# KeePass Data Model

*for use with Invantive SQL*

24.0

**invantive**
the **SQL** company

# Copyright

# Important Safety and Usage Information

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

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

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

# Contents

# 1 SQL Driver for KeePass

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 <u>user community</u>. Invantive Support or other users will try to help you.

Security-sensitive storage of keys and other confidential information.

The KeePass driver covers 7 tables and 67 columns.

## KeePass Clients

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

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

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

## Specifications

The SQL driver for KeePass does not support partitioning.

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

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

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

Changes and bug fixes on the KeePass SQL driver can be found in the <u>release notes</u>. There is currently no specific section on the <u>Invantive forums</u> for KeePass.Please reach out to other users of KeePass by leaving a question or contact request.

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

Alias: `KeePass`

Recommended alias: `kp`

Status: Non-production

Updated 30-05-2024 10:35 using Invantive UniversalSQL version 24.1.3-BETA+4689.

# 2 SQL Driver Attributes for KeePass

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

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

- Connection string: the connection string from the settings*.xml file and applied during log on.

- Set SQL statement: a set SQL-statement to be executed once connection has been established.

- Log on: value to be specified interactively by user during log on in a user interface.

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

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

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

The KeePass driver can be configured using the following attributes:

| Code | Description | Origin | Default Value | Set from Connection String | Set from Set SQL-Statement | Set from Drivers File | Set from Log On |
|------|-------------|--------|---------------|------|------|------|------|
| analysis-enforce-row-uniqueness | Enforce rows to be unique for software analysis. A fingerprint is calculated from the whole row of data when the primary key column is unknown. | Shared | False | ✓ | ✓ | ✓ | |
| 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 | | ✓ | ✓ | ✓ | |

| Code | Description | Origin | Default Value | Set from Connection String | Set from Set SQL-Statement | Set from Drivers File | Set from Log On |
|------|-------------|--------|---------------|------------|------------|------------|------------|
| invantive-sql-compress-sparse-arrays | Whether to compress sparse arrays in result sets during compression. | SQL Engine V1 | True | ✓ | ✓ | ✓ | |
| invantive-sql-correct-invalid-date | Whether to correct dates considered invalid since they are before 01-01-1753. When nullable, they are removed. Otherwise they are replaced by 01-01-1753. | SQL Engine V1 | False | ✓ | ✓ | ✓ | |
| invantive-sql-execution-profile-disk-path | itgen_pae_invantive_sql_execution_profile_disk_path | SQL Engine V1 | c:\temp\profiles | ✓ | ✓ | ✓ | |
| invantive-sql-execution-profile-to-disk | itgen_pae_invantive_sql_execution_profile_to_disk | SQL Engine V1 | True | ✓ | ✓ | ✓ | |
| invantive-sql-forward-filters-to-data-containers | Whether to forward filters to data containers. | SQL Engine V1 | True | ✓ | ✓ | ✓ | |
| invantive-sql-share-byte-arrays | Whether to share the memory used by identical byte arrays in result sets during compression. | SQL Engine V1 | True | ✓ | ✓ | ✓ | |
| invantive-sql-share-strings | Whether to share the memory used by identical strings in result sets during compression. | SQL Engine V1 | True | ✓ | ✓ | ✓ | |
| invantive-sql-shuffle-fetch-results-data-containers | Whether to shuffle results fetched from data containers. | SQL Engine V1 | False | ✓ | ✓ | ✓ | |
| invantive-use-cache | Whether to cache the results of a query. | SQL Engine V1 | True | ✓ | ✓ | ✓ | |
| log-native-calls-to-disk-max-events | Maximum number of call events to register from last activation. | Shared | | ✓ | ✓ | ✓ | |
| log-native-calls-to-disk-max-seconds | Maximum number of seconds to register calls from last activation. | Shared | | ✓ | ✓ | ✓ | |
| log-native-calls-to-disk-on-error | Registers native calls to data container backend as disk files when the call raised an error. | Shared | False | ✓ | ✓ | ✓ | |
| log-native-calls-to-disk-on-success | Registers native calls to data container backend as disk files when the call raised no error. | Shared | False | ✓ | ✓ | ✓ | |
| log-native-calls-to-trace | Log native calls to data container backend on the trace. | Shared | False | ✓ | ✓ | ✓ | |
| max-url-length-accepted | The maximum accepted URL length before raising an error. | Shared | 8000 | ✓ | ✓ | ✓ | |
| max-url-length-desired | The maximum desired URL length. | Shared | 8000 | ✓ | ✓ | ✓ | |
| maximum-length-identifiers | Non-default maximum length in characters of identifier names. | Shared | | ✓ | ✓ | ✓ | |
| partition-slot-based-rate-limit-length-ms | Total length in milliseconds across all slots of a partition-based rate limit. | Shared | 60000 | ✓ | | ✓ | |

| Code | Description | Origin | Default Value | Set from Connection String | Set from Set SQL-Statement | Set from Drivers File | Set from Log On |
|------|-------------|--------|---------------|----------------------------|-----------------------------|------------------------|-----------------|
| 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 row s to exchange per round trip; only effective on limited platforms such as AFAS Online. | Shared | | ✓ | ✓ | ✓ | |
| requests-parallel-max | Maximum number of parallel data requests from individual partitions on the data container. | Shared | 32 | ✓ | ✓ | ✓ | |
| slot-based-rate-limit-length-ms | Total length in milliseconds across all slots of a slot-based rate limit. | Shared | 60000 | ✓ | | ✓ | |
| slot-based-rate-limit-slots | Number of slots of a slot-based rate limit. Null means no slot-based rate limit. | Shared | | ✓ | | ✓ | |
| standardize-identifiers-casing | Rew rite all identifiers to the recommended standard platform-specific casing w hen changing a data model on a case-dependent platform. | Shared | True | ✓ | ✓ | ✓ | |
| standardize-identifiers | Rew rite all identifiers to the preferred standards as configured by standardize-identifiers-casing and maximum-length-identifiers. | Shared | True | ✓ | ✓ | ✓ | |

# 3 keepass_entities: KeePass Entities

Catalog: KeePass

Schema: KeePass

Label: Entities

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

## Parameters of Table Function

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

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

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

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

| Name | Data Type | Required | Default Value | Documentation |
|------|-----------|----------|---------------|---------------|
| keyfile | string | ☐ | | Key file for the KDBX file. Either password or key file must be specified. |
| password | string | ☐ | | Password for the KDBX file. Either password or key file must be specified. |
| path | string | ☑ | | Full path to the KDBX file with password entries. |

# Columns of Table Function

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

| Name | Data Type | Label | Required | Documentation |
|------|-----------|-------|----------|---------------|
| AutoType | boolean | | ☐ | |
| CustomBackGroundColor | string | | ☐ | |
| CustomForeGroundColor | string | | ☐ | |
| Expires | boolean | | ☐ | |
| ExpiryTime | datetime | | ☐ | |
| IconId | int32 | | ☐ | |
| id | string | | ☐ | |
| name | string | | ☐ | |
| notes | string | | ☐ | |
| OverrideUrl | string | | ☐ | |
| parent_entry_id | string | | ☐ | |
| password | string | | ☐ | |
| Tags | string | | ☐ | |
| title | string | | ☐ | |
| type | string | | ☐ | |
| url | string | | ☐ | |
| username | string | | ☐ | |

# 4 keepass_entry_custom_sequences: KeePass Entry Custom Sequences

Catalog: KeePass

Schema: KeePass

Label: Entry Custom Sequences

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

## Parameters of Table Function

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

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

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

| Name | Data Type | Required | Default Value | Documentation |
|---|---|---|---|---|
| entry_uuid | string | ☑ | | UUID of the entry to query. |
| keyfile | string | ☐ | | Key file for the KDBX file. Either password or key file must be specified. |
| password | string | ☐ | | Password for the KDBX file. Either password or key file must be specified. |
| path | string | ☑ | | Full path to the KDBX file with password entries. |

## Columns of Table Function

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

| Name | Data Type | Label | Required | Documentation |
|---|---|---|---|---|
| Sequence | string | | ☐ | |
| Target_window | string | | ☐ | |

# 5 keepass_entry_file_attachments: KeePass Entry File Attachments

Catalog: KeePass

Schema: KeePass

Label: Entry File Attachments

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

## Parameters of Table Function

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

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

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

| Name | Data Type | Required | Default Value | Documentation |
|---|---|---|---|---|
| entry_uuid | string | ☑ | | UUID of the entry to query. |
| keyfile | string | ☐ | | Key file for the KDBX file. Either password or key file must be specified. |
| password | string | ☐ | | Password for the KDBX file. Either password or key file must be specified. |
| path | string | ☑ | | Full path to the KDBX file with password entries. |

## Columns of Table Function

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

| Name | Data Type | Label | Required | Documentation |
|---|---|---|---|---|
| file | string | | ☐ | |

# 6 keepass_entry_history: KeePass Entries History

Catalog: KeePass

Schema: KeePass

Label: Entries History

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

## Parameters of Table Function

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

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

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

| Name | Data Type | Required | Default Value | Documentation |
|------|-----------|----------|---------------|---------------|
| entry_uuid | string | ☑ | | UUID of the entry to query. |
| keyfile | string | ☐ | | Key file for the KDBX file. Either password or key file must be specified. |
| password | string | ☐ | | Password for the KDBX file. Either password or key file must be specified. |
| path | string | ☑ | | Full path to the KDBX file with password entries. |

## Columns of Table Function

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

| Name | Data Type | Label | Required | Documentation |
|------|-----------|-------|----------|---------------|
| AutoType | boolean | | ☐ | |
| CustomBackGroundColor | string | | ☐ | |
| CustomForeGroundColor | string | | ☐ | |
| Expires | boolean | | ☐ | |
| ExpiryTime | datetime | | ☐ | |
| IconId | int32 | | ☐ | |

| Name | Data Type | Label | Required | Documentation |
|------|-----------|-------|----------|---------------|
| id | string | | ☐ | |
| name | string | | ☐ | |
| notes | string | | ☐ | |
| OverrideUrl | string | | ☐ | |
| parent_entry_id | string | | ☐ | |
| password | string | | ☐ | |
| Tags | string | | ☐ | |
| title | string | | ☐ | |
| type | string | | ☐ | |
| url | string | | ☐ | |
| username | string | | ☐ | |

# 7 keepass_entry_string_fields: KeePass Entry String Fields

Catalog: KeePass

Schema: KeePass

Label: Entry String Fields

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

## Parameters of Table Function

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

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

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

| Name | Data Type | Required | Default Value | Documentation |
|------|-----------|----------|---------------|---------------|
| entry_uuid | string | ☑ | | UUID of the entry to query. |
| keyfile | string | ☐ | | Key file for the KDBX file. Either password or key file must be specified. |
| password | string | ☐ | | Password for the KDBX file. Either password or key file must be specified. |

| Name | Data Type | Required | Default Value | Documentation |
|------|-----------|----------|---------------|---------------|
| path | string | ☑ | | Full path to the KDBX file with password entries. |

## Columns of Table Function

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

| Name | Data Type | Label | Required | Documentation |
|------|-----------|-------|----------|---------------|
| key | string | | ☐ | |
| value | string | | ☐ | |

# 8 keepass_file_metadata_binaries: KeePass File Metadata Binaries

Catalog: KeePass

Schema: KeePass

Label: File Metadata Binaries

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

## Parameters of Table Function

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

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

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

| Name | Data Type | Required | Default Value | Documentation |
|------|-----------|----------|---------------|---------------|
| keyfile | string | ☐ | | Key file for the KDBX file. Either password or key file must be specified. |
| password | string | ☐ | | Password for the KDBX file. Either password or key file must be specified. |

| Name | Data Type | Required | Default Value | Documentation |
|------|-----------|----------|---------------|---------------|
| path | string | ☑ | | Full path to the KDBX file w ith passw ord entries. |

## Columns of Table Function

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

| Name | Data Type | Label | Required | Documentation |
|------|-----------|-------|----------|---------------|
| Compressed | string | | ☐ | |
| Id | string | | ☐ | |
| Value | string | | ☐ | |

# 9 keepass_file_metadata: KeePass File Metadata

Catalog: KeePass

Schema: KeePass

Label: File Metadata

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

## Parameters of Table Function

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

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

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

| Name | Data Type | Required | Default Value | Documentation |
|------|-----------|----------|---------------|---------------|
| keyfile | string | ☐ | | Key file for the KDBX file. Either passw ord or key file must be specified. |
| passw ord | string | ☐ | | Passw ord for the KDBX file. Either passw ord or key file must be specified. |
| path | string | ☑ | | Full path to the KDBX file w ith passw ord entries. |

# Columns of Table Function

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

| Name | Data Type | Label | Required | Documentation |
|------|-----------|-------|----------|---------------|
| Color | string | | ☐ | |
| DatabaseDescription | string | | ☐ | |
| DatabaseName | string | | ☐ | |
| DatabaseNameChanged | datetime | | ☐ | |
| DefaultUsername | string | | ☐ | |
| DefaultUsernameChanged | datetime | | ☐ | |
| EntryTemplatesGroup | string | | ☐ | |
| EntryTemplatesGroupChanged | datetime | | ☐ | |
| Generator | string | | ☐ | |
| HeaderHash | string | | ☐ | |
| HistoryMaxItems | int32 | | ☐ | |
| HistoryMaxSize | int32 | | ☐ | |
| LastSelectedGroup | string | | ☐ | |
| LastTopVisibleGroup | string | | ☐ | |
| MaintenanceHistoryDays | int32 | | ☐ | |
| MasterKeyChanged | datetime | | ☐ | |
| MasterKeyChangeForce | int32 | | ☐ | |
| MasterKeyChangeRec | int32 | | ☐ | |
| ProtectNotes | boolean | | ☐ | |
| ProtectPassword | boolean | | ☐ | |
| ProtectURL | boolean | | ☐ | |
| ProtectUsername | boolean | | ☐ | |
| RecycleBinChanged | datetime | | ☐ | |
| RecycleBinEnabled | boolean | | ☐ | |
| RecycleBinUUID | string | | ☐ | |

# Index

## - A -

## - B -

## - C -

## - D -

## - E -

## - F -

## - G -

## - H -

## - I -

## - K -

## - L -

# - M -

MaintenanceHistoryDays     11
MasterKeyChanged     11
MasterKeyChangeForce     11
MasterKeyChangeRec     11
maximum-length-identifiers     1
max-url-length-accepted     1
max-url-length-desired     1

# - N -

name     4, 8
notes     4, 8

# - O -

OverrideUrl     4, 8

# - P -

parent_entry_id     4, 8
partition-slot-based-rate-limit-length-ms     1
partition-slot-based-rate-limit-slots     1
password     4, 6, 7, 8, 9, 10, 11
path     4, 6, 7, 8, 9, 10, 11
pre-request-delay-ms     1
ProtectNotes     11
ProtectPassword     11
ProtectURL     11
ProtectUsername     11

# - R -

RecycleBinChanged     11
RecycleBinEnabled     11
RecycleBinUUID     11
requested-page-size     1
requests-parallel-max     1

# - S -

Sequence     6
slot-based-rate-limit-length-ms     1
slot-based-rate-limit-slots     1
standardize-identifiers     1
standardize-identifiers-casing     1

# - T -

Tags     4, 8
Target_window     6
title     4, 8
type     4, 8

# - U -

url     4, 8
username     4, 8

# - V -

Value     9, 10