

Yuki Data Model

for use with Invantive SQL



Contents

| | | |
|------------|---|-----------|
| 1 | SQL Driver for Yuki API | 1 |
| 2 | SQL Driver Attributes for Yuki API | 2 |
| 3 | Schema: Accounting | 5 |
| 3.1 | Tables | 5 |
| 3.1.1 | BalancesByDate: Yuki Balances by Date | 5 |
| 3.1.2 | CheckOutstandingAdminItemsByReference | 7 |
| 3.1.3 | CheckOutstandingItemsByReference | 8 |
| 3.1.4 | FiscalBalancesByDate: Yuki Fiscal Balances by Date | 10 |
| 3.1.5 | FiscalNetRevenues: Yuki Fiscal Net Revenues | 11 |
| 3.1.6 | FiscalTransactions: Yuki Fiscal Transactions | 12 |
| 3.1.7 | NetRevenues: Yuki Net Revenues | 13 |
| 3.1.8 | OutstandingCreditorItems: Yuki Outstanding Creditor Items | 14 |
| 3.1.9 | OutstandingCreditorItemsByDate: Yuki Outstanding Creditor Items by Date | 16 |
| 3.1.10 | OutstandingDebtorItems: Yuki Outstanding Debtor Items | 18 |
| 3.1.11 | OutstandingDebtorItemsByDate: Yuki Outstanding Debtor Items by Date | 20 |
| 3.1.12 | Transactions: Yuki Transactions | 21 |
| 3.1.13 | TransactionsAndContacts: Yuki Transactions and Contacts | 23 |
| 3.1.14 | YearEndBalancesByDate: Yuki Year-End Balances by Date | 24 |
| 4 | Schema: AccountingInfo | 25 |
| 4.1 | Tables | 25 |
| 4.1.1 | DocumentsByTransaction: Yuki Documents by Transaction | 25 |
| 4.1.2 | FinancialPeriodsByYear | 27 |
| 4.1.3 | GLAccountSchemes: Yuki GL Account Schemes | 28 |
| 4.1.4 | ProjectBalancesByDate: Yuki Project Balances by Date | 29 |
| 4.1.5 | ProjectDetails: Yuki Project Details | 30 |
| 4.1.6 | Projects: Yuki Projects | 31 |
| 4.1.7 | RGSschemes: Yuki RGS Schemes | 33 |
| 4.1.8 | StartBalancesByYear | 34 |
| 4.1.9 | TransactionDetailsByDate: Yuki Transaction Details by Date | 35 |
| 5 | Schema: Archive | 37 |
| 5.1 | Tables | 37 |
| 5.1.1 | CostCategories: Yuki Cost Categories | 37 |
| 5.1.2 | Currencies: Yuki Currencies | 38 |
| 5.1.3 | DocumentBinaryDataById | 38 |
| 5.1.4 | DocumentById | 39 |
| 5.1.5 | DocumentFolderCounts | 41 |
| 5.1.6 | DocumentFolders | 42 |
| 5.1.7 | DocumentFolderTabsByFolderId | 42 |
| 5.1.8 | DocumentImageByDocumentId | 43 |
| 5.1.9 | DocumentImageCountByDocumentId | 44 |
| 5.1.10 | Documents | 45 |
| 5.1.11 | DocumentsByFolderId | 47 |
| 5.1.12 | DocumentsBySearchCriteria | 48 |
| 5.1.13 | DocumentsByTabId | 50 |
| 5.1.14 | Menus | 51 |
| 5.1.15 | PaymentMethods | 52 |

| | | |
|-------------|---|-----------|
| 6 | Schema: Contact | 52 |
| 6.1 | Tables | 52 |
| 6.1.1 | ContactsByDomainId | 52 |
| 7 | Schema: Domains | 53 |
| 7.1 | Tables | 53 |
| 7.1.1 | UsersByDomainId | 53 |
| 8 | Schema: General | 55 |
| 8.1 | Tables | 55 |
| 8.1.1 | Administrations: Yuki Administrations | 55 |
| 8.1.2 | Domains: Yuki Domains | 55 |
| 9 | Schema: Sales | 56 |
| 9.1 | Tables | 56 |
| 9.1.1 | SalesInvoiceSchemas: Yuki Sales Invoice Schemas | 56 |
| 10 | Schema: Views | 57 |
| 10.1 | Views | 57 |
| 10.1.1 | Contacts | 57 |
| 10.1.2 | Users | 58 |
| | Index | 59 |

1 SQL Driver for Yuki API

Invantive SQL is the fastest, easiest and most reliable way to exchange data with the Yuki 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](#). Other users or Invantive Support will try to help you to our best.

Yuki is online accounting software from Visma. Yuki specializes in automated processing of documents in the Netherlands.

Yuki API Clients

Invantive SQL is available on many user interfaces ("clients" in traditional server-client paradigm). All Invantive SQL 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 [Yuki Power BI connector](#) is based on the Invantive SQL driver for Yuki, 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 Yuki API into traditional databases such as SQL Server (on-premise and Azure), MySQL, PostgreSQL and Oracle is possible using [Invantive Data Replicator](#). Invantive Data Replicator automatically creates and maintains Yuki datawarehouses, possibly in combination with data from over 70 other (cloud) platforms. Data Replicator supports data volumes up to over 1 TB and over 5.000 companies. The on-premise edition of Invantive Bridge offers an Yuki ADO.net provider.

Finally, online web apps can be build for Yuki using App Online of [Invantive Cloud](#).

Monitor API Calls

When a query or DML-statement has been executed on Invantive SQL a developer can evaluate the actual calls made to the Yuki 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\NativeLogs folder Invantive SQL will create log files per API request and response.

Specifications

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

An introduction into the concepts of Invantive SQL such as databases, data containers and partitioning can be found in the [Invantive SQL grammar](#).

The configuration can be changed using various attributes during log on and use. A full list of configuration options is listed in the [driver attributes](#)^[2].

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

Changes and bug fixes on the Yuki SQL driver can be found in the [release notes](#). Get access to the Yuki community through the [Yuki section](#) of the Invantive forums.

Driver code for use in settings.xml: Yuki

Alias: yuki

Recommended alias: yki

Partition Label: Company

More technical documentation as provided by the supplier of the Yuki API on the native APIconnection used can be found at <http://help.yuki.nl/documents/homepage/startpagina-yuki-api-documentatie>.

General documentation on Yuki is available at <https://yukeworks.com>

Updated: 06-04-2021 14:39 using Invantive SQL version 20.1.393-BETA+3152.

2 SQL Driver Attributes for Yuki API

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

The Yuki 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.
- Drivers file: the providers.xml file (obsolete starting release 17.32).
- Log on: value to be specified interactively by user during log on in a user interface.

The connection string for Yuki can be found in the settings*.xml file used for the database. Settings*.xml files are typically located in the %USERPROFILE%\invantive folder in most deployment scenarios. The reference manuals contain instructions how to relocate the settings*.xml files. 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 Yuki 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 | Use for analysis only! Enforce rows to be unique. | Shared | False | ✓ | ✓ | ✓ | |
| api-url | URL of Yuki web service | Yuki | | ✓ | | ✓ | |
| 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 | ✓ | ✓ | ✓ | |
| 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\gle3.WS212\Inventive\Cache\http\gle3\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 | ✓ | ✓ | ✓ | |
| http-get-timeout-ms | HTTP GET timeout (ms) | | 300000 | ✓ | ✓ | ✓ | |
| 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-ms | HTTP POST timeout (ms) | | 300000 | ✓ | ✓ | ✓ | |
| ignore-xml-fatal-errors | Ignore fatal errors within the XML returned by the API. | Yuki | False | ✓ | ✓ | ✓ | |

| Code | Description | Origin | Default Value | Set from Connection String | Set from Set SQL-Statement | Set from Drivers File | Set from Log On |
|---|---|---------------|---------------|----------------------------|----------------------------|-----------------------|-----------------|
| ignore-xml-no-access-errors | Ignore no access errors within the XML returned by the API. | Yuki | True | ✓ | ✓ | ✓ | |
| 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-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 | Registers native calls to data container backend as disk files. | Shared | False | ✓ | ✓ | ✓ | |
| log-native-calls-to-trace | Log native calls to data container backend on the trace. | Shared | False | ✓ | ✓ | ✓ | |
| maximum-length-identifiers | Non-default maximum length in characters of identifier names. | Shared | | ✓ | ✓ | ✓ | |
| max-url-length-accepted | The maximum accepted URL length before raising an error. | Shared | 8000 | ✓ | ✓ | ✓ | |
| max-url-length-desired | The maximum desired URL length. | Shared | 8000 | ✓ | ✓ | ✓ | |
| 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 ms 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 | Shared | | ✓ | ✓ | ✓ | |

| Code | Description | Origin | Default Value | Set from Connection String | Set from Set SQL-Statement | Set from Drivers File | Set from Log On |
|---------------------------------|---|--------|---------------|----------------------------|----------------------------|-----------------------|-----------------|
| | effective on limited platforms such as AFAS Online | | | | | | |
| requests-parallel-max | Maximum number of parallel data requests from individual partitions on the data container. | Shared | 32 | ✓ | ✓ | ✓ | |
| result-set-memory-cache | Action: provide 'empty' to empty. | XML | | | ✓ | | |
| slot-based-rate-limit-length-ms | Total length in ms across all slots of a slot-based rate limit. | Shared | 60000 | ✓ | | ✓ | |
| slot-based-rate-limit-slots | Number of slots of a slot-based rate limit. Null means no slot-based rate limit | Shared | | ✓ | | ✓ | |
| standardize-identifiers | Rewrite all identifiers to the preferred standards as configured by standardize-identifiers-casing and maximum-length-identifiers. | Shared | True | ✓ | ✓ | ✓ | |
| standardize-identifiers-casing | Rewrite all identifiers to the recommended standard platform-specific casing when changing a data model on a case-dependent platform. | Shared | True | ✓ | ✓ | ✓ | |
| use-http-disk-cache-read | Whether to use HTTP responses from previous queries stored on disk to answer the current query. | Shared | True | ✓ | ✓ | ✓ | |
| use-http-disk-cache-write | Whether to memorize HTTP responses on disk. | Shared | True | ✓ | ✓ | ✓ | |
| use-http-memory-cache-read | Whether to use HTTP responses from previous queries stored in memory to answer the current query | | True | ✓ | ✓ | ✓ | |
| use-http-memory-cache-write | Whether to memorize HTTP responses in memory | | True | ✓ | ✓ | ✓ | |
| use-metadata-memory-cache | Whether to use the metadata in memory calculated previously Has only practical use during development on a XML provider. | 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: Accounting

3.1 Tables

3.1.1 BalancesByDate: Yuki Balances by Date

Catalog: Yuki

Schema: Accounting

Label: Balances by Date

The data in this table is partitioned per value of the COMPANY_CODE column.

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

Yuki API URL: <https://api.yukiworks.nl/ws/Accounting.asmx>

Topic: GLAccountBalance

Base XPath: /soap11:Envelope/soap11:Body/tns:GLAccountBalanceResponse

XPath: /tns:GLAccountBalanceResult/GLAccountBalance/GLAccount

Table Function Parameters

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

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

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

| Name | Data Type | Required | Default Value | Documentation |
|------------------|-----------|-------------------------------------|---------------|---------------|
| administrationID | string | <input type="checkbox"/> | | |
| transactionDate | datetime | <input checked="" type="checkbox"/> | | |

Table Function Columns

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

| Name | Data Type | Label | Required | Documentation |
|--------------|-------------|----------------|--------------------------|---------------|
| AMOUNT | decimal | Amount | <input type="checkbox"/> | |
| BALANCETYPE | string | Balance Type | <input type="checkbox"/> | |
| CODE | string | Code | <input type="checkbox"/> | |
| COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |
| COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| DESCRIPTION | string | Description | <input type="checkbox"/> | |
| DOMAINID | string(36) | | <input type="checkbox"/> | |
| DOMAINNAME | string(240) | | <input type="checkbox"/> | |

3.1.2 CheckOutstandingAdminItemsByReference

Catalog: Yuki

Schema: Accounting

The data in this table is partitioned per value of the COMPANY_CODE column.

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

Yuki API URL: <https://api.yukiworks.nl/ws/Accounting.asmx>

Topic: CheckOutstandingItemAdmin

Base

XPath: /soap11:Envelope/soap11:Body/tns:CheckOutstandingItemAdminResponse

XPath: /tns:CheckOutstandingItemAdminResult/OutstandingItems/Item

Table Function Parameters

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

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

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

| Name | Data Type | Required | Default Value | Documentation |
|------------------|-----------|-------------------------------------|---------------|---------------|
| administrationID | string | <input type="checkbox"/> | | |
| Reference | string | <input checked="" type="checkbox"/> | | |

Table Function Columns

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

| Name | Data Type | Label | Required | Documentation |
|--------------|------------|----------------|--------------------------|---------------|
| ADDRESSLINE1 | string | | <input type="checkbox"/> | |
| ADDRESSLINE2 | string | Address Line 2 | <input type="checkbox"/> | |
| CITY | string | City | <input type="checkbox"/> | |
| COCNUMBER | string | CoC Number | <input type="checkbox"/> | |
| COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |

| Name | Data Type | Label | Required | Documentation |
|------------------|-------------|-----------------|--------------------------|---------------|
| COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| CONTACTCODE | string | Contact Code | <input type="checkbox"/> | |
| CONTACTID | guid | Contact ID | <input type="checkbox"/> | |
| CONTACTNAME | string | | <input type="checkbox"/> | |
| COUNTRY | string | Country | <input type="checkbox"/> | |
| DATE | datetime | Date | <input type="checkbox"/> | |
| DESCRIPTION | string | Description | <input type="checkbox"/> | |
| DOMAINID | string(36) | | <input type="checkbox"/> | |
| DOMAINNAME | string(240) | | <input type="checkbox"/> | |
| DUEDATE | datetime | Due Date | <input type="checkbox"/> | |
| ID | guid | ID | <input type="checkbox"/> | |
| MAILADDRESSLINE1 | string | | <input type="checkbox"/> | |
| MAILADDRESSLINE2 | string | | <input type="checkbox"/> | |
| MAILCITY | string | Mail City | <input type="checkbox"/> | |
| MAILPOSTCODE | string | Mail ZIP Code | <input type="checkbox"/> | |
| OPENAMOUNT | decimal | Open Amount | <input type="checkbox"/> | |
| ORIGINALAMOUNT | decimal | Original Amount | <input type="checkbox"/> | |
| POSTCODE | string | ZIP Code | <input type="checkbox"/> | |
| REFERENCE | string | Reference | <input type="checkbox"/> | |
| TYPEDESCRIPTION | string | | <input type="checkbox"/> | |
| TYPEID | string | Type ID | <input type="checkbox"/> | |
| VATNUMBER | string | VAT Number | <input type="checkbox"/> | |

3.1.3 CheckOutstandingItemsByReference

Catalog: Yuki

Schema: Accounting

The data in this table is partitioned per value of the COMPANY_CODE column.

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

Yuki API URL: <https://api.yukiworks.nl/ws/Accounting.asmx>

Topic: CheckOutstandingItem

Base

XPath: /soap11:Envelope/soap11:Body/tns:CheckOutstandingItemResponse

XPath: /tns:CheckOutstandingItemResult/CheckOutstandingItems/Item

Table Function Parameters

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

parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

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

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

| Name | Data Type | Required | Default Value | Documentation |
|------------------|-----------|-------------------------------------|---------------|---------------|
| administrationID | string | <input type="checkbox"/> | | |
| Reference | string | <input checked="" type="checkbox"/> | | |

Table Function Columns

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

| Name | Data Type | Label | Required | Documentation |
|------------------|-------------|-----------------|--------------------------|---------------|
| ADDRESSLINE1 | string | | <input type="checkbox"/> | |
| ADDRESSLINE2 | string | Address Line 2 | <input type="checkbox"/> | |
| CITY | string | City | <input type="checkbox"/> | |
| COCONUMBER | string | CoC Number | <input type="checkbox"/> | |
| COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |
| COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| CONTACTCODE | string | Contact Code | <input type="checkbox"/> | |
| CONTACTID | guid | Contact ID | <input type="checkbox"/> | |
| CONTACTNAME | string | | <input type="checkbox"/> | |
| COUNTRY | string | Country | <input type="checkbox"/> | |
| DATE | datetime | Date | <input type="checkbox"/> | |
| DESCRIPTION | string | Description | <input type="checkbox"/> | |
| DOMAINID | string(36) | | <input type="checkbox"/> | |
| DOMAINNAME | string(240) | | <input type="checkbox"/> | |
| DUEDATE | datetime | Due Date | <input type="checkbox"/> | |
| ID | guid | ID | <input type="checkbox"/> | |
| MAILADDRESSLINE1 | string | | <input type="checkbox"/> | |
| MAILADDRESSLINE2 | string | | <input type="checkbox"/> | |
| MAILCITY | string | Mail City | <input type="checkbox"/> | |
| MAILPOSTCODE | string | Mail ZIP Code | <input type="checkbox"/> | |
| OPENAMOUNT | decimal | Open Amount | <input type="checkbox"/> | |
| ORIGINALAMOUNT | decimal | Original Amount | <input type="checkbox"/> | |

| Name | Data Type | Label | Required | Documentation |
|-----------------|-----------|------------|--------------------------|---------------|
| POSTCODE | string | ZIP Code | <input type="checkbox"/> | |
| REFERENCE | string | Reference | <input type="checkbox"/> | |
| TYPEDESCRIPTION | string | | <input type="checkbox"/> | |
| TYPEID | string | Type ID | <input type="checkbox"/> | |
| VATNUMBER | string | VAT Number | <input type="checkbox"/> | |

3.1.4 FiscalBalancesByDate: Yuki Fiscal Balances by Date

Catalog: Yuki

Schema: Accounting

Label: Fiscal Balances by Date

The data in this table is partitioned per value of the COMPANY_CODE column.

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

Yuki API URL: <https://api.yukiworks.nl/ws/Accounting.asmx>

Topic: GLAccountBalanceFiscal

Base

XPath: /soap11:Envelope/soap11:Body/tns:GLAccountBalanceFiscalResponse

XPath: /tns:GLAccountBalanceFiscalResult/GLAccountBalance/GLAccount

Table Function Parameters

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

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

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

| Name | Data Type | Required | Default Value | Documentation |
|------------------|-----------|-------------------------------------|---------------|---------------|
| administrationID | string | <input type="checkbox"/> | | |
| transactionDate | datetime | <input checked="" type="checkbox"/> | | |

Table Function Columns

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

| Name | Data Type | Label | Required | Documentation |
|--------------|-------------|----------------|--------------------------|---------------|
| AMOUNT | decimal | Amount | <input type="checkbox"/> | |
| BALANCETYPE | string | Balance Type | <input type="checkbox"/> | |
| CODE | string | Code | <input type="checkbox"/> | |
| COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |
| COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| DESCRIPTION | string | Description | <input type="checkbox"/> | |
| DOMAINID | string(36) | | <input type="checkbox"/> | |
| DOMAINNAME | string(240) | | <input type="checkbox"/> | |

3.1.5 FiscalNetRevenues: Yuki Fiscal Net Revenues

Catalog: Yuki

Schema: Accounting

Label: Fiscal Net Revenues

The data in this table is partitioned per value of the `COMPANY_CODE` column.

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

Yuki API URL: <https://api.yukiworks.nl/ws/Accounting.asmx>

Topic: `NetRevenueFiscal`

Base XPath: `/soap11:Envelope/soap11:Body/tns:NetRevenueFiscalResponse`

Table Function Parameters

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

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

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

| Name | Data Type | Required | Default Value | Documentation |
|------------------|-----------|--------------------------|---------------|---------------|
| administrationID | string | <input type="checkbox"/> | | |
| EndDate | datetime | <input type="checkbox"/> | | |
| StartDate | datetime | <input type="checkbox"/> | | |

Table Function Columns

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

| Name | Data Type | Label | Required | Documentation |
|------------------------|-------------|---------------------------|--------------------------|---------------|
| COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |
| COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| DOMAINID | string(36) | | <input type="checkbox"/> | |
| DOMAINNAME | string(240) | | <input type="checkbox"/> | |
| NETREVENUEFISCALRESULT | decimal | Net Revenue Fiscal Result | <input type="checkbox"/> | |

3.1.6 FiscalTransactions: Yuki Fiscal Transactions

Catalog: Yuki

Schema: Accounting

Label: Fiscal Transactions

The data in this table is partitioned per value of the `COMPANY_CODE` column.

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

Yuki API URL: <https://api.yukiworks.nl/ws/Accounting.asmx>

Topic: GLAccountTransactionsFiscal

Base

XPath: /soap11:Envelope/soap11:Body/tns:GLAccountTransactionsFiscalResponse

XPath: /tns:GLAccountTransactionsFiscalResult/GLAccountTransactions/GLAccountTransaction

Table Function Parameters

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

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four

parameters will use the default value for the fourth parameter and the specified values for the first three.

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

| Name | Data Type | Required | Default Value | Documentation |
|------------------|-----------|--------------------------|---------------|---------------|
| administrationID | string | <input type="checkbox"/> | | |
| EndDate | datetime | <input type="checkbox"/> | | |
| GLAccountCode | string | <input type="checkbox"/> | | |
| StartDate | datetime | <input type="checkbox"/> | | |

Table Function Columns

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

| Name | Data Type | Label | Required | Documentation |
|--------------------|-------------|-----------------------------|--------------------------|---------------|
| AMOUNT | decimal | Amount | <input type="checkbox"/> | |
| COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |
| COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| CONTACTNAME | string | | <input type="checkbox"/> | |
| DATE | datetime | Date | <input type="checkbox"/> | |
| DESCRIPTION | string | Description | <input type="checkbox"/> | |
| DOMAINID | string(36) | | <input type="checkbox"/> | |
| DOMAINNAME | string(240) | | <input type="checkbox"/> | |
| GLACCOUNTCODE | string | General Ledger Account Code | <input type="checkbox"/> | |
| ID | guid | ID | <input type="checkbox"/> | |
| PROJECTCODE | string | Project Code | <input type="checkbox"/> | |
| PROJECTDESCRIPTION | string | | <input type="checkbox"/> | |

3.1.7 NetRevenues: Yuki Net Revenues

Catalog: Yuki

Schema: Accounting

Label: Net Revenues

The data in this table is partitioned per value of the `COMPANY_CODE` column.

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

Yuki API URL: <https://api.yukiworks.nl/ws/Accounting.asmx>

Topic: NetRevenue

Base XPath: /soap11:Envelope/soap11:Body/tns:NetRevenueResponse

Table Function Parameters

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

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

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

| Name | Data Type | Required | Default Value | Documentation |
|------------------|-----------|--------------------------|---------------|---------------|
| administrationID | string | <input type="checkbox"/> | | |
| EndDate | datetime | <input type="checkbox"/> | | |
| StartDate | datetime | <input type="checkbox"/> | | |

Table Function Columns

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

| Name | Data Type | Label | Required | Documentation |
|------------------|-------------|--------------------|--------------------------|---------------|
| COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |
| COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| DOMAINID | string(36) | | <input type="checkbox"/> | |
| DOMAINTNAME | string(240) | | <input type="checkbox"/> | |
| NETREVENUERESULT | decimal | Net Revenue Result | <input type="checkbox"/> | |

3.1.8 OutstandingCreditorItems: Yuki Outstanding Creditor Items

Catalog: Yuki

Schema: Accounting

Label: Outstanding Creditor Items

The data in this table is partitioned per value of the COMPANY_CODE column.

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

Yuki API URL: <https://api.yukiworks.nl/ws/Accounting.asmx>

Topic: OutstandingCreditorItems

Base

XPath: /soap11:Envelope/soap11:Body/tns:OutstandingCreditorItemsResponse

XPath: /tns:OutstandingCreditorItemsResult/OutstandingCreditorItems/Item

Table Function Parameters

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

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

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

| Name | Data Type | Required | Default Value | Documentation |
|-------------------------|-----------|--------------------------|---------------|---------------|
| administrationID | string | <input type="checkbox"/> | | |
| includeBankTransactions | boolean | <input type="checkbox"/> | | |

Table Function Columns

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

| Name | Data Type | Label | Required | Documentation |
|--------------|-------------|----------------|--------------------------|---------------|
| ADDRESSLINE1 | string | | <input type="checkbox"/> | |
| ADDRESSLINE2 | string | Address Line 2 | <input type="checkbox"/> | |
| CITY | string | City | <input type="checkbox"/> | |
| COCONUMBER | string | CoC Number | <input type="checkbox"/> | |
| COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |
| COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| CONTACT | string | Contact | <input type="checkbox"/> | |
| CONTACTCODE | string | Contact Code | <input type="checkbox"/> | |
| CONTACTID | guid | Contact ID | <input type="checkbox"/> | |
| COUNTRY | string | Country | <input type="checkbox"/> | |
| DATE | datetime | Date | <input type="checkbox"/> | |
| DESCRIPTION | string | Description | <input type="checkbox"/> | |

| Name | Data Type | Label | Required | Documentation |
|------------------|-------------|-----------------|--------------------------|---------------|
| DOMAINID | string(36) | | <input type="checkbox"/> | |
| DOMAINNAME | string(240) | | <input type="checkbox"/> | |
| DUEDATE | datetime | Due Date | <input type="checkbox"/> | |
| EMAILHOME | string | | <input type="checkbox"/> | |
| EMAILWORK | string | | <input type="checkbox"/> | |
| ID | guid | ID | <input type="checkbox"/> | |
| MAILADDRESSLINE1 | string | | <input type="checkbox"/> | |
| MAILADDRESSLINE2 | string | | <input type="checkbox"/> | |
| MAILCITY | string | Mail City | <input type="checkbox"/> | |
| MAILPOSTCODE | string | Mail ZIP Code | <input type="checkbox"/> | |
| OPENAMOUNT | decimal | Open Amount | <input type="checkbox"/> | |
| ORIGINALAMOUNT | decimal | Original Amount | <input type="checkbox"/> | |
| PAYMENTMETHOD | string | | <input type="checkbox"/> | |
| PHONEHOME | string | | <input type="checkbox"/> | |
| PHONENAME | string | | <input type="checkbox"/> | |
| POSTCODE | string | ZIP Code | <input type="checkbox"/> | |
| RECIPIENTEMAIL | string | | <input type="checkbox"/> | |
| REFERENCE | string | Reference | <input type="checkbox"/> | |
| TYPEDESCRIPTION | string | | <input type="checkbox"/> | |
| TYPEID | string | Type ID | <input type="checkbox"/> | |
| VATNUMBER | string | VAT Number | <input type="checkbox"/> | |

3.1.9 OutstandingCreditorItemsByDate: Yuki Outstanding Creditor Items by Date

Catalog: Yuki

Schema: Accounting

Label: Outstanding Creditor Items by Date

The data in this table is partitioned per value of the COMPANY_CODE column.

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

Yuki API URL: <https://api.yukiworks.nl/ws/Accounting.asmx>

Topic: OutstandingCreditorItemsByDate

Base

XPath: /soap11:Envelope/soap11:Body/tns:OutstandingCreditorItemsByDateResponse

XPath: /tns:OutstandingCreditorItemsByDateResult/OutstandingCreditorItems/Item

Table Function Parameters

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

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

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

| Name | Data Type | Required | Default Value | Documentation |
|-------------------------|-----------|-------------------------------------|---------------|---------------|
| administrationID | string | <input type="checkbox"/> | | |
| endDate | datetime | <input checked="" type="checkbox"/> | | |
| includeBankTransactions | boolean | <input type="checkbox"/> | | |
| startDate | datetime | <input checked="" type="checkbox"/> | | |

Table Function Columns

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

| Name | Data Type | Label | Required | Documentation |
|------------------|-------------|----------------|--------------------------|---------------|
| ADDRESSLINE1 | string | | <input type="checkbox"/> | |
| ADDRESSLINE2 | string | Address Line 2 | <input type="checkbox"/> | |
| CITY | string | City | <input type="checkbox"/> | |
| COCONUMBER | string | CoC Number | <input type="checkbox"/> | |
| COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |
| COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| CONTACTCODE | string | Contact Code | <input type="checkbox"/> | |
| CONTACTID | guid | Contact ID | <input type="checkbox"/> | |
| CONTACTNAME | string | | <input type="checkbox"/> | |
| COUNTRY | string | Country | <input type="checkbox"/> | |
| DATE | datetime | Date | <input type="checkbox"/> | |
| DESCRIPTION | string | Description | <input type="checkbox"/> | |
| DOMAINID | string(36) | | <input type="checkbox"/> | |
| DOMAINNAME | string(240) | | <input type="checkbox"/> | |
| DUEDATE | datetime | Due Date | <input type="checkbox"/> | |
| EMAILHOME | string | | <input type="checkbox"/> | |
| EMAILWORK | string | | <input type="checkbox"/> | |
| ID | guid | ID | <input type="checkbox"/> | |
| MAILADDRESSLINE1 | string | | <input type="checkbox"/> | |

| Name | Data Type | Label | Required | Documentation |
|------------------|-----------|-----------------|--------------------------|---------------|
| MAILADDRESSLINE2 | string | | <input type="checkbox"/> | |
| MAILCITY | string | Mail City | <input type="checkbox"/> | |
| MAILPOSTCODE | string | Mail ZIP Code | <input type="checkbox"/> | |
| OPENAMOUNT | decimal | Open Amount | <input type="checkbox"/> | |
| ORIGINALAMOUNT | decimal | Original Amount | <input type="checkbox"/> | |
| PAYMENTMETHOD | string | | <input type="checkbox"/> | |
| PHONEHOME | string | | <input type="checkbox"/> | |
| PHONEWORK | string | | <input type="checkbox"/> | |
| POSTCODE | string | ZIP Code | <input type="checkbox"/> | |
| RECIPIENTEMAIL | string | | <input type="checkbox"/> | |
| REFERENCE | string | Reference | <input type="checkbox"/> | |
| TYPEDESCRIPTION | string | | <input type="checkbox"/> | |
| TYPEID | string | Type ID | <input type="checkbox"/> | |
| VATNUMBER | string | VAT Number | <input type="checkbox"/> | |

3.1.10 OutstandingDebtorItems: Yuki Outstanding Debtor Items

Catalog: Yuki

Schema: Accounting

Label: Outstanding Debtor Items

The data in this table is partitioned per value of the COMPANY_CODE column.

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

Yuki API URL: <https://api.yukiworks.nl/ws/Accounting.asmx>

Topic: OutstandingDebtorItems

Base

XPath: /soap11:Envelope/soap11:Body/tns:OutstandingDebtorItemsResponse

XPath: /tns:OutstandingDebtorItemsResult/OutstandingDebtorItems/Item

Table Function Parameters

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

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

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

| Name | Data Type | Required | Default Value | Documentation |
|-------------------------|-----------|--------------------------|---------------|---------------|
| administrationID | string | <input type="checkbox"/> | | |
| includeBankTransactions | boolean | <input type="checkbox"/> | | |

Table Function Columns

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

| Name | Data Type | Label | Required | Documentation |
|------------------|-------------|-----------------|--------------------------|---------------|
| ADDRESSLINE1 | string | | <input type="checkbox"/> | |
| ADDRESSLINE2 | string | Address Line 2 | <input type="checkbox"/> | |
| CITY | string | City | <input type="checkbox"/> | |
| COCONUMBER | string | CoC Number | <input type="checkbox"/> | |
| COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |
| COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| CONTACTCODE | string | Contact Code | <input type="checkbox"/> | |
| CONTACTID | guid | Contact ID | <input type="checkbox"/> | |
| CONTACTNAME | string | | <input type="checkbox"/> | |
| COUNTRY | string | Country | <input type="checkbox"/> | |
| DATE | datetime | Date | <input type="checkbox"/> | |
| DESCRIPTION | string | Description | <input type="checkbox"/> | |
| DOMAINID | string(36) | | <input type="checkbox"/> | |
| DOMAINNAME | string(240) | | <input type="checkbox"/> | |
| DUEDATE | datetime | Due Date | <input type="checkbox"/> | |
| EMAILHOME | string | | <input type="checkbox"/> | |
| EMAILWORK | string | | <input type="checkbox"/> | |
| ID | guid | ID | <input type="checkbox"/> | |
| MAILADDRESSLINE1 | string | | <input type="checkbox"/> | |
| MAILADDRESSLINE2 | string | | <input type="checkbox"/> | |
| MAILCITY | string | Mail City | <input type="checkbox"/> | |
| MAILPOSTCODE | string | Mail ZIP Code | <input type="checkbox"/> | |
| OPENAMOUNT | decimal | Open Amount | <input type="checkbox"/> | |
| ORIGINALAMOUNT | decimal | Original Amount | <input type="checkbox"/> | |
| PAYMENTMETHOD | string | | <input type="checkbox"/> | |
| PHONEHOME | string | | <input type="checkbox"/> | |
| PHONENAME | string | | <input type="checkbox"/> | |
| POSTCODE | string | ZIP Code | <input type="checkbox"/> | |
| RECIPIENTEMAIL | string | | <input type="checkbox"/> | |

| Name | Data Type | Label | Required | Documentation |
|-----------------|-----------|------------|--------------------------|---------------|
| REFERENCE | string | Reference | <input type="checkbox"/> | |
| TYPEDESCRIPTION | string | | <input type="checkbox"/> | |
| TYPEID | string | Type ID | <input type="checkbox"/> | |
| VATNUMBER | string | VAT Number | <input type="checkbox"/> | |

3.1.11 OutstandingDebtorItemsByDate: Yuki Outstanding Debtor Items by Date

Catalog: Yuki

Schema: Accounting

Label: Outstanding Debtor Items by Date

The data in this table is partitioned per value of the COMPANY_CODE column.

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

Yuki API URL: <https://api.yukiworks.nl/ws/Accounting.asmx>

Topic: OutstandingDebtorItemsByDate

Base

XPath: /soap11:Envelope/soap11:Body/tns:OutstandingDebtorItemsByDate
Response

XPath: /tns:OutstandingDebtorItemsByDateResult/OutstandingDebtorItems/Item

Table Function Parameters

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

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

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

| Name | Data Type | Required | Default Value | Documentation |
|-------------------------|-----------|-------------------------------------|---------------|---------------|
| administrationID | string | <input type="checkbox"/> | | |
| endDate | datetime | <input checked="" type="checkbox"/> | | |
| includeBankTransactions | boolean | <input type="checkbox"/> | | |
| startDate | datetime | <input checked="" type="checkbox"/> | | |

Table Function Columns

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

| Name | Data Type | Label | Required | Documentation |
|------------------|-------------|-----------------|--------------------------|---------------|
| ADDRESSLINE1 | string | | <input type="checkbox"/> | |
| ADDRESSLINE2 | string | Address Line 2 | <input type="checkbox"/> | |
| CITY | string | City | <input type="checkbox"/> | |
| COCONUMBER | string | CoC Number | <input type="checkbox"/> | |
| COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |
| COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| CONTACTCODE | string | Contact Code | <input type="checkbox"/> | |
| CONTACTID | guid | Contact ID | <input type="checkbox"/> | |
| CONTACTNAME | string | | <input type="checkbox"/> | |
| COUNTRY | string | Country | <input type="checkbox"/> | |
| DATE | datetime | Date | <input type="checkbox"/> | |
| DESCRIPTION | string | Description | <input type="checkbox"/> | |
| DOMAINID | string(36) | | <input type="checkbox"/> | |
| DOMAINNAME | string(240) | | <input type="checkbox"/> | |
| DUEDATE | datetime | Due Date | <input type="checkbox"/> | |
| ID | guid | ID | <input type="checkbox"/> | |
| MAILADDRESSLINE1 | string | | <input type="checkbox"/> | |
| MAILADDRESSLINE2 | string | | <input type="checkbox"/> | |
| MAILCITY | string | Mail City | <input type="checkbox"/> | |
| MAILPOSTCODE | string | Mail ZIP Code | <input type="checkbox"/> | |
| OPENAMOUNT | decimal | Open Amount | <input type="checkbox"/> | |
| ORIGINALAMOUNT | decimal | Original Amount | <input type="checkbox"/> | |
| PAYMENTMETHOD | string | | <input type="checkbox"/> | |
| POSTCODE | string | ZIP Code | <input type="checkbox"/> | |
| REFERENCE | string | Reference | <input type="checkbox"/> | |
| TYPEDESCRIPTION | string | | <input type="checkbox"/> | |
| TYPEID | string | Type ID | <input type="checkbox"/> | |
| VATNUMBER | string | VAT Number | <input type="checkbox"/> | |

3.1.12 Transactions: Yuki Transactions

Catalog: Yuki

Schema: Accounting

Label: Transactions

The data in this table is partitioned per value of the `COMPANY_CODE` column.

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

Yuki API URL: <https://api.yukiworks.nl/ws/Accounting.asmx>

Topic: GLAccountTransactions

Base

XPath: /soap11:Envelope/soap11:Body/tns:GLAccountTransactionsResponse

XPath: /tns:GLAccountTransactionsResult/GLAccountTransactions/GLAccountTransaction

Table Function Parameters

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

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

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

| Name | Data Type | Required | Default Value | Documentation |
|------------------|-----------|--------------------------|---------------|---------------|
| administrationID | string | <input type="checkbox"/> | | |
| EndDate | datetime | <input type="checkbox"/> | | |
| GLAccountCode | string | <input type="checkbox"/> | | |
| StartDate | datetime | <input type="checkbox"/> | | |

Table Function Columns

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

| Name | Data Type | Label | Required | Documentation |
|--------------|-------------|----------------|--------------------------|---------------|
| AMOUNT | decimal | Amount | <input type="checkbox"/> | |
| COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |
| COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| CONTACTID | guid | Contact ID | <input type="checkbox"/> | |
| CONTACTNAME | string | | <input type="checkbox"/> | |
| DATE | datetime | Date | <input type="checkbox"/> | |
| DESCRIPTION | string | Description | <input type="checkbox"/> | |

| Name | Data Type | Label | Required | Documentation |
|--------------------|-------------|-----------------------------|--------------------------|---------------|
| DOMAINID | string(36) | | <input type="checkbox"/> | |
| DOMAINNAME | string(240) | | <input type="checkbox"/> | |
| GLACCOUNTCODE | string | General Ledger Account Code | <input type="checkbox"/> | |
| ID | guid | ID | <input type="checkbox"/> | |
| PROJECTCODE | string | Project Code | <input type="checkbox"/> | |
| PROJECTDESCRIPTION | string | | <input type="checkbox"/> | |

3.1.13 TransactionsAndContacts: Yuki Transactions and Contacts

Catalog: Yuki

Schema: Accounting

Label: Transactions and Contacts

The data in this table is partitioned per value of the COMPANY_CODE column.

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

Yuki API URL: <https://api.yukiworks.nl/ws/Accounting.asmx>

Topic: GLAccountTransactionsAndContact

Base

XPath: /soap11:Envelope/soap11:Body/tns:GLAccountTransactionsAndContactResponse

XPath: /tns:GLAccountTransactionsAndContactResult/GLAccountTransactions/GLAccountTransaction

Table Function Parameters

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

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

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

| Name | Data Type | Required | Default Value | Documentation |
|------------------|-----------|--------------------------|---------------|---------------|
| administrationID | string | <input type="checkbox"/> | | |
| EndDate | datetime | <input type="checkbox"/> | | |

| Name | Data Type | Required | Default Value | Documentation |
|---------------|-----------|--------------------------|---------------|---------------|
| GLAccountCode | string | <input type="checkbox"/> | | |
| StartDate | datetime | <input type="checkbox"/> | | |

Table Function Columns

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

| Name | Data Type | Label | Required | Documentation |
|---------------|-------------|-----------------------------|--------------------------|---------------|
| AMOUNT | decimal | Amount | <input type="checkbox"/> | |
| COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |
| COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| CONTACTID | guid | Contact ID | <input type="checkbox"/> | |
| CONTACTNAME | string | | <input type="checkbox"/> | |
| DATE | datetime | Date | <input type="checkbox"/> | |
| DESCRIPTION | string | Description | <input type="checkbox"/> | |
| DOMAINID | string(36) | | <input type="checkbox"/> | |
| DOMAINNAME | string(240) | | <input type="checkbox"/> | |
| FILENAME | string | File Name | <input type="checkbox"/> | |
| GLACCOUNTCODE | string | General Ledger Account Code | <input type="checkbox"/> | |
| ID | guid | ID | <input type="checkbox"/> | |
| PROJECTCODE | string | Project Code | <input type="checkbox"/> | |
| SALESITEM | string | | <input type="checkbox"/> | |

3.1.14 YearEndBalancesByDate: Yuki Year-End Balances by Date

Catalog: Yuki

Schema: Accounting

Label: Year-End Balances by Date

The data in this table is partitioned per value of the `COMPANY_CODE` column.

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

Yuki API URL: <https://api.yukiworks.nl/ws/Accounting.asmx>

Topic: GLAccountBalanceYearEnd

Base

XPath: /soap11:Envelope/soap11:Body/tns:GLAccountBalanceYearEndResponse

XPath: /tns:GLAccountBalanceYearEndResult/GLAccountBalance/GLAccount

Table Function Parameters

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

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

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

| Name | Data Type | Required | Default Value | Documentation |
|------------------|-----------|-------------------------------------|---------------|---------------|
| administrationID | string | <input type="checkbox"/> | | |
| transactionDate | datetime | <input checked="" type="checkbox"/> | | |

Table Function Columns

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

| Name | Data Type | Label | Required | Documentation |
|--------------|-------------|----------------|--------------------------|---------------|
| AMOUNT | decimal | Amount | <input type="checkbox"/> | |
| BALANCETYPE | string | Balance Type | <input type="checkbox"/> | |
| CODE | string | Code | <input type="checkbox"/> | |
| COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |
| COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| DESCRIPTION | string | Description | <input type="checkbox"/> | |
| DOMAINID | string(36) | | <input type="checkbox"/> | |
| DOMAINNAME | string(240) | | <input type="checkbox"/> | |

4 Schema: AccountingInfo

4.1 Tables

4.1.1 DocumentsByTransaction: Yuki Documents by Transaction

Catalog: Yuki

Schema: AccountingInfo

Label: Documents by Transaction

The data in this table is partitioned per value of the COMPANY_CODE column.

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

Yuki API URL: <https://api.yukiworks.nl/ws/AccountingInfo.asmx>

Topic: GetTransactionDocument

Base

XPath: /soap11:Envelope/soap11:Body/tns:GetTransactionDocumentResponse

XPath: /tns:GetTransactionDocumentResult

Table Function Parameters

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

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

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

| Name | Data Type | Required | Default Value | Documentation |
|------------------|-----------|-------------------------------------|---------------|---------------|
| administrationID | string | <input type="checkbox"/> | | |
| transactionID | string | <input checked="" type="checkbox"/> | | |

Table Function Columns

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

| Name | Data Type | Label | Required | Documentation |
|--------------|-------------|----------------|--------------------------|---------------|
| COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |
| COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| DOMAINID | string(36) | | <input type="checkbox"/> | |
| DOMAINNAME | string(240) | | <input type="checkbox"/> | |
| FILEDATA | byte[] | File Data | <input type="checkbox"/> | |
| FILENAME | string | File Name | <input type="checkbox"/> | |

4.1.2 FinancialPeriodsByYear

Catalog: Yuki

Schema: AccountingInfo

The data in this table is partitioned per value of the COMPANY_CODE column.

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

Yuki API URL: <https://api.yukiworks.nl/ws/AccountingInfo.asmx>

Topic: GetPeriodDateTable

Base

XPath: /soap11:Envelope/soap11:Body/tns:GetPeriodDateTableResponse

XPath: /tns:GetPeriodDateTableResult

Table Function Parameters

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

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

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

| Name | Data Type | Required | Default Value | Documentation |
|------------------|-----------|-------------------------------------|---------------|---------------|
| administrationID | string | <input type="checkbox"/> | | |
| yearID | int32 | <input checked="" type="checkbox"/> | | |

Table Function Columns

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

| Name | Data Type | Label | Required | Documentation |
|---------------|-------------|-----------------|--------------------------|---------------|
| COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |
| COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| DOMAINID | string(36) | | <input type="checkbox"/> | |
| DOMAINTNAME | string(240) | | <input type="checkbox"/> | |
| ISO8601PERIOD | boolean | ISO 8601 Period | <input type="checkbox"/> | |
| NAME | string | Name | <input type="checkbox"/> | |

| Name | Data Type | Label | Required | Documentation |
|-------------|-----------|--------------|--------------------------|---------------|
| PERIOD | datetime | Period | <input type="checkbox"/> | |
| WHOLEPERIOD | string | Whole Period | <input type="checkbox"/> | |

4.1.3 GLAccountSchemes: Yuki GL Account Schemes

Catalog: Yuki

Schema: AccountingInfo

Label: GL Account Schemes

The data in this table is partitioned per value of the COMPANY_CODE column.

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

Yuki API URL: <https://api.yukiworks.nl/ws/AccountingInfo.asmx>

Topic: GetGLAccountScheme

Base

XPath: /soap11:Envelope/soap11:Body/tns:GetGLAccountSchemeResponse

XPath: /tns:GetGLAccountSchemeResult/tns:GlAccount

Table Function Parameters

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

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

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

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

Table Function Columns

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

| Name | Data Type | Label | Required | Documentation |
|------|-----------|-------|--------------------------|---------------|
| CODE | string | Code | <input type="checkbox"/> | |

| Name | Data Type | Label | Required | Documentation |
|--------------|-------------|----------------|--------------------------|---------------|
| COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |
| COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| DESCRIPTION | string | Description | <input type="checkbox"/> | |
| DOMAINID | string(36) | | <input type="checkbox"/> | |
| DOMAINNAME | string(240) | | <input type="checkbox"/> | |
| ISENABLED | boolean | Is Enabled | <input type="checkbox"/> | |
| SUBTYPE | string | Subtype | <input type="checkbox"/> | |
| TYPE | string | Type | <input type="checkbox"/> | |

4.1.4 ProjectBalancesByDate: Yuki Project Balances by Date

Catalog: Yuki

Schema: AccountingInfo

Label: Project Balances by Date

The data in this table is partitioned per value of the COMPANY_CODE column.

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

Yuki API URL: <https://api.yukiworks.nl/ws/AccountingInfo.asmx>

Topic: GetProjectBalance

Base

XPath: /soap11:Envelope/soap11:Body/tns:GetProjectBalanceResponse

XPath: /tns:GetProjectBalanceResult/tns:ProjectBalance

Table Function Parameters

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

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

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

| Name | Data Type | Required | Default Value | Documentation |
|------------------|-----------|-------------------------------------|---------------|---------------|
| administrationID | string | <input type="checkbox"/> | | |
| EndDate | datetime | <input checked="" type="checkbox"/> | | |

| Name | Data Type | Required | Default Value | Documentation |
|---------------|-----------|-------------------------------------|---------------|---|
| financialMode | int32 | <input checked="" type="checkbox"/> | | 1 (default) for fiscal, '0' for commercial. |
| GLAccountCode | string | <input type="checkbox"/> | | |
| projectCode | string | <input type="checkbox"/> | | |
| StartDate | datetime | <input checked="" type="checkbox"/> | | |

Table Function Columns

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

| Name | Data Type | Label | Required | Documentation |
|---------------|-------------|-----------------------------|-------------------------------------|---------------|
| AMOUNT | decimal | Amount | <input checked="" type="checkbox"/> | |
| COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |
| COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| DOMAINID | string(36) | | <input type="checkbox"/> | |
| DOMAINNAME | string(240) | | <input type="checkbox"/> | |
| GLACCOUNTCODE | string | General Ledger Account Code | <input checked="" type="checkbox"/> | |
| PROJECTCODE | string | Project Code | <input checked="" type="checkbox"/> | |
| PROJECTNAME | string | | <input type="checkbox"/> | |

4.1.5 ProjectDetails: Yuki Project Details

Catalog: Yuki

Schema: AccountingInfo

Label: Project Details

The data in this table is partitioned per value of the `COMPANY_CODE` column.

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

Yuki API URL: <https://api.yukiworks.nl/ws/AccountingInfo.asmx>

Topic: `GetProjectsAndID`

Base XPath: `/soap11:Envelope/soap11:Body/tns:GetProjectsAndIDResponse`

XPath: `/tns:GetProjectsAndIDResult/tns:Project`

Table Function Parameters

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

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

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

| Name | Data Type | Required | Default Value | Documentation |
|------------------|-----------|--------------------------|---------------|---------------------------------------|
| administrationID | string | <input type="checkbox"/> | | |
| searchOption | string | <input type="checkbox"/> | | All or Company or Code or Description |
| searchValue | string | <input type="checkbox"/> | | |

Table Function Columns

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

| Name | Data Type | Label | Required | Documentation |
|----------------|-------------|-----------------|--------------------------|---------------|
| BUDGETPURCHASE | decimal | Budget Purchase | <input type="checkbox"/> | |
| BUDGETSALES | decimal | Budget Sales | <input type="checkbox"/> | |
| CODE | string | Code | <input type="checkbox"/> | |
| COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |
| COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| COMPANY | string | Company | <input type="checkbox"/> | |
| CONTACT | string | Contact | <input type="checkbox"/> | |
| CONTACTID | guid | Contact ID | <input type="checkbox"/> | |
| DESCRIPTION | string | Description | <input type="checkbox"/> | |
| DOMAINID | string(36) | | <input type="checkbox"/> | |
| DOMAINNAME | string(240) | | <input type="checkbox"/> | |
| ENDDATE | datetime | End Date | <input type="checkbox"/> | |
| HID | int32 | Human ID | <input type="checkbox"/> | |
| ID | string | ID | <input type="checkbox"/> | |
| STARTDATE | datetime | Start Date | <input type="checkbox"/> | |
| TAGS | string | Tags | <input type="checkbox"/> | |

4.1.6 Projects: Yuki Projects

Catalog: Yuki

Schema: AccountingInfo

Label: Projects

The data in this table is partitioned per value of the COMPANY_CODE column.

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

Yuki API URL: <https://api.yukiworks.nl/ws/AccountingInfo.asmx>

Topic: GetProjects

Base XPath: /soap11:Envelope/soap11:Body/tns:GetProjectsResponse

XPath: /tns:GetProjectsResult/tns:Project

Table Function Parameters

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

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

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

| Name | Data Type | Required | Default Value | Documentation |
|------------------|-----------|--------------------------|---------------|---------------------------------------|
| administrationID | string | <input type="checkbox"/> | | |
| searchOption | string | <input type="checkbox"/> | | All or Company or Code or Description |
| searchValue | string | <input type="checkbox"/> | | |

Table Function Columns

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

| Name | Data Type | Label | Required | Documentation |
|----------------|-------------|-----------------|--------------------------|---------------|
| BUDGETPURCHASE | decimal | Budget Purchase | <input type="checkbox"/> | |
| BUDGETSALES | decimal | Budget Sales | <input type="checkbox"/> | |
| CODE | string | Code | <input type="checkbox"/> | |
| COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |
| COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| COMPANY | string | Company | <input type="checkbox"/> | |
| CONTACT | string | Contact | <input type="checkbox"/> | |
| CONTACTID | guid | Contact ID | <input type="checkbox"/> | |
| DESCRIPTION | string | Description | <input type="checkbox"/> | |
| DOMAINID | string(36) | | <input type="checkbox"/> | |

| Name | Data Type | Label | Required | Documentation |
|------------|-------------|------------|--------------------------|---------------|
| DOMAINNAME | string(240) | | <input type="checkbox"/> | |
| ENDDATE | datetime | End Date | <input type="checkbox"/> | |
| HID | int32 | Human ID | <input type="checkbox"/> | |
| ID | string | ID | <input type="checkbox"/> | |
| STARTDATE | datetime | Start Date | <input type="checkbox"/> | |
| TAGS | string | Tags | <input type="checkbox"/> | |

4.1.7 RGSSchemes: Yuki RGS Schemes

Catalog: Yuki

Schema: AccountingInfo

Label: RGS Schemes

The data in this table is partitioned per value of the COMPANY_CODE column.

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

Yuki API URL: <https://api.yukiworks.nl/ws/AccountingInfo.asmx>

Topic: GetRGSScheme

Base XPath: /soap11:Envelope/soap11:Body/tns:GetRGSSchemeResponse

XPath: /tns:GetRGSSchemeResult/tns:RGSEntry

Table Function Parameters

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

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

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

| Name | Data Type | Required | Default Value | Documentation |
|------------------|-----------|--------------------------|---------------|---------------|
| administrationID | string | <input type="checkbox"/> | | |
| rgsVersion | string | <input type="checkbox"/> | | |

Table Function Columns

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

| Name | Data Type | Label | Required | Documentation |
|------------------|-------------|--------------------|--------------------------|---------------|
| COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |
| COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| DOMAINID | string(36) | | <input type="checkbox"/> | |
| DOMAINNAME | string(240) | | <input type="checkbox"/> | |
| RGSDESCRIPTION | string | RGS Description | <input type="checkbox"/> | |
| RGSFLIPCODE | string | RGS Flip Code | <input type="checkbox"/> | |
| RGSREFERENCECODE | string | RGS Reference Code | <input type="checkbox"/> | |
| YUKICODE | string | Code | <input type="checkbox"/> | |
| YUKIDESCRIPION | string | Description | <input type="checkbox"/> | |
| YUKIENABLED | string | Is Enabled | <input type="checkbox"/> | |

4.1.8 StartBalancesByYear

Catalog: Yuki

Schema: AccountingInfo

The data in this table is partitioned per value of the COMPANY_CODE column.

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

Yuki API URL: <https://api.yukiworks.nl/ws/AccountingInfo.asmx>

Topic: GetStartBalanceByGIAccount

Base

XPath: /soap11:Envelope/soap11:Body/tns:GetStartBalanceByGIAccountResponse

XPath: /tns:GetStartBalanceByGIAccountResult/tns:AccountStartBalance

Table Function Parameters

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

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

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

| Name | Data Type | Required | Default Value | Documentation |
|------------------|-----------|-------------------------------------|---------------|---|
| administrationID | string | <input type="checkbox"/> | | |
| bookyear | int32 | <input checked="" type="checkbox"/> | | |
| financialMode | int32 | <input checked="" type="checkbox"/> | | '1' (default) voor Fiscaal, '0' voor Commercieel. |

Table Function Columns

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

| Name | Data Type | Label | Required | Documentation |
|--------------------|-------------|---------------------|--------------------------|---------------|
| ACCOUNTDESCRIPTION | string | Account Description | <input type="checkbox"/> | |
| ACCOUNTID | string | Account ID | <input type="checkbox"/> | |
| COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |
| COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| DOMAINID | string(36) | | <input type="checkbox"/> | |
| DOMAINNAME | string(240) | | <input type="checkbox"/> | |
| STARTBALANCE | string | Start Balance | <input type="checkbox"/> | |

4.1.9 TransactionDetailsByDate: Yuki Transaction Details by Date

Catalog: Yuki

Schema: AccountingInfo

Label: Transaction Details by Date

The data in this table is partitioned per value of the COMPANY_CODE column.

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

Yuki API URL: <https://api.yukiworks.nl/ws/AccountingInfo.asmx>

Topic: GetTransactionDetails

Base

XPath: /soap11:Envelope/soap11:Body/tns:GetTransactionDetailsResponse

XPath: /tns:GetTransactionDetailsResult/tns:TransactionInfo

Table Function Parameters

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

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

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

| Name | Data Type | Required | Default Value | Documentation |
|------------------|-----------|-------------------------------------|---------------|---|
| administrationID | string | <input type="checkbox"/> | | |
| EndDate | datetime | <input checked="" type="checkbox"/> | | |
| financialMode | int32 | <input checked="" type="checkbox"/> | | '1' (default) voor Fiscaal, '0' voor Commercieel. |
| GLAccountCode | string | <input type="checkbox"/> | | |
| StartDate | datetime | <input checked="" type="checkbox"/> | | |

Table Function Columns

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

| Name | Data Type | Label | Required | Documentation |
|-------------------|-------------|-----------------------------|--------------------------|---------------|
| COCONUMBER | string | CoC Number | <input type="checkbox"/> | |
| COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |
| COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| COMPANY | string | Company | <input type="checkbox"/> | |
| CONTACTCOUNTRY | string | Contact Country | <input type="checkbox"/> | |
| CONTACTHID | int32 | Contact HID | <input type="checkbox"/> | |
| CONTACTID | guid | Contact ID | <input type="checkbox"/> | |
| CONTACTZIPCODE | string | Contact ZIP Code | <input type="checkbox"/> | |
| CURRENCY | string | Currency | <input type="checkbox"/> | |
| CURRENCYRATE | decimal | Currency Rate | <input type="checkbox"/> | |
| DESCRIPTION | string | Description | <input type="checkbox"/> | |
| DOCUMENTFOLDER | string | Document Folder | <input type="checkbox"/> | |
| DOCUMENTFOLDERTAB | string | Document Folder Tab | <input type="checkbox"/> | |
| DOCUMENTID | string | Document ID | <input type="checkbox"/> | |
| DOCUMENTREFERENCE | string | Document Reference | <input type="checkbox"/> | |
| DOCUMENTTYPE | string | Document Type | <input type="checkbox"/> | |
| DOMAINID | string(36) | | <input type="checkbox"/> | |
| DOMAINNAME | string(240) | | <input type="checkbox"/> | |
| FULLNAME | string | Full Name | <input type="checkbox"/> | |
| GLACCOUNTCODE | string | General Ledger Account Code | <input type="checkbox"/> | |
| HID | string | Human ID | <input type="checkbox"/> | |

| Name | Data Type | Label | Required | Documentation |
|----------------------------------|-----------|-------------------------------------|--------------------------|--|
| ID | string | ID | <input type="checkbox"/> | |
| MUTATIONUSER | string | Mutation User | <input type="checkbox"/> | |
| PERIODID | string | Period ID | <input type="checkbox"/> | |
| PROJECT | string | Project | <input type="checkbox"/> | Project is identical to the concept 'Dossier'. |
| PROJECTCODE | string | Project Code | <input type="checkbox"/> | |
| TAXCODEDESCRIPTION | string | Tax Code Description | <input type="checkbox"/> | |
| TRANSACTIONAMOUNT | decimal | Transaction Amount | <input type="checkbox"/> | |
| TRANSACTIONAMOUNTFOREIGNCURRENCY | decimal | Transaction Amount Foreign Currency | <input type="checkbox"/> | |
| TRANSACTIONDATE | datetime | Transaction Date | <input type="checkbox"/> | |
| VATNUMBER | string | VAT Number | <input type="checkbox"/> | |

5 Schema: Archive

5.1 Tables

5.1.1 CostCategories: Yuki Cost Categories

Catalog: Yuki

Schema: Archive

Label: Cost Categories

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

Yuki API URL: <https://api.yukiworks.nl/ws/Archive.asmx>

Topic: CostCategories

Base XPath: /soap11:Envelope/soap11:Body

XPath: /tns:CostCategoriesResponse/tns:CostCategoriesResult/CostCategories/CostCategory

Table Columns

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

| Name | Data Type | Label | Required | Documentation |
|--------------|-------------|----------------|--------------------------|---------------|
| COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |
| COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| DESCRIPTION | string | Description | <input type="checkbox"/> | |
| DOMAINID | string(36) | | <input type="checkbox"/> | |
| DOMAINNAME | string(240) | | <input type="checkbox"/> | |
| ID | string | ID | <input type="checkbox"/> | |

5.1.2 Currencies: Yuki Currencies

Catalog: Yuki

Schema: Archive

Label: Currencies

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

Yuki API URL: <https://api.yukiworks.nl/ws/Archive.asmx>

Topic: Currencies

Base XPath: /soap11:Envelope/soap11:Body

XPath: /tns:CurrenciesResponse/tns:CurrenciesResult/Currencies/Currency

Table Columns

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

| Name | Data Type | Label | Required | Documentation |
|--------------|-------------|----------------|--------------------------|---------------|
| COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |
| COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| DEFAULT | boolean | Default | <input type="checkbox"/> | |
| DESCRIPTION | string | Description | <input type="checkbox"/> | |
| DOMAINID | string(36) | | <input type="checkbox"/> | |
| DOMAINNAME | string(240) | | <input type="checkbox"/> | |
| ID | string | ID | <input type="checkbox"/> | |

5.1.3 DocumentBinaryDataById

Catalog: Yuki

Schema: Archive

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

Yuki API URL: <https://api.yukiworks.nl/ws/Archive.asmx>

Topic: DocumentBinaryData

Base XPath: /soap11:Envelope/soap11:Body

XPath: /tns:DocumentBinaryDataResponse

Table Function Parameters

The following parameters can be used to control the behaviour of the table function DocumentBinaryDataById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a

pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

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

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

| Name | Data Type | Required | Default Value | Documentation |
|------------|-----------|-------------------------------------|---------------|---------------|
| documentID | string | <input checked="" type="checkbox"/> | | |

Table Function Columns

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

| Name | Data Type | Label | Required | Documentation |
|--------------------------|-------------|----------------|--------------------------|---------------|
| COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |
| COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| DOCUMENTBINARYDATARESULT | byte[] | | <input type="checkbox"/> | |
| DOMAINID | string(36) | | <input type="checkbox"/> | |
| DOMAINNAME | string(240) | | <input type="checkbox"/> | |

5.1.4 DocumentById

Catalog: Yuki

Schema: Archive

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

Yuki API URL: <https://api.yukiworks.nl/ws/Archive.asmx>

Topic: FindDocument

Base XPath: /soap11:Envelope/soap11:Body/tns:FindDocumentResponse

XPath: /tns:FindDocumentResult/Documents/Document

Table Function Parameters

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

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

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

| Name | Data Type | Required | Default Value | Documentation |
|------------|-----------|-------------------------------------|---------------|---------------|
| documentID | string | <input checked="" type="checkbox"/> | | |

Table Function Columns

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

| Name | Data Type | Label | Required | Documentation |
|-------------------|-------------|----------------|--------------------------|---------------|
| AMOUNT | decimal | Amount | <input type="checkbox"/> | |
| COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |
| COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| CONTACTNAME | string | | <input type="checkbox"/> | |
| CONTENTTYPE | string | | <input type="checkbox"/> | |
| CREATED | datetime | | <input type="checkbox"/> | |
| CREATOR | string | | <input type="checkbox"/> | |
| DOCUMENTDATE | datetime | | <input type="checkbox"/> | |
| DOMAINID | string(36) | | <input type="checkbox"/> | |
| DOMAINNAME | string(240) | | <input type="checkbox"/> | |
| FILENAME | string | File Name | <input type="checkbox"/> | |
| FILESIZE | int32 | | <input type="checkbox"/> | |
| FOLDERDESCRIPTION | string | | <input type="checkbox"/> | |
| FOLDERID | string | | <input type="checkbox"/> | |
| ID | string | ID | <input type="checkbox"/> | |
| MODIFIED | datetime | | <input type="checkbox"/> | |
| MODIFIER | string | | <input type="checkbox"/> | |
| SUBJECT | string | | <input type="checkbox"/> | |
| TABDESCRIPTION | string | | <input type="checkbox"/> | |
| TABID | string | | <input type="checkbox"/> | |
| TYPE | string | Type | <input type="checkbox"/> | |
| TYPEDESCRIPTION | string | | <input type="checkbox"/> | |

5.1.5 DocumentFolderCounts

Catalog: Yuki

Schema: Archive

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

Yuki API URL: <https://api.yukiworks.nl/ws/Archive.asmx>

Topic: DocumentFolderCounts

Base

XPath: /soap11:Envelope/soap11:Body/tns:DocumentFolderCountsResponse

XPath: /tns:DocumentFolderCountsResult/DocumentFolders/DocumentFolder

Table Function Parameters

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

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

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

| Name | Data Type | Required | Default Value | Documentation |
|------|-----------|--------------------------|---------------|---------------|
| year | int32 | <input type="checkbox"/> | | |

Table Function Columns

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

| Name | Data Type | Label | Required | Documentation |
|---------------|-------------|----------------|--------------------------|---------------|
| COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |
| COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| DOCUMENTCOUNT | int32 | | <input type="checkbox"/> | |
| DOMAINID | string(36) | | <input type="checkbox"/> | |
| DOMAINNAME | string(240) | | <input type="checkbox"/> | |
| ID | string | ID | <input type="checkbox"/> | |

5.1.6 DocumentFolders

Catalog: Yuki

Schema: Archive

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

Yuki API URL: <https://api.yukiworks.nl/ws/Archive.asmx>

Topic: DocumentFolders

Base XPath: /soap11:Envelope/soap11:Body/tns:DocumentFoldersResponse

XPath: /tns:DocumentFoldersResult/DocumentFolders/DocumentFolder

Table Function Columns

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

| Name | Data Type | Label | Required | Documentation |
|-----------------|-------------|----------------|--------------------------|---------------|
| COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |
| COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| DESCRIPTION | string | Description | <input type="checkbox"/> | |
| DOMAINID | string(36) | | <input type="checkbox"/> | |
| DOMAINNAME | string(240) | | <input type="checkbox"/> | |
| ICON | string | | <input type="checkbox"/> | |
| ID | string | ID | <input type="checkbox"/> | |
| PROCESSEDBYYUKI | boolean | | <input type="checkbox"/> | |

5.1.7 DocumentFolderTabsByFolderId

Catalog: Yuki

Schema: Archive

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

Yuki API URL: <https://api.yukiworks.nl/ws/Archive.asmx>

Topic: DocumentFolderTabs

Base

XPath: /soap11:Envelope/soap11:Body/tns:DocumentFolderTabsResponse

XPath: /tns:DocumentFolderTabsResult/DocumentFolderTabs/DocumentFolderTab

Table Function Parameters

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

parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

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

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

| Name | Data Type | Required | Default Value | Documentation |
|----------|-----------|-------------------------------------|---------------|---------------|
| folderID | string | <input checked="" type="checkbox"/> | | |

Table Function Columns

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

| Name | Data Type | Label | Required | Documentation |
|-----------------|-------------|----------------|--------------------------|---------------|
| COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |
| COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| DESCRIPTION | string | Description | <input type="checkbox"/> | |
| DOMAINID | string(36) | | <input type="checkbox"/> | |
| DOMAINNAME | string(240) | | <input type="checkbox"/> | |
| ID | string | ID | <input type="checkbox"/> | |
| PROCESSEDBYYUKI | boolean | | <input type="checkbox"/> | |

5.1.8 DocumentImageByDocumentId

Catalog: Yuki

Schema: Archive

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

Yuki API URL: <https://api.yukiworks.nl/ws/Archive.asmx>

Topic: DocumentImage

Base XPath: /soap11:Envelope/soap11:Body/tns:DocumentImageResponse

Table Function Parameters

The following parameters can be used to control the behaviour of the table function DocumentImageByDocumentId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the

execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

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

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

| Name | Data Type | Required | Default Value | Documentation |
|------------|-----------|-------------------------------------|---------------|---------------|
| documentID | string | <input checked="" type="checkbox"/> | | |
| maxHeight | int32 | <input checked="" type="checkbox"/> | | |
| maxWidth | int32 | <input checked="" type="checkbox"/> | | |

Table Function Columns

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

| Name | Data Type | Label | Required | Documentation |
|---------------------|-------------|----------------|--------------------------|---------------|
| COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |
| COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| DOCUMENTIMAGERESULT | byte[] | | <input type="checkbox"/> | |
| DOMAINID | string(36) | | <input type="checkbox"/> | |
| DOMAINNAME | string(240) | | <input type="checkbox"/> | |

5.1.9 DocumentImageCountByDocumentId

Catalog: Yuki

Schema: Archive

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

Yuki API URL: <https://api.yukiworks.nl/ws/Archive.asmx>

Topic: DocumentImageCount

Base

XPath: /soap11:Envelope/soap11:Body/tns:DocumentImageCountResponse

Table Function Parameters

The following parameters can be used to control the behaviour of the table function DocumentImageCountByDocumentId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the

execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

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

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

| Name | Data Type | Required | Default Value | Documentation |
|------------|-----------|-------------------------------------|---------------|---------------|
| documentID | string | <input checked="" type="checkbox"/> | | |

Table Function Columns

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

| Name | Data Type | Label | Required | Documentation |
|--------------------------|-------------|----------------|--------------------------|---------------|
| COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |
| COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| DOCUMENTIMAGECOUNTRESULT | int32 | | <input type="checkbox"/> | |
| DOMAINID | string(36) | | <input type="checkbox"/> | |
| DOMAINNAME | string(240) | | <input type="checkbox"/> | |

5.1.10 Documents

Catalog: Yuki

Schema: Archive

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

Yuki API URL: <https://api.yukiworks.nl/ws/Archive.asmx>

Topic: Documents

Base XPath: /soap11:Envelope/soap11:Body/tns:DocumentsResponse

XPath: /tns:DocumentsResult/Documents/Document

Table Function Parameters

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

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

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the 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"/> | | |
| startDate | datetime | <input type="checkbox"/> | | |

Table Function Columns

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

| Name | Data Type | Label | Required | Documentation |
|-------------------|-------------|----------------|--------------------------|---------------|
| AMOUNT | decimal | Amount | <input type="checkbox"/> | |
| COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |
| COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| CONTACTNAME | string | | <input type="checkbox"/> | |
| CONTENTTYPE | string | | <input type="checkbox"/> | |
| CREATED | datetime | | <input type="checkbox"/> | |
| CREATOR | string | | <input type="checkbox"/> | |
| DOCUMENTDATE | datetime | | <input type="checkbox"/> | |
| DOMAINID | string(36) | | <input type="checkbox"/> | |
| DOMAINNAME | string(240) | | <input type="checkbox"/> | |
| FILENAME | string | File Name | <input type="checkbox"/> | |
| FILESIZE | int32 | | <input type="checkbox"/> | |
| FOLDERDESCRIPTION | string | | <input type="checkbox"/> | |
| FOLDERID | string | | <input type="checkbox"/> | |
| ID | string | ID | <input type="checkbox"/> | |
| MODIFIED | datetime | | <input type="checkbox"/> | |
| MODIFIER | string | | <input type="checkbox"/> | |
| SUBJECT | string | | <input type="checkbox"/> | |
| TABDESCRIPTION | string | | <input type="checkbox"/> | |
| TABID | string | | <input type="checkbox"/> | |
| TYPE | string | Type | <input type="checkbox"/> | |
| TYPEDESCRIPTION | string | | <input type="checkbox"/> | |

5.1.11 DocumentsByFolderId

Catalog: Yuki

Schema: Archive

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

Yuki API URL: <https://api.yukiworks.nl/ws/Archive.asmx>

Topic: DocumentsInFolder

Base

XPath: /soap11:Envelope/soap11:Body/tns:DocumentsInFolderResponse

XPath: /tns:DocumentsInFolderResult/Documents/Document

Table Function Parameters

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

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

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the 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"/> | | |
| folderID | int32 | <input checked="" type="checkbox"/> | | |
| startDate | datetime | <input type="checkbox"/> | | |

Table Function Columns

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

| Name | Data Type | Label | Required | Documentation |
|--------------|-------------|----------------|--------------------------|---------------|
| AMOUNT | decimal | Amount | <input type="checkbox"/> | |
| COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |
| COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| CONTACTNAME | string | | <input type="checkbox"/> | |
| CONTENTTYPE | string | | <input type="checkbox"/> | |
| CREATED | datetime | | <input type="checkbox"/> | |

| Name | Data Type | Label | Required | Documentation |
|-------------------|-------------|-----------|--------------------------|---------------|
| CREATOR | string | | <input type="checkbox"/> | |
| DOCUMENTDATE | datetime | | <input type="checkbox"/> | |
| DOMAINID | string(36) | | <input type="checkbox"/> | |
| DOMAINNAME | string(240) | | <input type="checkbox"/> | |
| FILENAME | string | File Name | <input type="checkbox"/> | |
| FILESIZE | int32 | | <input type="checkbox"/> | |
| FOLDERDESCRIPTION | string | | <input type="checkbox"/> | |
| FOLDERID | string | | <input type="checkbox"/> | |
| ID | string | ID | <input type="checkbox"/> | |
| MODIFIED | datetime | | <input type="checkbox"/> | |
| MODIFIER | string | | <input type="checkbox"/> | |
| SUBJECT | string | | <input type="checkbox"/> | |
| TABDESCRIPTION | string | | <input type="checkbox"/> | |
| TABID | string | | <input type="checkbox"/> | |
| TYPE | string | Type | <input type="checkbox"/> | |
| TYPEDESCRIPTION | string | | <input type="checkbox"/> | |

5.1.12 DocumentsBySearchCriteria

Catalog: Yuki

Schema: Archive

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

Yuki API URL: <https://api.yukiworks.nl/ws/Archive.asmx>

Topic: SearchDocuments

Base XPath: /soap11:Envelope/soap11:Body/tns:SearchDocumentsResponse

XPath: /tns:SearchDocumentsResult/Documents/Document

Table Function Parameters

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

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

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

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

| Name | Data Type | Required | Default Value | Documentation |
|--------------|-----------|--------------------------|---------------|---|
| documentID | string | <input type="checkbox"/> | | |
| endDate | datetime | <input type="checkbox"/> | | |
| folderID | int32 | <input type="checkbox"/> | | |
| searchOption | string | <input type="checkbox"/> | | All or Creator or Contact or Subject or Tag or Type |
| searchText | string | <input type="checkbox"/> | | |
| startDate | datetime | <input type="checkbox"/> | | |
| tabID | int32 | <input type="checkbox"/> | | |

Table Function Columns

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

| Name | Data Type | Label | Required | Documentation |
|-------------------|-------------|----------------|--------------------------|---------------|
| AMOUNT | decimal | Amount | <input type="checkbox"/> | |
| COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |
| COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| CONTACTNAME | string | | <input type="checkbox"/> | |
| CONTENTTYPE | string | | <input type="checkbox"/> | |
| CREATED | datetime | | <input type="checkbox"/> | |
| CREATOR | string | | <input type="checkbox"/> | |
| DOCUMENTDATE | datetime | | <input type="checkbox"/> | |
| DOMAINID | string(36) | | <input type="checkbox"/> | |
| DOMAINNAME | string(240) | | <input type="checkbox"/> | |
| FILENAME | string | File Name | <input type="checkbox"/> | |
| FILESIZE | int32 | | <input type="checkbox"/> | |
| FOLDERDESCRIPTION | string | | <input type="checkbox"/> | |
| FOLDERID | string | | <input type="checkbox"/> | |
| ID | string | ID | <input type="checkbox"/> | |
| MODIFIED | datetime | | <input type="checkbox"/> | |
| MODIFIER | string | | <input type="checkbox"/> | |
| SUBJECT | string | | <input type="checkbox"/> | |
| TABDESCRIPTION | string | | <input type="checkbox"/> | |
| TABID | string | | <input type="checkbox"/> | |
| TYPE | string | Type | <input type="checkbox"/> | |
| TYPEDESCRIPTION | string | | <input type="checkbox"/> | |

5.1.13 DocumentsByTabId

Catalog: Yuki

Schema: Archive

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

Yuki API URL: <https://api.yukiworks.nl/ws/Archive.asmx>

Topic: DocumentsInTab

Base XPath: /soap11:Envelope/soap11:Body/tns:DocumentsInTabResponse

XPath: /tns:DocumentsInTabResult/Documents/Document

Table Function Parameters

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

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

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the 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"/> | | |
| startDate | datetime | <input type="checkbox"/> | | |
| tabID | int32 | <input checked="" type="checkbox"/> | | |

Table Function Columns

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

| Name | Data Type | Label | Required | Documentation |
|--------------|-------------|----------------|--------------------------|---------------|
| AMOUNT | decimal | Amount | <input type="checkbox"/> | |
| COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |
| COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| CONTACTNAME | string | | <input type="checkbox"/> | |
| CONTENTTYPE | string | | <input type="checkbox"/> | |
| CREATED | datetime | | <input type="checkbox"/> | |
| CREATOR | string | | <input type="checkbox"/> | |

| Name | Data Type | Label | Required | Documentation |
|-------------------|-------------|-----------|--------------------------|---------------|
| DOCUMENTDATE | datetime | | <input type="checkbox"/> | |
| DOMAINID | string(36) | | <input type="checkbox"/> | |
| DOMAINNAME | string(240) | | <input type="checkbox"/> | |
| FILENAME | string | File Name | <input type="checkbox"/> | |
| FILESIZE | int32 | | <input type="checkbox"/> | |
| FOLDERDESCRIPTION | string | | <input type="checkbox"/> | |
| FOLDERID | string | | <input type="checkbox"/> | |
| ID | string | ID | <input type="checkbox"/> | |
| MODIFIED | datetime | | <input type="checkbox"/> | |
| MODIFIER | string | | <input type="checkbox"/> | |
| SUBJECT | string | | <input type="checkbox"/> | |
| TABDESCRIPTION | string | | <input type="checkbox"/> | |
| TABID | string | | <input type="checkbox"/> | |
| TYPE | string | Type | <input type="checkbox"/> | |
| TYPEDESCRIPTION | string | | <input type="checkbox"/> | |

5.1.14 Menus

Catalog: Yuki

Schema: Archive

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

Yuki API URL: <https://api.yukiworks.nl/ws/Archive.asmx>

Topic: Menu

Base XPath: /soap11:Envelope/soap11:Body/tns:MenuResponse

XPath: /tns:MenuResult/Menu/MenuEntry

Table Columns

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

| Name | Data Type | Label | Required | Documentation |
|--------------|-------------|----------------|--------------------------|---------------|
| ALERT | int32 | | <input type="checkbox"/> | |
| COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |
| COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| DOMAINID | string(36) | | <input type="checkbox"/> | |
| DOMAINNAME | string(240) | | <input type="checkbox"/> | |
| ICON | string | | <input type="checkbox"/> | |
| ID | string | ID | <input type="checkbox"/> | |
| LINK | string | | <input type="checkbox"/> | |

| Name | Data Type | Label | Required | Documentation |
|------|-----------|-------|--------------------------|---------------|
| TEXT | string | | <input type="checkbox"/> | |

5.1.15 PaymentMethods

Catalog: Yuki

Schema: Archive

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

Yuki API URL: <https://api.yukiworks.nl/ws/Archive.asmx>

Topic: PaymentMethods

Base XPath: /soap11:Envelope/soap11:Body/tns:PaymentMethodsResponse

XPath: /tns:PaymentMethodsResult/PaymentMethods/PaymentMethod

Table Columns

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

| Name | Data Type | Label | Required | Documentation |
|--------------|-------------|----------------|--------------------------|---------------|
| COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |
| COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| DESCRIPTION | string | Description | <input type="checkbox"/> | |
| DOMAINID | string(36) | | <input type="checkbox"/> | |
| DOMAINTNAME | string(240) | | <input type="checkbox"/> | |
| ID | string | ID | <input type="checkbox"/> | |

6 Schema: Contact

6.1 Tables

6.1.1 ContactsByDomainId

Catalog: Yuki

Schema: Contact

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

Yuki API URL: <https://api.yukiworks.nl/ws/Contact.asmx>

Topic: SearchContacts

Base XPath: /soap11:Envelope/soap11:Body/tns:SearchContactsResponse

XPath: /tns:SearchContactsResult/Contacts/Contact

Table Function Parameters

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

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

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

| Name | Data Type | Required | Default Value | Documentation |
|---------------|-----------|-------------------------------------|---------------|--|
| active | string | <input type="checkbox"/> | | Both or Active or Inactive |
| domainId | string | <input checked="" type="checkbox"/> | | |
| modifiedAfter | datetime | <input type="checkbox"/> | | |
| searchOption | string | <input type="checkbox"/> | | All or Name or City or Postcode or Tag or Email or Website or Phone or Code or CoCNumber or VATNumber or BankAccount or ID |
| searchValue | string | <input type="checkbox"/> | | |

Table Function Columns

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

| Name | Data Type | Label | Required | Documentation |
|--------------|-------------|----------------|--------------------------|---------------|
| COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |
| COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| DOMAINID | string(36) | | <input type="checkbox"/> | |
| DOMAINNAME | string(240) | | <input type="checkbox"/> | |
| NAME | string | Name | <input type="checkbox"/> | |

7 Schema: Domains

7.1 Tables

7.1.1 UsersByDomainId

Catalog: Yuki

Schema: Domains

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

Yuki API URL: <https://api.yukiworks.nl/ws/Domains.asmx>

Topic: GetDomainUsers

Base XPath: /soap11:Envelope/soap11:Body/tns:GetDomainUsersResponse

XPath: /tns:GetDomainUsersResult/Users/User

Table Function Parameters

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

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

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

| Name | Data Type | Required | Default Value | Documentation |
|--------|-----------|-------------------------------------|---------------|---------------|
| domain | string | <input checked="" type="checkbox"/> | | |

Table Function Columns

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

| Name | Data Type | Label | Required | Documentation |
|--------------|-------------|----------------|--------------------------|---------------|
| COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |
| COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| DOMAINID | string(36) | | <input type="checkbox"/> | |
| DOMAINNAME | string(240) | | <input type="checkbox"/> | |
| ID | guid | ID | <input type="checkbox"/> | |
| LASTLOGIN | datetime | | <input type="checkbox"/> | |
| LOGIN | string | | <input type="checkbox"/> | |
| NAME | string | Name | <input type="checkbox"/> | |

8 Schema: General

8.1 Tables

8.1.1 Administrations: Yuki Administrations

Catalog: Yuki

Schema: General

Label: Administrations

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

Yuki API URL: <https://api.yukiworks.nl/ws/Accounting.asmx>

Topic: Administrations

Base XPath: /soap11:Envelope/soap11:Body/tns:AdministrationsResponse

XPath: /tns:AdministrationsResult/Administrations/Administration

Table Columns

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

| Name | Data Type | Label | Required | Documentation |
|--------------|-------------|----------------|--------------------------|---------------|
| ADDRESSLINE | string | Address Line | <input type="checkbox"/> | |
| CITY | string | City | <input type="checkbox"/> | |
| COCONUMBER | string | CoC Number | <input type="checkbox"/> | |
| COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |
| COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| COUNTRY | string | Country | <input type="checkbox"/> | |
| DOMAINID | string(36) | | <input type="checkbox"/> | |
| DOMAINTNAME | string(240) | | <input type="checkbox"/> | |
| ID | guid | ID | <input type="checkbox"/> | |
| NAME | string | Name | <input type="checkbox"/> | |
| POSTCODE | string | ZIP Code | <input type="checkbox"/> | |
| STARTDATE | datetime | Start Date | <input type="checkbox"/> | |
| VATNUMBER | string | VAT Number | <input type="checkbox"/> | |

8.1.2 Domains: Yuki Domains

Catalog: Yuki

Schema: General

Label: Domains

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

Yuki API URL: <https://api.yukiworks.nl/ws/Accounting.asmx>

Topic: Domains

Base XPath: /soap11:Envelope/soap11:Body/tns:DomainsResponse

XPath: /tns:DomainsResult/Domains/Domain

Table Columns

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

| Name | Data Type | Label | Required | Documentation |
|--------------|-------------|----------------|--------------------------|---------------|
| COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |
| COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| ID | guid | ID | <input type="checkbox"/> | |
| NAME | string | Name | <input type="checkbox"/> | |
| URL | string | URL | <input type="checkbox"/> | |

9 Schema: Sales

9.1 Tables

9.1.1 SalesInvoiceSchemas: Yuki Sales Invoice Schemas

Catalog: Yuki

Schema: Sales

Label: Sales Invoice Schemas

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

Yuki API URL: <https://api.yukiworks.nl/ws/Sales.asmx>

Topic: SalesInvoiceSchemaPath

Base XPath: /soap11:Envelope/soap11:Body

XPath: /tns:SalesInvoiceSchemaPathResponse

Table Function Parameters

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

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

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

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

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

Table Function Columns

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

| Name | Data Type | Label | Required | Documentation |
|-------------------------------|-------------|----------------|--------------------------|---------------|
| COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |
| COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| DOMAINID | string(36) | | <input type="checkbox"/> | |
| DOMAINTNAME | string(240) | | <input type="checkbox"/> | |
| SALESINVOICESCHEMA PATHRESULT | string | Result | <input type="checkbox"/> | |

10 Schema: Views

10.1 Views

10.1.1 Contacts

Catalog: Yuki

Schema: Views

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

View Columns

The columns of the view Contacts are shown below. Each column has an SQL data type.

| Name | Data Type | Label | Required | Documentation |
|---------------------|-------------|----------------|--------------------------|---------------|
| COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |
| COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| domain_COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |
| domain_COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| domain_ID | guid | ID | <input type="checkbox"/> | |
| domain_NAME | string | Name | <input type="checkbox"/> | |
| domain_URL | string | URL | <input type="checkbox"/> | |
| NAME | string | Name | <input type="checkbox"/> | |

10.1.2 Users

Catalog: Yuki

Schema: Views

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

View Columns

The columns of the view `Users` are shown below. Each column has an SQL data type.

| Name | Data Type | Label | Required | Documentation |
|---------------------|-------------|----------------|--------------------------|---------------|
| COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |
| COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| domain_COMPANY_CODE | string(36) | Partition Code | <input type="checkbox"/> | |
| domain_COMPANY_NAME | string(240) | Partition Name | <input type="checkbox"/> | |
| domain_ID | guid | ID | <input type="checkbox"/> | |
| domain_NAME | string | Name | <input type="checkbox"/> | |
| domain_URL | string | URL | <input type="checkbox"/> | |
| ID | guid | ID | <input type="checkbox"/> | |
| LASTLOGIN | datetime | | <input type="checkbox"/> | |
| LOGIN | string | | <input type="checkbox"/> | |
| NAME | string | Name | <input type="checkbox"/> | |

Index

- A -

Account Description 34
 Account ID 34
 ACCOUNTDESCRIPTION 34
 ACCOUNTID 34
 active 52
 Address Line 55
 Address Line 2 7, 8, 14, 16, 18, 20
 ADDRESSLINE 55
 ADDRESSLINE1 7, 8, 14, 16, 18, 20
 ADDRESSLINE2 7, 8, 14, 16, 18, 20
 administrationID 5, 7, 8, 10, 11, 12, 13, 14, 16, 18, 20, 21, 23, 24, 25, 27, 28, 29, 30, 31, 33, 34, 35, 37, 38, 39, 41, 42, 43, 44, 45, 47, 48, 50, 51, 52, 53, 55, 56, 57, 58
 ALERT 51
 AMOUNT 5, 10, 12, 21, 23, 24, 29, 39, 45, 47, 48
 50
 analysis-enforce-row-uniqueness 2
 api-url 2

- B -

Balance Type 5, 10, 24
 Balances by Date 5
 BalancesByDate 5
 BALANCETYPE 5, 10, 24
 bookyear 34
 Budget Purchase 30, 31
 Budget Sales 30, 31
 BUDGETPURCHASE 30, 31
 BUDGETSALES 30, 31
 bulk-delete-page-size-rows 2
 bulk-insert-page-size-bytes 2
 bulk-insert-page-size-rows 2

- C -

CheckOutstandingAdminItemsByReference 7
 CheckOutstandingItemsByReference 8
 CITY 7, 8, 14, 16, 18, 20, 55
 CoC Number 7, 8, 14, 16, 18, 20, 35, 55
 COCNUMBER 7, 8, 14, 16, 18, 20, 35, 55
 CODE 5, 10, 24, 28, 30, 31
 COMPANY 30, 31, 35

COMPANY_CODE 5, 7, 8, 10, 11, 12, 13, 14, 16, 18, 20, 21, 23, 24, 25, 27, 28, 29, 30, 31, 33, 34, 35, 37, 38, 39, 41, 42, 43, 44, 45, 47, 48, 50, 51, 52, 53, 55, 56, 57, 58

COMPANY_NAME 5, 7, 8, 10, 11, 12, 13, 14, 16, 18, 20, 21, 23, 24, 25, 27, 28, 29, 30, 31, 33, 34, 35, 37, 38, 39, 41, 42, 43, 44, 45, 47, 48, 50, 51, 52, 53, 55, 56, 57, 58

CONTACT 14, 30, 31

Contact Code 7, 8, 14, 16, 18, 20

Contact Country 35

Contact HID 35

Contact ID 7, 8, 14, 16, 18, 20, 21, 23, 30, 31, 35

Contact ZIP Code 35

CONTACTCODE 7, 8, 14, 16, 18, 20

CONTACTCOUNTRY 35

CONTACTHID 35

CONTACTID 7, 8, 14, 16, 18, 20, 21, 23, 30, 31, 35

CONTACTNAME 7, 8, 12, 16, 18, 20, 21, 23, 39, 45, 47, 48, 50

Administrations 55

Contacts 57

ContactsByDomainId 52

CONTACTZIPCODE 35

CONTENTTYPE 39, 45, 47, 48, 50

Cost Categories 37

CostCategories 37

COUNTRY 7, 8, 14, 16, 18, 20, 55

CREATED 39, 45, 47, 48, 50

CREATOR 39, 45, 47, 48, 50

Currencies 38

CURRENCY 35

Currency Rate 35

CURRENCYRATE 35

- D -

DATE 7, 8, 12, 14, 16, 18, 20, 21, 23

DEFAULT 38

DESCRIPTION 5, 7, 8, 10, 12, 14, 16, 18, 20, 21, 23, 24, 28, 30, 31, 35, 37, 38, 42, 52

Document Folder 35

Document Folder Tab 35

Document ID 35

Document Reference 35

Document Type 35

DocumentBinaryDataById 38

DOCUMENTBINARYDATARESULT 38

DocumentById 39

DOCUMENTCOUNT 41

DOCUMENTDATE 39, 45, 47, 48, 50

DOCUMENTFOLDER 35

DocumentFolderCounts 41

DocumentFolders 42
 DOCUMENTFOLDERTAB 35
 DocumentFolderTabsByFolderId 42
 documentID 35, 38, 39, 43, 44, 48
 DocumentImageByDocumentId 43
 DocumentImageCountByDocumentId 44
 DOCUMENTIMAGECOUNTRESULT 44
 DOCUMENTIMAGERESULT 43
 DOCUMENTREFERENCE 35
 Documents 45
 Documents by Transaction 25
 DocumentsByFolderId 47
 DocumentsBySearchCriteria 48
 DocumentsByTabId 50
 DocumentsByTransaction 25
 DOCUMENTTYPE 35
 domain_COMPANY_CODE 57, 58
 domain_COMPANY_NAME 57, 58
 domain_ID 57, 58
 domain_NAME 57, 58
 domain_URL 57, 58
 DOMAINID 5, 7, 8, 10, 11, 12, 13, 14, 16, 18, 20, 21, 23, 24, 25, 27, 28, 29, 30, 31, 33, 34, 35, 37, 38
 HID 30, 31, 35
 39, 41, 42, 43, 44, 45, 47, 48, 50, 51, 52, 53, 55, 56
 http-disk-cache-compression-level 2
 DOMAINNAME 5, 7, 8, 10, 11, 12, 13, 14, 16, 18
 http-disk-cache-directory 2
 20, 21, 23, 24, 25, 27, 28, 29, 30, 31, 33, 34, 35, 37
 http-disk-cache-ignore-write-errors 2
 38, 39, 41, 42, 43, 44, 45, 47, 48, 50, 51, 52, 53, 55
 http-disk-cache-max-age-sec 2
 56
 Domains 55
 Driver 1
 Due Date 7, 8, 14, 16, 18, 20
 DUEDATE 7, 8, 14, 16, 18, 20

- E -

EMAILHOME 14, 16, 18
 EMAILWORK 14, 16, 18
 End Date 30, 31
 endDate 11, 12, 13, 16, 20, 21, 23, 29, 30, 31, 35, 45, 47, 48, 50

- F -

File Data 25
 File Name 23, 25, 39, 45, 47, 48, 50
 FILEDATA 25
 FILENAME 23, 25, 39, 45, 47, 48, 50
 FILESIZE 39, 45, 47, 48, 50
 financialMode 29, 34, 35
 FinancialPeriodsByYear 27
 Fiscal Balances by Date 10

Fiscal Net Revenues 11
 Fiscal Transactions 12
 FiscalBalancesByDate 10
 FiscalNetRevenues 11
 FiscalTransactions 12
 FOLDERDESCRIPTION 39, 45, 47, 48, 50
 FOLDERID 39, 42, 45, 47, 48, 50
 force-case-sensitive-identifiers 2
 forced-casing-identifiers 2
 Full Name 35
 FULLNAME 35

- G -

General Ledger Account Code 12, 21, 23, 29, 35
 GL Account Schemes 28
 GLACCOUNTCODE 12, 21, 23, 29, 35
 GLAccountSchemes 28

- H -

HID 30, 31, 35
 http-disk-cache-compression-level 2
 DOMAINNAME 5, 7, 8, 10, 11, 12, 13, 14, 16, 18, 18
 http-disk-cache-directory 2
 20, 21, 23, 24, 25, 27, 28, 29, 30, 31, 33, 34, 35, 37
 http-disk-cache-ignore-write-errors 2
 38, 39, 41, 42, 43, 44, 45, 47, 48, 50, 51, 52, 53, 55
 http-disk-cache-max-age-sec 2
 56
 Domains 55
 Driver 1
 Due Date 7, 8, 14, 16, 18, 20
 DUEDATE 7, 8, 14, 16, 18, 20
 Human ID 30, 31, 35

- I -

ICON 42, 51
 ignore-xml-fatal-errors 2
 ignore-xml-no-access-errors 2
 includeBankTransactions 14, 16, 18, 20
 invantive-sql-compress-sparse-arrays 2
 invantive-sql-correct-invalid-date 2
 invantive-sql-forward-filters-to-data-containers 2
 invantive-sql-share-byte-arrays 2
 invantive-sql-share-strings 2
 invantive-sql-shuffle-fetch-results-data-containers 2
 invantive-use-cache 2
 Is Enabled 28, 33
 ISENABLED 28
 ISO 8601 Period 27
 ISO8601PERIOD 27

- L -

LASTLOGIN 53, 58
LINK 51
LOGIN 53, 58
log-native-calls-to-disk 2
log-native-calls-to-trace 2

- M -

Mail City 7, 8, 14, 16, 18, 20
Mail ZIP Code 7, 8, 14, 16, 18, 20
MAILADDRESSLINE1 7, 8, 14, 16, 18, 20
MAILADDRESSLINE2 7, 8, 14, 16, 18, 20
MAILCITY 7, 8, 14, 16, 18, 20
MAILPOSTCODE 7, 8, 14, 16, 18, 20
maxHeight 43
maximum-length-identifiers 2
max-url-length-accepted 2
max-url-length-desired 2
maxWidth 43
Menus 51
minimum-length-text 2
MODIFIED 39, 45, 47, 48, 50
modifiedAfter 52
MODIFIER 39, 45, 47, 48, 50
Mutation User 35
MUTATIONUSER 35

- N -

NAME 27, 52, 53, 55, 57, 58
Net Revenue Fiscal Result 11
Net Revenue Result 13
Net Revenues 13
NETREVENUEFISCALRESULT 11
NETREVENUERESULT 13
NetRevenues 13

- O -

Open Amount 7, 8, 14, 16, 18, 20
OPENAMOUNT 7, 8, 14, 16, 18, 20
Original Amount 7, 8, 14, 16, 18, 20
ORIGINALAMOUNT 7, 8, 14, 16, 18, 20
Outstanding Creditor Items 14
Outstanding Creditor Items by Date 16
Outstanding Debtor Items 18
Outstanding Debtor Items by Date 20

OutstandingCreditorItems 14
OutstandingCreditorItemsByDate 16
OutstandingDebtorItems 18
OutstandingDebtorItemsByDate 20

- P -

Partition Code 5, 7, 8, 10, 11, 12, 13, 14, 16, 18, 20, 21, 23, 24, 25, 27, 28, 29, 30, 31, 33, 34, 35, 37, 38, 39, 41, 42, 43, 44, 45, 47, 48, 50, 51, 52, 53, 55, 56, 57, 58
Partition Name 5, 7, 8, 10, 11, 12, 13, 14, 16, 18, 20, 21, 23, 24, 25, 27, 28, 29, 30, 31, 33, 34, 35, 37, 38, 39, 41, 42, 43, 44, 45, 47, 48, 50, 51, 52, 53, 55, 56, 57, 58
partition-slot-based-rate-limit-length-ms 2
partition-slot-based-rate-limit-slots 2
PAYMENTMETHOD 14, 16, 18, 20
PaymentMethods 52
PERIOD 27
Period ID 35
PERIODID 35
PHONEHOME 14, 16, 18
PHONEWORK 14, 16, 18
POSTCODE 7, 8, 14, 16, 18, 20, 55
pre-request-delay-ms 2
PROCESSEDBYYUKI 42
PROJECT 35
Project Balances by Date 29
Project Code 12, 21, 23, 29, 35
Project Details 30
ProjectBalancesByDate 29
PROJECTCODE 12, 21, 23, 29, 35
PROJECTDESCRIPTION 12, 21
ProjectDetails 30
PROJECTNAME 29
Projects 31

- R -

RECIPIENTEMAIL 14, 16, 18
REFERENCE 7, 8, 14, 16, 18, 20
requested-page-size 2
requests-parallel-max 2
Result 56
result-set-memory-cache 2
RGS Description 33
RGS Flip Code 33
RGS Reference Code 33
RGS Schemes 33
RGSDESCRIPTION 33

RGSFLIPCODE 33
 RGSREFERENCECODE 33
 RGSSchemes 33
 rgsVersion 33

- S -

Sales Invoice Schemas 56
 SALESINVOICESCHEMAPATHRESULT 56
 SalesInvoiceSchemas 56
 SALESITEM 23
 searchOption 30, 31, 48, 52
 searchText 48
 searchValue 30, 31, 52
 slot-based-rate-limit-length-ms 2
 slot-based-rate-limit-slots 2
 standardize-identifiers 2
 standardize-identifiers-casing 2
 Start Balance 34
 Start Date 30, 31, 55
 STARTBALANCE 34
 StartBalancesByYear 34
 STARTDATE 11, 12, 13, 16, 20, 21, 23, 29, 30, 31, 35, 45, 47, 48, 50, 55
 SUBJECT 39, 45, 47, 48, 50
 SUBTYPE 28

- T -

TABDESCRIPTION 39, 45, 47, 48, 50
 TABID 39, 45, 47, 48, 50
 TAGS 30, 31
 Tax Code Description 35
 TAXCODEDESCRIPTION 35
 TEXT 51
 Transaction Amount 35
 Transaction Amount Foreign Currency 35
 Transaction Date 35
 Transaction Details by Date 35
 TRANSACTIONAMOUNT 35
 TRANSACTIONAMOUNTFOREIGNCURRENCY
 TRANSACTIONDATE 5, 10, 24, 35
 TransactionDetailsByDate 35
 transactionID 25
 Transactions 21
 Transactions and Contacts 23
 TransactionsAndContacts 23
 TYPE 28, 39, 45, 47, 48, 50
 Type ID 7, 8, 14, 16, 18, 20
 TYPEDESCRIPTION 7, 8, 14, 16, 18, 20, 39, 45, 47, 48, 50

TYPEID 7, 8, 14, 16, 18, 20

- U -

URL 55, 57, 58
 use-http-disk-cache-read 2
 use-http-disk-cache-write 2
 use-http-memory-cache-read 2
 use-http-memory-cache-write 2
 use-metadata-memory-cache 2
 use-result-memory-cache 2
 Users 58
 UsersByDomainId 53

- V -

VAT Number 7, 8, 14, 16, 18, 20, 35, 55
 VATNUMBER 7, 8, 14, 16, 18, 20, 35, 55

- W -

Whole Period 27
 WHOLEPERIOD 27

- Y -

year 41
 Year-End Balances by Date 24
 YearEndBalancesByDate 24
 yearID 27
 Yuki 1, 5, 7, 8, 10, 11, 12, 13, 14, 16, 18, 20, 21, 23, 24, 25, 27, 28, 29, 30, 31, 33, 34, 35, 37, 38, 39, 41, 42, 43, 44, 45, 47, 48, 50, 51, 52, 53, 55, 56, 57, 58
 YUKICODE 33
 YUKIDESCRIPION 33
 YUKIENABLED 33

- Z -

ZIP Code 7, 8, 14, 16, 18, 20, 55

Copyright

(C) Copyright 2004-2020 Invantive Software B.V., the Netherlands. All rights reserved.

Alle rechten voorbehouden. Niets uit deze uitgave mag worden verveelvoudigd, opgeslagen in een geautomatiseerd gegevensbestand, of openbaar gemaakt, in enige vorm of op enige wijze, hetzij elektronisch, mechanisch, door fotokopieën, opnemen, of enig andere manier, zonder voorafgaande schriftelijke toestemming van de uitgever.

Ondanks alle aan de samenstelling van deze tekst bestede zorg, kan noch de schrijver noch de uitgever aansprakelijkheid aanvaarden voor eventuele schade, die zou kunnen voortvloeien uit enige fout, die in deze uitgave zou kunnen voorkomen.

Deze handleiding is een naslagwerk bedoeld om het gebruik te verduidelijken. Indien gegevens in de voorbeeldafbeeldingen overeenkomen met gegevens in uw systeem, dan is de overeenkomst toevallig.

Auteurs: Jan van Engelen, Michiel de Brieder, Mathijs Terhaag, Tanja Middelkoop, Guido Leenders, Tatjana Daka.

The JasperReports License, Version 1.0

Copyright (C) 2001-2004 Teodor Danciu(teodord@users.sourceforge.net).

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by Teodor Danciu (<http://jasperreports.sourceforge.net>).". Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.
4. The name "JasperReports" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact teodord@users.sourceforge.net.
5. Products derived from this software may not be called "JasperReports", nor may "JasperReports" appear in their name, without prior written permission of Teodor Danciu.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

