

To Dos From Chats

1) Meetings, Planning, and Team Coordination

To-Do	What you must do / deliver	When / Context
Attend the 4K meeting to review Caufero's "Suggested Improvements" PDF	Join the 90-minute review session and walk through the PDF with Joe and Oz (Kirk optional)	Meeting proposed for Jan 7–9, 2026 , likely 2–3 PM GMT
Participate in the "team meeting" before the 4K PDF review meeting	Be present for the broader team alignment meeting (tools, naming, task manager, workflow rules)	Joe hopes this happens before the 4K meeting
Join a brainstorming / whiteboard session about product variants and SKUs	Review Joe's current design and propose better TO-group and data model approaches	AZ Safety inventory refactor discussion

2) Server Access, Dev/Test/Prod Environment, and Virola Access Problems

To-Do	What you must do / deliver	When / Context
Log into the dev/test server and explore the 4K app safely	Use the server as a sandbox to learn tables, relationships, and structure	Server: azsafety.fmcloud.fm
Use the provided credentials and change password	Login as your first name (lowercase), password Caufero123 , then change it	Joe provided credentials
"Play" with the 4K copy to learn relationships and table structures	Explore tables, relationships, and TO groups enough to discuss improvement plan	Joe asked Cyril and Oz to do this
Help decide whether PBTS dev server or AZSafety server is sandbox/test	Evaluate pros/cons and align on one approach	Dev/test server strategy discussion
Learn what is installed on the dev server and what each tool is for	Confirm what each listed file is used for and where it helps the team	Joe's "Current State of Dev Server" post
Fix the Virola download problem for Joe	Help troubleshoot why Joe cannot download from Virola on two Mac minis even with permissions	Joe cannot access shared materials
Learn and apply Virola channel organization rules	Understand "Knowledge groups" vs "Discussion groups" and how to use them	Joe's channel organization guidance

To-Do	What you must do / deliver	When / Context
Use Virola Search tool to find posts across channels	Practice using Search so you can retrieve tips and links fast	Joe explicitly asked you to try it

3) Build Freeze, File Naming, and Release Control

To-Do	What you must do / deliver	When / Context
Learn the "build freeze authority" concept and how it will be enforced	Understand who calls freeze, what happens at freeze, and how work stops cleanly	Joe says PM must call freeze
Help define the file naming convention for dev/test/prod files	Decide between version numbers in filename vs suffixes like <u>_dev</u> and <u>_test</u> , and define the rule	Joe asked key question
Understand how naming ties into OttoFMS + OttoDeploy deployment	Learn which file name must match production and how dev/test names stay separate	Joe says naming still matters even with Otto
Learn the standard DEV → TEST → PROD flow	Learn: freeze after review + early tests, push build to TEST, test, then deploy to PROD	Kirk's Jan 2 workflow
Learn OttoDeploy rollback capability	Understand how rollback works, when it is used, and what the "prior version" means	Kirk notes quick rollback feature
Learn how the PM avoids deploying while other devs have risky WIP	Understand the "unfinished work can sit if not user accessible" strategy	Kirk's point about hidden WIP

4) Deployment Discipline and Multi-Developer Workflow Rules

To-Do	What you must do / deliver	When / Context
Study Kirk's 13-step manual migration order	Learn the exact sequence and why wrong order causes errors	Kirk says order is critical
Understand when the 13-step manual process applies	Learn it as the manual fallback and also for rolling live edits back into dev	Kirk clarified scope
Adopt the "Make changes → Review → Freeze → Build/Deploy" model	Be able to explain and follow this as team standard	Joe summarized it clearly
Understand the OTTO automated release sequence	Learn: pause prod, backup, clone dev, move, migrate data, rename to prod name, resume	OTTO Scheduled Transition list

To-Do	What you must do / deliver	When / Context
Apply "live code change" safety habits	Learn practices: copy scripts/layouts to edit in sandbox, copy accepted changes back, code review, documentation	Live Code Changes section
Learn the "script IDs matter, names do not" nuance	Understand why edits happen in original scripts and how backups restore content	Kirk's script ID explanation
Learn and adopt the backup/WIP duplication rules	Follow: duplicate for BACKUP, edit original, delete backup if validated, restore from backup if freeze is coming	Kirk's "Basic rules" list
Define a naming convention for BACKUP and WIP objects	Agree on suffix rules so the strategy stays consistent	Kirk says naming convention needed
Mirror production security changes back into dev	When users are added in production, replicate those changes in development too	Joe flagged this risk
Implement "Save As XML" change delta logging	Add a server-side scheduled script that generates timestamped Save As XML outputs	Kirk proposed it
Learn to use ModLog and similar tools on XML deltas	Use change logs so reviews and migration sequencing becomes clearer	Kirk referenced Geist tools
Use DDR as a safety snapshot before change sessions	Learn the DDR workflow and how FMPerception can restore scripts from DDR	Kirk suggested optional approach

5) Code Snippets, Reusable Code, and Shared Libraries

To-Do	What you must do / deliver	When / Context
Learn Raycast for FileMaker snippet capture	Install and understand how it captures FileMaker clipboard XML via AppleScript	Kirk recommended it
Study FmClipTools	Review the GitHub project and understand what it enables (clipboard XML convert/modify/save)	Link shared
Study the Raycast FileMaker Snippets setup guide	Learn how Raycast uses a GitHub repo for sharing snippets	Link shared
Help set up the shared GitHub repository for team snippets	Participate in repo structure decisions, how snippets are organized, and how team consumes them	Kirk asked "who will take it on"

To-Do	What you must do / deliver	When / Context
Review CodeSnips database (MonkeyBread) on the dev server	Explore how it stores and retrieves snippet blocks	Listed in server tools
Learn the "modular code database" concept on the dev server	Understand how reusable modules (bigger than snippets) are cataloged	Kirk described it

6) Documentation Review and Missing Visuals

To-Do	What you must do / deliver	When / Context
Read the coding guidelines document Kirk sent	Review it carefully and prepare questions/comments/changes	Kirk asked you to scan it
Create or help create a "quick start" TO Groups doc with visuals	Provide screenshots/examples that reinforce TO group concepts	Joe said the PDF lacked useful pictures

7) Inventory System Design Study and Refactor Support

To-Do	What you must do / deliver	When / Context
Learn the two SKU numbering schools of thought	Compare "ID + serialized variant" vs "metadata encoded in SKU digits"	Kirk's inventory notes
Study VIN decoding as a SKU analogy	Use VIN structure as a mental model for metadata-based SKUs	Link shared in chat
Learn BoM inventory model requirements	Understand assemblies, sub-assemblies, and atomic decomposition	Kirk's notes
Learn formula-based inventory requirements	Understand irreversibility after mixing and why component batch tracking matters	Kirk's notes
Learn batch / lot tracking rationale	Understand recall risk and why tracing component batches matters	Tylenol example idea
Learn item serialization requirements	Understand serial tracking for high value items and assemblies	Kirk's notes
Learn RMA and return-to-stock design requirements	Model "cannot resell as new" cases and inventory designations/states	Kirk's notes
Help Joe refactor Product Group / Product SKU TO groups in AZ Safety	Review his current design and suggest better TO group architecture for variant matrix	Joe asked for help

To-Do	What you must do / deliver	When / Context
Learn variant matrix modeling (one product → many SKUs)	Understand how to represent size/color and deep category hierarchies cleanly	Joe's T-shirt example
Explore BCS_d1 SKU Builder layout on the dev server	Learn categories, tree nodes, nested loops approach to SKU variation sets	Joe listed it as a key demo
Study "Item Collection" demo patterns	Learn Collection/Parent/Child/Sort with breadcrumbs and tree navigation UI	Listed in server tools

8) Mobile Strategy and Offline-First Architecture

To-Do	What you must do / deliver	When / Context
Move mobile strategy discussion to the Mobile channel	Continue the conversation in the dedicated mobile chat room	Kirk requested it
Think through mobile data architecture (mobile vs back office)	Define what data lives on device, what stays server-side, and how they sync	Kirk's Jan 30 note
Design the upgrade path for an offline mobile app	Define how mobile app upgrades keep local data intact	Kirk flagged upgrade retention challenge
Evaluate mobile contenders and write a recommendation	Compare: FM Go, WebDirect, Studio, Better Forms, custom native app	Joe listed contenders
Decide what "untethered" means for your use case	Define which functions must work with zero server and which can be optional	Kirk: untethered needs 100% for required functions
Assess iOS device feature access by platform choice	Confirm camera, barcode, signatures, geolocation support assumptions	Kirk warned web apps lose sensor access
Investigate Better Forms offline + sync claims	Validate what Better Forms can really do for offline, sync, and iOS feature access	Joe mentioned AI claims
Consider MirrorSync capabilities	Check if MirrorSync can avoid the separation model for code/data	Kirk mentioned it

9) Integration Learning: OData, APIs, Webhooks, and SimpleQ

To-Do	What you must do / deliver	When / Context
Learn what OData is and why it is useful	Understand query precision: select only needed fields, filter, sort at server	Joe's OData explanation

To-Do	What you must do / deliver	When / Context
Watch John Renfrew's OData lesson series (5 videos)	Watch the first linked video, then complete all five lessons	Joe recommended all five
Watch the Claris Engage 2025 OData talk (first video link shared)	View it and extract key takeaways relevant to your FileMaker work	Joe shared link
Review the OData Examples database on the dev server	Study the 45 OData examples including syntax, URLs, cURL and demo data	Server tool list
Learn API basics using the shared reference	Read the API explanation resource Joe linked	Postman "what is an API" link
Learn webhook basics using the shared reference	Read the webhook guide Joe linked	usewebhook.com link
Learn API vs webhook difference using the shared reference	Study pull vs push and event-driven payload delivery	Zapier link + chat explanation
Learn SimpleQ's role in your stack	Understand SimpleQ as message broker and that it uses webhooks as payload carriers	Kirk + Joe confirmation
Explore SimpleQ demo files on the dev server	Log in (admin/admin as stated) and understand Publisher/Subscriber pattern	Server tool list
Use FindJSONPath tool for JSON path testing	Practice extracting fields via JSON paths	Link shared by Kirk
Study JSONQuery 2.1 custom function	Learn querying/transforming JSON using that CF and where it helps	Server tool list

10) UI, UX, Layout Standards, and Design Resources

To-Do	What you must do / deliver	When / Context
Learn Button Bars vs Buttons and adopt the standard	Study why button bars are preferred and how to design with them consistently	Kirk + Oz + Joe discussion
Apply button bar segment labels as calculated UI elements	Practice using calculated segment labels for dynamic data, uniform spacing, icons, hover help	Kirk's explanation
Review Apple Human Interface Guidelines	Read and extract UI rules that apply to your mobile and desktop layouts	Link shared by Joe
Study FM Design University workflow design framework	Learn UX workflow design thinking and how it applies to FileMaker screens	Link shared by Joe
Use SVG icon repositories for UI work	Explore SVGRepo and Proof+Geist FM SVG library for layout icons	Links shared

11) FileMaker Containers: Learn, Verify, and Update Knowledge

To-Do	What you must do / deliver	When / Context
Review the "Container Fields Under the Hood" infographic	Understand how containers store files, server backup implications, and performance risks	Kirk shared it
Confirm secure storage encryption details in modern FileMaker	Follow up after Kirk "finds out" the new security details	Cyril asked, Kirk will check
Confirm media streaming behavior in modern FileMaker	Record that streaming is now HTTPS on port 443, and understand URL guessing risk	Kirk responded with updates
Update your folder structure best practices understanding	Note: duplicate filenames issue resolved, directory structure consolidated, review folder-size performance risk	Kirk responded
Investigate Windows folder performance issue relevance today	Check if the "512+ files per directory slowdown" still matters on modern Windows	Kirk unsure, flagged it

12) Virola Usage Learning

To-Do	What you must do / deliver	When / Context
Watch the official Virola videos	Use the Virola Messenger YouTube channel and watch the official videos	Kirk shared channel link
Apply the iPhone UI workaround	If Virola controls are cut off in portrait mode, switch to landscape	Kirk's tip

13) FileMaker 26.1 (ETS) Feature Tracking and Study

To-Do	What you must do / deliver	When / Context
Learn "Select Window by UUID" + <code>Get(WindowUUID)</code>	Understand what problems this solves and how you would use it	Kirk shared ETS note
Learn enhanced <code>GetRecordIDsFromFoundSet()</code> optionalName parameter	Understand specifying TO or portal name and what types 0–4 mean	Kirk shared ETS note
Study the new "Execute URL" script step (async + callback script)	Learn the async paradigm, callback handling, and supported cURL options matching Insert from URL	Kirk shared ETS note

To-Do	What you must do / deliver	When / Context
Track the dev community feedback on inspector palette changes	Understand the change (4 to 2) and implications on styles/workflow	Kirk noted major pushback
Capture the remaining "Execute URL" details from the original source	Your pasted log cuts off mid-sentence, so you need to pull the rest from the chat and save it	Your transcript ends mid-line

Quick checklist of the dev server apps Joe explicitly wants you to explore

App / Tool	What to do with it
4K Painting & Drywall	Learn structure and relationships
OttoDeploy	Learn builds, deployments, rollback
BCS_d1 (SKU Builder layout)	Study SKU variants via categories/tree nodes/nested loops
barcode_test	Learn barcode reading via iOS camera / container
BCS_StyleSheetTheme	Study standard theme/style naming patterns
ActionTrack_R1	Learn issue/feature tracking approach
CodeSnips	Learn snippet storage via MonkeyBread approach
Item Collection	Learn parent/child collections + breadcrumbs/tree UI patterns
InfoTools Helper	Learn Kirk's constraints/constructs philosophy + setup notes
JSONQuery 2.1	Learn JSON query/transform helper CF usage
Time_Picker	Review it and note the WebDirect limitation + look for the newer version Kirk mentioned
TaskControl	Learn multi-project, multi-user task management model
OData Examples	Work through example patterns
SimpleQ	Learn the broker + demos