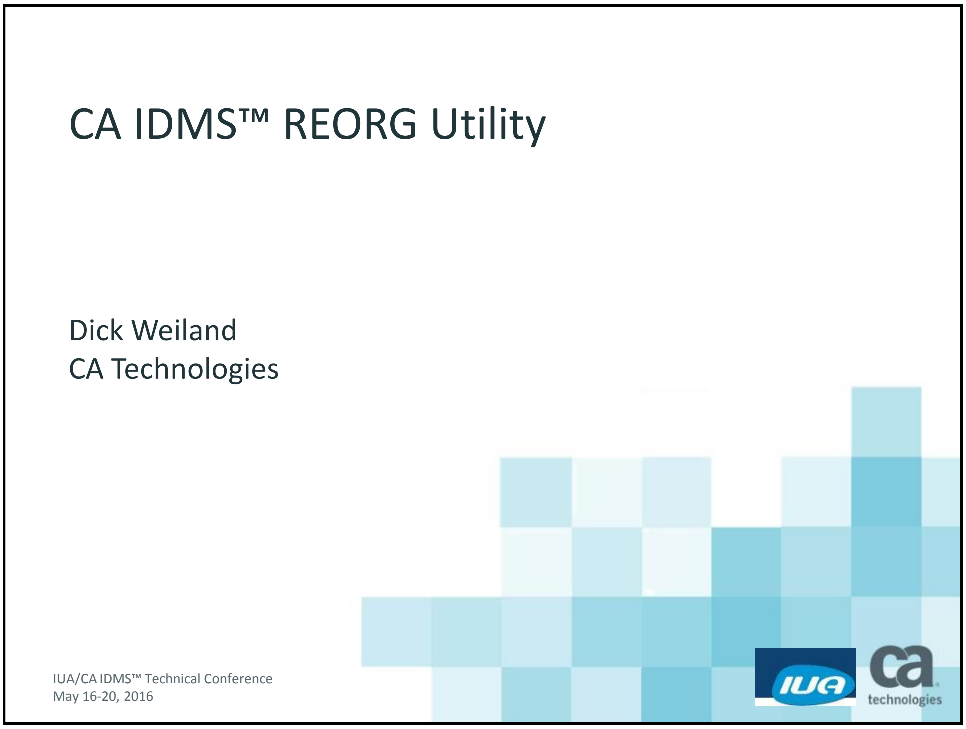
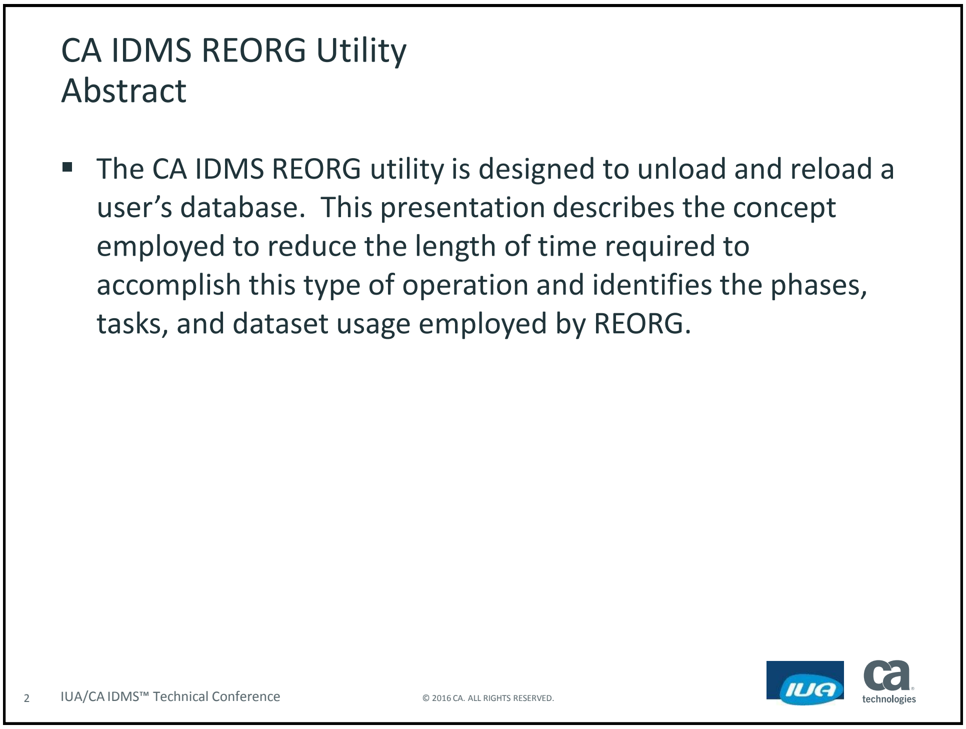


CA IDMS™ REORG Utility

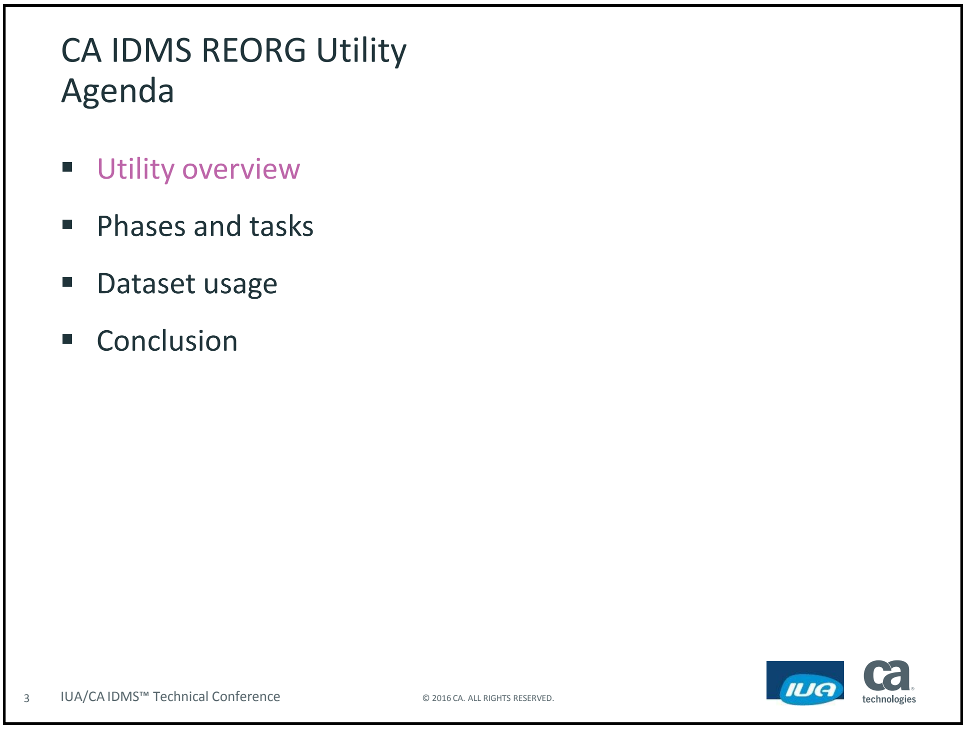
Dick Weiland
CA Technologies

IUA/CA IDMS™ Technical Conference
May 16-20, 2016



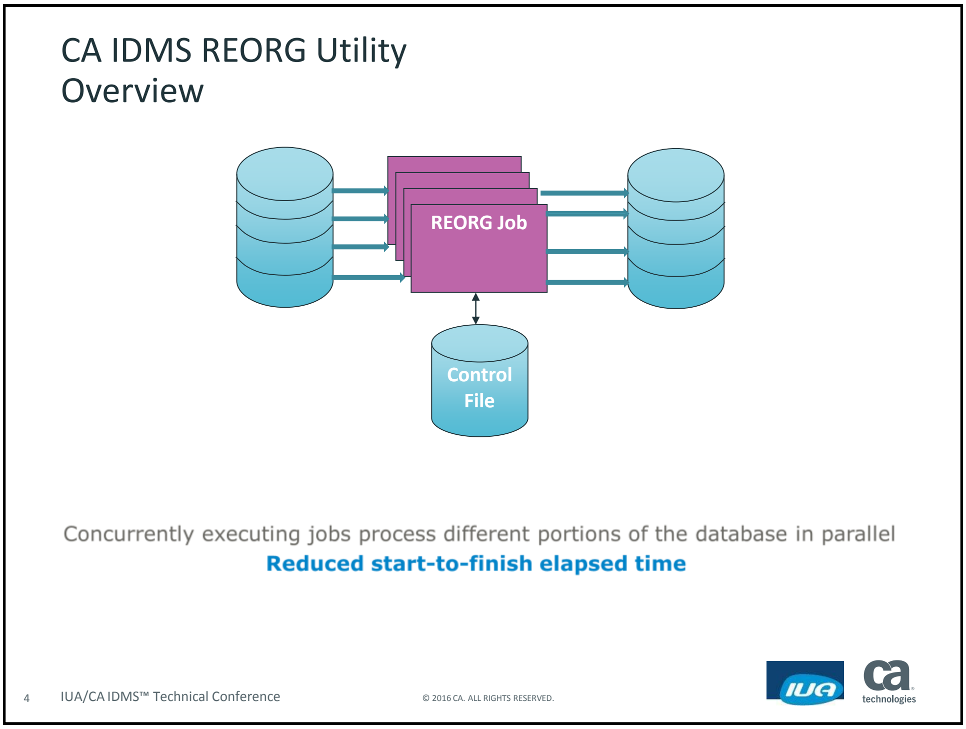
CA IDMS REORG Utility
Abstract

- The CA IDMS REORG utility is designed to unload and reload a user's database. This presentation describes the concept employed to reduce the length of time required to accomplish this type of operation and identifies the phases, tasks, and dataset usage employed by REORG.



CA IDMS REORG Utility
Agenda

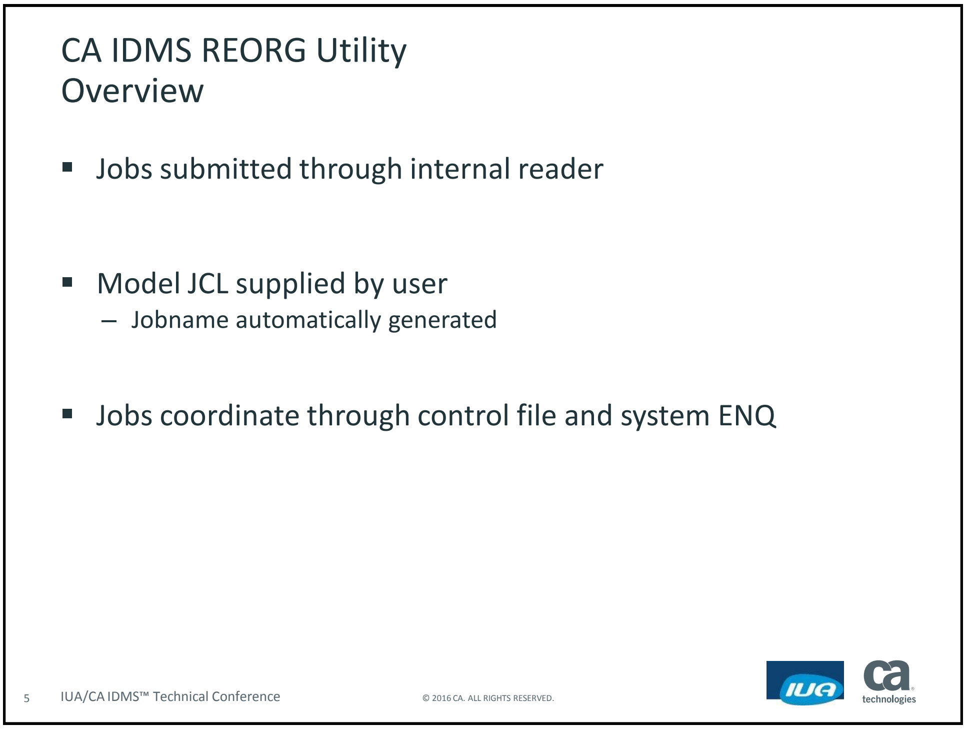
- Utility overview
- Phases and tasks
- Dataset usage
- Conclusion



CA IDMS REORG Utility
Overview

Concurrently executing jobs process different portions of the database in parallel

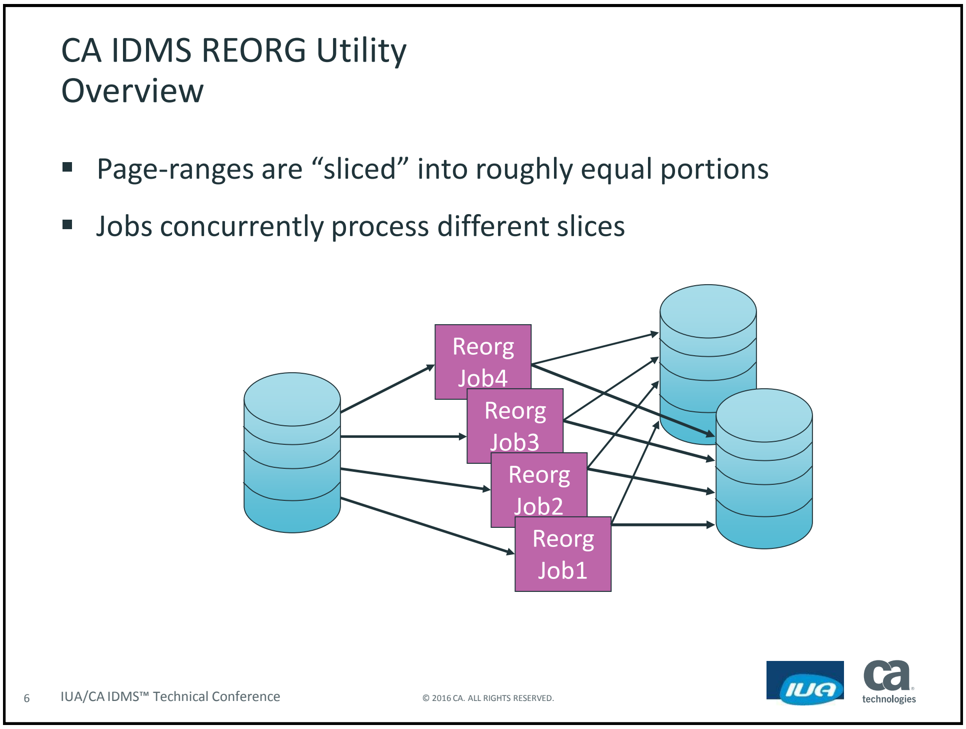
Reduced start-to-finish elapsed time



CA IDMS REORG Utility Overview

- Jobs submitted through internal reader
- Model JCL supplied by user
  - Jobname automatically generated
- Jobs coordinate through control file and system ENQ

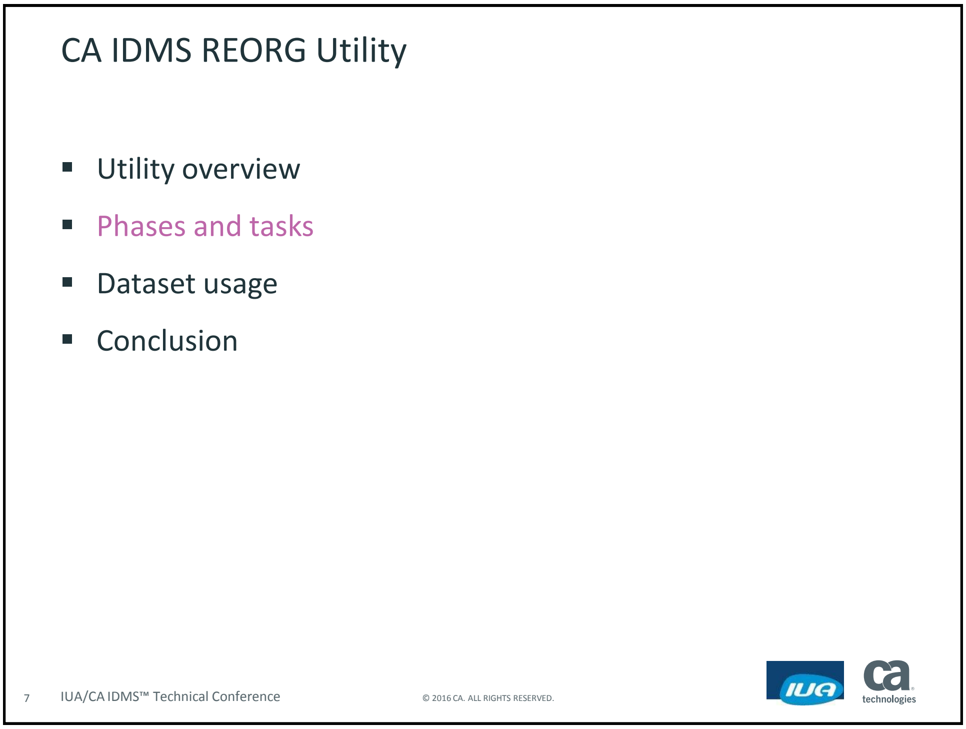
5 IUA/CA IDMS™ Technical Conference



CA IDMS REORG Utility Overview

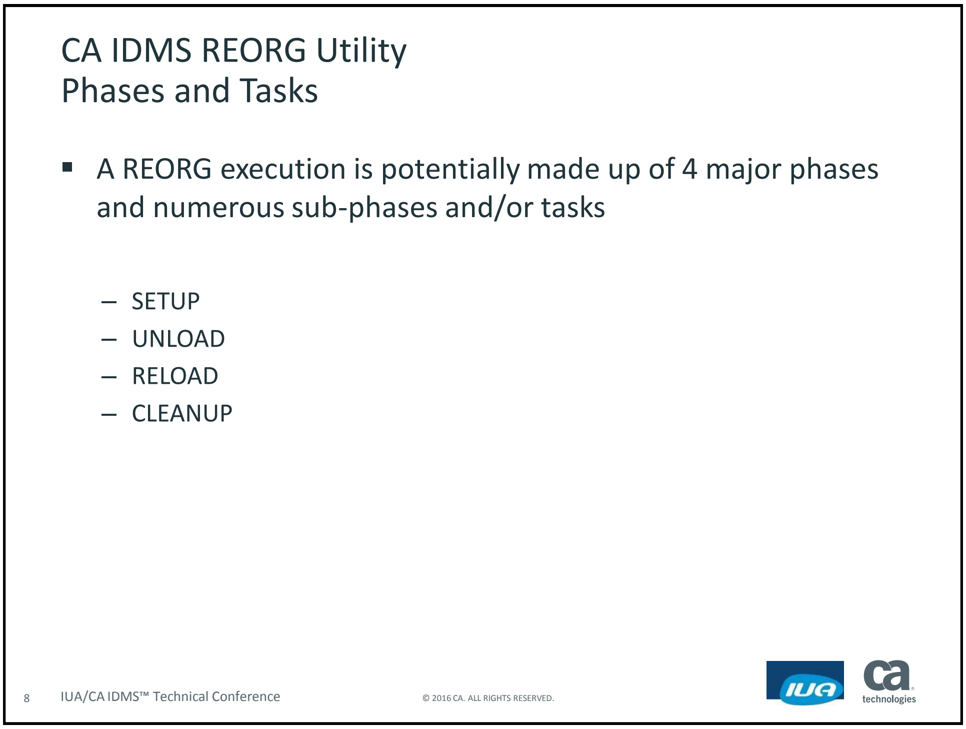
- Page-ranges are “sliced” into roughly equal portions
- Jobs concurrently process different slices

6 IUA/CA IDMS™ Technical Conference



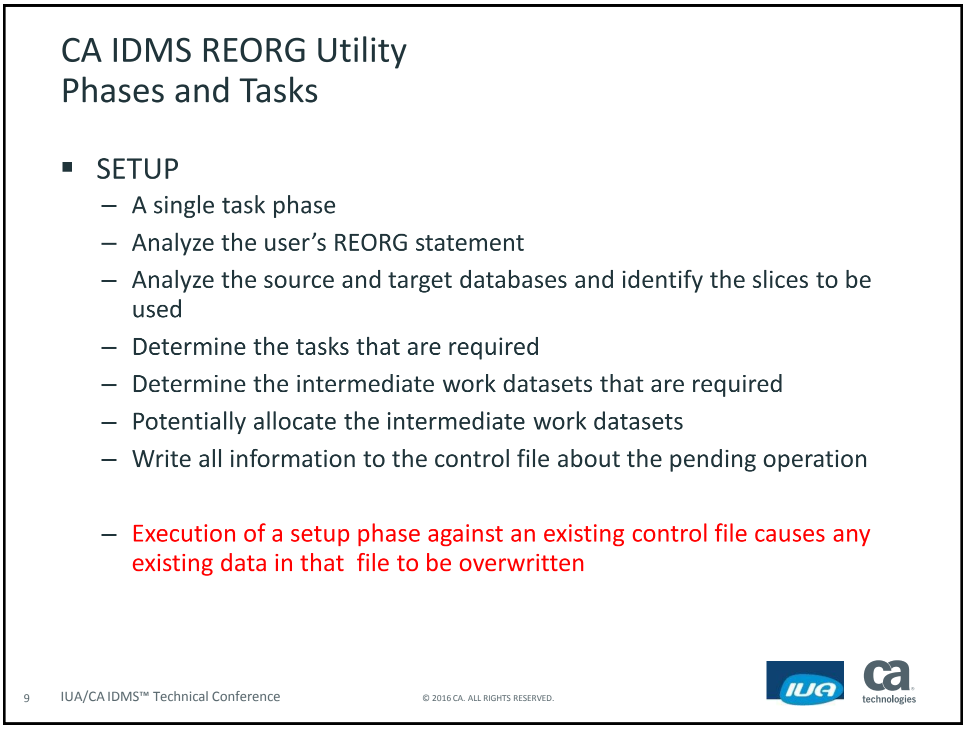
CA IDMS REORG Utility

- Utility overview
- Phases and tasks
- Dataset usage
- Conclusion



CA IDMS REORG Utility
Phases and Tasks

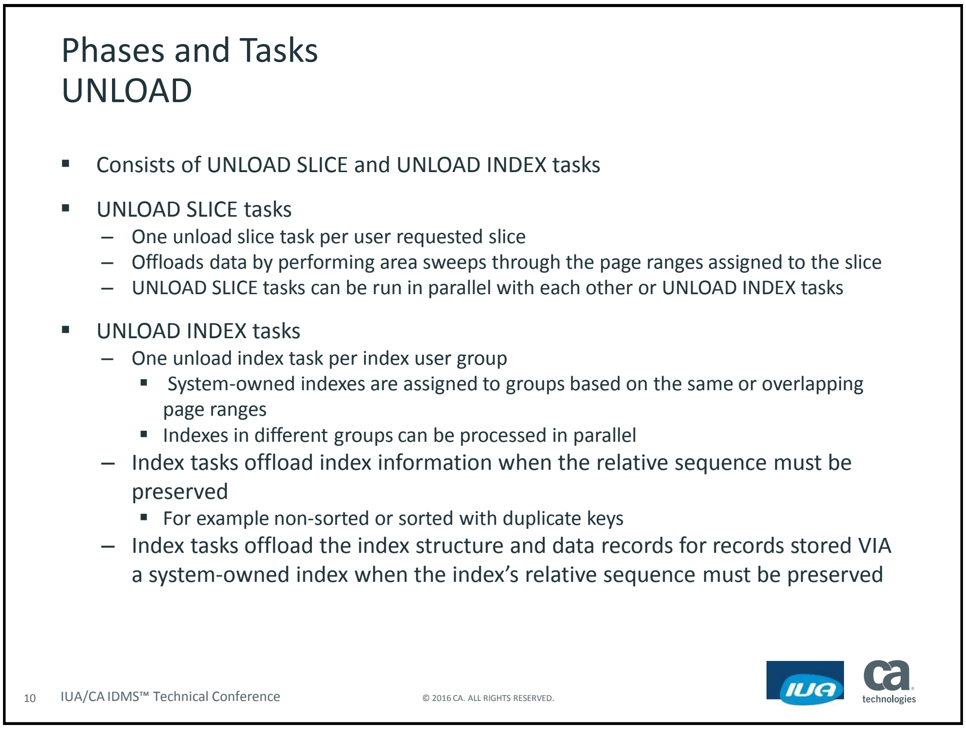
- A REORG execution is potentially made up of 4 major phases and numerous sub-phases and/or tasks
  - SETUP
  - UNLOAD
  - RELOAD
  - CLEANUP



CA IDMS REORG Utility
Phases and Tasks

- SETUP
  - A single task phase
  - Analyze the user's REORG statement
  - Analyze the source and target databases and identify the slices to be used
  - Determine the tasks that are required
  - Determine the intermediate work datasets that are required
  - Potentially allocate the intermediate work datasets
  - Write all information to the control file about the pending operation
  - Execution of a setup phase against an existing control file causes any existing data in that file to be overwritten

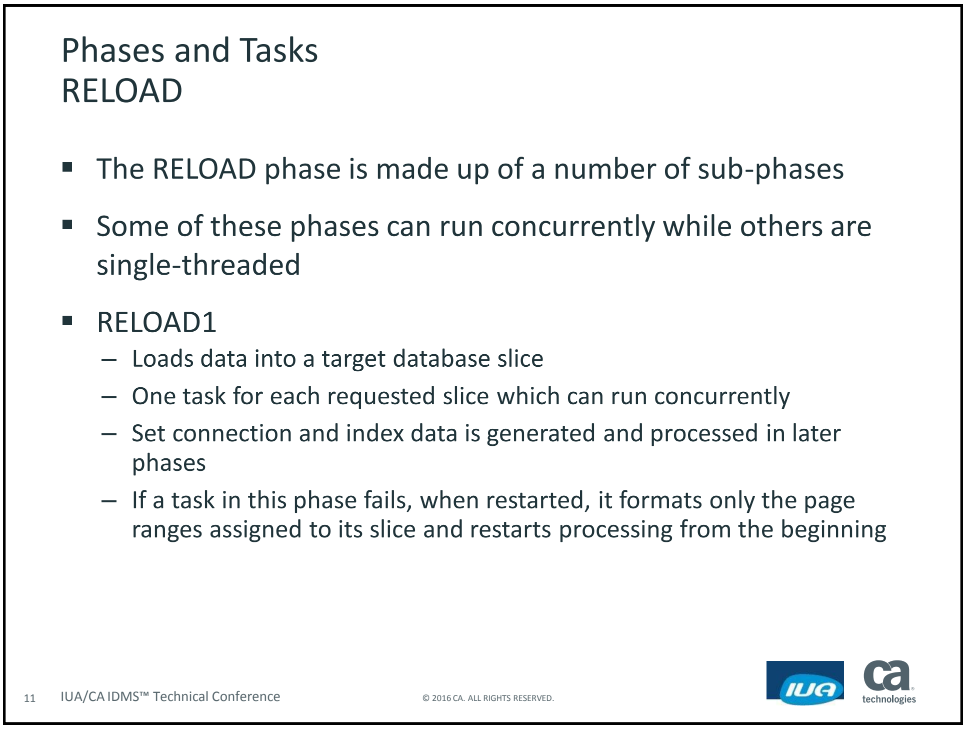
9 IUA/CA IDMS™ Technical Conference



Phases and Tasks
UNLOAD

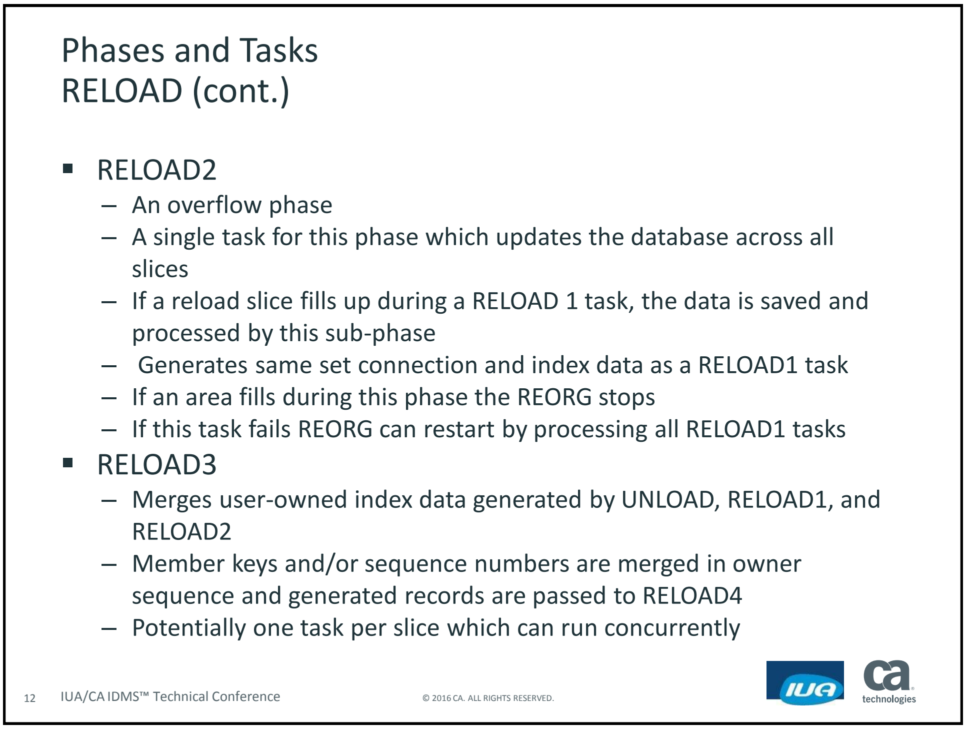
- Consists of UNLOAD SLICE and UNLOAD INDEX tasks
- UNLOAD SLICE tasks
  - One unload slice task per user requested slice
  - Offloads data by performing area sweeps through the page ranges assigned to the slice
  - UNLOAD SLICE tasks can be run in parallel with each other or UNLOAD INDEX tasks
- UNLOAD INDEX tasks
  - One unload index task per index user group
    - System-owned indexes are assigned to groups based on the same or overlapping page ranges
    - Indexes in different groups can be processed in parallel
  - Index tasks offload index information when the relative sequence must be preserved
    - For example non-sorted or sorted with duplicate keys
  - Index tasks offload the index structure and data records for records stored VIA a system-owned index when the index's relative sequence must be preserved

10 IUA/CA IDMS™ Technical Conference



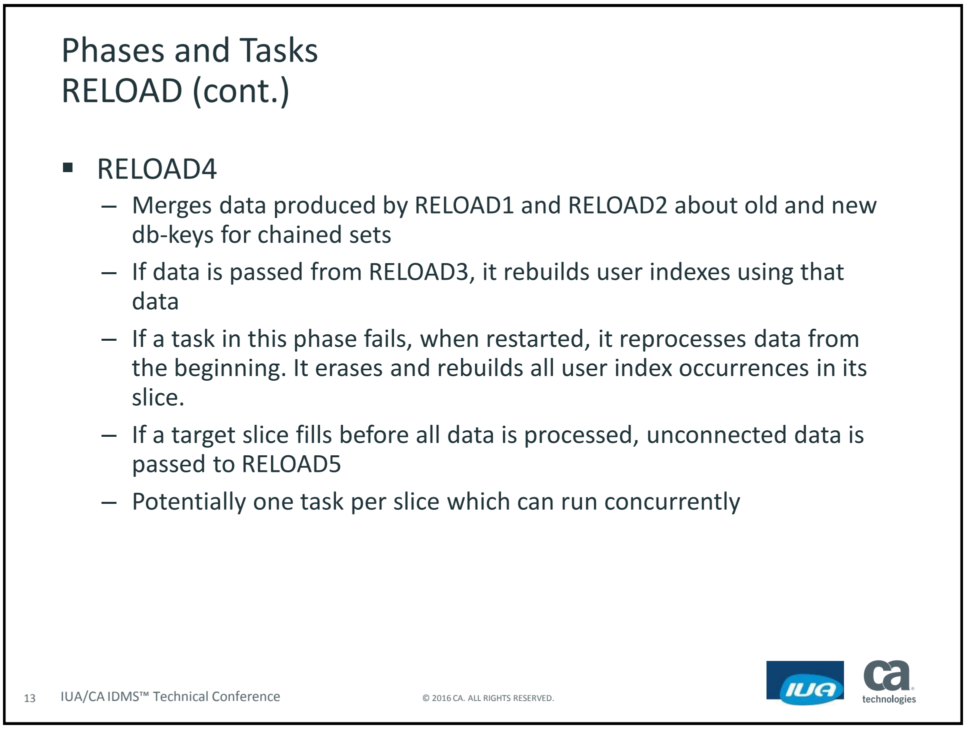
Phases and Tasks
RELOAD

- The RELOAD phase is made up of a number of sub-phases
- Some of these phases can run concurrently while others are single-threaded
- RELOAD1
  - Loads data into a target database slice
  - One task for each requested slice which can run concurrently
  - Set connection and index data is generated and processed in later phases
  - If a task in this phase fails, when restarted, it formats only the page ranges assigned to its slice and restarts processing from the beginning



Phases and Tasks
RELOAD (cont.)

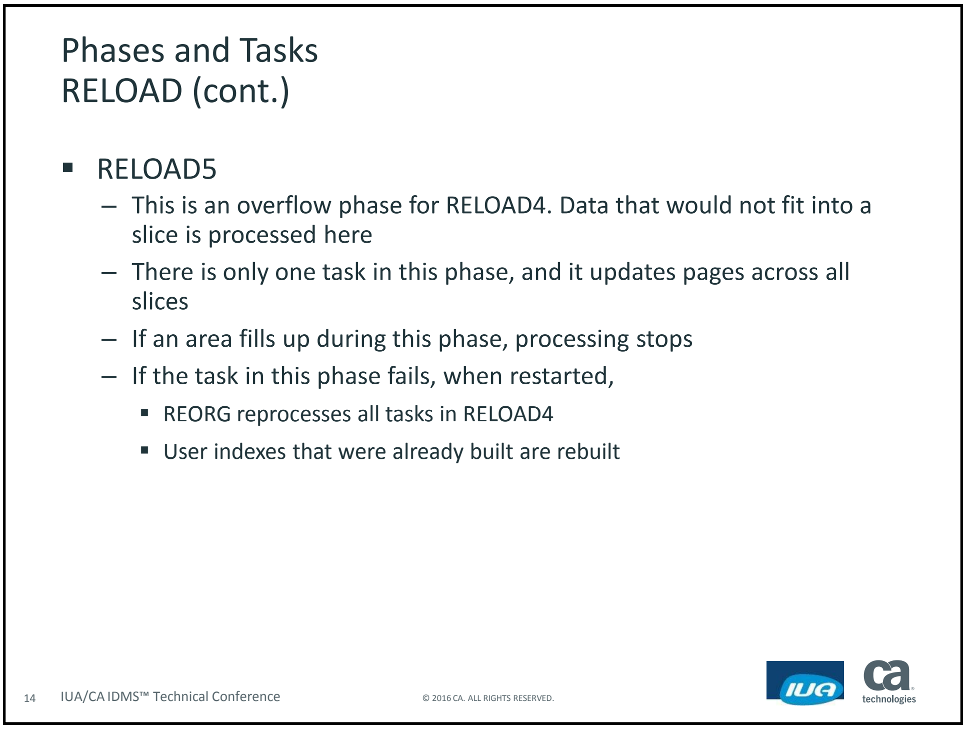
- RELOAD2
  - An overflow phase
  - A single task for this phase which updates the database across all slices
  - If a reload slice fills up during a RELOAD 1 task, the data is saved and processed by this sub-phase
  - Generates same set connection and index data as a RELOAD1 task
  - If an area fills during this phase the REORG stops
  - If this task fails REORG can restart by processing all RELOAD1 tasks
- RELOAD3
  - Merges user-owned index data generated by UNLOAD, RELOAD1, and RELOAD2
  - Member keys and/or sequence numbers are merged in owner sequence and generated records are passed to RELOAD4
  - Potentially one task per slice which can run concurrently



Phases and Tasks RELOAD (cont.)

- RELOAD4
  - Merges data produced by RELOAD1 and RELOAD2 about old and new db-keys for chained sets
  - If data is passed from RELOAD3, it rebuilds user indexes using that data
  - If a task in this phase fails, when restarted, it reprocesses data from the beginning. It erases and rebuilds all user index occurrences in its slice.
  - If a target slice fills before all data is processed, unconnected data is passed to RELOAD5
  - Potentially one task per slice which can run concurrently

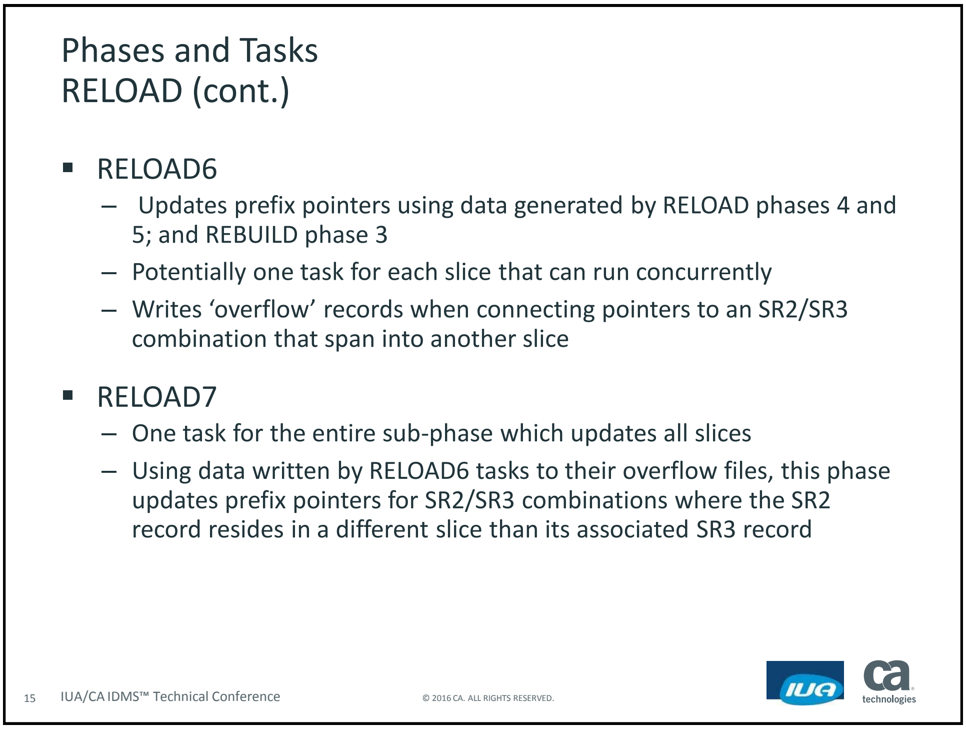
13 IUA/CA IDMS™ Technical Conference



Phases and Tasks RELOAD (cont.)

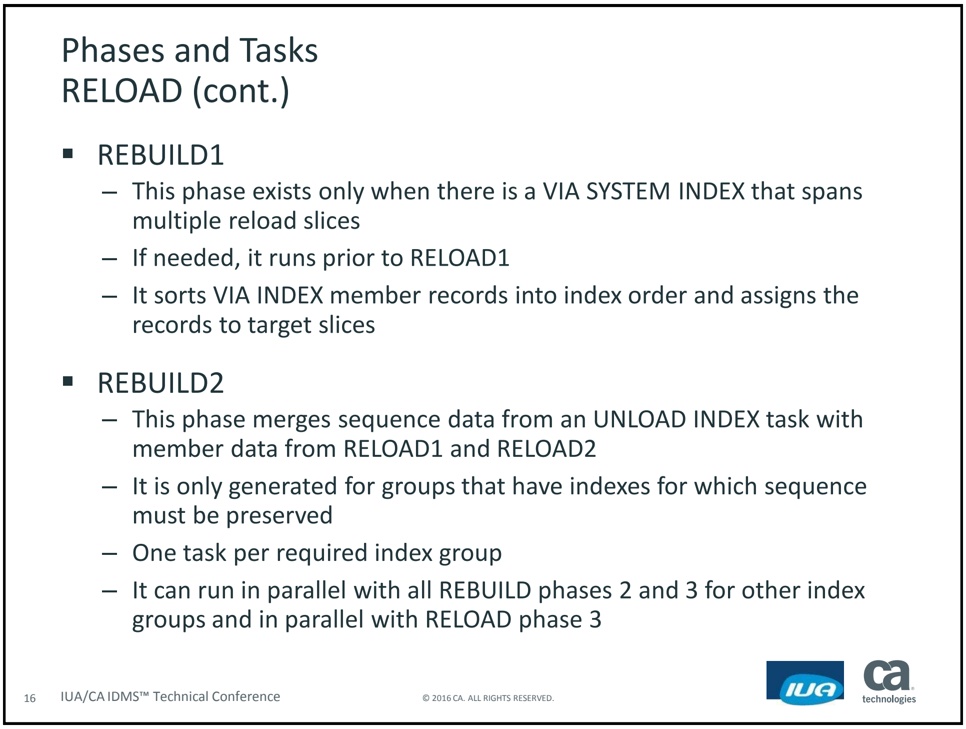
- RELOAD5
  - This is an overflow phase for RELOAD4. Data that would not fit into a slice is processed here
  - There is only one task in this phase, and it updates pages across all slices
  - If an area fills up during this phase, processing stops
  - If the task in this phase fails, when restarted,
    - REORG reprocesses all tasks in RELOAD4
    - User indexes that were already built are rebuilt

14 IUA/CA IDMS™ Technical Conference



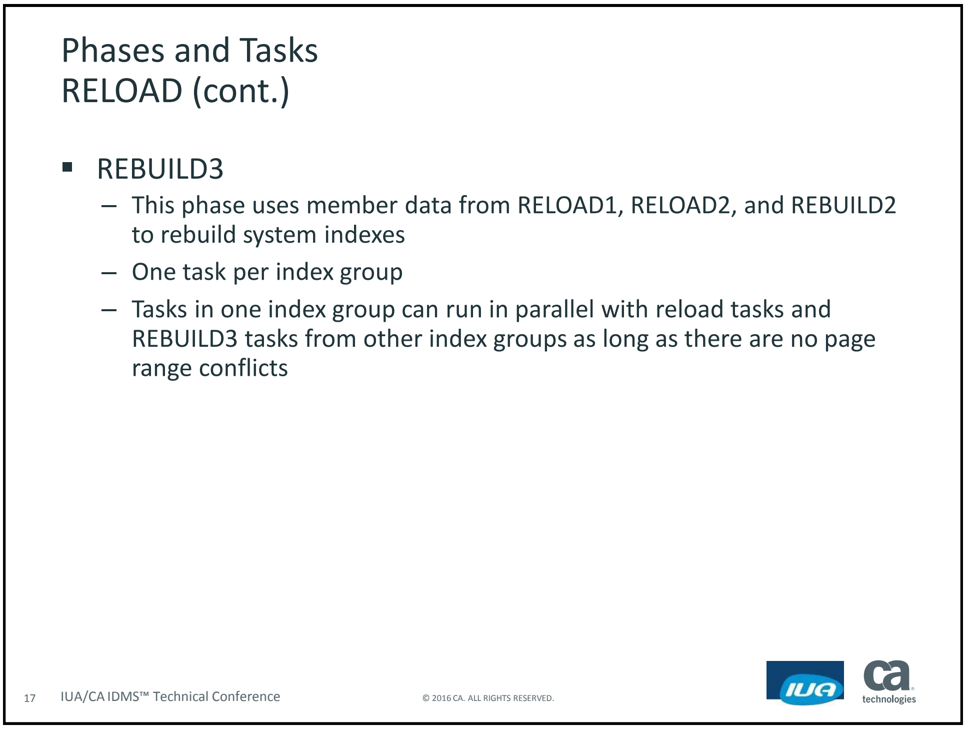
Phases and Tasks RELOAD (cont.)

- RELOAD6
  - Updates prefix pointers using data generated by RELOAD phases 4 and 5; and REBUILD phase 3
  - Potentially one task for each slice that can run concurrently
  - Writes ‘overflow’ records when connecting pointers to an SR2/SR3 combination that span into another slice
- RELOAD7
  - One task for the entire sub-phase which updates all slices
  - Using data written by RELOAD6 tasks to their overflow files, this phase updates prefix pointers for SR2/SR3 combinations where the SR2 record resides in a different slice than its associated SR3 record



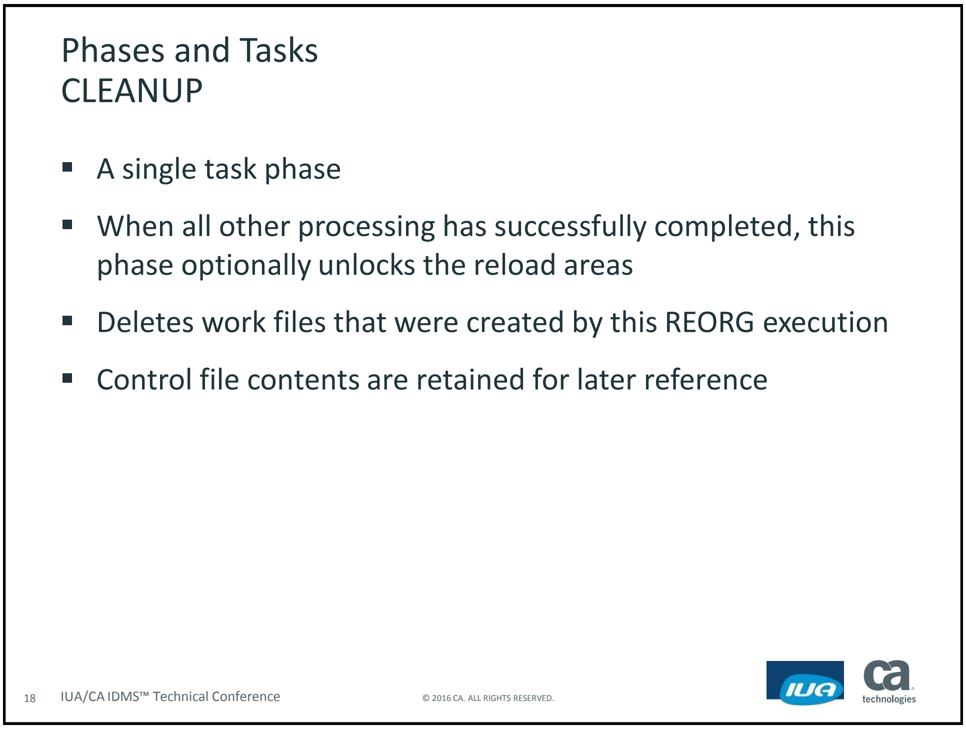
Phases and Tasks RELOAD (cont.)

- REBUILD1
  - This phase exists only when there is a VIA SYSTEM INDEX that spans multiple reload slices
  - If needed, it runs prior to RELOAD1
  - It sorts VIA INDEX member records into index order and assigns the records to target slices
- REBUILD2
  - This phase merges sequence data from an UNLOAD INDEX task with member data from RELOAD1 and RELOAD2
  - It is only generated for groups that have indexes for which sequence must be preserved
  - One task per required index group
  - It can run in parallel with all REBUILD phases 2 and 3 for other index groups and in parallel with RELOAD phase 3



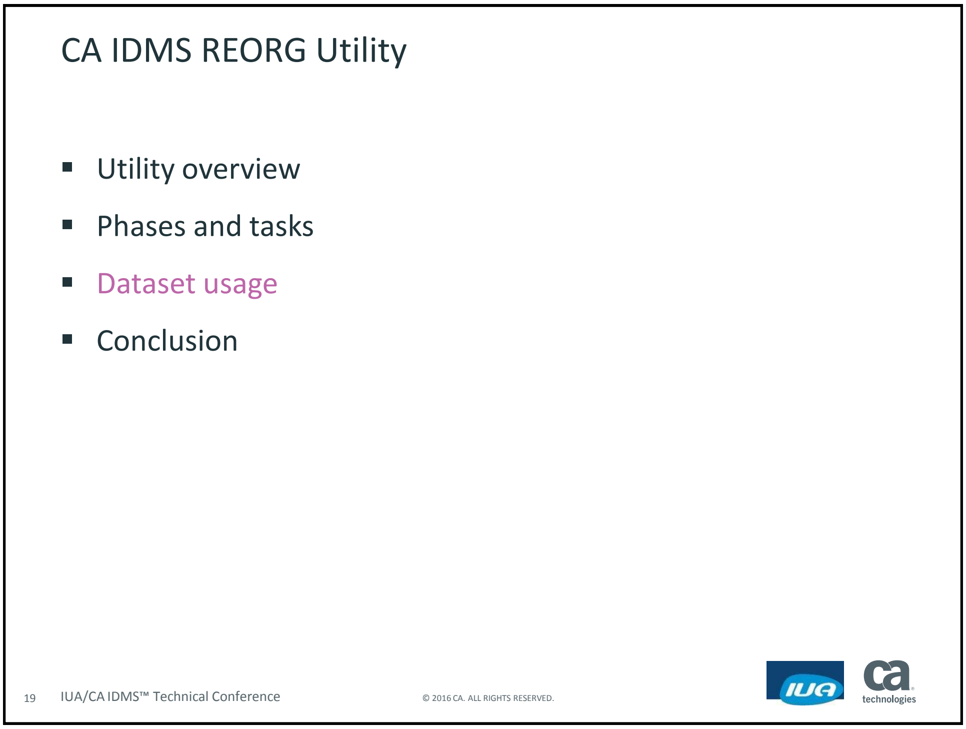
Phases and Tasks
RELOAD (cont.)

- REBUILD3
  - This phase uses member data from RELOAD1, RELOAD2, and REBUILD2 to rebuild system indexes
  - One task per index group
  - Tasks in one index group can run in parallel with reload tasks and REBUILD3 tasks from other index groups as long as there are no page range conflicts



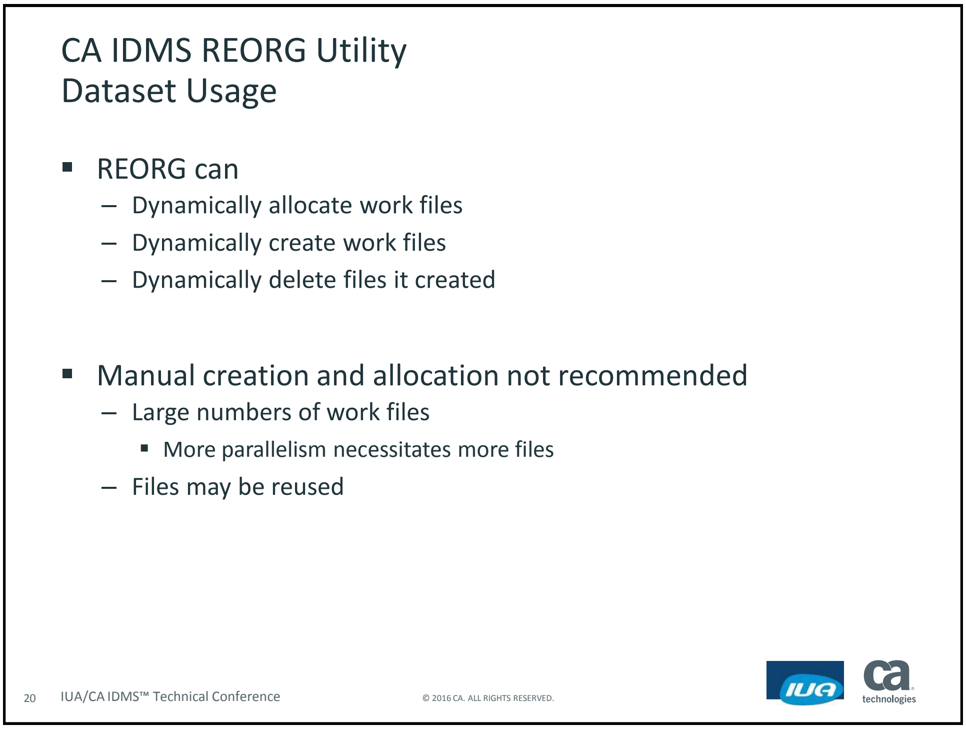
Phases and Tasks
CLEANUP

- A single task phase
- When all other processing has successfully completed, this phase optionally unlocks the reload areas
- Deletes work files that were created by this REORG execution
- Control file contents are retained for later reference



CA IDMS REORG Utility

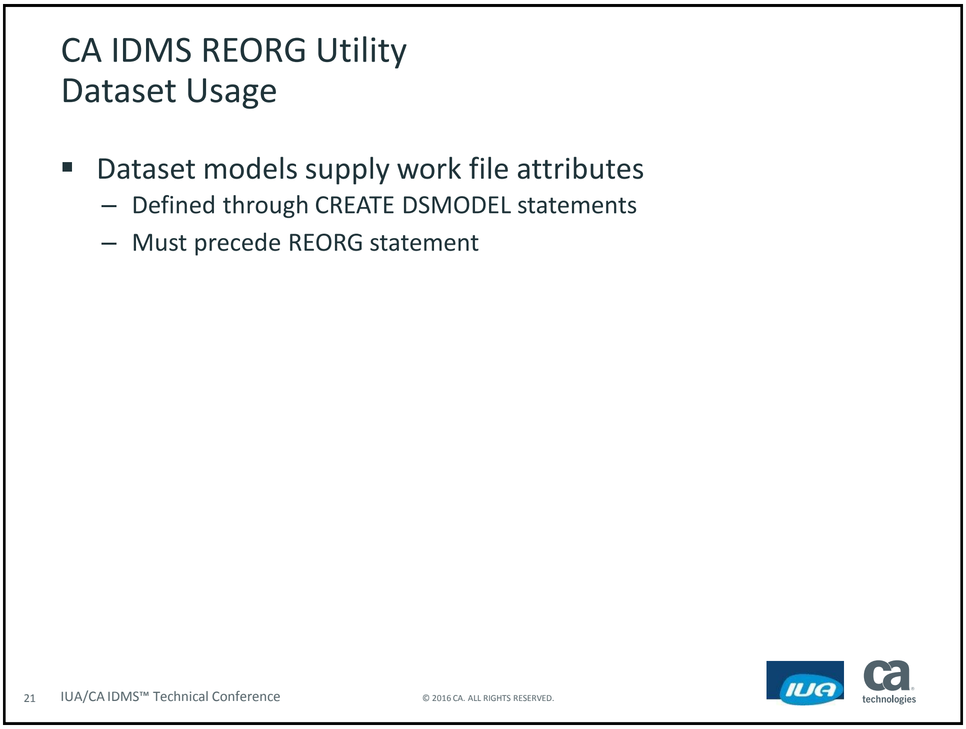
- Utility overview
- Phases and tasks
- Dataset usage
- Conclusion



CA IDMS REORG Utility
Dataset Usage

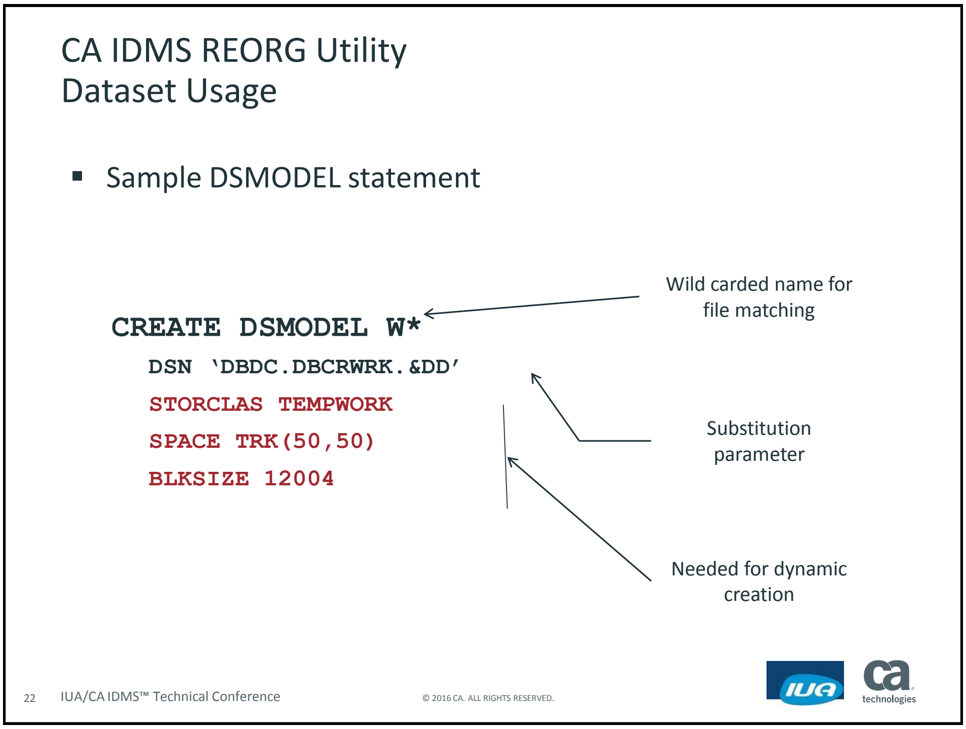
- REORG can
  - Dynamically allocate work files
  - Dynamically create work files
  - Dynamically delete files it created

- Manual creation and allocation not recommended
  - Large numbers of work files
    - More parallelism necessitates more files
  - Files may be reused



CA IDMS REORG Utility
Dataset Usage

- Dataset models supply work file attributes
  - Defined through CREATE DSMODEL statements
  - Must precede REORG statement



CA IDMS REORG Utility
Dataset Usage

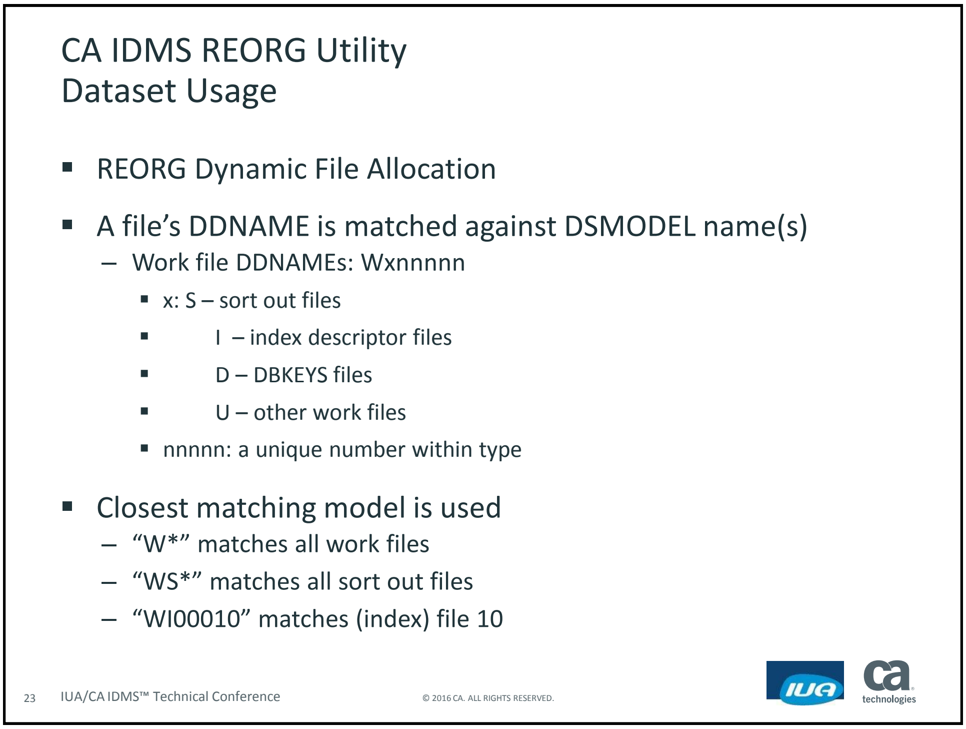
- Sample DSMODEL statement

```
CREATE DSMODEL W*
   DSN 'DBDC.DBCRWRK.&DD'
   STORCLAS TEMPWORK
   SPACE TRK(50,50)
   BLKSIZE 12004
```

Wild carded name for file matching

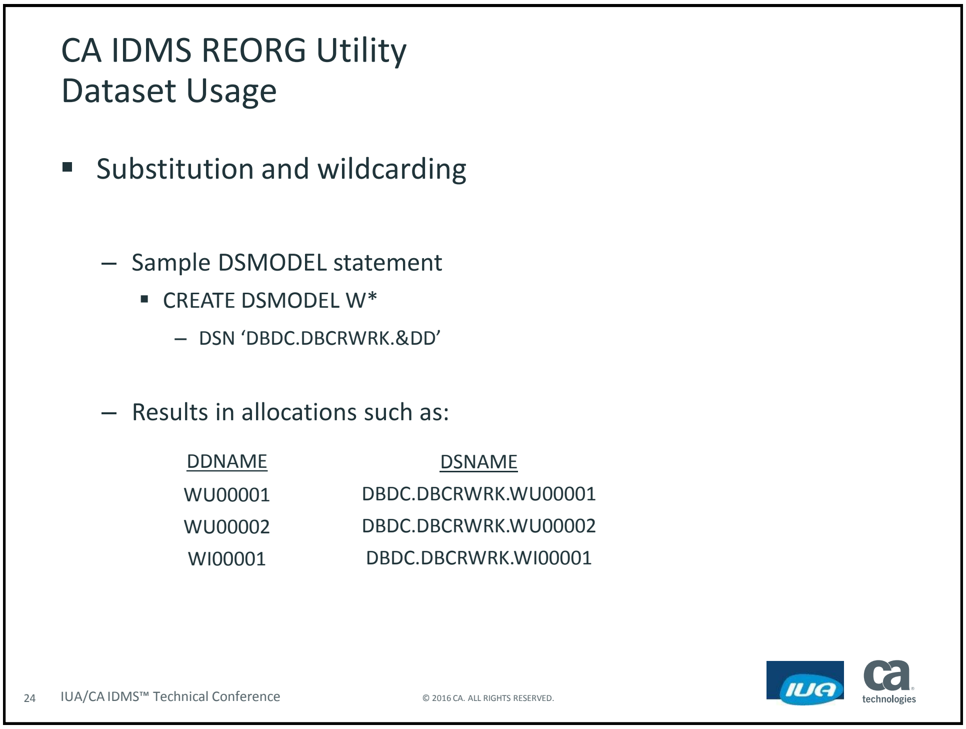
Substitution parameter

Needed for dynamic creation



CA IDMS REORG Utility
Dataset Usage

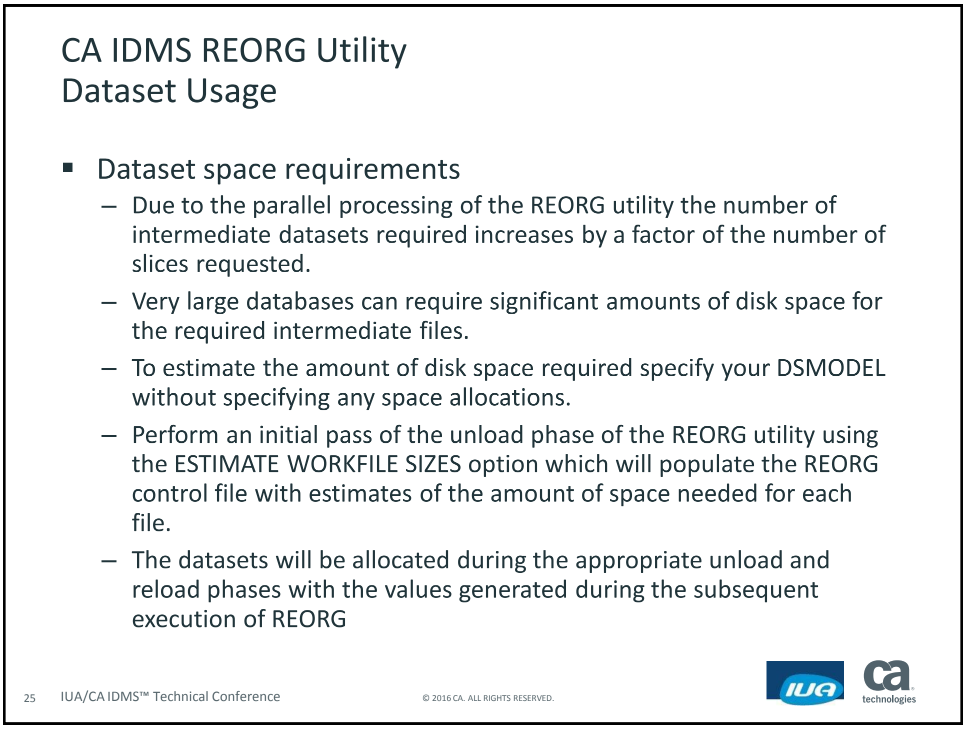
- REORG Dynamic File Allocation
- A file’s DDNAME is matched against DSMODEL name(s)
  - Work file DDNAMEs: Wxnnnnn
    - x: S – sort out files
    - I – index descriptor files
    - D – DBKEYS files
    - U – other work files
    - nnnnn: a unique number within type
- Closest matching model is used
  - “W*” matches all work files
  - “WS*” matches all sort out files
  - “WI00010” matches (index) file 10



CA IDMS REORG Utility
Dataset Usage

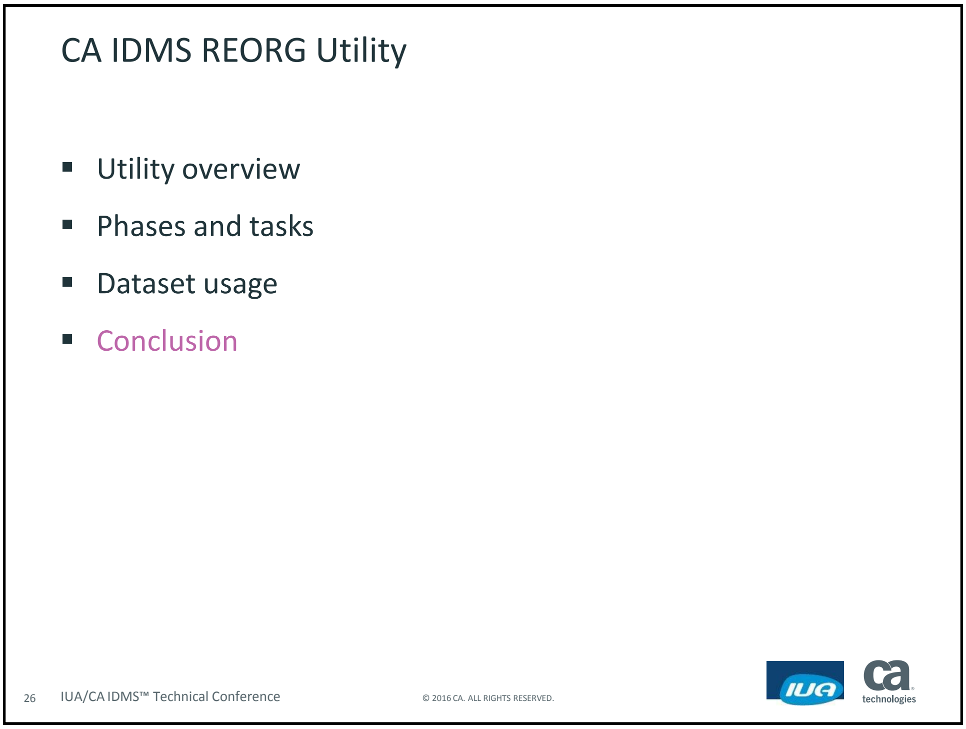
- Substitution and wildcarding
  - Sample DSMODEL statement
    - CREATE DSMODEL W*
      - DSN ‘DBDC.DBCRWRK.&DD’
  - Results in allocations such as:

DDNAME	DSNAME
WU00001	DBDC.DBCRWRK.WU00001
WU00002	DBDC.DBCRWRK.WU00002
WI00001	DBDC.DBCRWRK.WI00001



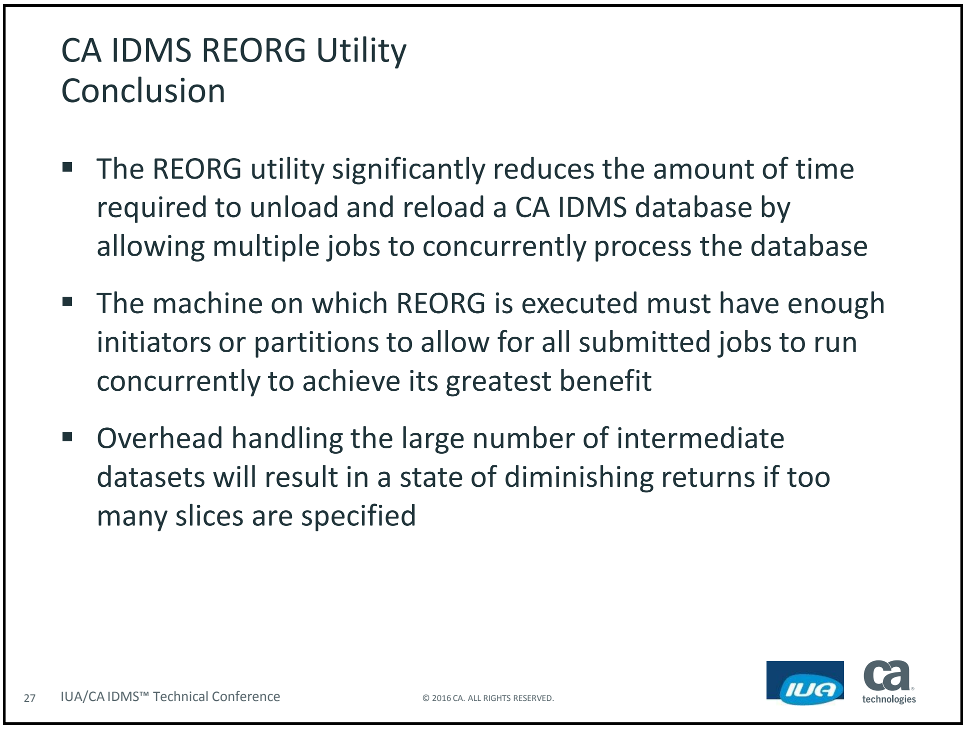
CA IDMS REORG Utility
Dataset Usage

- Dataset space requirements
  - Due to the parallel processing of the REORG utility the number of intermediate datasets required increases by a factor of the number of slices requested.
  - Very large databases can require significant amounts of disk space for the required intermediate files.
  - To estimate the amount of disk space required specify your DSMODEL without specifying any space allocations.
  - Perform an initial pass of the unload phase of the REORG utility using the ESTIMATE WORKFILE SIZES option which will populate the REORG control file with estimates of the amount of space needed for each file.
  - The datasets will be allocated during the appropriate unload and reload phases with the values generated during the subsequent execution of REORG



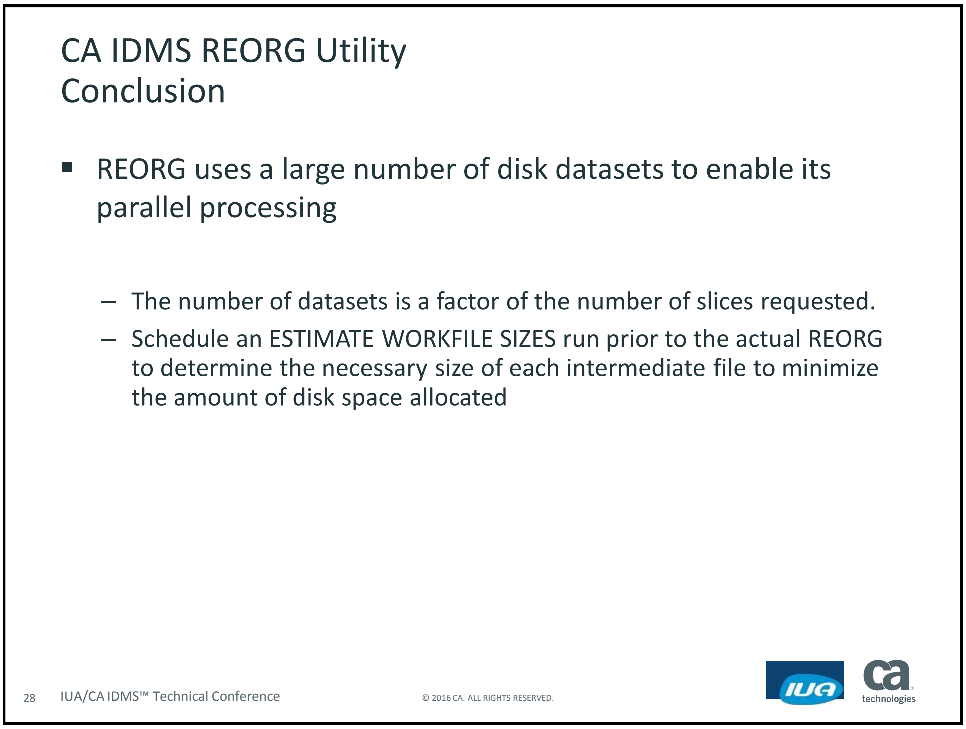
CA IDMS REORG Utility

- Utility overview
- Phases and tasks
- Dataset usage
- Conclusion



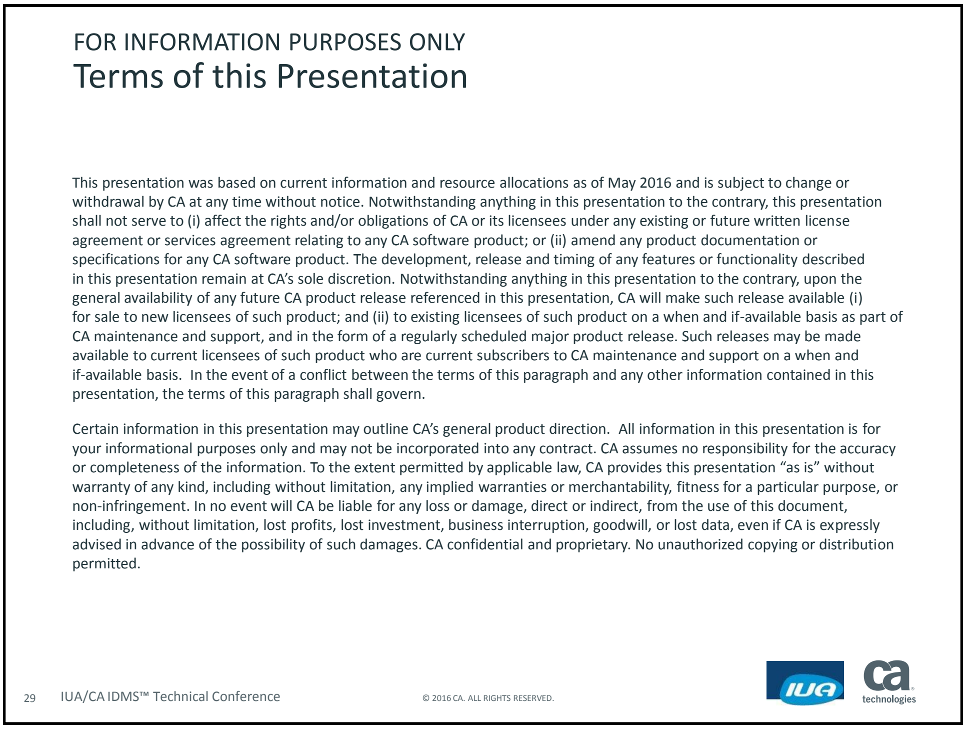
CA IDMS REORG Utility
Conclusion

- The REORG utility significantly reduces the amount of time required to unload and reload a CA IDMS database by allowing multiple jobs to concurrently process the database
- The machine on which REORG is executed must have enough initiators or partitions to allow for all submitted jobs to run concurrently to achieve its greatest benefit
- Overhead handling the large number of intermediate datasets will result in a state of diminishing returns if too many slices are specified



CA IDMS REORG Utility
Conclusion

- REORG uses a large number of disk datasets to enable its parallel processing
  - The number of datasets is a factor of the number of slices requested.
  - Schedule an ESTIMATE WORKFILE SIZES run prior to the actual REORG to determine the necessary size of each intermediate file to minimize the amount of disk space allocated



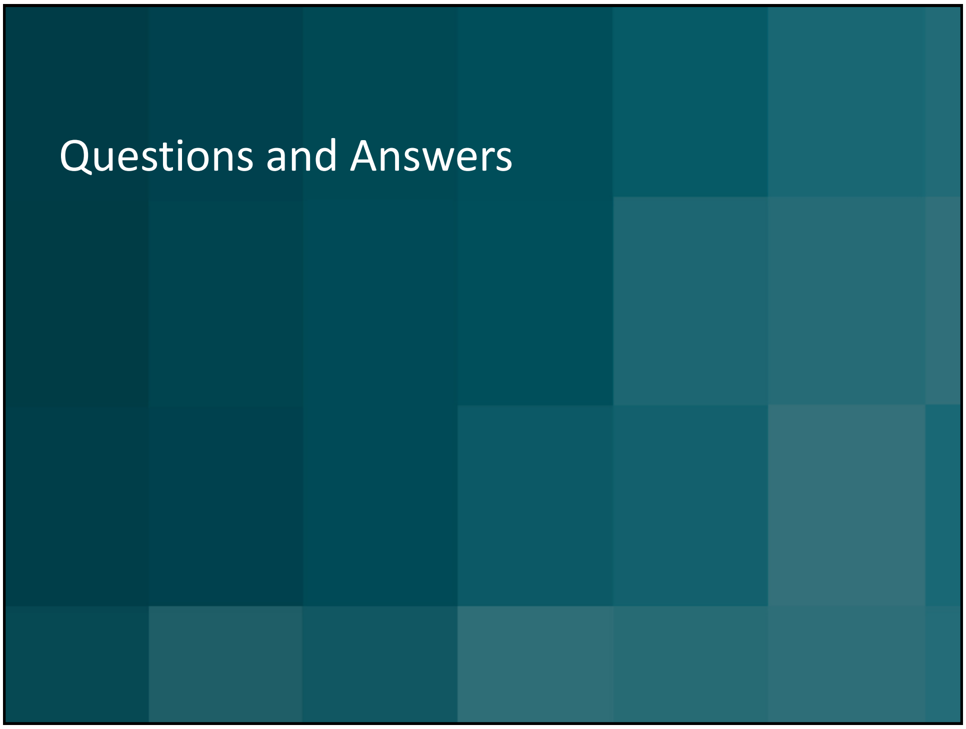
FOR INFORMATION PURPOSES ONLY

Terms of this Presentation

This presentation was based on current information and resource allocations as of May 2016 and is subject to change or withdrawal by CA at any time without notice. Notwithstanding anything in this presentation to the contrary, this presentation shall not serve to (i) affect the rights and/or obligations of CA or its licensees under any existing or future written license agreement or services agreement relating to any CA software product; or (ii) amend any product documentation or specifications for any CA software product. The development, release and timing of any features or functionality described in this presentation remain at CA's sole discretion. Notwithstanding anything in this presentation to the contrary, upon the general availability of any future CA product release referenced in this presentation, CA will make such release available (i) for sale to new licensees of such product; and (ii) to existing licensees of such product on a when and if-available basis as part of CA maintenance and support, and in the form of a regularly scheduled major product release. Such releases may be made available to current licensees of such product who are current subscribers to CA maintenance and support on a when and if-available basis. In the event of a conflict between the terms of this paragraph and any other information contained in this presentation, the terms of this paragraph shall govern.

Certain information in this presentation may outline CA's general product direction. All information in this presentation is for your informational purposes only and may not be incorporated into any contract. CA assumes no responsibility for the accuracy or completeness of the information. To the extent permitted by applicable law, CA provides this presentation "as is" without warranty of any kind, including without limitation, any implied warranties or merchantability, fitness for a particular purpose, or non-infringement. In no event will CA be liable for any loss or damage, direct or indirect, from the use of this document, including, without limitation, lost profits, lost investment, business interruption, goodwill, or lost data, even if CA is expressly advised in advance of the possibility of such damages. CA confidential and proprietary. No unauthorized copying or distribution permitted.

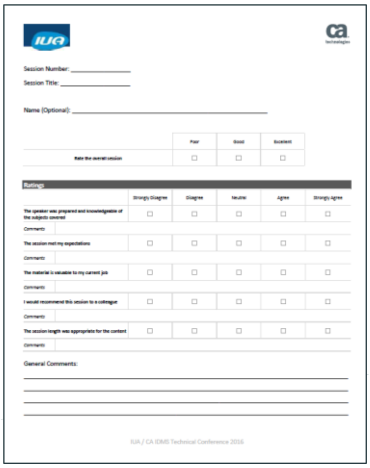
29 IUA/CA IDMS™ Technical Conference



Questions and Answers

Please Complete a Session Evaluation Form

- The number for this session is **D01**
- After completing your session evaluation form, place it in the envelope at the front of the room



Session Number: ___________

Session Title: ___________

Name (Optional): ______________________________

	Poor	Good	Excellent
Rate the overall session	□	□	□

Ratings

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
The speaker was prepared and knowledgeable of the subjects covered	□	□	□	□	□
Comments					
The session met my expectations	□	□	□	□	□
Comments					
The material is valuable to my current job	□	□	□	□	□
Comments					
I would recommend this session to a colleague	□	□	□	□	□
Comments					
The session length was appropriate for the content	□	□	□	□	□
Comments					

General Comments:

____

____

____

____

IUA / CA IDMS Technical Conference 2016