The 10 Golden Rules of ALE Optimization

Thomas Nittmann, Tim Yates
DataXstream

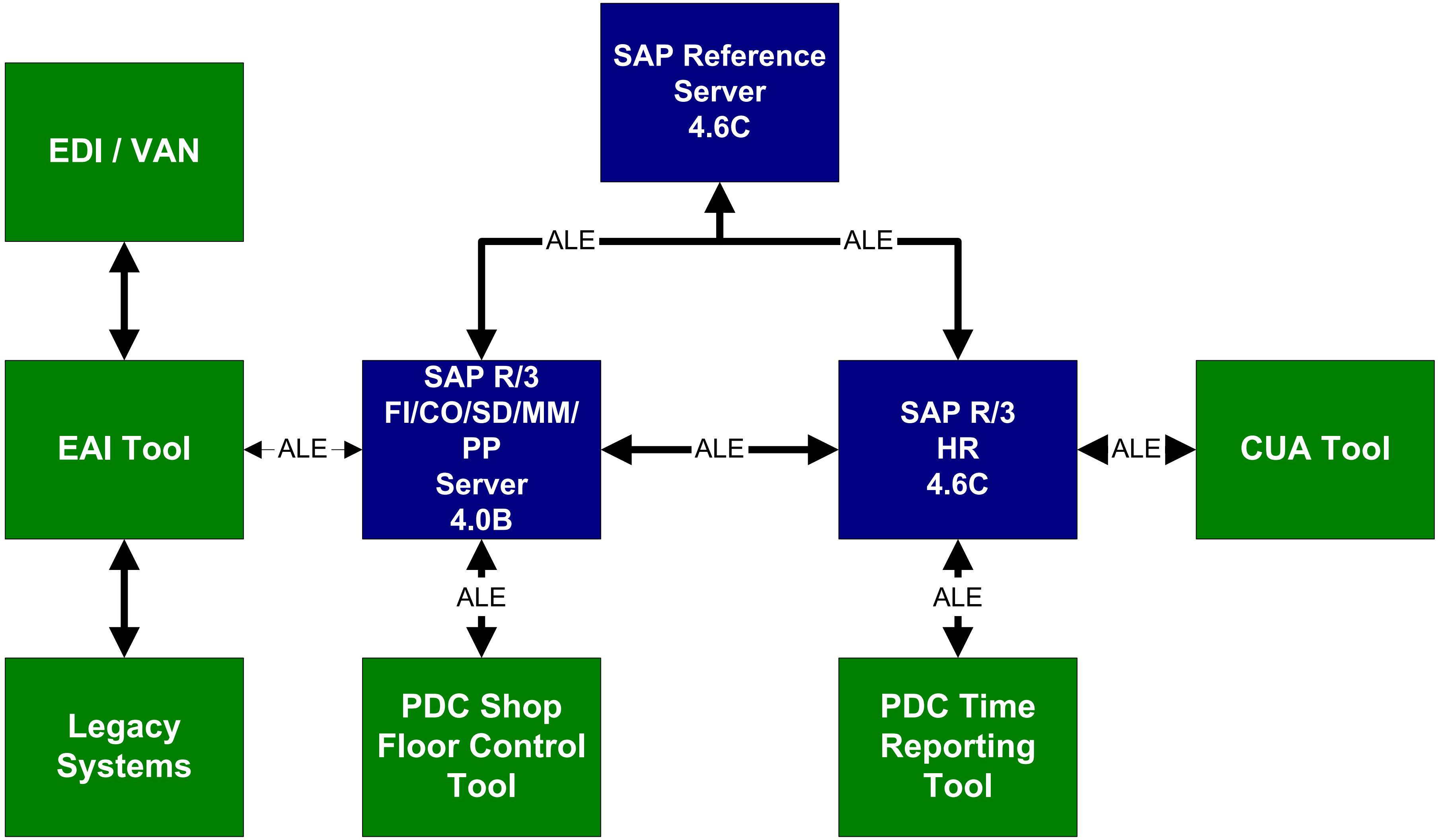
What We'll Cover

- Review of ALE integration scenarios
- Understanding ALE technology
- Optimizing ALE by understanding top 10 ALE performance hits
- Optimizing / sizing ALE interfaces during design phase

Review of ALE Integration Scenarios

- R/3 to R/3
  - Distributed R/3 systems
    - Master Data
    - Transactional Data
    - Configuration\Control Data
- R/3 to mySAP
  - APO, BW, CRM, EBP
- R/3 to 3rd Party Bolt-On Application
  - Warehouse Management Systems, PDC Integration: Production Order Confirmations, Time Reporting, Maintenance Order Confirmations
- R/3 to EAI Tool
  - Legacy Data Feeds, Electronic Data Interchange (EDI), Best of Breed Product Integration

ALE Integration Scenario



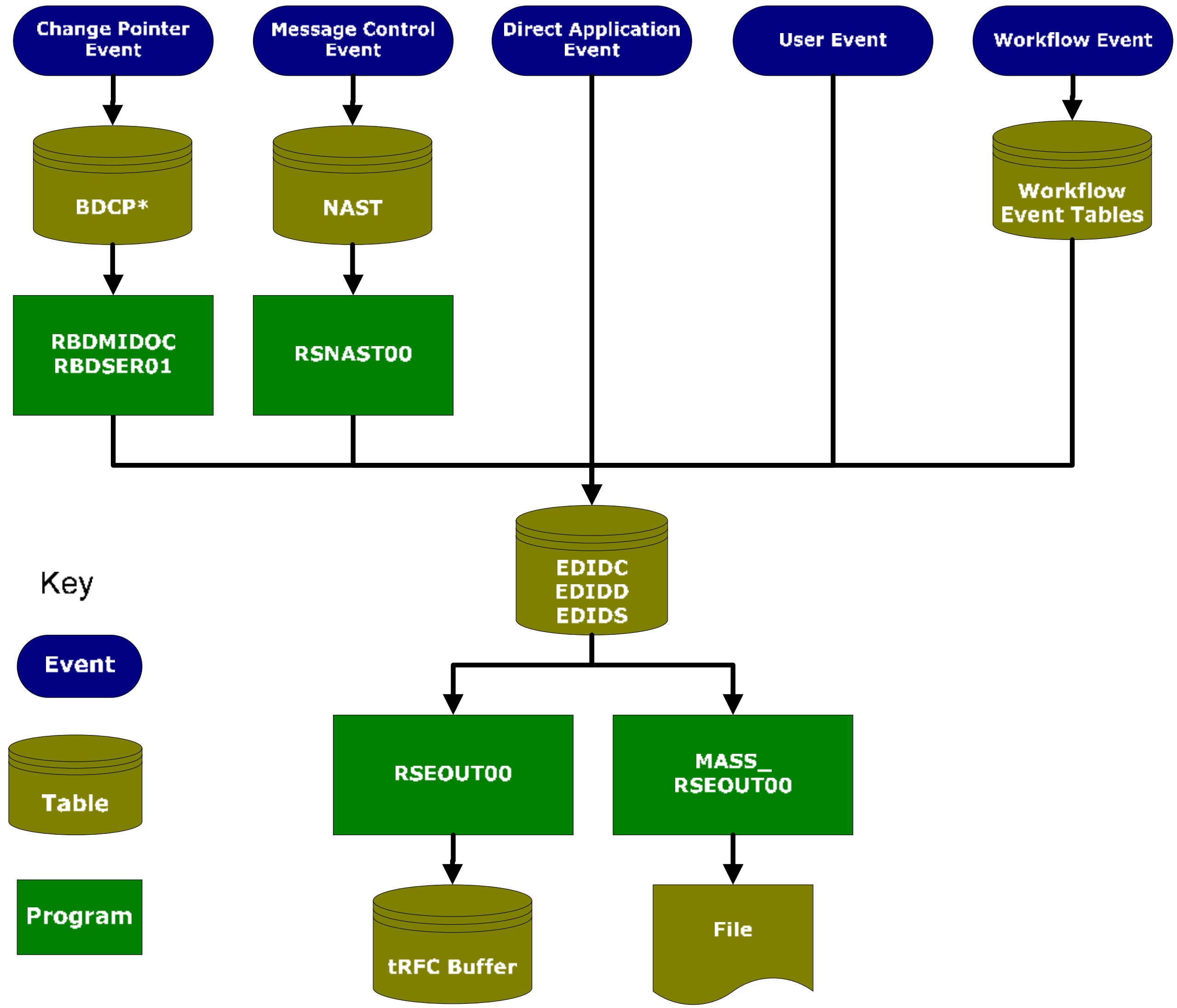
DATAXSTREAM Systems Integration & Architecture

SAP Professional Journal www.SAPpro.com

ALE Technology

- Application Layer
- Distribution Layer
  - Customer Distribution Model
  - Change Pointers
  - Message Control
  - IDocs
  - Ports
  - Partner Profiles
- Communication Layer
  - RFC Connections
  - tRFCs

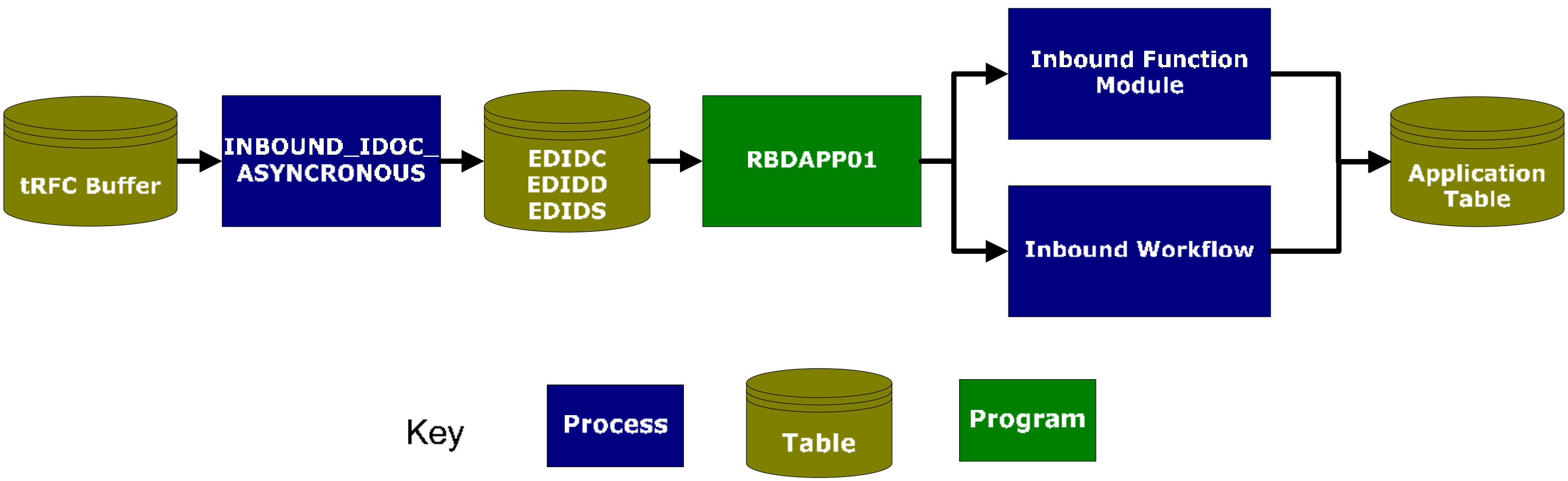
ALE Outbound Processing Technology



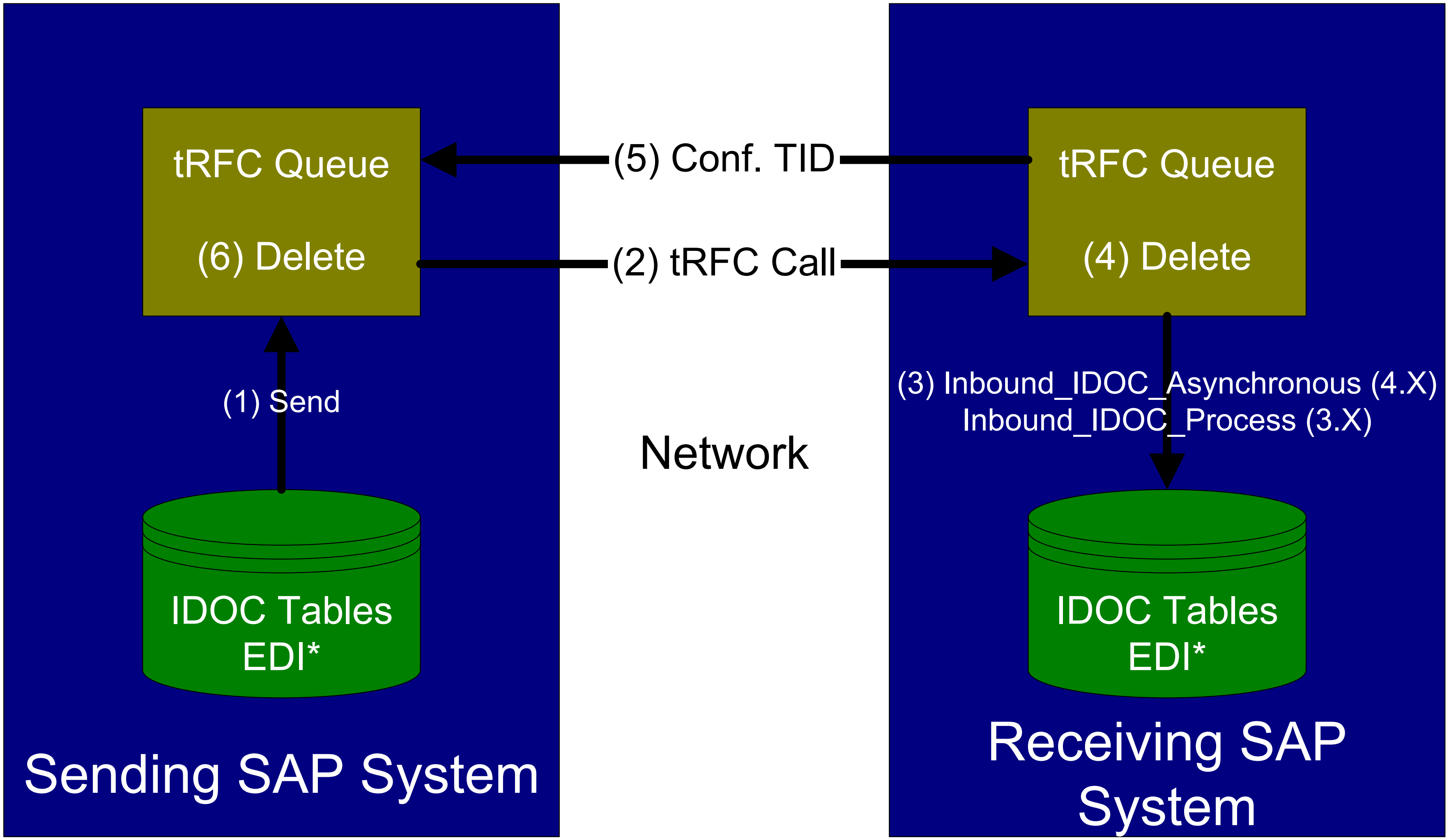
DATAXSTREAM Systems Integration & Architecture

SAP Professional Journal www.SAPpro.com

ALE Inbound Processing Technology



ALE Communication Layer Technology



DATAXSTREAM Systems Integration & Architecture

SAP Professional Journal www.SAPpro.com

ALE IDocs \ Message Types

- IDOC vs. Message Type
  - Message Type – Business Function
    - COGRP1 – Cost Center Hierarchy
    - COGRP6 – Profit Center Hierarchy
  - IDOC Type – Data Structure
    - COGRP01 – Hierarchy Data Structure
- IDOC Is Made Up of 3 Records
  - Control Record (1) – Structure EDIDC
  - Data Record (Many) – Segments
  - Status Record (Many) – Structure EDIDS

Top 10 ALE Performance Hits

1. Interference With Other Interfaces \ System Load
2. Archiving (IDOC, Workflow, Change Points, Message Control)
3. Inefficient User Exit Code
4. Workflow Error Handling
5. Automatic tRFC Batch Retry Switched ON
6. Using Immediate Processing
7. Using Single Processing Instead of Parallel Processing
8. Improper Sized Packets
9. Hardware and Basis Configuration
10. Standardization Without Interface Specific Considerations

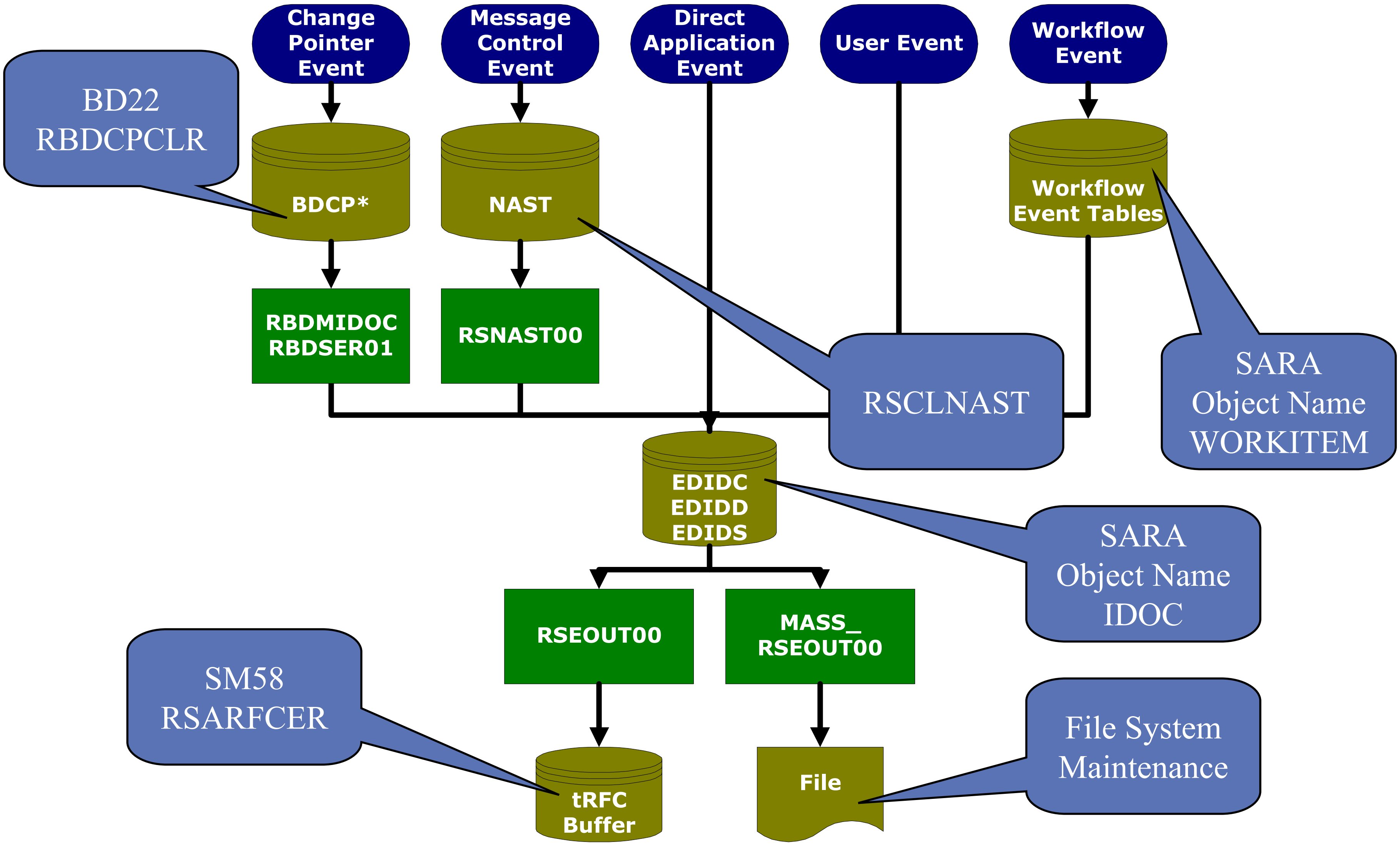
1) Interference and System Load - CS

- Because many different application areas leverage ALE technology it is important to understand their impact on each other.
- The impact of interfaces interacting and creating performance issues is typically not found until production problems arise.
- Coordinated scheduling of interfaces is very important. (will cover in scheduling).
- Other system load conditions can significantly impact ALE performance. Logon groups and dedicated ALE application servers can significantly reduce the effects.

2) Archiving - CS

- Many different applications leverage ALE technologies. This can significantly impact system performance as buffer tables and their indexes grow.
- Scheduling appropriate maintenance jobs is important in maintaining system performance in interfaces.
- While performance is not an issue in development and test environments, space usually is. Make sure your archiving plan is comprehensive.
- Also ensure that your archiving criteria are appropriate for the specific situation.

2) Archiving - CS (cont.)



What Does CS Mean?

Coordinated Scheduling

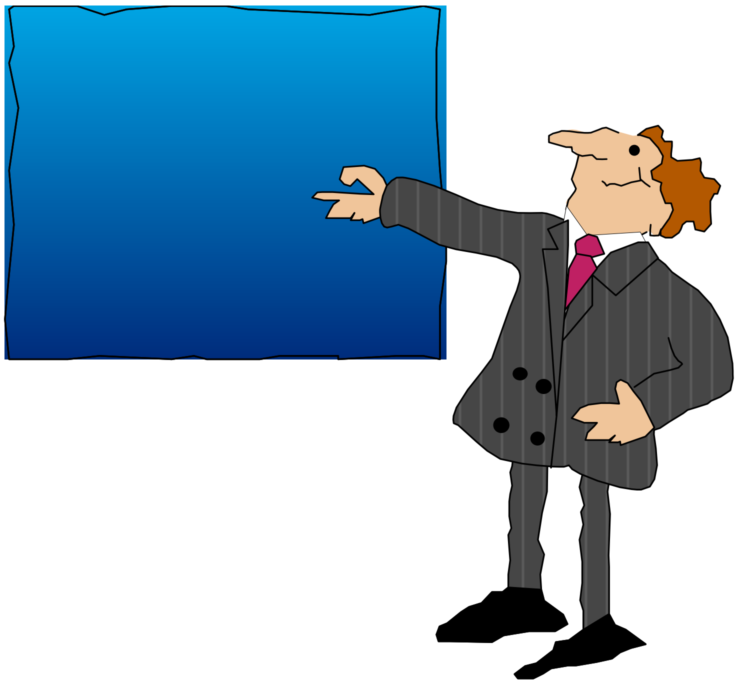
Common Sense

Commonly Skipped

3) User Exit Code

- Interface User Exits and Application Exits can significantly impact ALE performance.
- Inefficient User Exit code can decrease IDoc processing performance by more than 10-fold.
- Efficient ABAP Code is important when large volumes of IDocs are being processed.
- Also make sure that User Exit Code is being implemented in the most effective exit. It is possible that the same operation can be implemented more efficiently in another exit.

3) User Exit Demo



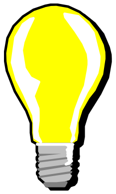
3) User Exit Demo (cont.)

- **Posting of Inbound ORDERS IDoc into R/3**
  - User Exit has been coded to recognize the IDoc Message Type: ORDERS_BAD and ORDERS_GOOD
  - ORDERS_BAD will run inefficient code
  - ORDERS_GOOD will run efficient code
  - Result: We will compare times of background posting jobs which post the two different sets of IDocs.

- **Demo Assumptions**
  - Same number of IDocs
  - Same IDoc Data
  - Same manipulation to IDoc by User Exit

4) Workflow Error Handling

- SAP ALE error handling is managed by SAP Workflow.
- Initial Workflow event linkages for all ALE Message Types are generated by running Transaction BD72.
- This linkage routes all errors to specified users inboxes
- Workflow processing creates overhead in the system
- If you are not using workflow to manage IDoc errors, Turn It Off.
- This will reduce system load when IDoc errors occur and will reduce disk usage by interfaces.



Tip

DATAXSTREAM Systems Integration & Architecture

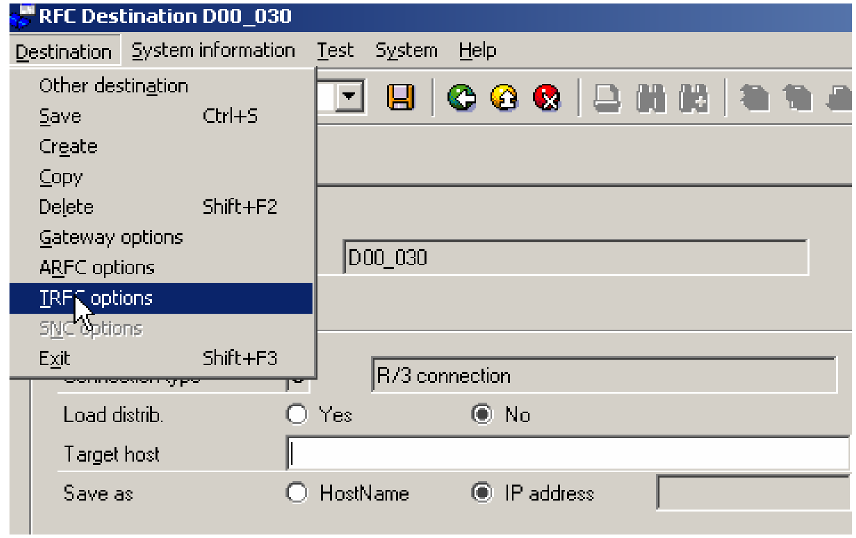
SAP Professional Journal www.SAPpro.com

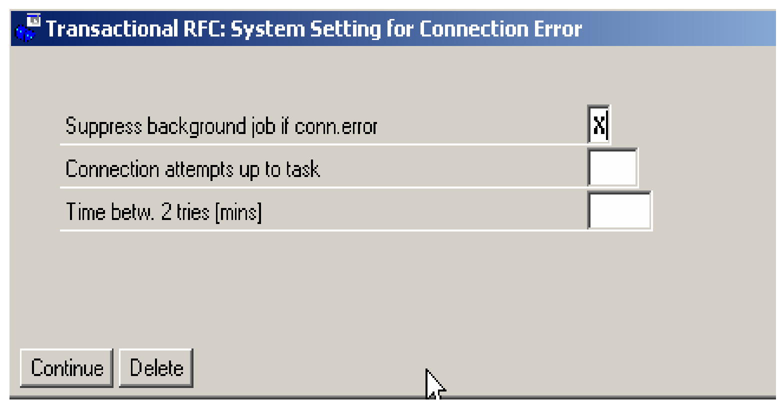
5) Automatic tRFC Batch Retry

- The tRFC Batch Retry Flag is defaulted ON when an RFC Destination is created.
- When a communication error occurs, the tRFC Batch schedules a Retry Job in the background, with a frequency of every 2 minutes.
- If you were sending 1000 IDocs with a packet size of 10, that would result in 100 background jobs executing every 2 minutes.
- This will result in ALE taking down the SAP batch job scheduler.

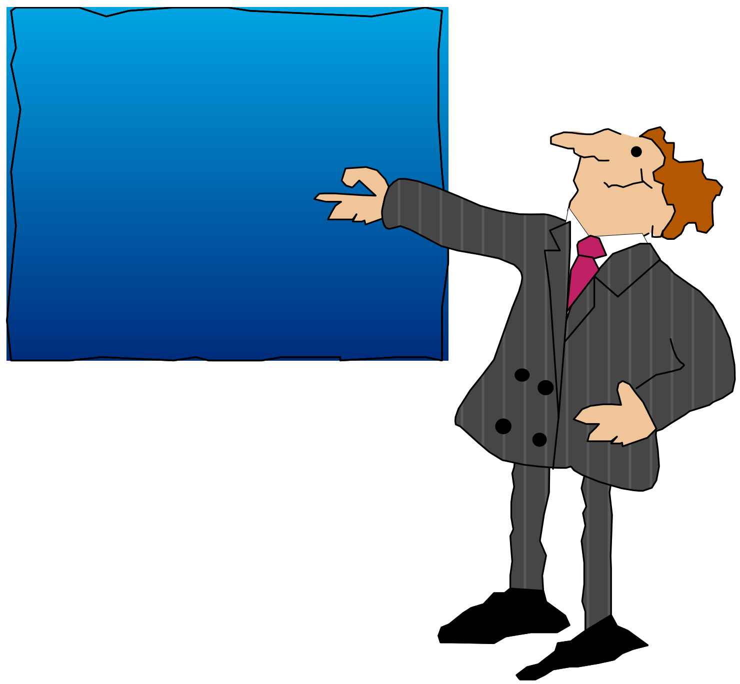
5) Automatic tRFC Batch Retry

- To turn off the Retry, you must edit the RFC Destination (SM59)
- Open the RFC Destination and select the following menu option
- Place an “X” in the Suppress Retry Box





5) Automatic tRFC Batch Retry Demo



5) Automatic tRFC Batch Retry Demo (cont.)

- R/3 communicating via ALE to a "Dummy System"
- When communication is triggered, the connection to the Dummy System will be set up so that the tRFC Retry is activated
- Result: aRFC Retry Jobs scheduled to retry until successful every 1 minutes
- Demonstration Assumptions
  - tRFC Retry Parameter Set To Default

6) Collecting IDocs vs. Immediate Processing

- When processing Outbound IDocs if the Dispatch Control is set to immediate, IDocs are packaged in packets of one IDoc per packet.
- When large data sends are initiated, this causes a separate tRFC Transaction to occur for every IDoc.
- This can overload the tRFC Buffer and eventually cause the Initiating Program to Roll Back.
- It also increases network load because of tRFC overhead.
- This can be corrected by setting the Dispatch Control on the Partner Profile to collect and by scheduling Program RSEOUT00 in the background.

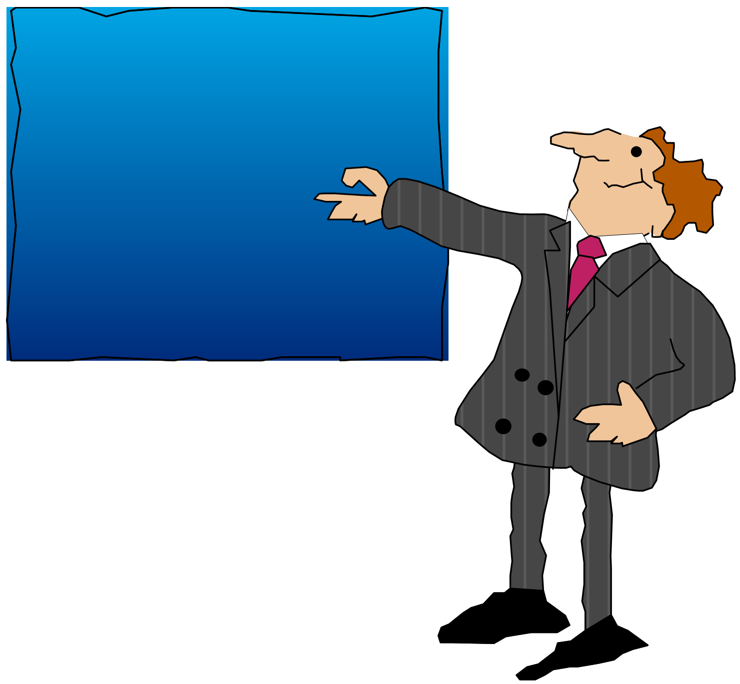
6) Collecting IDOCS Vs Immediate Processing (cont.)

- When processing inbound IDocs, if the Dispatch Control is set to Immediate, IDocs are processed serially.
- This will significantly decrease inbound processing throughput.
- This also causes the interface to create more system overhead by only processing one IDoc per update process.
- This can be corrected by setting the Dispatch Control on the Partner Profile to Collect.
- And scheduling the RBDAPP01 in the background.

7) Parallel Processing or Server Groups

- Background processing significantly improves interface performance. However, it does not allow you to distribute your interface load across multiple application servers.
- Parallel Processing \ Using Server Groups can significantly improve IDoc performance by distributing interface load across multiple SAP application servers for both inbound and outbound IDoc processing.
- IDocs are processed via multiple Dialog Work Processes vs. a single Dialog Work Process.
- It is important that RFC Server Groups are configured In (RZ12).

7) Parallel Processing Demo



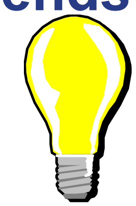
7) Parallel Processing Demo (cont.)

- Processing inbound ORDERS IDocs using Server Groups
- When processing inbound IDocs via RBDAPP01, there is an option to use Server Groups. If Server Groups are configured correctly, you can use this feature to start multiple processing threads to the application layer and distribute load across multiple servers.
- Demo Assumptions:
  - One server only, so will only demonstrate the Multithreading not Server Balancing.

8) Sizing IDoc Packets

- Improper packet sizing can reduce system performance at the communication layer, by reducing the effectiveness of the network, parallel processing, and can cause hardware memory management problems.
- Correct sizing of IDoc packets can be influenced by several variables.
  - Hardware
  - Network
  - IDoc Message Type

- SAP in its ALE Performance Tuning Workshop recommends 1000-2000 segments / packet



Tip

8) Sizing IDOC Packets (cont.)

- Packet size is controlled on the Partner Profile.
- The Value Input is the number of IDocs in a packet, not segments.
- Some analysis of the specific message type is required to set the packet size correctly.

8) Sizing IDOC Packets

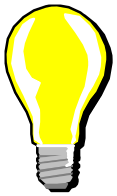
- **Example Sizing Exercise**
  - Message Type GLMAST
  - GLMAST01 IDoc Structure
    - E1SKA1M (1) – GL Account Master Data
      - E1SKATM (Many) – GL Account Master Data Text
      - E1SKB1M (Many) – GL Account Master Data Company Code
  - Sizing for initial loads or conversions via ALE interface
    - Number of Languages Chart (10)
    - Number of Company Codes Being Loaded (100)
    - Calculation
      - 1 E1SKA1M + 10 E1SKATM + 100 E1SKB1M = 111 Segments
    - Therefore a Packet Size of 10 to 20 is Appropriate
  - Sizing for daily volumes via ALE Interface
    - 2 Changes to Language Data With Interface cycle
    - 2 Changes to Company Code Data With Interface Cycle
    - Calculation
      - 1 E1SKA1M + 2 E1SKATM + 2 E1SKB1M = 5 Segments
    - Therefore a Packet Size of 200 to 400 is Appropriate

DATAXSTREAM Systems Integration & Architecture

SAP Professional Journal www.SAPpro.com

9) Hardware and Basis Configuration

- Inaccurate / improper sizing of your system for its interfaces can cause significant issues.
- This can lead to many poor architectural decisions.
- Proper planning is the key to eliminating these issues.
- When sizing a system, SAP lumps interfaces in the system overhead and sizes a system to 65% capacity.
- Use the 80/20 Rule. 80% of your interface will be sized appropriately. Look for the other 20% that are not.



Tip

9) Hardware and Basis Configuration (cont.)

- Often the network between integrated systems is overlooked.
- If you are exchanging large volumes of data, make sure the physical network between integrated systems is streamlined. Look for the bottlenecks before there are real issues.
- Minimize the number of physical hops between integrated systems.
- When encryption is required, make sure adequate encryption hardware is employed on the network.

9) Hardware and Basis Configuration (cont.)

- Table Space issues and Basis configuration are often problems in development and test environments and can also cause issues during initial data conversions.
- Check your table sizing for ALE Buffer Tables and Workflow Tables and make sure it is appropriate for all volumes
  - Development Testing, Conversion Testing, Performance Testing, Production Conversions, Production Interfaces
- Include changes to Basis configuration in conversion plans to facilitate better performance during these high load activities
  - Customize Server Groups, Customize Work Process Available, Adjust System Parameters, Buffer Tables

9) Hardware and Basis Configuration (cont.)

ztta/roll_first
ztta/roll_extension
abap/heap_area_dia
abap/heap_area_total
abap/heap_area_nondia
em/initial_size_MB
enqueue/table_size
rdisp/max_comm_entries
gw/max_conn

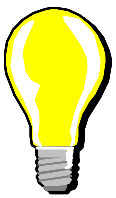
gw/max_overflow_size
rdisp/elem_per_queue
rdisp/appc_ca_blk_no
rdisp/rfc_use_quotas
rdisp/rfc_max_queue
rdisp/rfc_max_login
rdisp/rfc_max_own_login
rdisp/rfc_max_comm_entries
rdisp/rfc_max_own_used_wp

10) Standardization Without Interface Specific Considerations

- Often if ALE is leveraged in many areas on large SAP implementations, it is good to standardize on one way of doing things. This helps to reduce implementation issues and confusion.
- This, however, can often lead to problems with interfaces because of a rule or standard being applied when it is not appropriate.

10) Standardization Without Interface Specific Consideration

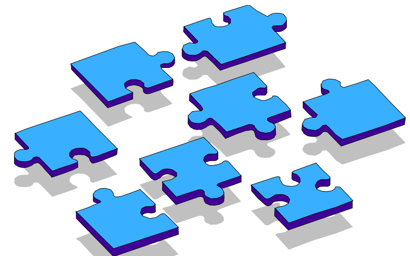
- **Use the 80-20 Rule. You can expect that 80% of your interfaces can follow a standard, but let the other 20% deviate when it is appropriate.**
- **Examples of standards that might impact overall performance**
  - **Always use A packet size of 20**
  - **Always activate Event Linkage for Workflow Error Handling**
  - **All interfaces will run at a specific frequency**
  - **Always use Change Pointers for tracking Master Data changes**
  - **Always use ALE / IDOC interfaces**
  - **Always use middleware**



Tip

Important Design Considerations

- Interface Big Picture
- Volumetrics
- Available Options
  - Input Method
- Hardware \ Disk Sizing
- Scheduling
- Test Planning
- Get in the Ball Park, No Satellites, No Microscopes



Building Block

Warning

ALE Big Picture (Design)

- **Understand total interface scope**
  - Many different areas of SAP leverage ALE technology and therefore impact performance.
- **Most developers are working on specific pieces**
  - Total scope of all interfaces is not always clear.
- **Understand the full scope**
  - R/3 to R/3 Integration
  - R/3 to 3rd Bolt On
  - EDI
  - R/3 to EAI Tools

ALE Volumetric Considerations (Design)

- Initial load volumes
- Frequency of records
- Expected peaks / distribution (hourly, daily, seasonal)

- This information drives hardware and disk sizing requirements, finding the other 20%.
- It should also drive your Interface design approach.
- Volume metrics example worksheet
- IDoc sizing example worksheet

ALE Available Options (Design)

BAPI

Calling a BAPI in the appropriate application transfers the data to the R/3 System. If BAPIs are used as interfaces to the SAP system, the same technique is used as for the continuous data transfer between R/3 systems or between non-SAP systems and R/3 Systems via ALE /IDocs.

Batch input

Batch input is a standard technique used to transfer large volumes of data into the R/3 System. In the process, the transaction flow is simulated, and the data is transferred as if it were entered online. The advantage of this procedure is that all the transaction checks are performed, which guarantees data consistency

Direct input

During direct input, the data in the data transfer file is first subjected to various checks, and is then imported directly into the R/3 System. The R/3 database is updated directly with the transferred data

ALE Job Scheduling (Design)

- Assumption: External job scheduling package available
- Least technical solution to optimizing ALE interfaces - CS
- Typically given little consideration
- External scheduling packages provide big returns when developing interfaces
- Benefits
  - Distribute loads via scheduling
  - Define rules to help avoid job pileups.
    - Did last job complete successfully?
    - Is last job still running?
    - Has job exceeded and acceptable time limit?
    - Is there another job running that will be negatively impacted?
  - Interface Early Warning System. If problems do occur schedule can notify appropriate people of an issue.

ALE Job Sequencing (Design)

- **Example of Job Sequencing for COSMAS Master Data Interface.**
  - Source System of COSMAS
    - RSARFCEX – Retry tRFC Buffer for errors
    - RBDMIDOC – Process Change Pointers for COSMAS
    - RSEOUT00 – Process Outbound COSMAS IDocs
    - RBDMOIND – Check tRFC Buffer Updated Status from 3 to 12
  - Target System of COSMAS
    - RBDAPP01 – Process Inbound COSMAS IDocs
    - RBDSTATE – Process ALEAUD IDocs
    - RSEOUT00- Process Outbound ALEAUD IDocs
  - Source System of COSMAS
    - RBDAPP01 – Process Inbound ALEAUD IDoc

  - Job Network Rules: Run Every 10 Minutes, Do Not Start Unless Last Network Completed Successfully, Notify If Any Job Runs More Than 15 Minutes, Notify If Any Job Returns An Error Code, Do Not Start If Networks X, Y and Z are Currently Running.

Near Real Time ALE Using Job Networks

- It is possible to extend the Job Networks to perform near real time Interfaces by starting Networks using System Events.
- Often there are Interface Processes that have to be updated in two different systems with very little delay.
- This technique allows you to use standard asynchronous interfaces to achieve near real time processes.

ALE Hardware\Sizing (Design)

- Hardware architecture can significantly impact performance.
- Table space sizing is also critical. Proper sizing is driven by Ball Park Volumetrics.
- A distributed architecture is expensive but segmenting applications on separate SAP instances and integrating them via ALE can also help in managing total system performance.

ALE Test Planning (Design)

- Interface performance testing can be very difficult. It is often hard to emulate all production conditions.
  - Hardware
  - User Load
- Put first things first. Do not try to put together a test plan that tests performance on low volume interfaces.
- If adequate performance testing is not possible, it is important to have significantly more monitoring of interfaces until an acceptable baseline is established.

Resources Available

- OSS Notes
- SAP Interface Advisor
- SAP Documentation

7 Key Points to Take Home

1. Understand Your Interfacing Big Picture
2. Archive Interface Objects
3. Plan Interface Job Networks and Use a Job Scheduler to Help Manage Load and Problems
4. Do Not Use Immediate Processing
5. Suppress tRFC Retry
6. Use Server Groups For Parallel Processing
7. Design It Smart from the Start

Your Turn!

Questions?