WinCC OA Archiving in a Nutshell

R. Kulaga
BE-ICS-FD

Topics

- Archiving in WinCC OA: past, present and future
- Current architecture of WinCC OA archiving
- Retrieval of historical data from WinCC OA – available functions
- Quick tour of the Oracle schema
- Metadata history keeping and fwRDBAPI component

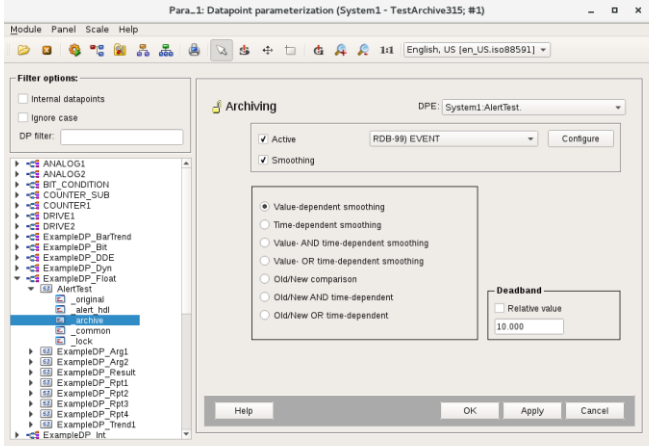
Archiving in WinCC OA

- Archiving of historical process values and alarms is one of the key functions of a SCADA system
    - Without it, the users can only see the current process snapshot

- In WinCC OA, archiving is performed by dedicated managers
    - In the past: Archive Manager (Database Manager for alarms)
    - Currently: RDB Archive Manager
    - In the future: NextGen Archiver Frontend Manager, communicating with various backends

What is archived?

- Value changes of DPEs with enabled `_archive` config (according to smoothing rules)



Para_1: Datapoint parameterization (System1 - TestArchive315; #1)

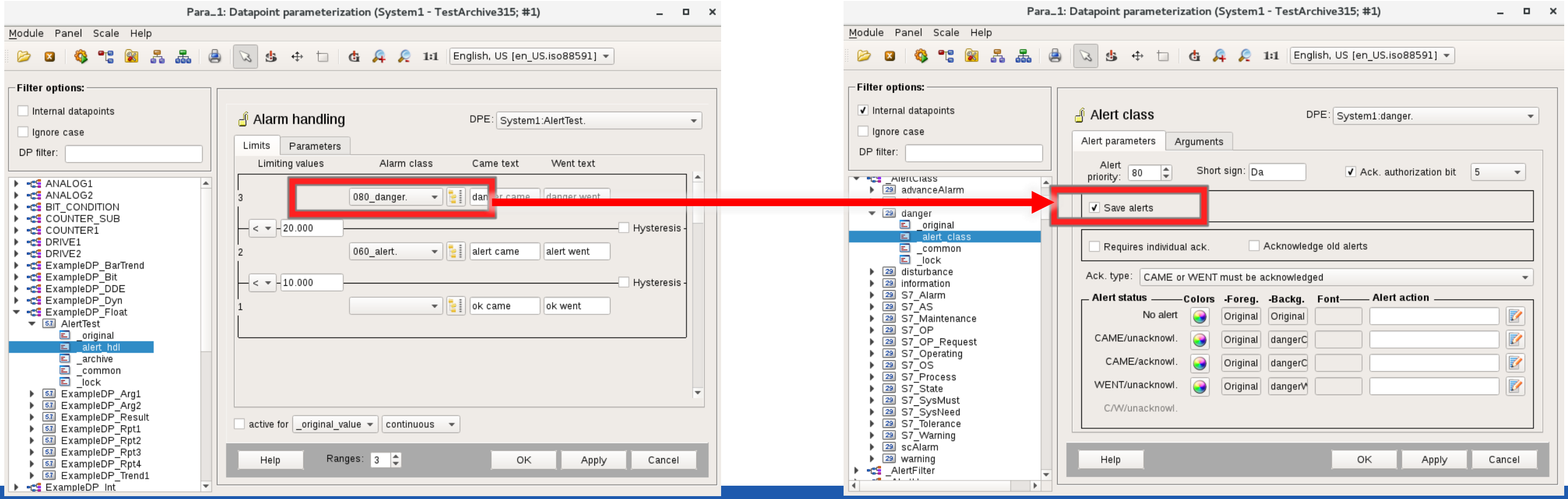
Filter options:
- ☐ Internal datapoints
- ☐ Ignore case
- DP filter:

Archiving — DPE: System1:AlertTest.

- ☑ Active — RDB-99) EVENT
- ☑ Smoothing
- ● Value-dependent smoothing
- ○ Time-dependent smoothing
- ○ Value- AND time-dependent smoothing
- ○ Value- OR time-dependent smoothing
- ○ Old/New comparison
- ○ Old/New AND time-dependent
- ○ Old/New OR time-dependent

Deadband
- ☐ Relative value
- 10.000

- Alarms with alarm classes that have storage of historical alarms enabled



Para_1: Datapoint parameterization (System1 - TestArchive315; #1)

Filter options:
- ☐ Internal datapoints
- ☐ Ignore case
- DP filter:

Alarm handling — DPE: System1:AlertTest.

Limits | Parameters

	Limiting values	Alarm class	Came text	Went text	
3		080_danger.	danger came	danger went	
	< 20.000				☐ Hysteresis
2		060_alert.	alert came	alert went	
	< 10.000				☐ Hysteresis
1			ok came	ok went	

☐ active for _original_value — continuous

Ranges: 3

Para_1: Datapoint parameterization (System1 - TestArchive315; #1)

Filter options:
- ☑ Internal datapoints
- ☐ Ignore case
- DP filter:

Alert class — DPE: System1:danger.

Alert parameters | Arguments

Alert priority: 80 — Short sign: Da — ☑ Ack. authorization bit 5

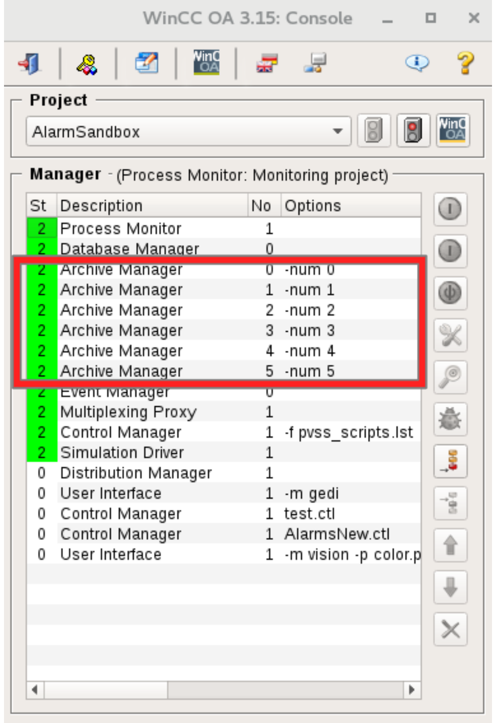
- ☑ Save alerts
- ☐ Requires individual ack.
- ☐ Acknowledge old alerts

Ack. type: CAME or WENT must be acknowledged

Alert status	Colors	-Foreg.	-Backg.	Font	Alert action
No alert		Original	Original		
CAME/unacknowl.		Original	dangerC		
CAME/acknowl.		Original	dangerC		
WENT/unacknowl.		Original	dangerW		
C/W/unacknowl.					

Value Archives (HDB) – (almost) past

- No longer supported at CERN
- History of value changes stored in proprietary file format in project directory
- Historical alarms stored in RAIMA databases (directly by the Database Manager)
- Tricky to configure properly – need to know approximate frequencies of datapoint changes
- Problems with archive consistency after crashes
- Not very scalable – all queries need to go through the Database Manager (direct queries not supported)



WinCC OA 3.15: Console

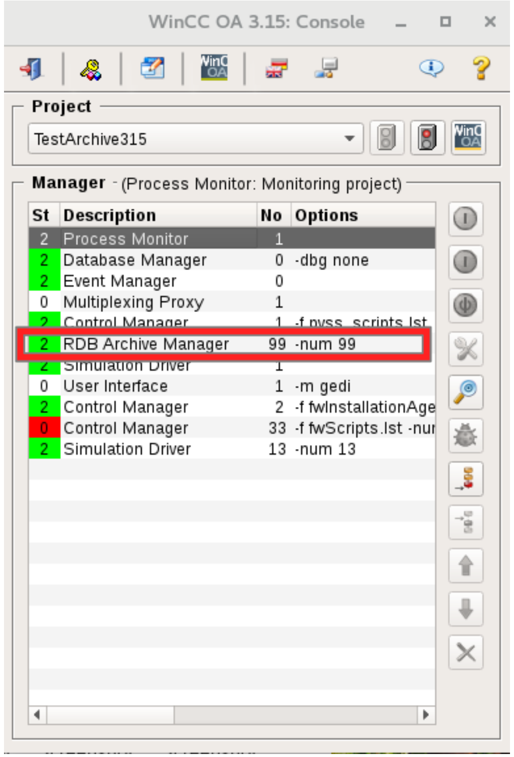
Project: AlarmSandbox

Manager - (Process Monitor: Monitoring project)

St	Description	No	Options
2	Process Monitor	1	
2	Database Manager	0	
2	Archive Manager	0	-num 0
2	Archive Manager	1	-num 1
2	Archive Manager	2	-num 2
2	Archive Manager	3	-num 3
2	Archive Manager	4	-num 4
2	Archive Manager	5	-num 5
2	Event Manager	0	
2	Multiplexing Proxy	1	
2	Control Manager	1	-f pvss_scripts.lst
2	Simulation Driver	1	
0	Distribution Manager	1	
0	User Interface	1	-m gedi
0	Control Manager	1	test.ctl
0	Control Manager	1	AlarmsNew.ctl
0	User Interface	1	-m vision -p color.p

RDB Archive Manager – present

- Almost all production projects currently use the RDB Archive Manager, which writes history to Oracle databases
- Developed in collaboration between ETM and CERN
- Used in parallel with Value Archives in several projects until EYETS 2016/2017
- Can handle high data throughputs when archive groups with limited sets of columns are used – e.g. QPS use case with 200,000 changes per second
- Data retrieval performance can vary greatly and is affected by:
    - Number of queried datapoints and their frequencies of changes
    - Whether or not *bonus values* are retrieved – *n* values from before the query start time and/or after query end time



Project

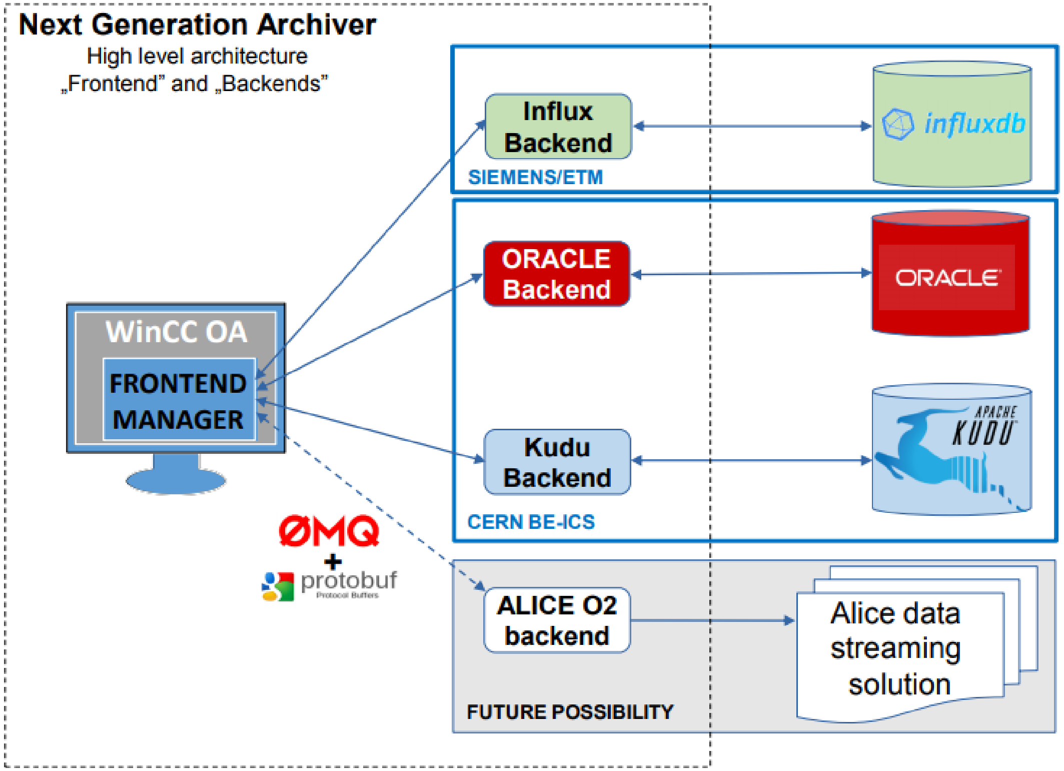
TestArchive315

Manager - (Process Monitor: Monitoring project)

St	Description	No	Options
2	Process Monitor	1	
2	Database Manager	0	-dbg none
2	Event Manager	0	
0	Multiplexing Proxy	1	
2	Control Manager	1	-f pvss_scripts.lst
2	RDB Archive Manager	99	-num 99
2	Simulation Driver	1	
0	User Interface	1	-m gedi
2	Control Manager	2	-f fwInstallationAge
0	Control Manager	33	-f fwScripts.lst -nur
2	Simulation Driver	13	-num 13

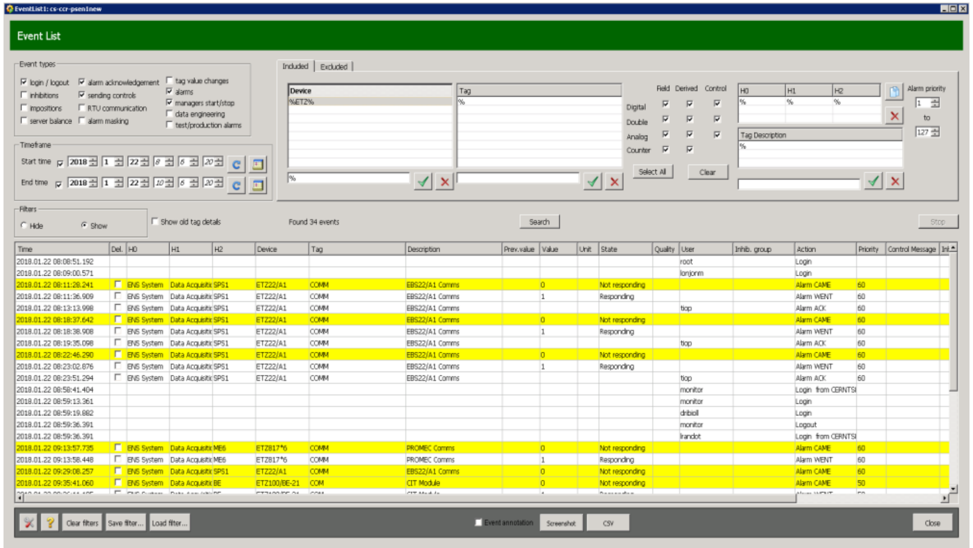
NextGeneration Archiver – future

- Developed in collaboration between CERN and ETM
- Main advantages:
  - Possibility to simultaneously archive to multiple storages
  - Backends implement a high-level communication protocol (ZMQ + Google ProtoBuf) that's not dependend on WinCC OA
- Can act as a general-purpose data *fanout*
  - ALICE O2 use case


Next Generation Archiver
High level architecture
„Frontend" and „Backends"

Accessing archived data from WinCC OA

- `dpGetPeriod()` / `alertGetPeriod()`
- `dpGetAsynch()`
- `dpQuery()`
- Custom SQL queries – e.g. in PSEN Event Screen



Event types
☑ login / logout ☑ alarm acknowledgement □ tag value changes
□ inhibitions ☑ sending controls ☑ alarms
□ impositions □ RTU communication ☑ managers start/stop
□ server balance □ alarm masking □ data engineering
□ test/production alarms

Timeframe
Start time ☑ 2018 1 22 8 6 20
End time ☑ 2018 1 22 10 6 20

Included | Excluded

Device: %ETZ%
Tag: %
%

	Field	Derived	Control
Digital	☑	☑	☑
Double	☑	☑	☑
Analog	☑	☑	☑
Counter	☑	☑	

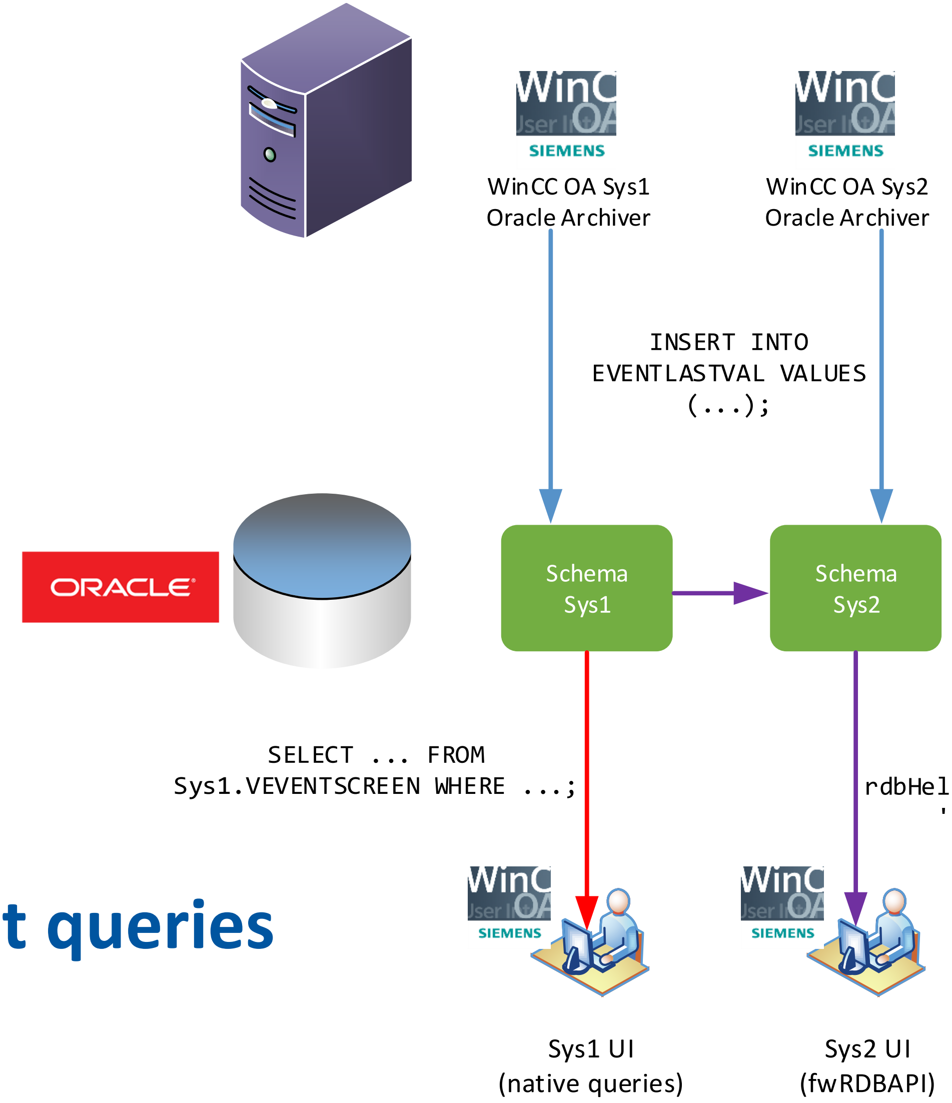
H0: % H1: % H2: %
Tag Description: %
Alarm priority 1 to 127

Filters ○ Hide ● Show
□ Show old tag details
Found 34 events

Time	Del.	H0	H1	H2	Device	Tag	Description	Prev.value	Value	Unit	State	Quality	User	Inhib. group	Action	Priority	Control Message	Inl
2018.01.22 08:08:51.192													root		Login			
2018.01.22 08:09:00.571													lonjonm		Login			
2018.01.22 08:11:28.241	□	ENS System	Data Acquisitic	SPS1	ETZ22/A1	COMM	EBS22/A1 Comms		0		Not responding				Alarm CAME	60		
2018.01.22 08:11:36.909	□	ENS System	Data Acquisitic	SPS1	ETZ22/A1	COMM	EBS22/A1 Comms		1		Responding				Alarm WENT	60		
2018.01.22 08:13:13.998	□	ENS System	Data Acquisitic	SPS1	ETZ22/A1	COMM	EBS22/A1 Comms						tiop		Alarm ACK	60		
2018.01.22 08:18:37.642	□	ENS System	Data Acquisitic	SPS1	ETZ22/A1	COMM	EBS22/A1 Comms		0		Not responding				Alarm CAME	60		
2018.01.22 08:18:38.908	□	ENS System	Data Acquisitic	SPS1	ETZ22/A1	COMM	EBS22/A1 Comms		1		Responding				Alarm WENT	60		
2018.01.22 08:19:35.098	□	ENS System	Data Acquisitic	SPS1	ETZ22/A1	COMM	EBS22/A1 Comms						tiop		Alarm ACK	60		
2018.01.22 08:22:46.290	□	ENS System	Data Acquisitic	SPS1	ETZ22/A1	COMM	EBS22/A1 Comms		0		Not responding				Alarm CAME	60		
2018.01.22 08:23:02.876	□	ENS System	Data Acquisitic	SPS1	ETZ22/A1	COMM	EBS22/A1 Comms		1		Responding				Alarm WENT	60		
2018.01.22 08:23:51.294	□	ENS System	Data Acquisitic	SPS1	ETZ22/A1	COMM	EBS22/A1 Comms						tiop		Alarm ACK	60		
2018.01.22 08:58:41.404													monitor		Login from CERNTSI			
2018.01.22 08:59:13.361													monitor		Login			
2018.01.22 08:59:19.882													dribiol		Login			
2018.01.22 08:59:36.391													monitor		Logout			
2018.01.22 08:59:36.391													lrandot		Login from CERNTSI			
2018.01.22 09:13:57.735	□	ENS System	Data Acquisitic	ME6	ETZ817*6	COMM	PROMEC Comms		0		Not responding				Alarm CAME	60		
2018.01.22 09:13:58.448	□	ENS System	Data Acquisitic	ME6	ETZ817*6	COMM	PROMEC Comms		1		Responding				Alarm WENT	60		
2018.01.22 09:29:08.257	□	ENS System	Data Acquisitic	SPS1	ETZ22/A1	COMM	EBS22/A1 Comms		0		Not responding				Alarm CAME	60		
2018.01.22 09:35:41.060	□	ENS System	Data Acquisitic	BE	ETZ100/BE-21	COM	CIT Module		0		Not responding				Alarm CAME	50		

□ Event annotation

Archiving and data retrieval in WinCC OA – data flows



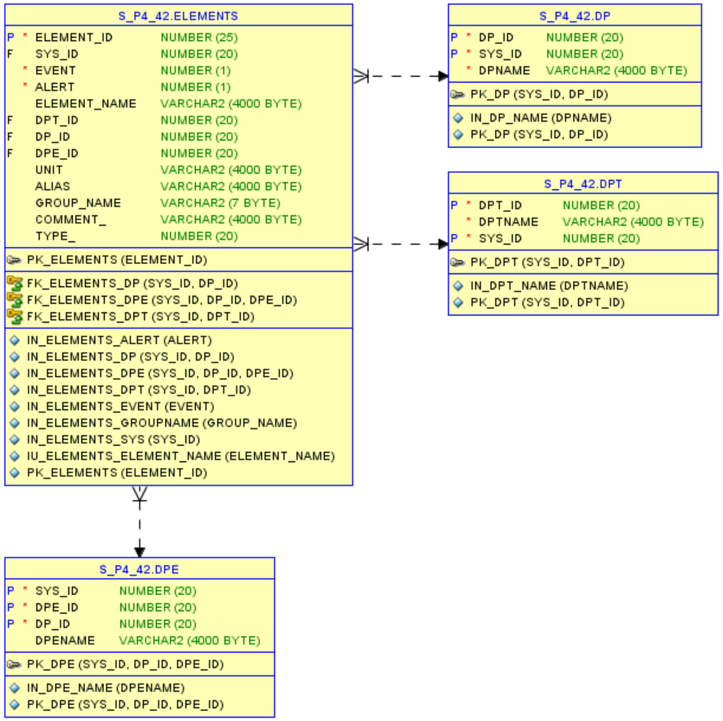
```
INSERT INTO
EVENTLASTVAL VALUES
(...);
```

```
SELECT ... FROM
Sys1.VEVENTSCREEN WHERE ...;
```

```
SELECT
rdbHelpers.dpGetPeriod(pSchema =>
'Sys1', ...) FROM DUAL;
```

Direct queries

RDB Archive Manager Oracle schema tables



S_P4_42.ELEMENTS

P	*	ELEMENT_ID	NUMBER (25)
F		SYS_ID	NUMBER (20)
	*	EVENT	NUMBER (1)
	*	ALERT	NUMBER (1)
		ELEMENT_NAME	VARCHAR2 (4000 BYTE)
F		DPT_ID	NUMBER (20)
F		DP_ID	NUMBER (20)
F		DPE_ID	NUMBER (20)
		UNIT	VARCHAR2 (4000 BYTE)
		ALIAS	VARCHAR2 (4000 BYTE)
		GROUP_NAME	VARCHAR2 (7 BYTE)
		COMMENT_	VARCHAR2 (4000 BYTE)
		TYPE_	NUMBER (20)

PK_ELEMENTS (ELEMENT_ID)

FK_ELEMENTS_DP (SYS_ID, DP_ID)
FK_ELEMENTS_DPE (SYS_ID, DP_ID, DPE_ID)
FK_ELEMENTS_DPT (SYS_ID, DPT_ID)

IN_ELEMENTS_ALERT (ALERT)
IN_ELEMENTS_DP (SYS_ID, DP_ID)
IN_ELEMENTS_DPE (SYS_ID, DP_ID, DPE_ID)
IN_ELEMENTS_DPT (SYS_ID, DPT_ID)
IN_ELEMENTS_EVENT (EVENT)
IN_ELEMENTS_GROUPNAME (GROUP_NAME)
IN_ELEMENTS_SYS (SYS_ID)
IU_ELEMENTS_ELEMENT_NAME (ELEMENT_NAME)
PK_ELEMENTS (ELEMENT_ID)

S_P4_42.DP

P	*	DP_ID	NUMBER (20)
P	*	SYS_ID	NUMBER (20)
	*	DPNAME	VARCHAR2 (4000 BYTE)

PK_DP (SYS_ID, DP_ID)

IN_DP_NAME (DPNAME)
PK_DP (SYS_ID, DP_ID)

S_P4_42.DPT

P	*	DPT_ID	NUMBER (20)
	*	DPTNAME	VARCHAR2 (4000 BYTE)
P	*	SYS_ID	NUMBER (20)

PK_DPT (SYS_ID, DPT_ID)

IN_DPT_NAME (DPTNAME)
PK_DPT (SYS_ID, DPT_ID)

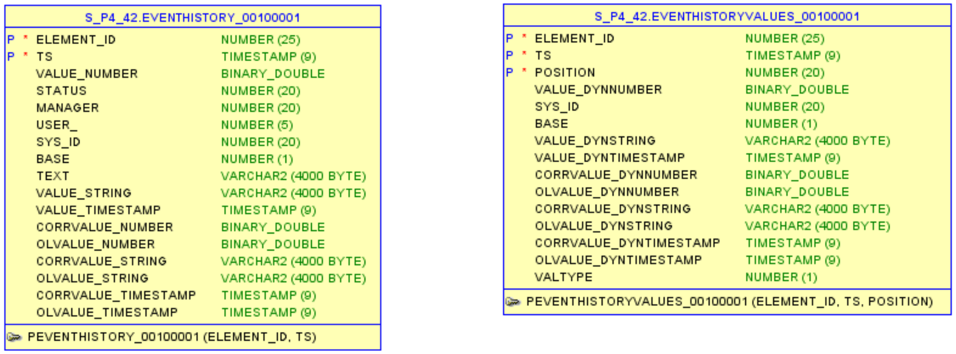
S_P4_42.DPE

P	*	SYS_ID	NUMBER (20)
P	*	DPE_ID	NUMBER (20)
P	*	DP_ID	NUMBER (20)
		DPENAME	VARCHAR2 (4000 BYTE)

PK_DPE (SYS_ID, DP_ID, DPE_ID)

IN_DPE_NAME (DPENAME)
PK_DPE (SYS_ID, DP_ID, DPE_ID)

Metadata



S_P4_42.EVENTHISTORY_00100001

P	*	ELEMENT_ID	NUMBER (25)
P	*	TS	TIMESTAMP (9)
		VALUE_NUMBER	BINARY_DOUBLE
		STATUS	NUMBER (20)
		MANAGER	NUMBER (20)
		USER_	NUMBER (5)
		SYS_ID	NUMBER (20)
		BASE	NUMBER (1)
		TEXT	VARCHAR2 (4000 BYTE)
		VALUE_STRING	VARCHAR2 (4000 BYTE)
		VALUE_TIMESTAMP	TIMESTAMP (9)
		CORRVALUE_NUMBER	BINARY_DOUBLE
		OLVALUE_NUMBER	BINARY_DOUBLE
		CORRVALUE_STRING	VARCHAR2 (4000 BYTE)
		OLVALUE_STRING	VARCHAR2 (4000 BYTE)
		CORRVALUE_TIMESTAMP	TIMESTAMP (9)
		OLVALUE_TIMESTAMP	TIMESTAMP (9)

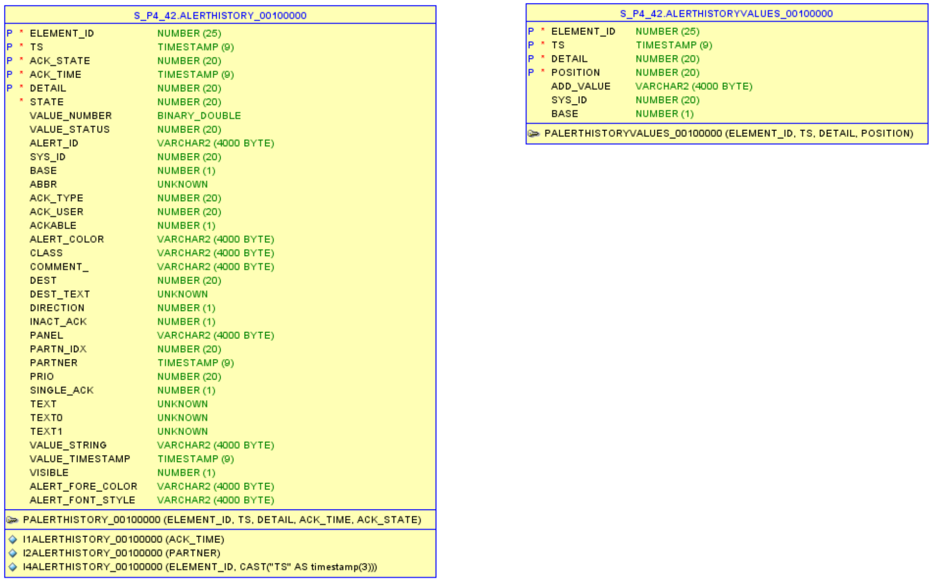
PEVENTHISTORY_00100001 (ELEMENT_ID, TS)

S_P4_42.EVENTHISTORYVALUES_00100001

P	*	ELEMENT_ID	NUMBER (25)
P	*	TS	TIMESTAMP (9)
P	*	POSITION	NUMBER (20)
		VALUE_DYNNUMBER	BINARY_DOUBLE
		SYS_ID	NUMBER (20)
		BASE	NUMBER (1)
		VALUE_DYNSTRING	VARCHAR2 (4000 BYTE)
		VALUE_DYNTIMESTAMP	TIMESTAMP (9)
		CORRVALUE_DYNNUMBER	BINARY_DOUBLE
		OLVALUE_DYNNUMBER	BINARY_DOUBLE
		CORRVALUE_DYNSTRING	VARCHAR2 (4000 BYTE)
		OLVALUE_DYNSTRING	VARCHAR2 (4000 BYTE)
		CORRVALUE_DYNTIMESTAMP	TIMESTAMP (9)
		OLVALUE_DYNTIMESTAMP	TIMESTAMP (9)
		VALTYPE	NUMBER (1)

PEVENTHISTORYVALUES_00100001 (ELEMENT_ID, TS, POSITION)

Events (datapoint changes)



S_P4_42.ALERTHISTORY_00100000

P	*	ELEMENT_ID	NUMBER (25)
P	*	TS	TIMESTAMP (9)
P	*	ACK_STATE	NUMBER (20)
P	*	ACK_TIME	TIMESTAMP (9)
P	*	DETAIL	NUMBER (20)
	*	STATE	NUMBER (20)
		VALUE_NUMBER	BINARY_DOUBLE
		VALUE_STATUS	NUMBER (20)
		ALERT_ID	VARCHAR2 (4000 BYTE)
		SYS_ID	NUMBER (20)
		BASE	NUMBER (1)
		ABBR	UNKNOWN
		ACK_TYPE	NUMBER (20)
		ACK_USER	NUMBER (20)
		ACKABLE	NUMBER (1)
		ALERT_COLOR	VARCHAR2 (4000 BYTE)
		CLASS	VARCHAR2 (4000 BYTE)
		COMMENT_	VARCHAR2 (4000 BYTE)
		DEST	NUMBER (20)
		DEST_TEXT	UNKNOWN
		DIRECTION	NUMBER (1)
		INACT_ACK	NUMBER (1)
		PANEL	VARCHAR2 (4000 BYTE)
		PARTN_IDX	NUMBER (20)
		PARTNER	TIMESTAMP (9)
		PRIO	NUMBER (20)
		SINGLE_ACK	NUMBER (1)
		TEXT	UNKNOWN
		TEXT0	UNKNOWN
		TEXT1	UNKNOWN
		VALUE_STRING	VARCHAR2 (4000 BYTE)
		VALUE_TIMESTAMP	TIMESTAMP (9)
		VISIBLE	NUMBER (1)
		ALERT_FORE_COLOR	VARCHAR2 (4000 BYTE)
		ALERT_FONT_STYLE	VARCHAR2 (4000 BYTE)

PALERTHISTORY_00100000 (ELEMENT_ID, TS, DETAIL, ACK_TIME, ACK_STATE)

I1ALERTHISTORY_00100000 (ACK_TIME)
I2ALERTHISTORY_00100000 (PARTNER)
I4ALERTHISTORY_00100000 (ELEMENT_ID, CAST("TS" AS timestamp(3)))

S_P4_42.ALERTHISTORYVALUES_00100000

P	*	ELEMENT_ID	NUMBER (25)
P	*	TS	TIMESTAMP (9)
P	*	DETAIL	NUMBER (20)
P	*	POSITION	NUMBER (20)
		ADD_VALUE	VARCHAR2 (4000 BYTE)
		SYS_ID	NUMBER (20)
		BASE	NUMBER (1)

PALERTHISTORYVALUES_00100000 (ELEMENT_ID, TS, DETAIL, POSITION)

Alarms

+ tables for configuration, logging, etc.

ALARM and EVENT history tables

- Value changes are stored in EVENT history tables
  - Index Organized Tables on (`ELEMENT_ID`, `TS`)
  - Partitioned on TS, typically daily
  - Number of kept partitions defines the data retention

- Alarms are stored in ALERT history tables
  - Index Organized Tables on (`ELEMENT_ID`, `TS`)
  - Support for partitioning on TS is planned in the next version of the schema

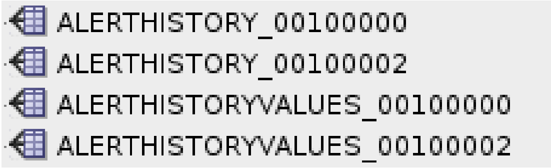
- `VEVENTSCREEN` and `VALERTSCREEN` views join history tables with current datapoint metadata (stored in the `ELEMENTS` table) and are used in the queries executed from managers when `queryFunction` is disabled

```
SELECT ... FROM
Sys1.VEVENTSCREEN WHERE ...;
```

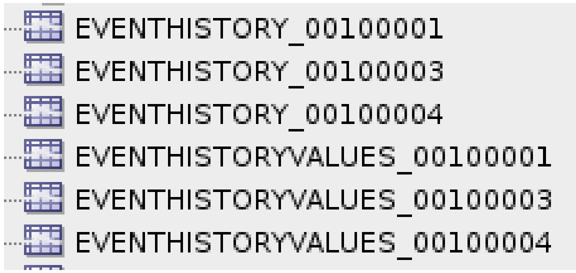
Archives and archive groups

- EVENT and ALERT are different archive groups
  - Each archived datapoint has an archive group selected (in the _archive config)
  - Different sets of columns can be used in different archive groups
  - Each archive group has its own data retention policy

- EVENTHISTORY*, ALERTHISTORY* tables are different archives within the same archive group
  - The mechanism can be used as a substitute for partitioning
  - Makes upgrades easier (e.g. switching from partitioning on system number to partitioning on timestamps)
  - Archives in a group are *glued* by <group_name>HISTORY views



ALERTHISTORY_00100000
ALERTHISTORY_00100002
ALERTHISTORYVALUES_00100000
ALERTHISTORYVALUES_00100002



EVENTHISTORY_00100001
EVENTHISTORY_00100003
EVENTHISTORY_00100004
EVENTHISTORYVALUES_00100001
EVENTHISTORYVALUES_00100003
EVENTHISTORYVALUES_00100004

Metadata storage, fwRDBAPI and the problem it solves

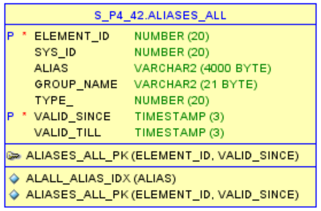
- Metadata tables (ELEMENTS, DPT, DP, DPE) in the schema only store data for datapoints that are currently archived

- The problem
  1. Every newly created datapoint in WinCC OA gets a new, monotonically increasing ID
  2. This ID is used on the Oracle schema side to uniquely identify all its elements
  3. When a datapoint is deleted and recreated with the same name (comment or alias), access to the existing history (saved with a different ID) is lost

- In order to remedy this, fwRDBAPI component enables queries on a specific DP name, comment or alias retrieve events that were archived with different IDs

fwRDBAPI – details

- Trigger on the ELEMENTS table keeps history of metadata in ELEMENTS_ALL, ALIASES_ALL, COMMENTS_ALL tables

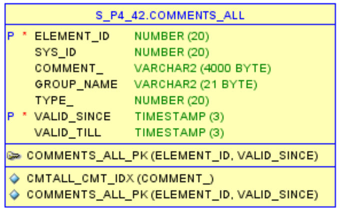


S_P4_42.ALIASES_ALL

P	*	ELEMENT_ID	NUMBER (20)
		SYS_ID	NUMBER (20)
		ALIAS	VARCHAR2 (4000 BYTE)
		GROUP_NAME	VARCHAR2 (21 BYTE)
		TYPE_	NUMBER (20)
P	*	VALID_SINCE	TIMESTAMP (3)
		VALID_TILL	TIMESTAMP (3)

ALIASES_ALL_PK (ELEMENT_ID, VALID_SINCE)

ALALL_ALIAS_IDX (ALIAS)
ALIASES_ALL_PK (ELEMENT_ID, VALID_SINCE)

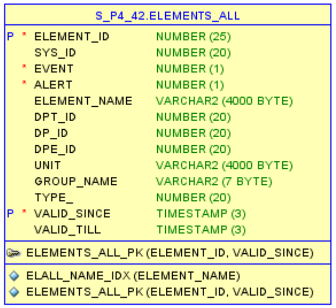


S_P4_42.COMMENTS_ALL

P	*	ELEMENT_ID	NUMBER (20)
		SYS_ID	NUMBER (20)
		COMMENT_	VARCHAR2 (4000 BYTE)
		GROUP_NAME	VARCHAR2 (21 BYTE)
		TYPE_	NUMBER (20)
P	*	VALID_SINCE	TIMESTAMP (3)
		VALID_TILL	TIMESTAMP (3)

COMMENTS_ALL_PK (ELEMENT_ID, VALID_SINCE)

CMTALL_CMT_IDX (COMMENT_)
COMMENTS_ALL_PK (ELEMENT_ID, VALID_SINCE)



S_P4_42.ELEMENTS_ALL

P	*	ELEMENT_ID	NUMBER (25)
		SYS_ID	NUMBER (20)
	*	EVENT	NUMBER (1)
	*	ALERT	NUMBER (1)
		ELEMENT_NAME	VARCHAR2 (4000 BYTE)
		DPT_ID	NUMBER (20)
		DP_ID	NUMBER (20)
		DPE_ID	NUMBER (20)
		UNIT	VARCHAR2 (4000 BYTE)
		GROUP_NAME	VARCHAR2 (7 BYTE)
		TYPE_	NUMBER (20)
P	*	VALID_SINCE	TIMESTAMP (3)
		VALID_TILL	TIMESTAMP (3)

ELEMENTS_ALL_PK (ELEMENT_ID, VALID_SINCE)

ELALL_NAME_IDX (ELEMENT_NAME)
ELEMENTS_ALL_PK (ELEMENT_ID, VALID_SINCE)

- When queryFunction option is enabled in the config file, events or alarms are retrieved using PL/SQL functions from rdbHelpers package in the schema, instead of using the default SQL queries generated by the UI/CTRL manager

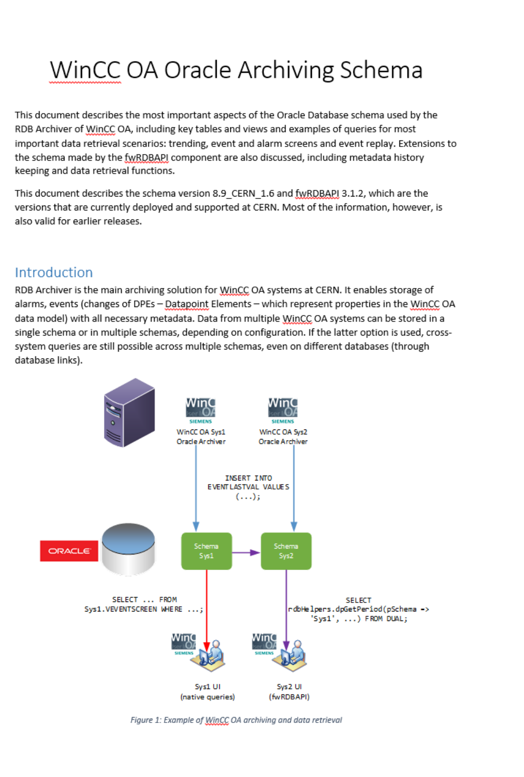
```
      SELECT ... FROM
Sys1.VEVENTSCREEN WHERE ...;
```

```
                 SELECT
rdbHelpers.dpGetPeriod(pSchema =>
      'Sys1', ...) FROM DUAL;
```

- We are aiming at integrating the functionality of fwRDBAPI directly in the schema in the NextGeneration Archiver

Want to learn more?

- More detailed description of the schema and the most important queries is available in EDMS 2013374 (working copy)



WinCC OA Oracle Archiving Schema

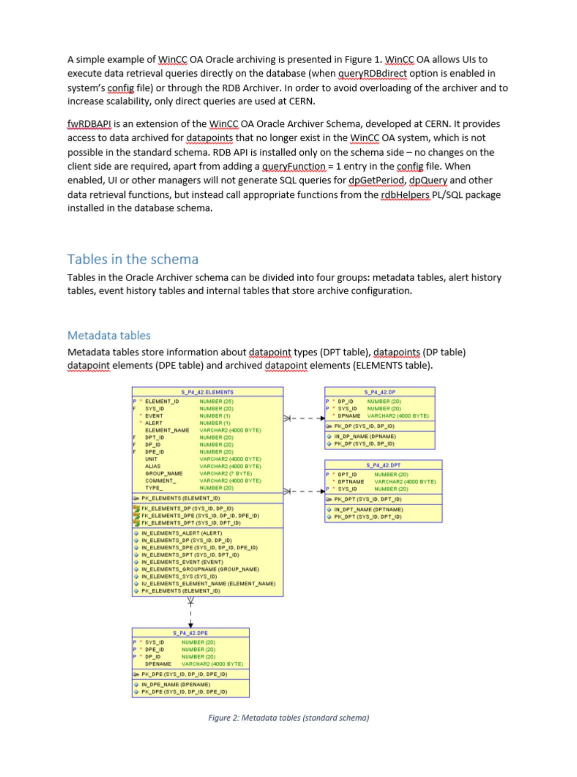
This document describes the most important aspects of the Oracle Database schema used by the RDB Archiver of WinCC OA, including key tables and views and examples of queries for most important data retrieval scenarios: trending, event and alarm screens and event replay. Extensions to the schema made by the fwRDBAPI component are also discussed, including metadata history keeping and data retrieval functions.

This document describes the schema version 8.9_CERN_1.6 and fwRDBAPI 3.1.2, which are the versions that are currently deployed and supported at CERN. Most of the information, however, is also valid for earlier releases.

Introduction

RDB Archiver is the main archiving solution for WinCC OA systems at CERN. It enables storage of alarms, events (changes of DPEs – Datapoint Elements – which represent properties in the WinCC OA data model) with all necessary metadata. Data from multiple WinCC OA systems can be stored in a single schema or in multiple schemas, depending on configuration. If the latter option is used, cross-system queries are still possible across multiple schemas, even on different databases (through database links).

Figure 1: Example of WinCC OA archiving and data retrieval



A simple example of WinCC OA Oracle archiving is presented in Figure 1. WinCC OA allows UIs to execute data retrieval queries directly on the database (when queryRDBdirect option is enabled in system's config file) or through the RDB Archiver. In order to avoid overloading of the archiver and to increase scalability, only direct queries are used at CERN.

fwRDBAPI is an extension of the WinCC OA Oracle Archiver Schema, developed at CERN. It provides access to data archived for datapoints that no longer exist in the WinCC OA system, which is not possible in the standard schema. RDB API is installed only on the schema side – no changes on the client side are required, apart from adding a queryFunction = 1 entry in the config file. When enabled, UI or other managers will not generate SQL queries for dpGetPeriod, dpQuery and other data retrieval functions, but instead call appropriate functions from the rdbHelpers PL/SQL package installed in the database schema.

Tables in the schema

Tables in the Oracle Archiver schema can be divided into four groups: metadata tables, alert history tables, event history tables and internal tables that store archive configuration.

Metadata tables

Metadata tables store information about datapoint types (DPT table), datapoints (DP table) datapoint elements (DPE table) and archived datapoint elements (ELEMENTS table).

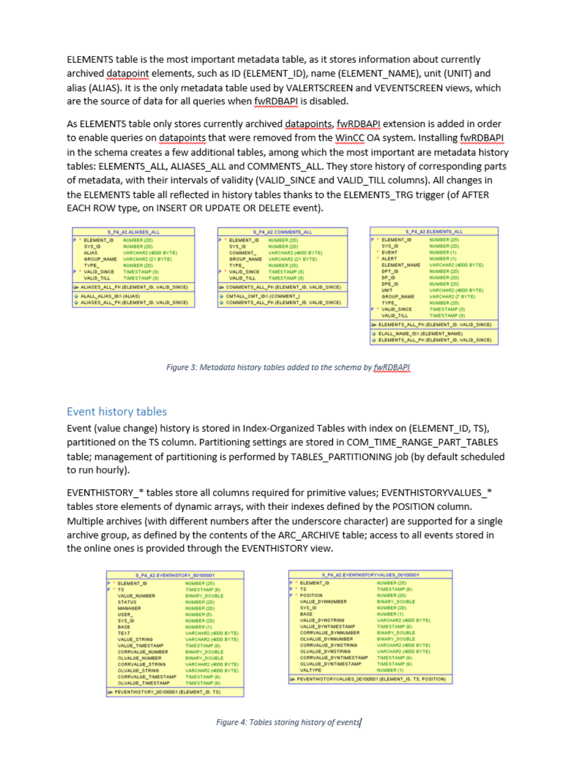
S_P4_42.ELEMENTS		
P *	ELEMENT_ID	NUMBER (25)
F	SYS_ID	NUMBER (20)
*	EVENT	NUMBER (1)
*	ALERT	NUMBER (1)
	ELEMENT_NAME	VARCHAR2 (4000 BYTE)
F	DPT_ID	NUMBER (20)
F	DP_ID	NUMBER (20)
F	DPE_ID	NUMBER (20)
	UNIT	VARCHAR2 (4000 BYTE)
	ALIAS	VARCHAR2 (4000 BYTE)
	GROUP_NAME	VARCHAR2 (7 BYTE)
	COMMENT_	VARCHAR2 (4000 BYTE)
	TYPE_	NUMBER (20)
	PK_ELEMENTS (ELEMENT_ID)	
	FK_ELEMENTS_DP (SYS_ID, DP_ID)	
	FK_ELEMENTS_DPE (SYS_ID, DP_ID, DPE_ID)	
	FK_ELEMENTS_DPT (SYS_ID, DPT_ID)	
	IN_ELEMENTS_ALERT (ALERT)	
	IN_ELEMENTS_DP (SYS_ID, DP_ID)	
	IN_ELEMENTS_DPE (SYS_ID, DP_ID, DPE_ID)	
	IN_ELEMENTS_DPT (SYS_ID, DPT_ID)	
	IN_ELEMENTS_EVENT (EVENT)	
	IN_ELEMENTS_GROUPNAME (GROUP_NAME)	
	IN_ELEMENTS_SYS (SYS_ID)	
	IU_ELEMENTS_ELEMENT_NAME (ELEMENT_NAME)	
	PK_ELEMENTS (ELEMENT_ID)	

S_P4_42.DP		
P *	DP_ID	NUMBER (20)
P *	SYS_ID	NUMBER (20)
*	DPNAME	VARCHAR2 (4000 BYTE)
	PK_DP (SYS_ID, DP_ID)	
	IN_DP_NAME (DPNAME)	
	PK_DP (SYS_ID, DP_ID)	

S_P4_42.DPT		
P *	DPT_ID	NUMBER (20)
*	DPTNAME	VARCHAR2 (4000 BYTE)
P *	SYS_ID	NUMBER (20)
	PK_DPT (SYS_ID, DPT_ID)	
	IN_DPT_NAME (DPTNAME)	
	PK_DPT (SYS_ID, DPT_ID)	

S_P4_42.DPE		
P *	SYS_ID	NUMBER (20)
P *	DPE_ID	NUMBER (20)
P *	DP_ID	NUMBER (20)
	DPENAME	VARCHAR2 (4000 BYTE)
	PK_DPE (SYS_ID, DP_ID, DPE_ID)	
	IN_DPE_NAME (DPENAME)	
	PK_DPE (SYS_ID, DP_ID, DPE_ID)	

Figure 2: Metadata tables (standard schema)



ELEMENTS table is the most important metadata table, as it stores information about currently archived datapoint elements, such as ID (ELEMENT_ID), name (ELEMENT_NAME), unit (UNIT) and alias (ALIAS). It is the only metadata table used by VALERTSCREEN and VEVENTSCREEN views, which are the source of data for all queries when fwRDBAPI is disabled.

As ELEMENTS table only stores currently archived datapoints, fwRDBAPI extension is added in order to enable queries on datapoints that were removed from the WinCC OA system. Installing fwRDBAPI in the schema creates a few additional tables, among which the most important are metadata history tables: ELEMENTS_ALL, ALIASES_ALL and COMMENTS_ALL. They store history of corresponding parts of metadata, with their intervals of validity (VALID_SINCE and VALID_TILL columns). All changes in the ELEMENTS table all reflected in history tables thanks to the ELEMENTS_TRG trigger (of AFTER EACH ROW type, on INSERT OR UPDATE OR DELETE event).

S_P4_42.ALIASES_ALL		
P *	ELEMENT_ID	NUMBER (20)
	SYS_ID	NUMBER (20)
	ALIAS	VARCHAR2 (4000 BYTE)
	GROUP_NAME	VARCHAR2 (21 BYTE)
	TYPE_	NUMBER (20)
P *	VALID_SINCE	TIMESTAMP (3)
	VALID_TILL	TIMESTAMP (3)
	ALIASES_ALL_PK (ELEMENT_ID, VALID_SINCE)	
	ALALL_ALIAS_IDX (ALIAS)	
	ALIASES_ALL_PK (ELEMENT_ID, VALID_SINCE)	

S_P4_42.COMMENTS_ALL		
P *	ELEMENT_ID	NUMBER (20)
	SYS_ID	NUMBER (20)
	COMMENT_	VARCHAR2 (4000 BYTE)
	GROUP_NAME	VARCHAR2 (21 BYTE)
	TYPE_	NUMBER (20)
P *	VALID_SINCE	TIMESTAMP (3)
	VALID_TILL	TIMESTAMP (3)
	COMMENTS_ALL_PK (ELEMENT_ID, VALID_SINCE)	
	CMTALL_CMT_IDX (COMMENT_)	
	COMMENTS_ALL_PK (ELEMENT_ID, VALID_SINCE)	

S_P4_42.ELEMENTS_ALL		
P *	ELEMENT_ID	NUMBER (25)
	SYS_ID	NUMBER (20)
*	EVENT	NUMBER (1)
*	ALERT	NUMBER (1)
	ELEMENT_NAME	VARCHAR2 (4000 BYTE)
	DPT_ID	NUMBER (20)
	DP_ID	NUMBER (20)
	DPE_ID	NUMBER (20)
	UNIT	VARCHAR2 (4000 BYTE)
	GROUP_NAME	VARCHAR2 (7 BYTE)
	TYPE_	NUMBER (20)
P *	VALID_SINCE	TIMESTAMP (3)
	VALID_TILL	TIMESTAMP (3)
	ELEMENTS_ALL_PK (ELEMENT_ID, VALID_SINCE)	
	ELALL_NAME_IDX (ELEMENT_NAME)	
	ELEMENTS_ALL_PK (ELEMENT_ID, VALID_SINCE)	

Figure 3: Metadata history tables added to the schema by fwRDBAPI

Event history tables

Event (value change) history is stored in Index-Organized Tables with index on (ELEMENT_ID, TS), partitioned on the TS column. Partitioning settings are stored in COM_TIME_RANGE_PART_TABLES table; management of partitioning is performed by TABLES_PARTITIONING job (by default scheduled to run hourly).

EVENTHISTORY_* tables store all columns required for primitive values; EVENTHISTORYVALUES_* tables store elements of dynamic arrays, with their indexes defined by the POSITION column. Multiple archives (with different numbers after the underscore character) are supported for a single archive group, as defined by the contents of the ARC_ARCHIVE table; access to all events stored in the online ones is provided through the EVENTHISTORY view.

S_P4_42.EVENTHISTORY_00100001		
P *	ELEMENT_ID	NUMBER (25)
P *	TS	TIMESTAMP (9)
	VALUE_NUMBER	BINARY_DOUBLE
	STATUS	NUMBER (20)
	MANAGER	NUMBER (20)
	USER_	NUMBER (5)
	SYS_ID	NUMBER (20)
	BASE	NUMBER (1)
	TEXT	VARCHAR2 (4000 BYTE)
	VALUE_STRING	VARCHAR2 (4000 BYTE)
	VALUE_TIMESTAMP	TIMESTAMP (9)
	CORRVALUE_NUMBER	BINARY_DOUBLE
	OLVALUE_NUMBER	BINARY_DOUBLE
	CORRVALUE_STRING	VARCHAR2 (4000 BYTE)
	OLVALUE_STRING	VARCHAR2 (4000 BYTE)
	CORRVALUE_TIMESTAMP	TIMESTAMP (9)
	OLVALUE_TIMESTAMP	TIMESTAMP (9)
	PEVENTHISTORY_00100001 (ELEMENT_ID, TS)	

S_P4_42.EVENTHISTORYVALUES_00100001		
P *	ELEMENT_ID	NUMBER (25)
P *	TS	TIMESTAMP (9)
P *	POSITION	NUMBER (20)
	VALUE_DYNNUMBER	BINARY_DOUBLE
	SYS_ID	NUMBER (20)
	BASE	NUMBER (1)
	VALUE_DYNSTRING	VARCHAR2 (4000 BYTE)
	VALUE_DYNTIMESTAMP	TIMESTAMP (9)
	CORRVALUE_DYNNUMBER	BINARY_DOUBLE
	OLVALUE_DYNNUMBER	BINARY_DOUBLE
	CORRVALUE_DYNSTRING	VARCHAR2 (4000 BYTE)
	OLVALUE_DYNSTRING	VARCHAR2 (4000 BYTE)
	CORRVALUE_DYNTIMESTAMP	TIMESTAMP (9)
	OLVALUE_DYNTIMESTAMP	TIMESTAMP (9)
	VALTYPE	NUMBER (1)
	PEVENTHISTORYVALUES_00100001 (ELEMENT_ID, TS, POSITION)	

Figure 4: Tables storing history of events

Thank you for your attention!