CON8737
Oracle GoldenGate 12.2

New Features Deep Dive

Jagdev Dhillon – VP Product Development
Mahesh Subramaniam - Director Product Development
Nick Wagner - Director of PM
Oracle GoldenGate Development
October, 2015

ORACLE OPEN WORLD
October 25–29, 2015
San Francisco

ORACLE

Program Agenda

1. Review & Strategic Initiatives
2. GoldenGate New Features Preview
3. GoldenGate Enhancements
4. Q&A

Oracle GoldenGate and Data Integration

Over 10K Customers Worldwide

Communications

Finance / Banking

Media

Services

Energy/Industrial

Insurance / Health

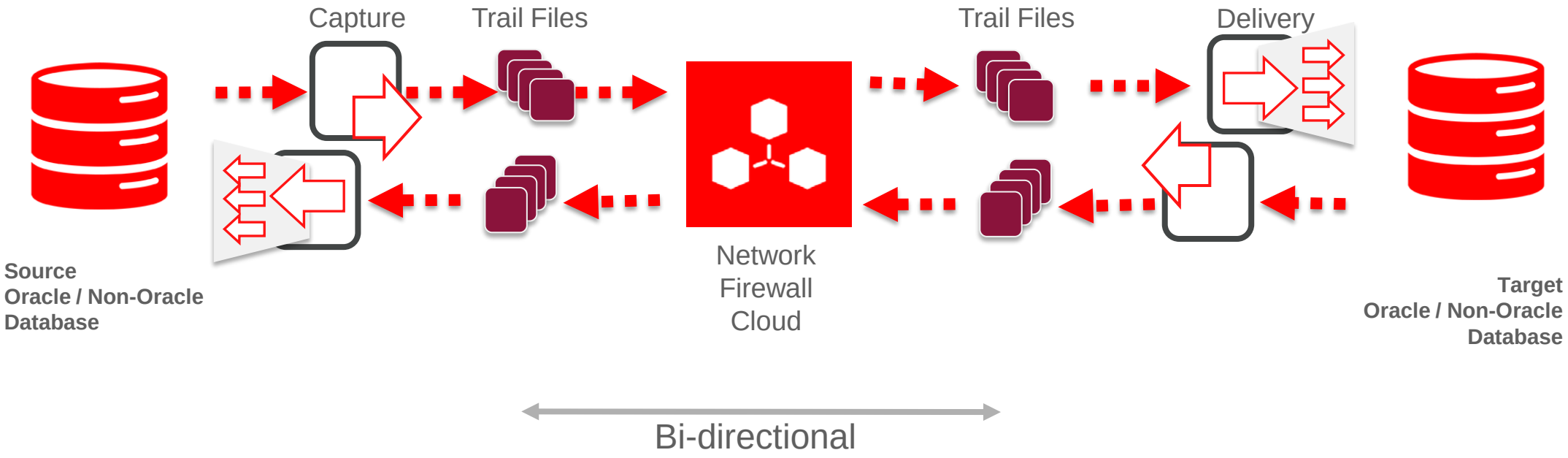
Retail

Other

ORACLE

Oracle GoldenGate

Architecture



Oracle GoldenGate 12.1 Review

Optimized for Oracle Database 12c

Multitenant and Cloud-based Real-Time Replication

Integrated Delivery for the Oracle Database

Leveraging lightweight Streaming API built Exclusively for Oracle GoldenGate

Coordinated Delivery for All Databases

Orchestrates the High-Speed Apply Processes & Simplifies Setup and Management

Improved Ease of Use

Automatic Discard File, Enhanced Debugging, and Schema Wildcarding

Expanded Heterogeneity

12c Brings Support for New Databases and Enhancements to Existing Supported Platforms

Enhanced High Availability

Integration with Data Guard FSFO for Automated & Transparent failover of Components

Tighter Security

Integration with the Oracle Credential Store and Oracle Wallet for encrypted user details

Expanded Oracle Application and Technology Support

Active/Active ATG, Low Downtime E-Business Suite Migrations and Coherence Integration

Oracle GoldenGate 12.1 Patchset Review

Optimized for Oracle Database

Support for Edition Based Redefinition, support for AnyData, and UDT's, CTAS with DML. Capture from ADG (Classic)

Integrated Delivery Enhancements

Dependency aware Batching, Support for Streams DML/DDL Handlers, Error Queue Support

Integrated Extract

Use TAG based filtering for Active/Active, Share mining dictionary for multiple captures

Enhanced Cloud Support

SOCKS V5 support for secure transport of data between cloud and on-premise

Stream to GoldenGate Conversion Utility

Tools on MOS for easier migration from Streams to GoldenGate.

Column Level Character Support

Enable minimal downtime when cleaning up character data to be Unicode compliant using DMU

Strategic Initiatives for Oracle GoldenGate

- Reduce Operational Costs and Complexity
  - Build intelligence directly into components and reduce manual configuration steps
  - Automatic recovery for more failure cases
- Improve Performance, Scalability, Reliability of Replication
  - Improved performance for IE and IR.
- Heterogeneous Support
  - Non-relational targets including Big Data ecosystems (e.g Kafka, HDFS)
  - Better integration with Database HA capabilities.
- Cloud Support
  - Secure support for private, public, and hybrid clouds

Program Agenda

1. Review & Strategic Initiatives
2. GoldenGate New Features Preview
3. GoldenGate Enhancements
4. Q&A

Quick Quiz… Identify the missing parameter

Ext1.prm:

EXTRACT ext1

USERIDALIAS ggs_admin

DDL include mapped

RMTTRAIL $data/ggs12.2/a1

TABLE hr.*;

Rep1.prm

REPLCAT rep1

USERIDALIAS ggs_replicat

DDL include all

MAP hr.*, TARGET hr.*;

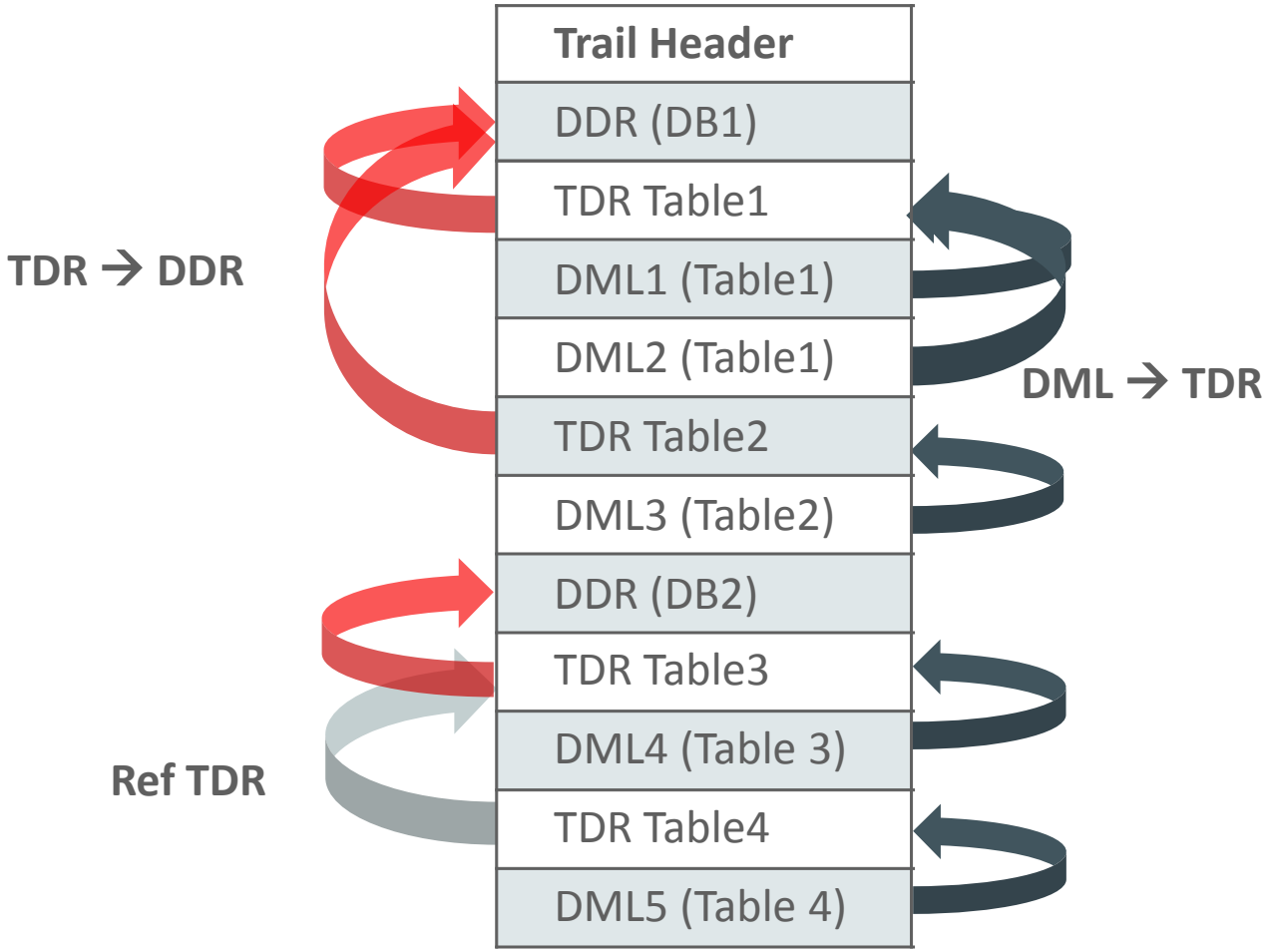
Self-describing Trail Files

No more SOURCEDEFS

- Simpler to configure replication
  - Eliminates the SOURCEDEFS or ASSUMETARGETDEFS parameters
  - Supports replication even if source and target have different structures or different databases
  - Handles multiple catalogs with different character sets and time zones using one trail
  - Ability to configure DDL replication among more than 2 Oracle databases
- Eliminate many manual steps and reduces errors during replication
  - Metadata information in the trail file is accurate unlike ASSUMETARGETDEFS which assumes target table has the same internal structure as source table
- Logdump has been modified to provide additional information

Self-describing Trail Files

No Need for SOURCEDEFS or ASSUMETARGETDEFS



DDR – Database Definition Record
TDR – Table Definition Record

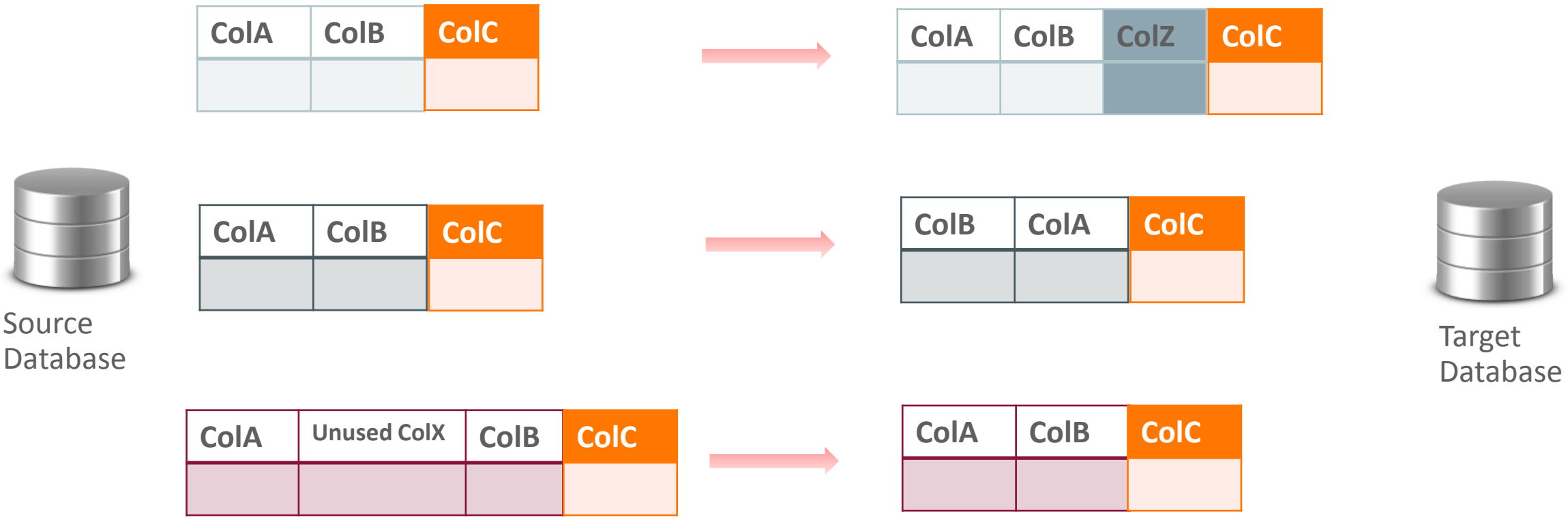
- Metadata records used to interpret DML records instead of SOURCEDEFS or ASSUMETARGETDEFS
- Each trail file contains a Database Definition Record (DDR) before first occurrence of a DML record or a SEQUENCE from a particular database
- Each trail file contains a Table Definition Record (TDR) before first occurrence of a DML record for a particular table
  - TDR contains table and column definition including column number, data types, column lengths, etc.
- DML records have a reference to the TDR and no longer contain the object name
  - Typically results in smaller trail files
- SEQUENCE records have a reference to the DDR and no longer contain the SEQUENCE name

Simplified User Experience

- New Installations
  - Automatically get metadata in trails by default
  - No need to create and maintain source definitions files
  - Easier configuration and manageability
- Existing Installations
  - Metadata in trail generated by default if FORMAT RELEASE 12.2 (Recommended)
    - Ignores SOURCEDEFS and ASSUMETARGETDEFS
    - Use GLOBALS parameter (NO_USE_TRAILDEFS) to retain old behavior of using SOURCEDEFS or ASSUMETARGETDEFS
    - Use SOURCEDEFS OVERRIDE and ASSUMETARGETDEFS OVERRIDE to force old behavior for specific files

Examples of New Use Cases with DDL Replication

Seamless DDL replication between tables with different structures (Oracle-to-Oracle)



Automatic Heartbeat Table

Built-in Mechanism to Monitor End-to-End Replication Lag

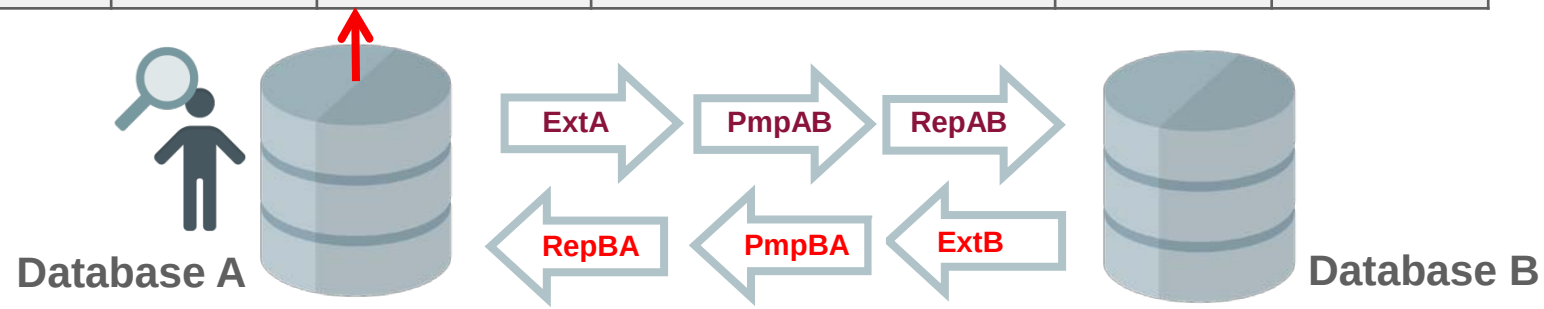
- Intelligent Functionality
  - Automatically discovers replication topology
    - Unidirectional, bi-directional, N-way, …
  - Automatically propagates heartbeats along replication paths
- Database views and tables to view replication lags
  - Shows incoming and outgoing lags in replication paths in each database for active-active scenarios
- Easy to configure
  - Execute GGSCI command **ADD HEARTBEATTABLE** at each database

Simple Bidirectional Replication Example

- Enable Heartbeat functionality by executing GGSCI command '**ADD HEARTBEATTABLE**' at each database
  - Creates required heartbeat tables, views and jobs
  - Updates heartbeat every 60s by default

In Database A: GG_LAG View contents

LOCAL_ DATABASE	REMOTE_ DATABASE	INCOMING_PATH	INCOMING_ LAG	INCOMING_ HEARTBEAT_ AGE	OUTGOING_PATH	OUTGOING_ LAG	OUTGOING_ HEARTBEAT_ AGE
A	B	ExtB==>PmpBA==>RepBA	1.066182	14.150614	ExtA==>PmpAB==>RepAB	1.391718	25.387458



Monitoring Lag using GG_LAG View

Column Name	Data type	Description
LOCAL_DATABASE	VARCHAR2(30)	Local database name
CURRENT_LOCAL_TS	TIMESTAMP(6)	Current timestamp in UTC time zone
REMOTE_DATABASE	VARCHAR2(30)	Remote database name
INCOMING_PATH	VARCHAR2(4000)	Group names on the incoming flow
INCOMING_LAG	NUMBER	Period of time between remote database generating heartbeat and local database receiving heartbeat
INCOMING_HEARTBEAT_AGE	NUMBER	CURRENT_LOCAL_TS minus most recent heartbeat timestamp of remote database
OUTGOING_PATH	VARCHAR2(4000)	Group names on the outgoing flow
OUTGOING_LAG	NUMBER	Period of time between local database generating heartbeat and remote database receiving heartbeat
OUTGOING_HEARTBEAT_AGE	NUMBER	CURRENT_LOCAL_TS minus recent heartbeat timestamp of local database

Additional Tables and Views

- GG_LAG_HISTORY View
  - Historical heartbeat lag

- GG_HEARTBEAT and GG_HEARTBEAT_HISTORY Tables
  - Underlying tables for the views
  - Get lag for each process on the path

Column Name	
INCOMING_HEARTBEAT_TS	} Extract lag
INCOMING_EXTRACT_TS	
INCOMING_ROUTING_TS	} Pump lag
INCOMING_REPLICAT_TS	} Replicat lag
OUTGOING_HEARTBEAT_TS	
OUTGOING_EXTRACT_TS	
OUTGOING_ROUTING_TS	
OUTGOING_REPLICAT_TS	
…	

Parameter Files – Simplified Operational Experience

- New standalone utility **checkprm** for validation
  - Can validate parameter files offline before deployment
  - Can be run on one platform (e.g., Oracle RDBMS on Linux) to validate another platform (e.g., DB2 on z/OS)
- New **INFO PARAM** GGSCI command to obtain definitions of parameters
- New **SEND [process_name] GETPARAMINFO** GGSCI command to get current running parameters including defaulted values
  - Provides framework to dynamically change parameter values in future releases

Checkprm – Offline Parameter File Validation Utility

Example parameter file with errors

```
1  replicat rep1
2  userid password
3  macro #mac
4  begin
5  mapexclude
6  end;
7  ddlerror default discard
8  ddl include all extra stuff
9  ddloptions report
10 discardfile ./dirrpt/rep1.dsc purge, maxbytes 2147483647
11 ASSUMETARGETDEFS
12 #mac()
13 HANDLECOLLISIONS
14 table asdf;
15 map tkggu1.*, TARGET tkggu2.* ;
16 map SYSTEM.TKGG_SRC_HB, TARGET SYSTEM.TKGG_TGT_HB ;
17 map tkggmarkeruser.tkggmarker, target tkggmarkeruser.tkggmarker ;
18 sqlexec "call tkgg_util.tkgg_proc_hb@dbs1('rep1')"
```

Output of running *checkprm* utility

```
$ checkprm test1.prm
(test1.prm) line 2: Parsing error, option [password] for parameter [user_id] is missing a required value.
(test1.prm) line 2: Parsing error, parameter [user_id] is missing a required value.
(test1.prm) line 8: Parsing error, option [include] for parameter [ddl] has unrecognized value "extra".
(test1.prm) line 8: Parsing error, option [include] for parameter [ddl] has unrecognized value "stuff".
(test1.prm) line 10: Parsing error, value "2147483647" is out of legal range (1 - 2147483646) for [max_bytes].
(test1.prm) line 12: Parsing error, parameter [map_exclude] is missing a required value.
Original expression [#mac()] -- Expanded expression [mapexclude]
(test1.prm) line 14: Parameter [table] is not valid for this configuration and will be ignored.

2015-02-21 19:20:21  INFO    OGG-10139  Parameter file test1.prm:  Validity check: FAIL.
```

ORACLE

INFO PARAM: Display Static Information of a Parameter

```
param name   : port
component(s): MGR
mode(s)      : none
platform(s) : all platforms
versions     :
database(s) :
type         : integer
default      : 7809
      range : 1 - 65535
description : TCP IP port number for the Manager process
scope        : global
status       : current
resolve      : merge
mandatory    : false
dynamic      : false
relations    : none
```

This parameter can only be used for the Manager process

Default value and valid range of values

SEND [process] GETPARAMINFO

Runtime Parameter Values including defaults

```
GGSCI (slc08usu) 1> send rep1 getparaminfo
Sending GETPARAMINFO request to REPLICAT REP1 ...

GLOBALS

ggschema                          : gg
trailbyteorder                    : LITTLEENDIAN
checkpointtable                   : gg.chkpt

/scratch/zyin/ggbin/zyin_qabugfix2/dirprm/rep1.prm

replicat                          : rep1
discardfile                       : ./dirrpt/mr.dsc
     purge(discardfile)           : <enabled>
batchsql                          : <enabled>
eofdelay                          : 0
userid                            : gg
     password(userid)             : ********
map                               : gg.src
     target(map)                  : gg.dst
maxtransops                       : 10000
```

Display all parameters loaded from parameter file into Replicat *rep1*, plus those parameters that the *rep1* has accessed so far

Display only one parameter

```
GGSCI (slc08usu) 2> send mgr getparaminfo port
Sending getparaminfo request to MANAGER ...

/scratch/zyin/ggbin/zyin_qabugfix2/dirprm/mgr.prm

port                                : 15000
```

Send the output to the file "mgrfile.out" instead of the console

```
GGSCI (slc08usu) 1> send mgr getparaminfo file mgrfile.out
```

Program Agenda

1. Review & Strategic Initiatives
2. GoldenGate New Features Preview
3. GoldenGate Enhancements
4. Q&A

Transparent Integration with Oracle Clusterware

Achieve GoldenGate high availability in a cluster configuration.

- GoldenGate is managed/monitored by Oracle Clusterware.
- XAG ensures that GoldenGate can tolerate server failures by moving processing to another available server (instance failover in a cluster or Data Guard failover).
- Transparent Integration with Clusterware
  - Before: GoldenGate administrators have to use XAG's AGCTL to manage the GoldenGate instance.
  - After: GoldenGate administrators can continue using GGSCI to start/stop manager (still have to use AGCTL to register GoldenGate instance with Clusterware)

Enable Transparent Integration with Clusterware

- Add parameter “XAG_ENABLE” to GLOBALS to enable this feature.
  - Syntax: **XAG_ENABLE**
  - The feature is disabled by default.
- Use **AGCTL** to register GoldenGate instance with Clusterware
- GGSCI command “**START/STOP MANAGER**” is passed to XAG and the manager is started/stopped by XAG.
- Use “AUTOSTART” and “AUTORESTART” to make sure that ER processes are restarted by the manager when they abend. If an ER process runs into repeated failures on restart, thereby exhausting all restart attempts, XAG will failover the entire GoldenGate instance to another available node.

Fetch from Active Data Guard

- Remove almost all impact from source database.
  - Still need source db for startup validations, registration and some metadata lookups
  - Enable with **FETCH_USER_ID** ggadmin@adg_inst password pwd
    or **FETCHUSERIDALIAS** ggadmin_adginst
- Aware of applied SCN on ADG to ensure fetch consistency
  - **DBOPTIONS [NO_]FETCH_TIMEOUT** <seconds> (Default 30 secs)
  - **DBOPTIONS FETCH_CHECK_FREQ** <seconds> (Default 3 secs)
    - Wait this many seconds between checks for required ADG current_scn.
  - **DBOPTIONS FETCH_RETRY_COUNT** <count>
    - Check ADG this many times before reporting progress.
    - Will report required SCN and current SCN and if MRP is down

New GoldenGate Extended Metrics

- Real-time insight into GoldenGate processes
  - Exposed with a RESTful Interface
  - Ability to integrate with 3rd party products
  - Ability to record metrics for diagnosis by GoldenGate support / development
- New Metrics
  - Status and Configuration Information
  - Process and Thread Level Metrics for Extract, Pump and Replicat
  - Database Statistics for Extract and Replicat & Network Statistics for Pump
  - In-flight transactions and queue statistics for Extract
  - Table statistics for Replicat

New GoldenGate Extended Metrics

Fine-grained Performance Monitoring

- Access to Monitoring Point through Restful Web Services

  http://<hostname>:<mgr_port>/mpoints**x**

- Real-time insight into GoldenGate client programs

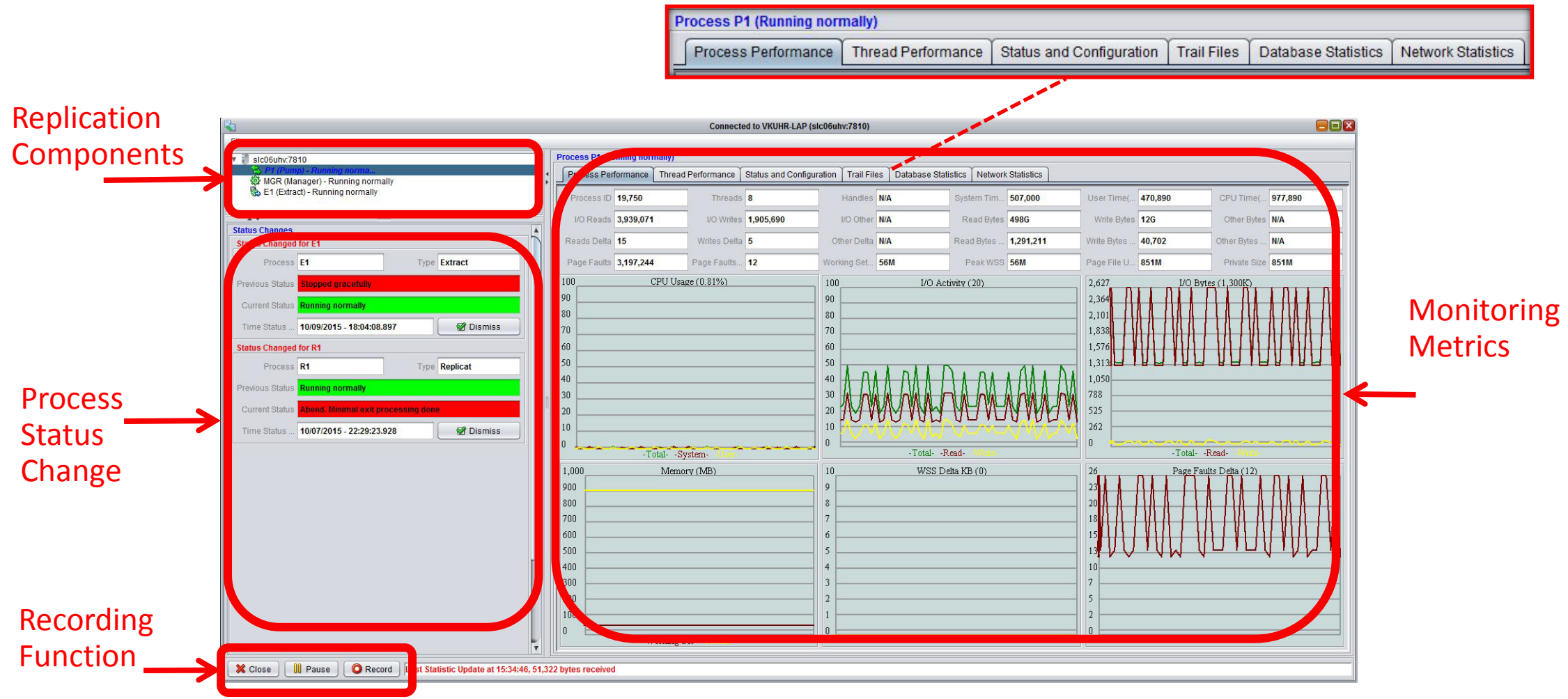
E1 - Thread Performance	
thread-id	23619
thread-name	ReaderThread-1
thread-function	/net/slc06uhv/scratch/vkuhr/view_storage/vkuhr_slc06uhv_v6oc1/tklocal/ggtest/install/extract (ggs::gglib::AsyncReader::AsyncReader::ReaderThread(void*)+0) [0x6db8f0]
thread-start-time	649771430
thread-current-stack	/lib64/libpthread.so.0(read+0x2d) [0x3c0ce0e54d]

- Requires “ENABLEMONITORING” in GLOBALS

Graphical real-time instance monitoring

Utility available on

https://java.net/projects/oracledi/pages/OracleGoldenGate



Oracle Data Pump Integration for Table Instantiation

Integration with Oracle Datapump

- At Source Oracle Database
  - ADD TRANDATA / SCHEMATRANDATA automatically prepares tables
  - Oracle Datapump export will automatically generate import actions to set instantiation CSN for each table at target upon import
- At Target Database
  - Datapump import will populate system tables and views with instantiation CSNs
  - New Replicat parameter (DBOPTIONS ENABLE_INSTANTIATION_FILTERING) to enable table level instantiation filtering
  - Start replicat, who will query instantiation CSN on any new mapping and filter records accordingly
    - Filters out DDL and DML records based on each table's instantiation CSN
    - Eliminates need for HANDLE_COLLISIONS or specification of individual MAP for each imported table with the @FILTER(@GETENV('TRANSACTION','CSN') clause

Simple Usage

1. ADD TRANDATA / SCHEMATRANDATA on tables to be instantiated
2. Stop the Replicat (on the target)
3. Start EXTRACT with proper TABLE statement

4. EXPORT tables using Oracle Datapump
5. Import tables using Oracle Datapump utility
6. Start Replicat with DBOPTIONS ENABLE_INSTANTIATION_FILTERING

Option to set Instantiation CSN manually

- GGSCI command at Target database to set instantiation CSN manually
  - **SET_INSTANTIATION_CSN** <csn> **FOR** <table_name> **FROM** <source_database_name>
    - source_database_name is the GLOBAL_NAME of the source database from query:

      Select global_name from global_name;
  - Simpler alternative to specifying @FILTER(@GETENV('TRANSACTION','CSN')
  - Used when target tables instantiated using alternate mechanism or when source database tables were not prepared prior to export.

Replacing Oracle CDC with Oracle GoldenGate

- Main use case is for feeding Informatica or other ETL tools
- Sample OGG parameter files to fill in additional metadata details on the target
  - Uses INSERTALLRECORDS with Tokens to fill in SCN, timestamp and operation type details
- New subscriptions objects for use with OGG
  - New table to maintain subscription high and low water marks
- Rebuilt PURGE_WINDOW and EXTEND_WINDOW procedures
- Added new procedures for adding and removing subscriptions

Improved Trail File Recovery

- For use when Replicat abends due to missing or corrupt trail file
  - If the trail is corrupt, delete the trail file first
- Any missing trails are now automatically rebuilt by bouncing the Extract Pump.
- Once trail files have been restored, restart the Replicat
  - Do not use NOFILTERDUPTRANSACTIONS
- Requires at least 1 valid, complete trail on the target
  - Due to this, you may want to modify your PURGEOLDEXTRACTS parameter
- Backported to 12.1.2.1.8

Additional New Features

- Support for Invisible Columns (Oracle Only)
  - New parameter - MAPINVISIBLECOLUMNS
  - Requires Oracle Integrated Extract and Oracle 12c
  - The invisible column can be part of an index, including primary key and unique index
- 9 digit trail file sequences
  - New default is 9 digits (AA123456789)

Program Agenda

1. Review & Strategic Initiatives
2. GoldenGate New Features Preview
3. GoldenGate Enhancements
4. Q&A


ORACLE®