

External Tag Provider Reference

Important

If you are using the default Tag provider, none of this is applicable. The following information is only valid for new Tag providers that you create using the **External Tag Provider** type.

The information provided here requires an understanding of Tags and how they work. It is an advanced reference to how the tables of external Tags providers are structured, and an overview of the concepts of Tag execution. If you are a new user, it is suggested that you read the [Understanding Tags](#) section first.

SQLT Table Descriptions

Basic Concepts and Data Flow

Tags operate through tables created in the database.

Tag Configuration Table Names and Descriptions	
sqlt_core	The core Tag information table, has one entry per Tag. Defines fundamental properties like data type, as well as the current value of the Tag. Is monitored by the provider to determine value and configuration changes.
sqlt_meta	Provides additional properties for Tags. Only consulted when Tag configuration has changed.
sqlt_as	Provides alert state configuration for Tags which utilize alerting.
sqlt_perm	Provides custom permission settings for Tags set to use them.
Operations Table Names and Descriptions	
sqlt_sc	Contains the definitions of scan classes, which dictate how Tags are executed.
sqlt_sci	Contains an entry for each scan class from sqlt_sc, for each driver currently driving Tags. Used to verify that drivers are properly executing.
sqlt_drv	Contains an entry for each Tags driver. Only really used for browsing Tags.
sqlt_err	Contains errors that have occurred executing Tags.
sqlt_wq	The "write queue". All write requests are entered into this table, where the driver will detect and execute them. The result will be written back by the driver, and will be noticed by the provider.

Tag Execution Concepts

Polling – Many operations require polling of the database by either the driver or the provider. To ensure efficiency, all polling operations utilize indexed timestamp fields. This allows the database to do very little work when nothing has changed.

Tag Configuration – Tags are configured by inserting or modifying the appropriate entries in the configuration tables above. Configuration change is signaled to the provider by updating the **configchange** of **sqlt_core** to be the current time. Deleting a Tag works by setting its **deleted** column and then **touching** config change. This will inform all drivers and providers to remove the Tag from memory. At some point later, a daemon will delete the Tag information from the database.

Tag Execution, drivers – Each Tag has a **drivername** property that indicates which driver is responsible for executing it. Other drivers and providers with different names will treat the Tag as an **external** Tag – a Tag driven by a different entity – and will only monitor its value.

Tag Execution, scan classes – Each Tag is assigned to a scan class. The idea is that scan classes will define how often the Tag should execute, as well as provide more advanced options like leased and driven execution. In reality, the Tag driver is free to execute Tags as it desires, but it is important to understand how the scan classes and the **sqlt_sci** table are expected to work, as that is how the provider will verify that the Tags are being executed.

Tag Monitoring – Both providers and drivers generally monitor Tag value and configuration changes. In general, the entities will monitor Tags whose **drivername** isn't equal to their own, which for providers means all Tags, since providers don't have a driver name. Monitoring

occurs by selecting the Tag values (or any information desired) from the appropriate table where one of the indexed timestamp columns is greater than the last checked time. The provider/driver will then store that time in memory as the last check, and will use it in the next poll.

Table Reference

The following is a reference list for the table structures of all the tables listed above. In general, all integer time values are in milliseconds.

sqlt_core		
Column	Data Type	Notes
id	integer	Auto-incrementing, unique id for the Tag
name	string	Name of Tag
path	string	Folder path, in form of <code>path/to/</code>
drivename	string	Name of driver responsible for executing Tags
tagtype	integer / TagType enum	The type of Tag - that is, OPC, DB, and so on
datatype	integer / DataType enum	The type of data provided by the Tag
enabled	integer (0 or 1)	Whether the Tag is enabled for execution
accessrights	integer / AccessRightsenum	Access permissions for the Tag
scanclass	integer	ID of the scan class for the Tag
intvalue	integer	Value column used if Tag has integer data
floatvalue	double	Value column for float/real data
stringvalue	string	Value column for string data
datevalue	datetime	Value column for date data
dataintegrity	integer / DataQualityenum	Current quality of the value
deleted	integer (0 or 1)	Whether the Tag is deleted or not
valuechange	datetime	The last time that the value changed
configchange	datetime	The last time that the Tag's config changed

sqlt_sc		
Column	Data Type	Notes
id	integer	Auto-incrementing unique id
name	string	Name of the scan class
lorate	integer	The slower rate to run at, in milliseconds. Only rate used if scan class mode is direct
hirate	integer	Higher rate, in ms. Only used if scan class is driver or leased
drivingtagpath	string	Path to Tag to watch if mode is driven
comparison	integer / Comparison enum	Operation to apply to driving Tag in driven mode
comparevalue	double	Value to compare driving Tag to for driven mode
mode	integer / Scan class mode enum	The mode of the scan class
staletimeout	integer	Time, in milliseconds, before scan class is determined to not be running
leaseexpire	datetime	The time that the lease should expire, if using leased mode
configchange	datetime	The last time that the scan class has been modified
deleted	integer (0 or 1)	Whether the scan class has been deleted

sqlt_sci

sqlt_meta		
Column	Data Type	Notes
tagid	Integer	ID of tag that the property belongs to
name	string	The well-known property name
intval	integer	Value, if property has integer type
floatval	double	Value if property has float type
stringval	string	Value, if property has string type

sqlt_as		
Column	Data Type	Notes
id	integer	Unique id of alert state
statename	string	Name of alert state
severity	integer /Severity enum	
low	double	Low setpoint
high	double	High setpoint
flags	integer / Alert Flags	Flags that dictate how the state acts
lotagpath	string	Path to Tag that provides low setpoint, if low driven flag is set
hitagpath	string	Path to Tag that Provides high setpoint, if high driven flag is set
timedeadband	double	Time deadband value
timedbunits	integer / TimeUnits enum	Time deadband units

sqlt_perm		
Column	Data Type	Notes
tagid	integer	ID of Tag that the permission belongs to

Column	Data Type	Notes
sc_id	integer	The id of the scan class represented
drivename	string	The driver executing this instance
lastexec	datetime	Last time that the scan class executed
lastexecrate	integer	The rate of the scan class at last execution
lastecexduration	integer	Time, in ms, that the scan class took to execute
lastexecopcwrite	integer	Writes to OPC performed during last execution
lastexecopcreads	integer	Value updates from OPC processed in last execution
lastexecdbwrites	integer	Writes to DB performed during last execution
lastexecdbreads	integer	Value updates from the database processed during the last execution
lastecexdelay	integer	The delay between when the scan class should have ran and when it actually ran for the last execution
avgexecduration	integer	The average duration time of the scan class, in ms
execcount	integer	The number of times the scan class has executed
nextexec	datetime	The next time that the scan class should execute

sqlt_wq		
Column	Data Type	Notes
id	integer	Auto-incrementing unique id for the write operation
tagid	integer	ID of the Tag to write to

rolename	string	Name of the role that this permission is applied to
accessrights	integer / AccessRights enum	Access rights for the given role on the given Tag

sqlt_drv		
Column	Data Type	Notes
name	string	Name of the Tag drive
ipaddr	string	Address of browser server, blank or null if browsing isn't available
port	integer	Port of browse server

intvalue	integer	Value, if Tag has integer data type
floatvalue	double	Value, if Tag has float or real data type
stringvalue	string	Value, if Tag has string data type
datevalue	datetime	Value, if Tag has date data type
responsecode	integer / Write Responseenum	The state of the write request. When created, the response code should be set to 2 - Pending
responsemsg	string	Write error if operation failed
t_stamp	datetime	The time that the write request was created
objectid	integer	ID of the object with the error
objectype	integer / Object Type enum	The type of object. Used with objectid to identify the item that caused the message
lifecycleid	integer/ Lifecycle enum	When the message was generated
msgtype	integer / Message Type enum	
errormsg	string	The primary message
stack	string	Additional error information
t_stamp	datetime	When the message was generated