



Model Design II: Tunnel

*Presented by:
Roger Williams
Dimensional Insight, Inc.*



Model Design : Yesterday, Today and Tomorrow



Model Design :
Yesterday, Today and Tomorrow



■ Diver Pro/Net

- Ad Hoc
- Product Focus
- Data analysis and Reporting
- Self-directed "data mining"
- Bottom-up approach
- User-controlled
- Response Less Issue
- Diver training

■ DivePort

- Targeted
- Solution Focus
- Information delivery
- "Just-in-time" info
- Top-down approach
- Developer-controlled
- Response Issue
- Application training



■ Diver Pro/Net

- Fewer Models
- Larger Models
- Many Dimensions
- Many Infos
- More Memory

■ DivePort

- More Models
- Smaller Models
- Fewer Dimensions
- Fewer Infos
- Less Memory



- DivePort Today
 - Reports-on-line
 - Column page layout
 - Ad Hoc models
 - External QuickView Models
 - Traditional Model re-design
 - dimensions
 - block factor
 - skip_build
 - Rely on external links to Pro/Net Diver
 - Same old, same old



- DivePort Tomorrow
 - Applications
 - Dashboards
 - Scorecards
 - Manual page layout
 - Non-traditional Model design
 - More, smaller Models
 - Canisters
 - Tunnel
 - Parameters
 - Less attention paid to Pro/Net Diver
 - New learning curve



Model Design : Making the Transition



Model Design :
Making the Transition



Model Design : Making the Transition

Yesterday

1. Define data
 2. Explore data relationships
 3. Model design
 4. **AD HOC Models**
 5. Create reports
- Same old, same old
 - Integrator
 - DiveMaster
 - Builder - Ad Hoc Models
 - ProDiver
 - DivePort

↓ Hospital

↓ Product Name

DRG

Physician

Patient



Model Design : Making the Transition

↓ Hospital
↓ Product Name

DRG

Physician

Patient

Tomorrow

1. Define report(s)
 2. Identify data
 3. Model design
 4. **DETAIL/INDEXED** Models
 5. Create reports
- Same old, same old
 - New Learning Curve
 - Detail/Indexed Models`
 - Tunnel
 - Parameter Passing
 - DiveRef Links
 - Report Templates



The Diver Solution

- Integrator
 - Tunnel
 - Model input object
 - List input object
 - Builder output object
- Builder
 - Tunnel
 - Building detail models
- Diver
 - Tunnel
 - Reference Links
- DivePort
 - Portal variables



Model Design : Show and Tell



Model Design :
Show and Tell



DRG - Physician - Patient Example

Physician Performance
Management (PPM) Application



Model Design : Show and Tell - Example

Ad Hoc Model

- 13 dimensions +1
- 21 summaries
- 25 infos
- 69 detail dimensions
- 21 detail summaries
- 0 detail infos

- 97833 records
- 346.4 meg
- elapsed time 0:01:00

Detail / Indexed Model

- 3 dimensions +1
- 21 summaries
- 0 infos
- 69 detail dimensions
- 21 detail summaries
- 0 detail infos

- 97833 records
- 89.7 meg
- elapsed time 0:00:12



Builder

- Build indexed.mdl from indexed.dsc

DivePort

- DRG
 - Marker Portlet
 - drg.mrk (uses indexed.mdl)
 - contains these QuickViews
 - » Hospital
 - » Product Name
 - Click on DRG value



DivePort

- Physician

- diveref on DRG

```
diverref:report["physician.dvp  
{Hospital="$(Hospital)",Product Name="$(Product Name)",DRG="$TEXT"}  
.Attending Physician,  
_Report_Template=physician.rep]
```

- physician.dvp

- » access physician.tnl and controls columns and calcs

- physician.tnl

- » reads indexed.mdl to access records for DRG selected and builds a "real" model physician_tnl.mdl

- physician.rep

- » report format applied

- » contains diveref for patient report

- click on Physician value



DivePort

- Patient

- diverref on Physician

```
diverref:report["patient.dvp\  
.Patient,  
Attending Physician=\"$TEXT\  
_Report_Template=patient.rep]
```

- patient.dvp

- » access physician_tnl.mdl and controls columns & calcs

- patient.rep

- » report format applied



Builder

- Multiple real Models
- Multiple detail/indexed Models

DivePort (v6.3)

- Lots of Portlets
 - Measures
 - Indicators
 - Markers
- Integrator / Tunnel scripts
 - Same script w/ PARMS used > 40 on one page
 - Execution / memory shared
 - Need to react to user selections



Model Design : Tunnel and Other Useful Stuff



Model Design :
Tunnel and Other Useful Stuff



The Diver Solution: Tunnel

- What is Tunnel ?
- Model Types
- Model Building Options
- Indexed Access to Detail Models
- Tunnel Parameters
- Tunnel in DivePort
- Configuration - Set-up & Security
- What Are You Doing?





TUNNEL - What is it?

Integrator script initiated by a DI Client, executed on a DiveLine server which returns a Model.

Integrator-based provides wide range of functionality and possibilities.

How and when to implement Tunnel ?

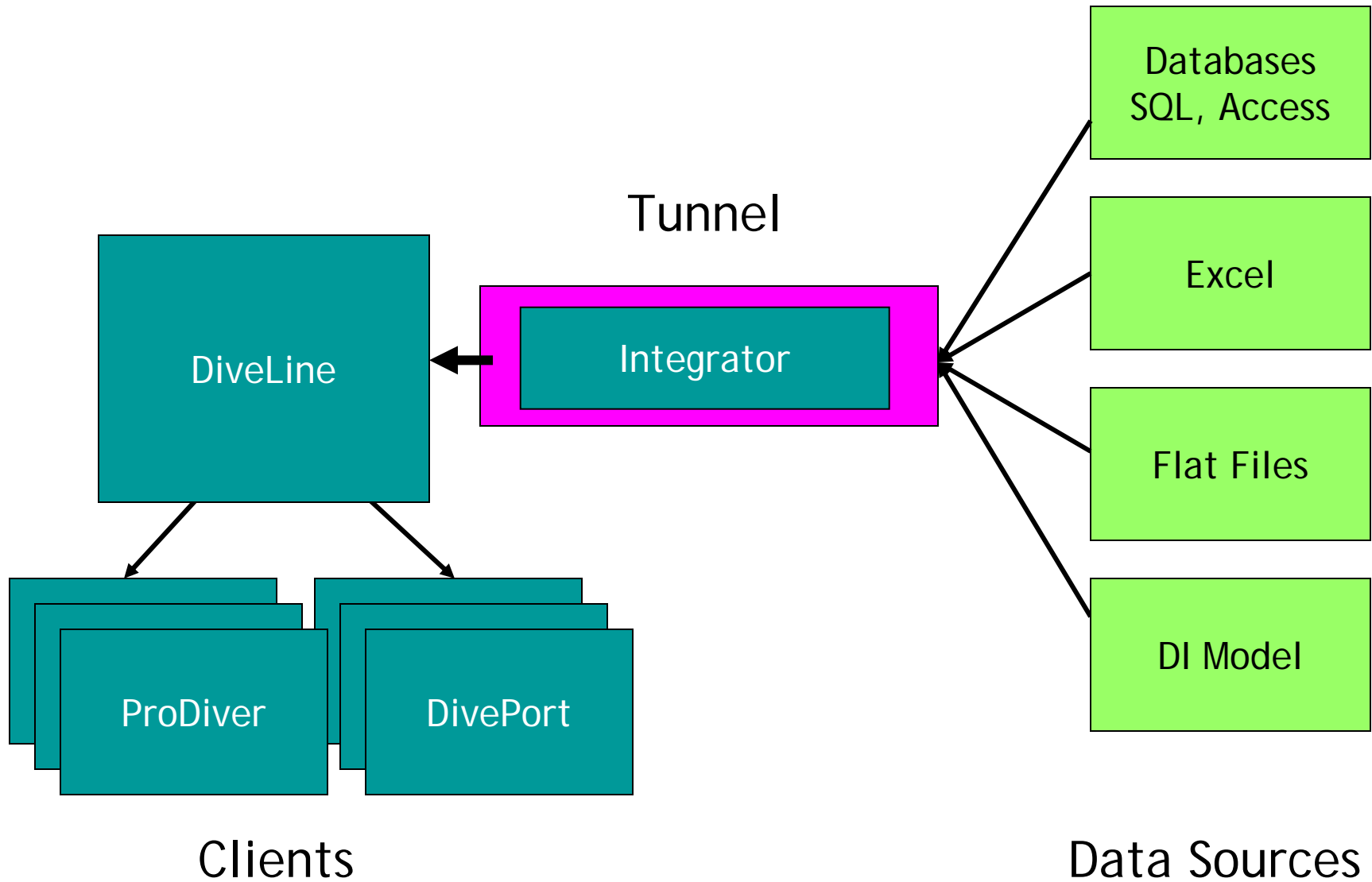


TUNNEL - What is it?

- Models built on demand
 - "Dynamic" Memory Model
 - "Static" Physical Model
 - Properties identical to regular Model
- Integrator Enabled
 - Embedded within DiveLine
 - ".tnl" script executed by DiveLine directly
 - object_type="tunnel" attribute in OUTP object
- Understood by DiveLine Clients
 - Pro/Net/Cell Diver can open ".tnl" script
 - "diveref=" links can reference ".tnl" scripts
 - All DiveLine clients can open Tunnel Markers



TUNNEL Flow





Model Types

- Traditional “Ad Hoc” Models
 - Shotgun approach
 - Fewer in number
 - Possible response time issues
- Small “Report” Models
 - Targeted
 - Greater in number
- “Canister” Models
 - Multiple metrics in same column
 - Multiple data summarizations



Model Building Options

- Traditional “Batch”
- “On Demand” or “Real-Time”
- “Near Real-Time”





“On-Demand” / “Real-Time”

- Integrator (.tnl)
 - access data (flat files, SQL Server)
 - transform data
 - Model design
 - build Model (Memory Model)
 - present Model

- DivePlan (.dvp)
 - combine Models, Tunnel scripts
 - define columns, calcs, groups, etc.

- Marker (.mrk)
 - define Report
 - define appearance

- DI Client
 - Only DiveLine clients (not Diver)



“Batch”

vs

“Real-Time”

- Integrator (.int)
- Builder (.dsc)

Static
Physical Model
(.mdl)

- DivePlan (.dvp)
- Marker (.mrk)
- DI Client

- Integrator (.tnl)

Dynamic
Memory Model
(.tnl)

- DivePlan (.dvp)
- Marker (.mrk)
- DI Client



“Batch”

vs

“Real-Time”

- Large data volumes
- Wide production windows
- Pre-calculated data views
- Report consistency

- Response Time

- Small data volumes
- Secondary dives
- Up-to-the-minute data
- Different metrics

- Report inconsistency



“Near Real-Time” Process

- Hybrid: “Batch” process run “On-Demand”
- Multiple Integrator TASKS
 - Build physical “static” Model
 - Marker based upon “static” Model
- Report consistency
 - “On-Demand” Model constantly changing
 - “Batch” Model is “static”
- Manage impact on users



Tunnel Integrator Script

"On-Demand"

```
object 'OUTP' "out" {
  input          = "filter",
  output_type   = "tunnel",
  dimensions     = {
    "Supplier_Code" ,
    "Product_Code"  ,
    "Sales_State"   ,
    "Store_Number"  ,
    "Date_Of_Data"  ,
  },
  summary        = {
    "Case_Qty"
  },
};
```

"Near-Real Time"

```
object 'OUTP' "out" {
  input          = "filter",
  output_type   = "builder",
  output        = "sales.mdl",
  dimensions     = {
    "Supplier_Code" ,
    "Product_Code"  ,
    "Sales_State"   ,
    "Store_Number"  ,
    "Date_Of_Data"  ,
  },
  summary        = {
    "Case_Qty"
  },
};
```

```
object 'OUTP' "out_display" {
  output_type    = "tunnel",
  return_model   = "sales.mdl",
};
```



Tunnel Integrator Script

- Extension “.tnl”
- One or more TASKS
- OUTP object w/ output type = “tunnel”
- “Physical” or “Memory” Model



Tunnel Data Source

- “Live” production data
 - SQL Server
- Any Integrator source data
 - “Staged” production data
 - denormalized Integrator data
 - DI Models
- Indexed Access to Detail Models



Indexed Access to Detail Models

- Create an indexed Detail Model

Model Name	Detail_Filter.mdl		
# records	382,543,798		
Block factor	100,000		
Model Size	30.9 GB		
Elapsed time	1:20:40		
2 Core Dimensions			
	Supplier_Code	95 values	
	Sales_State	51 values	
5 Detail Dimensions			
	Supplier_Code	Sales_State	Date_Of_Data
	Store_Number	Product_Code	
1 Detail Summary			
	Case_Qty		

- Create a .tnl script



Indexed Access to Detail Models

```
object 'TASK' "Detail_Filter_Model" {
  inputs = {
    "QV_values" ,
    "model"      ,
  },
  output = "out",
};
```

```
object 'INPT' "QV_Values" {
  input_type = "list",
  columns    = { "Supplier_Code" , "Sales_State" },
  values     = {{ "KAI"           , "AZ"           }},
};
```

```
object 'INPT' "model" {
  input_type    = "model",
  model         = "../Models/Detail_Filter.mdl",
  detail       = "true",
  filter_input  = "QV_Values",
  filter_columns = {
    "Supplier_Code" ,
    "Sales_State"   ,
  },
};
```

```
object 'OUTP' "out" {
  output_type = "tunnel",
  input       = "model",
  dimensions  = {
    "Supplier_Code" ,
    "Product_Code" ,
    "Sales_State"   ,
    "Store_Number"  ,
    "Date_Of_Data"  ,
  },
  summary     = { "Case_Qty" },
};
```



Indexed Access to Detail Models

- Filter
 - Supplier_Code = KAI
 - Sales_State = AZ
 - Records = 140 (out of 382,543,798)
- Sequential file = 0:05:48 (.txt)
- Indexed Model = 0:00:03 (.mdl)
- FAST for small amounts of data



Tunnel Parameters

- Automatic variables passed to “.tnl” by DiveLine
- Report Palette & Item Links via “diveref=”
- Uses :
 - Filter data
 - Identify INPT/OUTP files
 - Read/write to specific directories



Tunnel Parameters

- Automatic variables passed to “.tnl” by DiveLine
 - `$_user` User name
 - `$_pid` Process ID (pid)
 - `$_diveline` Dataroot directory
 - `$_diveline_data` Dataroot/data



Tunnel Parameters

- "Near-Real Time"

```
object 'PARM' "parms" {  
    parms = { { name = "_USER" , default = "xxx" }, }  
};
```

```
object 'OUTP' "out" {  
    input          = "filter",  
    output_type    = "builder",  
    output         = `c:/DI_Users/$_USER/sales.mdl`,  
    dimensions     = {  
        "Supplier_Code" ,  
        "Product_Code"  ,  
        "Sales State"   ,  
        "Store_Number"  ,  
        "Date_Of_Data"  ,  
    },  
    summary        = {  
        "Case_Qty"     ,  
    },  
};
```

```
object 'OUTP' "out_display" {  
    output_type    = "tunnel",  
    return_model   = `c:/DI_Users/$_USER/sales.mdl`,  
};
```



Tunnel Parameters

- Report Palette & Item Links via “diveref=”
 - Parameters in “{...}” between quoted Tunnel script name and the “.” separating the dive_dimension

`diveref:table["sales.tnl"{Month="Jan"}.Salesrep, Region="North"]`

- \$TEXT
- QuickView Values
- Report Palette Variables



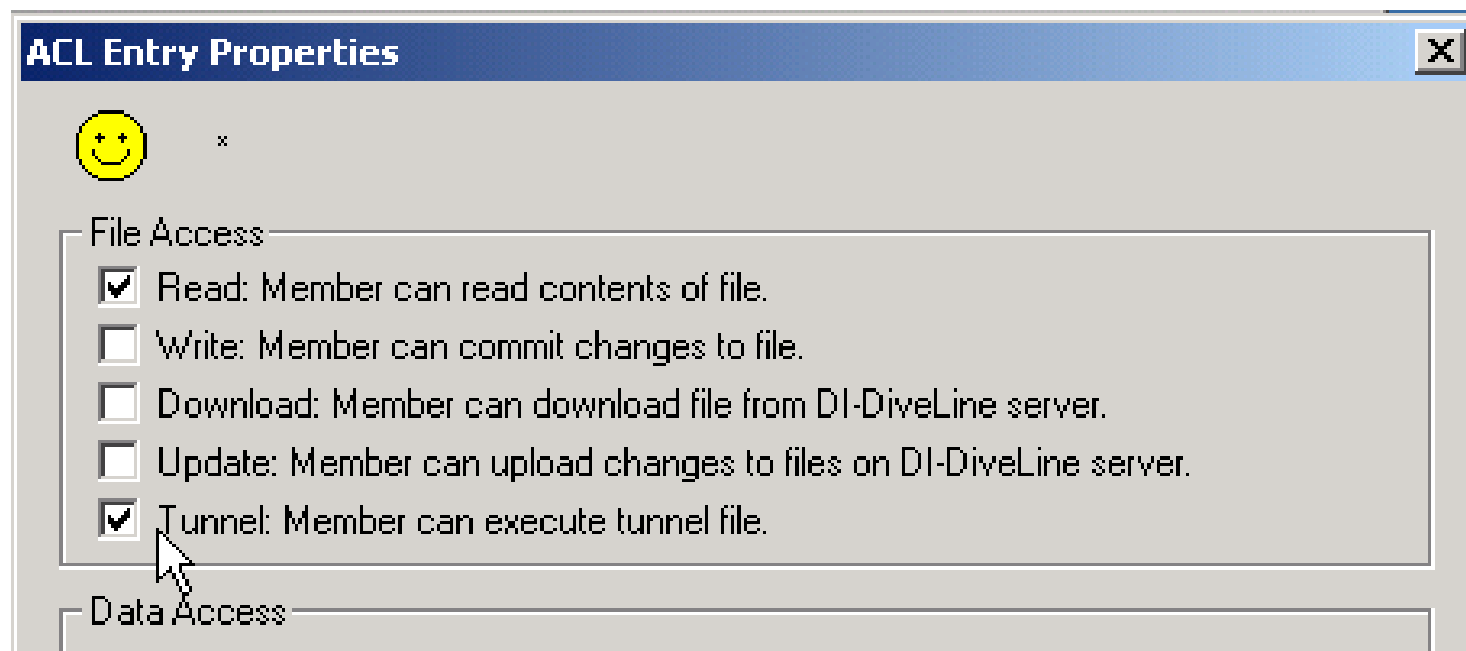
Tunnel in DivePort

- Never Cached
- Multiple “Real-Time” Reports may show different results
- Create “Static” Model to view “Near Real-Time” Reports with consistent results
- Combine “Static” Model with “Real-Time” Model



Configuration - DiveLine Setup

- File ACL (Access Control List) must be set to allow Tunnel file execution





- DiveLine ACL treats “.tnl” just like a Model.
- Access is granted or denied on the Memory Model not on the source data.
- Additional Integrator objects required to limit data processed and contained in Memory Model.



“Real-Time” Report

- Create TUNNEL Integrator script
 - INPTs to access desired data
 - PROCs to manipulate data
 - OUTP to build a Memory Model (output_type=tunnel)
- Create Marker
 - Open .tnl in ProDiver
 - Dive into Memory Model
 - Create Report
 - Save as .mrk
- Add to DivePort Page
 - Add a Portlet specifying saved .mrk



“Near Real-Time” Report

- Create TUNNEL Integrator script
 - TASK 1
 - INPTs to access desired data
 - PROCs to manipulate data
 - OUTP to build a “static” disk Model (output_type=builder)
 - TASK 2
 - OUTP to open “static” disk Model
- Create Marker
 - Open “static” disk Model in ProDiver
 - Dive into “static” disk Model
 - Create Report
 - Save as .mrk
- Add to DivePort Page
 - Add a Portlet specifying saved .mrk



“Study” or “Panel” Analysis

- Create Patient ID File
 - Open “standard” Model in ProDiver
 - Perform analysis
 - Dive on Patient ID
 - Save Marker to DI_USER_user folder
- Create TUNNEL Integrator script
 - TASK 1
 - INPTs to access desired data (Patient ID file just saved)
 - PROCs to manipulate data
 - OUTP to build a “static” disk Model (output_type=builder) and save into DI_USER_user folder
 - TASK 2
 - OUTP to open “static” disk Model
- Add to DivePort Page
 - Add a Portlet specifying saved .mrk



“Combo” - Static Model & Real-Time Model

- Create Static Model (actual)
- Create TUNNEL Integrator script (goals)
- Create DivePlan
 - Static Model
 - Tunnel script
- Create Marker
- Update TUNNEL input (goal) using
 - Lookup Manager
 - Future DivePort Input Portlet



“Combo” - Static Model & Real-Time Model

- Sales_Goals.tnl

```
object 'INPT' "in" {
    input_type      = "Filein",
    file_type       = "column_headers",
    filename        = "../models/sales_goals.tbl" };

object 'OUTP' "out" {
    output_type     = "tunnel",
    input           = "in",
    dimensions      = {
        "Team",
        "Salesperson"
    },
    summary         = {"Goal" },
};
```

- Sales_Actual_Goals.dvp

```
merge_dbs={
    {
        dbname="sales_actual.mdl",
        diveline_dbname="sales_actual.mdl",
    },
    {
        dbname="sales_goals.tnl",
        diveline_dbname="sales_goals.tnl",
    }
},
```



TUNNEL

What are you doing ?



Model Design II: Tunnel

*Presented by:
Roger Williams
Dimensional Insight, Inc.*