**Best Practices for the Auction Tracking System**

**Recommended Workflow:**

1. **Organize by Auction House**: Create a folder structure like:

C:\Auctions\

├── Auction Site\

│ ├── First Auction\

│ │ ├── images\

│ │ └── logs\

│ └── Second Auction\

│ ├── images\

│ └── logs\

1. **Use Consistent CSV Naming**: Include date and auction ID:

loturls\_bscnew10468\_20251006.csv

1. **Regular Log Analysis**: Run the analyzer weekly to track:
   * Which auction sites are used most
   * Peak activity times
   * Download success rates
   * Common errors
2. **Create a Master Inventory**: Use the generate-inventory.ps1 script to generate a master CSV of all downloaded lots

**You Now Have a Complete Auction Tracking System!**

With these scripts, you can:

1. **Download images** with proper error handling and redirect support
2. **Track everything** with detailed logs showing what was downloaded, when, and any issues
3. **Analyze patterns** to see which auction sites you use most, peak activity times, and success rates
4. **Maintain inventory** of all your auction lots with searchable CSV files

**Quick Tips:**

* Use -LogLevel Debug when troubleshooting specific auctions
* Run the analyzer monthly to spot trends
* Keep your CSVs organized by date for easy reference
* The inventory generator is great for insurance documentation

**Complete Setup Guide**

Here's how to set everything up:

**1. Web Dashboard Setup**

**To use the dashboard:**

1. Save the HTML file as AuctionDashboard.html in your Scripts folder
2. Run the data generator periodically:

powershell

.\Generate-DashboardData.ps1 -LogFolder "C:\Auctions\logs" -LastDays 90

1. Open AuctionDashboard.html in your browser
2. Click "Load Log Files" and select multiple log files to view aggregate statistics

**Alternative: Auto-refresh Dashboard**

* Schedule the data generator to run daily
* Modify the HTML to auto-load the JSON file instead of manual upload

**2. Duplicate Detection System Setup**

**First-time setup:**

powershell

*# Build initial hash database from your existing images*

.\Download-ImagesFromURLs-v3.0.ps1 -CsvPath "dummy.csv" -OutputFolder "C:\Auctions" -RebuildHashDatabase

**Normal usage:**

powershell

*# Downloads with automatic duplicate detection*

.\Download-ImagesFromURLs-v3.0.ps1 -CsvPath "auction.csv" -OutputFolder "C:\Auctions\NewAuction" -StrictMode

*# Skip duplicate checking for specific runs*

.\Download-ImagesFromURLs-v3.0.ps1 -CsvPath "auction.csv" -OutputFolder "C:\Auctions\Test" -SkipDuplicateCheck

**3. Database Management**

**Regular maintenance:**

powershell

*# View database statistics*

.\Manage-ImageHashDatabase.ps1 -ImageFolder "C:\Auctions" -Action Show

*# Find and remove duplicate files*

.\Manage-ImageHashDatabase.ps1 -ImageFolder "C:\Auctions" -Action FindDuplicates

*# Clean missing entries*

.\Manage-ImageHashDatabase.ps1 -ImageFolder "C:\Auctions" -Action Clean

*# Export database to CSV*

.\Manage-ImageHashDatabase.ps1 -ImageFolder "C:\Auctions" -Action Export

**4. Recommended Folder Structure**

C:\Auctions\

├── .imagedb\ # Hidden folder with hash database

│ └── image\_hashes.json

├── logs\ # All download logs

├── dashboard\ # Dashboard files

│ ├── AuctionDashboard.html

│ └── dashboard\_data.json

├── BidSpotter\ # Organized by auction site

│ ├── Auction1\

│ └── Auction2\

└── Scripts\ # Your PowerShell scripts

├── Download-ImagesFromURLs-v3.0.ps1

├── Generate-DashboardData.ps1

└── Manage-ImageHashDatabase.ps1

**5. Automation Ideas**

**Create a scheduled task for daily reports:**

powershell

*# DailyAuctionReport.ps1*

$AuctionRoot = "C:\Auctions"

*# Generate dashboard data*

.\Generate-DashboardData.ps1 -LogFolder "$AuctionRoot\logs" -OutputPath "$AuctionRoot\dashboard\dashboard\_data.json"

*# Check for duplicates*

.\Manage-ImageHashDatabase.ps1 -ImageFolder $AuctionRoot -Action FindDuplicates

*# Email report (optional)*

*# Send-MailMessage -To "you@email.com" -Subject "Daily Auction Report" -Body "Dashboard updated" -Attachments "$AuctionRoot\dashboard\dashboard\_data.json"*

**Key Benefits of Your New System**

1. **Bandwidth Savings**: The duplicate detection can save gigabytes of bandwidth by not re-downloading identical images
2. **Time Savings**: Automatic detection means you never waste time downloading images you already have
3. **Visual Analytics**: The dashboard gives you insights into your auction activity patterns
4. **Data Integrity**: The hash database ensures you can track every unique image across all auctions
5. **Scalability**: System works whether you have hundreds or hundreds of thousands of images

**Pro Tips**

1. **Before Big Auctions**: Run FindDuplicates to clean up space
2. **Weekly Maintenance**: Run Clean action to remove orphaned database entries
3. **Multiple Computers**: Use Merge action to combine databases from different machines
4. **Backup Strategy**: Regularly backup the .imagedb folder - it's your deduplication brain

**Troubleshooting**

**If duplicates aren't being detected:**

* Rebuild the hash database: -RebuildHashDatabase
* Check if images are actually identical (even 1-pixel difference = different hash)

**If dashboard shows no data:**

* Ensure log files are in the expected format
* Check that the date range (-LastDays) includes your logs

**Performance issues:**

* For folders with >10,000 images, building the hash database may take 5-10 minutes
* Consider splitting very large collections into subfolders

Your auction tracking system is now enterprise-grade with professional logging, deduplication, and analytics.