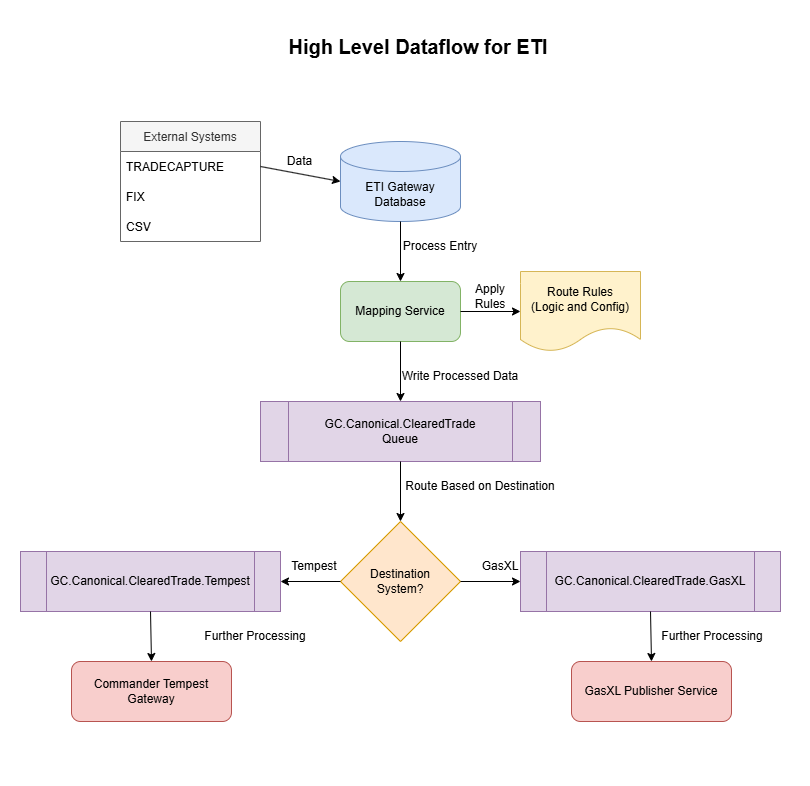
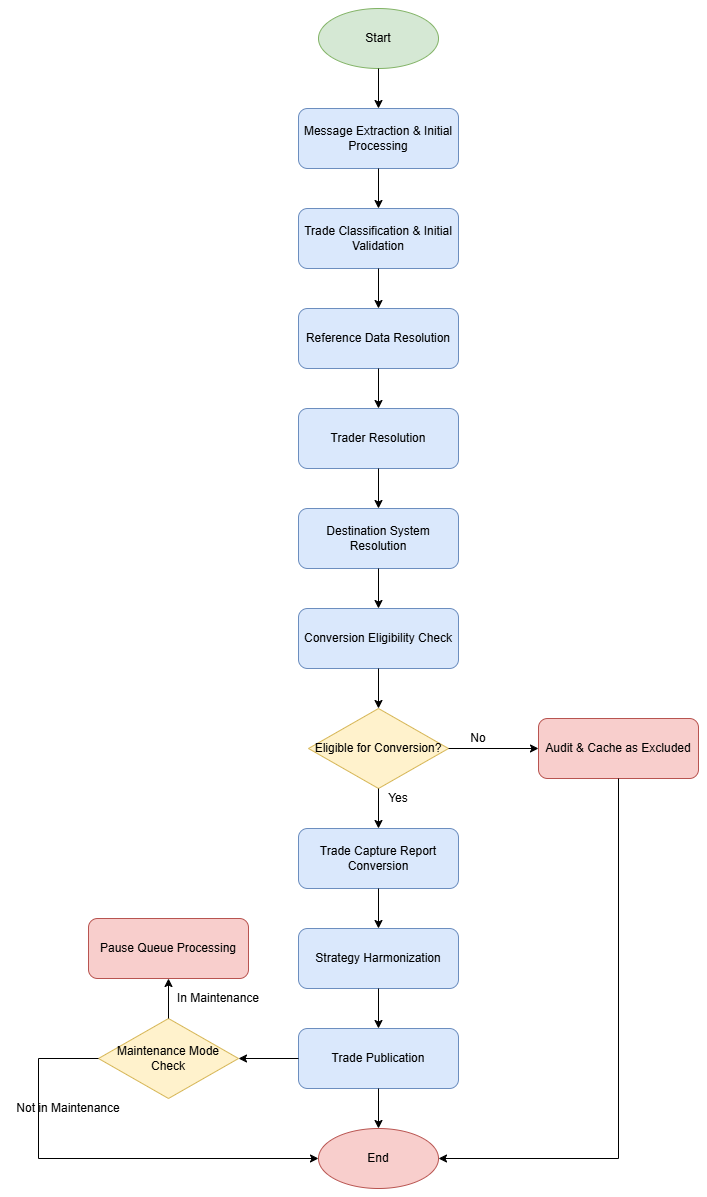
# **High Level Dataflow for ETI**

* Data is received from an external system, which could be TRADECAPTURE, FIX, or CSV.
* This data is then written to the ETI Gateway Database.
* Each entry is processed using a mapping service, which stamps the destination system using Route rules (logic and config).
* The processed data is written to the GC.Cananical.ClearedTrade Queue.
* The Queue uses the destination system to determine whether the data goes to GC.Cananical.ClearedTrade.Tempest or GC.Cananical.ClearedTrade.GasXL.
* If the destination is Tempest, the data is sent to the Commander Tempest Gateway for further processing.
* If the destination is GasXL, the data is sent to the GasXL Publisher service for further processing.



# **Mapping Service**

The message processing system is designed to handle incoming trade messages from various external systems, enrich them with additional data, resolve reference information, determine appropriate routing, and convert them to a canonical format before publication.



Key components include:

* **MessageExtractor**: Responsible for the initial processing of incoming messages
* **MapperMessageProcessor**: Maintains correlation keys and processes messages
* **TradeCaptureRecord**: Data structure representing the trade with enriched properties
* **MappingService**: Provides reference data resolution capabilities
* **SymbolConvertService**: Handles security identifier conversions
* **CommanderAPI**: External system for resolving user and reference data
* **TradeConverter**: Converts trade data between different formats
* **TradePublisher**: Publishes processed trades to downstream systems

## Message Extraction Process

The message processing flow begins with the MessageExtractor component, which is responsible for receiving and initially processing incoming messages from external systems. This component serves as the entry point for all trade-related messages that need to be processed by the system.

### MessageExtractor ProcessMessage

When a message arrives from an external system, the MessageExtractor component invokes its ProcessMessage method to begin processing. The ProcessMessage method acts as a gateway, ensuring that only valid messages proceed to the next stage of processing. Invalid or malformed messages are rejected at this stage, with appropriate error handling and logging.

### MapperMessageProcessor:Process

Once the message passes through the MessageExtractor, it is handed off to the

MapperMessageProcessor component, which invokes its Process method. A critical function of this method is to maintain a consistent correlation key throughout the message processing lifecycle.

## TradeCaptureRecord Creation and Enrichment

After establishing the correlation key, the MapperMessageProcessor creates a TradeCaptureRecord object from the incoming message. This object serves as the primary data structure for representing the trade throughout most of the processing pipeline.

During creation of the TradeCaptureRecord, several properties are enriched with data from the original message. The table below shows the mapping between source fields in the incoming message and target fields in the TradeCaptureRecord:

|  |  |  |
| --- | --- | --- |
| **Source Field** | **Target Field** | **Description** |
| trdCaptRpt.Header.SourceSystem | application.eti.source | Source system identifier |
| trdCaptRpt.TrdSummary.TrdSubTyp | application.eti.trade\_type | Type of trade |
| trdCaptRpt.TrdLegs[0].ContractCode | application.eti.contract\_code | Contract code from first trade leg |
| trdCaptRpt.TrdSummary?.TxnTm | application.eti.execution\_time | Transaction/execution timestamp |
| trdCaptRpt.TrdSummary?.TraderId | application.eti.trader | Trader identifier |
| trdCaptRpt.TrdSummary?.ClearingBroker | application.eti.broker | Clearing broker information |
| trdCaptRpt.TrdSummary?.ClearingAccount | application.eti.account | Clearing account information |

## Bilateral Platts Partials Detection

After the TradeCaptureRecord is created and enriched, the system performs a check to determine if the trade represents Bilateral Platts partials. This is determined by examining the LastQty field of the first trading leg.

## Message Caching and Publication Decision

The system maintains a cache of processed messages to handle various scenarios, including duplicate detection and partial trade aggregation. At this stage, a decision is made regarding whether to publish the message immediately or continue with further processing:

If the message is already in the cache and is not classified as Bilateral Platts partials, it is published immediately. Otherwise, processing continues to the next stage

## Initial Rule Application

Before proceeding to the next major processing stage, the system applies initial exclude and hold rules from the configuration. These rules determine whether certain trades should be excluded from further processing or held for manual review.

The rules are typically defined in a configuration file and are applied based on various criteria, such as trade type, source system, and other trade attributes. This initial rule application helps filter out trades that should not proceed through the full processing pipeline, improving efficiency and reducing the risk of processing invalid or unwanted.

# Trade Classification and Initial Validation

## Pre-conversion Clean-up Workarounds

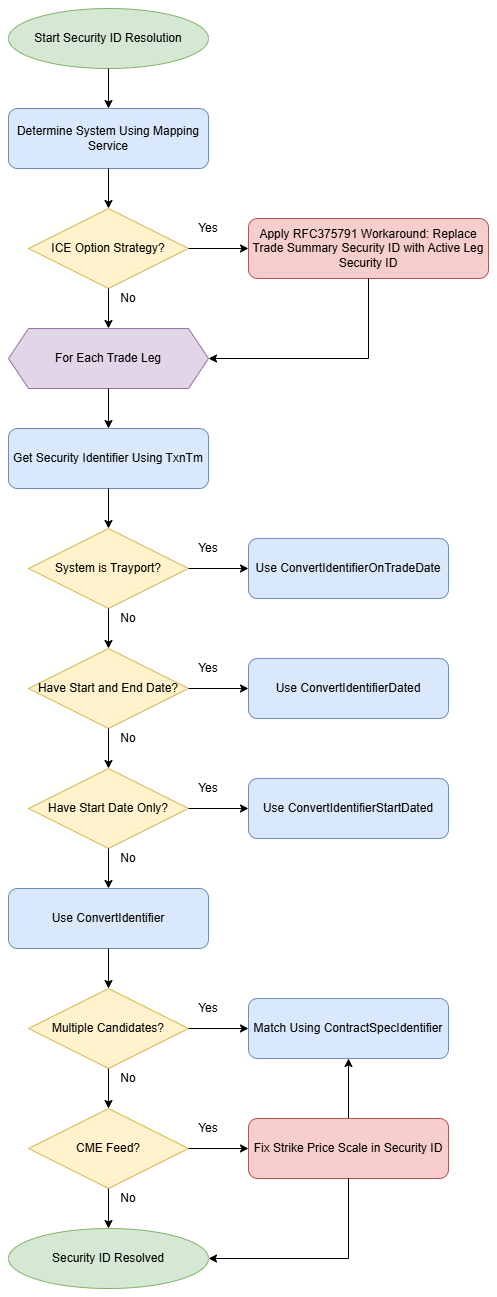
Before proceeding with the main processing flow, the system applies several pre-conversion clean-up workarounds to address known issues with specific data sources. These workarounds ensure that the data is in a consistent format before further processing.

* **TT Prices Workaround** - Trading Technologies (TT) sometimes provides price information in a format that requires normalization. This workaround adjusts the price format to ensure consistency with the system's expected format, preventing downstream processing errors.
* **TT Clearing Broker Workaround** - For trades originating from Trading Technologies, the clearing broker information may require special handling. This workaround ensures that the clearing broker is correctly identified and formatted according to system requirements.
* **ICE Security IDs Workaround** - Intercontinental Exchange (ICE) security identifiers sometimes require special handling due to format inconsistencies. This workaround normalizes ICE security IDs to ensure they can be properly resolved in subsequent processing steps.
* **Mizuho Trader-specific Subaccounts Workaround** - For trades involving Mizuho traders, subaccount information may require special handling. This workaround ensures that trader-specific subaccounts are correctly processed and associated with the appropriate entities.
* **Physical Trade Exclusion Logic** - The system implements specific logic for handling physical trades, particularly those originating from ICE. Physical trades are excluded from processing if the following conditions are met: 1. ICE did not provide an exchange contract code 2. Users have not set up a mapping from hub and product. This exclusion prevents the system from attempting to process physical trades that lack the necessary reference data for proper handling.

## Security ID Resolution

A critical step in the initial validation process is ensuring that the SecurityId is properly resolved. This involves:

1. Checking if the SecurityId is present in the incoming message
2. Validating the format of the SecurityId
3. Ensuring the SecurityId can be mapped to known instruments



## Decision Points in Initial Validation

The table below summarizes the key decision points during the trade classification and initial validation stage:

|  |  |  |
| --- | --- | --- |
| **Decision Point** | **Condition** | **Action** |
| Bilateral Platts  Partials | TrdLegs[0].LastQty == 25 | Classify as Bilateral Platts partials |
| Physical Trade  Exclusion | ICE source + No exchange contract code + No user mapping | Exclude trade |
| Security ID  Resolution | SecurityId cannot be resolved | Flag for review or exclude |

## Validation Outcomes

The initial validation process can result in several outcomes:

1. **Continue Processing**: The trade passes all validation checks and proceeds to the next stage
2. **Exclude Trade**: The trade is excluded from further processing due to failing specific validation criteria
3. **Hold for Review**: The trade is placed on hold for manual review due to potential issues
4. **Cache for Later Processing**: The trade is cached for later processing, possibly waiting for related trades

# Reference Data Resolution

## Overview of Reference Data Resolution

After the initial validation, the system proceeds to resolve various reference data elements associated with the trade. This resolution process is critical for ensuring that the trade is properly categorized, routed, and processed. The reference data resolution stage involves interactions with multiple external systems and services to gather the necessary information.

## Mapping Service Integration

The system integrates with a dedicated mapping service to resolve reference data. This service provides a centralized repository of mapping information that translates between different identifier systems, entity references, and other trade-related data.

### Security ID Normalization

One of the primary functions of the reference data resolution stage is to normalize security identifiers. This process ensures that all security IDs are in a consistent format that can be recognized and processed by downstream systems.

The security ID normalization process follows a complex decision tree, as illustrated in the following diagram:

#### Security ID Resolution Diagram

The process varies depending on the source system and available date information:

|  |  |  |  |
| --- | --- | --- | --- |
| **System** | **Security ID Resolution**  **Method** | **Parameters** | **Description** |
| Trayport | ConvertIdentifierOnTradeDate | Identifier, TradeDate | Converts identifier using trade date context |
| Other (with start/ end date) | ConvertIdentifierDated | Identifier, StartDate,  EndDate | Converts identifier using date range |
| Other (with start date only) | ConvertIdentifierStartDated | Identifier, StartDate | Converts identifier using start date only |
| Other (no dates) | ConvertIdentifier | Identifier | Basic identifier conversion without date context |

For ICE option strategies, a special workaround (RFC375791) is applied, replacing the trade summary security ID with the active leg security ID.

For CME feeds, additional handling is required for the strike price scale in the security ID, as the scale cannot be reliably determined from the security ID itself and must be obtained from a dedicated field.

## Broker Trade Resolution

The system resolves broker trades by matching trade information with known broker entities. This resolution process involves:

1. Identifying the broker from the trade information
2. Validating the broker against known entities
3. Resolving any broker-specific attributes or requirements

## Trade Hold Determination

During reference data resolution, the system determines if trades should be placed on hold based on various criteria:

Trades may be placed on hold for several reasons: - Missing required reference data - Matching specific hold rules defined in the configuration - Requiring manual review due to unusual characteristics or potential issues.

**Key components involved in this stage include**: - TradeCaptureRecord: The primary data structure representing the trade - MappingService: Provides reference data resolution capabilities -

SymbolConvertService: Handles security identifier conversions - CommanderAPI: External system for resolving additional reference data

These components work together to ensure that all necessary reference data is properly resolved before proceeding to the next processing stage.

## Special Cases and Exception Handling

The reference data resolution process includes handling for several special cases and exceptions:

* **Multiple Security ID Candidates** - When multiple security ID candidates are returned from the conversion process, the system attempts to match using the ContractSpecIdentifier. This requires getting the known tradable instrument type using the Commander API.
* **Inactive Legs** - For inactive trade legs, security identifier resolution is attempted within a try block, with failures being logged but not necessarily causing the entire process to fail.
* **Missing Reference Data** - When essential reference data cannot be resolved, the system follows configurable rules to determine whether to exclude the trade, place it on hold, or attempt alternative resolution methods.
* **Audit and Logging** - Throughout the reference data resolution process, the system maintains detailed audit records and logs to track: - Resolution attempts and outcomes - Data transformations - External service calls - Error conditions and handling

# Trader Resolution

## Overview of Trader Resolution

The trader resolution stage is a critical component of the message processing flow, ensuring that trades are properly attributed to the correct traders and that trader-specific rules and configurations are applied. This process involves resolving both external and internal trader identifiers and handling special cases for different trade types.

## Default Trader Resolution

### ResolveDefaultTrader from XML

The system uses a conditional approach to resolve default traders:

This resolution is only performed when either: - There is no Glencore trader identified in the trade, OR there is no Glencore-trader-specific clearing account

The default trader information is retrieved from XML configuration files, which map various trade attributes to default trader identifiers. This ensures that even when explicit trader information is missing, the system can still attribute the trade to an appropriate trader based on predefined rules.

## Position Transfer Trader Resolution

For trades classified as type "Transfer", the system uses a specialized resolution process:

The ResolvePositionTransferTrader function applies specific logic for position transfer trades, which may involve different trader attribution rules compared to standard trades. This specialized handling ensures that position transfers are properly tracked and attributed within the system.

## Trader Resolution Process

The trader resolution process involves several steps:

* External Trader Resolution - For external traders, the system resolves the user by passing the TraderId to the Commander API:
* Internal Trader Resolution - For internal traders, the system resolves the user by passing the InternalTraderId to the Commander API: This resolution ensures that internal trader references are properly mapped to the correct entities.
* Special Cases in Trader Resolution - For Singapore or ST Shipping traders, special handling is applied to ensure trades are booked for the trader's company: Glencore Commodities Limited Trades - A special case exists for Glencore Commodities Limited trades booking into Tempest

## Trades on Hold

If a trade is determined to be on hold during the trader resolution process, the system performs additional actions:

These actions ensure that: 1. The hold status is properly audited for compliance and troubleshooting purposes 2. The trade is marked as excluded in the cache to prevent further processing until the hold condition is resolved

## Decision Points in Trader Resolution

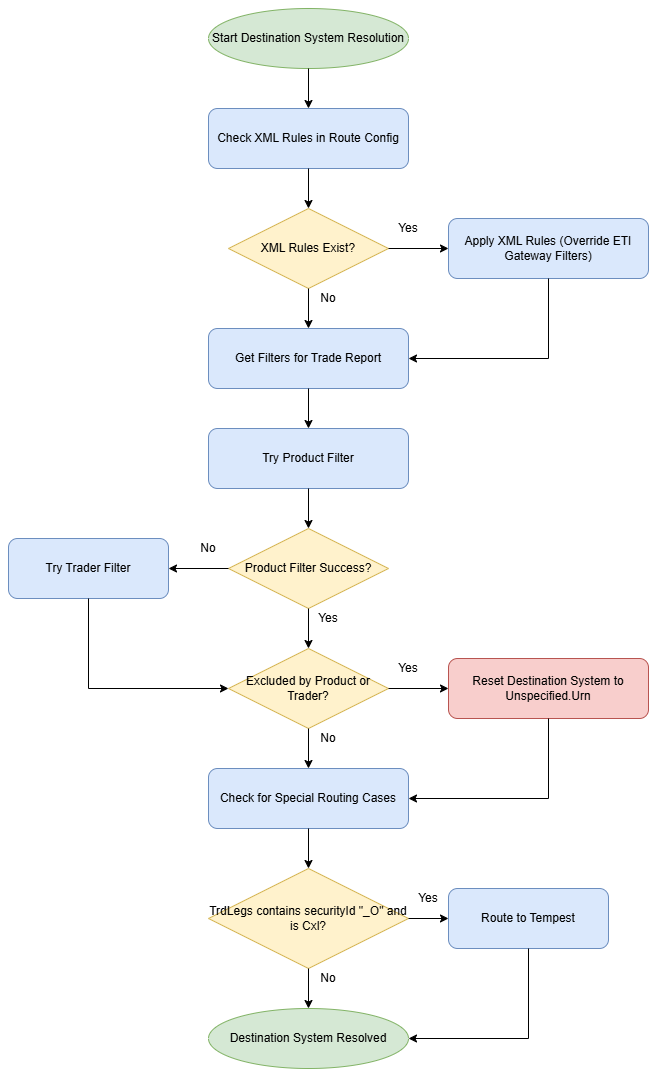
The trader resolution process includes several key decision points:

|  |  |  |
| --- | --- | --- |
| **Decision Point** | **Condition** | **Action** |
| Default Trader  Resolution | No Glencore trader OR No Glencore-trader specific clearing account | Use ResolveDefaultTrader from  XML |
| Position Transfer  Trader | Trade type is Transfer | Use  ResolvePositionTransferTrader |
| Hold Status | Trade on hold and not already excluded | Audit and cache as excluded |

# Destination System Resolution

## Overview of Destination System Resolution

The destination system resolution stage is responsible for determining which downstream system should receive the processed trade. This decision is based on a combination of configuration rules, trade attributes, and special case handling. The process ensures that trades are routed to the appropriate systems for further processing.



## XML Configuration Rules

### Route Configuration Settings

The first step in destination system resolution is to check for any XML rules defined in the Route configuration settings:

These XML rules take precedence over the ETI Gateway filters and can override the default routing behavior. The rules are typically defined based on business requirements and may specify routing based on various trade attributes.

### XML Rule Application

When XML rules are present, they are applied according to their defined priority and conditions. The rules may specify: - Destination systems for specific contract codes - Routing rules for specific traders - Special handling for certain trade types - Override conditions for default routing behaviour

* Filter Application - After checking XML rules, the system retrieves and applies any filters that are relevant to the trade report: These filters may be based on: - Contract code - Trader identifier - Other trade attributes
* Destination System Determination - By default, the system sets the destination to "Tempest.Urn": However, this default can be overridden based on the application of filters:
* Product Filter - The system first attempts to apply product filters:
* Trader Filter - If product filtering doesn't yield a result, the system falls back to trader filters:
* Exclusion Handling - If the trade is excluded by either product or trader filters, the destination system is reset to "Unspecified.Urn":

This setting effectively marks the trade as not eligible for routing to any specific destination system.

## Special Routing Cases

The system includes additional logic to handle special routing cases:

### Cancellation with Option Identifier

A specific rule routes cancellations where the trade legs contain a security ID with "\_O" suffix to Tempest: This special case ensures that option cancellations are properly routed regardless of other filtering rules.

## Decision Points Summary

The key decision points in the destination system resolution process are summarized in the following table:

|  |  |  |
| --- | --- | --- |
| **Decision**  **Point** | **Condition** | **Action** |
| XML Rules | XML rules exist in Route config | Apply XML rules (override ETI Gateway filters) |
| Product Filter | Product filter succeeds | Use destination from product filter |
| Trader Filter | Product filter fails and trader filter  succeeds | Use destination from trader filter |
| Exclusion | Product or trader excluded | Reset destination system to  Unspecified.Urn |
| Special  Routing | TrdLegs contains securityId "\_O" and is  Cxl | Route to Tempest |

## Component Interactions

The destination system resolution process involves interactions between multiple system components:

* TradeCaptureRecord: Contains the trade information being processed
* RouteConfiguration: Provides XML rules and filter definitions
* MapperMessageProcessor: Applies routing logic based on configuration
* FlowRules: May influence routing decisions based on business rules

These components work together to ensure that trades are routed to the appropriate destination systems based on their characteristics and applicable business rules.

# Reference Data Resolution for Destination System

## Overview of Destination-Specific Reference Data

After determining the destination system, the process continues with resolving reference data specific to that destination. This stage ensures that all necessary reference data for the chosen destination system is properly populated, enabling correct interpretation and processing of the trade at its destination.

## Exchange Determination Logic

The system uses different approaches to determine the exchange based on the source system:

Source System-Based Exchange Determination

If the exchange can be determined by the source system:

In this case, the system: 1. looks up the exchange based on the source system 2. Looks up the counterparty based on the source system

* **Trayport-Specific Logic** - If the source system is Trayport, special handling is required: This special handling accounts for broker-executed exchange deals in Trayport, where the broker's name may appear in all counterparty/venue fields, with the exchange only shown as part of the Trayport instrument.
* **OTC and ICE with ICEClear** - For OTC trades or ICE trades with ICEClear as the counterparty: These trades may be subject to contract filters, which can exclude certain trades based on contract-specific criteria.
* **Clearer Feeds** - For feeds coming from clearers. In this case, the system uses the counterparty information provided in the clearer feed.

## Reference Data Population

After determining the exchange and counterparty information, the system populates new reference data fields:

This process populates several key fields:

1. **Counterparty**: The entity on the other side of the trade
2. **ClearingAccount**: The account used for clearing the trade
3. **TradingVenue**: The venue where the trade was executed
4. **CounterPartyBook**: The book associated with the counterparty

These fields are essential for proper trade processing at the destination system and ensure that the trade is correctly attributed and accounted for.

## Special Cases

* **Broker-Executed Exchange Deals** - For Trayport trades, special handling is required for broker-executed exchange deals: In these cases, the broker's name may appear in all counterparty/venue fields, and the exchange is only shown as part of the Trayport instrument. The system must extract this information to properly populate the reference data.
* **ICE with ICEClear as Counterparty** - For ICE trades with ICEClear as the counterparty: These trades are typically excluded in the contract filters, as they require special handling that is outside the normal processing flow.

## Component Interactions

The reference data resolution for destination systems involves interactions between multiple components:

* TradeCaptureRecord: Contains the trade information being processed
* MappingService: Provides reference data resolution capabilities
* RouteConfiguration: Contains rules that may affect reference data resolution
* CommanderAPI: May be used to resolve additional reference data

These components work together to ensure that all necessary reference data is properly resolved and populated for the destination system.

# Conversion Eligibility Check

Before proceeding to convert the trade capture report to the destination system's format, the system performs a series of checks to determine if the trade is eligible for conversion. This stage acts as a final validation gate before committing resources to the conversion process.

## Eligibility Conditions

The system evaluates several conditions to determine conversion eligibility:

* **Destination System Check -** The first check verifies that a valid destination system has been determined.If the destination system is "Unspecified.Urn", the trade is not eligible for conversion. This typically occurs when: - The trade was excluded by product or trader filters - Required routing information could not be determined - The trade matches specific exclusion criteriaWhen this condition is met, the system audits what is missing (trader, contract, etc.) to provide context for why the conversion was not performed.
* **UnlinkedEfs Check** - Using flow rules, the system checks if the trade represents an unlinked EFS (Exchange for Swap): Unlinked EFS trades require special handling and are not eligible for standard conversion. The system audits these cases for later processing or manual intervention.
* **EFP Trade Type for NYMEX Floor Check -** The system also checks for a specific condition related to EFP (Exchange for Physical) trade types on the NYMEX floor:This check identifies EFP trades on the NYMEX floor that should be excluded from conversion based on specific business rules.

## Decision Points Summary

The key decision points in the conversion eligibility check are summarized in the following table:

|  |  |  |
| --- | --- | --- |
| **Decision Point** | **Condition** | **Action** |
| Destination System | Destination system is Unspecified.Urn | Do not proceed with conversion |
| UnlinkedEfs | UnlinkedEfs detected via flow rules | Do not proceed with conversion |
| EFP Trade Type | ExcludeEfpTradeTypeForNymexFloor is true | Do not proceed with conversion |
|  |  |  |

## Outcome of Eligibility Check

The outcome of the conversion eligibility check is a boolean decision: - **True**: The trade is eligible for conversion and proceeds to the next stage - **False**: The trade is not eligible for conversion and is excluded from further processing

Trades that fail the eligibility check may be: - Flagged for manual review - Queued for later processing after certain conditions are met.

# Trade Capture Report Conversion

## Overview of Conversion Process

Once a trade has passed the eligibility check, it proceeds to the conversion stage. This critical process transforms the trade capture report from the exchange format to the Commander vocabulary, ensuring that all data elements are properly translated and formatted for the destination system.

The security ID conversion is one of the most complex parts of the trade conversion process. It involves cloning the trade report and updating the security IDs using a sophisticated logic flow:

* **System-Specific Conversion Logic** - The conversion logic varies depending on the system:
* **Multiple Candidates Handling -** When multiple security ID candidates are returned:
* **CME Feed Special Handling -** For CME feeds, special handling is required for the strike price scale:
* **Trader Conversion** - The trader conversion process resolves both external and internal traders:
* **External Trader Resolution -**
* **Internal Trader Resolution -**
* **Internal Company Resolution -** The internal company resolution process determines the company associated with the trade:
* **Execution Broker Resolution -** The execution broker is updated using reference data:

## Security ID Resolution Methods

The following table summarizes the methods used for security ID resolution based on system and available date information:

|  |  |  |  |
| --- | --- | --- | --- |
| **System** | **Security ID Resolution**  **Method** | **Parameters** | **Description** |
| Trayport | ConvertIdentifierOnTradeDate | Identifier, TradeDate | Converts identifier using trade date context |
| Other (with start/ end date) | ConvertIdentifierDated | Identifier, StartDate,  EndDate | Converts identifier using date range |
| Other (with start date only) | ConvertIdentifierStartDated | Identifier, StartDate | Converts identifier using start date only |
| Other (no dates) | ConvertIdentifier | Identifier | Basic identifier conversion without date context |

# Strategy Harmonization

## Overview of Strategy Harmonization

After the trade capture report conversion, the system performs strategy harmonization, particularly for ICE Spread and Composite Markets. This process ensures that complex trading strategies are properly represented and that redundant or duplicate information is eliminated.

## ICE Spread and Composite Markets Strategy Harmonization

* **Leg Exclusion Logic** - For ICE Spread and Composite Markets, the system applies special logic to exclude legs that represent strips when the underlying are also represented in other legs: This exclusion prevents redundant representation of the same underlying instruments, which could lead to double-counting or other processing errors.
* **Strip Identification -** The system identifies strips using specific attributes in the leg data:
* **Underlying Representation Check -** To determine if underlying are represented in other legs:This check ensures that all underlying instruments in a strip are properly represented in other legs before excluding the strip leg.
* **Strategy Attribute Handling -** In some cases, the system may need to override the Strategy attribute with a default value:When OverrideStrategyWithDefault is set to true, the Strategy attribute is removed, allowing the Tempest Publisher to infer it later based on standard rules.

## Decision Points in Strategy Harmonization

The key decision points in the strategy harmonization process are summarized in the following table:

|  |  |  |
| --- | --- | --- |
| **Decision Point** | **Condition** | **Action** |
| Leg Exclusion | Leg represents strip AND underlyings represented in other legs | Exclude leg |
| Strategy  Attribute | OverrideStrategyWithDefault is true | Remove Strategy  attribute |

These decision points guide the system's behaviour during the strategy harmonization process, ensuring that complex trading strategies are properly represented.

## Outcome of Strategy Harmonization

The outcome of the strategy harmonization process is a trade capture record with: - Properly harmonized legs for ICE Spread and Composite Markets - Appropriate strategy attribute handling based on configuration - Elimination of redundant or duplicate information

# Trade Publication

## Overview of Trade Publication

The trade publication stage represents the final phase of the message processing flow. After all validations, transformations, and harmonisations have been completed, the system converts the trade to its final format and publishes it to the destination system.

* Strategy Attribute Handling - Before publishing, the system checks if the strategy attribute should be overridden with a default value: When OverrideStrategyWithDefault is set to true, the Strategy attribute is removed, allowing the Tempest Publisher to infer it later based on standard rules.
* Trade Conversion to Canonical Format - The system converts the trade to a CanonicalClearedTrade format, which is the standard format used for publication: This conversion ensures that the trade is in a consistent format that can be understood by all destination systems.
* Final Exclusion Check - Before publication, the system performs a final check to determine if the trade has been excluded: The system also checks if the trade matches any exclusion or hold rules after conversion:

## Trade Publication

If the trade passes all checks, it is published to the destination system:

The publication process involves sending the trade to the appropriate destination system in the expected format.

## Maintenance Mode Handling

After attempting to publish the trade, the system checks if the recipient is in maintenance mode:

If the recipient is in maintenance mode, the system pauses queue processing for a configured duration (PauseOnMaintenanceModeMinutes). This prevents the system from continuously attempting to send trades to a system that is temporarily unavailable.

## Decision Points in Trade Publication

The key decision points in the trade publication process are summarized in the following table:

|  |  |  |
| --- | --- | --- |
| **Decision Point** | **Condition** | **Action** |
| Strategy Handling | OverrideStrategyWithDefault is true | Remove Strategy  attribute |
| Trade Exclusion | Trade is excluded | Do not publish |
| Post-Conversion  Rules | Trade matches exclusion or hold rules after conversion | Do not publish |
| Maintenance Mode | Recipient is in MaintenanceMode | Pause queue processing |

These decision points guide the system's behaviour during the trade publication process, ensuring that only appropriate trades are published and that system resources are managed efficiently.