



GS523 – Power Platform for Compliance Workflows in Globalization Studio (Revised)

As businesses scale and compliance obligations grow more complex, automation becomes critical, not just for tax and document submission, but for what happens *after*.

Globalization Studio in Dynamics 365 provides the core execution engine for tax, document, and electronic invoicing features, but what if you could add dynamic reactions to those outcomes?

That's where **Microsoft Power Platform** becomes your compliance co-pilot.

What This Article Covers

In this article, you'll learn how to:

- Leverage **existing data events** and submission logs for compliance monitoring
- Use **custom alerts and business rules** to notify the right people
- Tap into **existing and virtual data entities**, with no custom code
- Design an architected, modular approach, even without a live demo

⚠ **Note:** Configuration screenshots are not included as this guidance is based on a theoretical approach. The concepts explained here are derived from real implementations, but not demonstrated in the current demo environment.

Why Combine Power Platform with Globalization Studio?

Think of it this way:

🧠 **Globalization Studio** → Handles logic and submission

🤖 **Power Platform** → Reacts, escalates, and communicates

This combination unlocks:

- End-to-end **compliance visibility**
 - **Real-time user alerts** and case escalation
 - Integration with **Teams, Outlook, DevOps, SharePoint**, and more
-



1. Use Existing Data Events & Virtual Entities for Submission Monitoring

Dynamics 365 already produces structured records for:

- Submission logs
- Error messages
- Processing results

These logs can be:

- Read through

Dataverse virtual Entities

Virtual entity metadata sync status AX : Operations					
Standard view *					
Filter					
Physical name	Description	Failure message	Failure retr...	Refresh ne...	Last refresh time
ElectronicMessageItemsEntity	Electronic messag...		0		
ElectronicMessageItemsLogDetailsEntity	Electronic messag...		0		
ElectronicMessageItemsLogEntity	Electronic messag...		0		
ElectronicMessageItemsLogV2Entity	Electronic messag...		0		
ElectronicMessageItemStatusEntity	Electronic messag...		0		
ElectronicMessageProcessingEntity	Electronic messag...		0		
ElectronicMessagesEntity	Electronic messag...		0		
ElectronicMessagesLogEntity	Electronic messag...		0		
ElectronicMessageStatusEntity	Electronic messag...		0		

Entity Name	Description	Typical Use Case	Power Platform Benefit
ElectronicMessageItemsEntity	Represents individual message items (e.g., one invoice, one file).	Track submission status of each invoice or document.	Trigger flows for failed items, push Teams/email alerts per document.
ElectronicMessageItemsLogDetailsEntity	Detailed technical log	Investigate submission errors, failed	Provide error detail context in Power



Entity Name	Description	Typical Use Case	Power Platform Benefit
ElectronicMessageItemsLogEntity	for each message item.	payloads, or integration issues.	Automate alerts or dashboards.
	Summary log for message items.	Monitor submission summary per item, including timestamps and outcomes.	Aggregate status history into Power BI for trend analysis.
ElectronicMessageItemsLogV2Entity	Enhanced version of message item log with extended schema.	Use in newer ER versions for more granular logging and retry logic.	Compatible with newer solutions; enriches compliance audit reports.
ElectronicMessageItemStatusEntity	Tracks individual item statuses (e.g., Created, Processing, Failed, Sent).	Visualize document life cycle through each processing step.	Create workflow rules based on item status changes.
ElectronicMessageProcessingEntity	Captures processing step results (e.g., Sign, Store, Submit).	Audit which pipeline steps succeeded or failed for each message.	Identify automation failure points and inform next steps.
ElectronicMessagesEntity	Header or parent message grouping related items together.	Manage multi-document batches (e.g., invoice group submission).	Initiate Power Automate flows per batch header for group processing.
ElectronicMessagesLogEntity	Log data at the full message (batch) level.	Track message group submission outcomes, response logs.	Correlate batch-level success/failure and integrate with issue trackers.



Entity Name	Description	Typical Use Case	Power Platform Benefit
ElectronicMessageStatusEntity	Tracks the overall status of a full message group.	Determine if a full document set was processed successfully.	Condition Power Automate flows to proceed or alert based on group status.

Data Events

Manage Security Options		
Business events		
Business event catalogue Data event catalogue Endpoints Active business events Inactive business events Active data events Inactive data events Errors Security		
+ Activate		
Category	Business event ID	Name
Electronic messages	DataEvent_ElectronicMess...	Electronic message items (ElectronicMessageItemsEntity) created
Electronic messages	DataEvent_ElectronicMess...	Electronic message items (ElectronicMessageItemsEntity) deleted
Electronic messages	DataEvent_ElectronicMess...	Electronic message items (ElectronicMessageItemsEntity) updated
Electronic messages	DataEvent_ElectronicMess...	Electronic message items log details (ElectronicMessageItemsLogDetailsEntity) created
Electronic messages	DataEvent_ElectronicMess...	Electronic message items log details (ElectronicMessageItemsLogDetailsEntity) deleted
Electronic messages	DataEvent_ElectronicMess...	Electronic message items log details (ElectronicMessageItemsLogDetailsEntity) updated
Electronic messages	DataEvent_ElectronicMess...	Electronic message items log (ElectronicMessageItemsLogEntity) created
Electronic messages	DataEvent_ElectronicMess...	Electronic message items log (ElectronicMessageItemsLogEntity) deleted
Electronic messages	DataEvent_ElectronicMess...	Electronic message items log (ElectronicMessageItemsLogEntity) updated
Electronic messages	DataEvent_ElectronicMess...	Electronic message items log V2 (ElectronicMessageItemsLogV2Entity) created
Electronic messages	DataEvent_ElectronicMess...	Electronic message items log V2 (ElectronicMessageItemsLogV2Entity) deleted
Electronic messages	DataEvent_ElectronicMess...	Electronic message items log V2 (ElectronicMessageItemsLogV2Entity) updated
Electronic messages	DataEvent_ElectronicMess...	Electronic message item statuses (ElectronicMessageItemStatusEntity) created
Electronic messages	DataEvent_ElectronicMess...	Electronic message item statuses (ElectronicMessageItemStatusEntity) deleted
Electronic messages	DataEvent_ElectronicMess...	Electronic message item statuses (ElectronicMessageItemStatusEntity) updated
Electronic messages	DataEvent_ElectronicMess...	Electronic message processing (ElectronicMessageProcessingEntity) created
Electronic messages	DataEvent_ElectronicMess...	Electronic message processing (ElectronicMessageProcessingEntity) deleted
Electronic messages	DataEvent_ElectronicMess...	Electronic message processing (ElectronicMessageProcessingEntity) updated
Electronic messages	DataEvent_ElectronicMess...	Electronic messages (ElectronicMessagesEntity) created
Electronic messages	DataEvent_ElectronicMess...	Electronic messages (ElectronicMessagesEntity) deleted
Electronic messages	DataEvent_ElectronicMess...	Electronic messages (ElectronicMessagesEntity) updated
Electronic messages	DataEvent_ElectronicMess...	Electronic messages log (ElectronicMessagesLogEntity) created
Electronic messages	DataEvent_ElectronicMess...	Electronic messages log (ElectronicMessagesLogEntity) deleted
Electronic messages	DataEvent_ElectronicMess...	Electronic messages log (ElectronicMessagesLogEntity) updated
Electronic messages	DataEvent_ElectronicMess...	Electronic message statuses (ElectronicMessageStatusEntity) created
Electronic messages	DataEvent_ElectronicMess...	Electronic message statuses (ElectronicMessageStatusEntity) deleted
Electronic messages	DataEvent_ElectronicMess...	Electronic message statuses (ElectronicMessageStatusEntity) updated
Electronic messages	DataEvent_EMActionClass...	Action class settings (EMActionClassEntity) created

These data events allow you to **react to key lifecycle changes** in the electronic message framework, such as:

- A new invoice submission starting (created)
- A file being logged, signed, or sent (updated)
- A document being rejected, retried, or deleted (deleted)

These events don't process the document, they **notify you when something important happens**, enabling **real-time automation**, **exception handling**, and **compliance visibility** via Power Automate.



When and Why to Use These Data Events

Usage Area	How These Data Events Help
Compliance Monitoring	Trigger workflows when a document is rejected, fails submission, or receives a status update.
Exception Management	Automatically assign failed or rejected items to business users or finance teams.
Audit Logging	Log changes to submission lifecycle in SharePoint or Dataverse for audit trail visibility.
Status Notifications	Inform internal teams or external partners when a document status changes (e.g., “Sent” or “Failed”).
Integration Triggers	Call external systems (e.g., SAP, email gateways, case management) when electronic records are created.

☒ Best Practices & Recommendations

Recommendation	Why It Matters
Use created and updated events only	These are the most useful for automation; deleted is typically noise unless auditing deletions.
Avoid tight coupling to internal tables	Use virtual entities or Dataverse instead of accessing D365 directly inside flows.
Log every triggered flow	For traceability and audit, log each triggered business event to a central store like SharePoint or Dataverse.
Filter intelligently inside the flow	Not every update means failure, add conditions like Status = Failed before triggering alerts.
Disable unused events	Keep the environment clean and performant by disabling business events not relevant to your use cases.
Always secure your endpoints	When using Power Automate HTTP triggers or Azure Logic Apps, ensure token or certificate authentication.
Use separate flows for Dev, UAT, and Prod	Avoid cross-environment noise; follow standard ALM practices for compliance flows.



2. Use Custom Alerts to Notify for the changes

Electronic Submission Log Tables

Table Name	Purpose	Key Field(s)	How It Works	Best Practices
BusinessDocumentExecutionResult	Stores the overall outcome of a document submission or pipeline execution. Each record represents one execution run.	Status (e.g., Succeeded, Failed, InProgress)	Created automatically during ER pipeline execution (e.g., electronic invoicing, format transformation, or storage). It logs the top-level result of the pipeline process.	Use Status to monitor success/failure of document runs. Trigger Power Automate flows when Status = Failed. Filter by execution date or legal entity to isolate issues.
BusinessDocumentExecutionResultLog	Stores detailed step-by-step messages linked to each execution (e.g., validation result, error, success note).	Message (free text: error message, success logs, validation outcome)	For each execution result, multiple log entries are stored here. These logs capture runtime activity like signing, transformation, storage, or submission response from a government portal.	Always log both success and error messages for traceability. Display Message values in submission audit screens or notifications. Join with Executio

✓ Use **Power Automate to listen alerts** use business events , no dependency on X++ or hardcoding.



You can setup alerts directly from Table Browser

The screenshot shows the Dynamics 365 Table Browser interface. The table being viewed is 'BusinessDocumentExecutionResultLog'. The table has columns: 'modifiedDateTime', 'BusinessDocumentExecutionResult', 'Classification', 'Code', 'CustomData', 'LogLevel', 'Message', and 'Timestamp'. The table contains 30 rows of data. On the right side, the 'Create alert rule' dialog box is open. The 'Alert me when' section is highlighted with a red circle. It shows 'Table name' as 'Action logs' and 'Event' as 'is set for'. The 'Alert me until' section is also visible, showing 'No end date'. The 'Alert me with' section shows 'Subject' as 'Field LogLevel in table Action...' and 'Message' as 'Solution: The invoice signer is not spec...'. The 'Send email' section is also visible, with 'Send externally' set to 'Yes'.



3. Use Existing Data Entities

Dynamics 365 exposes many useful **data entities**



Target entities | AX : Operations

My view * v

Filter

Entity	Staging table
Available statuses for action after change electronic messages	EMActionStatusToStaging
Available statuses for action to change electronic messages	EMActionStatusFromStaging
Electronic message action	EMActionStaging
Electronic message additional fields	EMMessagesAdditionalFieldsStaging
Electronic message additional fields for processing	EMProcessingMessageAdditionalFieldsStaging
Electronic message item statuses	ElectronicMessageItemStatusStaging
Electronic message item types	EMItemTypeStaging
Electronic message items	ElectronicMessageItemsStaging
Electronic message items additional fields	EMItemsAdditionalFieldsStaging
Electronic message items log	ElectronicMessageItemsLogStaging
Electronic message items log details	ElectronicMessageItemsLogDetailsStaging
Electronic message items log V2	ElectronicMessageItemsLogV2Staging
Electronic message processing	ElectronicMessageProcessingStaging
Electronic message statuses	ElectronicMessageStatusStaging
Electronic messages	ElectronicMessagesStaging
Electronic messages log	ElectronicMessagesLogStaging

Use **data entities** for:

- Archival/export
- Reprocessing or data cleansing
- Historical reporting with large data sets

Scenario	Recommended Entity Type	Why
Monitoring submission results	Virtual Entity	Lightweight, real-time view (e.g., failed invoices)
Triggering compliance alerts	Virtual Entity	Can trigger Power Automate flows without data movement
Bulk error log exports to Excel/Data Lake	Data Entity	Staging tables are optimized for bulk reads
Reprocessing or correcting logs	Data Entity	Allows manual data manipulation via staging
Audit dashboards (Power BI)	Virtual Entity (for live), Data Entity (for historical)	Mix both , live for dashboard tiles, data entities for trend analysis



Scenario	Recommended Entity Type	Why
Custom ALM/DevOps processes	Data Entity	Better control with versioning and import/export scripts

Example Compliance Workflows (Conceptual)

Scenario	Outcome
B2G invoice to Spain fails	Auto-alert in Teams and log entry in SharePoint
Rejected XML file	Assign a Power Automate task to compliance officer
New format version deployed	Trigger a review/acknowledgment by test team
Tax override detected in TCS	Notify tax owner for secondary approval

Real-World Application Scenarios

Even without configuration access, a solution architect can design:

- An **event-driven compliance framework**
- A **scalable notification model** with environment separation (Dev/UAT/Prod)
- Modular flows for **audit-readiness and governance**

“Power Platform gives you visibility. Globalization Studio gives you control. Together, they give you confidence.”

Best Practices from the Field

Tip	Why It Matters
Use standardized flow names	Easy to trace and maintain
Define owner/role per flow	Critical for audits and support
Use centralized logging	Avoid redundant alerting
Secure all endpoints	Use Azure AD auth, tokens, and IP filters
Test in layers	Keep your compliance flows loosely coupled



Related Articles

- [GS508 – ALM for Compliance Features](#)
- [GS514 – Routing and Storage](#)
- [GS517 – Logs and Error Management](#)
- [GS528 – Archival and Compliance Audit](#)

Coming Up Next

In [GS524 – Embedding Business Rules and Exceptions in Globalization Studio](#), we'll show you how to take compliance personalization further by embedding dynamic rules directly into your features.

You'll learn how to:

- Apply applicability rules using context variables
- Add validations in Format Designer to catch errors early
- Control pipeline steps based on customer type, amount, or geography
- Design real-world use cases like B2B vs. B2G logic in Spain

 Continue reading: [GS524 – Business Logic and Rule Embedding](#) →