FILEMAKER MASTER CLASS – BROADER VIEW: DETAIL

2015 [locations/dates to be determined]
Presenter: Ray Cologon, Director of Development, NightWing Enterprises Pty Ltd

SESSION FORMAT:
The class will take place over three days and will comprise six substantive and wide-ranging topics.

Topics will commence with a presentation and overview of the key points, including or followed by demos as appropriate. Subsequently for each main topic, questions and discussion will be invited, after which a workshop format will be adopted to explore key ideas emerging from the discussion.

Examples and proof-of-concept files will be available to participants, and further examples contributed by participants during the classes will (with permission) also be made available to attendees.

Interaction between participants is encouraged as part of a mutual learning, exploration and professional development process.

Detailed Content Listing:

Day 0 (The evening before formal classes begin)

- Dinner in a restaurant (not covered by the event fee)
- Introduction of Presenter and Participants
- Overview of the Schedule

Day 1:

1. Introduction

- Logistics and Orientation
  - Operational Details for the 3-Day class
  - Background to the Master Class
  - Review of Content and Topic Depth vs Breadth
  - Protocols for Participation and Discussion
- Framework and Purpose of the Master Class
  - Concepts: Advanced Techniques and Approaches
    - What Sets FileMaker Apart
    - The Platform vs the Individual Tools
    - Scope and Limitations; Playing to Inherent Strengths
    - Deciding when to (or not to) 'Push the Envelope'
  - New Challenges to Rethink & Reappraise
  - Finding Opportunities, Vistas and Wider Horizons
  - Opportunities for Exchange & Collaboration with Peers
  - Bringing Advanced FileMaker Practice into Focus

2. Perspectives on Solution Design

- Taking the "Design Rethink" Opportunity
  - Layering within Layouts
  - Tools for Achieving Visual 'Depth'
  - Slide Controls as a Dimensional Framework
  - Programmatic Visibility and Overlaying Options
  - Popovers as an Interleaving/Layering Tool
- The Simplicity Principle as a Solution Design Axiom
  - When Too Simple is Too Hard!
  - Solutions that "Handles Complexity so that the User Won't Have To"
  - A Core Paradigm: Natural Progression, Logical Sequencing, Temporal Flow
  - The Right Amount of Complexity is Empowering (for the User)
- Evolving Concepts of User Interaction Modelling
  - Establishing Logical Sequences
  - Mirroring Patterns from the Physical World
  - Approach: From Wide-View to Close-Up
  - The Short-Term Memory "Handful"
  - Natural Hierarchal Organising Methods
- Getting More out of FileMaker Interface Elements
  - Appreciating the Importance of Space
  - Establishing Consistency - the Implicit Framework of your Solution
  - Representing Conceptual Relationships Visually
  - The Dark Art of Object Metaphors:
    Harnessing Familiarity to Engender (or Mimic) an Intuitive User Experience

- Doing More with Less: Visual Dynamism Options
  - Establishing Depth (and its meanings vis-á-vis proximity)
  - FileMaker 13 Dimensionality and Spatial Illusion
  - New Tools for Layering Objects and Content
  - Space in the Third Dimension
- Achieving Precision, Subtlety & Depth with Object/Attribute Controls
  - Padding as a Dimension of Space
  - Shadows (vs the Flat Universe of v12)
  - Intelligent Use of Object States
  - Transparency and Conditional Formatting
- Harnessing the New FileMaker Design Surface
  - Evolution and Directions of the Platform
  - The Shift to a Developer Focus
  - Review of Specific Interface Innovations
  - Layout Tools (Styles, Themes, Slide Control, Popover...)
  - Scope of Action (WebDirect, iOS Gestures...)
  - Other (Hidden Attributes, Script & Calc Operations...)
- Designing for Device-specific Operability
  - The Mac / Win / iOS / WebD challenge
  - User Interaction Modeling for Context
  - Toolsets for Delivering a Diverse User Experience
  - Integrated (branching) multi-interface solutions
  - Alternate application or 'front end' options

3. Data and Structure: FileMaker Application Models

- Reviewing Data Modelling and Entity Relationships
  - The Conventional ERD and FileMaker
  - Perspectives on Table Structure
  - Data Models as a Codification of Syntax
  - Mapping Requirements to Form a Model
  - Identifying Associative Entities
- A Deeper Look at the Handling of Associative Entities in FileMaker
  - Considering uses and Limits of Conventional Join Handling:
    - One to One, One to Many
    - Many to Many & Join Tables
  - Multi-Key Functionality
  - Compound Key Joins
  - Matrix Key Methodologies
- Conceptual Separation between Tables and Table Aliases (TOs)
  - ERD vs Relationships Graph
    - Graph to support Data (Calcs) and Schema
    - Graph to Support Interface and Operations (eg Scripts)
    - Graph to Support Reporting
  - Avoidance of Circular References
  - TOGs and Multiple Named Aliases
- Graph Management and its Implications
  - FileMaker Data Caching Model
  - Origins and Benefits
  - Ramifications and Performance Hurdles
  - Graph Design Methodologies
  - Merits of a Delineated Management Approach/Strategy
  - Alternative Strategies and their Respective Pros and Cons
- Optimal Use of Themes and Styles in FileMaker
  - FileMaker's Visual Hierarchical Principle
  - Where the CSS Resides (and the significance of the Object Library)
  - Theme Portability and Adaptability
  - Building Your Own Visual Library
  - Updating the Theme (Master Copy), not the Solution
- File Structures: Data Separation & Multi-Application Solution Options
  - The Conventional "Data Separation Model" (and the Separation Update "Myth")
  - Multi-Application Solutions

- Understanding Context Dependencies: Refresh Issues
- Data Archiving and 'Vertical Segmentation'
- ESS and Distributed Data Models

– Multi-Platform & Multi-Device Design Considerations
- Available Script Commands by Platform
- Reviewing Scripts for Alternate Deployment Scenarios
- Branching vs Parallel Processes
- Error Handling across Platforms and Devices

– Understanding and Managing File Interdependencies
- Cascading References
- Mixed storage via ESS
- Circular External Source References – The Solution Exit Conundrum
- Efficiency Implications
  - Adopting Cascading Dependencies as the Default File Architecture
  - File Dependencies and Closing Script Behavior
- Strategic Alternatives:
  - Use of Control Architectures to Ameliorate File Dependency Issues

– Versioning and its Implications and Practicalities
- Roles of Version and Subversion
- Build Incrementation Options and Merits
- Value and Options for Detailed Version History
- Other Related Support Considerations
- Live Development Issues, Risks and Risk Mitigation
- Version Incrementation and Migration Tools
- Discussion of 'Delta Systems' Demo

Day 2:

4. Process Control: Advanced Scripting Techniques

– Abstraction and the Principles of Modular and Reusable Code
- The Uses and Abuses of Sub-Scripts
- Cascading States and Script Threading
- The Call-Stack and Button Attributes — Halt • Exit • Resume • Pause
- Local Variables and Script Zero

– Precise Timing Control in Loop and Interaction Applications
- Get(CurrentTimeUTCMilliseconds) [formerly: Get(UTCmSecs)]
- Double-Click in FileMaker Interaction Modeling
- Loop-Based Precision Scripted Pauses
- Loop Pause Interface and Refresh Considerations

– Optimal Solution Script Architectures
- Meta-Scripts for Error Handling
- Generic Handler Scripts for Navigation, User Feedback Dialogs etc.
- Use of Parameters to Extend the Scope of Individual Scripts
- Options for Passing/Parsing Multiple Parameters

– Management of Lengthy Script Sequences and User Feedback Options
- Script Logging and Alternate Debugging Techniques
- Block Code Commenting for Lengthy Scripts
- Use of 'Status Pane' Feedback Text
- Flash Dialogs and Progress Bars in v13
- Management of Allow User Abort States

– The Concept of "Processing in Place" vs Macro Coding
- Traditional Scripting Frameworks and "User Interaction Replication"
- Uses of the "Allow Create" Relationship Attribute
- Making Use of Pivot Relationships
- Multi-Directional Pivot Structures
- Hierarchic (Record Set) Replication

– Server and Client Trade-Offs including Perform Script on Server
- Context-Sensitive Expression Evaluations
- Load-Balancing between Host and Client
- Robots and Server-Side Scripting
- Perform Script on Server [ ] Script Command

– Advanced Control and Automation Techniques using Script Triggers
- Understanding Limitations of Script Triggers

- Controlling Trigger Execution (eg Suppressing Triggers Resulting from Scripted Actions)
- Use of OnTimer Triggers for Automation
- Approaches to Using OnKeystroke Triggers

– Code Architectures for Solution Management and Control
- Control Files and their Uses
- The Anatomy of Stubs, Models, Utility Files
- Versioning and the DevTool Concept
- Automated Migration and Roll-out

– Control Systems for Managing Version Deployment
- Different types of External Data Dependencies
- Scenarios Where File Dependency Issues Arise
- File Triangulation Methodologies
- Conceptual Map of Control File Processing

5. Code Logic: The FileMaker Calculation Engine

– Steps Towards Optimal Handling of Code and Data
- The "Swiss Cheese" Proposition
- Strategies for Minimising of File Stress
- Development Techniques for Creation of "Clean" Code
- Deployment Practices for Optimal Integrity

– Logical Dependencies, Redundancy & Code Efficiency
- Active Record Limits for FileMaker Calculation Dependency Management
- Behavior & Uses of Global Calculations
- Uses of Indirection and Associated Issues
- Considerations Regarding File Dependencies

– Logic in the Interface Layer
- File Security as a Bridge Between Schema and Interface
- Layout Calculations and their Uses
- Calculating Script Parameter Values
- Image State-Aware Objects
- Declaring Variables from Within the Layout

– Compound Syntax Challenges
- Concatenated and Nested Calc Expressions and Their Uses
- Formation of Complex Expressions Using Let( ) and Calculation Variables
- Use of $local Variables within Calc Syntax
- Optimisation of Long/Complex Expressions
- Judicious Use of Calculation Commenting

– Code Optimisation and the FileMaker Calc Engine
- Writing Calc Code for Brevity
- Elimination of Redundancy and Repetition
  - DRY vs SSOT as Rules of Thumb
  - Let( ) as a Frame for Clean Syntax
  - Writing Self-Commenting Code
- Minimisation of Dependency (Reduction of Dependency Depth)

– Native Calculation Power vs Extensibility with Plug-Ins and Custom Functions
- Calculation Simplification Using Custom Functions
- Function and Calc Dependency on Plug-Ins
- Extension of Calculation Scope Via Recursion
- Custom Functions and Performance Considerations
- The Plug-In API — Issues and Limitations

– Examples of Alternate Code Models for Complex Calculations
- Conventional, Legacy and Longhand Code Formation and Syntax
- Setting and Controlling Calculation Context
- Applications of Calc, Local and Global Variables in Calculation Expressions
- Understanding and Using Head and Tail Recursion

– Techniques for Controlling Calculation Dependency
- Localisation of FileMaker Dependency Management
- Referential Basis of Calculation Dependency
- Use of Evaluate( ) and GetField( ) to Break Managed Dependencies
- Use of Evaluate and Null References to Create/Introduce Managed Dependencies

Day 3:

6. Designing for Scalability

- FileMaker Development Life-Cycle Overview and Options
  - Conventional Single-Developer Systems
  - Version Control and Staged Deployments
  - Iterative Cycling and FileMaker:
    Design > Build > Test > Deploy
  - Segmented Development
  - Team Development Strategies
- Frameworks for Design Validation and Load Testing
  - Development Environments & Custom Tools
  - Design Briefs and Prototypes
  - The Uses of Stub Files and Mock-Ups
  - The Pristine Development Master Copy
  - Migration Utilities:
    - Legacy Systems and ETL Processing
    - Version Updates
- Design and Development Techniques that Scale vs Those that Fail
  - Indexing and Performance
  - Leanness and Efficiency of Code and Structure
  - Dynamic Values and Cascading Dependencies
    - Unstored Calculations
    - Summary Fields
  - Graph Optimisation
- Special Considerations and Optimisations for Large-Scale Solutions
  - Elimination of Unstored Values
  - Reduction of Data Volume
  - Uses of Data Cubing Techniques
  - Modularisation for Scale and Performance
  - Star Implementations and the Use of SQL data
- Understanding the Nuances of Indices and their Implications.
  - Value Index and Word Index - Which for What?
  - Deconstructing the "Minimal" Index
  - Value Index Entry Length and its Implications
  - Alternate Character Set (Language) Indexing
  - Index Optimisation
- Understanding FileMaker's Caching Systems
  - Theoretical Model for Caching Operations
  - Cache and Refresh Issues:
    — The Use and Abuse of Record Commit
  - Caching and Graph Complexity
  - Use of Control File Strategies for Cache Manipulation
- Advanced Control of Referential Integrity in FileMaker
  - Overview of basic Referential Integrity Management
    — Implications of Loss of Referential Integrity
    — Other Forms of Integrity Compromise
  - Procedural (eg Scripted/Automated) Data Cleansing
  - Methods for Dynamic Application of Integrity Constraints
- Local-Code/Remote-Data and WAN Optimisation
  - Option 1: Heavy Resources (eg graphics) are Local
  - Option 2: Interface/Application is Local
  - Option 3: Static Reference Data is Local
  - Option 4: Remote Data Segmentation

7. Advanced Development Techniques & Quality Assurance

- Value and Trade-Offs of Standards-Based Development
  - Benefits of Standards-Based Development
  - Standards vs Intra-Solution Consistency
  - The Trouble with Standards is:
    ... There Are So Many of Them!
  - Issues with Rigid Standards Frameworks
- Considering a Frameworks-Based Approach

- Guidelines and Calls To Action
- Principles Trump Practice
- Rationale Survives Version Changes, Prescription (Frequently) doesn't
- Frameworks Encourage Individual Judgement and Responsibility

– Effective Models for "Harmony" with FileMaker
- Getting the Balance Right - "How standard is standard enough?"
- "Standards" that Result in Forfeiture of Inherent Advantages of FileMaker
- Playing to FileMaker's Strengths vs Bending the Spoon

– The Anatomy of Error – Fault Tracing in FileMaker
- Tracing Solution Logic and the FileMaker DDR
- The Principle of Elimination - Halving
- Errors that Don't Occur when the Script Debugger is in Operation
- Script Logging and Error Reporting Techniques

– The Value of Stubs and Maquettes in Testing and Analysis
- Definition and Scope of Action of a Stub
- Stubs vs Stand-In Systems for Staged Development
- Sand-Pit Files as a Testing Mechanism
  —Replication and Isolation of Problems in a New, Clean File

– Quality Assurance in the Context of the FileMaker Development Life-Cycle
- Code Test vs Application Test vs User Acceptance Testing
- Validity and Importance of Review in the Development Cycle
- Options and Strategies for Q.A.

– Etiology* and Remediation of Database File Corruption
- Common Encounters with Corruption
  - Index Corruption and its Symptoms
  - Object and Image Corruption (Clipboard-related)
- Uses of the File Recovery Toolset
- Considerations Regarding Development on Hosted Files

** derived from the Greek αιτιολογία (aitiologia) - the study of causation, or origination*

– Innovative Deployment Models for the FileMaker Platform
- Uses of Launcher Files
- The 'Pregnant' Host (Data) File
- Satellite Files and Sync Updater Architectures
- Mixed Modal Multi-Component Interfaces

– Custom Environments for Solution Management
- The Use(s) of a Master Development Tool
- Viewer Layouts for Data Validation and Sandboxing
- Migration and Update Functionality
- Code Stubbing and Test Operations

8. Conclusion

– Review, Q&A, Discussion
- Question and Answer sessions
- Open Discussion
  - Alternative Perspectives and Experiences
  - Exploring nuances of topics covered
- Areas for further investigation

– Feedback, Evaluations and Follow-Up
- Appraisal of the Master Class Experience
- Materials and Notes for Attendees
- Looking to the Future
- Considering Support and Q.A. Arrangements
- Opportunities for Ongoing Dialog

– Closure / Thank you!