



# CMSAPI Protocol Change Log

## Contents

CMSAPI Protocol Change Log .....	1
CMS 2.15 (Protocol Version: 1.43) .....	1
1. cmsapi_1.proto.....	2
For Users: .....	3
For Developers:.....	3
2. common_1.proto .....	3
For Developers:.....	4
3. profile_1.proto.....	4
For Users: .....	6
For Developers:.....	6
4. traderouting_1.proto.....	7
For Users: .....	8
For Developers:.....	8
5. shared_1.proto .....	8
For Users: .....	9
For Developers:.....	9
CMS 2.14 (Protocol Version: 1.42) .....	9
1. cmsapi_1.proto.....	9
Protocol Version Update: .....	10
New Fields: .....	10
Enhanced Validation:.....	10
Impact on Users .....	10
2. common_1.proto .....	10



Enhanced Validation:.....	10
Erasable Fields:.....	10
Impact on Users .....	10
3. login_1.proto .....	11
Extended Risk Parameters:.....	11
New Message:.....	11
Impact on Users .....	11
4. metadata_1.proto .....	11
New Fields: .....	11
New Messages:.....	12
Breaking Changes:.....	12
Impact on Users .....	12
5. risk_parameter_value_1.proto .....	12
New Message:.....	12
Impact on Users .....	12
6. risk_parameters_1.proto .....	12
New Fields: .....	12
New Message:.....	13
Impact on Users .....	13
7. traderouting_1.proto.....	13
New Fields: .....	13
New Enums: .....	13
Breaking Changes:.....	14
New Enum Values:.....	14
Impact on Users .....	14
CMS 2.13 (Protocol Version: 1.41) .....	14
1. api_limit_1.proto.....	14
Changes .....	14
Impact on Users .....	15



2. cmsapi_1.proto.....	15
Changes .....	15
Impact on Users .....	15
3. common_1.proto .....	15
Changes .....	15
Impact on Users .....	16
4. location_1.proto .....	16
Changes .....	16
Impact on Users .....	16
5. login_1.proto .....	16
Changes .....	16
Impact on Users .....	16
6. metadata_1.proto .....	16
Changes .....	17
Impact on Users .....	17
7. route_1.proto .....	17
Changes .....	17
Impact on Users .....	17
8. session_context_1.proto .....	17
o Changes .....	17
Impact on Users .....	17
9. shared_1.proto .....	17
Changes .....	17
Impact on Users .....	18
CMS 2.12 (Protocol Version: 1.40) .....	18
Overall Impact .....	18
1. cmsapi_1.proto.....	18
Changes .....	18
Impact on Users .....	19



2. common_1.proto .....	19
Changes .....	19
Impact on Users .....	20
3. profile_1.proto .....	20
Changes .....	20
Impact on Users .....	21
4. risk_parameters_1.proto .....	21
Changes .....	21
Impact on Users .....	21
5. traderouting_1.proto.....	22
Changes .....	22
Impact on Users .....	22
CMS 2.11 (Protocol Version: 1.39) .....	22



## CMS 2.15 (Protocol Version: 1.43)

- Summary of Protocol Changes (CMS 2.15, January, 2026)
  - Below is a summary of key functional changes in the protocol files that affect users and developers. Each numbered section describes updates to a single .proto file, groups changes by category, and details impacts and migration guidance. Breaking changes are called out only where actual breaking edits occurred.

### 1. cmsapi\_1.proto

This file update increments the protocol minor version and deprecates several sales-series-scoped fields/messages in favor of profile-scoped equivalents. Some message definitions that previously lived here were relocated into profile\_1.proto.

- **Protocol minor version** increment from 42 to 43.
- Multiple profile- and sales-series-related fields/messages are marked obsolete and documented to be replaced by profile-scoped messages.
- Two message definitions ([ProfileSearchRequest](#) and [ProfileSearchResultRecord](#)) were moved out of this file into profile\_1.proto (relocation only; see profile\_1.proto for details).
  - **Deprecations and renames**
  - **InformationRequest**
  - [customer\\_sales\\_series\\_authorization\\_list\\_request](#) (field 9) is now represented as [obsolete\\_customer\\_sales\\_series\\_authorization\\_list\\_request](#) with `deprecated = true` and a comment directing clients to [profile\\_scope\\_request.profile\\_to\\_profile\\_links\\_request](#).
  - **SearchRequest**
  - [profile\\_search\\_request](#) (field 4) is now [obsolete\\_profile\\_search\\_request](#) (`deprecated = true`). Clients are directed to use [profile\\_scope\\_request.profile\\_search\\_request](#).
  - [sales\\_series\\_search\\_request](#) retained as [obsolete\\_sales\\_series\\_search\\_request](#) (`deprecated = true`). Use [profile\\_scope\\_request.profile\\_search\\_request](#) instead.
  - **OperationRequest**
  - [update\\_customer\\_sales\\_series\\_authorization\\_list](#) (field 14) renamed/marked [obsolete\\_update\\_customer\\_sales\\_series\\_authorization\\_list](#) (`deprecated = true`) with guidance to use [profile\\_scope\\_request.update\\_profile\\_to\\_profile\\_links](#).
  - Clarifying comment added: trade-routing logins cannot be linked to profiles with non-individual legal type.
    - **InformationResult / SearchResult / OperationResult**
  - Corresponding result fields for the above requests have been renamed/marked obsolete (e.g., [obsolete\\_customer\\_sales\\_series\\_authorization\\_list\\_result](#), [obsolete\\_profile\\_record](#), [obsolete\\_sales\\_series\\_record](#),



obsolete\_update\_customer\_sales\_series\_authorization\_list\_result) with comments pointing to profile-scoped results.

- **Moved definitions (non-breaking relocation)**
- [ProfileSearchRequest](#) and [ProfileSearchResultRecord](#): removed from cmsapi\_1.proto and provided in profile\_1.proto. This is a relocation; the messages themselves continue to exist but under profile\_1.proto.

#### *For Users:*

- End-user behavior is not expected to change immediately. Server-side preference is shifting toward profile-scoped endpoints, so UI flows that previously surfaced sales-series-specific listings may begin to rely on profile-scoped results. Example: profile search widgets should prepare for results produced by the profile scope search endpoint instead of a CMS-wide search message.

#### *For Developers:*

- Actionable migration steps

1. Replace usages of deprecated request/result fields with the profile-scoped equivalents:

- [InformationRequest.customer\\_sales\\_series\\_authorization\\_list\\_request](#) to [profile\\_scope\\_request.profile\\_to\\_profile\\_links\\_request](#).
- [SearchRequest.profile\\_search\\_request](#) to [profile\\_scope\\_request.profile\\_search\\_request](#).
- [OperationRequest.update\\_customer\\_sales\\_series\\_authorization\\_list](#) and its result to [profile\\_scope\\_request.update\\_profile\\_to\\_profile\\_links](#) and [profile\\_scope\\_result.update\\_profile\\_to\\_profile\\_links](#).

2. Update imports/usages for moved messages:

- If your code referenced [ProfileSearchRequest](#) or [ProfileSearchResultRecord](#) from cmsapi\_1.proto, change imports to profile\_1.proto and regenerate code. This is a file relocation—adjust build/import paths accordingly.

3. Update protocol-version to 1.43.

## **2. common\_1.proto**

This file update adds a new search matching rule, opens NamedEntity for extensions, and clarifies profile legal-type relationships in comments (semantic validation guidance).

- Added a new SearchOption matching rule.
- NamedEntity now declares extension slots (tag range 100..199).
- Clarified [legal\\_type](#) semantics in Profile comments (no field number/type changes).
  - **SearchOption**



- **New enum value**
- [SearchOption.SearchMatchingRule](#): added EXISTS (value = 5).
- Purpose: allow tests for existence/non-existence of a parameter (text: "1" = exists, "0" = not exists).
  - **NamedEntity extensions**
  - **Extension slots declared**
- NamedEntity now declares extension tags 100..199 to permit protocol extensions attached to NamedEntity without changing the base message.
  - **Profile legal-type clarification (comments only)**
  - **Validation/semantics**
- Comments clarify:
- Child Admin/Customer profiles linked to parent Admin/Customer profiles must be of [LegalType](#) INDIVIDUAL.
- Parent Admin/Customer profile cannot be [LegalType](#) INDIVIDUAL.
- These are semantic/comment clarifications—no numeric changes.

#### *For Developers:*

- Actionable guidance
1. Add support for [SearchOption.SearchMatchingRule](#) = EXISTS:
  2. Handle [NamedEntity](#) extensions:
    - Be aware the [NamedEntity](#) message now allows extensions in tags 100..199. Update code generation or descriptor handling to consume these extensions where applicable.
  3. Enforce [legal\\_type](#) relationship rules:
    - a. A child Admin/Customer profile linked to a parent Admin/Customer profile must be [LegalType](#) INDIVIDUAL.
    - b. A parent Admin/Customer profile must not be [LegalType](#) INDIVIDUAL.

### **3. profile\_1.proto**

This file introduces profile-scoped request/result messages, consolidates profile search structures, imports validation constraints, and inherits relocated profile search message definitions from [cmsapi\\_1.proto](#).

- Added import of [Validation/constraints\\_1.proto](#) for field-level validation annotations.
- Introduced [ProfileScopeRequest/ProfileScopeResult](#) enhancements for profile-to-profile link operations and profile search.



- Added [ProfileToProfileLinksRequest/Result](#) and [ProfileSearchRequest/ProfileSearchResult](#) types.
- Moved [ProfileSearchRequest](#) and [ProfileSearchResultRecord](#) into this file from `cmsapi_1.proto` (relocation).
  - **New imports**
- Added import "Validation/constraints\_1.proto" to reference validation annotations (e.g., `cms_profile_id`).
  - **ProfileScope and profile-to-profile link improvements**
  - [ProfileScopeRequest](#)
- Replaced previous `profile_to_profile_links_request` type with a new [ProfileToProfileLinksRequest](#) (field 15).
- Added `profile_search_request` (field 25) to support profile-scoped searches.
  - **ProfileScopeResult**
- Added `profile_search_result` (field 17) to return profile search results.
  - **ProfileToProfileLinks messages**
  - [ProfileToProfileLinksRequest](#)
- New request with `profile_id` (validated via `cms_profile_id` constraint), optional `top`, and optional `linked_profiles_types` (enum constrained to `common_1.ProfileType`).
  - **ProfileToProfileLinksResult**
- Contains repeated links and an `is_complete` flag (default true).
- Extends `common_1.NamedEntity` using the `NamedEntity` extension slots to carry link attributes (number, `profile_type`, `is_removed`).
- **ProfileToProfileLink**
- Link structure used by the result; no structural breaking change noted.
  - **Profile search enhancements**
  - [ProfileSearchRequest](#)
- Expanded and moved here; `SearchCriteria` enum updated with new values: `PARENT_PROFILE_ID`, `IS_SELF_SERVICE`, `PARENT_CUSTOMER_PROFILE_ID`, [LEGAL\\_TYPE](#), `LINKED_PROFILE_ID`.
- Some prior values were renamed to `OBSOLETE_*` to mark moved semantics.
- Supports `search_options`, `all_match_mode`, `top`, `include_removed`.
  - **ProfileSearchResult & ProfileSearchResultRecord**
- [ProfileSearchResult](#) contains repeated [ProfileSearchResultRecord](#) and `is_complete` flag.
- [ProfileSearchResultRecord](#) now includes `parent_profiles` (repeated `common_1.NamedEntity`), `legal_type` (`common_1.LegalType`), and retains some deprecated fields as `obsolete_*`.
  - **Moved definitions (relocation)**
  - [ProfileSearchRequest](#) and [ProfileSearchResultRecord](#) were removed from `cmsapi_1.proto` and added here. This is a file relocation; message IDs and semantics are preserved.



- Breaking Changes:
  - The following is an actual breaking change for clients that import/ reference the moved messages from cmsapi\_1.proto:
    - [ProfileSearchRequest](#) and [ProfileSearchResultRecord](#) were removed from cmsapi\_1.proto and relocated to profile\_1.proto. Clients that import or refer to these messages from cmsapi\_1.proto must update their imports and generated bindings to reference profile\_1.proto.
- Details for enum/field changes:
  - Several SearchCriteria values were added and some prior criteria were renamed to OBSOLETE\_\*; if client code references criteria by enum value or ordinal, update to the new names/values. Verify numeric enum values in your generated bindings if you rely on numeric values rather than symbolic names.

#### *For Users:*

- Profile-search and profile-linking features gain richer metadata (parent\_profiles, [legal\\_type](#)) and clearer paging semantics (is\_complete). UI components should be able to present "show more" affordances and additional profile metadata. Overall functionality becomes more explicit and capable.

#### *For Developers:*

- Migration and action items
1. Update imports and generated bindings:
    - Change .proto imports to include profile\_1.proto where [ProfileSearchRequest](#) or [ProfileSearchResultRecord](#) are used. Regenerate code for all languages and update build configurations.
  2. Replace old profile-to-profile link request usage:
    - Replace previous common\_1.[EntityDataRequest](#) usages for profile\_to\_profile\_links with [ProfileToProfileLinksRequest](#) and populate profile\_id, optional top, linked\_profiles\_types appropriately.
  3. Handle is\_complete flags:
    - Respect [ProfileToProfileLinksResult.is\\_complete](#) and [ProfileSearchResult.is\\_complete](#) to determine whether paging/truncation occurred. If false, request further pages or surface "more results" in the UI.
  4. Update search criteria usage:



- Use the new SearchCriteria enum names (PARENT\_PROFILE\_ID, IS\_SELF\_SERVICE, LINKED\_PROFILE\_ID, etc.) and stop using OBSOLETE\_\* criteria. If you map criteria by numeric values, verify numeric assignments in the regenerated bindings.

#### 5. Validation constraints:

- [ProfileToProfileLinksRequest](#).profile\_id uses a cms\_profile\_id constraint (constraints\_1.field). If you perform client-side validation, ensure IDs conform to the expected format to reduce server-side validation errors.

#### 6. Extensions and code generation:

- Since ProfileToProfileLinksResult extends common\_1.NamedEntity (extension slots declared in common\_1.proto), regenerate language bindings and update generators to include extension handling. If your generator does not support extensions, ensure your code can still access required link fields through the primary result fields.

#### 7. Testing checklist:

- Verify profile search flows with new criteria and paging.
- Verify profile-to-profile link listing and is\_complete behavior when top is used.
- Confirm serialization/compatibility for moved messages after updating imports.

## 4. traderouting\_1.proto

This update adds validation annotations and introduces a mechanism for clients to indicate which fields should be cleared during account updates. It also marks one existing field as erasable and constrains its length.

- Added import of Validation/constraints\_1.proto to enable field-level annotations.
- Added cleared\_fields repeated uint32 field to Account to support explicit field erasure during updates.
- back\_office\_brokerage\_account\_number (field 25) now includes constraints\_1.field annotations indicating it is erasable and has string.max\_length = 256.
  - **Validation import**
- Imported Validation/constraints\_1.proto for annotation metadata.
  - **Account: cleared\_fields**
  - **New field**
- repeated uint32 cleared\_fields = 26
- Purpose: list of field IDs to be cleared during an update operation (partial-update semantics).
  - **Account: erasable annotation**
  - **back\_office\_brokerage\_account\_number**



- Remains field number 25.
- Annotated with `constraints_1.field`: tag "erasable" and `string.max_length = 256` to document/validate erasable and max-length semantics.

#### *For Users:*

- No immediate UI changes required. Users may gain the ability to clear certain account fields (for example, removing a brokerage account number) when clients expose this capability. Input for `back_office_brokerage_account_number` is constrained to 256 characters—attempts to set longer values will fail validation.

#### *For Developers:*

- Actionable technical guidance

##### 1. Partial-update behavior:

- Implement logic to honor `Account.cleared_fields`: when a field ID is present in `cleared_fields`, the server/client should clear/reset that field on the resource rather than leaving it unchanged. Ensure your mapping of numeric field IDs to semantic fields is correct (e.g., 25 = `back_office_brokerage_account_number`).

##### 2. Validation and constraints:

- Enforce `string.max_length = 256` for `back_office_brokerage_account_number` in client-side validation where applicable or rely on server-side validation to catch violations.
- If your validation pipeline consumes `constraints_1.field` annotations, update it to recognize the "erasable" tag.

##### 3. Serialization/compatibility:

- No existing field numeric renumbering occurred; `back_office_brokerage_account_number` remains 25 and `cleared_fields` occupies 26. Review any custom numeric-ID assumptions to avoid conflicts.

##### 4. Erase semantics & security:

- Implement clear/erase semantics so that a cleared field removes persisted data rather than being treated as missing. Ensure only authorized operations can include `cleared_fields` and that appropriate access control is enforced.

##### 5. Testing:

- Test update flows that include `cleared_fields` and confirm that the intended fields are cleared and validation behaves as expected.



## 5. shared\_1.proto

This update is a documentation/cleanup change: a stray comment was removed and duplicative deprecation commentary on the Text.format field was cleaned up. No structural or numeric changes were made.

- Removed an extraneous top-of-file comment and a duplicated inline deprecation comment.
- The Text.message format field remains present as optional string format = 3 and remains marked deprecated = true.

### *For Users:*

- No user-visible changes. Localization and message-format behavior remain unchanged.

### *For Developers:*

- No code changes required for binary compatibility: field numbers and types did not change.

### Notes and general recommendations

- Where messages or fields were relocated (moved between files) but retained their message/field IDs and semantics, these relocations are not breaking at the wire format level; they are breaking only in build/import references. Update .proto import statements and regenerate language-specific bindings.
- The only file-level breaking change in this set is the relocation of [ProfileSearchRequest](#) and [ProfileSearchResultRecord](#) from cmsapi\_1.proto to profile\_1.proto. No field renumbering was described for these moved messages; nevertheless, update imports and regenerate code to ensure namespace and packaging correctness.

## CMS 2.14 (Protocol Version: 1.42)

- Summary of Protocol Changes (CMS 2.14, November, 2025)
  - Below is a summary of key functional changes in the protocol files that affect users. The focus is on changes that impact user interaction with the system, with special attention to breaking changes that require code updates



## 1. cmsapi\_1.proto

- This update introduces a minor protocol version of increment and adds a new optional field to enhance the ObtainDemoLogin message with additional client information. It also updates validation rules for phone numbers.

### *Protocol Version Update:*

- Incremented `PROTOCOL_VERSION_MINOR` from 41 to 42 to reflect backward-compatible changes.

### *New Fields:*

- Added `client_ip_address` as an optional string field in the ObtainDemoLogin message to allow trusted clients to specify the real client IP address they act on behalf of.

### *Enhanced Validation:*

- Updated the `phone_number` option in ObtainDemoLogin to include a `protocol_version` constraint: ``protocol_version : ">=1.42.0" ``.

### *Impact on Users*

- **For Users:** No direct impact on user interactions. The system now supports additional client IP information for trusted clients, which may improve service accuracy.
- **For Developers:** Ensure that any integrations using the ObtainDemoLogin message can handle the new `client_ip_address` field if applicable. Update validation logic to accommodate the new protocol version constraint for phone numbers.

## 2. common\_1.proto

- This update enhances address and contact information management by introducing erasable fields and a new enum for address validation status.

### *Enhanced Validation:*

- Added `protocol_version` constraints for `state_code` and `zip_code` options to ensure version-specific validation.

### *Erasable Fields:*

- Marked multiple fields in Address and ContactInformation as erasable to support simplified profiles. This includes fields like `country`, `city`, `zip`, `address`, `email`, `phone`, `fax`, and `address`.

### *Impact on Users*

- **For Users:** No direct impact on login or authentication workflows. Password policies will be managed through alternative mechanisms.



- **For Developers:** IMMEDIATE ACTION REQUIRED:
  - Remove all code referencing the PasswordPolicy message.
  - Refactor password policy display/configuration logic.
  - Update data models and API contracts.
  - Check for serialized data containing PasswordPolicy.
  - Update unit and integration tests.

### 3. login\_1.proto

- This update introduces new messages and extends existing risk parameter levels to enhance login scope management and risk parameter configuration.

#### *Extended Risk Parameters:*

- Added new optional fields for various risk parameter levels, such as `all_product_group_position_limit_multiplier`, `commodity_long_option_position_limit_multiplier`, and `commodity_short_option_position_limit_multiplier`, marked as erasable.

#### *New Message:*

- Introduced `LoginPasswordSet` message to allow direct password setting under specific security policies.

#### *Impact on Users*

- **For Users:** Users may experience enhanced login management capabilities, including more flexible risk parameter settings and direct password updates under certain conditions.
- **For Developers:** Update client applications to support the new `LoginScopeRequest` and `LoginScopeResult` messages. Implement logic to handle new risk parameter fields and ensure compliance with security policies when using `LoginPasswordSet`.

### 4. metadata\_1.proto

- This update introduces new request and result messages to enhance metadata retrieval capabilities, specifically focusing on product groups and metadata exchanges. It also includes a breaking change with the removal of a message.

#### *New Fields:*

- Added `metadata_exchange_list_request` and `product_group_list_request` as optional fields in `MetadataScopeRequest` to request metadata exchanges and product groups.
- Introduced `metadata_exchange_list_result` and `product_group_result` as optional fields in `MetadataScopeResult` to return lists of metadata exchanges and product groups.



- Added `product_group_ids` to `ExchangeGroup` and `product_group_id` to `FungibleCommodity` to associate commodities with product groups.

#### *New Messages:*

- Introduced `MetadataExchangesListResult` and `MetadataExchange` to handle metadata exchange information.
- Added `ProductGroupResult` and `ProductGroup` to manage product group data, including trade routing system identifiers and names.

#### *Breaking Changes:*

- **BREAKING CHANGE:** Removed `CommodityInstrumentTypesRequest` message. Clients using this message must refactor to use `common_1.DataRequest`` for similar functionality.

#### *Impact on Users*

- **For Users:** Users can now request and view detailed lists of metadata exchanges and product groups directly through the system interface. This may result in new options or filters in user interfaces where metadata is displayed.
- **For Developers:** Update client applications to replace any usage of `CommodityInstrumentTypesRequest` with `common_1.DataRequest``. Ensure that new fields and messages are integrated into existing data retrieval and processing workflows. Review and update any serialization logic to accommodate new message structures.

## 5. `risk_parameter_value_1.proto`

- This update introduces a new message to handle risk parameter values based on position size, enhancing the flexibility of risk management configurations.

#### *New Message:*

- Added `RiskParameterDoublePerPositionSize` to define risk parameter values applicable to specific position sizes, including fields for `position_size` and `value`.

#### *Impact on Users*

- **For Users:** No direct impact on user interfaces. The changes are backend-focused, enhancing the precision of risk parameter configurations.
- **For Developers:** Integrate the new `RiskParameterDoublePerPositionSize` message into risk management systems where position size-specific risk parameters are required. Ensure that any logic handling risk parameters is updated to consider this new message type.

## 6. `risk_parameters_1.proto`

- This update expands the risk parameter framework to include product group levels and position size-specific margin multipliers, providing more granular control over risk settings.



#### *New Fields:*

- Added `product_group_levels` to `RiskParameters` to manage risk parameters at the product group level.
- Introduced `position_sized_margin_multipliers` and `position_sized_eth_margin_multipliers` across various levels (e.g., `AllCommoditiesLevel`, `UsCommoditiesLevel`) to apply margin multipliers based on position size.
- Added `commodity_long_option_position_limit` and `commodity_short_option_position_limit` to manage optional position limits for commodities.

#### *New Message:*

- Introduced `ProductGroupLevel` to define risk parameters specific to product groups, including fields for `product_group_id` and `product_group_position_limit`.

#### *Impact on Users*

- **For Users:** Users may experience more tailored risk management settings, potentially affecting trading limits and margin requirements based on product groups and position sizes.
- **For Developers:** Implement the new `ProductGroupLevel` and position size-specific fields in risk management systems. Update any existing logic to accommodate these new configurations, ensuring that risk parameters are correctly applied at the appropriate levels. Review and adjust any data models or API contracts to incorporate these changes.

## 7. `traderouting_1.proto`

- This update enhances the trade routing protocol by introducing new fields and enums to support MiFID compliance and cross-trade control, as well as reordering existing enum values.

#### *New Fields:*

- Added `mifid_info` to `AccountAndUserInfo` for MiFID-related information.
- Added `mifid_trading_venue_transaction_id` to `OrderFill` and `LegFill` for MiFID transaction identification.
- Added `back_office_brokerage_account_number` to `Account` for back office processing.
- Added `cross_trade_control_type` and `include_synthetic_strategy_legs` to `AccountRouteRecord` for cross-trade control.
- Added `allow_cross_trade_control` to `RouteInformation` to indicate cross-trade control permissions.



### *New Enums:*

- Introduced [CrossTradeControlType](#) with values CROSS\_TRADE\_CONTROL\_TYPE\_NONE, CROSS\_TRADE\_CONTROL\_TYPE\_REJECT\_ORDER, and CROSS\_TRADE\_CONTROL\_TYPE\_CANCEL\_ORDERS.

### *Breaking Changes:*

- **BREAKING CHANGE:** Enum value reordering in OrderTransactionStatus. TRADE\_AMENDED changed from 28 to 29, TRADE\_BROKEN changed from 29 to 28. Clients using numeric enum values must update to avoid incorrect interpretation.

### *New Enum Values:*

- Added REJECTED\_BY\_CQG to both OrderChain and OrderTransactionStatus enums.
- Added FILLS\_ALLOCATION to OrderTransactionStatus.

### *Impact on Users*

- **For Users:** Users will not experience immediate changes in the interface or workflows. However, the system will now handle additional MiFID compliance data and cross-trade controls.
- **For Developers:** IMMEDIATE ACTION REQUIRED:
- Update any code that uses numeric values for TRADE\_AMENDED and TRADE\_BROKEN to reflect the new enum mappings.
- Ensure integration with MiFID-related fields (mifid\_info, mifid\_trading\_venue\_transaction\_id) is correctly implemented.
- Review and update any logic related to cross-trade control using the new [CrossTradeControlType](#) enum.
- Update serialized data and integration tests to accommodate these changes.

## CMS 2.13 (Protocol Version: 1.41)

- Summary of Protocol Changes (CMS 2.13)
  - Below is a summary of key functional changes in the protocol files that impact users. Minor edits, such as spelling corrections, are excluded for clarity. This summary highlights changes affecting user interaction with the system.

### 1. api\_limit\_1.proto

#### *Changes*

- **New Import:** Added Validation/constraints\_1.proto to enable advanced validation rules.
- **Enhanced Validation for ApiLimitRequestEntry:**



- `limit_type`: Now explicitly tied to `.api_limit_1.ApiLimitType` enum and marked required with `(constraints_1.field)` validation.
- `entity_name`: Enforces ASCII string format for compatibility.

#### *Impact on Users*

- **For Users:** Limit type requests now require valid enum values, which may necessitate updates to input forms for compliance.
- **For Developers:** Update code to handle new validation constraints, ensuring `limit_type` uses the correct enum and `entity_name` is ASCII-compliant.

## 2. cmsapi\_1.proto

#### *Changes*

- **Protocol Version Update:** `PROTOCOL_VERSION_MINOR` incremented to 41 (from 40).
- **New Import:** Added `Validation/constraints_1.proto` for stricter field validation.
- **Deprecated Features:**
  - Messages like `SendWelcomeEmail` and `SendCredentialResetEmail` remain deprecated; use equivalents in `profile_1.proto`.
- **New Validation Rules:**
  - Added `one_of` constraints for messages like `SearchRequest` and `AdminRequest` to ensure only one optional "request" field is set.
  - Added `required_if` constraints to enforce `profile_id` over obsolete `customer_id` in messages like `SendWelcomeEmail` and `SendCredentialResetEmail`.
  - Fields like `login_id` and `profile_id` now use `cms_login_id` and `cms_profile_id` validations with specific expected types (e.g., `CMS_LOGIN_TRADE_ROUTING`, `CMS_PROFILE_CUSTOMER`).

#### *Impact on Users*

- **For Users:** Stricter input validation (e.g., ASCII-only fields, specific login/profile types) enhances data integrity but may increase form submission errors if inputs are non-compliant.
- **For Developers:** Update client applications for version 1.41 compatibility, handle new validation rules, and transition to `profile_1.proto` for deprecated messages.

## 3. common\_1.proto

#### *Changes*

- **New Import:** Added `Validation/constraints_1.proto` for field validation.
- **Enhanced Validation Rules:**
  - Fields like `user_name`, `password`, `client_app_id`, `client_version`, `private_label`, and `access_token` now enforce ASCII string format.
  - `protocol_version_minor` and `protocol_version_major` marked as required.



- Fields like user\_id, login\_id, and profile\_id use cms\_login\_id or cms\_profile\_id validations with specific expected types.
- Added required\_if constraints for LoginRequest to enforce access\_token usage when other fields are absent.
- Added erasable constraints for User message fields like user\_name.

#### *Impact on Users*

- **For Users:** ASCII-compliant inputs are required for login and profile fields, potentially requiring updates to data entry processes.
- **For Admins:** Enhanced control over user data with erasable fields and specific type validations improves management but demands adherence to new constraints.
- **For Developers:** Update code to comply with ASCII string requirements and new required\_if/erasable constraints.

## 4. location\_1.proto

### *Changes*

- **New Import:** Added Validation/constraints\_1.proto.
- **New Validation Rules:**
  - Added one\_of constraint to ensure only one "request" field in LocationRequest.
  - brokerage\_ids field enforces ASCII string format.

### *Impact on Users*

- **For Users:** Connection point requests require ASCII-compliant brokerage\_ids, potentially affecting input processes.
- **For Developers:** Ensure code handles new one\_of constraints and ASCII validation for brokerage\_ids.

## 5. login\_1.proto

### *Changes*

- **New Import:** Added Validation/constraints\_1.proto.
- **Enhanced Validation:**
  - brokerage\_id and billing\_brokerage\_id enforce ASCII format.
  - entitlement\_service\_id marked as required.
  - login\_id fields use cms\_login\_id validation with specific expected login types.
  - Added required\_if constraint for UpdateEntitlementsRequest to enforce entitlement\_service\_to\_setusage.

### *Impact on Users*

- **For Users:** Service entitlement updates require precise login ID formats, which may impact form submissions.
- **For Developers:** Update code to handle new validation rules and required fields.



## 6. metadata\_1.proto

### *Changes*

- **New Import:** Added Validation/constraints\_1.proto.
- **New Validation Rules:**
  - Added one\_of constraint to ensure only one "request" field in MetadataRequest.

### *Impact on Users*

- **For Users:** Contract and instrument type searches enforce single-request constraints, simplifying submissions but requiring precise inputs.
- **For Developers:** Ensure code respects one\_of constraints for request fields.

## 7. route\_1.proto

### *Changes*

- **New Import:** Added Validation/constraints\_1.proto.
- **New Validation Rules:**
  - Added one\_of constraint to ensure only one "request" field in RoutesRequest.
  - route\_codes and route\_group\_ids enforce ASCII string format.

### *Impact on Users*

- **For Users:** Route-related requests require ASCII-compliant inputs, potentially affecting data entry.
- **For Developers:** Update code to handle new validation and one\_of constraints.

## 8. session\_context\_1.proto

- *Changes*
- **New Import:** Added Validation/constraints\_1.proto.
- **New Validation Rules:**
  - Added one\_of constraint to ensure only one "request" field in SessionContextRequest.

### *Impact on Users*

- **For Users:** Session context requests enforce single-request constraints, simplifying interactions.
- **For Developers:** Ensure code respects one\_of constraints for session context requests.

## 9. shared\_1.proto

### *Changes*

- **Updated Comments:** Improved clarity in comments (e.g., "Original order is sent to the execution system" instead of "execution system").



- **New Enum Value:** Added FILLS\_ALLOCATION to OrderStatus for handling allocation order fills and cancellations.
- **Updated ResultCode Range:** Success codes range adjusted to 0–99 (from 0–100).

#### *Impact on Users*

- **For Users:** The new FILLS\_ALLOCATION status provides better visibility into allocation order processes.
- **For Developers:** Update code to handle the new FILLS\_ALLOCATION status and adjusted ResultCode range.

## CMS 2.12 (Protocol Version: 1.40)

- Summary of Protocol Changes (CMS 2.12)
  - Below is a summary of key functional changes in the protocol files that affect users. The focus is on changes that impact user interaction with the system, with minor edits (e.g., spelling fixes) omitted for clarity.
  - **[Important note]** We have added proto file level validation rules. They are disabled during the transfer period, but the rules will be applied starting from July, 18th 2025, please, check the messages used by your applications according to the validation rules in Validation/constraints\_1.proto.

## Overall Impact

- **Improved Data Accuracy:** Strict validation for addresses, contacts, and demo access enhances data quality but requires users to be more precise when entering information.
- **Greater Control:** Trading hours configuration and flexible cloning of users/accounts give admins more management options.
- **Enhanced Trading:** New margin parameters and complex strategy support make trading more flexible but may require adjustments to existing setups.
- **Migration to New Features:** Deprecated messages and fields (e.g., email-sending functions) necessitate code updates to ensure future compatibility.
- **Simplified Authentication:** SSO support streamlines login for users with external systems.

## 1. cmsapi\_1.proto

### *Changes*

- **Protocol Version Update:** Protocol version updated to 1.40 (from 1.39), with PROTOCOL\_VERSION\_MINOR incremented to 40.



- **New Validation Rules for Demo Access** (ObtainDemoLogin):
  - Fields first\_name, last\_name, email, and phone are now mandatory with strict checks:
    - First name: Up to 20 characters, valid name characters only.
    - Last name: Up to 25 characters.
    - Email: Must comply with RFC 5322.
    - Phone: Validated with country code.
  - Field product\_id must be convertible to TYPE\_INT32.
- **Deprecated Features:**
  - Messages send\_welcome\_email and send\_credential\_reset\_email are marked as deprecated ([deprecated=true]). Use equivalents in profile\_1.proto instead.
  - Results send\_welcome\_email\_result and send\_credential\_reset\_email\_result are also deprecated.
- **New Import:** Added Validation/constraints\_1.proto to support new validation rules.

### *Impact on Users*

- **For Users:** Requesting demo access now requires entering data that adheres to strict rules (e.g., valid email and phone formats). This improves data quality but may demand more attention during form submission.
- **For Developers:** Update code to use new messages from profile\_1.proto for sending emails. Compatibility checks for version 1.40 may be needed for client applications.

## 2. common\_1.proto

### *Changes*

- **New Validation Rules:**
  - For Address: Fields country (ISO 3166-1, 2 characters), state, city (up to 25 characters), zip (up to 9 characters), address, and address\_2 (up to 58 characters, Latin characters only) now have strict validation.
  - For ContactInformation: Fields first\_name (up to 20 characters), last\_name (up to 25 characters), and email (RFC 5322) are validated. Phone numbers are tied to country codes.
- **Flexible User Cloning** (CloneUser):



- Added SettingType enum (MARKET\_RISK\_PARAMETERS, RISK\_SETTINGS, PRODUCTS\_AND\_SERVICES) and setting\_types field to specify which settings to copy during cloning. If not provided, all settings will be copied.
- **Trading Hours Management:**
  - New TradingTimes message allows setting regular (rth\_start\_time) and extended (eth\_start\_time) trading hours for each day of the week (per ISO 8601).
- **Single Sign-On Support:**
  - Added SSO\_LOGIN to AuthActivityType for Single Sign-On (SSO) authentication.
- **New Import:** Added Validation/constraints\_1.proto.

#### *Impact on Users*

- **For Users:** Entering address and contact information now requires adherence to strict formats (e.g., Latin-only addresses, valid email). This ensures better data but may complicate form filling.
- **For Admins:** Configuring trading hours provides greater control over trading schedules. User cloning is now more customizable with selective setting copying.
- **For SSO Users:** SSO support simplifies login via external systems, enhancing user experience.

### 3. profile\_1.proto

#### *Changes*

- **New Email-Sending Messages:**
  - SendWelcomeEmail: Sends welcome emails for customer profiles (requires profile\_id, login\_id).
  - SendCredentialResetEmail: Handles password resets for customer profiles (includes operation field for special operations).
  - SendLoginNameEmail: Sends login name emails for customer profiles.
- **Trading Hours Management:**
  - Added trading\_times\_request and trading\_times\_set to ProfileScopeRequest for requesting and setting trading hours.
  - New messages: ProfileTradingTimesResult, ProfileTradingTimes, and ProfileTradingTimesSet for configuring time zones and trading schedules by day. Available to users with related new permissions.
- **Improved Risk Parameter Requests:**



- In ProfileBrokerageClassRiskParametersRequest, added profile\_brokerage\_id as an alternative to brokerage\_class (now optional).

#### *Impact on Users*

- **For Users:** New email-sending features improve system interaction (e.g., receiving login names or resetting passwords).
- **For Admins:** Configuring trading hours for brokerage profiles allows precise control over trading windows, including time zones.
- **For Developers:** Using profile\_brokerage\_id simplifies risk parameter requests, potentially streamlining integration.

## 4. risk\_parameters\_1.proto

#### *Changes*

- **Refined Margin Parameters:**
  - Fields margin\_multiplier, ms\_instrument\_group\_margin\_multiplier, and max\_purchasing\_power now require a maximum of 3 decimal places (e.g., 0.001 to 100 for margins).
- **New Risk Parameters:**
  - eth\_margin\_multiplier: Margin multiplier for extended trading hours (applies if trading hours are configured).
  - liquidation\_only\_after\_fn\_date: Allows only position liquidation after the FN date for contracts (across AllCommoditiesLevel, UsCommoditiesLevel, NonUsCommoditiesLevel, ExchangeLevel, FungibleCommodityLevel). Supported for account, broker, sales series. Not available to Order Handlers.
  - ms\_flat\_margin in PurchasingPowerParameters: If set, only available margin (Flat Margin minus margin of open positions/orders) is checked for opening positions.

#### *Impact on Users*

- **For Traders:** The 3-decimal-place limit requires adjusting existing margin settings to comply with new rules;
  - liquidation\_only\_after\_fn\_date restricts trading after specific dates, potentially affecting strategies;
  - ms\_flat\_margin simplifies margin checks, benefiting high-volume trading.
- **For Brokers:** eth\_margin\_multiplier enables flexible margin settings for extended trading hours but requires trading hours configuration.



## 5. traderouting\_1.proto

### *Changes*

- **Flexible Account Cloning** (CloneAccount):
  - Added SettingType enum (MARKET\_RISK\_PARAMETERS, RISK\_PARAMETERS, ROUTING\_SETUP, AUTHORIZED\_LOGINS, ADDITIONAL\_IDENTIFIERS, TYPE\_AND\_ORIGIN\_OVERRIDES, SERVICE\_GROUPS) and excluded\_setting\_types field to exclude specific settings during cloning. Fields excluded from cloning will have values as for newly created account. If ROUTING\_SETUP is excluded from cloning, but MARKET\_RISK\_PARAMETERS is not excluded, then only market risk parameters of levels ALL/US/Non US will be cloned.
- **Support for Complex Strategies:**
  - Added nested\_legs to OrderContractLeg for nested strategies in orders.
  - In LegFill, leg\_number is deprecated, replaced by leg\_path to indicate the path in the strategy tree (e.g., "/1/2").
- **Enhanced Account Information:**
  - Added brokerage\_name and brokerage\_id to AccountAndUserInfo for broker identification.
- **Deprecated Feature:**
  - Field externally\_provided\_execution in AccountSettings is deprecated ([deprecated=true]).

### *Impact on Users*

- **For Admins:** Account cloning is more customizable with the ability to exclude specific settings.
- **For Traders:** nested\_legs support enables complex trading strategies with nested orders, ideal for advanced users.
- **For Users:** brokerage\_name and brokerage\_id improve broker transparency in trading operations.
- **For Developers:** Transition to leg\_path from leg\_number is required, and usage of externally\_provided\_execution should be reviewed.

## CMS 2.11 (Protocol Version: 1.39)

- No changes since CMS 2.10