222, 228, 387
 Visma Severa 1, 6, 7, 8, 9, 11, 12, 14, 15, 16, 18, 19, 21, 22, 24, 25, 26, 28, 29, 30, 31, 32, 33, 35, 38, 39, 40, 42, 43, 44, 45, 46, 47, 49, 51, 56, 57, 58, 59, 60, 63, 64, 69, 70, 74, 75, 76, 81, 84, 86, 87, 88, 89, 91, 92, 93, 94, 95, 98, 99, 100, 101, 103, 104, 106, 107, 108, 110, 112, 115, 118, 119, 122, 125, 127, 128, 129, 130, 131, 133, 134, 135, 136, 137, 138, 139, 140, 147, 148, 149, 151, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 173, 174, 175, 176, 177, 178, 179, 181, 184, 186, 189, 191, 193, 194, 195, 196, 202, 207, 209, 210, 211, 212, 217, 222, 228, 233, 236, 239, 241, 244, 247, 248, 249, 250, 251, 252, 253, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 270, 272, 274, 277, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 310, 311, 312, 313, 314, 315, 316, 318, 319, 321, 324, 326, 327, 328, 330, 332, 337, 338, 339, 342, 343, 348, 350, 351, 352, 353, 354, 356, 359, 360, 361, 363, 365, 366, 367, 369, 371, 372, 374, 376, 377, 378, 379, 380, 381, 382, 384, 387
 VismaSevera 1

Work Type GUID 70, 84, 95, 118, 179, 181, 184, 186, 189, 191, 193, 207, 251, 253, 255, 256, 267, 270, 272, 343, 348, 356, 361, 363, 365, 367, 369, 371, 372, 374, 384
 Work Types 380
 Work Types by Phase GUID 380
 Work Types Changed since 381
 Workday Information by Date Range and User GUID 377
 WorkDayInfoByDateRangeAndUserGUID 377
 WORKESTIMATE 84, 251, 253, 255, 256, 348
 WorkTypeByGUID 378
 WorkTypeByHourGUID 379
 WORKTYPECODE 70, 179, 181, 184, 186, 189, 191, 207, 267, 270, 272, 343, 384
 WORKTYPEGUID 70, 84, 95, 118, 179, 181, 184, 186, 189, 191, 193, 207, 251, 253, 255, 256, 267, 270, 272, 343, 348, 356, 361, 363, 365, 367, 369, 371, 372, 374, 378, 384
 workTypeInfo_Code 99, 360
 workTypeInfo_GUID 99, 360
 workTypeInfo_IsActive 99, 360
 workTypeInfo_IsDefault 99, 360
 workTypeInfo_IsProductive 99, 360
 workTypeInfo_Name 99, 360
 workTypeInfo_SalesAccountGUID 99, 360
 WorkTypes 380
 WorkTypesByPhaseGUID 380
 WorkTypesChangedSince 381

- Y -

PEAR 47, 103, 328
 Your Reference 51, 110, 112, 115, 119, 122, 125, 139, 196, 202, 212, 217, 222, 228, 332, 384, 387
 YOURREFERENCE 51, 110, 112, 115, 119, 122, 125, 139, 196, 202, 212, 217, 222, 228, 332, 387

- W -

WEBSITE 60, 129, 148, 149, 151, 339
 WEIGHT 57, 59, 69, 98, 129, 298, 299, 376
 Work Estimate 84, 251, 253, 255, 256, 348
 Work Type by GUID 378
 Work Type by Hour GUID 379
 Work Type Code 70, 179, 181, 184, 186, 189, 191, 207, 267, 270, 272, 343, 384



invantive the **SQL** company

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