

Buy a Suppressor | Scope Vendor integration Quote

Website: <https://buyasuppressor.com/>

Project Review & Scope

We have evaluated the requirements and explored available approaches for integrating RSR data and automated workflows into WooCommerce. Below are two primary options:

- **Option A:** Plugin-based integration (with optional customization for automated order submission)
- **Option B:** Fully custom integration (FTP or API)

Option A is more cost-effective and significantly faster to implement. However, Option B offers full customization and real-time data management if required at a later stage.

Option A - Plugin Option

We've found a [plugin](#) for **RSR Group** that appears to offer the functionality we need. **However, since we haven't used this plugin before and there's no demo link available, we'll need to contact the plugin's support team to understand the setup process and how it works.**

1. PLUGIN OVERVIEW

Data Feed by TTG for WooCommerce is a specialized plugin developed by **Texas Technology Group (TTG)** specifically for firearms dealers using WooCommerce. Its primary goal is to automate the import and syncing of real-time firearms inventory from major distributors, including RSR Group, directly into a WooCommerce store, eliminating the need for manual CSV uploads or custom coding.

Core Purpose:

The plugin imports distributor data such as products, inventory, prices, and images into

WooCommerce automatically. Designed "for gun dealers, by gun dealers," it ensures compliance with federal regulations (e.g., FFL/SOT flags) and is capable of handling large product catalogs, often exceeding 100,000 SKUs.

RSR Compatibility:

RSR Group is fully supported as one of TTG's primary feeds. No custom setup is required for RSR; the plugin pulls inventory, stock levels, prices, and images directly from RSR via TTG's API, providing seamless integration.

Plugin Philosophy:

The plugin prioritizes automation over manual intervention, bridging the RSR DirectConnect ecosystem (FTP/API data as described in the RSR manual) with WooCommerce. It allows dealers to manage large inventories efficiently, with features like hourly stock updates, compliance checks, and automated image syncing.

2. FUNCTIONALITY PROVIDED BY RSR GROUP

The plugin leverages RSR's data feeds (e.g., [rsrinventory-new.txt](#), [IM-QTY-CSV.csv](#)) but processes them through TTG's Data Feed API for WooCommerce. This eliminates the need for direct FTP or API coding. The plugin is effectively "plug-and-play" for RSR, providing the following key features:

Bulk Product Import

- **Functionality:** Imports the full RSR catalog, including firearms, accessories, and ammunition. Filters can be applied by brand, category, price, SKU, or stock. Supports catalogs with 50,000+ SKUs.
- **RSR Data Tie-In:** Uses [rsrinventory-new.txt](#) (77 fields including SKU, Description, DealerPrice) and [IM-NEW-SKUS.csv](#) for new items.
- **WooCommerce Impact:** Automatically creates or updates products, mapping fields like title, SKU, price, and categories. Includes a one-click cleanup option for imported items.

Real-Time Inventory Sync

- **Functionality:** Automatically updates stock levels hourly, hiding out-of-stock items to prevent overselling.
- **RSR Data Tie-In:** Pulls data from [IM-QTY-CSV.csv](#), which updates every five minutes.
- **WooCommerce Impact:** Stock levels remain accurate in real-time; integrates with FFL Checkout to ensure compliance during firearm sales.

Dynamic Pricing

- **Functionality:** Automatically applies fixed or percentage markup to RSR dealer prices while respecting MAP policies.
- **RSR Data Tie-In:** Maps Retail MAP Price (Field #71) and Dealer Price.
- **WooCommerce Impact:** Regular and sale prices are updated automatically; violations of MAP pricing are flagged.

Image Sync

- **Functionality:** Downloads product images and hosts them via TTG's built-in CDN for fast loading and SEO optimization.
- **RSR Data Tie-In:** Pulls from the [/images/](#) folder in RSR feeds using the Image Name field (#15).
- **WooCommerce Impact:** Images are attached to the media library and automatically optimized for mobile and desktop.

State & Compliance Restrictions

- **Functionality:** Filters or excludes products based on state/county restrictions and flags FFL/SOT-required items.

- **RSR Data Tie-In:** Parses state restriction fields (16–56) and CA Prop 65 flag (Field #76).
- **WooCommerce Impact:** Checkout is blocked for restricted states; integrates with FFL Checkout to enable transfer dealer selection.

Vendor Approval Handling

- **Functionality:** Automatically holds or flags items that require vendor approval.
- **RSR Data Tie-In:** Checks Vendor Approval Required flag (Field #77).
- **WooCommerce Impact:** Items are added to an admin “Pending Review” queue for manual approval.

Deleted/New Item Management

- **Functionality:** Automatically removes deleted items and adds new ones to WooCommerce.
- **RSR Data Tie-In:** Monitors `rsrdeletedinv.txt` (2-week retention) and `IM-NEW-SKUS.csv`.
- **WooCommerce Impact:** Deleted products are trashed; new items are published automatically.

Marketplace Feeds

- **Functionality:** Generates XML or API feeds for marketplaces like Gun.deals, AmmoSeek, and others.
- **RSR Data Tie-In:** Exports products synced from RSR in compliant formats.
- **WooCommerce Impact:** External listings are automated; no additional certification or manual effort is needed.

Automation Level:

The plugin operates with high automation through WP-Cron background jobs and maintains logs for any failed syncs. After setup, no manual intervention is required for inventory and pricing updates.

Setup Steps and Time Estimates:

1. Install & Activate

- **Description:** Download the plugin from WP.org, upload it to WooCommerce, activate, and complete basic configuration.
- **Notes:** Free installation; one-time setup fee applies after trial.

2. Enter RSR Credentials

- **Description:** Input RSR Account Number, Username, and Password (from RSR Dealer Toolbox).
- **Notes:** TTG API handles authentication; connection can be tested via the plugin UI.

3. Configure Feed Filters

- **Description:** Select the RSR feed, apply brand, price, or category filters, and map products to WooCommerce categories.
- **Notes:** The UI is user-friendly with previews; video tutorials are available.

4. Set Pricing & Compliance

- **Description:** Define markup rules, enable state restrictions, and integrate FFL Checkout for compliance.
- **Notes:** MAP rules are automatically respected; recommend testing sync on 100 SKUs initially.

5. Run Initial Sync & Test

- **Description:** Trigger full import of products, stock, and images; configure hourly WP-Cron sync.
- **Notes:** Initial sync may take a few hours for 50,000+ SKUs; subsequent syncs run hourly in the background.

Requirment

- We will require [Gun Data Feed](#) Plugin for this project.

Estimated Time & Hours (Plugin Integration): 16-18 hours

Limitations for Plugin: While the plugin fully automates product import and stock syncing, it does not place orders to RSR automatically. For automated order submission, integration with RSR's API (as per the custom plugin) is necessary. Custom RSR-related features are available upon request.

Solution: Plugin with customization to place orders

If you choose **Option A** and purchase the plugin, then we will create a **custom plugin** to automatically submit orders to RSR using their API. The estimated time and effort required for developing this custom plugin for place orders automatically.

Deliverables:

- Auto Submit Order
- End Consumer Name
- Serial Number Sync
- Order Status Sync
- FFL Dealer Lookup

- Vendor Approval Hold
 - Error Handling
 - Admin UI
 - Cron Sync
-

Estimated Time & Hours(Custom Plugin) The estimated time for **development and testing** is 25-30 hours. (Only plugin development)

Total Hours for Option A + Custom plugin for order synchronization: 36 - 48 Hours

Option B - Custom Integration

1. FTP INTEGRATION

Overview

RSR provides **batch data files** via secure FTP for periodic synchronization. This method is best suited for **nightly updates**, **backup feeds**, or sites where inventory changes are not frequent. We will create a custom plugin to implement this functionality.

Scope of work

- **Full Product Catalog (rsrinventory-new.txt)** — complete inventory list with 77 fields, updated every 2 hours.
- **Drop-Ship Catalog (fulfillment-inv-new.txt)** — products available for drop-shipping, updated every 2 hours.
- **Real-Time Quantity (IM-QTY-CSV.csv)** — stock levels updated every 5 minutes.
- **New/Deleted Items (IM-NEW-SKUS.csv / rsrdeletedinv.txt)** — new SKUs added in the last 4 weeks and products deleted within 2 weeks.

- **Product Images (/images/)** — high- and low-resolution images for all products.

Format: Semicolon-delimited ASCII text (.txt or .csv).

Deliverables:

The FTP plugin provides multiple automated functionalities to manage product data efficiently:

- 1. Automated FTP Pull**

The plugin establishes a secure SSL connection to the RSR FTP server and downloads the necessary files automatically. This eliminates manual downloading, ensuring inventory is regularly updated without intervention.

- 2. Delta Import**

Instead of importing the entire catalog every time, the plugin detects which SKUs have changed and updates only those. This improves performance, reduces server load, and speeds up synchronization.

- 3. Image Auto-Download**

Product images are automatically downloaded from the /images/ folder. Each image is matched with the corresponding product based on the “Image Name” field and attached to the product gallery in WooCommerce.

- 4. State Restriction Engine**

Certain products may have sales restrictions by state (fields 16–56 in the feed). The plugin reads these fields and automatically blocks purchases in restricted regions, ensuring compliance.

- 5. MAP Price Enforcement**

Minimum Advertised Price (MAP) restrictions are enforced by hiding product prices if the price is below the threshold set by RSR.

- 6. Vendor Approval Filter**

Items flagged with “Y” or “F” by the vendor are automatically held in WooCommerce until approved, reducing the risk of selling restricted products unintentionally.

7. Deleted Item Purge

Products listed in `rsrdeletedinv.txt` are automatically removed from WooCommerce to prevent selling discontinued or deleted SKUs.

8. Error Logging

All import failures or issues are logged, and administrators receive notices in the backend. This allows for quick troubleshooting and ensures data integrity.

Estimated Time & Hours

- The estimated time for **development and testing** is **75-80 hours**.

Limitations / Considerations

FTP integration is **not suitable for real-time stock updates**. For live inventory, the **API integration** is recommended.

While FTP integration is highly useful, it has certain limitations:

1. No Real-Time Quantity

- Stock levels may not reflect immediate changes.
- If live inventory tracking is critical, the **API integration** is recommended.

2. Large File Sizes (~200MB)

- Downloading or importing very large files can cause memory or timeout issues.
- Mitigation: Use chunked imports or batch processing to prevent server overload.

3. FTP Downtime (3–5 AM)

- If the RSR FTP server is offline during scheduled syncs, imports may fail.

- Mitigation: Schedule syncs after 6 AM or implement retry logic.

4. No Push Updates

- FTP requires periodic polling; changes are not pushed automatically.
- Mitigation: Use **WP-Cron** to automate regular syncs.

5. Semicolon Conflicts in Data

- If product data contains semicolons, parsing errors may occur.
- Mitigation: Strict `fgetcsv(..., ';')` parsing ensures accurate field separation.

Conclusion

FTP integration is ideal for:

- **Nightly bulk updates**
- **Offline data processing**
- **Backup inventory feeds**

Not suitable for live stock updates — for real-time inventory and automated order management, use the **RSR API integration**.

2. API INTEGRATION

Overview:

The RSR **REST JSON API** allows WooCommerce stores to communicate with RSR in real-time. Unlike FTP, which requires periodic polling, the API provides instant access to critical data and

actions, enabling fully automated operations. We will create a **custom plugin** to implement this functionality.

Scope of Work

- **Live Inventory** — Query current stock for each SKU in real-time, ensuring accurate availability for customers.
- **Order Submission** — Submit orders automatically from WooCommerce to RSR without manual entry.
- **Order Status Tracking** — Monitor shipments, backorders, and delivery progress directly from WooCommerce.
- **Serial Number Retrieval** — Retrieve serial numbers for regulated items automatically and attach them to the order.

Deliverables:

1. Live Stock Check

- Queries the [/get-items](#) endpoint before adding products to the cart.
- Prevents purchase if stock is zero, displaying a user-friendly message.
- Helps avoid overselling and ensures inventory accuracy in real-time.

2. Auto Order Submission

- Sends WooCommerce orders automatically to RSR via [/place-order](#).
- Eliminates manual order entry, reducing human errors and processing time.
- Supports full order metadata including quantities, shipping info, and product SKUs.

3. Serial Number Import

- Automatically retrieves serial numbers using [/check-order](#).
- Adds serial numbers to WooCommerce order notes for compliance and tracking.
- Essential for regulated items like firearms and serialized products.

4. FFL Validation

- Checks [/get-transfer-dealers](#) to verify FFL locations.
- Customers can select a valid dealer during checkout.
- Ensures legal compliance and reduces order rejection risk.

5. Error Mapping & Notifications

- Handles API errors such as invalid SKUs, insufficient stock, or restricted items.
- Converts raw API error codes into **clear, actionable messages** for admins or customers.
- Provides an admin-friendly log for troubleshooting.

6. Caching Layer

- Frequently accessed endpoints, like [/get-items](#), are cached for 5 minutes using WooCommerce transients.
- Reduces repeated API calls, improves performance, and helps avoid throttling.

7. Webhook Fallback / Polling

- Since RSR does not provide webhooks, WP-Cron is used to poll [/check-order](#) every 15 minutes.
- Ensures order status and serial numbers are updated automatically without manual intervention.

8. Admin Interface for API Management

- Settings page in WooCommerce allows the admin to manage API credentials, caching options, and polling frequency.
 - Logs errors and updates to provide transparency and easier troubleshooting.
-

The API integration is developed in **structured phases** to ensure robustness, reliability, and compliance:

1. API Review:

- Study RSR API documentation and sample requests/responses.
- Understand authentication, endpoints, and payload requirements.

2. Authentication Setup:

- Configure **X-API-Key** and other credentials securely.
- Test API connectivity via Postman or similar tools.

3. Client Class Development:

- Create PHP class for API communication.
- Handle GET/POST requests, errors, retries, and logging.

4. Stock Sync:

- Fetch live stock from **/get-items** and update WooCommerce inventory in real-time.
- Handle edge cases such as zero stock, partial stock, or negative values.

5. Order Mapping:

- Map WooCommerce order fields to RSR `/place-order` payload.
- Include shipping info, quantities, metadata, and special flags for compliance.

6. FFL Integration:

- Implement `/get-transfer-dealers` endpoint.
- Add FFL dropdown selection at checkout.
- Store selected FFL in order metadata.

7. Status Polling:

- Poll `/check-order` every 15 minutes using WP-Cron.
- Import serial numbers and update order statuses automatically.

8. Caching Layer:

- Implement caching for `/get-items` and `/get-transfer-dealers` using transients.
- Reduces API load and prevents throttling.

9. Admin UI:

- Provide settings page, error logs, and API monitoring tools.
- Allows non-technical admins to manage integration safely.

10. Testing:

- Test 500+ orders including restricted SKUs, partial shipments, and serialized items.

- Validate stock accuracy, order submission, FFL compliance, and serial number import.

11. Documentation:

- Create README, setup instructions, and inline code comments for future maintenance.
-

Estimated Time & Hours

- The estimated time for **development and testing** is **90-95 hours**.

Limitations / Considerations:

1. Rate Limits Unknown

- High-frequency API requests may trigger throttling.
- Mitigation: Cache frequently accessed endpoints and implement retries for failed requests.

2. Firearm Name / Storename Required

- Certain fields (e.g., **Storename**) are mandatory for firearms orders.
- Mitigation: Add checkout fields or map WooCommerce store name automatically.

3. Vendor Approval Needed

- Some SKUs may require manual review before orders can be processed.
- Mitigation: Orders flagged by vendors are held in an admin queue until approved.

4. JSON Only, No XML

- API communicates strictly via JSON.
- Mitigation: Plugin parses JSON automatically; XML is not supported.

5. Dependency on API Availability

- If RSR API is temporarily unavailable, stock checks or order submissions may fail.
- Mitigation: Plugin logs failures and retries automatically.

The RSR API integration is ideal for WooCommerce stores that require:

- **Live inventory tracking**
- **Automated order submission**
- **Serial number management**
- **FFL and compliance verification**

Primary method for WooCommerce due to real-time accuracy, automated workflows, and reduced manual intervention compared to FTP polling.

3. EDI INTEGRATION

Big-business electronic data exchange (like Walmart uses)

EDI is not recommended for small WooCommerce stores because it is complex, costly to maintain, and not well-suited for WordPress + WooCommerce environments. EDI systems are typically designed for large enterprise platforms that have dedicated infrastructure, higher transaction volumes, and specialized support teams. For WooCommerce, this approach often creates unnecessary technical overhead and integration challenges. Additionally, implementation of this option is out of our wheelhouse.

Estimated Time & Hours

- The estimated time for **development and testing**

Option A

- Purchased Plugin + Custom plugin(Order sync): 36-48 **hours**

Option B

- FTP integration: 75 - 80 **hours**

- API integration: 90 -95 **hours**

Conclusion: Option A is more cost-effective and significantly faster to implement. However, Option B offers full customization and real-time data management if required at a later stage.

Notes:

- This timeframe may increase or decrease depending on how quickly we receive responses to queries and confirmations.
- Any changes or additional feedback may impact the estimated hours.
- If we encounter real-time issues while working with the resources provided by RSR, we may need to resolve them immediately, which could require additional hours.

