INTEGRIGY

Oracle Database 21c
New Security Features

January 28, 2021

Stephen Kost
Chief Technology Officer
Integrigy Corporation

Phil Reimann
Director of Business Development
Integrigy Corporation

About Integrigy

ERP Applications

Oracle E-Business Suite
and PeopleSoft

Databases

Oracle, Microsoft SQL Server,
DB2, Sybase, MySQL, NoSQL

Products

AppSentry

ERP Application and Database
Security Auditing Tool

Validates and Audits Security

AppDefend

Enterprise Application Firewall
for Oracle E-Business Suite
and PeopleSoft

Protects Oracle EBS & PeopleSoft

Services

Verify Security

Security Assessments

ERP, Database, Sensitive Data, Pen Testing

Ensure Compliance

Compliance Assistance

SOX, PCI, HIPAA, GLBA

Build Security

Security Design Services

Auditing, Encryption, DMZ

Integrigy Research Team

ERP Application and Database Security Research

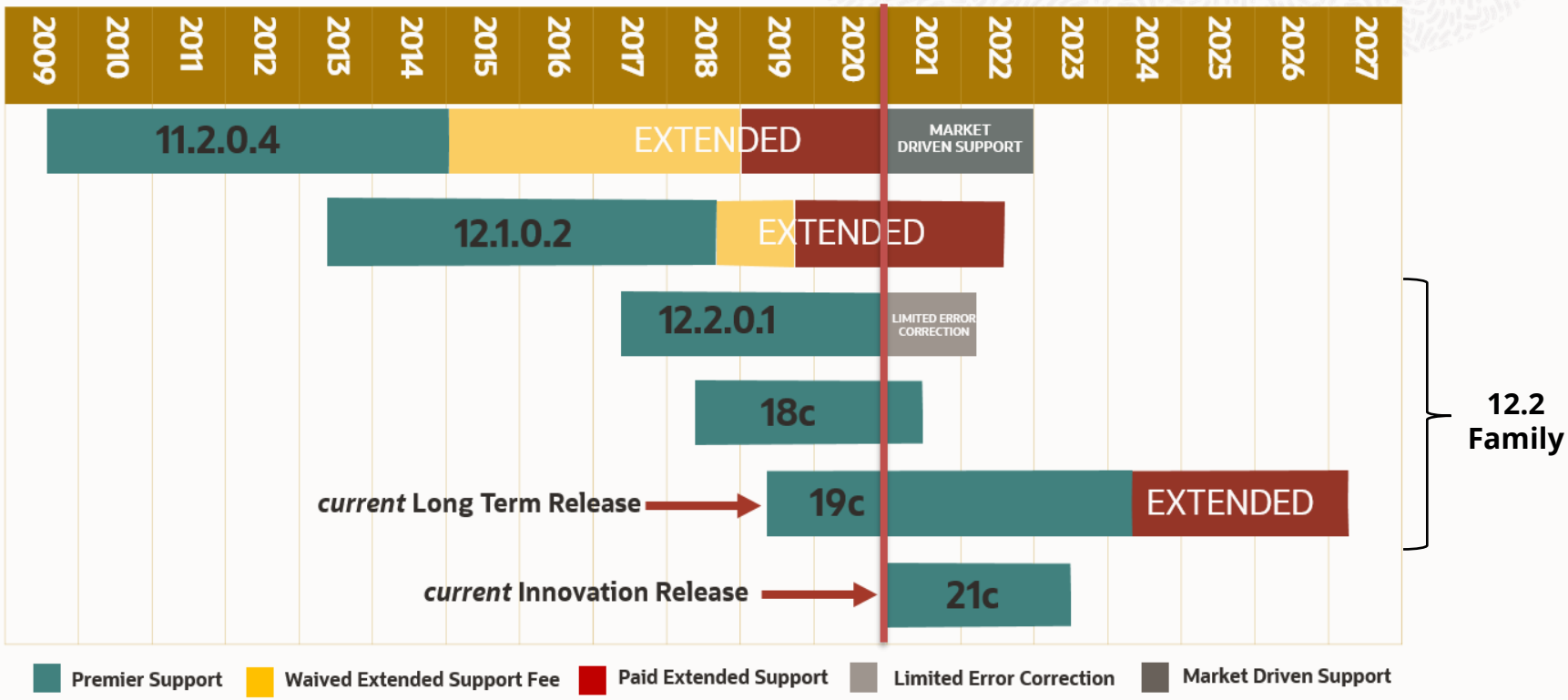
ORACLE
Gold Partner

Agenda

1. Oracle 21c Blockchain Table
2. Oracle 21c Unified Auditing Enhancements
3. Oracle 21c Gradual Database Password Rollover
4. Oracle 21c Other New Security Features
5. Q & A

Oracle Database Releases

Database Releases and Support Timelines



Source – Oracle Database (RDBMS) Releases Support Status Summary (Doc ID 161818.1) – January 15, 2021

Agenda

1. Oracle 21c Blockchain Table
2. Oracle 21c Unified Auditing Enhancements
3. Oracle 21c Gradual Database Password Rollover
4. Oracle 21c Other New Security Features
5. Q & A

Oracle Blockchain Tables

- Insert-only, tamper proof table
- Rows are chained using a cryptographic hashing approach
- Data cannot be modified by DBAs or other users
- Optional row signing by user for additional fraud protection
- Generally, operates as a standard database table
- Common use cases are for audit trails, compliance data, ledgers, and chain of custody or provenance information
- **Available in 19.10 (January 2021)**

ORDERS TABLE

ID	User	Value	Hash
1	Tom	500	ADSJS
2	Carol	176	%SHS
3	Steve	500	SH@1
4	John	176	DHD3
5	Mike	332	*EGG
6	Sarah	632	AH11
7	Eve	25	LIO$
8	Prisha	850	SHS4

BLOCKCHAIN TABLE

Blockchain Table Creation

`CREATE BLOCKCHAIN TABLE <table> (<columns)`	▪ Create table DDL similar to normal tables
`NO DROP [ UNTIL <0+> DAYS IDLE ]`	▪ `NO DROP` without days will prevent table from ever being dropped ▪ Use `1 DAYS` during testing so table can be dropped ▪ Don't use `0 DAYS` as this may cause errors
`NO DELETE { [ LOCKED ] | (UNTIL <16+> DAYS AFTER INSERT [ LOCKED ]) }`	▪ `NO DELETE` prevents rows from ever being deleted – cannot be changed ▪ `UNTIL number DAYS AFTER INSERT` prevents rows from deleted for x days ▪ `LOCKED` does not allow setting to be changed ▪ Retention periods can only be increased
`HASHING USING sha2_512 VERSION v1`	▪ `sha2_512` hash and `v1` version are fixed in this version

Blockchain Table DDL and DML

`DROP TABLE`	▪ Cannot drop until after NO DROP days has expired ▪ ORA-05723: drop blockchain table <> not allowed
`ALTER TABLE`	▪ Cannot modify table structure (add, drop, modify columns) or move tablespace ▪ ORA-05715: operation not allowed on the blockchain table
`DROP TABLESPACE`	▪ ORA-05723: drop blockchain table <> not allowed
`TRUNCATE TABLE`	▪ Never allowed ▪ ORA-05715: operation not allowed on the blockchain table
`UPDATE`	▪ Never allowed ▪ ORA-05715: operation not allowed on the blockchain table
`DELETE`	▪ Never allowed – use DBMS_BLOCKCHAIN_TABLE.DELETE_EXPIRED_ROWS ▪ ORA-05715: operation not allowed on the blockchain table

Add ORA-05723 and ORA-05715 to list of monitored Oracle error messages.

Blockchain Table Hidden Columns

ORABCTAB_INST_ID$	▪ RAC instance ID
ORABCTAB_CHAIN_ID$	▪ Each table may have up to 32 chains (0-31) in current use to allow for parallelism
ORABCTAB_SEQ_NUM$	▪ Row number in a chain
ORABCTAB_CREATION_TIME$	▪ Row creation timestamp, always UTC
ORABCTAB_USER_NUMBER$	▪ USER_ID of the user who inserted row (see DBA_USERS)
ORABCTAB_HASH$	▪ Calculated row hash (SHA2_512, v1)
ORABCTAB_SIGNATURE$, ORABCTAB_SIGNATURE_ALG$, ORABCTAB_SIGNATURE_CERT$	▪ Signature information when row signing is used ▪ Signature based on certificate and ORABCTAB_HASH$
ORABCTAB_SPARE$	▪ Future use

Blockchain Table Data Dictionary View

{CDB|DBA|ALL|USER}_ BLOCKCHAIN_TABLES	▪ Information about blockchain tables including row retention period, table retention period, and hashing algorithm used to chain rows ▪ View over the SYS.BLOCKCHAIN_TABLE$ table

```
SELECT row_retention "Row Retention Period", row_retention_locked "Row Retention Lock",
table_inactivity_retention "Table Retention Period", hash_algorithm "Hash Algorithm"
FROM dba_blockchain_tables WHERE table_name='BANK_LEDGER';

Row Retention Period Row Retention Lock   Table  Retention Period Hash Algorithm
-------------------- ------------------  ------------------------- --------------
                  16 YES                                        31 SHA2_512
```

DBMS_BLOCKCHAIN_TABLE Package

DELETE_EXPIRED_ROWS	▪ Deletes all expired rows or rows prior to a date ▪ Must have DELETE on table in order to delete rows
VERIFY_ROWS	▪ Verifies all rows or rows between two timestamps and optionally signatures for each row ▪ Must have SELECT on table in order to verify rows
VERIFY_TABLE_BLOCKCHAIN	▪ Verifies all rows or rows between two timestamps and optionally signatures for each row ▪ Must have SELECT on table in order to verify rows
SIGN_ROW	▪ Signs a row – user must be the one who inserted the row ▪ A row can only be signed once ▪ Must have INSERT on table in order to sign a row ▪ Must also have SELECT on table to sign a row as instance id, chain id, and row id are required

Blockchain Table Integrity

Oracle Database 21c Database Administrator's Guide

"An important aspect of maintaining the integrity of blockchain table data is to ensure that all rows are intact. Computing a signed digest provides a snapshot of the metadata and data about the last row in all chains at a particular time. You must store this information in [an external] repository. Signed digests generated at various times comprise the input to the DBMS_BLOCKCHAIN_TABLE.VERIFY_TABLE_BLOCKCHAIN procedure. Use this procedure to verify the integrity of rows created between two *specified times."*

- **Use Integrigy AppSentry to periodically retrieve, store, and verify the integrity of all blockchain tables**
  - Detects all blockchain tables
  - Generates a signed digest for each blockchain table
  - Stores the signed digests for each blockchain table
  - Verified the last signed digest to confirm the integrity of the blockchain table

Blockchain Table Observations

- As blockchain tables are new to Oracle Database 21c, should not be immediately used for critical data as issues and bugs may be encountered for the next 12 months.
- Blockchain tables should not be used for high volume transactional tables due to overhead required for the blockchain.
- No margin for error in determining DROP and DELETE days, so blockchain tables must be well designed from the beginning.
- Carefully review the restrictions and limitations for blockchain tables such as allowed data types (e.g., no TIMESTAMP WITH TIME ZONE) and Oracle Data Pump restrictions.
- Signing rows requires a certificate for each database user although most applications use a single database account.
- Use in combination with Oracle TDE tablespace encryption and Table Compression to help protect against direct manipulation of data by editing data files.

Agenda

1. Oracle 21c Blockchain Table
2. **Oracle 21c Unified Auditing Enhancements**
3. Oracle 21c Gradual Database Password Rollover
4. Oracle 21c Other New Security Features
5. Q & A

Oracle 21c Unified Auditing Enhancements

- **Unified Audit Policy Configuration Changes Effective Immediately**
- **Unified Audit Policies Enforced on the Current User**
- **SYSLOG Destination for Common Unified Audit Policies**
- Predefined Unified Audit Policies for DOD Security Technical Implementation Guides Compliance (STIG)
- Auditing for Oracle XML DB HTTP and FTP Services
- Unified Auditing on an Editioned Object Now Applies to All Its Editions
- Deprecation of Traditional Auditing – will be removed in 22c|23c

Unified Audit Policy Configuration Changes Effective Immediately

Old Traditional and Unified Auditing Behavior	▪ Change in traditional auditing or unified audit policy only takes effect for new database sessions
New Unified Auditing Behavior	▪ Change in unified auditing policy takes effect immediately in the current session and all other sessions ▪ Exceptionally useful if potential malicious logons are discovered

Unified Audit Policies Enforced on the Current User

- **When is the current user different from the login user –**
  - Definer rights procedure execution
  - Trigger execution
  - Functions and procedures that are executed during the evaluation of views

Old Traditional and Unified Auditing Behavior	▪ Traditional auditing and unified audit policies are enforced on the user who owned the top-level user session (that is, the login user session) in which the SQL statement is executed
New Unified Auditing Behavior	▪ Unified audit policies are enforced on the current user who executes the SQL statement ▪ Full visibility into use of privileges

SYSLOG Destination for Common Unified Audit Policies

- **New initialization parameter UNIFIED_AUDIT_COMMON_SYSTEMLOG**
  - CDB level initialization parameter
  - UNIFIED_AUDIT_SYSTEMLOG is a PDB level initialization parameter
- **Enables all unified audit records from common unified audit policies to be consolidated**
- **Only on UNIX platforms, not Windows**

Agenda

1 Oracle 21c Blockchain Table

2 Oracle 21c Unified Auditing Enhancements

3 **Oracle 21c Gradual Database Password Rollover**

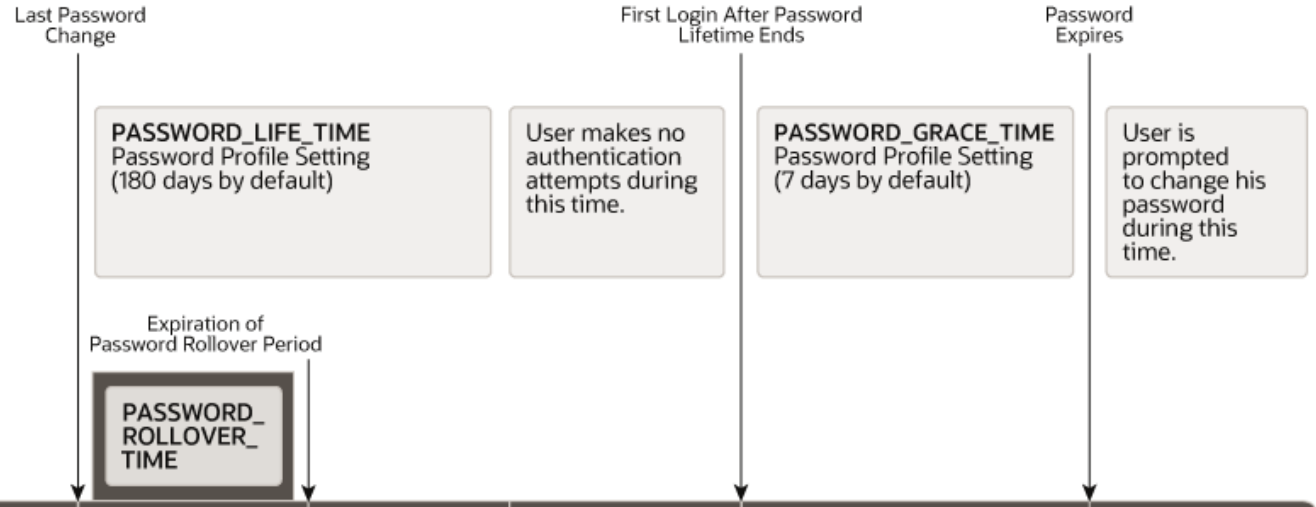
4 Oracle 21c Other New Security Features

5 Q & A

Gradual Database Password Rollover

- **Allows for database passwords to be gracefully changed to eliminate typical downtime required for database password changes**
  - Effectively a database account has two passwords simultaneously
- **Password Rollover is enabled using database user profiles**
  - New `PASSWORD_ROLLOVER_TIME` profile parameter
- **Intended only for database service accounts**
- **Only for database authenticated accounts**
  - Not GLOBAL, an EXTERNAL or NO AUTHENTICATION
  - Not Centrally Managed Users
  - Not administrative connections that use external password files

Gradual Database Password Rollover



Phase number:	1a	1b	2	3	4
DBA_USERS. ACCOUNT_STATUS :	OPEN & IN ROLLOVER	OPEN		EXPIRED (GRACE)	EXPIRED
Errors during phase:	None			ORA-28002 : The password will expire in *n* days	ORA-28001 : The password has expired
Prompted for new password?	No			No	Yes

PASSWORD_ROLLOVER_TIME User Profile Parameter

- New user profile parameter `PASSWORD_ROLLOVER_TIME` controls the time the old password is active in rollover.
- Set in days and can be fraction of days (1/24) to the second.

Default	Minimum	Maximum

Disabled
`0 or null`
	One Hour
	`1/24`
		The lower of –
		`1) 60 days`
		`2)PASSWORD_LIFE_TIME`
		`3)PASSWORD_GRACE_TIME`

DBA_USERS

- DBA_USERS ACCOUNT_STATUS column adds new status “IN ROLLOVER”
- The password rollover period begins immediately upon the password change

```
SQL> CREATE USER integrigy_test IDENTIFIED BY Password123# PROFILE integrigy;

User INTEGRIGY_TEST created.

SQL> SELECT username, account_status FROM dba_users WHERE username = 'INTEGRIGY_TEST’;

USERNAME          ACCOUNT_STATUS
_________________ _________________
INTEGRIGY_TEST    OPEN

SQL> ALTER USER integrigy_test IDENTIFIED BY Password1234#;

User INTEGRIGY_TEST altered.

SQL> SELECT username, account_status FROM dba_users WHERE username = 'INTEGRIGY_TEST';

USERNAME          ACCOUNT_STATUS
_________________ ____________________
INTEGRIGY_TEST    OPEN & IN ROLLOVER
```

Gradual Database Password Rollover

Start Gradual Password Rollover (user or admin)	`alter user <> identified by <p_new>;`
	
	`Current password = <p_new>, rollover password = <p_old>`
	▪ Sets the new password and uses the old password for the rollover period

End Gradual Password Rollover (user or admin)	`alter user <> expire password rollover period;` `Current password = <p_new>,` ~~`rollover password = <p_old>`~~ ▪ Expires the rollover period and removes the old password from use

Change Password during Rollover (keep old, set new) (user or admin)	`alter user <> identified by <p_new>;`
	`alter user <> identified by <p_new2>;`
	
	`Current password = <p_new2>, rollover password = <p_old>`
	▪ During the rollover, the old password (rollover password) is constant
	▪ To change rollover password, need to expire password rollover period first

Gradual Database Password Rollover Recommendations

- **Only should be used for active database service and application accounts**
  - Do not use for schema owners
- **Do not use for applications where the application must be shutdown to change the password**
  - Oracle E-Business Suite (see MOS Note ID 1674462.1)
  - PeopleSoft (see MOS Note ID 2559419.1)
- **After the password has been changed in remote application or server, use EXPIRE PASSWORD ROLLOVER PERIOD to remove old password**
- **Keep PASSWORD_ROLLOVER_TIME to a minimum**
  - 1 day should be sufficient for most applications

Agenda

1 Oracle 21c Blockchain Table

2 Oracle 21c Unified Auditing Enhancements

3 Oracle 21c Gradual Database Password Rollover

4 **Oracle 21c Other New Security Features**

5 Q & A

SEC_CASE_SENSITIVE_LOGON Removed

- **SEC_CASE_SENSITIVE_LOGON initialization parameter controls if database passwords are case sensitive (true) or not case sensitive (false)**
  - Introduced in 11g
  - Used for backward compatibility with applications that do not support case sensitive passwords
  - Deprecated in 19c
- **SEC_CASE_SENSITIVE_LOGON removed in 21c**
- **When upgrading, need to review database accounts with only 10g password authentication**

Common Mandatory Profiles

- **New `MANDATORY` option in `CREATE PROFILE`**
  - Mandatory profile is in the CDB root, not modifiable by PDB administrators
  - Only common users in the CDB root with ALTER PROFILE can modify
  - One mandatory profile is assigned to the CBD and all PDBs (ALTER SYSETM)
  - A different mandatory profile may be assigned to individual PDBs (init.ora)
- **Enforce minimum password length and other password complexity in all PDBs with a mandatory profile in the CDB**
  - Password verify function in the CDB mandatory profile executes before the database account profile password verify function for all local and common accounts

Agenda

1. Oracle 21c Blockchain Table
2. Oracle 21c Unified Auditing Enhancements
3. Oracle 21c Gradual Database Password Rollover
4. Oracle 21c Other New Security Features
5. Q & A

Integrigy Contact Information

Stephen Kost
Chief Technology Officer
Integrigy Corporation

web – **www.integrigy.com**

e-mail – **info@integrigy.com**

blog – **integrigy.com/oracle-security-blog**

youtube – **youtube.com/integrigy**

linkedin – **linkedin.com/company/integrigy**

twitter – **twitter.com/integrigy**