Managing Development, Test and Production Environments

Mark Rousculp  |  Director of Software Development
A conversation about infrastructure

- Phased Deployment
- Environments
- Moving things around
- Migration Advantages
- Integrating with Bridge
The Takeaway

- Why is a phased deployment strategy useful to me?
- What is the best way to set up my environments?
- How can DI tools help me to manage my phased deployment strategy?
Phased Deployment - What is it?

- Term used in software and IT

- Deployment split into stages
  - New content *Development*
  - Project *Testing* for validation
  - Accepted project enters *Production*

- Development, Test, Production
  - Also called DTP, DTAP, DTAPB, DTEP...
Advantages for Diver Content Developers

- Projects are tested before meeting users
- New content can be formally Accepted
- Manage migrations and upgrades
- Your IT team is doing this already
  - OS patches and upgrades
  - Server hardware
  - New Diver versions
Phased Deployment Workflow - Loops

- Development loop reaches milestone
  - Project by Project
  - Managed transition points
- Move to Test loop for validation
  - ETL process
  - Markers and Pages
  - Performance
- Deploy accepted content to Production
Development Loop

- **What it’s for**
  - New pages, dashboards & reports
  - Maintenance of existing projects

- **Who is using it**
  - Content Developers:
    - ProDiver users
    - Workbench users

- **Hardware**
  - Support dev user population
  - Run ETL process
  - Storage for entire data set
Test Loop

- **What it’s for**
  - Validating data, reports, ETL
  - Checking performance

- **Who is using it**
  - Testers, Developers
    - *May* be the same people

- **Hardware**
  - Support test user population
    - Load testing
  - Run ETL process
  - Storage for entire data set
Production Loop

- **What it’s for**
  - Serving Accepted data

- **Who is using it**
  - End Users:
    - Portal & Dashboard users
    - Non-developer Analysts

- **Hardware**
  - Support full user population under load
  - Fast Storage for Models and cBases
  - Storage for ETL
Loops in Summary

- Present unique challenges
- Have different requirements
- Should be isolated from each other

How?
What is an Environment

- An Environment is a Machine
- May be a physical machine or a VM

Why use Environments rather than DiveLines?
- Data sequestration between Environments
- Separate Diver from the Operating System
- Flexibility and freedom to tinker
To Virtualize or not to Virtualize

- Virtual Machine advantages
  - Flexibility
  - Backups
  - Price

- Virtual Machine disadvantages
  - VM server hardware limitations
  - Resource conflicts (RAM, I/O)
Dedicated Hardware Advantages

- Avoid resource conflicts
  - Network bandwidth
  - I/O
  - Memory

- Size to need

- Optimize for Diver requirements
DI Recommends

✅ Dedicated VM server
  – Host environments in role-based VMs
  – Efficient use of resources

✅ Role-based physical machines
  – A good alternative in lieu of VM expertise
  – Can be more cost effective in some cases

✅ In-chassis storage
  – Getting a SAN right is hard
  – Drives are bigger than ever
Three Loops, Three Environments

Development Environment:
• Support a few power users
• Data access speed less critical

Test Environment:
• Match Production hardware
• Mounts validation tools

Production Environment:
• Optimized for performance
• Sized to load
Hardware Life Cycle

- **How old is too old?**
  - DI replaces hardware every 3-4 years

- **Processors**
  - Processor generations are 1-2 years
  - Cheaper, faster, more efficient, more cores

- **Storage & Memory**
  - Solid State Drives
  - More chassis slots
  - 16 and 32 GB memory modules
Sizing Environments

- **Processors**
  - 3 GHz+ Haswell or Ivy Bridge EP

- **Memory**
  - 8+ GB per core *minimum*
  - Sized up to your cBases and users

- **Storage**
  - What do we mean by fast storage?
    - SSDs
    - 15k rpm SAS drives
    - Models and cBases in-chassis
Connecting Environments

- Phased Deployment
- Moving things around
- Migration Advantages
- Integrating with Bridge
Storage, Scripts, Data - Options

- Copy Everything
  - Easier to set up and maintain
  - Often good enough

- Just the Scripts
  - You plan to validate ETL anyway
  - Hardware or network considerations
Copy Everything

- Copy Projects, Models and cBases between environments
  - SSH, SFTP, or protocol of your choice

- Everything is automatically in synch

- ETL *need* not be run
Just the Scripts

- Copy scripts and configurations, then re-run ETL
  - Smaller, faster copy
  - ETL is hardened against old data

- ETL is validated as a part of process

- DI does not mandate any particular method
Phased Deployment

Environments

Migration

Advantages

Integrating with Bridge
7.0 Migration Over Time

Finance

Spectre KPI

Bridge

6.4

Finance

KPI

Direct
Phased Deployment and Migration

- Every representation is a manageable risk.
- With side-by-side installations, the advantages multiply.
- Each trunk contains multiple projects that could collide.
- Phased deployment minimizes those risks.
- Goal: End users see only improvements.
Phased Deployment

Environment

Integrating with Bridge

Migration Advantages
Navigating through the Bridge

- Users can be permitted to see only certain content
- Prohibited content doesn’t appear
- Many environments are easily managed on one page
Managing your Dev, Test and Prod environments

Any questions?
find us during & after the show:

- **Blog** -
  www.dimins.com/blog/

- **LinkedIn** -
  www.linkedin.com/company/dimensional-insight/

- **Twitter** -
  #DIUC15
  @DI_tweet

- **Facebook** -
  www.facebook.com/dimensdiver/