

# GENERAL DYNAMICS

## Land Systems



## DRM : Going Up the Learning Curve on 2nd and 3rd Implementations - From a Finance Users Perspective.

Tim Treloar – GDLS Finance

Nov 30, 2011

# DRM implementation at GDLS

## Agenda



- Company Overview
- DRM Overview
- Implementation History
- Manual vs. Automated Hierarchies Exports
- Data Input
- Department Master - Interfaces
- Validations vs Verifications
- Security/Access with additional Dimension
- Other issues to consider
- Finance users doing more with the Automator

# DRM implementation at GDLS

## Company Overview

- Land Systems provides a full spectrum of land and amphibious combat systems, subsystems and components worldwide, specifically to the domestic and foreign militaries
- Land Systems, like most GD Businesses was built through acquisition, specifically:
  - Chrysler Defense (1982)
  - General Motors (GM) Defense (2003)
- Land Systems is made up a few key business units:
  - GDLS - US
  - GDLS - Canada
  - Robotics
- Land Systems has approximately 7,500 employees worldwide in Australia, Canada, Egypt, Kuwait, Morocco, Saudi Arabia and the United States.



© Image Copyright: **Land Systems**  
(<http://www.gdls.com>)

# DRM implementation at GDLS

## MDM/DRM

- Master Data: Consistent, official set of identifiers, attributes and hierarchies for core entities.
- Master Data Management (MDM): A discipline used by business and IT to ensure uniformity, stewardship and accountability of shared information assets.
- Data Relationship Management (DRM): A combination of applications and technologies that consolidates, leans, and augments this corporate master data, and synchronizes it with all applications, business processes, and analytical tools.



# DRM implementation at GDLS

## DRM Overview

---

- A tool to create/maintain your dimensions/hierarchies (Ragged and Balanced hierarchies)
- Useful when you need to feed multiple systems (not just Hyperion)
- Data integrity Controls
  - Verification (Batch testing)
  - Validations (Real time testing)
- Functionality to help input of properties values
  - Pick lists
  - Lookup tables
  - Formulas to create properties
  - Automate for mass changes
- Export Tools
  - Exports easy to create (Parent/child)
  - Can export from transaction logs to assist with audits
- Versioning

# DRM implementation at GDLS

## Implementation History (IT outsourced to CSC)

---

- Contracts implemented in June 2010
  - Requirements Gathering (4 months)
  - Implementation tasks done in parallel (3 months)
  - CSC technical implementation (5 weeks)
- Department Master implemented in August 2011
  - Requirements Gathering (2 months)
  - Implementation Tasks done in parallel (3 months)
  - CSC technical implementation (6 weeks)
- Additional hierarchy created for Contracts in Sept 2011
  - Requirements Gathering (3 weeks)
  - Implementation tasks (2 weeks)
  - CSC technical implementation (1 week)
    - Export from DRM in parent/child (Less than a day)
    - Most work done outside of DRM, on systems receiving data

# DRM implementation at GDLS

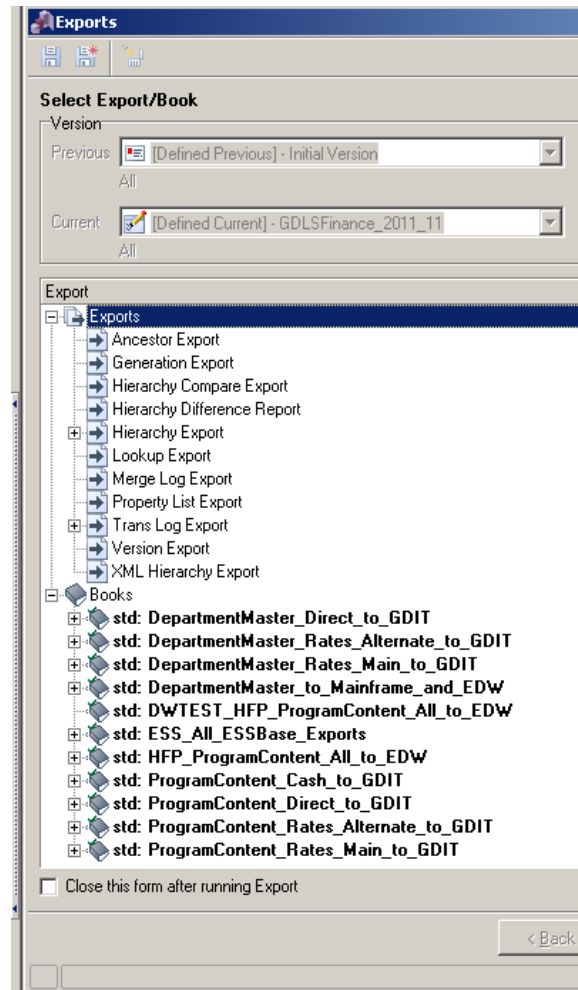
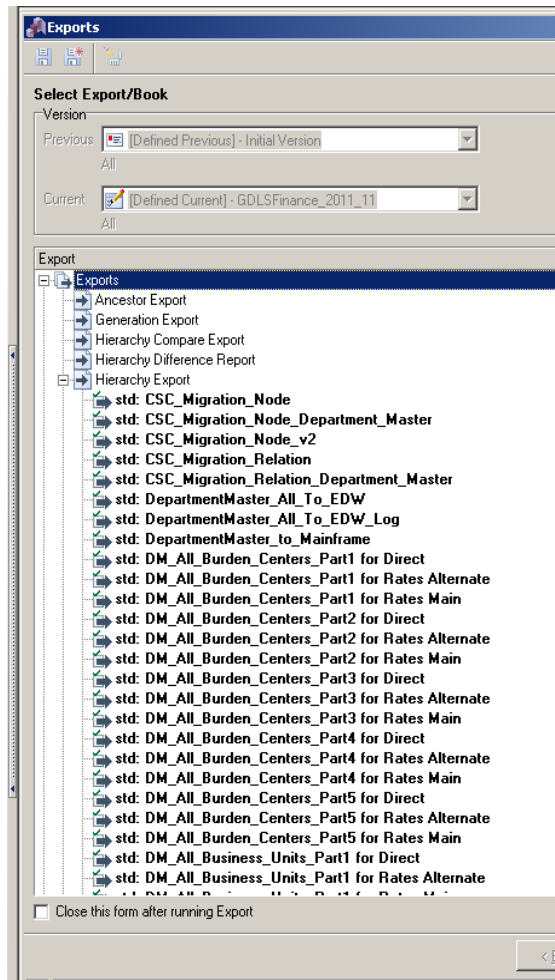
## Manual vs. Automated Hierarchies Exports

---

- Requirement
  - 8 Departmental Hierarchies needed for Hyperion applications
- Issue
  - Adding new departments could require creating a new department in main hierarchy and inserting into 6 other hierarchies
- Solution
  - One department hierarchy were created in DRM to support 7 Hierarchies in Hyperion Apps
  - The export functionality & additional properties were used to assist with creation of additional hierarchies
  - Separate Parent/Child export created for each level of hierarchy
  - Export Books created to run select groups of exports
  - Export files combined with a Unix script on an Informatica server
- Drawbacks
  - Additional Properties were needed to support using just one hierarchy in DRM
  - Additional Cost for development, couldn't use out of box DRM

# DRM implementation at GDLS

## Exports & Export Books



# DRM implementation at GDLS

## Departments: Data Input automation

---

- Issue: 29 Properties for each dept is too much inputting
- Formula's used
  - 7 of the properties are created via formula
  - Only Admins can create the formulas
  - Example: IF(ListContains(06;13;36;88;8E;92,PropValue(DM\_EDW\_LABOR\_CLUSTER),;),YES,NO)
- Lookup Tables created
  - 7 of the properties are created via lookup tables
  - Functional Admins maintain tables
- Values inherited from Parent
  - 3 of the properties are inherited from Parents
  - Users do have to input at parent level
- Remaining 12 values are input
  - Lists maintained to pick from for most properties
  - Functional Admins maintain lists

# DRM implementation at GDLS

## Departments: Data Input tab

The screenshot displays the 'DEPARTMENT\_MASTER - GDLSFinance\_2011\_11' application window. On the left, a tree view shows the hierarchy of departments under 'ALL\_BURDEN\_CENTERS', with 'L9590' selected. The main area shows a detailed data entry form for this department.

Property	Value	S...
Department Number	L9590	
Dept Name	GDSS FRINGE	
Location	MICH	
Plant	GBC	
Tax Entity	GDDB	
Functional Sort	CDC	
Manpower	N	
Business Unit	CSSC	
Function	E	
Forecast Dept	N	
Inventory Org	N	
Close Date Selection		
Close Date		
Close Flag	<input type="checkbox"/>	
Transfer To Dept		
State	MI	
State Name	MICHIGAN	
Available Hours	AVHRNONE	
Manufacturing Labor Type	Support	
Other Departmental Rollups (from ...)	OTHER_DEPARTMENTS	
Fringe Rollup (from LC)	9590	
Fringe Rollup Name	GDSS FRINGE_	
Direct Indirect (from LC)	I	
Employee Type	Management	
G&A Center	PSGA	
G&A Center Description	CSSC G&A	
Business Unit Name	CUSTOMER SERVICES AND SUPPORT	
Function Name	PRODUCTION DELIVERY & SUPPORT	
MfgCompCst	NO	
MinMax	NO	
Node was Changed	<input checked="" type="checkbox"/>	
CHANGED BY	kochensp	
CHANGED ON DATE	2011/10/26 05:24:11 AM	
ENTERED BY	adamczyk	
ENTERED ON	1995/02/27	

# DRM implementation at GDLS

## Lookup tables

The screenshot shows a 'Properties' dialog box with the 'Lookup Table' tab selected. The 'Current' property is 'DM\_LOOKUP\_STATE\_NAME' with a description of 'DM\_LOOKUP\_STATE\_NAME' and ID '1075'. The table below lists various lookup keys and their corresponding result values.

Lookup Key	Result Value
AF	AFGHANISTAN
AK	ALASKA
AL	ALABAMA
AR	ARKANSAS
ATR	AUSTRIA
AUS	AUSTRALIA
CA	CALIFORNIA
CNA	CANADA
EGT	EGYPT
EUR	EUROPE
FL	FLORIDA
FNN	FOREIGN
GA	GEORGIA

At the bottom of the dialog, there is an 'Add' button, two input fields for 'Lookup Key' and 'Result Value', and a note: 'To delete a row, select the desired row and enter the key combo Ctrl/Del to delete it.' Below these are 'OK', 'Cancel', and 'Apply' buttons.

# DRM implementation at GDLS

## Department Master - Interfaces

- We needed Department Master data feed directly to multiple locations
  - Oracle
    - Projects
    - General Ledger
  - Mainframe (This also feeds others)
  - Essbase
  - Data warehouse for query tools
- Oracle only supports loading data to interface tables through API's (Application Programming interface). Most of the tables we need to load department data to are not supported by API's.
- All Oracle data for departments is manually re-entered



# DRM implementation at GDLS

## Validations vs Verifications

---

- Validations
  - Validations are done in “Real time” when saving changes.
  - Cannot save if violates validation
- Verifications
  - Verification require running a batch.
  - Can save if violates verifications
- Examples of tests
  - Test of Length
  - Test of blanks
  - Testing for unique names between multiple hierarchies (dept and contracts) is basic functionality, and not a validation. But it is done in real time.
- Why not pick all as validations?
  - There are times when we need to get a new code in ASAP for Hyperion Apps, but we don't have all information. We put verifications on “Critical” properties, as opposed to all Properties.

# DRM implementation at GDLS

## Security/Access with additional Dimension

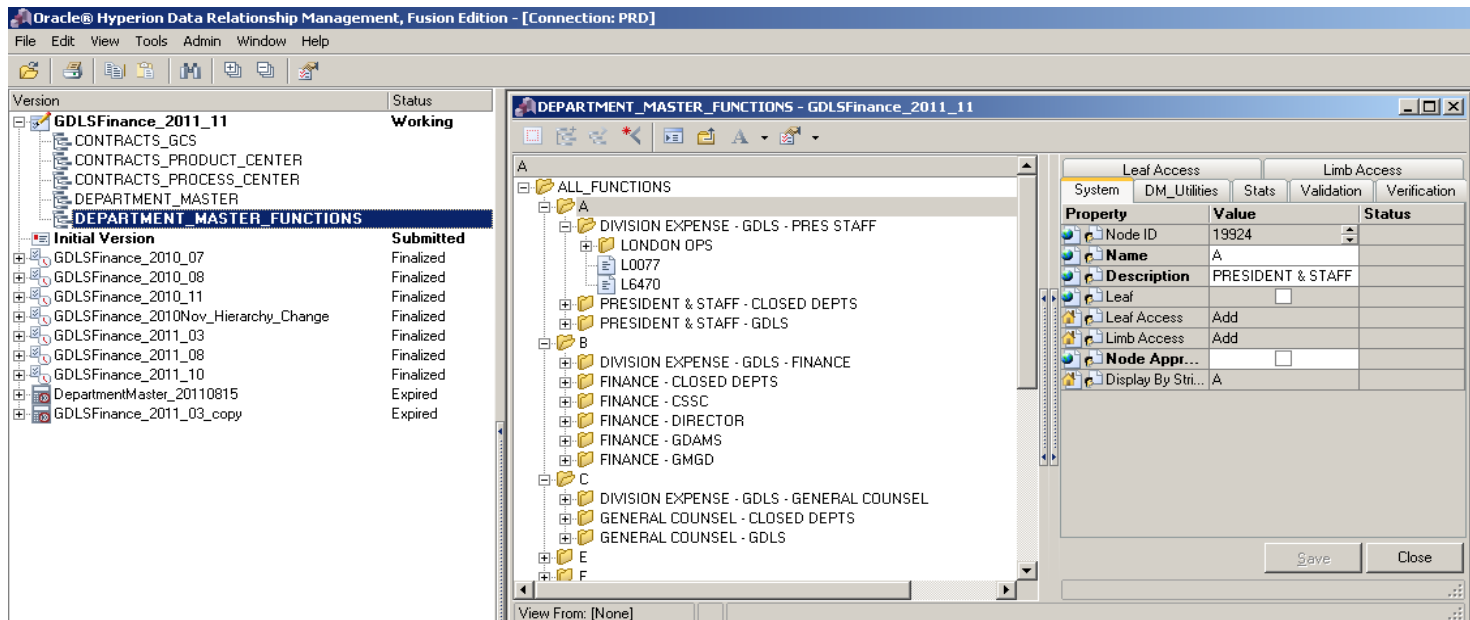
---

- With initial contract implementation, all finance users were set up as functional admins.
- Functional admins can change all hierarchies within a version. We did not want users entering department master data to have access to Contracts, or vice versa.
- Needed both contracts and departments in same version so we could enforce unique names between department master and program content.
- We still wanted some of the functional admin rights in finance.
  - Update Lists
  - Update lookups
  - Version Control
- During Dept Master implementation, it was decided to maintain two functional admins and set up all others as “users”.
- It cost a few days of CSC rework to change security around

# DRM implementation at GDLS

## Security/Access on Hierarchy

- One of the department hierarchies in DRM was set up for the functional area's reporting based on their org. charts
- Access was given to users in each function to adjust as needed.
- They have total control over their areas, but no access to others



# DRM implementation at GDLS

## Other issues to consider

---

- Sort Order Property
  - DRM exports in the order set in properties
  - Sequence in Data Warehouse used to retain sort order
  - Essbase/ODI creates in order received
- Storage Type Property
  - Requirement for a “Never Share” indicator at all non-zero level members
- Running Exports
  - We have two sets of users who can run Exports (Dept & Contracts)
  - DRM will allow multiple exports at the same time
  - Because of way our export runs through Informatica server, we can't have two exports running at the same time
  - I would suggest discussing this potential issue with IT developers

# DRM implementation at GDLS

## Finance users doing more with the Automator

---

- Process used to create initial hierarchies
  - Data maintained in Excel throughout many iterations
  - CSC created a loading process from an Excel file using a staging table, PL/SQL and the DRM import tool.
  - Files sent to CSC admin for loading. Loads took around an hour
  - Cost GDLS finance budget money
- Process used to create additional hierarchy: All done by Finance
  - Created Top Node in DRM
  - Created Parent/Child file in Excel
    - Once program office determined roll-up, this took about 30 minutes
  - Used Automator functionality in DRM
  - Loads took about 10 minutes

### Format of File

Action/Version/Hierarchy Name/Child/Parent/Node is leaf ( ) = false 1 = true

Actions used: “Add” for upper levels, “Insert” for lowest level

# DRM implementation at GDLS

## Q&A



# DRM implementation at GDLS

---

**Back-up Slides**

# DRM implementation at GDLS

## Set up - Security/Access

- Three options were evaluated for setting up access to DRM
- Criteria for selection:
  - Limit the number of users that could interact with DRM
  - GDLS finance to administer the data as much as possible
  - GDLS finance should not get too involved in the technical details



# DRM implementation at GDLS

## Security – Option A

Access Type*	Example of who might fill the role	Typical Tasks	Notes:
<b>DRM Administrators</b>	Technology Team	<ul style="list-style-type: none"> <li>● Create Property Definitions</li> <li>● Create Users</li> <li>● Configure User Level Security</li> <li>● Create/Assign Validations</li> </ul>	<ul style="list-style-type: none"> <li>● At least one required in DRM to perform DRM configuration such as creating properties.</li> <li>● Can see and edit all content in DRM</li> </ul>
<b>DRM Functional Administrators</b>	Functional Leaders on Business Teams (SMEs)	<ul style="list-style-type: none"> <li>● Update LOV property values</li> <li>● Maintain User Level Security</li> <li>● Create Versions/Hierarchies</li> <li>● Create Standard (shared) Exports and Queries</li> <li>● Add/Move Nodes</li> <li>● Edit Property Values</li> <li>● Assign Validations</li> </ul>	<ul style="list-style-type: none"> <li>● Can see and edit ALL content in DRM.</li> <li>● Would maintain some content such as LOV values unless the restricted (limited) value sets are managed via additional small hierarchies</li> </ul>
<b>DRM Users</b>	Business Content Management Specialists	<ul style="list-style-type: none"> <li>● Add/Move Nodes</li> <li>● Update Property values</li> <li>● Run Exports</li> <li>● Run Queries</li> <li>● Edit Property Values</li> </ul>	<ul style="list-style-type: none"> <li>● Rights are dependent on assignment to Node Access Groups and the rights that Node Access Group is given on individual nodes.</li> <li>● Would require additional security if restricted value sets are managed via small hierarchies as opposed to LOVs.</li> </ul>

# DRM implementation at GDLS

## Security – Option B

Access Type*	Example of who might fill the role	Typical Tasks	Notes:
<b>DRM Administrators</b>	Technology Team	<ul style="list-style-type: none"> <li>● Create Property Definitions</li> <li>● Create Users</li> <li>● Configure and Maintain User Level Security</li> <li>● Create/Assign Validations</li> <li>● Update LOV property values</li> <li>● Create Versions/Hierarchies</li> <li>● Create Standard (shared) Exports and Queries</li> </ul>	<p>At least one required in DRM to perform DRM configuration such as creating properties. Can see and edit all content in DRM</p> <ul style="list-style-type: none"> <li>● Would maintain some content such as LOV values unless the restricted (limited) value sets are managed via additional small hierarchies</li> <li>● Would need to create all versions – roll monthly versions</li> </ul>
<b>DRM Users</b>	Business Content Management Specialists	<ul style="list-style-type: none"> <li>● Add/Move Nodes</li> <li>● Update Property values</li> <li>● Run Exports</li> <li>● Run Queries</li> <li>● Edit Property Values</li> </ul>	<ul style="list-style-type: none"> <li>● Rights are dependent on assignment to Node Access Groups and the rights that Node Access Group is given on individual nodes.</li> <li>● Would require additional security if restricted value sets are managed via small hierarchies as opposed to LOVs.</li> </ul>

# DRM implementation at GDLS

## Security – Option C (Picked)

Access Type*	Example of who might fill the role	Typical Tasks	Notes:
<b>DRM Administrators</b>	Technology Team	<ul style="list-style-type: none"> <li>● Create Property Definitions</li> <li>● Create Users</li> <li>● Create/Assign Validations</li> </ul>	<ul style="list-style-type: none"> <li>● At least one required in DRM to perform DRM configuration such as creating properties.</li> <li>● Can see and edit all content in DRM</li> </ul>
<b>DRM Functional Administrators</b>	Functional Leaders on Business Teams (SMEs)	<ul style="list-style-type: none"> <li>● Update LOV property values</li> <li>● Create Versions/Hierarchies</li> <li>● Create Standard (shared) Exports and Queries</li> <li>● Add/Move Nodes</li> <li>● Edit Property Values</li> <li>● Assign Validations</li> </ul>	<ul style="list-style-type: none"> <li>● Can see and edit ALL content in DRM</li> <li>● Would maintain some content such as LOV values.</li> </ul>